Finite element simulation of the aluminum extrusion process : shape prediction for complex profiles

Citation for published version (APA):
Rens, van, B. J. E. (1999). *Finite element simulation of the aluminum extrusion process : shape prediction for complex profiles*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mechanical Engineering]. Technische Universiteit Eindhoven. https://doi.org/10.6100/IR527121

DOI:
10.6100/IR527121

Document status and date:
Published: 01/01/1999

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 16. Nov. 2023

Finite Element Simulation
# of the
Aluminum Extrusion Process

Shape prediction for complex profiles

This research project was carried out for Alcoa Europe, Extrusions & End Products, location Drunen, The Netherlands and was partly funded through a PBTS grant. The support is gratefully acknowledged.

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Rens, Bas J.E. van

Finite element simulation of the aluminum extrusion process :
shape prediction for complex profiles / by Bas J.E. van Rens. - Eindhoven :
Technische Universiteit Eindhoven, 1999. - Proefschrift.
ISBN 90-386-2731-9
NUGI 841
Trefwoorden: extrusie / aluminium / eindige elementenmethode / vrije oppervlakken ;
vormvoorspelling / mesh-generatie ; richtingsafhankelijke verfijning / wrijving ; Coulomb /
vaste stof-vloeistof interactie ; gereedschapsdeformatie
Subject headings: extrusion / aluminum / finite element method / free surfaces ; shape
prediction / mesh generation ; directional refinement / friction ; Coulomb / solid-fluid
interaction; tooling deformation

Cover : Desiree van Doorne Grafische Vormgeving, Mierlo, The Netherlands
Reproduction : Universiteitsdrukkerij TU Eindhoven, Eindhoven, The Netherlands

Finite Element Simulation of the Aluminum Extrusion Process

Shape prediction for complex profiles

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Eindhoven,
op gezag van de Rector Magnificus, prof.dr. M. Rem,
voor een commissie aangewezen door het College voor Promoties
in het openbaar te verdedigen op
woensdag 24 november 1999 om 16.00 uur

door

Bas Jan Emile van Rens

geboren te Eindhoven

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. F.P.T. Baaijens
en
prof.dr.ir. H.E.H. Meijer

Copromotor:
dr.ir. W.A.M. Brekelmans

Voor Anne-Marie

Contents

Summary **xi**

Notations **xiii**

1 Introduction **1**
1.1 Practical aspects of aluminum extrusion 1
1.2 Finite element modeling of extrusion 3
1.3 Scope of the thesis 4

2 Mathematical modeling of extrusion **7**
2.1 Conservation laws 8
2.2 Constitutive equations 8
2.3 Flow problem 9
2.4 Mechanical boundary conditions 12
  2.4.1 Friction 12
  2.4.2 Free surface 14
2.5 Thermal problem 14
2.6 Thermal boundary conditions 15
2.7 Changing aluminum boundary 16
  2.7.1 Three field approach 16
  2.7.2 Structure problem - Free surface shape 18
  2.7.3 Structure problem - Die shape 20

3 Finite element implementation **21**
3.1 Discretizing the Stokes problem 21
  3.1.1 MINI element 23
  3.1.2 Choice of the bubble shape 25
  3.1.3 Non-constant viscosity 28
  3.1.4 Friction 29
3.2 Discretizing the temperature problem 35
  3.2.1 Streamline Upwind / Petrov-Galerkin method 36
3.3 Discretizing the changing boundary problems 37
  3.3.1 Structure problem - Free surface 37
  3.3.2 Structure problem - Die deflection 39

3.3.3 Mesh problem . . . 40
3.4 Solution procedure . . . 41
3.4.1 Staggered approach . . . 41
3.4.2 Linear solver . . . 42

4 Mesh generation 47
4.1 Meshing the surfaces - Paving generator . . . 48
4.1.1 Input data . . . 50
4.1.2 Generating triangles with the current boundary nodes . . . 50
4.1.3 Adding a layer of elements . . . 51
4.1.4 Merging nodes on the current boundary . . . 51
4.1.5 Smoothing the mesh . . . 52
4.1.6 Resulting meshes . . . 53
4.2 Meshing the surfaces - Expansion generator . . . 54
4.3 Meshing the volumes - Expansion generator . . . 55
4.3.1 Generating prisms . . . 55
4.3.2 Generating tetrahedra . . . 55
4.4 Meshing the volumes - Unstructured generator . . . 57
4.4.1 Input data . . . 57
4.4.2 Generate elements with the current boundary nodes . . . 58
4.4.3 Adding a node to generate a tetrahedron . . . 58
4.4.4 Applying the Delaunay algorithm . . . 59
4.4.5 Smoothing the mesh . . . 59
4.4.6 Deleting unacceptable elements . . . 59
4.4.7 Resulting mesh of the die package . . . 60
4.5 Stretching the bearing . . . 61
4.6 Extension to hollow dies . . . 61

5 Evaluation of the finite element model 67
5.1 Relevance of the separate problems . . . 67
5.2 Mesh sensitivity . . . 71
5.2.1 Reference mesh . . . 72
5.2.2 Mesh coarseness . . . 73
5.2.3 Orientation of the envelope mesh . . . 75
5.3 Experimental validation . . . 76

6 Extrusion of complex profiles 81
6.1 Flat profile . . . 81
6.2 Hollow profile . . . 84
6.3 Polymer extrusion . . . 90

7 Conclusions and recommendations 99
7.1 Conclusions . . . 99
7.2 Recommendations . . . 101
7.2.1 Experimental validation . . . 101

7.2.2 Exploitation of the solution fields . . . 101
7.2.3 Parallelization . . . 103

Bibliography **112**

Samenvatting **113**

Dankwoord **115**

Curriculum vitae **116**

Summary

Aluminum extrusion is a bulk forming process in which a hot aluminum billet is placed in a container and subsequently pressed through a die. The die plays a crucial role in this process because it ensures that the flow of the aluminum is balanced, such that the profile meets the requirements e.g. concerning dimensions and straightness of the extruded profile. In the current engineering practice the design of extrusion dies is primarily a trial and error based process. Trial-pressings, used to assess the performance of a die, are an integral part of this process. However, trial-pressings are time consuming and expensive and the translation of errors in the extruded profile to adjustments of the die design is complicated and, generally, requires years of learning experience. Therefore, the objective of this thesis is to develop a finite element model that can, at least partly, replace these trial-pressings by numerical simulations.

The aluminum extrusion process is modeled as a Stokes flow problem, coupled to a thermal convection-diffusion problem. The flow problem is formulated in an Eulerian, spatially fixed, reference frame which is well suited to describe processes in which large deformations occur, such as extrusion. The Eulerian approach, however, requires special attention in case the material domain changes. This occurs at the outflow of the die where the aluminum is not constrained by the tooling and can deform due to flow imbalances. It also occurs at the aluminum-die interface where the die deforms due to the extrusion pressures. The deforming aluminum domain is dealt with by employing a three field approach. In this approach the shape of the profile at the outflow of the die and the shape of the die are computed separately. The effect of changes in the profile shape and/or the die shape on the aluminum flow is accounted for by adapting the (aluminum) mesh.

The constitutive behavior of aluminum is modeled as a non-Newtonian fluid of which the viscosity depends on the temperature and the shear rate. Furthermore, the frictional behavior between aluminum and tooling is described by adopting Coulomb's law of friction.

The flow problem is discretized using the MINI element which enables the application of powerful tetrahedral mesh generators while a minimal number of unknowns is introduced for each element in the mesh. Additionally, the Lagrange multipliers that incorporate the effect of the frictional surface tractions on the aluminum flow, are discretized according to a new method that makes it possible to obtain accurate results near sharp edges on relatively coarse meshes. Finally, the thermal problem is discretized using linear tetrahedral elements that are stabilized using the SUPG (Streamline Upwinding / Petrov-Galerkin) method.

The domains associated with the aluminum and the die package are complex. As a result mesh generation poses a serious problem in the analysis of aluminum extrusion. To overcome this problem, new meshing algorithms have been developed that produce directionally refined meshes on which gradients in the solution fields can be captured accurately while a minimal number of elements is created. Moreover, these algorithms contain a number of post-processing steps to incorporate the varying bearing length and the complex geometries of dies for hollow profiles. The newly developed method of generating meshes makes it possible to create a mesh for any die (design) with only a minimal effort from the user.

Based on the comparison of various numerical experiments it is concluded that it is important to incorporate the temperature dependence of the viscosity, as well as the shape of the outflow surface, and occasionally the deflection of the die. It can also be observed that the value that is assigned to the (Coulomb) friction coefficient has a substantial influence on both the flow and the temperature fields of the aluminum. Comparison with industrially obtained experimental data shows that the model predicts the flow field and the resulting profile shape accurately, even for very complex profiles.

In conclusion the model that has been developed in this thesis is able to predict the performance of a complex extrusion die accurately, within time spans that can definitely compete with the time necessary to perform a trial-pressing.

Notations

Entities

a	scalar
$\vec{a}$	vector
$\boldsymbol{A}$	tensor
$\boldsymbol{I}$	identity tensor
$\underline{\underline{A}}$	matrix
$\underline{a}$	column

Operators

$\vec{\nabla} a$	gradient
$\underline{\underline{A}}^t$	transpose
$\underline{\underline{A}}^{-1}$	inverse
$\boldsymbol{A}^c$	conjugate
$\mathrm{tr}(\boldsymbol{A})$	trace
$\det(\boldsymbol{A})$	determinant
$\boldsymbol{A}^d$	deviator, $\boldsymbol{A}^d = \boldsymbol{A} - \frac{1}{3}\mathrm{tr}(\boldsymbol{A})\boldsymbol{I}$
$\boldsymbol{A} \cdot \boldsymbol{B}$	inner product
$\boldsymbol{A} : \boldsymbol{B}$	double innner product

Chapter 1

Introduction

Aluminum extrusion is a bulk forming process used to manufacture a great diversity of aluminum profiles. Since there are almost no constraints on the cross-sectional shape of these profiles, extrusion allows the incorporation of a high level of functionality into one profile. This makes extruded aluminum profiles increasingly popular with designers. For example, the profile of a mast fulfills the functions of a light structure that supports the sail, a rail through which the sail can be guided, a surface that is esthetically pleasing and a shape that is aerodynamical. Each functionality requires that the profile meets certain tolerances with respect to the cross-sectional shape and/or metallurgical state. The current trend in profile design is to incorporate an ever higher level of functionality into one profile which results in ever decreasing tolerance ranges for the profile. This poses a considerable challenge on the design of the aluminum extrusion tools.

In Section 1.1 the extrusion process is explained. The importance of the die in this process is elucidated along with the drawbacks that are inherently associated with the trial and error process usually involved in die design. In Section 1.2 the benefits of using finite element simulations of extrusion to assist in the die design process are indicated. Also, the shortcomings of the finite element analyses that can be found in the literature are identified and are used as a basis to formulate the goals for this thesis. Finally, in Section 1.3 an outline of the rest of the thesis is given.

1.1 Practical aspects of aluminum extrusion

Aluminum extrusion is a process in which a hot billet is placed in a container and pressed through a die. The objective of this process is to obtain a straight profile with a cross-sectional shape that meets the specifications defined by the customer. A profile that conforms to the specifications can only be produced if the aluminum velocities at the exit of the die are balanced. The exit velocities are balanced when the velocity component in extrusion direction is constant over the profile cross-section and when the components perpendicular to the extrusion direction are zero. The first requirement ensures that the profile is straight, whereas the second requirement ensures that the profile complies to the predefined cross-sectional shape.

It is obvious that the geometry of the die has a dominant influence on the balancing of the exit flow and thus on the shape of the profile.

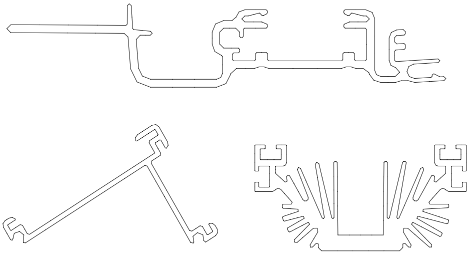


Figure 1.1: Examples of flat profiles.

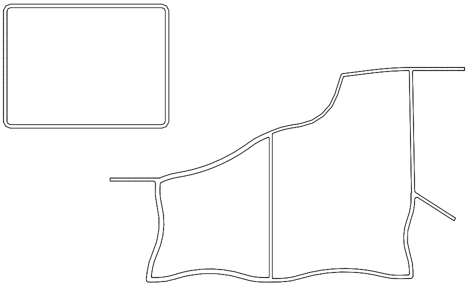


Figure 1.2: Examples of hollow profiles.

Extrusion dies can be classified into four groups with increasing levels of complexity. The specification of the groups is closely related to the type of profile that the die produces. The first group encompasses single opening dies that produce one ’flat’ profile out of each billet. Flat profiles do not contain enclosed hollow sections (see Figure 1.1). The second group contains multiple opening dies of which each opening renders a separate flat profile. Typically, these dies generate two to six profiles out of each billet. The third group contains single opening dies designed for hollow profiles. Some examples of hollow profiles are plotted in Figure 1.2. The last group is made up out of multiple opening dies for hollow profiles. The die for a flat profile consists of a die plate that contains a pocket and a bearing (see Figure 1.3). The die for a hollow profile consists of a more complicated die plate that also contains a welding chamber, and of an additional bridge-part (Figure 1.4).

The increasing level of complexity between the four groups is based on the following reasoning. For a flat profile die, only the pocket and the bearing have to be designed in such a manner that a balanced exit flow is obtained. If the die has multiple openings this is more complicated, because the separate profiles then have to be mutually balanced as well. Also, when the profile is hollow not only the design of the die plate is more complicated because the shape of the welding chamber has to be devised, but the shape of the additional bridge-part has to be defined as well.

In current engineering practice, die design is a process that is based on trial and error. When designing a flat profile die, the pocket shape and the bearing shape have to be defined. The pocket shape is given by the pocket contour and the pocket length, the bearing shape by the bearing contour and the bearing length. Moreover, the bearing length can vary along the bearing contour for flow balancing purposes. The effect of, for instance, local variations in the bearing length on the balancing of the aluminum exit flow cannot be fully predicted a priori. Therefore, a die is designed according to the experience of the designer and/or by using expert systems. After design, the die is manufactured and, subsequently, tested in a trial-pressing. If the die performs satisfactorily it is ready for production, otherwise it is corrected, or, if the distortions of the profile are too severe, it is redesigned. Subsequent to the correction or the redesign and remanufacturing, the die is tested again. This process is repeated until a die that produces acceptable profiles is obtained. For some dies a considerable

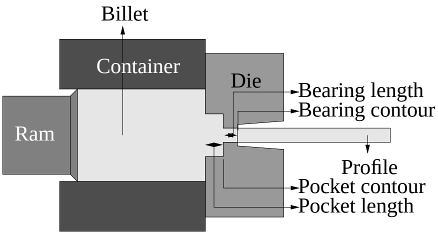


Figure 1.3: Die for flat profiles.

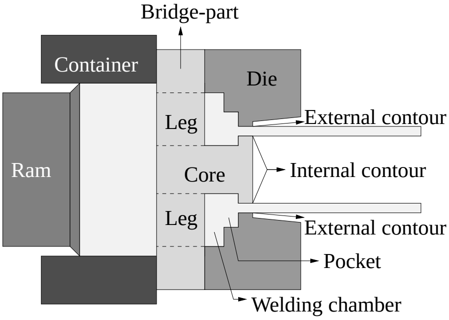


Figure 1.4: Die for hollow profiles.

number of these iterations are required before a satisfactory die design has been obtained. The number of iterations that is actually required depends highly on the level of complexity of the profile. Since the complexity of the profiles is consistently increasing and the tolerances on the profiles are constantly narrowing, the number of necessary trial-pressings is becoming larger in the extrusion industry.

The trial and error process has a number of important drawbacks. Obviously, the trial-pressing, the correction, the redesign and the remanufacturing are expensive in terms of man hours, machine time and raw material. A less apparent drawback is that, due to the unpredictability of the number of trial-pressings that will be required for a certain profile, it is very difficult to give an accurate estimate of the time that is necessary to realize an adequate die design. As a result, the term of delivery that is agreed upon with the customer has to allow for possibly unnecessary trial-pressings to avoid the risk of a late delivery. Lastly, sometimes there is no consensus on the mechanisms that cause the failing of a die and the opinions deviate on the methods that have to be used to adjust a die.

1.2 Finite element modeling of extrusion

To reduce the drawbacks of the trial and error process dealt with in the previous section, aluminum extrusion companies have invested large amounts of money and effort to limit the number of trial-pressings. Partly these efforts have aimed at improving the expert systems, and partly these efforts have been put into replacing the trial-pressings by numerical computations which predict what the profile shape would be for a given die design. Predictions can be obtained using analytical upper-bound models but these are only applicable to very simple profile shapes and require extensive simplifications of the aluminum material model (Solomon et al., 1998; Støren, 1993). The approach most commonly pursued in predicting the performance of a die is the finite element method because it allows for complex profile shapes and material models. In this work the finite element method will be applied as well.

Until now finite element computations have only been moderately successful. Most computations have been restricted to two-dimensional (2D) settings due to the prohibitively large computational times associated with modeling extrusion in a three-dimensional (3D) configuration (Hanssen et al., 1998). Nevertheless, some 3D computations have been performed

(Lishnij et al., 1998) but these computations are restricted to very simple profile shapes with a high level of symmetry (Mooi, 1996). The modeling of simple profiles allow the manual generation of the 3D discretization of the domain while the symmetry can be exploited to model only a representative part of the domain in order to reduce the size of the model to a manageable level (Legat and Marchal, 1993). Moreover, simplifications with respect to e.g. the interaction of the aluminum with the die and/or the influence of thermal effects, are often introduced. The simplifications reduce the order of non-linearity of the numerical models to arrive at acceptable computational times. The applicability of these simplified models is limited since the underlying assumptions make the results only valid for straightforward profile shapes for which the die design does not pose problems in practice. Furthermore, the large preprocessing time associated with the, partly manual, discretization of the 3D domain is a limiting factor on the applicability of these computations in extrusion practice.

The goal of this work is to improve the applicability of finite element computations to the extrusion practice by eliminating the shortcomings discussed above. To do so, an advanced description of the aluminum behavior is adopted, including thermal effects. Also, the interaction between the aluminum and the die is modeled accurately. Additionally, the discretization process is highly automated, such that only a limited amount of input from the user is required, to reduce the amount of time needed for preprocessing drastically. Lastly, the models are implemented in such a manner that they require acceptable computational times, that can compete with the time needed to perform a trial-pressing. As a result of these improvements complex 3D profiles can be analyzed accurately and efficiently.

1.3 Scope of the thesis

In Chapter 2 of this thesis a set of equations will be derived which model the extrusion process. Starting from the integral conservation laws the irrelevant terms will be identified and eliminated. This leads to a Stokes flow problem that is coupled to a thermal convection-diffusion problem. The equations will be formulated in an Eulerian, i.e. spatially fixed, reference frame. The Eulerian reference frame requires that special attention is paid to changing domain boundaries which occur when the profile is free to move after the exit of the die, or when the tooling deflects. To this end, two additional problems are introduced that will be discussed as well. The first problem models the shape of the aluminum profile that leaves the die. This free surface problem is reformulated in an innovative manner which leads to two independent convection problems that are defined on the surface of the profile. The second problem describes the shape of the die package as it deflects under the extrusion pressures.

Chapter 3 will discuss how the equations derived in Chapter 2 can be discretized using the finite element method. For the discretization of the flow problem the MINI element is used. This is a mixed, stabilized element in which the velocity and the pressure are equally interpolated. The boundary conditions at the aluminum-tooling interface are modeled with a Coulomb friction model. The Coulomb friction is incorporated into the flow problem using a Lagrange multiplier method that is discretized using a new, more accurate method. For the discretization of the thermal problem a stabilized method, the Streamline Upwind / Petrov-Galerkin (SUPG) method is employed to capture the temperature field in the convection dominated areas of the aluminum domain. The discretization of the outflow shape problem is performed by applying the SUPG method as well, while the discretization of the die deflection problem requires the use of Lagrange multipliers to incorporate the interaction between the various parts of the tooling. After all the separate problems have been discretized a non-linear, coupled system of equations results. The staggered approach, applied to decouple and linearize this system of equations into several linear systems, is outlined, followed by an overview of the preconditioned iterative solvers that are used to solve the resulting linear systems of equations.

The spatial discretization of the computational domain, the mesh generation, is presented in Chapter 4. To keep the number of elements to a minimum without jeopardizing the accuracy of the solution fields, special algorithms are developed to generate the surface and volume meshes. These mesh generators produce high aspect ratio elements that are stretched in the direction where low gradients occur in the solution field and compressed in the direction where the gradients are high. The meshes resulting from these dedicated mesh generators cannot be used directly for extrusion. The varying bearing length and the geometry of the bridge-part can only be captured through specially designed post-processing steps. These steps will be discussed last.

Chapter 5 is concerned with the accuracy of the finite element model that has been elaborated in the previous two chapters. In the first section, the influence of each of the coupled problems on the solution fields that are computed for the aluminum flow is investigated. Also, the effect of altering the coefficient related to the friction boundary conditions is demonstrated. In the next section, it is shown that the high aspect ratio meshes are sufficiently fine to compute the solution field accurately. This is done by comparing the results of computations on meshes, generated along the strategy explained in Chapter 4, to those obtained on very refined meshes. In the final section the performance of the finite element model is assessed by comparing the results of the analysis to the results that are obtained in extrusion practice for a die that has been specially designed for this comparison.

Three examples of finite element computations on extrusion are presented in Chapter 6. In the first example the shape of a profile that was extruded in practice with a die that did not perform satisfactorily, is predicted. The computations display the same behavior as observed in reality and render insight into the question why the alterations that were made to the die did not suffice to improve the performance of the die. The second example concerns the assessment of the performance of a die designed for a hollow profile. In practice this die performed well, which is confirmed by the computations of the flow field. The last example illustrates the general applicability of the tool that has been developed. It concerns the computation of the flow field associated with the extrusion of a polymer through a fictive die. In this example the influence of the shape of the out-flowing polymer on the computed velocity field is very strong. Moreover, in this example the polymer is not pressed through the die using a ram, but by applying a twin-screw extruder, which is incorporated in the model as well.

In Chapter 7 conclusions are drawn and recommendations for future work are given.

Chapter 2

Mathematical modeling of extrusion

The extrusion process is numerically analyzed using the finite element method. To derive the set of equations that have to be solved, a Lagrangian or an Eulerian approach can be adopted. Eulerian approaches, which are characterized by a stationary spatial reference frame, are especially useful for those processes where large deformations occur, because the computational domain (the mesh) is spatially fixed and not ’attached’ to the material. This eliminates the problems of large mesh distortions that can occur in a Lagrangian approach. However, when the shape of the material domain is unknown a priori, or changes during the process, this poses a problem in an Eulerian approach. In that case additional equations have to be introduced to ensure that the material and the computational domain become and/or remain coincident.

This chapter will start with the conservation laws for mass, momentum and energy within an Eulerian framework. These equations are simplified using the process characteristics of extrusion and the constitutive behavior of aluminum to arrive at a mechanical Stokes problem combined with a thermal convection-diffusion problem. The boundary conditions needed to solve these problems are presented, where particular attention is paid to the friction between the aluminum and the die in the mechanical problem.

Thereafter, the problems which are intrinsically caused by using the Eulerian approach in combination with a changing material domain are considered. At the outflow of the die the aluminum boundary is not constrained by tooling and can therefore deform. The equations that determine the deformations at the outflow are discussed. Also, the die can deform significantly due to the process forces. Since the aluminum is in contact with the die, this affects the shape of the aluminum domain. The equations necessary to determine the deformations of the die will be discussed last.

2.1 Conservation laws

Without loss of generality the conservation laws can either be formulated in terms of a Lagrangian or an Eulerian description. In this thesis the latter formulation is adopted for several reasons. First, the aluminum undergoes extremely large deformations during extrusion as can be deduced from the experiments presented in e.g. (Valberg et al., 1996) and (Kialka and Misiolek, 1996), where a grid was attached to aluminum during extrusion. In a Lagrangian finite element approach large deformations would necessitate remeshing to ensure acceptable element quality throughout the simulation. Second, a velocity based formulation, which is typical for an Eulerian framework, follows naturally from the constitutive model that is adopted for the aluminum in the following section.

Within the Eulerian framework, where the relevant physical fields are expressed as functions of the spatial position $\vec{x}$, and the time t, the conservation laws are given by:

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{v}) = 0 \tag{2.1}$$

$$\rho \frac{\partial \vec{v}}{\partial t} + \rho \vec{v} \cdot \left(\vec{\nabla} \vec{v} \right) = \vec{\nabla} \cdot \boldsymbol{\sigma} + \rho \vec{g} \tag{2.2}$$

$$\rho \left(\frac{\partial e}{\partial t} + \vec{v} \cdot \left(\vec{\nabla} e \right) \right) = \boldsymbol{\sigma} : \boldsymbol{D} - \vec{\nabla} \cdot \vec{h} + \rho r \tag{2.3}$$

where ρ represents the density, $\vec{v}$ the velocity vector, $\boldsymbol{\sigma}$ the Cauchy stress tensor, $\vec{g}$ the distributed force per unit of mass related to gravity, e the specific internal energy, $\boldsymbol{D} = \frac{1}{2}(\vec{\nabla}\vec{v} + (\vec{\nabla}\vec{v})^c)$ the rate of deformation tensor, $\vec{h}$ the heat flux, and r the internal heat source. These conservation laws are used as a basis to derive the equations that govern the flow and the temperature of the aluminum during extrusion.

2.2 Constitutive equations

The Cauchy stress $\boldsymbol{\sigma}$ in (2.2) can be split into a hydrostatic and a deviatoric part:

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \boldsymbol{\tau} \tag{2.4}$$

where p is the hydrostatic pressure and $\boldsymbol{\tau}$ the deviatoric part of the stress tensor. The deviatoric part of the stress tensor is associated to the rheological behavior of the material. This part will be considered first. Then attention will be focused on a mutual comparison of the separate terms in (2.1) and (2.2). It will be shown that for extrusion a steady state, incompressible description is adequate, and that inertia effects and gravity can be omitted.

For extrusion conditions the deviatoric stress in the aluminum is determined by the temperature T, and the strain rate according to (Lalli and DeArdo, 1990; Wright et al., 1996):

$$\boldsymbol{\tau} = 2\eta(\boldsymbol{D}, T)\boldsymbol{D}^d \tag{2.5}$$

where $\eta(\boldsymbol{D}, T)$ represents the viscosity which is defined as:

$$\eta(\boldsymbol{D}, T) = \frac{\sinh^{-1}\left(\left(\frac{Z}{A}\right)^{\frac{1}{n}}\right)}{3\beta\dot{\gamma}} \tag{2.6}$$

where β, A, and n are constants that depend on the type of aluminum. The variable $\dot{\gamma}$ represents the strain rate and Z is the Zener-Hollomon parameter which can be interpreted as a temperature compensated strain rate (Garafolo, 1963). These variables are defined by, respectively:

$$\dot{\gamma} = \sqrt{\frac{2}{3}\boldsymbol{D} : \boldsymbol{D}}, \qquad Z = \dot{\gamma} \exp\left(\frac{\Delta H}{RT}\right)$$

where ΔH is a constant related to the activation energy, and R the gas constant. In this thesis the material parameters as suggested in (Sample and Lalli, 1987) will be used, see Table 2.1.

ΔH	$1.772 \cdot 10^{5}$	[J mol^{-1}]	R	8.321	[J mol^{-1}K^{-1}]
β	$3.224 \cdot 10^{-8}$	[m^{2} N^{-1}]	A	$1.445 \cdot 10^{12}$	[s^{-1}]
n	4.750	[-]			

Table 2.1: The material parameters used in the constitutive model for aluminum.

If the parameters in Table 2.1 are used to plot the viscosity as a function of the respectively the shear rate $\dot{\gamma}$ and the temperature T the plots in Figure 2.1. By comparing Figure 2.1(a) with Figure 2.1(b) it can be seen that the influence of the the shear rate $\dot{\gamma}$ on the viscosity η is much larger than that of the temperature T.

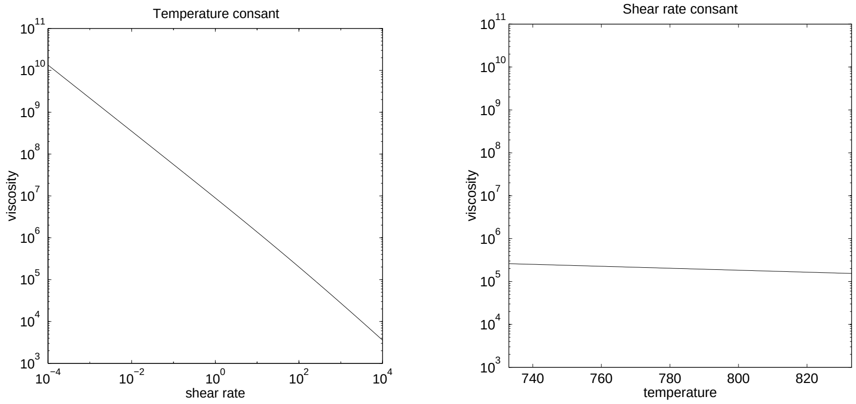
2.3 Flow problem

An approach that is commonly pursued to analyze aluminum extrusion is to model the process as an incompressible Stokes flow. To investigate the correctness of this simplification the conservation law of mass (2.1) is reformulated:

$$\frac{1}{\rho}\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot \vec{v} + \vec{v} \cdot \frac{1}{\rho}\vec{\nabla}\rho = 0$$

assuming the density to be a function of the temperature and the pressure this leads to:

$$\frac{1}{\rho}\frac{\partial \rho}{\partial T}\frac{\partial T}{\partial t} + \frac{1}{\rho}\frac{\partial \rho}{\partial p}\frac{\partial p}{\partial t} + \vec{\nabla} \cdot \vec{v} + \vec{v} \cdot \frac{1}{\rho}\frac{\partial \rho}{\partial T}\vec{\nabla}T + \vec{v} \cdot \frac{1}{\rho}\frac{\partial \rho}{\partial p}\vec{\nabla}p = 0$$



(a) $T = 783$ [K]

(b) $\dot{\gamma} = 10^2$ [s^{-1}]

Figure 2.1: Viscosity η for values of the shear rate $\dot{\gamma}$ and temperature T, typical for extrusion.

with $\alpha = -\frac{1}{\rho}\frac{\partial \rho}{\partial T}$ and $\frac{1}{\kappa} = \frac{1}{\rho}\frac{\partial \rho}{\partial p}$ this is rewritten as:

$$-\alpha\frac{\partial T}{\partial t} + \frac{1}{\kappa}\frac{\partial p}{\partial t} + \vec{\nabla}\cdot\vec{v} - \alpha\vec{v}\cdot\vec{\nabla}T + \frac{1}{\kappa}\vec{v}\cdot\vec{\nabla}p = 0 \tag{2.7}$$

where α and κ are the thermal expansion coefficient and the compressibility modulus, respectively, which are both assumed constant. Next, the constitutive model for aluminum, (2.5) is substituted into (2.2) to render:

$$\rho\frac{\partial\vec{v}}{\partial t} + \rho\vec{v}\cdot\left(\vec{\nabla}\vec{v}\right) = \vec{\nabla}\cdot\left(-p\boldsymbol{I} + 2\eta(\boldsymbol{D}, T)\boldsymbol{D}^d\right) + \rho\vec{g} \tag{2.8}$$

Now the characteristic velocity v_0, length l_0, time t_0, viscosity η_0, gravity g_0, density ρ_0, offset temperature T_0 and temperature range ΔT are introduced to scale the variables and operators present in (2.7) and (2.8) according to:

$$\begin{array}{lll} \vec{v} = v_0\vec{v}^*, & \eta = \eta_0\eta^*, & p = \dfrac{\eta_0 v_0}{l_0}p^* \\ \vec{g} = g_0\vec{g}^*, & \rho = \rho_0\rho^*, & \vec{\nabla} = \dfrac{1}{l_0}\vec{\nabla}^* \\ \dfrac{\partial}{\partial t} = \dfrac{1}{t_0}\dfrac{\partial}{\partial t^*}, & \boldsymbol{D} = \dfrac{v_0}{l_0}\boldsymbol{D}^*, & T = T_0 + \Delta T\ T^* \end{array} \tag{2.9}$$

With the scaling the conservation of mass in (2.7) and the conservation of momentum in (2.8) can be rewritten to:

$$\begin{aligned} & Sr\,\alpha\Delta T\left(\frac{\partial T^*}{\partial t^*}\right) + Sr\,\frac{1}{\kappa}\frac{v_0\eta_0}{l_0}\left(\frac{\partial p^*}{\partial t^*}\right) + \\ & \qquad + \left(\vec{\nabla}^*\cdot\vec{v}^*\right) - \alpha\Delta T\left(\vec{v}^*\cdot\vec{\nabla}^* T^*\right) + \frac{1}{\kappa}\frac{v_0\eta_0}{l_0}\left(\vec{v}^*\cdot\vec{\nabla}^* p^*\right) = 0 \end{aligned} \tag{2.10}$$

$$\begin{aligned} & Sr\left(\frac{\partial\vec{v}^*}{\partial t^*}\right) + \left(\vec{v}^*\cdot\vec{\nabla}^*\vec{v}^*\right) = \\ & \qquad = \frac{1}{Re}\left(\frac{1}{\rho^*}\vec{\nabla}^*\cdot\left(-p^*\boldsymbol{I} + 2\eta^*(\boldsymbol{D}^*)^d\right)\right) + \frac{1}{Fr}\left(\vec{g}^*\right) \end{aligned} \tag{2.11}$$

where the dimensionless numbers Sr, Re and Fr are the Strouhal, the Reynolds and the Froude numbers, respectively, which are defined according to:

$$Sr = \frac{l_0}{t_0 v_0}, \qquad Re = \frac{\rho_0 v_0 l_0}{\eta_0}, \qquad Fr = \frac{v_0^2}{g_0 l_0} \tag{2.12}$$

For aluminum extrusion characteristic values for the parameters in (2.10) and (2.11) are:

$$\begin{aligned} & v_0 = 10^0\ [\mathrm{m\,s^{-1}}], && \eta_0 = 10^5\ [\mathrm{kg\,m^{-1}s^{-1}}], && g_0 = 10^1\ [\mathrm{m\,s^{-2}}] \\ & l_0 = 10^{-3}\ [\mathrm{m}], && t_0 = 10^2\ [\mathrm{s}], && \Delta T = 10^2\ [\mathrm{K}] \\ & \rho_0 = 10^3\ [\mathrm{kg\,m^{-3}}], && \alpha = 10^{-7}\ [\mathrm{K^{-1}}], && \kappa = 10^{11}\ [\mathrm{kg\,m^{-1}s^{-2}}] \end{aligned} \tag{2.13}$$

Substitution renders the following order of magnitude estimations:

$$\begin{aligned} & Sr\,\alpha\Delta T = \mathcal{O}(10^{-10}), && Sr\,\frac{1}{\kappa}\frac{v_0\eta_0}{l_0} = \mathcal{O}(10^{-8}), && \\ & \alpha\Delta T = \mathcal{O}(10^{-5}) && \frac{1}{\kappa}\frac{v_0\eta_0}{l_0} = \mathcal{O}(10^{-3}) && \\ & Sr = \mathcal{O}(10^{-5}), && \frac{1}{Re} = \mathcal{O}(10^5), && \frac{1}{Fr} = \mathcal{O}(10^{-2}) \end{aligned} \tag{2.14}$$

If the negligible entries in (2.10) and (2.11) are removed, an incompressible Stokes problem emerges:

$$\vec{\nabla}\cdot\vec{v} = 0 \tag{2.15}$$

$$\vec{\nabla}\cdot\left(-p\boldsymbol{I} + 2\eta(\boldsymbol{D},T)\boldsymbol{D}\right) = \vec{0} \tag{2.16}$$

where the scaling has been dropped and use has been made of the fact that $\boldsymbol{D}^d = \boldsymbol{D}$ as a result of (2.15).

2.4 Mechanical boundary conditions

In order to solve the differential equation (2.16), subject to the incompressibility constraint (2.15), boundary conditions need to be prescribed. In the extrusion process two distinct zones can be identified. In the container and the die the aluminum is constrained by the tooling which exerts surface tractions due to contact and friction (see also Figure 2.2). It is assumed that the aluminum remains in contact with the tooling throughout the analysis. In the other zone, at the outflow of the die, the aluminum does not experience any surface forces and the boundary is therefore stress free. The boundary conditions for these zones are governed by equations formulated in the following.

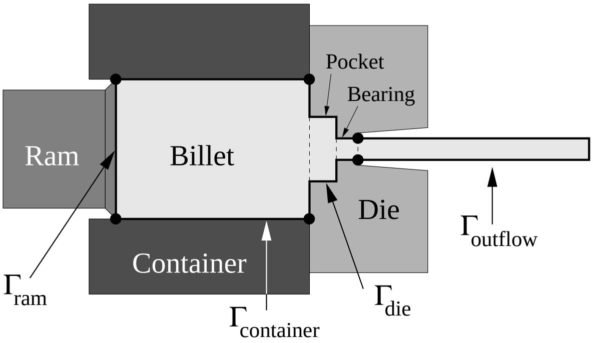


Figure 2.2: Surface definition for extrusion.

2.4.1 Friction

Friction in aluminum extrusion is a complex and hardly understood phenomenon. Due to the high pressures and temperatures it is extremely difficult to perform in situ experiments. Consequently, most information on friction phenomena is obtained by studying the bearing surface after extrusion (Abtahi et al., 1996; Clode and Sheppard, 1990; Welo et al., 1996). These studies render some insight into the mechanisms that govern friction but do not supply sufficient information to specify the constitutive relations that can be applied to model friction in an extrusion analysis. Yet, there are indications that some models are more suitable than others.

To model friction, a variety of constitutive laws can be used to define the surface tractions that are exerted on a material when it is sliding along a surface. Usually, the tangential surface tractions, $(\boldsymbol{I}-\vec{n}\vec{n})\cdot\boldsymbol{\sigma}\cdot\vec{n}$, with $\vec{n}$ the outward unit surface normal vector, are related to the size of the normal surface traction $\vec{n}\cdot\boldsymbol{\sigma}\cdot\vec{n}$ and to the relative sliding velocity along the tooling $(\vec{v}-\vec{v}_{tool})$. As a point of departure a Norton-Hoff type law is considered which gives the following relation (Moal and Massoni, 1995):

$$(\boldsymbol{I}-\vec{n}\vec{n})\cdot\boldsymbol{\sigma}\cdot\vec{n} = -\mu\|\vec{n}\cdot\boldsymbol{\sigma}\cdot\vec{n}\|^{\alpha}\|\vec{v}-\vec{v}_{tool}\|^{\beta-1}(\vec{v}-\vec{v}_{tool}) \tag{2.17}$$

where μ represents the friction coefficient, and α and β are additional constitutive parameters. Notice that for this law the dimension of μ depends on the choice of α and β. For extrusion the velocity of the tooling is given by (see also Figure 2.2):

$$\vec{v}_{tool} = \begin{cases} \vec{0} & \forall \vec{x} \in \Gamma_{die} \cup \Gamma_{container}, \\ \vec{v}_{ram} & \forall \vec{x} \in \Gamma_{ram} \end{cases} \tag{2.18}$$

For the particular values of $\alpha = 0$, $\beta = 1$, (2.17) reduces to the Norton friction law. For $\alpha = 0$, $\beta = 0$, (2.17) represents the constant friction law, while for $\alpha = 1$, $\beta = 0$, the Coulomb friction law is recovered.

In extrusion analysis all of these laws have been used to model friction (see e.g. (Welo et al., 1996) for Norton, (Yang, 1994) for constant friction, and (Mooi, 1996) for Coulomb). However, in extrusion practice it has been observed that locally increasing the length of the bearings above a certain value, does not further improve the balance of the extrudate. This suggests that towards the exit of the bearing the frictional traction decreases and even vanishes for large bearing lengths. This trend cannot be captured by adopting the constant friction law. Also, the Norton law is incapable of describing this behavior since it predicts that a particle that is sliding along the bearing will experience a frictional traction proportional to the sliding velocity. Since the velocity of a particle will not change significantly within the bearing, the Norton law suggests a nearly constant friction. Furthermore, there are indications that the sliding velocity has very little influence on the friction forces in metal-metal contact (Anand, 1993; Anand and Tong, 1993). Additionally, experiments in extrusion with non-parallel bearings reveal that the contact pressure plays a crucial role (Abtahi et al., 1996; Welo et al., 1996). Therefore, in this thesis a Coulomb friction law is employed.

Coulomb's law of friction may be expressed (Heege et al., 1995) by a friction criterion (2.19) which defines the limits of the frictional traction in relation to the normal traction, a slip rule (2.20) which prescribes the frictional tractions to be opposed to the sliding velocity, and a complementarity condition (2.21) which ensures that stick and slip are mutually exclusive conditions:

$$\|(\boldsymbol{I} - \vec{n}\vec{n}) \cdot \boldsymbol{\sigma} \cdot \vec{n}\| \leq -\mu(\vec{n} \cdot \boldsymbol{\sigma} \cdot \vec{n}) \tag{2.19}$$

$$\frac{\vec{v} - \vec{v}_{tool}}{\|\vec{v} - \vec{v}_{tool}\|} = -\frac{(\boldsymbol{I} - \vec{n}\vec{n}) \cdot \boldsymbol{\sigma} \cdot \vec{n}}{\|(\boldsymbol{I} - \vec{n}\vec{n}) \cdot \boldsymbol{\sigma} \cdot \vec{n}\|} \tag{2.20}$$

$$(\|\vec{v} - \vec{v}_{tool}\|)\,(\|(\boldsymbol{I} - \vec{n}\vec{n}) \cdot \boldsymbol{\sigma} \cdot \vec{n}\| + \mu(\vec{n} \cdot \boldsymbol{\sigma} \cdot \vec{n})) = 0 \tag{2.21}$$

In every point on the aluminum surface that is in contact with the tooling, the stress state and the velocity must satisfy these three equations. Also, in case of sliding the size of the frictional surface tractions, $\mu(\vec{n} \cdot \boldsymbol{\sigma} \cdot \vec{n})$ should not exceed the local yield stress of the aluminum $\dot{\gamma}\eta$. In extrusion this requirement is implicitly met because numerical experiments show that, in the areas where the aluminum slides along the tooling, the contact pressures are considerably lower than the yield stress of the aluminum there.

Due to the difficulties of performing in situ experiments it is impossible to measure the Coulomb friction coefficient, μ. Therefore, the friction coefficient has been estimated by performing a number of simulations with varying Coulomb coefficients for a die design that rendered a well balanced exit flow in extrusion practice. Of these computations the flow field that was obtained with a Coulomb friction coefficient of 0.4 [-] displayed the most uniform exit flow. Therefore in this work a Coulomb friction coefficient of 0.4 [-] will be used.

2.4.2 Free surface

The boundary of the outflow at the end of the die is stress free since surface tractions are assumed to be absent. The boundary condition applied at the outflow is then given by:

$$\boldsymbol{\sigma} \cdot \vec{n} = \vec{0} \qquad \forall \vec{x} \in \Gamma_{outflow} \tag{2.22}$$

2.5 Thermal problem

The conservation of energy law in (2.3) that governs the thermal problem can be simplified based on the assumption that no heat is generated or lost due to phase changes or chemical reactions, i.e. by removing $\vec{r}$ from (2.3). Furthermore, it is assumed that Fourier's law, given by $\vec{h} = -\lambda \vec{\nabla} T$, holds for the heat flux, and that the specific internal energy is only a function of the temperature, i.e. $e = \rho c_p T$, which follows directly from the incompressible, viscous material model that is used for aluminum. If the mechanically dissipated heat is computed using the flow field from (2.15) and (2.16), the following expression for the conservation of energy law results:

$$\rho c_p \frac{\partial T}{\partial t} + \rho c_p \vec{v} \cdot \vec{\nabla} T = 2\eta \boldsymbol{D} : \boldsymbol{D} + \lambda \vec{\nabla} \cdot \left(\vec{\nabla} T \right) \tag{2.23}$$

where c_p represents the specific heat capacity, and λ the heat conduction coefficient, which are both assumed constant. Equation (2.23) can be scaled with the use of (2.9) to render:

$$Sr \left(\rho^* \frac{\partial T^*}{\partial t^*} \right) + \left(\rho^* \vec{v}^* \cdot \vec{\nabla}^* T^* \right) = \frac{Br}{Pe} 2 \left(\eta^* \boldsymbol{D}^* : \boldsymbol{D}^* \right) + \frac{1}{Pe} \vec{\nabla}^* \cdot \left(\vec{\nabla}^* T^* \right) \tag{2.24}$$

where the dimensionless numbers Fo, Pe and Br are the Fourier, the Péclet and the Brinkman numbers, respectively, which are defined according to:

$$Sr = \frac{l_0}{t_0 v_0}, \qquad Pe = \frac{\rho_0 c_p v_0 l_0}{\lambda}, \qquad Br = \frac{\eta_0 v_0^2}{\Delta T \lambda} \tag{2.25}$$

To evaluate the importance of each of the terms, the characteristic values in (2.13) are adopted along with:

$$c_p = 10^3 \text{ [J kg}^{-1}\text{K}^{-1}\text{]}, \qquad \lambda = 10^2 \text{ [J m}^{-1}\text{s}^{-1}\text{K}^{-1}\text{]} \tag{2.26}$$

Substitution renders the following order of magnitude estimations:

$$Sr = \mathcal{O}(10^{-5}), \qquad \frac{Br}{Pe} = \mathcal{O}(10^{0}), \qquad \frac{1}{Pe} = \mathcal{O}(10^{-1}) \tag{2.27}$$

If the irrelevant entries in (2.24) are removed the following convection diffusion problem emerges:

$$\rho c_p \vec{v} \cdot \vec{\nabla} T - \lambda \vec{\nabla} \cdot \left(\vec{\nabla} T \right) = 2\eta \boldsymbol{D} : \boldsymbol{D} \tag{2.28}$$

where the scaling has been dropped.

2.6 Thermal boundary conditions

In order to model the thermal boundary conditions of extrusion accurately it is necessary to perform a complete heat transfer analysis of the press and the surrounding (instationary) air. Since such an analysis is complicated, the boundary conditions are usually simplified by assuming that the ram has a constant temperature and that the container, the die and the outflow are insulated (see e.g. (Mooi, 1996)):

$$\begin{cases} T = T_{ram} & \forall \vec{x} \in \Gamma_{ram} \\ \vec{\nabla} T \cdot \vec{n} = 0 & \forall \vec{x} \in \Gamma_{container} \cup \Gamma_{die} \cup \Gamma_{outflow} \end{cases} \tag{2.29}$$

In this assumption the heat transfer across the boundary of the aluminum domain is solely attributed to convection. This neglects the heat transfer due to diffusion, i.e. due to temperature gradients at the aluminum boundary.

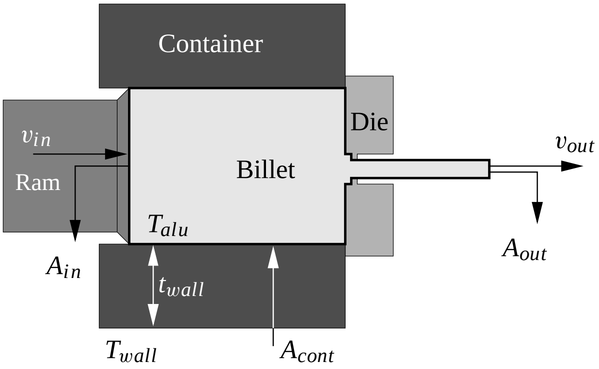


Figure 2.3: Simplified extrusion model.

To investigate the correctness of this assumption a simplified heat model of the container is analyzed (see Figure 2.3). The contribution of convection and diffusion to the thermal power that is inserted into the aluminum at the boundary can be computed as follows. The order of magnitude of the thermal power put into the aluminum due to convection P_{conv}, is:

$$\begin{aligned} P_{conv}\ [\mathrm{W}] &= v_{in} A_{in} \rho_0 c_p T_{in} - v_{out} A_{out} \rho_0 c_p T_{out} \\ &= v_{in} A_{in} \rho_0 c_p (-\Delta T) \\ &= -\mathcal{O}(10^{-2})\mathcal{O}(10^{-2})\mathcal{O}(10^{3})\mathcal{O}(10^{3})\mathcal{O}(10^{2}) = -\mathcal{O}(10^{4})\ [\mathrm{W}] \end{aligned} \tag{2.30}$$

where $v_{in} = \mathcal{O}(10^{-2})$ [ms^{-1}], $A_{in} = \pi r^2 = \pi \mathcal{O}(10^{-1})^2 = \mathcal{O}(10^{-2})$ [m^2] are used, and ρ, c_p and ΔT as introduced previously.

The thermal power that is inserted into the aluminum due to diffusion P_{diff} is governed by temperature gradients at the container, the die and the outflow wall. Compared to the container surface area, the die surface area is less than 10% and the outflow surface area for the modeled profile lengths is less than 1%. The container consists of a steel shell that is in contact with the aluminum on the inside and that is kept at a constant temperature on the outside. The thickness of this shell is $t_{wall} = \mathcal{O}(10^{-1})$ [m] and the temperature drop over the thickness of this shell is $T_{wall} - T_{alu} = -\mathcal{O}(10^{1})$ [K] which renders a temperature gradient $\vec{\nabla} T = -\mathcal{O}(10^{2})$ [K m^{-1}] at in the shell. In extrusion practice similar temperature gradients have been measured in the die and the profile. These measurements, combined with the relative surface areas justify the assumption that the power inserted through temperature gradients is mainly governed by the container. This leads to the following order of magnitude estimate:

$$\begin{aligned} P_{diff} \text{ [W]} &= A_{cont}\lambda_{steel}\frac{T_{wall}-T_{alu}}{t_{wall}} \\ &= \mathcal{O}(10^{-1})\mathcal{O}(10^{1})\frac{-\mathcal{O}(10^{1})}{\mathcal{O}(10^{-1})} = -\mathcal{O}(10^{2}) \text{ [W]} \end{aligned} \tag{2.31}$$

where $A_{cont} = 2\pi rl = 2\pi\mathcal{O}(10^{-1})\mathcal{O}(10^{-1}) = \mathcal{O}(10^{-1})$ [m^2] is used, as well as $\lambda_{steel} = \mathcal{O}(10^{1})$ [Jm^{-1}s^{-1}K^{-1}]. It should be noted that the same order of magnitude estimates are obtained for P_{conv} and P_{diff} if real values, rather than order of magnitude estimates, are substituted for the variables in (2.30) and (2.31), respectively.

A comparison between P_{conv} and P_{diff} reveals that the power related to convection is considerably larger than that related to diffusion. Therefore the boundary conditions in (2.29) are a justifiable simplification and will be used in this thesis.

2.7 Changing aluminum boundary

To obtain accurate results from the finite element computations it is of paramount importance that the computational domain, the mesh, coincides with the aluminum domain. However, the shape of the aluminum domain cannot be predicted a priori. It is therefore impossible to generate a mesh that is coincident with the shape of the aluminum domain. Specifically, the shape of the outflow surface and of the die surface change depending on the process conditions. The shape of the outflow surface will not coincide with the original one if the calculated velocities at this (by definition stress-free) surface have a component perpendicular to the boundary of the original surface. The shape of the die surface will change if the die deflects due to the process forces. Strictly speaking the deflections of the ram and the container influence the shape of the aluminum domain as well. However, the deformations of the ram and the container are so small that they can be neglected. As a consequence of these changing boundaries, the mesh has to be adapted to ensure that it remains, or becomes, coincident with the aluminum.

2.7.1 Three field approach

The varying shape of the aluminum is handled in the following way (van Rens et al., 1998a) (see also Figure 2.4). As a first approximation, it is assumed that the die does not deform and that the aluminum exits the die according to the domain initially specified, which is perfectly straight in general. Using this point of departure, the shape of the aluminum domain Ω is defined and a mesh, denoted by triangularization $\mathcal{T}^k$ consisting of tetrahedral elements k, can be generated. Then an iterative procedure is started. Within this procedure the first step consists of computing the flow field of the aluminum using the current aluminum mesh. From this flow field the aluminum velocities, $\vec{v}$ at the outflow surface can be computed, as well as the surface traction $-p\vec{n}+\boldsymbol{\tau}\cdot\vec{n}$ that the aluminum experiences at the die surface. In the second step the new shape of the outflow surface is computed, using the velocities at the surface, and the new shape of the die is computed, using the interface tractions between the aluminum and the die. From these new shapes the aluminum surface displacements $\vec{u}$, required to keep the boundary of the aluminum mesh conform with the outflow surface and the die surface, can be computed. In the third step the new node positions of the aluminum mesh are computed using the displacements of the boundary. These new node positions define a new mesh $\mathcal{T}^k$ on the aluminum domain Ω. With the new mesh the next iteration is started.

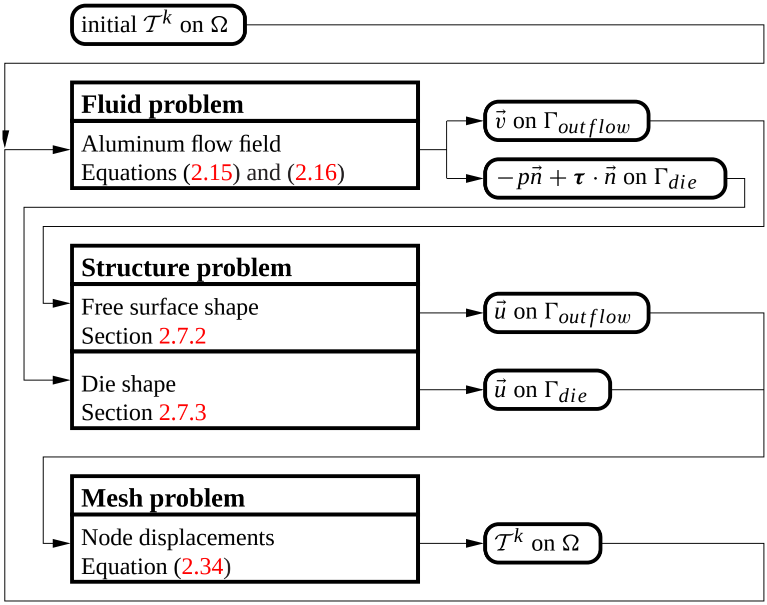

initial $\mathcal{T}^k$ on Ω
Fluid problem
Aluminum flow field
Equations (2.15) and (2.16)
$\vec{v}$ on $\Gamma_{outflow}$
$-p\vec{n}+\boldsymbol{\tau}\cdot\vec{n}$ on Γ_{die}
Structure problem
Free surface shape
Section 2.7.2
Die shape
Section 2.7.3
$\vec{u}$ on $\Gamma_{outflow}$
$\vec{u}$ on Γ_{die}
Mesh problem
Node displacements
Equation (2.34)
$\mathcal{T}^k$ on Ω

Figure 2.4: The incorporation of a varying aluminum shape.

The iterative procedure described above can be characterized as a three field approach (Farhat et al., 1998; Lesoinne and Farhat, 1998). Other procedures that fall within this category are the Arbitrary Euler Lagrange method and the dynamic mesh method. In such an approach three fields with corresponding problems are defined. The first field, the fluid field, involves the velocities and pressures that are present in the aluminum flow problem that was discussed in Section 2.3. The second field describes the displacements of the structure with which the aluminum domain interacts. In this work, two structure problems are present, the outflow surface shape problem and the flexible die shape problem. The third field is related to the displacements of the aluminum mesh. The problem related to this field provides the relations that govern the displacements of the nodes in the mesh as a function of the changing boundary. For this purpose the mesh is treated as a pseudo structure. Since it is only defined for computational purposes, the mechanical properties of the pseudo structure are not relevant. Therefore, linear elastic behavior is used.

To solve the problems that are present in the three field approach depicted in Figure 2.4, their exact form has to be defined. To do so, the displacements $\vec{u}_s$ and $\vec{u}_m$ are defined on the domains of the structure and the mesh, respectively. Then these problems can be written according to:

$$\text{flow problem} \begin{cases} \vec{\nabla} \cdot \vec{v} = 0 \\ \vec{\nabla} \cdot (-p\boldsymbol{I} + 2\eta \boldsymbol{D}) = \vec{0} \end{cases} \tag{2.32}$$

$$\text{structure problem} \begin{cases} \mathcal{G}(\vec{u}_s) = 0 \end{cases} \tag{2.33}$$

$$\text{mesh problem} \begin{cases} \vec{\nabla} \cdot \left(\boldsymbol{C} : \frac{1}{2} \left(\vec{\nabla}\vec{u}_m + (\vec{\nabla}\vec{u}_m)^c \right) \right) = \vec{0} \end{cases} \tag{2.34}$$

where $\mathcal{G}$ represents the function that defines the shape of the structure, and $\boldsymbol{C}$ the fourth order Hookean elasticity tensor. The exact form of $\mathcal{G}$ will be discussed in more detail for both the free surface and the flexible die in Section 2.7.2 and Section 2.7.3, respectively.

2.7.2 Structure problem - Free surface shape

The outflow surface is modeled by a membrane structure of zero thickness. This membrane does not exert any surface tractions on the aluminum, which corresponds to a stress-free outflow condition for the aluminum. The shape of the structure is fully defined by the set of particle tracks belonging to those particles that are located on the outflow surface. Because a particle that starts on the surface remains on the surface, the shape of the structure is determined by those particles that are on the outflow surface when the aluminum leaves the die.

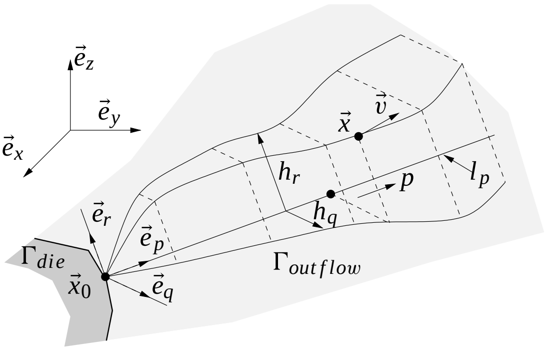


Figure 2.5: Variable definition for particle track.

Let the principal flow direction of the aluminum at the outflow be denoted by the unit vector $\vec{e}_p$ (see also Figure 2.5). Also, $\vec{e}_q$ and $\vec{e}_r$ represent the unit vectors that constitute an orthogonal basis with $\vec{e}_p$. Then, the trajectory of one particle can be described by the length measuring coordinate p along the line l_p, and two variables defined on l_p, $h_q(p)$ and $h_r(p)$

which represent the distance from l_p in $\vec{e}_q$- and $\vec{e}_r$-direction respectively. As a result, the position of a particle $\vec{x}$ can be defined as:

$$\vec{x}(\vec{x}_0, p) = \vec{x}_0 + p\vec{e}_p + h_q(p)\vec{e}_q + h_r(p)\vec{e}_r \tag{2.35}$$

where $\vec{x}_0$ is the starting point of the particle track.

The tangent to the track of a particle and the local velocities of that particle have to be codirectional for any point along the track. Considering the velocity field to be constant during the extrusion process, the shape of the particle track is defined by (van Rens et al., 1999c):

$$v_q = v_p \frac{dh_q}{dp}, \qquad v_r = v_p \frac{dh_r}{dp} \tag{2.36}$$

where $v_p = (\vec{v} \cdot \vec{e}_p)$, $v_q = (\vec{v} \cdot \vec{e}_q)$ and $v_r = (\vec{v} \cdot \vec{e}_r)$ represent the components of the velocities in the $\vec{e}_p$,$\vec{e}_q$,$\vec{e}_r$ reference frame. The equations in (2.36) can be solved on the line l_p to render the fields h_q and h_r, related to one particle. The boundary conditions that are supplied to (2.36) should ensure that the particle track is on the outflow surface as the aluminum leaves the die and can be expressed as:

$$h_q = 0, \quad h_r = 0 \qquad \text{for} \qquad \vec{x} = \vec{x}_0, \qquad \vec{x}_0 \in \Gamma_{die} \cap \Gamma_{outflow} \tag{2.37}$$

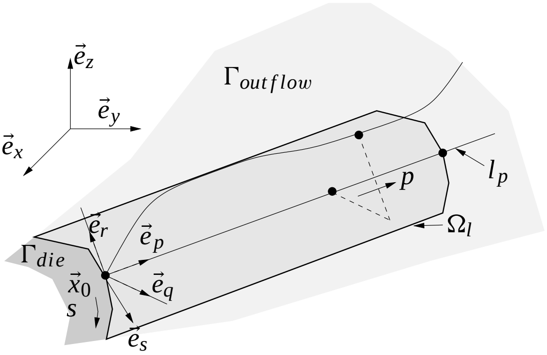


Figure 2.6: Surface definition of Ω_l.

Since it is not feasible in reality to solve the fields h_q and h_r separately for every particle track that starts at the die exit, equations (2.36) are redefined on the surface Ω_l that is constituted by the set of all the lines l_p (see also Figure 2.6):

$$\Omega_l = \left\{ \bigcup l_p ; l_p | \vec{x}_0 \in \Gamma_{die} \cap \Gamma_{outflow} \right\} \tag{2.38}$$

To do so, a local orthogonal basis is defined by $\vec{e}_p$ and $\vec{e}_s$ on Ω_l, along with their respective length measuring coordinates p and s. If the fields h_q and h_r are redefined on Ω_l to be of a function of both p and s, the position of a particle on the outflow surface can be written as:

$$\vec{x}(p, s) = \vec{x}_0(s) + p\vec{e}_p + h_q(p, s)\vec{e}_q + h_r(p, s)\vec{e}_r \tag{2.39}$$

Also, the local gradient operator on Ω_l is defined as $\vec{\nabla}_l = \vec{e}_p \frac{\partial}{\partial p} + \vec{e}_s \frac{\partial}{\partial s}$ and the vector field $\vec{w} = v_p \vec{e}_p$ is constructed. Then (2.36) can be rewritten in terms of the local coordinates p and s on Ω_l.

$$v_q = \vec{w} \cdot \vec{\nabla}_l h_q, \qquad v_r = \vec{w} \cdot \vec{\nabla}_l h_r \tag{2.40}$$

which are convection equations that have to be solved on Ω_l to render the distances h_q and h_r between the outflow surface and Ω_l for a given point $\vec{x}(p, s)$. The advantage of the formulation introduced here is that for a given velocity field the shape of outflow surface can be determined instantly by solving two convection equations. This removes the need for pseudo-time stepping schemes that are commonly used to determine the shape of a free surface flow.

2.7.3 Structure problem - Die shape

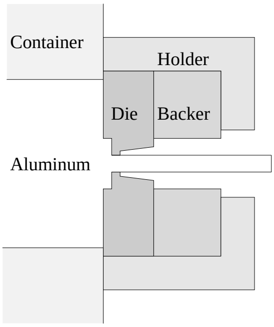


Figure 2.7: Configuration of the die package.

The interaction between the aluminum and the ram, the container and the die causes changes to the aluminum domain. However, only the deformation of the die package has a significant effect (see also Figure 2.7). The die package consists of a steel die and backer that are placed inside a steel die holder which in turn is placed against a bolster (not shown). All of these components deform to a certain extent under the extrusion pressures. In this work only the elastic behavior of the die and the backer are considered, because these are the only two that have been found to deform significantly in extrusion practice.

The elastic deformation of the die and the backer is modeled in a Lagrangian framework as a quasi-static process. For static processes and elastic material behavior the conservation of momentum law reduces to:

$$\vec{\nabla} \cdot \left(\boldsymbol{C} : \frac{1}{2} \left(\vec{\nabla} \vec{u}_s + (\vec{\nabla} \vec{u}_s)^c \right) \right) = \vec{0} \tag{2.41}$$

where the fourth order Hookean elasticity tensor $\boldsymbol{C}$ is fully defined by the Young's modulus, $E = 2.1 \cdot 10^{11} [\mathrm{Nm}^{-2}]$, and the Poisson's ratio, $\nu = 0.3$.

To solve (2.41) subject to the tractions that the aluminum exerts on the die, the boundary conditions that describe the interaction between the tool parts and the surrounding support have to be imposed. These conditions are based on the assumptions that contact between the different parts of the tooling is frictionless, and that the die holder is rigid.

Chapter 3

Finite element implementation

It is impossible to arrive at analytical solutions for the coupled partial differential equations that were derived in the previous chapter, especially for the complex geometries representative for aluminum extrusion. For this reason either the differential equations and/or the relevant geometry have to be simplified to such a level that analytical solutions can be obtained (see e.g. (Støren, 1993)), or the solutions of the differential equations have to be approximated using numerical methods. In this work a numerical method, the finite element method, will be used because the simplifications needed to attain an analytical solution are too constraining to obtain realistic solutions.

Within the finite element framework the partial differential equations that represent each of the distinguishable problems have to be spatially discretized. The discretization of the (aluminum flow) Stokes problem, is described in Section 3.1, where special attention is paid to the MINI element and the incorporation of friction through Lagrange multipliers. The discretization of the thermal convection-diffusion problem is handled in Section 3.2. There, the Streamline Upwind / Petrov-Galerkin (SUPG) method, necessary to stabilize the resulting system of equations, will be outlined. The discretization of the additional problems that result from the application of the three field approach to handle the varying aluminum shape will be discussed in Section 3.3. There, the structure problems and the mesh problem associated with the free-surface shape and the elastic die deformation, will each be treated separately.

After discretization each problem reduces to a system of algebraic equations. For the problem that describes the aluminum flow, the system consists of a non-linear Stokes problem that is coupled to a non-linear convection diffusion problem. This system of equations has to be decoupled and linearized before it can be solved. The iterative procedure that is used to decouple and linearize the system of equations is presented in Section 3.4. Also the solvers that are used to solve the linearized systems will be presented.

3.1 Discretizing the Stokes problem

The Stokes problem associated with extrusion, governed by (2.15) and (2.16), can be approximated using the classical Galerkin finite element method. To do so first the spaces of trial solutions $\mathcal{V}$ (for the velocity $\vec{v}$) and $\mathcal{P}$ (for the pressure p) are defined on the aluminum domain Ω with boundary Γ:

$$\mathcal{V} = \left\{ \vec{v} \in \left(H^1(\Omega) \right)^n \right\} \qquad \mathcal{P} = \left\{ p \in L^2(\Omega) \right\} \tag{3.1}$$

where $n \in \{1, 2, 3\}$ represents the dimension of the problem, and $H^1(\Omega)$ and $L^2(\Omega)$ are the standard Hilbert and Lebesque spaces, respectively. The variational form of the Stokes problem can then be written:

$$\begin{cases} \text{find } (\vec{v}, p) \in \mathcal{V} \times \mathcal{P} \text{ such that} & \\ \int_\Omega 2\eta \boldsymbol{D}_w : \boldsymbol{D}_v d\Omega - \int_\Omega \left(\vec{\nabla} \cdot \vec{w} \right) p \, d\Omega = \int_\Gamma \vec{w} \cdot \vec{s} \, d\Gamma & \forall \vec{w} \in \mathcal{V} \\ \int_\Omega q \left(\vec{\nabla} \cdot \vec{v} \right) d\Omega = 0 & \forall q \in \mathcal{P} \\ BC(\vec{v}, p) = 0 & \text{on } \Gamma \end{cases} \tag{3.2}$$

where $\boldsymbol{D}_w$ and $\boldsymbol{D}_v$ represent the rate of deformation tensor related to the fields $\vec{w}$ and $\vec{v}$, respectively. The variable $\vec{s} = (-p\boldsymbol{I} + 2\eta \boldsymbol{D}_v) \cdot \vec{n}$ denotes the surface tractions on Γ. The function $BC(\vec{v}, p)$ is used to introduce the appropriate boundary conditions on the boundary of the aluminum domain, Γ, and will be discussed further on in this chapter. Next, the finite dimensional subspaces $\mathcal{V}^h \subset \mathcal{V}$ and $\mathcal{P}^h \subset \mathcal{P}$ are defined on a family of discretizations $\mathcal{T}^k$ of the domain. To accommodate the meshing issues to be considered in the next chapter, $\mathcal{T}^k$ is supposed to contain elements, K, of triangular (2D) or tetrahedral (3D) form. Now the following subspaces are chosen:

$$\mathcal{V}^h = \left\{ \vec{v}^h \in \mathcal{V} \ ; \vec{v}^h|_K \in (P_1(K))^n, \forall K \in \mathcal{T}^k \right\} \tag{3.3}$$

$$\mathcal{P}^h = \left\{ p^h \in \mathcal{P} \ ; p^h|_K \in P_1(K), \forall K \in \mathcal{T}^k \right\} \tag{3.4}$$

where $P_k(K)$ denotes the space of polynomial functions of degree k or less and thus $P_1(K)$ denotes the space of linear functions on the element K. Now the problem formulation (3.2) can be written in terms of $\vec{v}^h \in \mathcal{V}^h$ and $p^h \in \mathcal{P}^h$:

$$\begin{cases} \text{find } (\vec{v}^h, p^h) \in \mathcal{V}^h \times \mathcal{P}^h \text{ such that} & \\ \int_\Omega 2\eta \boldsymbol{D}_{w^h} : \boldsymbol{D}_{v^h} d\Omega - \int_\Omega \left(\vec{\nabla} \cdot \vec{w}^h \right) p^h d\Omega = \int_\Gamma \vec{w}^h \cdot \vec{s} \, d\Gamma & \forall \vec{w}^h \in \mathcal{V}^h \\ \int_\Omega q^h \left(\vec{\nabla} \cdot \vec{v}^h \right) d\Omega = 0 & \forall q^h \in \mathcal{P}^h \\ BC(\vec{v}^h, p^h) = 0 & \text{on } \Gamma \end{cases} \tag{3.5}$$

where the weighting functions are chosen identically, in conjunction with the (Bubnov-) Galerkin method. It is well-known that, for the given choices of $\mathcal{V}^h$ and $\mathcal{P}^h$, the problem definition (3.5) is ill-posed because it does not satisfy the inf-sup or Babuška-Brezzi (LBB) stability condition (see e.g. (Bathe, 1982; Hughes, 1987) and references therein). As a result, spurious pressure modes will cause severe oscillations in the pressure solution and the velocity solution will lock in many cases, irrespective of the mesh size.

Two approaches exist that are generally employed to remedy the ill-posedness of (3.5) (Russo, 1996). The first method is to chose the polynomials that span the spaces of trial solutions $\vec{v}^h$ and p^h of different order. In that case the combination of these function spaces has to be chosen such that the LBB condition will be satisfied. An example is the triangular or tetrahedral P_2P_1 element implemented in standard FE packages, where the velocity is interpolated quadratically and the pressure linearly. However, these elements introduce a large number of degrees of freedom for each element which results in (actually unacceptably) large systems of equations to be solved.

The second approach is to stabilize the linear elements by modifying the Stokes equation through the inclusion of a certain amount of diffusivity to the formulation, without affecting the consistency of the solution (see (Franca et al., 1993) for an overview). An example of such an approach is the method indicated as Galerkin Least Squares (GLS) (Hughes et al., 1994). This allows to circumvent the compatibility requirements imposed on $\mathcal{V}^h$ and $\mathcal{P}^h$ by the LBB condition and thus enables the use of equal order trial spaces. A disadvantage of these methods is that they usually require the selection of a stabilizing parameter that is a function of the mesh size and the viscosity, which makes the method less suitable in case of strong variations in the element sizes and viscosities within Ω. However, in the formulation of the MINI element (Pierre, 1995), which is also a stabilized element, the stabilization parameter follows directly from the formulation. Therefore, the MINI element has been selected to be applied in this research.

3.1.1 MINI element

In the MINI element an enriched space of trial functions $\mathcal{V}_+^h$ is constructed out of the linear trial space $\mathcal{V}^h$, and the space of bubbles $\mathcal{V}_b^h$:

$$\mathcal{V}_+^h = \mathcal{V}^h \oplus \mathcal{V}_b^h \tag{3.6}$$

Bubbles are functions which are equal to zero at element boundaries and positive within the element to which they are associated. The bubble space is generally taken to be

$$\mathcal{V}_b^h = \left\{ \vec{v}_b^h \in \left(H_0^1(K) \right)^n ; \vec{v}_b^h|_K \in (B_k(K))^n , \forall K \in \mathcal{T}^k \right\} \tag{3.7}$$

where $B_k(K)$ represent the space of bubbles constructed out of polynomials of degree k or less and all basis functions $\phi \in B_k(K)$ satisfy:

$$\begin{cases} \phi(\vec{x}) > 0 & \forall \vec{x} \in K \setminus \partial K \\ \phi(\vec{x}) = 0 & \forall \vec{x} \in \partial K \\ \phi(\vec{x}) = 1 & \vec{x} = \text{geometrical center of } K \end{cases} \tag{3.8}$$

where ∂K indicates the boundary of the element K. The trial velocity solution $\vec{v}_+^h \in \mathcal{V}_+^h$ can now be split into a linear and a bubble contribution in a unique way:

$$\vec{v}_+^h = \vec{v}^h + \vec{v}_b^h \quad \forall(\vec{v}^h, \vec{v}_b^h) \in \mathcal{V}^h \times \mathcal{V}_b^h \tag{3.9}$$

Using this split of variables, the discretized form of (3.5) at the element level, can be written as:

$$\begin{cases} \text{find } (\vec{v}^h, \vec{v}_b^h, p^h) \in \mathcal{V}^h \times \mathcal{V}_b^h \times \mathcal{P}^h \text{ such that} \\ \int_K 2\eta \boldsymbol{D}_{w^h+w_b^h} : \boldsymbol{D}_{v^h+v_b^h} dK - \int_K \left(\vec{\nabla} \cdot (\vec{w}^h + \vec{w}_b^h)\right) p^h dK = \\ \qquad\qquad \int_{\partial K} (\vec{w}^h + \vec{w}_b^h) \cdot \vec{s}_k \, d\partial K \qquad \forall(\vec{w}^h, \vec{w}_b^h) \in \mathcal{V}^h \times \mathcal{V}_b^h \\ \int_K q^h \left(\vec{\nabla} \cdot (\vec{v}^h + \vec{v}_b^h)\right) dK = 0 \qquad\qquad \forall q^h \in \mathcal{P}^h \end{cases} \tag{3.10}$$

where $\vec{s}_k$ represents the surface tractions acting on $d\partial K$.

The operations that will be performed to simplify (3.10) are based on the property of the bubble space that for any constant tensor $\boldsymbol{A}$ on element K the following holds:

$$\int_K \boldsymbol{A} : \left(\vec{\nabla} \vec{v}_b^h\right) dK = 0 \tag{3.11}$$

which is a direct result of the fact that, by construction, $\vec{v}_b^h = \vec{0}$ on ∂K. This property, together with the observation that, as a result from the linear interpolation, $\vec{\nabla} \vec{v}^h$ is constant within an element leads to the orthogonality of the subspaces:

$$\int_K \vec{\nabla} \vec{v}^h : \vec{\nabla} \vec{v}_b^h dK = 0 \qquad \forall(\vec{v}^h, \vec{v}_b^h) \in \mathcal{V}^h \times \mathcal{V}_b^h \tag{3.12}$$

The orthogonality property (3.12) can be applied to simplify (3.10) if the viscosity (at the element level) is assumed constant for the moment:

$$\begin{cases} \text{find } (\vec{v}^h, \vec{v}_b^h, p^h) \in \mathcal{V}^h \times \mathcal{V}_b^h \times \mathcal{P}^h \text{ such that} \\ 2\eta \int_K \boldsymbol{D}_{w^h} : \boldsymbol{D}_{v^h} dK - \int_K \left(\vec{\nabla} \cdot \vec{w}^h\right) p^h dK = \int_{\partial K} \vec{w}^h \cdot \vec{s}_k \, d\partial K \\ \qquad\qquad\qquad\qquad \forall \vec{w}^h \in \mathcal{V}^h \\ 2\eta \int_K \boldsymbol{D}_{w_b^h} : \boldsymbol{D}_{v_b^h} dK - \int_K \left(\vec{\nabla} \cdot \vec{w}_b^h\right) p^h dK = \int_{\partial K} \vec{w}_b^h \cdot \vec{s}_k \, d\partial K \\ \qquad\qquad\qquad\qquad \forall \vec{w}_b^h \in \mathcal{V}_b^h \\ \int_K q^h \left(\vec{\nabla} \cdot \vec{v}^h\right) dK + \int_K q^h \left(\vec{\nabla} \cdot \vec{v}_b^h\right) dK = 0 \qquad \forall q^h \in \mathcal{P}^h \end{cases} \tag{3.13}$$

Integration of (3.13) for one element leads to a linear system of equations with the following structure:

$$\begin{bmatrix} \underline{\underline{K}} & \underline{\underline{0}} & \underline{\underline{L}} \\ \underline{\underline{0}} & \underline{\underline{K}}_b & \underline{\underline{L}}_b \\ -\underline{\underline{L}}^t & -\underline{\underline{L}}_b^t & \underline{\underline{0}} \end{bmatrix} \begin{bmatrix} \underline{v} \\ \underline{v}_b \\ \underline{p} \end{bmatrix} = \begin{bmatrix} \underline{f} \\ \underline{f}_b \\ \underline{0} \end{bmatrix} \tag{3.14}$$

where $\underline{v}$, $\underline{v}_b$, and $\underline{p}$ are columns containing the nodal velocities and pressures and $\underline{f}$, and $\underline{f}_b$ are columns with nodal loads. The matrices in (3.14) are related to the integrals over K. Since the bubble velocities, $\vec{v}_b$ are only defined inside the element considered, the system in (3.14) can be reduced by static condensation of the bubble degrees of freedom to render:

$$\begin{bmatrix} \underline{\underline{K}} & \underline{\underline{L}} \\ -\underline{\underline{L}}^t & \underline{\underline{L}}_b^t \underline{\underline{K}}_b^{-1} \underline{\underline{L}}_b \end{bmatrix} \begin{bmatrix} \underline{v} \\ \underline{p} \end{bmatrix} = \begin{bmatrix} \underline{f} \\ \underline{\underline{L}}_b^t \underline{\underline{K}}_b^{-1} \underline{f}_b \end{bmatrix} \tag{3.15}$$

The structure of this element matrix is similar to that obtained with other stabilization techniques. An added feature of this structure is that the zero diagonal matrix, which is still present in (3.14), has disappeared (the zero matrix tends to cause problems with iterative solvers).

It is obvious that the stabilization solely stems from the bubble. Therefore it is expected that the shape of the bubble will have some influence on the performance of the element. This influence is investigated next.

3.1.2 Choice of the bubble shape

Recently much effort has been invested in determining optimal shapes of bubbles. An important and promising result from these efforts is the residual free bubble (see e.g. (Brezzi et al., 1997; Franca and Farhat, 1995; Franca and Russo, 1997)). The essence of these bubbles is that their shape is not defined a priori but depends on the solution of a differential equation that is defined on each element. However, these residual-free bubbles are still in a very exploratory phase and are therefore not applied in this work.

The selection of the bubble will be restricted to invariant bubbles, i.e. bubbles of which the shape is determined a priori. Typical shapes of invariant bubbles that are within $B(K)$ are the cubic bubble, used in the standard MINI element, and the piecewise linear bubble. For these bubbles, the shape functions can be written in terms of the baricentric coordinates λ_i defined on K as (Pierre, 1995):

$$\phi_{cubic} = (n+1)^{n+1} \prod_{i=1}^{n+1} \lambda_i \tag{3.16}$$

$$\phi_{linear} = (n+1) \min_{i=1}^{n+1} \lambda_i \tag{3.17}$$

with n, as in the rest of this work, the dimension of the domain. Also, it should be noted that even though the ϕ_{cubic} is a quartic in 3D, the term cubic will be used to refer to these bubbles in 3D as well. It has been proven in (Pierre, 1995) that the space of bubbles can be extended with shapes that do not satisfy the first two requirements in (3.8) without losing the orthogonality property. An example of these bubble shapes is the quadratic bubble function:

$$\phi_{quadratic} = 2 - \sum_{i=1}^{n+1} (n+1)(\lambda_i)^2 \tag{3.18}$$

where it can easily be verified that in the geometrical center $\phi = 1$, but that the conditions $\phi > 0$ within the element, and $\phi = 0$ on the boundary, are violated.

The optimal bubble shape should render sufficient stabilization at an acceptable computational cost. For the Stokes problem an equilateral MINI element with a cubic bubble renders the optimal level of stabilization. However, in extrusion modeling grids will be applied with highly distorted elements and it has been proven in (Pierre, 1995) that the level of stabilization is reduced for distorted cubic MINI elements. For these distorted elements the performance of the quadratic element is expected to be better. Furthermore, to integrate the cubic bubbles analytically, integration rules that are exact for polynomials of order 6 in 3D and 4 in 2D, are required. Such integration rules are computationally rather time consuming. These problems may be overcome by applying lower order integration rules, but this may result in a poor performance of the stabilization and thus in degenerated pressure solutions.

To investigate the influence of using lower order bubbles and lower order integration the linear system (3.14) is constructed for four different 2D MINI elements. The first has a cubic bubble and is integrated using a four point Gauss rule which is only exact up to polynomials of degree 3. The second, third and fourth elements considered here have cubic, quadratic and linear bubbles respectively that are all integrated analytically. To compare the elements, a dimensionless Stokes problem with an external distributed body force (per unit volume, $\vec{f} \neq \vec{0}$) is considered:

$$\begin{cases} \vec{\nabla} \cdot \vec{v} = 0 \\ \vec{\nabla} \cdot (-p\boldsymbol{I} + 2\eta\boldsymbol{D}) = \vec{f} \end{cases} \tag{3.19}$$

Starting with a divergence free velocity field and a constant pressure field, according to:

$$\begin{cases} \vec{v}_{exact} = (1-y^2)\vec{e}_x + (1-x^2)\vec{e}_y \\ p_{exact} = 0 \end{cases} \tag{3.20}$$

the necessary body force can be derived to be (considering $\eta = 1$ for the dimensionless problem):

$$\vec{f} = \left(2\vec{e}_x + 2\vec{e}_y\right) \tag{3.21}$$

The appropriate Dirichlet boundary conditions are prescribed on the boundary of the domain for the velocities $\vec{v}$ and at the corners of the domain for the pressures p.

Two test cases are examined. For the first test a square domain, $-2 \leq (x, y) \leq 2$, is meshed with an unstructured mesh that mainly consists of equilateral elements (see Figure 3.1). The characteristic mesh length h, is defined as:

$$h^2 = \frac{\text{area}}{\text{number of elements}}$$

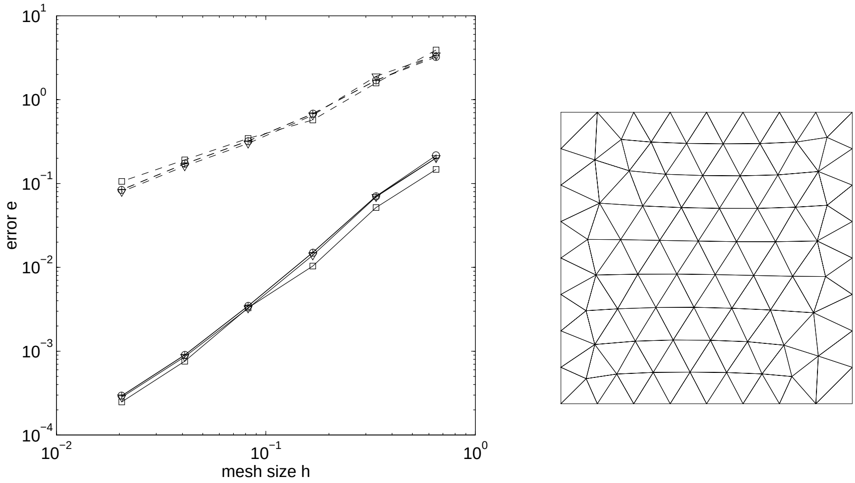


Figure 3.1: Left: Errors in the pressure e_p, dashed lines (- -), and in the velocity e_v, solid lines (—), for the numerically integrated cubic bubble (+) and for the analytically integrated cubic bubble (▽), quadratic bubble (○) and the linear bubble (□). Right: Example mesh for $h \approx 1/3$.

For several values of h, the maximal error for the velocity and the pressure field are computed according to:

$$e_v = \max(\|\vec{v} - \vec{v}_{exact}\|) \qquad e_p = \max(|p - p_{exact}|) \tag{3.22}$$

The results are plotted in Figure 3.1. For the second test the domain again spans the square $-2 \leq (x, y) \leq 2$. However, the mesh consists of stretched elements. The results are plotted in Figure 3.2.

For both types of meshes the expected linear convergence for the pressure and quadratic convergence for the velocity can be observed. In an absolute sense, the errors are slightly larger in the stretched mesh. In both cases the errors that are obtained using the quadratic analytically integrated bubble and the cubic numerically integrated bubble coincide. This is caused by the fact that the element matrices of the quadratic analytically integrated and the cubic numerically integrated bubble are exactly identical.

It can be observed that for the equilateral mesh the choice of the bubble has almost no influence on the prediction of the velocity field or on the pressure field even though the performance of the linear bubble slightly decreases for finer grids. For the stretched mesh the velocities are very similar for all elements. Yet, the prediction of the pressures is better for the element with the quadratic bubble. It is proven in (Pierre, 1995) that for the quadratic bubble this difference is more pronounced in 3D. The analytically integrated quadratic bubble appears to be numerically equivalent to the numerically integrated cubic bubble. Since the latter bubble is straightforward to implement in the finite element package that is used, it will be used in the following.

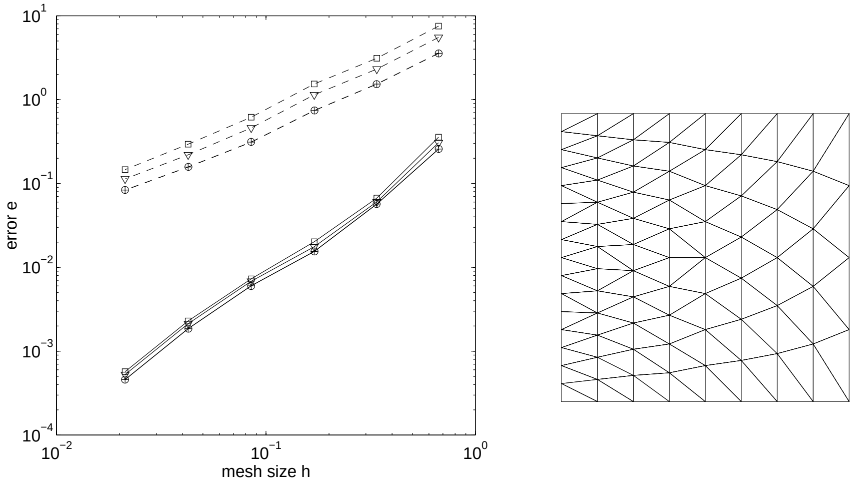


Figure 3.2: Left: Errors in the pressure e_p, dashed lines (- -) , and in the velocity e_v, solid lines (—), for the numerically integrated cubic bubble (+) and for the analytically integrated cubic bubble (▽), quadratic bubble (○) and the linear bubble (□). Right: Example mesh for $h \approx 1/3$.

3.1.3 Non-constant viscosity

In the derivation of (3.13) the viscosity, η, was assumed to be constant. However, for aluminum the viscosity is a function of the shear rate tensor, $\boldsymbol{D}$, and the temperature, T, which are both not constant at the element level. Consequently, the viscosity is not constant across an element, which implies that the orthogonality property (3.12) cannot be used to derive the simplified form (3.13) of the Stokes problem. To overcome this difficulty the viscosity is made constant within an element. For this purpose, two new piecewise constant solution fields are introduced to map the shear rate and the temperature onto solution fields that are constant within an element. The associated solution spaces read:

$$\mathcal{D}^h = \left\{ \boldsymbol{d}^h \in \left(L^2(\Omega)\right)^{n\times n}; \boldsymbol{d}^h|_K \in P_0(K), \forall K \in \mathcal{T}^k \right\} \tag{3.23}$$

$$\Theta^h = \left\{ \theta^h \in L^2(\Omega); \theta^h|_K \in P_0(K), \forall K \in \mathcal{T}^k \right\} \tag{3.24}$$

If the viscosity is now defined to be a function of the mapped quantities $\boldsymbol{d}^h$ and θ^h which are constant within one element, the problem formulated in (3.10) can be written as:

$$\begin{cases} \text{find } (\vec{v}^h, \vec{v}_b^h, p^h, \boldsymbol{d}^h, \theta^h) \in \mathcal{V}^h \times \mathcal{V}_b^h \times \mathcal{P}^h \times \mathcal{D}^h \times \Theta^h \text{ such that} \\ 2\eta(\boldsymbol{d}^h, \theta^h) \int_K \boldsymbol{D}_{w^h + w_b^h} : \boldsymbol{D}_{v^h + v_b^h} dK - \int_K \left(\vec{\nabla} \cdot \left(\vec{w}^h + \vec{w}_b^h\right)\right) p^h dK = \\ \qquad\qquad \int_{\partial K} (\vec{w}^h + \vec{w}_b^h) \cdot \vec{s}_k \, d\partial K \qquad \forall (\vec{w}^h, \vec{w}_b^h) \in \mathcal{V}^h \times \mathcal{V}_b^h \\ \int_K q^h \vec{\nabla} \cdot \left(\vec{v}^h + \vec{v}_b^h\right) dK = 0 \qquad \forall q^h \in \mathcal{P}^h \\ \int_K \boldsymbol{e}^h : (\boldsymbol{d}^h - \boldsymbol{D}_{v^h + v_b^h}) dK = 0 \qquad \forall \boldsymbol{e}^h \in \mathcal{D}^h \\ \int_K \varsigma^h (\theta^h - T) dK = 0 \qquad \forall \varsigma^h \in \Theta^h \end{cases} \tag{3.25}$$

Since $\boldsymbol{D}_{v^h}$ is constant within the element, application of the orthogonality relation (3.11) to the third equation in (3.25) gives $\boldsymbol{d}^h = \boldsymbol{D}_{v^h}$. Using this relation, (3.25) can be simplified to render the new form of (3.13):

$$\begin{cases} \text{find } (\vec{v}^h, \vec{v}_b^h, p^h, \theta^h) \in \mathcal{V}^h \times \mathcal{V}_b^h \times \mathcal{P}^h \times \Theta^h \text{ such that} \\ 2\eta(\boldsymbol{D}_{v^h}, \theta^h) \int_K \boldsymbol{D}_{w^h} : \boldsymbol{D}_{v^h} dK - \int_K \left(\vec{\nabla} \cdot \vec{w}^h\right) p^h dK = \\ \qquad\qquad \int_{\partial K} \vec{w}^h \cdot \vec{s} \, d\partial K \qquad \forall \vec{w}^h \in \mathcal{V}^h \\ 2\eta(\boldsymbol{D}_{v^h}, \theta^h) \int_K \boldsymbol{D}_{w_b^h} : \boldsymbol{D}_{v_b^h} dK - \int_K \left(\vec{\nabla} \cdot \vec{w}_b^h\right) p^h dK = \\ \qquad\qquad \int_{\partial K} \vec{w}_b^h \cdot \vec{s} \, d\partial K \qquad \forall \vec{w}_b^h \in \mathcal{V}_b^h \\ \int_K q^h \left(\vec{\nabla} \cdot \vec{v}^h\right) dK + \int_K q^h \left(\vec{\nabla} \cdot \vec{v}_b^h\right) dK = 0 \qquad \forall q^h \in \mathcal{P}^h \\ \int_K \varsigma^h (\theta^h - T) dK = 0 \qquad \forall \varsigma^h \in \Theta^h \end{cases} \tag{3.26}$$

Clearly, the last equation of (3.26) can be interpreted as averaging of the temperature within one element and can be performed as a preprocessing step. The remaining set of equations can then be written in the same form as (3.15).

3.1.4 Friction

To incorporate the effect of the frictional surface tractions, denoted by $\boldsymbol{\sigma} \cdot \vec{n}$ in (2.19)-(2.21), into the discretized system (3.26), these tractions have to be defined in terms of discretized entities. To this end one of three approaches is commonly pursued: a Lagrange multiplier method (Baaijens et al., 1986), an augmented Lagrange method (Heege and P.Alart, 1992), or a penalty method (Shimizu and Sano, 1995). Here the Lagrange multiplier method is adopted to avoid the introduction of a penalty parameter (Baaijens, 1987). In this method a Lagrange multiplier, $\vec{\lambda} = \boldsymbol{\sigma} \cdot \vec{n}$, which can be considered as a traction, is used to enforce the friction equations on the boundary Γ_{tool}. Furthermore the inequalities in (2.19)-(2.21) have to be reformulated into a set of equalities. This is done by the introduction of three distinct frictional states that allow the formulation of a set of equalities associated with each state.

Formulation of friction using three distinct states

To rewrite the inequalities in (2.19)-(2.21) into equalities, 3 distinct states are introduced, being stick, slip and release, comparable to the approach suggested in (Brekelmans, 1989). The friction states 'stick' and 'slip' follow directly from the complementarity condition. The friction state 'release', on the other hand, is introduced to cope with negative contact pressures. An unrealistic negative contact pressure implies that the aluminum would loose contact with the die and create a free surface that might regain contact again. To accurately capture this behavior a contact analysis would be required which introduces numerous numerical instabilities. Moreover, in extrusion practice it has been observed that, even if the aluminum releases from the die, the gap between the die and the aluminum is minimal. Therefore, in the separate state 'release', negative contact pressures are coped with by prescribing that the frictional tractions are zero and by neglecting the gap between the aluminum and the die. Neglecting this gap implies that the velocities of the aluminum remain tangential to the tooling surface.

Associated with each state, a set of equations is defined which enforce that the material either sticks to the tooling, slides along the tooling or 'releases' (i.e. slips along the tooling with zero friction tractions). With $\vec{n}$ the outward unit normal to the aluminum surface, a function $\vec{\mathcal{F}}(\vec{v}, \vec{\lambda})$ is defined according to:

$$\vec{\mathcal{F}}(\vec{v}, \vec{\lambda}) = \begin{cases} \text{stick} & \begin{cases} (\vec{v} - \vec{v}_{tool}) = \vec{0} \end{cases} \\ \text{slip} & \begin{cases} (\vec{v} - \vec{v}_{tool}) \cdot \vec{n}\vec{n} + \\ \qquad \vec{\lambda} \cdot (\boldsymbol{I} - \vec{n}\vec{n}) - \mu(\vec{\lambda} \cdot \vec{n}) \frac{\vec{v} - \vec{v}_{tool}}{\|\vec{v} - \vec{v}_{tool}\|} = \vec{0} \end{cases} \\ \text{release} & \begin{cases} (\vec{v} - \vec{v}_{tool}) \cdot \vec{n}\vec{n} + \vec{\lambda} \cdot (\boldsymbol{I} - \vec{n}\vec{n}) = \vec{0} \end{cases} \end{cases} \tag{3.27}$$

The function $\vec{\mathcal{F}}(\vec{v}, \vec{\lambda})$ is applied in an iterative procedure that proceeds as follows. At the start of the first iteration each point on the boundary is estimated to be in either the stick, the slip or the release state. For this distribution of states on the friction boundary, the solution field can be obtained by combining $\vec{\mathcal{F}}(\vec{v}, \vec{\lambda})$ with the equations that govern the flow of the aluminum. For the solution field computed using this set of equations, it is assessed whether for every point on the friction boundary the inequalities in (2.19)-(2.21) are satisfied. For each state these inequalities can be simplified according to:

$$\begin{cases} \text{stick} & \|\vec{\lambda} \cdot (\boldsymbol{I} - \vec{n}\vec{n})\| + \mu\vec{\lambda} \cdot \vec{n} \leq 0 \\ \text{slip} & (\vec{v} - \vec{v}_{tool}) \cdot (\vec{\lambda} \cdot (\boldsymbol{I} - \vec{n}\vec{n})) \leq 0 \\ \text{release} & -\vec{\lambda} \cdot \vec{n} \leq 0 \end{cases} \tag{3.28}$$

If the inequalities are not satisfied for a point on the boundary, the state of that point is adjusted accordingly and a new iteration is started. This process is continued until (2.19)-(2.21) are satisfied on the entire friction boundary.

Discretization of the Lagrange multipliers

Before $\vec{\mathcal{F}}(\vec{v}, \vec{\lambda})$ can be incorporated into the finite element formulation, the Lagrange multipliers that operate on the friction boundary, $\mathcal{T}^{k_\lambda}$, have to be discretized. The discretization has to be such that it does not introduce spurious modes or locking in the other degrees of freedom of the velocity field (Brezzi et al., 1996). It is therefore subject to similar stability considerations as the discretization of the pressure degrees of freedom in the mixed formulation. In (Bertrand et al., 1997) it has been suggested that, for piecewise linear integration of the Lagrange multiplier space, the distance between two adjacent points of $\mathcal{T}^{k_\lambda}$ should be larger than the edge length of the elements in $\mathcal{T}^k$ to avoid locking of the solution field. As a result, it is complicated to generate a discretization of the surface on which the Lagrange multiplier is applied. However, if each component of $\vec{\lambda}$ is approximated with a piecewise constant the distance between two points in $\mathcal{T}^{k_\lambda}$ is less constrained and can be equal to the edge length in $\mathcal{T}^k$ (Bertrand et al., 1997). This piecewise constant approximation is referred to as the virtual finite element method. In this method N control points, $\vec{x}_{c_i}, i = 1, 2, ..., N$ are introduced on the boundary. At these control points the friction equations are enforced locally (comparable to the collocation method). This leads to the following subspace of trial solutions for $\vec{\lambda}$:

$$\Lambda^h = \left\{ \vec{\lambda}^h \in \left(L^2(\Gamma_{tool}) \right)^n ; \vec{\lambda}^h|_{\vec{x}_{c_i}} \in P_0(\vec{x}_{c_i}), \forall i \in 1, ..., N \right\} \tag{3.29}$$

Now the appropriate weighting function that expresses the local operation of the Lagrange multiplier is a Dirac function $\delta(\vec{x})$ which defines the subspace:

$$\mathcal{Z}^h = \left\{ \vec{\zeta}^h = \sum_{i=1}^{N} \vec{z}_i \; \delta(\vec{x} - \vec{x}_{c_i}) \right\} \tag{3.30}$$

In principle friction should be modeled on the entire tooling surface Γ_{tool}. However, both numerical experiments and observations in extrusion practice indicate that the aluminum sticks to the ram and the container surface. Therefore, at these surfaces stick is assumed a priori and friction is only modeled on those parts of the tooling surface associated with the die. Then, using the enriched solution space for the velocities $\mathcal{V}_+^h$ (3.6) (rather than the linear space, $\mathcal{V}^h$), the linear space for the pressures $\mathcal{P}^h$ and the piecewise constant space for the

Lagrange multipliers Λ^h along with their spaces of weighting functions, the flow problem in (3.5) can be written as:

$$\begin{cases} \text{find } (\vec{v}^h_+, p^h, \vec{\lambda}^h) \in \mathcal{V}^h_+ \times \mathcal{P}^h \times \Lambda^h \text{ such that} & \\ \int_\Omega 2\eta \boldsymbol{D}_{w^h_+} : \boldsymbol{D}_{v^h_+} d\Omega - \int_\Omega \left(\vec{\nabla} \cdot \vec{w}^h_+\right) p^h d\Omega & \\ \qquad - \int_{\Gamma_{die}} \vec{w}^h_+ \cdot \vec{\lambda}^h d\Gamma = \int_{\Gamma_{tool} \setminus \Gamma_{die}} \vec{w}^h_+ \cdot \vec{s} d\Gamma & \forall \vec{w}^h_+ \in \mathcal{V}^h_+ \\ \int_\Omega q^h \left(\vec{\nabla} \cdot \vec{v}^h\right) d\Omega = 0 & \forall q^h \in \mathcal{P}^h \\ \int_{\Gamma_{die}} \vec{\zeta}^h \cdot \left(\vec{\mathcal{F}}(\vec{v}^h_+, \vec{\lambda}^h)\right) \partial\Gamma = 0 & \forall \vec{\zeta}^h \in \mathcal{Z}^h \\ \vec{v}^h = \vec{v}_{tool} & \text{on } \Gamma_{tool} \setminus \Gamma_{die} \end{cases} \tag{3.31}$$

where the surface integral over the surface tractions $\vec{s}$ is not taken over the die area or the outflow area. This is allowable because on the die surface the surface tractions are introduced by the friction Lagrange multipliers and on the outflow surface the surface tractions are zero, according to a stress-free outflow.

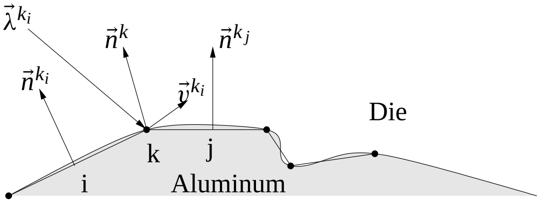


Figure 3.3: Vectors associated with node k on a discretized, curved boundary.

Since the friction boundary coincides with a domain boundary it is most convenient to locate the control points on the nodes of the boundary. This implies that the friction is enforced at the nodal level and therefore (3.27) and (3.28) have to be written in terms of nodal entities. The velocities in these relations are readily available but the surface normal $\vec{n}^k$ poses a serious problem (see also Figure 3.3). In node k the direction of the normal is not uniquely defined for the discretized boundary. In fact, if there are m element faces (for a 2D flow m=2) connected to node k, there are m normals defined in node k: $\vec{n}^{k_1}$, ..., $\vec{n}^{k_m}$, where $\vec{n}^{k_i}$ represents the face normal of the ith face connected to k. Only if $\vec{n}^{k_i} = \vec{n}^{k_j}, \forall i, j \in [1, m]$ the normal in k is uniquely defined. Since this is not the case for curved boundaries, as exemplified in Figure 3.3, the normal $\vec{n}^k$ has to be determined in an appropriate manner.

Defining the nodal normal on curved surfaces

To define the normal the new reconstruction method first suggested in (van Rens et al., 1998b) will be employed since it is basically independent of the discretization and renders accurate results on coarse grids. The method is based on using the local flow field to determine the direction of the local normal and can be explained as follows. The normal $\vec{n}^k$ in node k is defined to be a weighted average of the normals on the element faces connected to the node under consideration:

$$\vec{n}^k = \frac{\sum_{i=1}^{m} w_i \vec{n}^{k_i}}{\| \sum_{i=1}^{m} w_i \vec{n}^{k_i} \|} \tag{3.32}$$

with w_i the weight of each face. Usually these weights are determined by, for example, the surface area of the associated element faces (Brekelmans, 1989), or local smoothing techniques (Fourment et al., 1997). However, in the approach of (van Rens et al., 1998b) the weight factors, w_i are determined such that $\vec{n}^k$ becomes positioned in the plane that is perpendicular to the direction of the average velocity $\vec{v}$ in a thin boundary layer near k. This approach is based on the observation that when material is sliding along a surface, the velocity in a thin boundary layer is mainly perpendicular to the surface normal.

To determine w_i such that $\vec{n}^k$ is perpendicular to $\vec{v}$ in the vicinity of k an iterative method is used. Initially the weights, w_i are chosen equal for all faces connected to k. Then the velocity field is computed. Next the velocity field is used to update w_i in the following manner. For each element face e connected to node k with position vector $\vec{x}^k$, the velocity $\vec{v}^{p_i}$ is calculated in a material point p_i located at

$$\vec{x}^{p_i} = \vec{x}^k - \theta \vec{n}^{k_i} \tag{3.33}$$

where $\theta > 0$ represents the thickness of the boundary layer, which should be small compared to the local element size. From $\vec{v}^{p_i}$ the component orthogonal to $\vec{n}^{k_i}$ and thus tangential to the element face can be computed:

$$\vec{v}_{t_i}^{p_i} = (\boldsymbol{I} - \vec{n}^{k_i} \vec{n}^{k_i}) \cdot \vec{v}^{p_i} \tag{3.34}$$

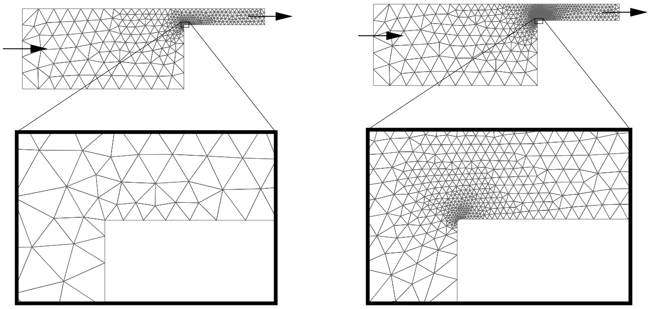
The weight w_i of each element normal is now defined to be the length of the tangential velocity in the sampling point:

$$w_i = \| \vec{v}_{t_i}^{p_i} \| \tag{3.35}$$

It can be verified that in a two dimensional flow ($m = 2$) this definition of w_i is correct since it leads to a normal in k that is perpendicular to the average velocity in the two sampling points. In a three dimensional configuration the proposed method is less transparent. The definition of w_i is then still based on the principle that the velocity in the boundary layer should be tangential to the boundary and therefore only the tangential component of the velocity should influence the direction of $\vec{n}^k$.

The benefits of the proposed definition of $\vec{n}^k$ can be illustrated by considering the flow field that can be computed for a 2D extrusion problem. Four velocity fields are computed on a relatively coarse mesh (750 elements, see Figure 3.4(a)) for the following definitions of the normal $\vec{n}^k$ in the node k on the edge at the entrance of the outflow channel:

(a) normal of upstream element face
(b) normal of downstream element face
(c) average direction of (a) and (b)
(d) the flow dependent normal discussed above



(a) Coarse mesh *(b)* Fine mesh

Figure 3.4: The meshes near the outlet

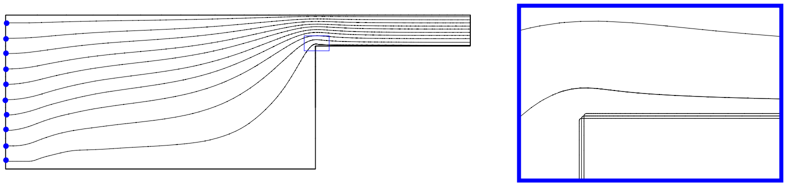
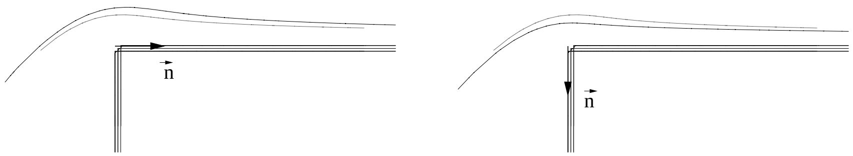


Figure 3.5: The particle tracks resulting from a fine mesh (triple lines indicate boundary)

The velocity fields are compared to a velocity field that is obtained on a very fine mesh where several elements were positioned along the arc at the entrance of the outflow channel (2500 elements, see Figure 3.4(b)). For the velocity fields on the fine and the coarse mesh particle tracks are calculated. The starting positions of the particles are chosen at the inflow as indicated with the dots in Figure 3.5. The particle tracks near the re-entrant corner of the fine mesh (Figure 3.5) are compared to the corresponding tracks in the coarse mesh in Figure 3.6.

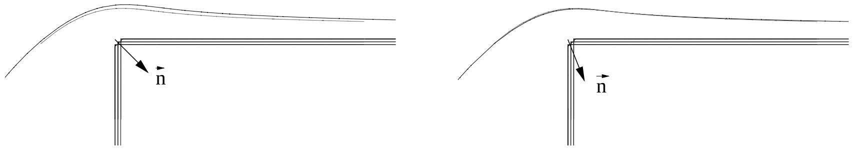
The results in Figure 3.6 indicate that the new method of defining the normal on a curved surface renders more accurate results for a coarse mesh. For the substitute normal that is based on the flow field, the position and shape of the particle track on the coarse mesh practically coincides with that of the fine mesh, while the other methods show significant deviations.

It is remarked that the extra computational costs that are introduced by the method are small. This is due to the fact that the iterative procedure, necessary to determine the direction of the normals, can be executed simultaneously within the iterative procedure that is necessary to solve the friction equations. In the computations the number of iterations that are necessary to solve the friction equations are only increased by 20% as a result of the concurrent determining of the direction of the normals.



(a) Upstream normal

(b) Downstream normal



(c) Average normal

(d) Flow normal

Figure 3.6: Particle tracks of coarse mesh (solid line) and fine mesh (dashed line)

3.2 Discretizing the temperature problem

As a point of departure, the thermal problem of extrusion, governed by (2.28), is approximated with the classical Galerkin finite element method. Analogous to the Stokes problem in the previous section, the variational form can be derived by introducing trial solution and weighting function spaces for the thermal problem which can be chosen to be equal, according to the (Bubnov)-Galerkin method. To reduce the meshing effort, the same spatial discretization $\mathcal{T}^k$, with equal order of interpolation, is employed for the flow and the temperature problems. Then, automatically the subspace in which the temperature trial solutions and weighting functions are defined, results:

$$\mathcal{T}^h = \left\{ T \in H^1(\Omega); \, T|_K \in P_1(K), \forall K \in \mathcal{T}^k \right\} \tag{3.36}$$

where, again, $P_1(K)$ denotes the space of linear functions on the element K. By employing $\mathcal{T}^h$, equation (2.28) can be written in the weak form:

$$\begin{cases} \text{find } T^h \in \mathcal{T}^h \text{ such that} \\ \rho c_p \int_\Omega s^h \vec{v} \cdot \vec{\nabla} T^h d\Omega - \lambda \int_\Omega \vec{\nabla} s^h \cdot \vec{\nabla} T^h d\Omega = \\ \qquad \int_\Omega s^h 2\eta \boldsymbol{D}_{v^h} : \boldsymbol{D}_{v^h} d\Omega - \int_\Gamma s^h \phi \, d\Gamma \ \forall s^h \in \mathcal{T}^h \\ T^h = T_{ram} \qquad\qquad \text{on } \Gamma_{ram} \end{cases} \tag{3.37}$$

where $\phi = \lambda \vec{\nabla} T \cdot \vec{n}$ represents the heat flux across the boundary Γ.

For systems described by (3.37) in which the convection term is dominant the solution of the temperature will display oscillations. This problem can be overcome by adding stabilizing terms to the formulation. One of the most commonly employed methods is the Streamline Upwind / Petrov-Galerkin (SUPG) method that was first suggested in (Brooks and Hughes, 1982).

3.2.1 Streamline Upwind / Petrov-Galerkin method

In the SUPG method the weighting functions are selected such that they do not lie in the same space as the trial functions. Within each element the weighting function $s^h \in \mathcal{T}^h$ is modified by adding an upwinding term. The formulation of the upwinding term is based on obtaining the same sort of diffusion that is present in a 1D convection-diffusion problem using a standard first order finite difference method with an Il'in upwind scheme (Segal, 1993). The modified weighting function $\tilde{s}^h$ then becomes:

$$\tilde{s}^h = s^h + \frac{h\alpha}{2} \frac{\vec{v} \cdot \vec{\nabla} s^h}{\|\vec{v}\|} \tag{3.38}$$

where h is the characteristic element size width, and α is a dimensionless upwinding parameter that has to be chosen appropriately. For triangular or tetrahedral elements the value of h in (3.38) is non-trivial. It has been proposed in (Mizukami, 1985) to define h as ”the longest distance a particle can travel through the element, given a velocity field $\vec{v}$ ”. This proposal for h is adopted in this work. Apart from h, the parameter α has to be defined. From the Il'in scheme it can be derived that the optimal value of α is given by

$$\alpha = \coth(\beta) - \frac{1}{\beta}, \text{ with, } \beta = \frac{\|\vec{v}\| h}{2(\lambda/\rho c_p)} \tag{3.39}$$

Since the evaluation of the hyperbolic co-tangent is numerically expensive many approximations exist. In this thesis the so-called critical approximation (Segal, 1993) is used, which reads $\alpha = 1 - 1/\beta$.

Employing the upwinding shape functions renders the following weak SUPG formulation for an element K with boundary ∂K:

$$\begin{cases} \text{find } T^h \in \mathcal{T}^h \text{ such that} \\ \rho c_p \int_K s^h \vec{v} \cdot \vec{\nabla} T^h dK - \lambda \int_K \vec{\nabla} s^h \cdot \vec{\nabla} T^h dK + \\ \quad \rho c_p \int_K \left(\frac{h\alpha}{2} \frac{\vec{v} \cdot \vec{\nabla} s^h}{\|\vec{v}\|} \right) \vec{v} \cdot \vec{\nabla} T^h dK - \lambda \int_K \vec{\nabla} \left(\frac{h\alpha}{2} \frac{\vec{v} \cdot \vec{\nabla} s^h}{\|\vec{v}\|} \right) \cdot \vec{\nabla} T^h dK - \\ \quad \int_K \left(\frac{h\alpha}{2} \frac{\vec{v} \cdot \vec{\nabla} s^h}{\|\vec{v}\|} \right) 2\eta \boldsymbol{D}_{v^h} : \boldsymbol{D}_{v^h} dK + \int_{\partial K} \left(\frac{h\alpha}{2} \frac{\vec{v} \cdot \vec{\nabla} s^h}{\|\vec{v}\|} \right) \phi_k d\partial K = \\ \qquad \int_K s^h 2\eta \boldsymbol{D}_{v^h} : \boldsymbol{D}_{v^h} dK - \int_{\partial K} s^h \phi_k d\partial K \qquad\qquad \forall s^h \in \mathcal{T}^h \end{cases} \tag{3.40}$$

where ϕ_k represents the heat flux across the element boundary.

3.3 Discretizing the changing boundary problems

Each of the three problems that govern the changing boundary of the aluminum domain are discretized using the finite element method. The discretization of the first problem, the fluid problem, has been discussed in Section 3.1. The discretization of the second problem, the structure problem, is discussed in detail, where the discretization of the free surface shape problem and the deforming die problem are dealt with in Section 3.3.1 and Section 3.3.2, respectively. The discretization of the third problem, the deforming fluid mesh problem, which is essentially a standard elastic problem, will be handled briefly, for completeness, in Section 3.3.3.

Each of the problems is discretized such that the nodes for each domain coincide on the interface boundaries, i.e. on the outflow and the die boundaries. Also the discretization is realized such that the interpolation of the stresses and the displacements are of the same order on both sides of the interface boundary. This discretization makes it possible to easily map solution fields between the problems.

3.3.1 Structure problem - Free surface

In the free surface problem the two independent convection equations in (2.40) have to be solved on Ω_l to obtain the fields $h_q(p,s)$ and $h_r(p,s)$ which define the distance between the domain Ω_l and the aluminum outflow surface, $\Gamma_{outflow}$ for any point $\vec{x}(p,s)$ on Ω_l. It would be most convenient if the surface mesh on $\Gamma_{outflow}$ could be used to solve (2.40), for two reasons. First the velocity field $\vec{v}$, and thus the vector field $\vec{w}$ needed in the convection equations, is readily available in the nodes of $\Gamma_{outflow}$. Second, no separate mesh would have to be generated on Ω_l. However, Ω_l generally does not coincide with $\Gamma_{outflow}$, which poses a problem.

This problem is remedied by defining a new domain Ω_n that coincides with $\Gamma_{outflow}$ and by rewriting (2.40) in terms of the variables that define Ω_n (see also Figure 3.7). On the surface Ω_n the local, orthogonal unit vectors $\vec{e}_\zeta$ and $\vec{e}_\xi$ are defined, along with length measuring coordinates ζ and ξ in their respective directions. The axes are oriented such that $\vec{e}_p \cdot \vec{e}_\xi = 0$ and $\vec{e}_s \cdot \vec{e}_\zeta = 0$. This implies that $\vec{e}_\zeta$ has the same direction as the projection of

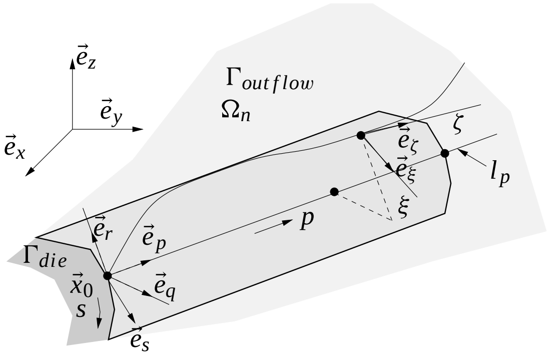


Figure 3.7: Surface definition of Ω_n.

$\vec{e}_p$ onto Ω_n. The surface velocity $\vec{w} = v_p \vec{e}_p$ on Ω_l, is projected onto Ω_n to render:

$$\vec{w}_n = \vec{w} \cdot (\vec{e}_\zeta \vec{e}_\zeta + \vec{e}_\xi \vec{e}_\xi) = v_p (\vec{e}_p \cdot \vec{e}_\zeta) \vec{e}_\zeta \tag{3.41}$$

Then, it is easily verified that (2.40) can be defined on Ω_n through the application of the local gradient operator $\vec{\nabla}_n = \vec{e}_\zeta \frac{\partial}{\partial \zeta} + \vec{e}_\xi \frac{\partial}{\partial \xi}$ along the lines:

$$\vec{w}_n \cdot \vec{\nabla}_n h_q = v_q (\vec{e}_p \cdot \vec{e}_\zeta)^2 \qquad \vec{w}_n \cdot \vec{\nabla}_n h_r = v_r (\vec{e}_p \cdot \vec{e}_\zeta)^2 \tag{3.42}$$

where the equivalence between (2.40) defined on Ω_l and (3.42) defined on Ω_n can be verified by applying (3.41) to (3.42) and by observing that $\frac{\partial}{\partial \zeta} = \frac{\partial}{\partial p} \frac{\partial p}{\partial \zeta} = \frac{\partial}{\partial p} (\vec{e}_p \cdot \vec{e}_\zeta)$ holds for the given definitions of $\vec{e}_\zeta$ and $\vec{e}_\xi$.

The discretization of Ω_n, $\mathcal{T}^{k_n}$, is constructed such that it is conform with the aluminum mesh on $\Gamma_{outflow}$. Also, the space of trial solutions for h_q and for h_r on Ω_n is constructed out of linear polynomials:

$$\mathcal{Q}^h = \left\{ h^h \in H^1(\Omega_n);\, h^h|_K \in P_1(K), \forall K \in \mathcal{T}^{k_n} \right\} \tag{3.43}$$

Using this space of trial solutions the weak formulation of (3.42) can be written in the following form:

$$\begin{cases} \text{find } h_q^h, h_r^h \in \mathcal{Q}^h \times \mathcal{Q}^h \text{ such that} & \\ \int_{\Omega_n} s_q^h \vec{w}_n \cdot \vec{\nabla}_n h_q = \int_{\Omega_n} s_q^h v_q \left(\vec{e}_p \cdot \vec{e}_\zeta\right)^2 \, d\Omega & \forall s_q^h \in \mathcal{Q}^h \\ \int_{\Omega_n} s_r^h \vec{w}_n \cdot \vec{\nabla}_n h_r = \int_{\Omega_n} s_r^h v_r \left(\vec{e}_p \cdot \vec{e}_\zeta\right)^2 \, d\Omega & \forall s_r^h \in \mathcal{Q}^h \\ h_q = h_r = 0 & \text{on } \Gamma_{die} \cap \Gamma_{outflow} \end{cases} \tag{3.44}$$

Because (3.44) is a convection problem, the same stability issues as those in the convection-diffusion temperature problem arise. Therefore, (3.44) is stabilized through the application of the SUPG method that has been described in Section 3.2.1 with an $\alpha = 1$.

The advantage of formulating the free surface in terms of the two independent convection problems in (3.44), as has been introduced here, is that the solutions of these convection problems renders the shape of the outflow surface instantly, without the need for (pseudo-) time integration. Also, the problem is defined on a surface which implies that the resulting system of equations will be small and, therefore, numerically inexpensive to solve.

3.3.2 Structure problem - Die deflection

In the discretization of the differential equations that govern the deflection of the die and the backer, the frictionless interaction between the various parts of the tooling requires special attention. The frictionless interaction between the die, the backer and the holder on the die-backer interface Γ_{db} and the holder interface Γ_h can be expressed as:

$$(\boldsymbol{I} - \vec{n}_b\vec{n}_b)\vec{s}_d = (\boldsymbol{I} - \vec{n}_b\vec{n}_b)\vec{s}_b = \vec{0} \qquad \forall\vec{x} \in \Gamma_{db} \cup \Gamma_h \tag{3.45}$$

where $\vec{n}_d$ represents the normal on the die surface, $\vec{n}_b$ the normal on the backer surface, $\vec{s}_d = \boldsymbol{\sigma}_d \cdot \vec{n}_d$ the surface tractions on the die, and $\vec{s}_b = \boldsymbol{\sigma}_b \cdot \vec{n}_b$ the surface tractions on the backer. It should be noted that, based on the geometry of the die and the backer, $\vec{n}_d = -\vec{n}_b$ holds on Γ_{db}, which makes it possible to express the requirement that the die and the backer should not penetrate each other as:

$$(\vec{u}_d - \vec{u}_b) \cdot \vec{n}_d\vec{n}_d = \vec{0} \qquad \forall\vec{x} \in \Gamma_{db} \tag{3.46}$$

Also, the holder is assumed to be rigid which reduces (3.46) to:

$$\vec{u}_d \cdot \vec{n}_d\vec{n}_d = \vec{u}_b \cdot \vec{n}_b\vec{n}_b = \vec{0} \qquad \forall\vec{x} \in \Gamma_h \tag{3.47}$$

For the discretization of the die deformation problem, the tooling domain is split into the two relevant domains, the die Ω_d with boundary Γ_d, and the backer Ω_b with boundary Γ_b, (see also Figure 2.7). The discretizations of these domains are performed such that the meshes are conforming at the interface surfaces between the aluminum and the die, Γ_{alu}, and between the die and the backer, Γ_{db}. This facilitates the mapping of stresses and displacements between the various domains. Consequently, a linear interpolation of the displacements of the die, $\vec{u}_d$, and the backer $\vec{u}_b$ is used:

$$\mathcal{U}^h = \left\{\vec{u}^h, \in \left(H^1(\Omega_d \cup \Omega_b)\right)^n ; \left(\vec{u}^h|_K\right) \in (P_1(K))^n , \forall K \in \mathcal{T}_h^d \cup \mathcal{T}_h^b\right\} \tag{3.48}$$

The constraints given in (3.45) and (3.47) are respectively Neumann and Dirichlet type boundary conditions that can be applied in a straightforward manner. However, to impose the constraints expressed in (3.45) the introduction of surface tractions that enforce (3.45) on Γ_{db} is required. These surface tractions are applied through the use of a Lagrange multiplier $\vec{\lambda}$ which has to be discretized as well. The discretization of the Lagrange multiplier is performed in the same manner as discussed in Section 3.1.4, i.e. by introducing a control point with position $\vec{x}_c$ for every surface node in Γ_{db}, and by approximating the components of the force vector $\vec{\lambda}$ by piecewise constants in each control point. Then the weak form of the variational problem now becomes:

$$\begin{cases} \text{find } (\vec{u}_d^h, \vec{u}_b^h, \vec{\lambda}^h) \in \mathcal{U}^h \times \mathcal{U}^h \times \mathcal{L}^h \text{ such that} \\ \int_{\Omega_d} \boldsymbol{\varepsilon}^h_{w_d} : \boldsymbol{C} : \boldsymbol{\varepsilon}^h_{u_d} d\Omega = \int_{\Gamma_{alu}} \vec{w}_d^h \cdot (p\boldsymbol{I} - 2\eta \boldsymbol{D})\, \vec{n}_d d\Gamma + \\ \quad \int_{\Gamma_{db}} \vec{w}_d^h \cdot \vec{\lambda}^h d\Gamma + \int_{\Gamma_d \setminus (\Gamma_{alu} \cup \Gamma_{db})} \vec{w}_d^h \cdot \vec{s}_d \Gamma \qquad \forall \vec{w}_d^h \in \mathcal{U}^h \\ \int_{\Omega_b} \varepsilon^h_{w_b} : \boldsymbol{C} : \varepsilon^h_{u_b} d\Omega = \\ \quad \int_{\Gamma_{db}} \vec{w}_b^h \cdot \vec{\lambda}^h d\Gamma_{db} + \int_{\Gamma_d \setminus \Gamma_{db}} \vec{w}_b^h \cdot \vec{s}_b \Gamma \qquad \forall \vec{w}_b^h \in \mathcal{U}^h \\ \int_{\Gamma_{db}} \vec{\zeta}^h \cdot \left((\vec{u}_d^h - \vec{u}_b^h) \cdot \vec{n}_d \vec{n}_d \right) \, d\Gamma = 0 \qquad \forall \vec{\zeta}^h \in \mathcal{Z}^h \\ \vec{u}_d^h \cdot \vec{n} = \vec{u}_b^h \cdot \vec{n} = 0 \qquad \text{on } \Gamma_h \end{cases} \tag{3.49}$$

where $\boldsymbol{\varepsilon}_u = \frac{1}{2}\left(\vec{\nabla}\vec{u} + (\vec{\nabla}\vec{u})^c\right)$ and $\mathcal{L}^h$, $\mathcal{Z}^h$, $\vec{\lambda}^h$ and $\vec{\zeta}^h$ have been introduced in Section 3.1.4.

3.3.3 Mesh problem

As mentioned before, the mesh of the aluminum domain is considered to be an elastic pseudo-structure. Since the mesh domain Ω_m with boundary Γ_m coincides with the fluid domain Ω, the discretization of the mesh problem $\mathcal{T}^{k_m}$ is taken identical to the discretization $\mathcal{T}^k$ of the fluid domain. The space of trial solutions $\mathcal{U}^h$ for the mesh displacements $\vec{u}_m$ on Ω_m is constructed out of linear polynomials:

$$\mathcal{U}^h = \left\{ \vec{u}_m^h \in \left(H^1(\Omega_m) \right)^n ; \vec{u}_m^h|_K \in (P_1(K))^n \,, \forall K \in \mathcal{T}^{k_m} \right\} \tag{3.50}$$

The material properties of an elastic structure are fully determined by the Young's modulus E and the Poisson's ratio ν. Since the pseudo-structure has no physical relevance the actual values of Young's modulus and Poisson's ratio do not really matter; here $E = 1[\text{N m}^{-2}]$ and $\nu = 0$ are used. Then, considering that the mesh displacements at the interface boundary, $\Gamma_{interface} = \Gamma_{outflow} \cup \Gamma_{die}$, are prescribed by the structural displacements, of the outflow surface and the die $\vec{u}_s$, the weak form of the mesh problem becomes:

$$\begin{cases} \text{find } \vec{u}_m^h \in \mathcal{U}^h \text{ such that} \\ \int_{\Omega_m} \boldsymbol{\varepsilon}^h_{w_m} : \boldsymbol{\varepsilon}^h_{u_m} d\Omega = \int_{\Gamma_m} \vec{w}_m^h \cdot \vec{s}_m \, d\Gamma \qquad \forall \vec{w}_m^h \in \mathcal{U}^h \\ \vec{u}_m^h = \vec{0} \qquad \text{on } \Gamma_m \setminus \Gamma_{interface} \\ \vec{u}_m^h = \vec{u}_s \qquad \text{on } \Gamma_{interface} \end{cases} \tag{3.51}$$

where $\boldsymbol{\varepsilon}_{u_m} = \frac{1}{2}\left(\vec{\nabla}\vec{u}_m + (\vec{\nabla}\vec{u}_m)^c\right)$ is the elastic strain tensor and $\vec{s}_m = \boldsymbol{\varepsilon}_{u_m} \cdot \vec{n}_m$, with $\vec{n}_m$ the outward pointing normal, represent the surface tractions acting on Γ_m. The advantage of taking $\nu = 0$ is that after elaboration of (3.51) the individual components of the mesh displacements are uncoupled and can be solved separately.

3.4 Solution procedure

The discretized partial differential equations derived in the previous sections each render a, possibly non-linear, system of algebraic equations. The solution of each system is dependent on the solution of the other systems, which is handled through the staggered approach depicted in Figure 2.4. The system of equations related to the fluid problem is non-linear and has to be linearized by applying a staggered approach as well. In the following, first the staggered approach that is applied will be discussed, followed by the linear solvers which are applied to solve the linearized systems.

3.4.1 Staggered approach

A staggered approach consists of solving the system of equations related to one problem while assuming that the solution of the other problems does not change. This decouples the system of equations enabling the independent solution of each system. As a result, the solution of each of the systems can be performed using the method that is best suited for that system, which makes the staggered approach very appealing.

The staggered approach that is used here is based on the scheme depicted in Figure 2.4 and can be summarized as follows (see also Figure 3.8). First an estimate of the shape of the aluminum domain Ω_0 is made and this is discretized to obtain a mesh $\mathcal{T}_0^k$ on Ω_0. With this mesh the velocity of the aluminum is computed by solving the fluid problem. The solution of the fluid problem renders the velocity on the outflow surface and the surface tractions that the aluminum exerts on the die. With these velocities and tractions known, the updated shape of the outflow surface and the updated shape of the die can be computed by solving the structure problems. The changing shape of the outflow surface and the deflection of the die define the displacements of the surface of the aluminum and thus the shape of the new aluminum domain Ω_1. With the displacements of the aluminum surface known, the displacements of the nodes in the mesh can be computed by solving the mesh problem. This renders the new mesh $\mathcal{T}_1^k$ on Ω_1, which can be used to enter the fluid problem again.

The system of equations related to the fluid problem consists of a flow problem coupled to a temperature problem because the viscosity $\eta(\boldsymbol{D}, T)$ is a function of both the shear rate and the temperature field. The coupling between the flow problem and the temperature problem is dealt with by applying a staggered approach within the solution of the fluid problem. First the temperature is held constant and the flow field is solved. Next the flow field is held constant and the temperature field is solved. The solution of the flow problem for a given temperature has to be obtained iteratively because both the viscosity η and the friction function $\vec{\mathcal{F}}$ are non-linear functions of the flow field. To this end a Picard scheme is employed that uses the flow field of the previous iteration to compute the viscosity and the friction function that is used in the current iteration. Also the solution of the temperature problem for a given flow field has to be obtained iteratively because the viscosity is a nonlinear function of the temperature. To solve the temperature field, again a Picard scheme is used.

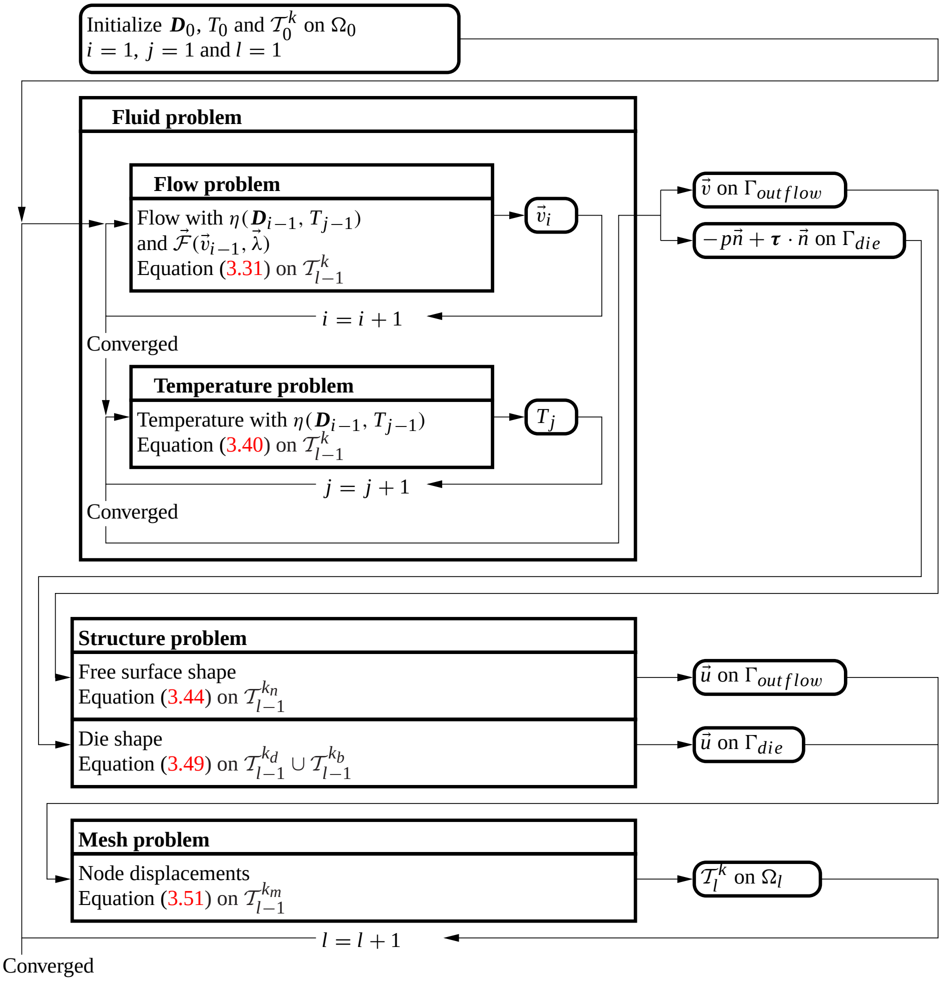

Initialize $\boldsymbol{D}_0$, T_0 and $\mathcal{T}_0^k$ on Ω_0
$i = 1$, $j = 1$ and $l = 1$
Fluid problem
Flow problem
Flow with $\eta(\boldsymbol{D}_{i-1}, T_{j-1})$
and $\vec{\mathcal{F}}(\vec{v}_{i-1}, \vec{\lambda})$
Equation (3.31) on $\mathcal{T}_{l-1}^k$
$\vec{v}_i$
$i = i + 1$
Converged
Temperature problem
Temperature with $\eta(\boldsymbol{D}_{i-1}, T_{j-1})$
Equation (3.40) on $\mathcal{T}_{l-1}^k$
T_j
$j = j + 1$
Converged
$\vec{v}$ on $\Gamma_{outflow}$
$-p\vec{n} + \boldsymbol{\tau} \cdot \vec{n}$ on Γ_{die}
Structure problem
Free surface shape
Equation (3.44) on $\mathcal{T}_{l-1}^{k_n}$
Die shape
Equation (3.49) on $\mathcal{T}_{l-1}^{k_d} \cup \mathcal{T}_{l-1}^{k_b}$
$\vec{u}$ on $\Gamma_{outflow}$
$\vec{u}$ on Γ_{die}
Mesh problem
Node displacements
Equation (3.51) on $\mathcal{T}_{l-1}^{k_m}$
$\mathcal{T}_l^k$ on Ω_l
$l = l + 1$
Converged

Figure 3.8: The iterative procedure.

3.4.2 Linear solver

Each of the linearized systems of algebraic equations that results from the staggered approach can be written in a matrix-vector format:

$$\underline{\underline{A}}\;\underline{x} = \underline{b} \tag{3.52}$$

where the vector $\underline{x}$ is related to the entities that have to be solved (e.g. nodal velocities, pressures, displacements), the matrix $\underline{\underline{A}}$ is constructed from the element contributions, and the vector $\underline{b}$ follows from external loads on the element.

For the linearized systems of equations associated with aluminum extrusion the solutions have to be obtained using a linear iterative solver because the number of degrees of freedom is of such magnitude ($\mathcal{O}(10^5 - 10^6)$) that direct linear solvers would require unrealistic amounts of memory and solution time. The robustness and effectiveness of an iterative solver decreases when the size and/or the condition number of the system matrix, $\kappa(\underline{\underline{A}}) = \|\underline{\underline{A}}\| \, \|\underline{\underline{A}}^{-1}\|$, increases. The condition number of $\underline{\underline{A}}$, will increase when the elements that are used to assemble $\underline{\underline{A}}$ become more distorted or when the material properties display larger variations.

In the following the attention will be focused on the linearized system of equations that is related to the flow problem because this is the most challenging to solve. This system has the highest number of degrees of freedom, the mesh of the domain is very distorted, and the variations in the material parameters over the domain are severe. If friction is omitted for ease of notation, the system of equations associated with the flow problem can be written as (see Section 3.1.1):

$$\begin{bmatrix} \underline{\underline{K}} & \underline{\underline{L}}^t \\ -\underline{\underline{L}} & \underline{\underline{S}} \end{bmatrix} \begin{bmatrix} \underline{v} \\ \underline{p} \end{bmatrix} = \begin{bmatrix} \underline{f} \\ \underline{g} \end{bmatrix} \tag{3.53}$$

This system can be solved using a one-level or a two-level strategy. In the one-level strategy the velocity field $\underline{v}$ and the pressure field $\underline{p}$ are solved simultaneously. Examples of this strategy are the integrated method in which the original system is solved and the penalty function method, in which a small amount of compressibility is introduced to enable the elimination of the pressure degrees of freedom at the element level, after which a smaller, positive definite system can be obtained (see e.g. (Cuvelier et al., 1986)). In the two-level strategy the pressure and the velocity field are solved separately. The velocities are written as:

$$\underline{v} = \underline{\underline{K}}^{-1}(\underline{f} - \underline{\underline{L}}^t \underline{p}) \tag{3.54}$$

Then the relation for the pressures can be written as:

$$\left(\underline{\underline{L}} \, \underline{\underline{K}}^{-1} \underline{\underline{L}}^t + \underline{\underline{S}} \right) \underline{p} = \underline{g} + \underline{\underline{L}} \, \underline{\underline{K}}^{-1} \underline{f} \tag{3.55}$$

After the pressures have been solved, the velocities are computed through back substitution. Examples of this strategy are the two-level penalty function method and the pressure-matrix method ((Quarteroni and Valli, 1994)).

In this work the integrated approach is used because it has been proven to converge faster than the two-level pressure matrix method (Ramage and Wathen, 1994; Wille, 1994). Also the integrated method renders better results than the two-level penalty method (Silvester and Atanga, 1992). The one-level penalty method cannot be used in combination with the MINI element because this method is based on the static condensation of the pressure degrees of freedom at the element level. This condensation can only be performed if the pressure field is approximated with a trial space that is discontinuous across the element boundaries, which is not the case for the MINI element. An additional disadvantage of a method based on a penalty parameter is that the condition number of the matrix becomes very high due to the very small numbers that appear in the $\underline{\underline{S}}$-matrix as a result of the added diffusivity. As discussed above this higher condition number has a negative influence on the efficiency of the iterative solver.

Iterative solvers can be split into solvers based on Arnoldi's method and those based on Lanczos' method. Since the matrix in (3.53) is square and positive definite, but non-symmetric, a considerable restriction is posed on the iterative solvers that can be applied. Of the remaining solvers based on Arnoldi's method the Generalized Minimum Residual (GMRES) method requires the least computation time for the same system compared to other solvers based on this method (e.g. Full Orthogonalization Process (FOM) and Direct Quasi GMRES (DQGMRES)) (Saad, 1996). A disadvantage of these methods is that they require a large amount of memory and computational effort because for every iteration an additional solution vector has to be stored and an additional vector-vector multiplication has to be performed. This introduces a serious problem for large systems since the number of iterations required increases with the number of degrees of freedom in the system. This problem can be overcome by restarting GMRES after a fixed number of iterations. However, convergence of the restarted GMRES method is not guaranteed.

An alternative to the solvers based on Arnoldi's method are the iterative solvers based on Lanczos' method. Examples of these methods are the Bi Conjugate Gradient (BICG) method, the Bi Conjugate Gradient Stabilized (BICGSTAB) method which is an improvement of BICG, and the Transpose-Free Quasi Minimal Residual (TFQMR) method which is closely related to DQGMRES. A disadvantage of these methods is that they are less robust. However, for large systems this is compensated by the fact that they need to store only a few vectors and require less computational effort per iteration. Numerical experiments with a number of test problems using the different solvers have indicated that BICGSTAB will require the least amount of computational time (Saad, 1996). Therefore the BICGSTAB method is adopted in this work.

The robustness and convergence rate of the BICGSTAB solver is improved by using a preconditioner. The matrix-vector equation is then multiplied with a preconditioning matrix $\underline{\underline{P}}$ to render:

$$\underline{\underline{P}}\,\underline{\underline{A}}\,\underline{x} = \underline{\underline{P}}\,\underline{b} \tag{3.56}$$

after which this system is solved. The purpose of this preconditioning is to reduce the condition number of the linear system and thus increase the convergence of the iterative solver. The ideal preconditioning matrix is $\underline{\underline{P}} = \underline{\underline{A}}^{-1}$, because this matrix renders the exact solution immediately. In reality $\underline{\underline{P}}$ is chosen such that it is a good approximation of $\underline{\underline{A}}^{-1}$. The three possibilities for $\underline{\underline{P}}$ discussed here are diagonal, Gauss Seidel and ILU(0) preconditioning matrices. The interested reader is referred to (Saad, 1996) for a complete overview of the existing possibilities.

In diagonal preconditioning the matrix is chosen such that $\underline{\underline{P}} = \left(\mathrm{diag}(\underline{\underline{A}})\right)^{-1}$ which is very cheap to compute but is not a very good approximation of $\underline{\underline{A}}^{-1}$. A slightly better preconditioning of $\underline{\underline{A}}$ is obtained by performing one or several iterations of a Gauss Seidel iterative solver before applying the real solver. The advantage of the Gauss Seidel solver over the solvers based on Arnoldi's or Lanczos' method is that it is computationally cheap per iteration; however the rate of convergence is much smaller. The ILU(0), Incomplete LU preconditioner is based on a LU-factorization of $\underline{\underline{A}}$ where the entries in the lower triangular $\underline{\underline{L}}$ and the upper triangular $\underline{\underline{U}}$ matrices are only filled if the corresponding entries in $\underline{\underline{A}}$ are nonzero. This method is more expensive than the other preconditioning methods discussed here but it renders a better approximation of $\underline{\underline{A}}^{-1}$. Numerical experiments indicate that the diagonal and the Gauss Seidel precondition do not decrease the condition number of the flow problem sufficiently to guarantee convergence of the BICGSTAB solver. Therefore the ILU(0) method is used for the flow problem, even though it is more expensive.

In summary, the flow problem is solved by applying an integrated approach in which the BICGSTAB algorithm is applied to solve both the velocity and the pressure field simultaneously. The system is preconditioned with the ILU(0) approach. For the temperature and the outflow problem the systems are also non-symmetrical and the BICGSTAB algorithm with ILU(0) preconditioning is used for these problems as well. The systems associated with the die deflection and the mesh deformation problem are symmetrical, positive definite and well conditioned, so the conjugate gradient (CG) solver with the cheaper Gauss Seidel preconditioning is used for these systems.

Chapter 4

Mesh generation

To arrive at the systems of algebraic equations that have been presented in the previous chapter, it is crucial that spatial discretizations of the relevant domains are available. However, the generation of these discretizations, from now on referred to as meshes, poses an enormous challenge due to the complex shapes that are associated with aluminum extrusion. As a result, existing meshing methods either fail or generate an unacceptably large number of elements for these complex domains. Therefore, new, dedicated meshing algorithms are presented that generate meshes with which the solution field can be captured accurately while the number of elements is kept to a minimum. To make these dedicated algorithms as robust and flexible as possible they are restricted to the generation of triangular surface and tetrahedral volume elements.

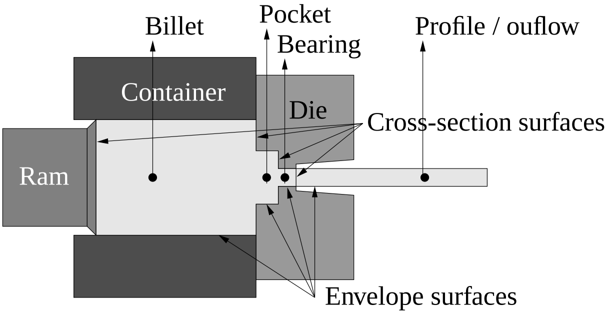


Figure 4.1: Cross-section and envelope surfaces.

For a standard flat profile two types of surfaces define the shape of the die and thus of the aluminum domain (see also Figure 4.1). Surfaces of the first type are oriented perpendicular to the extrusion direction and will be referred to as cross-section surfaces. Surfaces of the second type are tangential to the extrusion direction and will be called envelope surfaces. The cross-section surfaces of the aluminum are fully defined by the contours that define the die, i.e. those related to the container, the pocket and the bearing. The envelope surfaces are defined by these cross-sectional contours, combined with the lengths of the container, the pocket, the bearing and the profile. It should be noted that the length of the bearing can vary along its contour and that the length of the container is determined by the ram position.

Before a 3D volume mesh of the aluminum can be generated, the enclosing surfaces have to be meshed. Therefore, first the methods that have been devised to mesh the surfaces are discussed, where the generation of the cross-section meshes and the envelope meshes will be considered separately. This is followed by the method that has been developed to mesh the pocket, the bearing and the outflow volumes. Thereafter, the method that is adopted to discretize the billet volume is explained and finally the attention is focused on the meshing of the die.

As a first step volume meshes will be created with a constant bearing length. In extrusion practice, however, the bearing length varies. The mesh obtained in the first step is therefore adapted by local stretching to account for the varying bearing length. The methods used to adapt the mesh are described in detail below.

The geometry of the aluminum domain associated with a hollow profile is too complex to be fully captured using cross-section and envelope surfaces. The additional operations that have been designed to capture these complex shapes in the discretization of the aluminum will be discussed last.

4.1 Meshing the surfaces - Paving generator

Extrusion profiles are often thin walled, which implies that the flow through the cross-section is characterized by different length scales; the length scale in the direction of the profile wall is much larger than that perpendicular to the wall (see Figure 4.2). Therefore, the cross-section mesh should be directionally refined perpendicular to the profile wall. Several methods exist to generate directionally refined meshes for domains as depicted in Figure 4.2. These methods can be split into mapped and unstructured procedures. Here an unstructured mesh generator is applied because it is more suitable for complex domains.

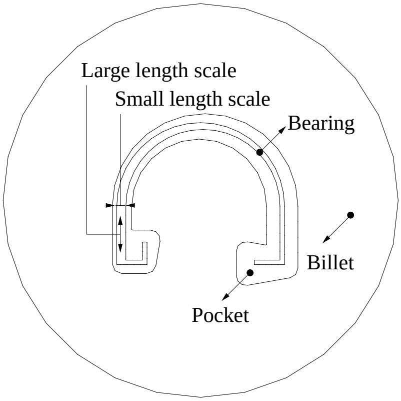
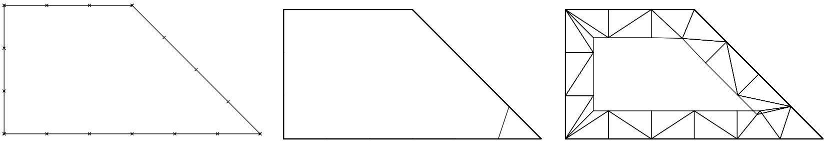
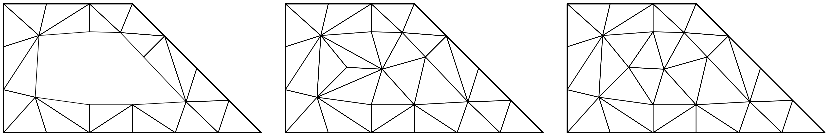


Figure 4.2: Contours for a simple profile.

Unstructured mesh generators can globally be split up into two classes, Delaunay triangulation type generators (Zheng et al., 1996) and paving or plastering generators (Blacker and Stephenson, 1991). Delaunay type generators construct grids between just the boundary nodes of the domain. Since this often generates very low-quality elements, points are added to the interior of the domain in order to meet quality criteria for the mesh. Directionally refined meshes can be generated by defining different quality criteria for different directions (Gobeau et al., 1995). However, for complex geometries the direction of the refinement varies throughout the domain, which makes the definition of the refinement direction rather cumbersome. Paving generators, on the other hand, add the elements by proceeding along the boundary between the gridded and the ungridded part of the domain, adding one element layer at a time (see Figure 4.3). An advantage of paving, as it has been indicated in (van Rens et al., 1998g), is that the thickness of each layer can be controlled to render directionally refined meshes. Therefore a paving algorithm is used.



(a) Input data *(b)* Use existing nodes *(c)* Adding a layer



(d) Merge close nodes *(e)* Domain meshed *(f)* Smoothing

Figure 4.3: The principle of paving.

The paving algorithm consists of the following steps (van Rens et al., 1998c) (see also Figure 4.3):

(a) Process the input data
    (-) Repeat
        (b) Generate triangles using existing nodes on the current boundary
        (c) Add a layer of quadrilaterals and split into triangles
        (d) Merge nodes on the new boundary that are close
    (e) Until entire domain is meshed
(f) Smooth the mesh to improve the element geometry

In the following each of these steps will be discussed briefly.

4.1.1 Input data

The input for a paver consists of the discretized boundaries of the domain and a measure for the thickness of the first layer of elements to be generated. Additionally an evolution or growth factor is used to indicate how this thickness should evolve during the paving process. Typically, the discretized boundaries are obtained (almost) directly from the Computer Aided Design (CAD) package with which the die has been designed. The thickness and the growth factor are then the only parameters that have to be supplied for each contour that defines the die.

The contours related to physical boundaries, such as the bearing opening, will be referred to as permanent boundaries (Blacker and Stephenson, 1991). The permanent boundaries remain constant during mesh generation. The boundary between the ungridded domain and the gridded domain will be referred to as the current boundary (see Figure 4.4). The current boundary continuously evolves during mesh generation and eventually vanishes when the entire domain has been meshed.

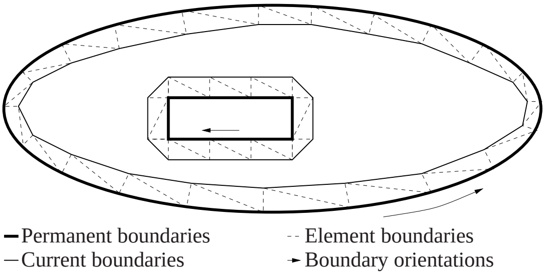


Figure 4.4: Permanent and current boundaries.

For every node i on the permanent boundary the starting thickness of the layers t_i^0 is prescribed. Also a growth factor f_i is specified for every node (see Figure 4.5). This growth factor indicates the ratio between the thickness t_i^n of a layer n and the thickness t_i^{n-1} of the previous layer $n-1$:

$$t_i^n = f_i \, t_i^{n-1} \tag{4.1}$$

A directional refinement can then be achieved by setting the thickness in each node to a fraction of the length of the boundary line elements it is connected to. If this fraction is small, elements with a high aspect ratio will result.

4.1.2 Generating triangles with the current boundary nodes

Before adding nodal points to pave a subsequent layer of elements a check is made to ensure that there is enough space between opposing parts of the current boundary to add a new layer of elements. If there is not enough space, adding a layer would imply the generation of overlapping elements. To avoid overlapping elements, bridging triangles are added at these locations. By adding these triangles the current boundary is altered such that no overlapping

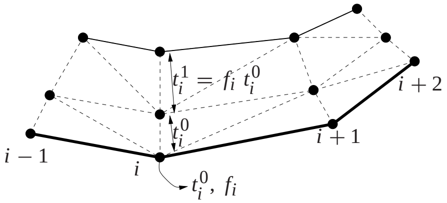


Figure 4.5: Thickness in node i for various layers.

elements will be generated in a subsequent paving step. This approach ensures, as an added benefit, that no superfluous nodes are generated.

The generation of new triangles using the current boundary nodes is attempted by checking all the line segments on this boundary. For a line segment e_i, spanned by nodes i and $i+1$, triangles are generated by selecting as a third node j any of the other nodes on the current boundary. The quality of all these triangles is evaluated. A triangle is considered to have an acceptable quality if all its corners are sharp and the distance h between the line segment e_i and the node j satisfies:

$$h < \frac{t_i^n + t_{i+1}^n}{2} + t_j^n \tag{4.2}$$

Of all the triangles that are acceptable the aspect ratio, i.e. the longest edge length of the element divided by the shortest edge length, is assessed. The triangle with the lowest aspect ratio is then generated and the current boundary is updated. This process is repeated until no more acceptable triangles can be generated.

4.1.3 Adding a layer of elements

In order to continue the meshing procedure, new nodes have to be generated. These new nodes are positioned at a distance t_i^n along the normals associated with the existing nodes on the current boundary. The normal in a node is defined to be the average normal of the line segments connected to that node. Between the existing and the new nodes quadrilaterals elements are constructed, which are subsequently divided into two triangles each (Figure 4.5).

A slightly different procedure must be followed at sharp angles in the current boundary in order to avoid distorted elements. Let α_i denote the angle between the normals $\vec{n}_{i-1}$ and $\vec{n}_i$ which belong to the line elements e_{i-1} and e_i, respectively (see Figure 4.6). For $\alpha_i > \frac{\pi}{3}$ the two new quads that share node i would become too distorted in node i. To prevent this, additional nodes are inserted (nodes j' and j'' in Figure 4.6). The number of additional nodes is given by $m = \alpha_i \div \frac{\pi}{3}$. This definition of m leads to a small number of extra nodes while maintaining an acceptable element shape. Generating elements using the additional nodes is straightforward.

4.1.4 Merging nodes on the current boundary

To keep the mesh as efficient as possible the number of nodes on the newly created boundary is reduced to a minimum by merging nodes that are almost coincident. Two nodes are merged

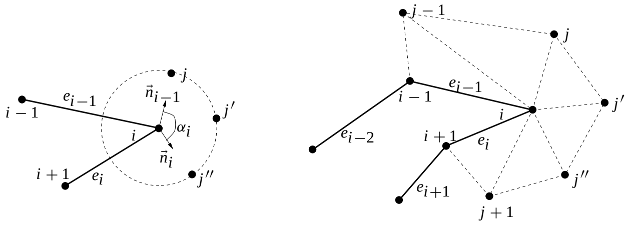


Figure 4.6: Inserting two additional nodes and triangles for $\alpha_i > \frac{2\pi}{3}$.

if the distance between the nodes is less than the average layer thickness defined in these nodes. The replacing node is positioned exactly between the two original nodes. Both the thickness and the growth factor in the new node are the average of the values in the nodes it replaces.

4.1.5 Smoothing the mesh

After the entire domain has been gridded the quality of the mesh is improved by flipping and subsequent node shifting. Flipping is used to replace elements with large angles. To obtain a balance between meshing effort and mesh quality it is only performed on elements that contain a corner with an internal angle greater than 0.6π. The process of element flipping consists of joining a triangle with that neighboring triangle that shares the longest edge with the triangle to be flipped. These two triangles then form a quad; the common edge of the triangles is one of the diagonals of the quad. Next, the quad is split along the other diagonal to render two new triangles (see also Figure 4.7).

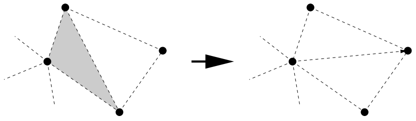


Figure 4.7: Flipping an element with a bad corner (shaded).

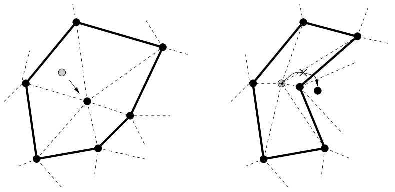
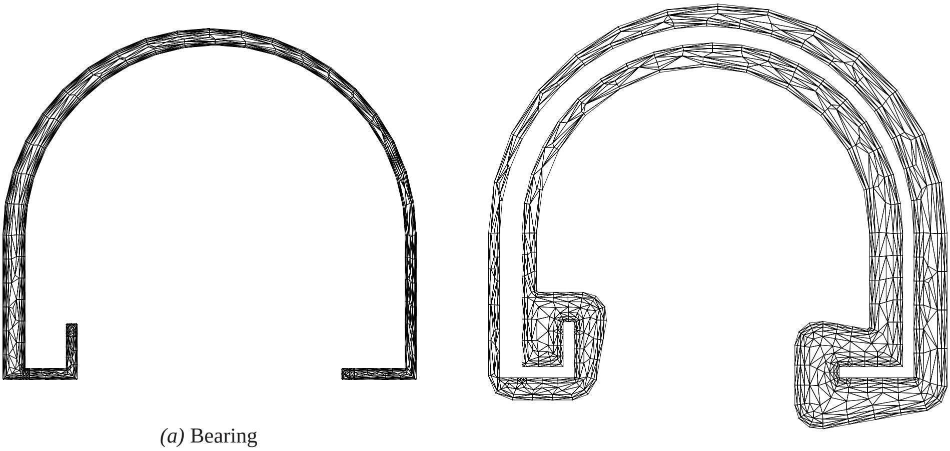


Figure 4.8: Only shifting nodes inside convex polygons.

If flipping does not improve the mesh, an attempt is made to shift the node of the triangle in which the large angle occurs. The displacement of the node should neither change the refinement direction of the mesh nor turn elements inside out. Therefore, a node is only shifted if the polygon formed by the elements connected to that node is convex. The node is then moved to the geometrical center of the polygon, as depicted in Figure 4.8.

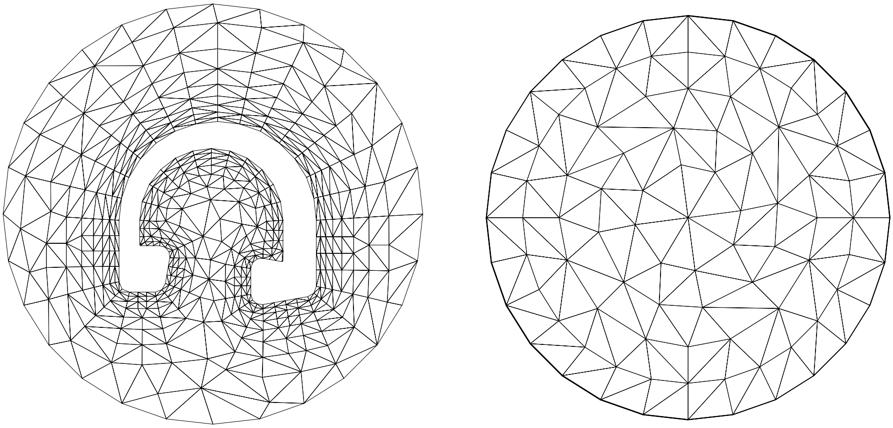
4.1.6 Resulting meshes

In Figure 4.9 the meshes resulting from the procedure introduced above have been depicted for the example configuration in Figure 4.2. The discretization of each contour is identical for every mesh it occurs in, which facilitates the meshing of the envelopes. As can be seen, high aspect ratio triangles have been generated for the bearing ro render many elements across the thickness of the profile because high gradients in the solution fields are to be expected there. The mesh of the ram surface is much coarser because near the the ram the gradients will be very low.



(a) Bearing

(b) Pocket



(c) Billet

(d) Ram

Figure 4.9: Cross-section meshes (the billet and the ram are scaled differently).

4.2 Meshing the surfaces - Expansion generator

The meshing of the envelopes is performed in two steps. First, quads are generated from the line segments in the discretized cross-section contours. Next, these quads are subdivided into triangles. This method of meshing requires that the discretizations of the two contours that bound the envelope are identical.

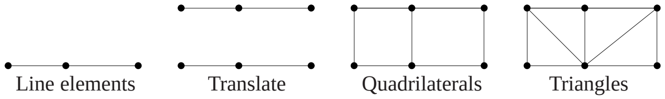


Figure 4.10: Generating triangles.

The quads are created by translating the line segments by a prescribed distance to render a new discretized contour. The line segments from the two contours are combined into quads (see Figure 4.10). This is repeated until the entire envelope surface is meshed with quads. By varying the translation distance between each contour, different levels of refinement can be obtained in extrusion direction.

To obtain a triangular surface mesh the quads have to be split into triangles. As indicated in Figure 4.10 a quad can be split into two triangles in two different ways, depending on the choice of the orientation of the diagonal in the quad. Since, as a result of the rectangular shape of the quad, both orientations will render triangles with the same quality, either orientation is acceptable. The diagonals are therefore oriented such that their direction alternates for adjacent quads. This eliminates the mesh orientation that would result from setting all diagonals in the same direction.

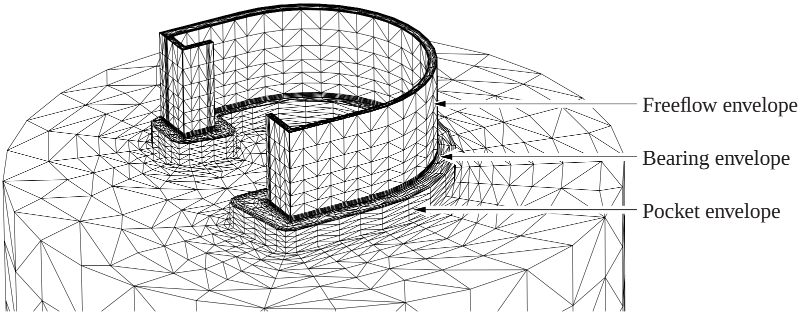


Figure 4.11: Mesh of the envelope surfaces.

In Figure 4.11 the envelope (and the cross-section) surfaces generated for the example problem are plotted. In this figure the alternating orientation of the diagonals on the enveloping surfaces can clearly be seen. It can also be observed that the mesh of the bearing envelope is more refined than the meshes of the freeflow and the pocket envelopes. This is done to appropriately capture the high gradients in the solution field in the bearing.

4.3 Meshing the volumes - Expansion generator

Of all the volumes that have to be filled with tetrahedral elements the volumes of the pocket, the bearing and the outflow are descretized using the expansion mesh generator discussed in this section. The volumes of the billet and the die are meshed using the generator discussed in the next section. The expansion generator applied in this work generates tetrahedra (tets) in two steps which will be treated in more detail in the following. First, prisms are created from the triangles in the cross-section meshes. Next, the prisms are subdivided into tetrahedra. The volumes that can be meshed with this expansion generator are restricted to volumes of which the top and the bottom surface have topologically equivalent meshes and of which the envelope surface has been created using the expansion surface generator dealt with in Section 4.2.

4.3.1 Generating prisms

The creation of the prisms is performed in a process similar to the creation of the quads in the envelope surfaces. Each triangle of the cross-section mesh is translated in the extrusion direction to render a new cross-section mesh (see also Figure 4.12). Subsequently, the triangles of both cross-section meshes are combined into prisms. The process of translating and combining is repeated until the entire volume is filled. To ensure that the nodes on the contour of each cross-section coincide with the nodes of the envelope mesh that encloses the volume, the distances over which the cross-section meshes are translated, are equal to the translation distances that have been used in the generation of the envelope mesh.

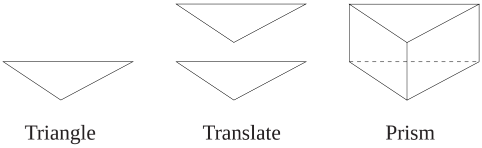


Figure 4.12: Generating prisms.

4.3.2 Generating tetrahedra

After the prisms have been generated they are split into tetrahedra. To do so, each rectangular face of a prism has to be split into two triangles (see also Figure 4.13). The split has to be performed such, that the diagonals introduced on the rectangular faces of two adjacent prisms have matching directions. This will be referred to as the matching-criterion.

With the diagonals set, each prism can be subdivided into tets using one of two paradigms. The first consists of introducing a node in the baricenter of the prism. This node then serves as the common top node which, when combined with each of the 8 triangles on the sides of the prism, generates 8 tets. The second method entails the splitting of the prisms into 3 tets, without introducing an internal node in each prism. The second method is applied in this work because it reduces the number of nodes and elements. Furthermore, it can easily be verified that the worst aspect ratio of the elements that are created by applying the 3-tet split

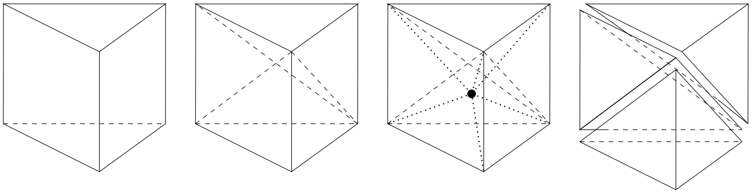

Prism Diagonals 8 tetrahedra 3 tetrahedra

Figure 4.13: Splitting a prism into 8 or 3 tetrahedra.

is always better than the worst aspect ratio of the elements that are obtained with the 8-tet split.

A disadvantage of the 3-tet split is that it imposes an extra requirement on the orientation of the diagonals of each prism. The orientation of the diagonals should not only match for adjacent prisms, but within one prism the diagonals should be oriented such that the 3-tet split is possible (Löhner, 1993). The 3-tet criterion dictates that both orientations of the diagonals should occur on the rectangular faces of an individual prism (see also Figure 4.14).

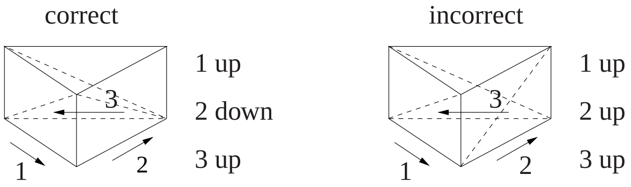


Figure 4.14: Diagonal orientation with respect to the 3-tet criterion.

An iterative procedure is used to select the orientation of the diagonals in a pattern that meets the additional 3-tet requirement (van Rens et al., 1998c). With the predefined diagonal orientations of the faces in the envelope mesh as a starting point a loop over the prisms is performed. For those prisms in which the orientation of the diagonal has been set for one or two rectangular faces, the orientation is set for one additional face. The orientation is chosen such that the diagonals of both this prism and of the adjacent prism obey the matching criterion and do not violate the 3-tet criterion. However, it is not possible to satisfy the matching criterion and the 3-tet criterion for both prisms in case both prisms already have two diagonals set and the 3-tet criterion in each prism requires non-matching diagonal orientations on their mutual face. In this case the orientation of the diagonal is set such that the current prism meets the 3-tet criterion, which implies that the adjacent prism obtains a diagonal configuration that does not satisfy the 3-tet criterion. The prisms with incorrect diagonal configurations are dealt with next. This loop is repeated until the diagonals of all the faces are set.

Due to the possible incompatibility between the matching and the 3-tet criterion described above, it is likely that the mesh contains prisms for which the 3-tet criterion is not satisfied. The orientation of the diagonals in these prisms is corrected as follows. For each prism that violates the 3-tet criterion the orientation of one of the diagonals is changed while the diagonal of the adjacent prism is altered accordingly to ensure that the matching criterion is not violated. Of the three rectangular faces in the incorrect prism on which the diagonal can be swapped that face is selected for which the adjacent prism still conforms to the 3-tet criterion after the swapping. If none of the adjacent prisms allow the swapping of a diagonal, the diagonal direction of one face is altered nevertheless, thus causing the 3-tet criterion to be violated in the prism adjacent to this face. Consequently, it is possible that after the alteration of the diagonal directions other prisms violate the 3-tet criterion. These prisms are identified by performing a new scan for incorrect prisms, which are adjusted in turn. This is repeated until all prisms satisfy the 3-tet criterion (typically 1 to 2 iterations are required). To avoid repetitions, the orientation of each diagonal can be altered only once in this process.

4.4 Meshing the volumes - Unstructured generator

The tetrahedral mesh of the billet (or the die package) cannot be generated using the expansion generator discussed in Section 4.3 because this generator requires that the volume has topologically equivalent cross-section meshes on both ends of an envelope mesh. The mesh of the ram surface is much coarser than that of the combined bearing, pocket and billet surface (see Figure 4.9). Therefore, an unstructured mesh generator is used to generate the tetrahedral mesh of the billet (the die package is considered separately in Section 4.4.7).

It is beyond the scope of this thesis to give a detailed description of the unstructured mesh generator and therefore only the basic ideas are presented here. The unstructured mesh generator is based on ideas similar to the paving approach introduced in Section 4.1. It consists of the following steps:

```
(a)  Process the input data
     (-)  Repeat
          (b)  Generate tetrahedra using existing nodes on the current boundary
          (c)  Try to add one new node to generate one new tetrahedron
               -    on success goto (b), else goto (d)
          (d)  Apply Delaunay method to resolve complex areas
          (f)  Smooth the mesh
          (g)  Delete elements with unacceptable shape or size
     (h)  Until entire domain is meshed
```

The concepts behind each of the steps will be discussed briefly.

4.4.1 Input data

The input consists of the triangulation data for the volume boundary. As in the paving algorithm a distinction is made between the permanent, physical boundary and the current, constantly evolving boundary. Before elements have been generated the permanent and the current boundary coincide. While elements are generated the current boundary changes and eventually vanishes.

The triangulation of the permanent boundary is used to compute the edge length distribution of the boundary mesh. The edge lengths on the boundary are interpolated inside the volume to obtain a field of desired edge lengths for the entire domain. This field of edge lengths is used to determine the coarseness of the mesh everywhere in the domain.

4.4.2 Generate elements with the current boundary nodes

Before adding new nodes a check is made whether elements can be generated using the existing nodes on the current boundary. This is done by considering all triangles on the current boundary. For each triangle, tets are generated by combining the triangle with all the other nodes on the current boundary. For each tet, the lengths of the edges that connect the node to the triangle are computed. Also, the angles between these edges and the triangle surface are determined. The edge lengths and angles are compared for every tet associated with one triangle. If at least one tet has edge lengths and angles that are acceptable the tet that is most resembling to an equilateral tet is generated. To determine whether the edge lengths of a new tet are acceptable, they are compared to the local desired edge length that is stored in the field computed in step (a), Section 4.4.1. The angles of a tet are acceptable if they are within preset limits. After a tet is generated the current boundary is updated. Then the loop over the current boundary is repeated until no additional tets can be generated.

4.4.3 Adding a node to generate a tetrahedron

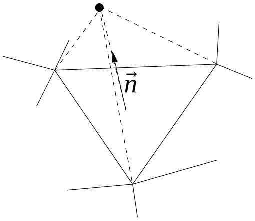


Figure 4.15: Generating a tet by adding a node.

If no more tets can be created from the current boundary mesh a new node has to be generated to continue the meshing procedure. The new node is created using a triangle in the current boundary as basis (see also Figure 4.15) and is positioned along the normal of the triangle that starts in the geometrical center of the triangle. The position is selected such that the edges of the new tet between the triangle and the new node have the desired length. If the new node is positioned inside the unmeshed part of the domain, the node and the tet are maintained and the current boundary is updated. Then step (b) is repeated to generate elements with the updated current boundary. If the new node is positioned inside an existing tet, the node and the tet are discarded and the next triangle is used to create a new tet by adding a node. This is repeated until a node and tet have been generated or all triangles have been considered. If all the triangles have been tried and no new nodes could be generated the current boundary will be very complex. In this case a Delaunay mesh generator will be employed for the remaining unmeshed domain.

4.4.4 Applying the Delaunay algorithm

The Delaunay algorithm is used to deal with complex current boundaries. For a detailed description the Delaunay method the interested reader is referred to (Zheng et al., 1996) or (Joe, 1991) and references therein. Here it is only mentioned that the Delaunay method is capable of generating a tetrahedral mesh for almost any boundary triangulation without introducing additional nodes. One of the few cases for which it fails, is a boundary that is topologically equivalent to a prism with incorrect boundary orientations as depicted in Figure 4.14. Then an additional node has to be added to generate 8 tets. The major drawback of Delaunay's method is that it often generates elements with very poor aspect ratios. Therefore it is only used if the element generation algorithms in steps (b) and (c) have failed.

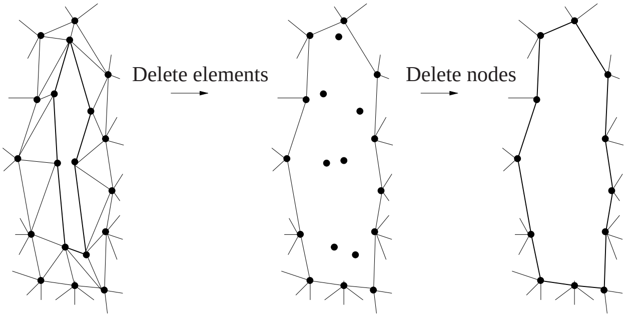


Figure 4.16: Removing the current boundary.

The current boundary is not used for the Delaunay algorithm because the region it encloses can be very slender, which may result in extremely deformed tets. Therefore the current boundary is redefined by eliminating all the elements connected to nodes on the current boundary (see also Figure 4.16). After the element elimination the nodes that were on the current boundary are no longer connected to any elements and are therefore deleted as well. Next, the Delaunay algorithm is applied to the redefined boundary.

4.4.5 Smoothing the mesh

The mesh is smoothed to increase the quality of the tets. This smoothing is performed using a standard Lagrangian smoothing algorithm. This algorithm consists of a loop over all nodes in the mesh. For each node the polyhedron that is spanned by the elements connected to the node is constructed. Then the node is translated to the geometrical center of that polyhedron. It should be noted that it is possible that elements become overlapping during this rather crude smoothing method. If this happens the inside-out elements are dealt with in the next step.

4.4.6 Deleting unacceptable elements

At the end of one meshing iteration two types of unacceptable elements can occur in the mesh. The first type of unacceptable elements are those elements that overlap other elements as a result of the node repositioning during the smoothing of the mesh. Such elements are never acceptable and are all marked for deletion. The second type are those elements which have edge lengths that deviate too much from the locally desired edge length. Of all the elements that have unsuitable edge lengths only the worst element is marked for deletion.

The marked elements are deleted along with all the elements that share a node with the marked elements. Since the nodes of the marked elements are not connected to any elements anymore after the element deletion, these are deleted as well. The deletion of marked elements causes the reformation of a current boundary which necessitates another meshing iteration. The mesh generation is terminated if no more elements have been marked for deletion.

4.4.7 Resulting mesh of the die package

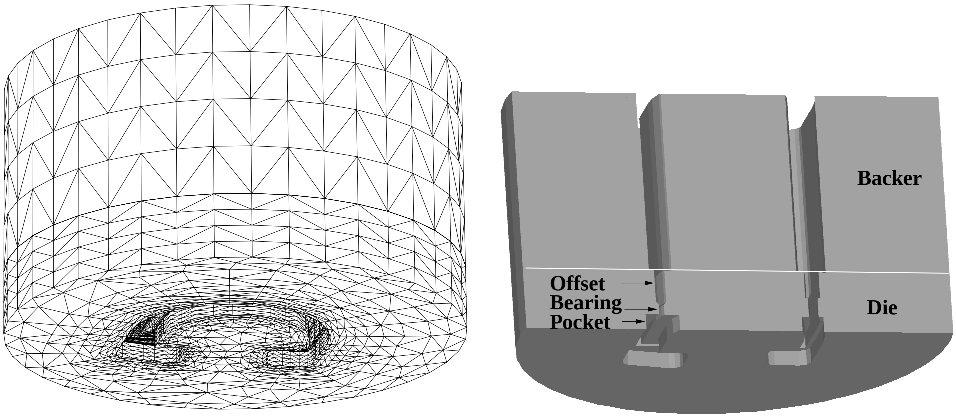


Figure 4.17: The volume mesh of the die package.

The mesh of the die package depicted in Figure 4.17 has been generated using the unstructured mesh generator. The resulting mesh is plotted on the left whereas an intersection of the die package is plotted on the right. In the intersection plane of the die the pocket, the bearing and the outflow offset can be identified. The outflow offset ensures that the aluminum of an unbalanced profile does not touch the die when the profile exits the die under an angle.

The domain of the die and the backer are meshed separately. At the aluminum-die interface the surface mesh of the die domain is taken in conformity with the mesh of the aluminum surface to simplify the modeling of the interactions between these domains. Also, the surface mesh of the backer at the die-backer interface is identical to the surface mesh of the die to simplify the discretization of the Lagrange multiplier that is used to model the frictionless contact between the die and the backer. Since the gradients in the displacement field will be larger in the die domain than in the backer domain the element size in the die domain is smaller than in the backer domain.

4.5 Stretching the bearing

In the example mesh generated in the previous sections the bearing has a constant length in extrusion direction. However, in reality the bearing length varies along the bearing contour and numerical experiments indicate that the length of the bearing has a significant influence on the outcome of the simulations (van Rens et al., 1998d). Therefore, the constant bearing length mesh is adapted in a number of steps to incorporate the varying bearing length (van Rens et al., 1998e). These steps are visualized in Figure 4.18, but only for the bearing volume mesh. Of course, these operations are applied to the outflow volume mesh and the die mesh as well to ensure the meshes remain compatible. In each of the steps nodes are translated in the extrusion direction while the topology of the mesh remains the same. This implies that different bearing lengths can be studied by repeating only the stretching operation, i.e. without having to generate a new mesh.

The mesh of the aluminum bearing volume with a constant bearing length, as depicted in Figure 4.18(a), serves as a point of departure. The length of the bearing is only specified in a discrete number of control nodes along the bearing contour and varies linearly between these nodes. In the first step the control nodes are translated to the specified position (Figure 4.18(b)).

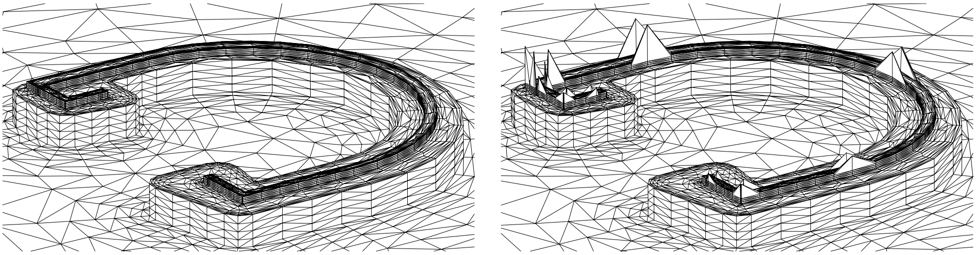
In the second step all the other nodes on the contour are translated to the real bearing geometry. The translation distance for these nodes is computed by a linear interpolation between the control nodes (Figure 4.18(c)). The linear interpolation is performed by solving a diffusion problem on the line segments that constitute the contour of the bearing. In this diffusion problem the translation distances are computed by prescribing the bearing lengths in the control points as Dirichlet boundary conditions.

It can be seen in Figure 4.18(c) that the cross-section surface that separates the aluminum bearing volume from the outflow volume has become very distorted. Therefore, in the third step, this surface is smoothed by repositioning the nodes that belong to this surface (Figure 4.18(d)). Many methods exist to compute the distance over which the surface nodes have to be translated (see e.g. (Tezduyar et al., 1992) and (Johnson and Tezduyar, 1994)). Here, it is obtained in the same manner as in the previous step, by solving a diffusion problem on the surface. In this problem the translation distances for the surface are computed, using Dirichlet boundary conditions to prescribe the previously computed translation distances on the bearing contour.

Lastly, in the fourth step, the volume mesh of the bearing is smoothed by translating the nodes in this volume (Figure 4.18(e)). Again, the translation distances for the nodes in the volumes are obtained by solving a diffusion problem for the bearing volume. In this diffusion problem the previously computed translation distances of the cross-section surface are imposed as Dirichlet boundary conditions. When both the bearing and the outflow meshes are plotted after the stretching, Figure 4.18(f) is obtained.

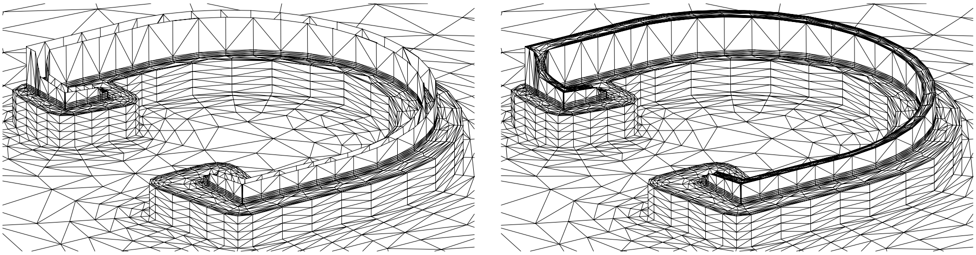
4.6 Extension to hollow dies

For hollow profiles the shape of the aluminum domain is considerably more complicated than for flat profiles. This is caused by the fact that the die is constructed out of two parts, instead



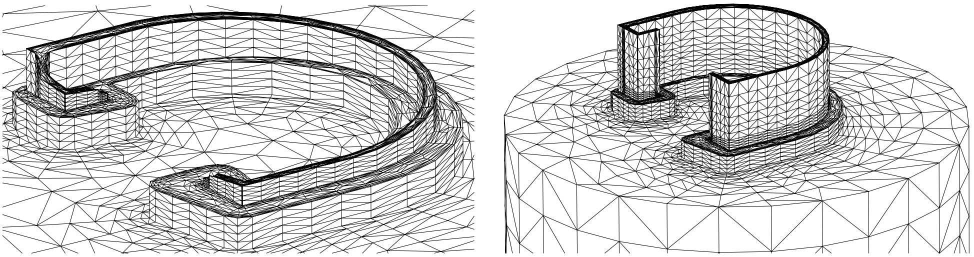
(a) Constant bearing length

(b) Moving the control points



(c) Linear interpolation of contours

(d) Smoothing the surface



(e) Smoothing the bearing

(f) Final mesh with outflow

Figure 4.18: Stretching the bearing.

of one (see Figure 4.19). The first part is the die plate which determines the external contour of the profile. In the die plate a welding chamber is added to allow the aluminum flow, that has split to pass the legs, to weld together again before entering the pocket of the die. The second part, the bridge part, consists of a core that determines the internal contour of the profile and the legs which support the core.

The core and the die plate consist of surfaces that are either perpendicular or tangential to the extrusion direction. This implies that they can be meshed with the methods described

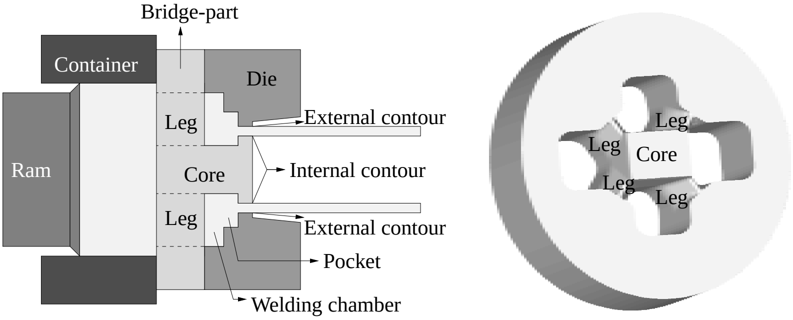
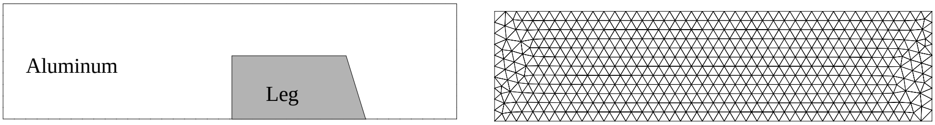


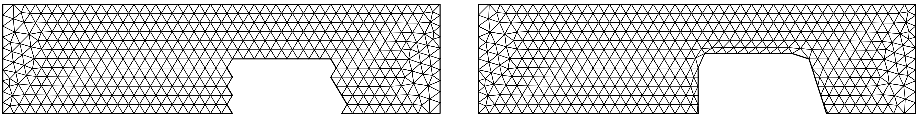
Figure 4.19: Extrusion of hollow profiles (left) and the bridge part of a die (right).

earlier in this chapter. However, as can be seen in Figure 4.19, the complex geometry of the legs cannot be described with just perpendicular or tangential surfaces. Therefore, the meshing of the aluminum domain associated with hollow profiles is performed in several steps (see also Figure 4.20):



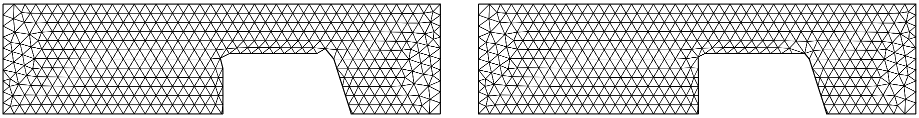
(a) Geometry

(b) Initial mesh



(c) Internal elements removed

(d) Translate onto boundary, smooth



(e) Intersecting elements removed

(f) Translate onto boundary, final mesh

Figure 4.20: Removing elements in four steps.

(a) The aluminum geometry is constructed taking only the geometry of the core into account. The geometry of the legs is described separately.

(b) The aluminum geometry is meshed, incorporating the geometry of the core but neglecting the geometry of the legs, using the methods discussed in the previous sections.

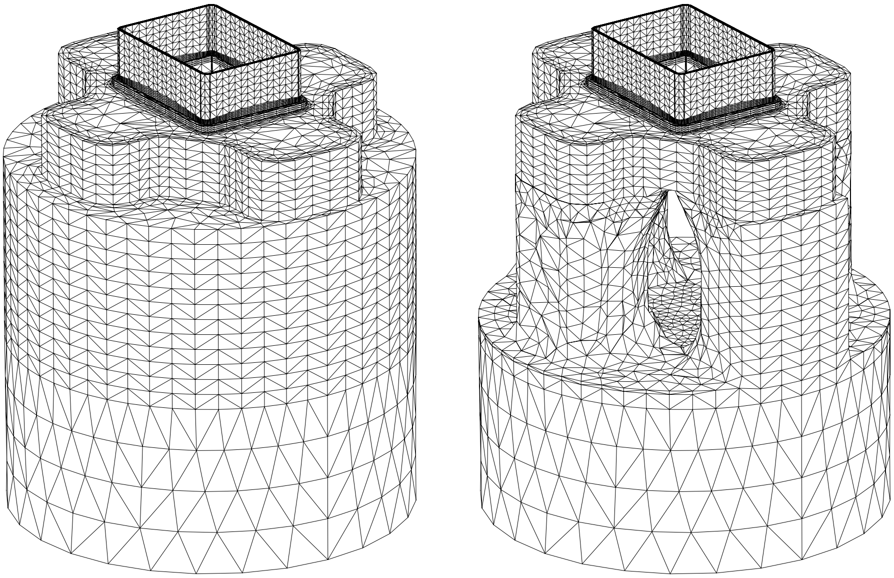
(c) The elements of which all nodes are positioned inside the geometry of the legs are removed.

(d) After the removal of the elements a set of external faces has been generated. These newly formed external faces are connected to nodes that are inside the legs. These nodes are translated onto the surface of the legs. During the translation the new positions of the nodes are determined such that they do not cause elements to become overlapping. The translation is followed by a local smoothing of the mesh around the translated nodes. To realize this the elements that are connected to newly formed external faces are marked. The nodes of these elements that are not connected to external faces are repositioned to increase the quality of the elements.

(e) It is possible that near sharp edges some elements intersect the legs, even though their nodes are on or outside the boundary of the legs. Therefore, the elements that have their geometrical center inside the geometry of the legs are deleted. The number of elements deleted in this step is generally very small.

(f) The translation of step (d) is repeated. In this case the newly formed external faces are connected to nodes that are positioned on or outside the geometry of the legs and these nodes have to be translated onto the surface of the legs as well.

In Figure 4.21 an example is given of an initial and final mesh belonging to the aluminum domain of a rectangular hollow profile.



(a) Initial mesh

(b) Final mesh

Figure 4.21: Generating the aluminum domain mesh for a rectangular hollow profile.

Chapter 5

Evaluation of the finite element model

In this chapter the finite element model developed in the previous chapters will be evaluated. First, it is investigated whether the separate modeling problems that are coupled in the staggered scheme have a significant influence on the results of the finite element computations. To this end, a number of extrusion simulations is performed for one die design, while in each of these simulations the influence of one of the coupled phenomena is neglected, or altered. The results of these simulations are compared to results that are obtained for a reference configuration that incorporates all the coupled problems, see Section 5.1. Second, the influence of the mesh coarseness on the accuracy of the solution field will be examined. This is done by comparing the solution fields that are obtained on a very fine mesh to the solution fields that have been computed on meshes with coarsenesses that are typically used in this thesis. The results of this comparison are presented in Section 5.2. Third, in Section 5.3, the predictions of finite element simulations are compared to data that have been measured in extrusion practice.

5.1 Relevance of the separate problems

In this section results will be presented which indicate that the solution of each of the separate problems that constitute the staggered scheme depicted in Figure 3.8 has a considerable influence on the flow field of the aluminum. Also, it will be shown that the selected value for the friction coefficient plays a crucial role in the outcome of the simulations. The data that justifies these conclusions is obtained by computing the solution fields for a number of test problems. In each test problem either one of the separate phenomena is neglected, or the friction coefficient is altered (van Rens et al., 1999a). The flow fields from these test problems are compared to a reference flow field that is computed using an analysis in which each of the separate problems is accounted for.

For the actual comparison the profile visualized in Figure 4.2 is used. The mesh of the aluminum domain is shown in Figure 4.11, and the mesh of the die-package is depicted in

Figure 4.17. The radius of the billet is in the order of magnitude of 130 [mm], the thicknesses of the thick and the thin legs are 6 [mm] and 3 [mm], respectively, and the area reduction from billet to profile is close to a factor 40. For each problem the prescribed ram velocity equals $6.3 \cdot 10^{-3}$ [m s^{-1}] and the imposed ram temperature is 733 [K]. As mentioned in Section 2.4.1, the reference Coulomb friction coefficient μ is taken to be 0.4 [-].

First the attention will be focused on the solution fields that are obtained for the reference problem. The streamlines of the aluminum are plotted in Figure 5.1. The notable corners in the streamlines near the pocket and the die are indicative for the relatively high velocity gradients in those areas.

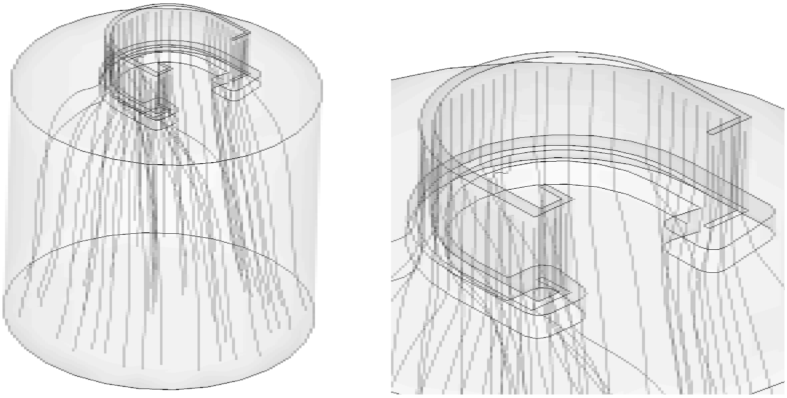


Figure 5.1: Streamlines of the aluminum.

The desired and the computed shapes of the profile are plotted in Figure 5.2 for the reference problem, where the geometrical deviations of the computed profile shape from the nominal shape have been multiplied by a factor 10 for visualization purposes. It can be concluded that the die is not optimally balanced, because the profile tends to flow to the right.

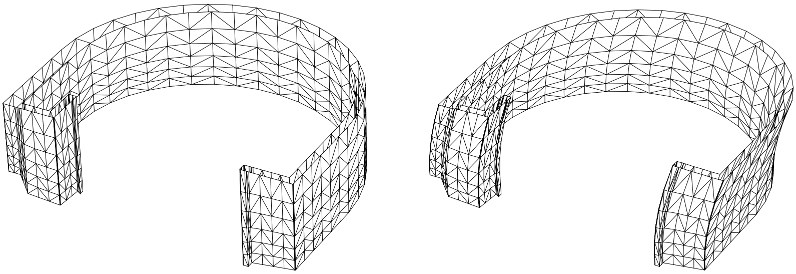


Figure 5.2: Undeformed and deformed outflow surface (deformations scaled with factor 10).

Due to the u-shape of the profile, the die has a large lip in the middle which is connected to the rest of the die through a relatively thin bridge, between the inward pointing legs of the die. Such lips are notoriously weak sections of a die in which considerable deformations occur as a result of the high extrusion pressure. The shape of the deformed die is plotted for a number of cross-sections in Figure 5.3, where the deformations of the die have been magnified with a factor 50. It can be observed that the lip bends out as a result of the interaction with the aluminum. Also, at the interface between the die and the backer it can be seen that the die and the backer can move freely with respect to each other. This is in agreement with the frictionless contact condition that has been used.

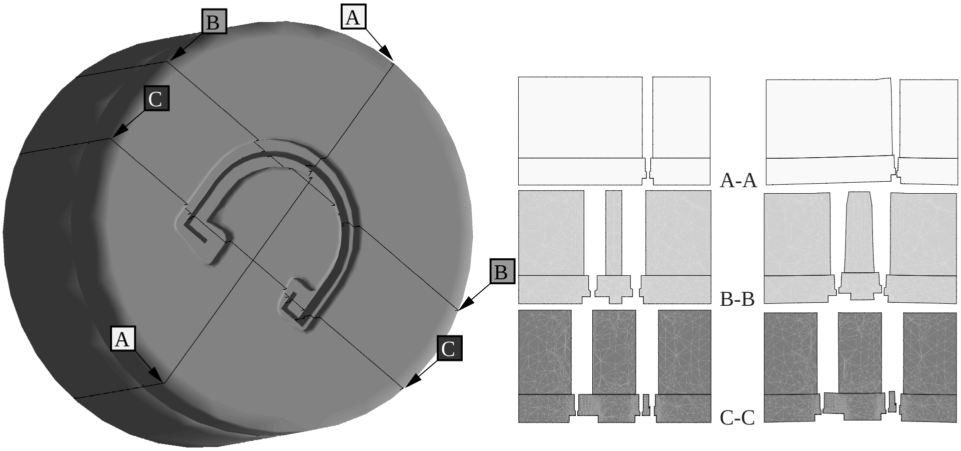


Figure 5.3: Left: Die package with the cross-section planes A-A, B-B and C-C indicated. Right: The undeformed and deformed shapes of the cross-sections (deformations scaled with factor 50).

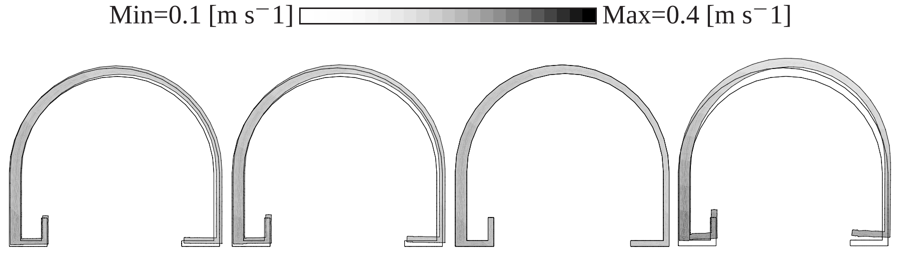
With the reference problem as a point of departure, the test problems in Table 5.1 are defined. Three test problems are formulated that neglect the influence of either the deformation of the die, the deformation of the outflow surface, or the temperature dependence of the viscosity, on the flow field of the aluminum. Additionally, four test problems are constructed in which the friction coefficient, μ is altered to respectively, 0.0, 0.2, 0.75, and ∞. It should be noted that the first and the last choice of μ represent a free slip and a full stick boundary condition for the aluminum in the die, respectively.

In Figure 5.4 the velocities in extrusion direction are presented for the reference problem and the test problems. The velocity contours are plotted in a cross-section of the computed (and consequently deformed) profile shape. In these plots the outline of the desired profile shape is also depicted as a reference. The deformations of the outflow surface have been magnified with a factor 5 to make the shapes more distinguishable. This magnification causes the boundary of the profiles to become somewhat jagged even though they are relatively smooth as can be seen in Figure 5.5, in which the temperature distribution at the outflow has been plotted in the computed cross-sections without additional scaling.

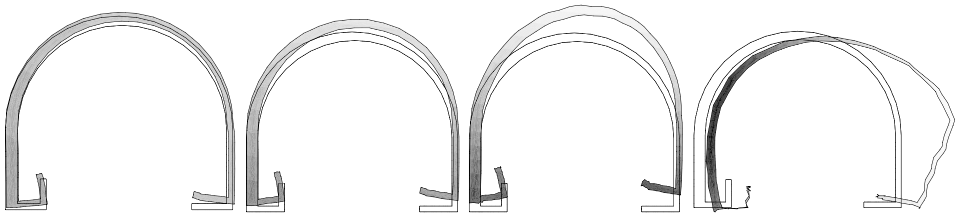
By comparing Figure 5.4(a) to Figure 5.4(b)-(d) it can be established that each of the coupled problems has an influence on the velocity field and, consequently, on the computed

Problem	Die deformation	Outflow deformation	Temperature dependent viscosity	Friction coefficient
Reference	yes	yes	yes	$\mu = 0.4$
Test 1	no	yes	yes	$\mu = 0.4$
Test 2	yes	no	yes	$\mu = 0.4$
Test 3	yes	yes	no	$\mu = 0.4$
Test 4	yes	yes	yes	$\mu = 0.0$
Test 5	yes	yes	yes	$\mu = 0.2$
Test 6	yes	yes	yes	$\mu = 0.75$
Test 7	yes	yes	yes	$\mu = \infty$

Table 5.1: Definition of the reference and test problems.



(a) Ref, $\mu = 0.4$ *(b)* T1, rigid die *(c)* T2, rigid outflow *(d)* T3, const. temperature



(e) T4, $\mu = 0$ *(f)* T5, $\mu = 0.2$ *(g)* T6, $\mu = 0.75$ *(h)* T7, $\mu = \infty$

Figure 5.4: Velocities in extrusion direction. The difference between the computed and the nominal shape is magnified with a scaling factor 5.

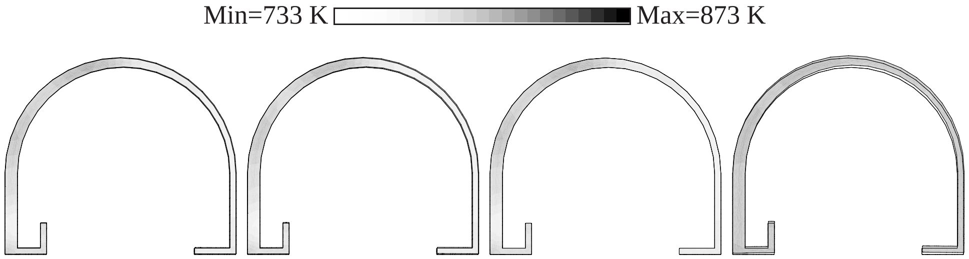
shape of the aluminum. Here, the flow field differs the least from the reference flow field for problem T1, where the deformation of the die is neglected, and the flow field deviates the most for problem T3, where the effect of the temperature on the viscosity is left out of consideration. The velocity field for test problem T2, where the changing shape of the outflow is neglected, deviates only slightly from the reference problem, because the computed outflow shape of the reference problem is almost identical to the desired outflow shape. It can, therefore, be expected that these differences become more pronounced if the computed profile shape differs more from the desired profile shape. Furthermore, problem T2 lacks the information about the final shape of the profile, which makes the simulation results more difficult to interpret. It can be concluded that the temperature problem and the free outflow problem should always be incorporated. The deformation of the die only needs to be considered if the die is expected to have weak sections such as the lip in the die for this example.

The large influence of the friction modeling on the flow field of the aluminum becomes apparent when Figure 5.4(a) is compared to Figure 5.4(e)-(h). For the full stick assumption in problem T7 the outflow velocities are depicted in Figure 5.4(h); the full stick assumption is physically unrealistic and it can be seen that it also renders an unrealistic flow field. For the friction coefficients $\mu = 0.0$, $\mu = 0.2$ and $\mu = 0.75$ the balance of the flow field is worse than in the reference problem. This can be concluded from the distorted cross-sections that are associated with the problems T4, T5 and T6 in Figure 5.4(e)-(g). For the coefficients $\mu = 0.0$ and $\mu = 0.2$, the exit velocity of the thick leg is slightly higher than that of the thin leg. This can be attributed to the low value of the friction coefficient which reduces the balancing effect of the longer bearings in the thick leg. For a friction coefficient $\mu = 0.75$ the opposite can be observed since the higher friction coefficient increases the balancing effect of the longer bearings and restricts the flow in the thick leg.

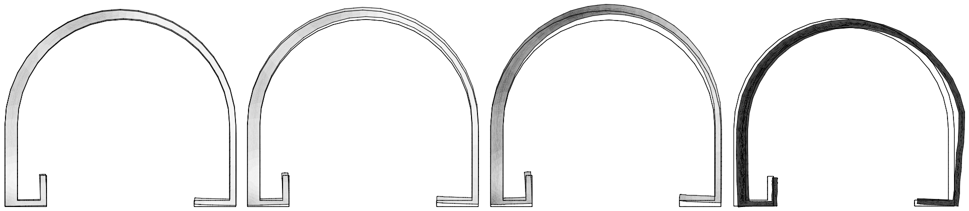
In Figure 5.5 the temperatures in a cross-section of the profiles are plotted. The temperature contours are depicted in the computed profile shape without additional scaling while the nominal profile shape is outlined as a reference. The differences in the temperature fields between the problems T1-T4 are relatively small. Even when the temperature is computed as in T3, where the influence of the temperature on the viscosity is neglected, the temperature is only marginally overestimated. For the varying friction coefficients on the other hand, the temperature does increase considerably as the friction coefficient μ is increased above 0.4, where it should be noted that the temperatures in T7 exceed the maximum of the temperature scale by as much as 50 [K]. Enlarging the friction coefficients leads to temperature increases because the shear rates inside the bearing become higher. The temperatures do not decrease if the friction coefficient is lowered below 0.4. This indicates that in these cases the temperature of the profile is mainly determined by the deformations that the aluminum experiences before it enters the bearing.

5.2 Mesh sensitivity

To assess the sensitivity of the finite element analyses to the size and orientation of the mesh, two tests are performed on a cylindrical hollow profile (see also Figure 5.6). First, a comparison is made between the solution fields that are computed on meshes with different element sizes. Second, the effect of the diagonal orientation in the envelope mesh on the computed velocity fields is assessed. In both cases the solution fields are compared to the ’exact’ solution for this profile. This exact solution is obtained on an extremely fine referenced mesh.



(a) Ref, $\mu = 0.4$ *(b)* T1, rigid die *(c)* T2, rigid outflow *(d)* T3, const. temperature



(e) T4, $\mu = 0$ *(f)* T5, $\mu = 0.2$ *(g)* T6, $\mu = 0.75$ *(h)* T7, $\mu = \infty$

Figure 5.5: Temperatures in a cross-section of the profile. Both the computed (solid) and the nominal (outline) shape are depicted.

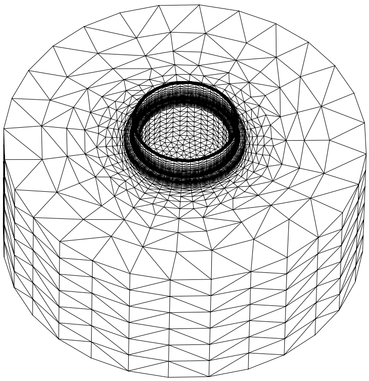


Figure 5.6: One of the meshes of the hollow profile.

5.2.1 Reference mesh

To compute the ’exact’ solution fields for the extrusion of the hollow cylindrical profile depicted in Figure 5.6, an extremely fine mesh is required. Since it would be computationally infeasible to perform the analysis with a very fine 3D mesh, the rotational symmetry of the profile is exploited to compute the velocity fields for the exact solution on a 2D mesh. The 2D mesh was generated using the paving algorithm introduced in Section 4.1 and the element size was determined such that the flow field and the temperature field did not change upon further refinement of the mesh. This mesh contains about 30 elements across the thickness of the profile and is shown in Figure 5.7.

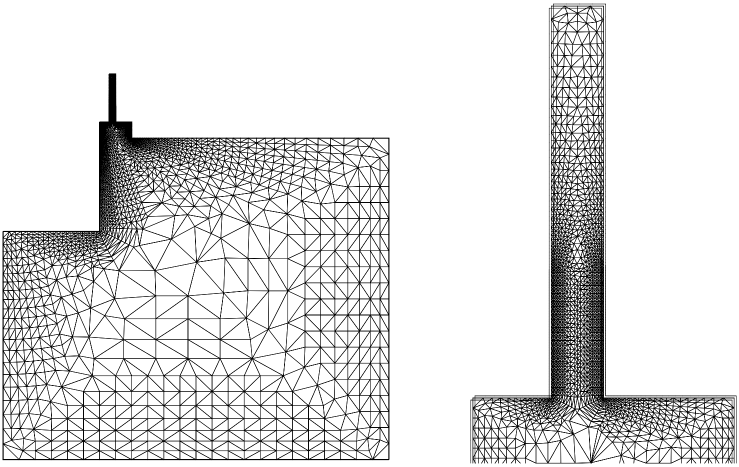
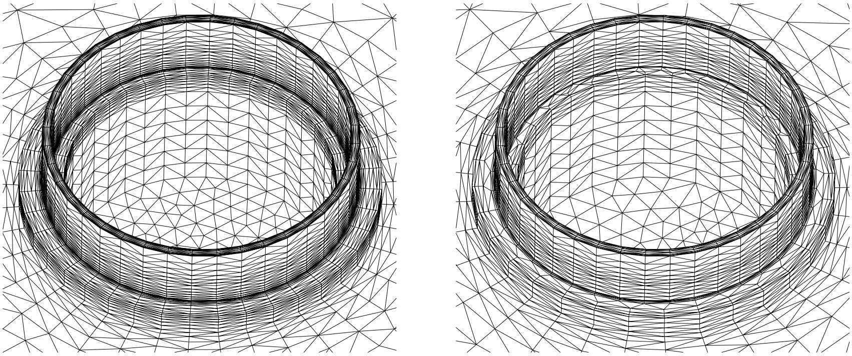


Figure 5.7: Left: 2D mesh, used to compute the 'exact' solution. Right: Detail of the die outflow area.

5.2.2 Mesh coarseness

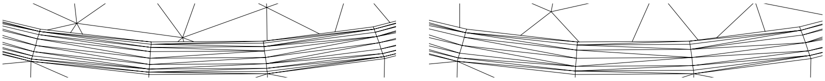
The domains associated with aluminum extrusion contain a sharp edge at the entrance of the pocket and at the entrance of the bearing. Especially the sharp edge near the entrance of the bearing has to be captured accurately since the gradients near that edge are rather high. To correctly compute the flow field near this edge, the special discretization of the friction normal has been introduced in Section 3.1.4 of this thesis. Additionally, the dedicated mesh generators developed in this work are used to create small elements near that edge. In the present section it is investigated whether these measures are sufficient to describe the global solution fields properly or a further refinement is necessary.

Two meshes are generated for the aluminum domain depicted in Figure 5.6. The first mesh is much coarser than the 2D mesh (see Figure 5.8(a)). However, when compared to the meshes that have typically been used in this work for the aluminum domain, it is still rather refined near the entrance of the bearing. To avoid unrealistically large numbers of degrees of freedom the discretization is coarsened away from the bearing entrance. The cross-section surface mesh of the bearing volume contains approximately 8 elements across the thickness of the profile. The distribution of these elements is strongly biased towards the boundaries of the cross-section, such that more slender elements with higher aspect ratios are located at the boundaries of the mesh than in the middle. This bias is achieved by using a small starting thickness combined with a large growth factor in the paving algorithm described in



(a) Fine mesh *(b)* Coarse mesh

Figure 5.8: The fine and the coarse mesh.



(a) Fine mesh *(b)* Coarse mesh

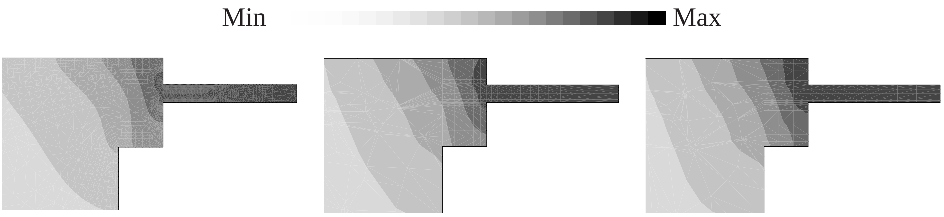
Figure 5.9: The bearing cross-section mesh for the fine and the coarse discretization.

Section 4.1 (see also Figure 5.9(a)). Moreover, the envelope mesh for the bearing contains about 7 elements in extrusion direction. In the other meshes presented in this work between 6 to 8 elements are generated across the thickness of the profile and the bias towards the boundaries is less pronounced. Also, usually only 4 to 5 elements are used for the bearing envelope mesh.

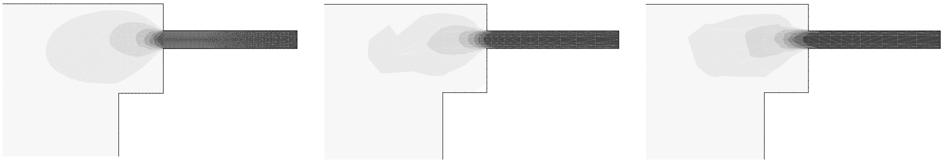
The second mesh is rather coarse near the die entrance compared to the meshes that have typically been used in this work. It contains only 6 elements across the thickness of the profile that are moderately biased towards the boundary of the cross-section (see also Figure 5.9(b)). Also, only 4 elements in extrusion direction are used in the envelope mesh of the bearing.

In Figure 5.10 the velocity field and the temperature field are depicted for the three different discretizations. The same contour levels have been used for each set of contour plots. The temperature fields that are computed on both the fine and the coarse mesh render results that are very similar to the exact solution. This can be observed in Figure 5.10(a). For a coarser mesh the exit temperature is slightly overestimated. The error in the maximum exit temperature of the aluminum is less than 1% of the temperature range for the fine mesh whereas it is about 3% for the coarse mesh.

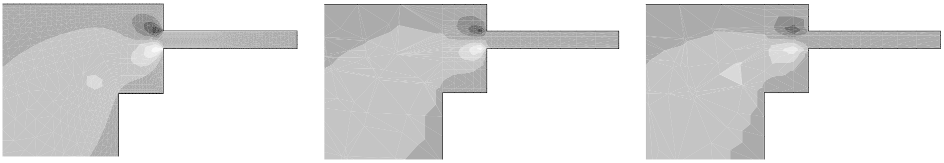
In Figure 5.10(b) and (c) the contour plots for respectively the axial and the radial components of the velocity field are plotted. It can be observed that these contour plots are also



(a) Temperatures



(b) Axial velocities



(c) Radial velocities

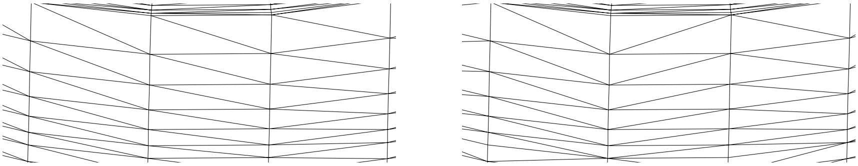
Figure 5.10: Solution fields for extremely fine, fine and coarse mesh.

very similar for the mesh sizes under consideration. In fact, for the fine mesh the maximum axial velocity (in extrusion direction) deviates by less than 1% of the maximum velocity from the exact solution field and for the coarse mesh by about 2%. Moreover, the maximum and minimum radial velocities differ by less than 1% of the maximum total velocity for both the fine and the coarse mesh.

From the results presented above it can be concluded that the meshes used in this work, which contain about 6 to 8 elements across the thickness of the profile wall and which are refined to obtain the smallest elements near the die entrance, are sufficiently fine to capture the important features of the flow field accurately. Refining these meshes to more than 8 elements across the thickness will hardly render a further improvement of the solution field. It is, however, important that the cross-section meshes are biased towards the bearing contour to generate elements that are as small as computationally feasible at the bearing inflow edge.

5.2.3 Orientation of the envelope mesh

As pointed out in Section 4.2, the direction of the diagonal that splits a quad on the envelope surface into two triangles can be chosen freely. To investigate the influence of this orientation on the computed flow field, two patterns of diagonal orientations on the envelope surface are considered. First a unidirectional diagonal orientation is applied to generate the envelope meshes (see also Figure 5.11(a)). Second the alternating diagonal orientation, that is employed throughout this thesis, is used (see also Figure 5.11(b)).



(a) Unidirectional *(b)* Alternating

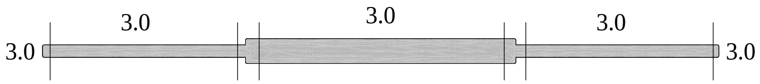
Figure 5.11: Close-up of envelope meshes for different diagonal orientations.

For both meshes the velocity fields are computed. Based on the rotational symmetry of the domain, the tangential component of the velocity field (i.e. the component in the cross-sectional plane, along the wall of the bearing) should be zero. However, for both meshes the tangential component deviates from zero. This deviation is $\mathcal{O}(10^{-4})$ times the magnitude of the local velocity vector for the unidirectional diagonal mesh, which is negligible. Nevertheless, this error is reduced by a factor 20 for the alternating diagonal orientation. Even though the tangential velocity components are relatively very small for both orientation patterns, this comparison supports the choice of the alternating diagonal orientation that is used in this work.

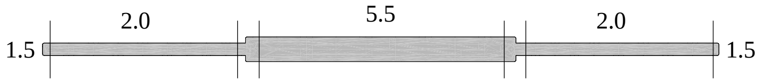
5.3 Experimental validation

In this section the results of finite element simulations will be compared to measurements that have been performed in situ at the extrusion plant of Alcoa Europe, location Drunen, The Netherlands. These validations are aimed at establishing whether the simulations can capture the trends in the influence of the bearing length on the balance of the aluminum flow that are observed in practice. A disadvantage of performing the experiments in an extrusion plant is that the control of the process parameters is not as good as in a laboratory setting. The in situ circumstances may cause unwanted variations between two pressings in, amongst others, the ambient temperature and moisture, the initial billet and die temperature and/or the extrusion speed,. However, a laboratory press was not available to realize an experimental set-up. Therefore, an experiment has been arranged such that the results can be interpreted in a reliable manner in spite of possible variations in the process parameters. This has been achieved by determining the influence of different bearing length distributions for a given profile in one pressing.

To this end, a die was designed to generate two flat, nominally identical profiles which are twice as thick in the middle as at the ends (see also Figure 5.12). The opening for the first profile has a constant bearing length along its contour, indicated in Figure 5.12(a). The



(a) Constant bearing length



(b) Varying bearing length

Figure 5.12: Profile shape for experimental validation; The bearing lengths are specified.

opening of the other profile was machined such that the bearing was considerably longer at the locations where the cross-section of the profile was thicker. This bearing length distribution (specified in Figure 5.12(b)) follows from the empirical relations which are normally used in die design.

The design of the bearing lengths was aimed at obtaining approximately equal exit velocities for both profiles. However, during the experiment the exit velocities showed a considerable difference between the two profiles. The profile from the constant bearing length opening had a higher exit velocity than that from the varying bearing length opening. In fact, the velocity difference between the two profiles was of such magnitude that after extruding one billet the length of the profile from the constant bearing length opening had a length of 52[m] whereas the profile from the varying bearing length opening was 42[m] long. This imples that the exit speed of the varying bearing length profile was 81% of that of the constant bearing length profile. This length difference was also clearly visible during the extrusion because it resulted in the folding of the profile with the higher velocity as can be seen in Figure 5.13. This unexpected difference in the exit velocities of the two profiles is very illustrative of the complexity of the die design process.

The experiments further revealed that the flow of the profile from the constant bearing length opening was not balanced. This can be concluded from the start-up sections of the profiles, that are depicted in Figure 5.14. If the profile is well balanced this section is straight. This is the case for the profile from the opening with the varying bearing length that is depicted in Figure 5.14(b). For the constant bearing length profile shown in Figure 5.14(a) the thin sides clearly lag behind the thicker center of the profile.

Also, geometrical measurements were taken from the cooled profiles. These measurements show that the constant bearing length profile was 0.5[mm] wider than the other profile. Additionally, the long cross-sectional axis of the profile from the varying bearing length opening was slightly curved whereas that of the constant bearing length profile was almost straight. This is surprising because, judging from the balancing of the profiles, the constant bearing length profile is expected to be more prone to shape deviations.

For the numerical analysis the aluminum domain associated with the die design was

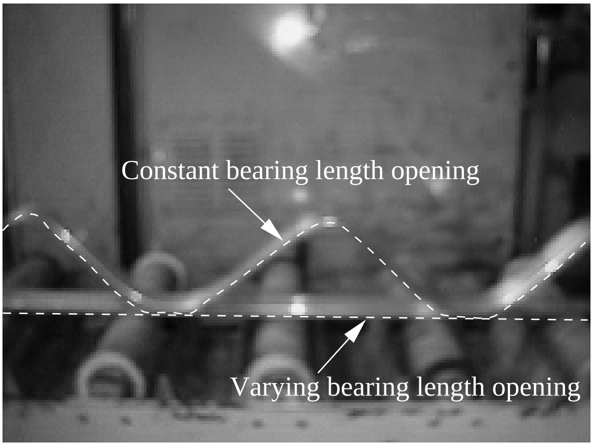
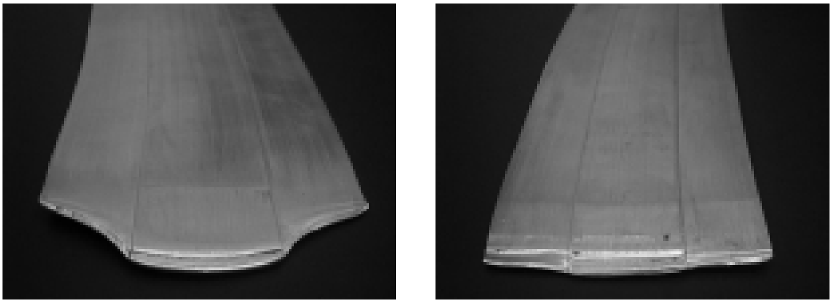

Constant bearing length opening
Varying bearing length opening

Figure 5.13: Picture of the two profiles begin extruded from left to right. The top profile exits with the higher velocity, which causes it to obtain a wavy shape in extrusion direction.



(a) Constant bearing length

(b) Varying bearing length

Figure 5.14: First parts of the profiles.

meshed rendering the discretization depicted in Figure 5.15. The different bearing length distributions for the two profiles can be clearly distinguished. This mesh was used to compute the solution fields within the aluminum domain. In the computation of the solution fields the deformation of the die is neglected since it is not expected that the die will deflect significantly in this case. Otherwise, the material parameters that have been introduced previously for the aluminum and the friction are used in the analysis.

The nominal and the computed profile shapes are plotted in Figure 5.16, where the deviations of the predicted profile shape from the desired shape are magnified with a factor 10. The exit velocities are plotted in Figure 5.17, for a cross-section of both profiles. The deformations of the profiles have been multiplied by a factor 5 for visualization purposes. The outline of the desired profile shape is plotted as a reference.

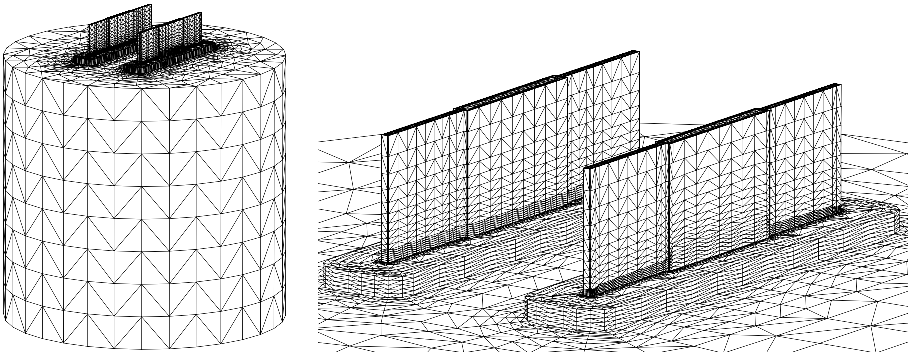


Figure 5.15: Mesh for the aluminum domain of the two exit die. The varying bearing length opening is situated in front of the constant bearing length opening.

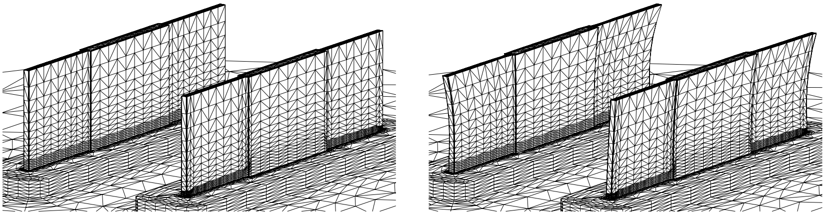
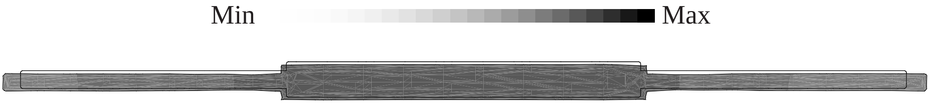
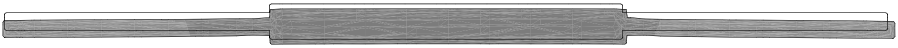


Figure 5.16: Desired (left) and computed (right) profile shapes. The deviations from the desired shape are magnified with a factor 10.



(a) Constant bearing length



(b) Varying bearing length

Figure 5.17: Exit velocities in a cross-section of the profiles. The deformation of the profiles are magnified with a factor 5.

It should be noted that the velocity field is not exactly symmetrical, as can be observed in Figure 5.17(b) where the computed profile for the varying bearing length is slightly shifted to the right. This is caused by a rigid body rotation of the profile at the exit of the die that is due to an asymmetry in the flow field. The error in the flow field which causes the asymmetry is less than 0.1% of the exit velocities and therefore negligible. The asymmetry in the flow field can be attributed to the asymmetrical mesh that is used for the analysis. The effects of this asymmetry become noticeable because the deformations of the profile are scaled.

From Figure 5.17 it can be concluded that the trends that were observed in the extrusion practice are predicted well. The difference in exit velocities is captured correctly by the computations. The average computed exit velocity for the profiles from the constant bearing length opening is 19.2$[\mathrm{m\,s^{-1}}]$, while that of the varying bearing length profile is 16.7$[\mathrm{m\,s^{-1}}]$, which is 87% of the velocity of the constant bearing length profile. The computed ratio between the exit velocities is close to the ratio that was measured in the experiment (87% vs. 81%). Also, the balancing within each profile is represented accurately. For the constant bearing length the velocities are higher in the center of the profile than at the tips, while for the varying bearing length the exit velocities are practically constant throughout the cross-section. Finally, the differences in the shape are computed correctly as well. In Figure 5.17(a) the computed profile cross-section is wider than the desired shape, while the computed and the desired cross-sections are almost identical in Figure 5.17(b). Additionally, the profile in Figure 5.17(b) is slightly bent down at the tips, which matches the deformations observed in the experiments.

Chapter 6

Extrusion of complex profiles

In this chapter three examples will be presented of finite element simulations of the extrusion of complex profiles. The first example concerns a flat profile that has been extruded in practice. The die that was designed for this profile did not perform satisfactorily during the extrusion. The finite element simulation that will be discussed, predicts the observed imbalance and gives a good indication of how the die should be corrected. In the second example an existing hollow profile with very thin walls and two chambers will be analyzed. The finite element model predicts that the die for this profile renders an almost completely balanced flow field, which was observed in aluminum extrusion practice as well. The third example concerns the extrusion of a fictive polymer profile. This example is included to underline the flexibility of the modeling tool that has been presented in this thesis. Also, this example demonstrates the necessity to include the shape of the outflowing profile in the model.

6.1 Flat profile

In this section the extrusion of the flat profile depicted in Figure 6.1, will be analyzed. The die that was initially designed for the extrusion of this profile did not produce a balanced exit flow, which resulted in a profile that was too severely curved in the extrusion direction. Based on the bent shape of the profile it was concluded that the left tip of the profile lagged behind. The die was adjusted accordingly; i.e. by locally enlarging the cross-section of the pocket to stimulate the flow of aluminum to the left tip of the aluminum. However, the modified die gave no improvement and still failed to produce acceptable profiles. A renewed adjustment appeared to be necessary.

Determining the shortcomings of a die design based on the geometry of the extruded profile is very difficult. Often conflicting theories can be applied to explain the same observed failure mechanism of the die. As a result, it is not uncommon that, for a complex die such as the one considered in this section, the wrong theory is adhered and the design is altered erroneously, as was the case in this example. The extrusion simulation that will be presented here is performed to understand why the alternations to the die did not improve the balance of the outflow. Also, the solution fields from this simulation will indicate how the die should

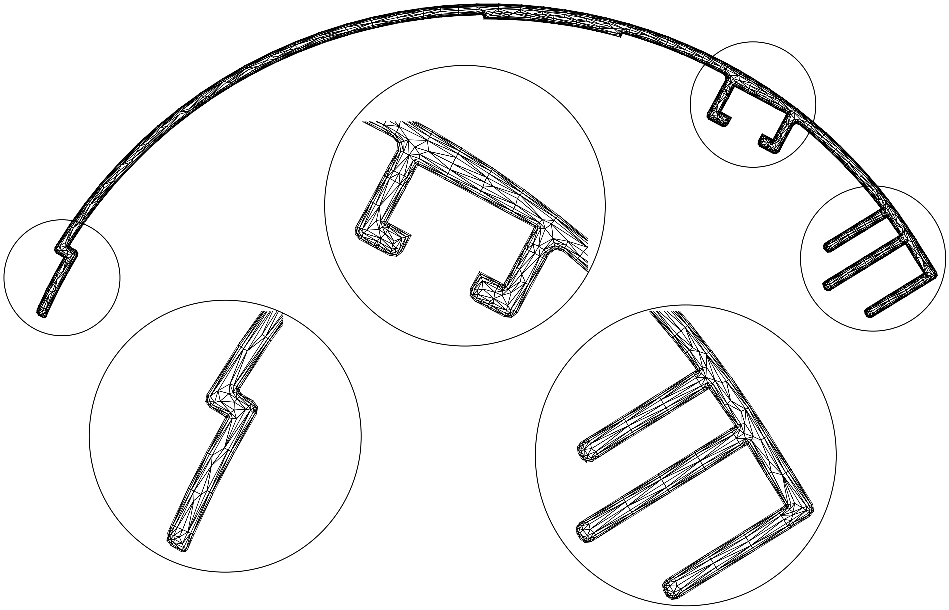


Figure 6.1: Mesh of the bearing cross-section. The circles correspond to enlarged details.

be adjusted to obtain an appropriate profile geometry.

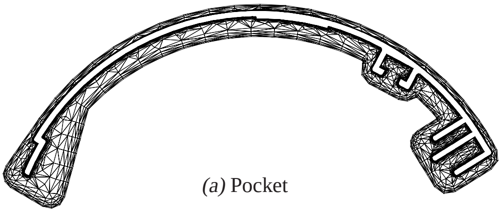
The die for the profile in Figure 6.1 is more complicated than a standard die. It consists of the normal bearing and pocket, but it has an additional pocket-like recess that is positioned between the billet and the actual pocket. This extra recess is called a feeder. The cross-sectional surface mesh for the aluminum in the bearing is plotted in Figure 6.1 and the meshes for the pocket, the feeder and the billet are shown in Figure 6.2.

Combination of these cross-sectional meshes with the appropriate envelope surfaces renders an enclosing volume that can be filled with tetrahedra using the expansion and unstructured mesh generators presented in this work. The resulting mesh of the aluminum domain contains 260,000 elements and is depicted in Figure 6.3.

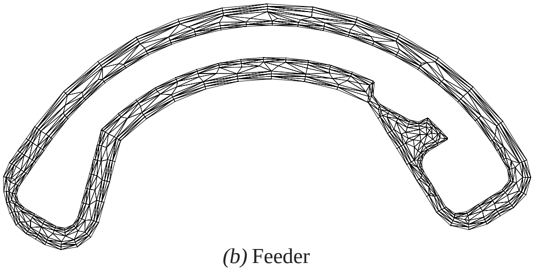
The computation of the solution fields associated with this mesh was performed in approximately 2.5 days of computer time (real time, i.e. not CPU time) on an Silicon Graphics O200 work station with a R10,000 processor that operates at 195 MHz. The computation required about 6 iterations over the 3 field problem to arrive at the converged solution fields. It should be noted that within these 3 field iterations the deformation of the die was not incorporated because it was expected that the die would only deflect marginally.

In Figure 6.4 the temperature field is plotted in the computed profile shape. The deviations of the outflow shape from the nominal profile shape are magnified with a factor 10 for visualization purposes. The velocities at which the aluminum exits the die are plotted in Figure 6.5 in a cross-section of the deformed profile.

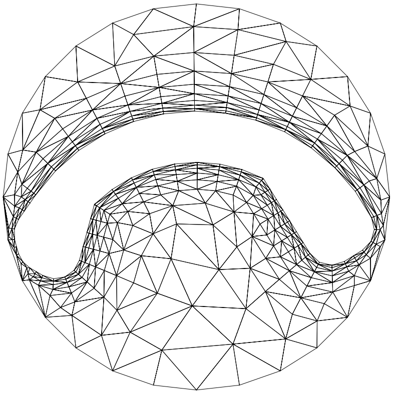
In Figure 6.4 it can be seen that in the computations the section of the profile on the right



(a) Pocket



(b) Feeder



(c) Billet

Figure 6.2: Cross-section surface meshes.

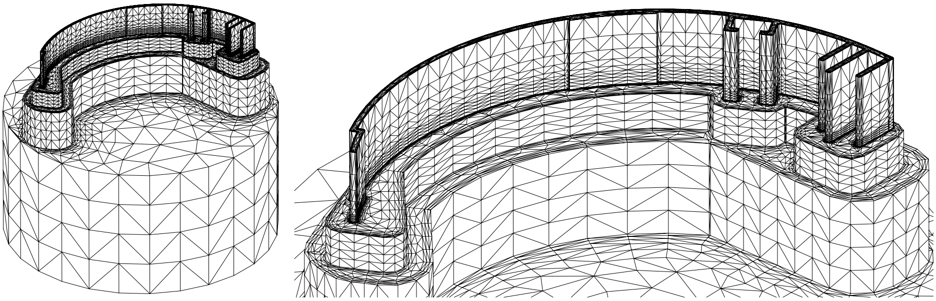


Figure 6.3: Mesh of the aluminum domain.

hand side bends inward as has been observed in practice. This is an indication that the exit velocities on the right hand side of the profile are too high. A logical conclusion is then that the velocities of the left tip should be increased by stimulating the flow of aluminum to this part of the profile. However, the exit velocity field in Figure 6.5 shows that this conclusion is not completely correct.

Based on the velocities in extrusion direction, depicted in Figure 6.5, the following can be concluded. The aluminum at the right hand side of the profile does indeed exit the bearing opening at a flow rate that is higher than the average flow rate. However, the aluminum at the left hand side flows at velocities that are too high as well. It is the top of the arc that lags behind and probably causes the most serious distortions of the profile. Moreover, the flow field within the fork on the right hand side of the profile is unbalanced as well. The tips of the fork lag behind slightly which causes the entire fork to bend inwards. In conclusion, this simulation suggests that the die should be altered to allow aluminum to flow more freely to the arc of the profile and to the tips of the fork. The fact that the suggested alterations to

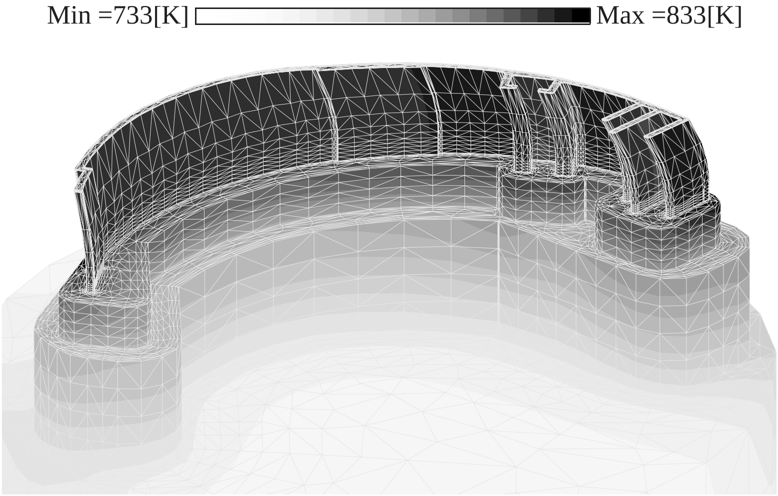


Figure 6.4: Temperature field for the deformed profile. The deformations of the profile are magnified with a factor 10.

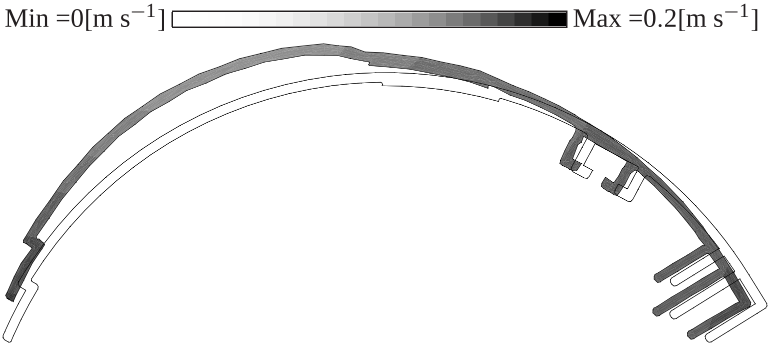


Figure 6.5: Velocity field for a cross-section of the deformed profile. The deformations of the profile are magnified with a factor 10.

the design are very different from the adaptations that were performed in reality, probably explains the failing of the redesigned die that occurred in practice. A new simulation with a die that has been altered along the lines suggested here would have to be performed, to assess the correctness of this assumption.

6.2 Hollow profile

In this section the extrusion of a complex hollow profile is simulated. The profile consists of two hollow chambers as can be seen in Figure 6.6 and is relatively thin-walled, while the wall thickness varies with a factor 2 within the cross-section of the profile. The combination of the multiple chambers, the thin walls, and the wall thickness variations complicates the design of the die considerably. The die design that is used in the simulation was obtained in extrusion practice after several trial and error iterations and renders an aluminum flow that is balanced to such a degree that the geometry of the profile meets the shape specifications of the customer.

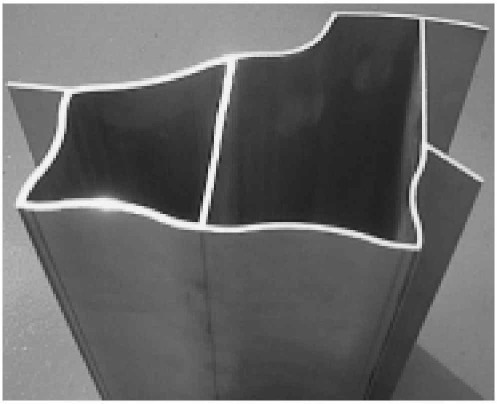


Figure 6.6: Hollow profile with 2 chambers.

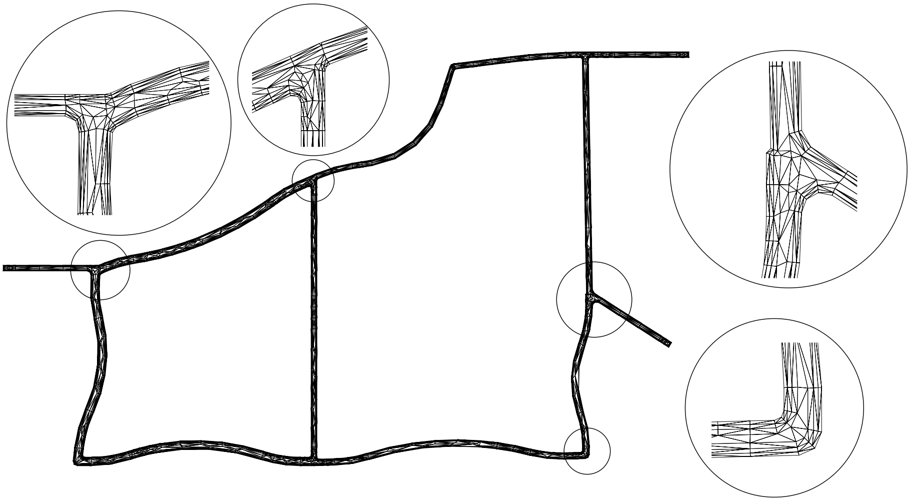
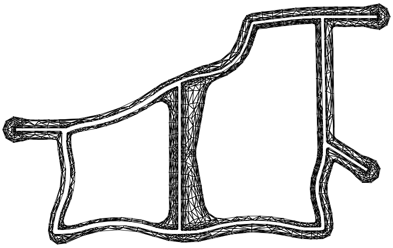
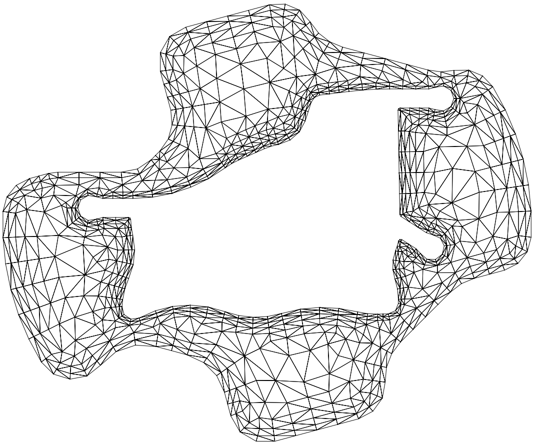


Figure 6.7: Mesh of the aluminum cross-section in the bearing. The circles correspond to enlarged details.

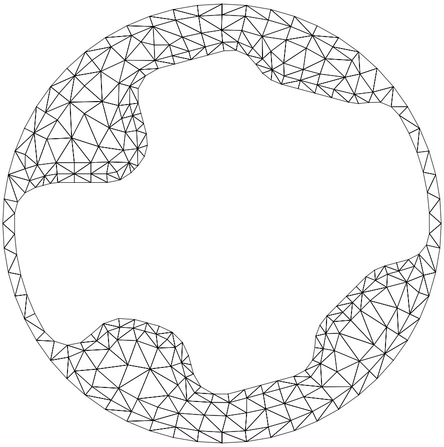
It was already indicated in Section 4.6 that dies for hollow profile are considerably more sophisticated than those for flat profiles. This is mainly caused by the fact that these dies consist of two parts instead of one, where the second part is the bridge part. To mesh the aluminum domain occupying the hollow die for this profile the approach that was proposed



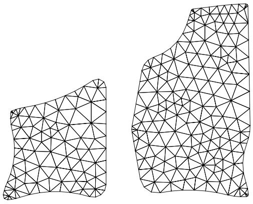
(a) Pocket



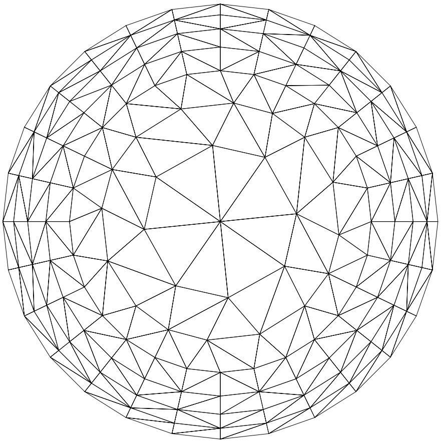
(b) Welding chamber



(c) Billet



(d) Bottoms of cores



(e) Ram

Figure 6.8: Cross-section surface meshes (billet and ram surfaces are scaled differently).

in Section 4.6 was adopted. In this approach the domain is initially discretized to obtain a provisional mesh. In this provisional mesh the presence of the legs is neglected. The first relevant cross-section surface of the aluminum in the provisional mesh, is the bearing surface of which the mesh is shown in Figure 6.7. The second surface is the pocket cross-section of which the mesh is depicted in Figure 6.8 consisting of one external surface related to the die plate and two internal surfaces related to the core of the bridge part. The third and fourth surfaces are cross-sections of the welding chamber and the billet, respectively. The meshes of these surfaces are also depicted in Figure 6.8. The fifth surface is constructed out of the bottoms of the cores and the mesh is depicted in Figure 6.8 as well, along with the sixth and final mesh of the ram surface.

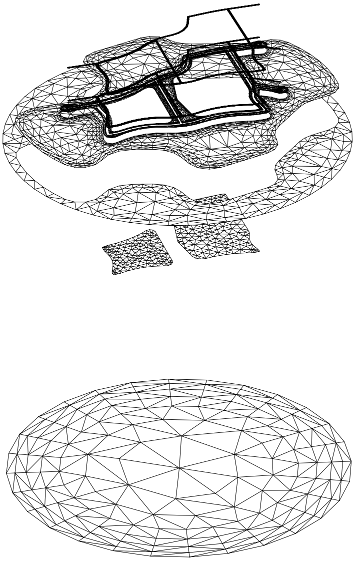


Figure 6.9: Positioned cross-section meshes.

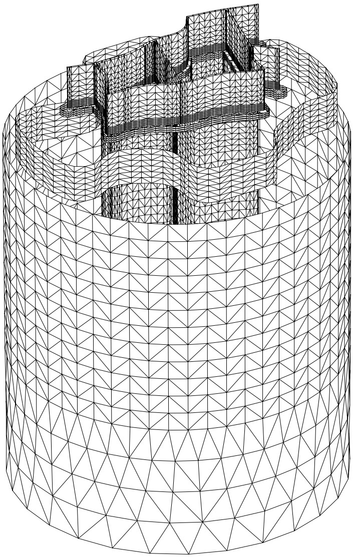


Figure 6.10: Envelope meshes.

After these cross-section meshes are positioned at the correct heights (see Figure 6.9) they can be combined with the appropriate envelope meshes (see Figure 6.10) to create the volume mesh. The resulting discretization is the provisional mesh that is drawn in Figure 6.11 in which the geometry of the legs of the bridge part visualized in Figure 6.12, still has to be incorporated. The provisional mesh contains 350,000 elements.

The geometry of the legs of the bridge part is taken into account by eliminating the appropriate elements and repositioning the remaining nodes as explained in Section 4.6. Since in this method the geometry of the bridge part is only used to determine whether points are positioned inside or outside the bridge part, the quality of the surface mesh of this bridge part is allowed to be very low. This enables the application of standard CAD surface meshes, which are generally of extremely poor quality, for the representation of the bridge part geometry. Clearly, this is advantageous because no remeshing of the bridge part surface is required. After incorporation (of the legs) of the bridge part and the stretching of the bearing the mesh in Figure 6.13 is obtained. This resulting mesh contains 320,000 elements.

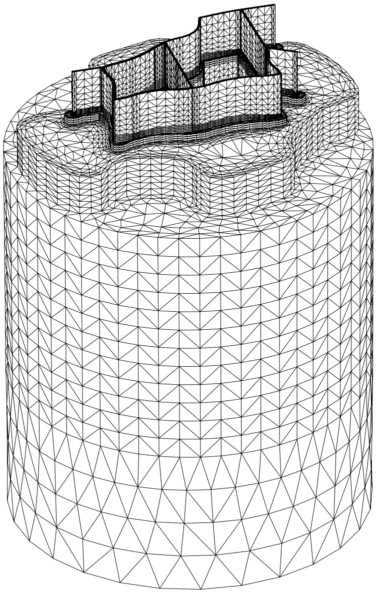


Figure 6.11: Provisional volume mesh.

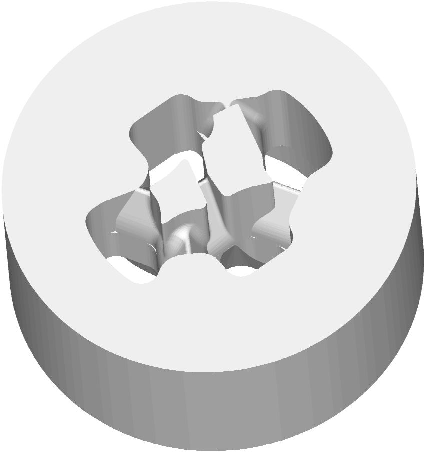


Figure 6.12: Geometry of the bridge part.

The computation of the solution fields associated with this mesh was performed in approximately 3 days of computer time (real time, i.e. not CPU time). Within the computation less than 10 iterations over the 3 field problem were necessary to calculate the converged solution fields. It should be noted that within these 3 field iteration process the deformations of the die were not incorporated because it was expected that these would be relatively small.

In Figure 6.14 the temperature fields are plotted in the computed profile shape. The deviations of the outflow shape from the nominal profile shape are magnified with a factor 10 for visualization purposes. It can be observed that, contrary to flat profiles, the aluminum heats up considerably before entering the pocket and the bearing. This heating occurs in the legs, where the shear rates are relatively high. Also, it can be concluded that the computed profile hardly deviates from its nominal shape. This corresponds to the well balanced profiles that were extruded in practice with this die design.

The fact that the profile is well balanced is further illustrated in Figure 6.15, where the velocities at which the aluminum exits the die are plotted in a cross-section of the deformed profile. The differences between the computed shape and the nominal shape have been magnified with a factor 10. As can be observed, the only significant deviation from the nominal shape, apart from rigid body movements, occurs in the part of the profile that separates the two chambers. The middle of this part is slightly off-centered to the left, which is precisely what is observed in the extruded profile. It can also be concluded that the velocity of the aluminum in extrusion direction is practically constant throughout the cross-section. It should be noted that this prediction of an almost constant exit velocity is a non-trivial result, considering the flow imbalances that were observed in the results of the previous section.

Until now there was a complete lack of knowledge on the distribution of the aluminum

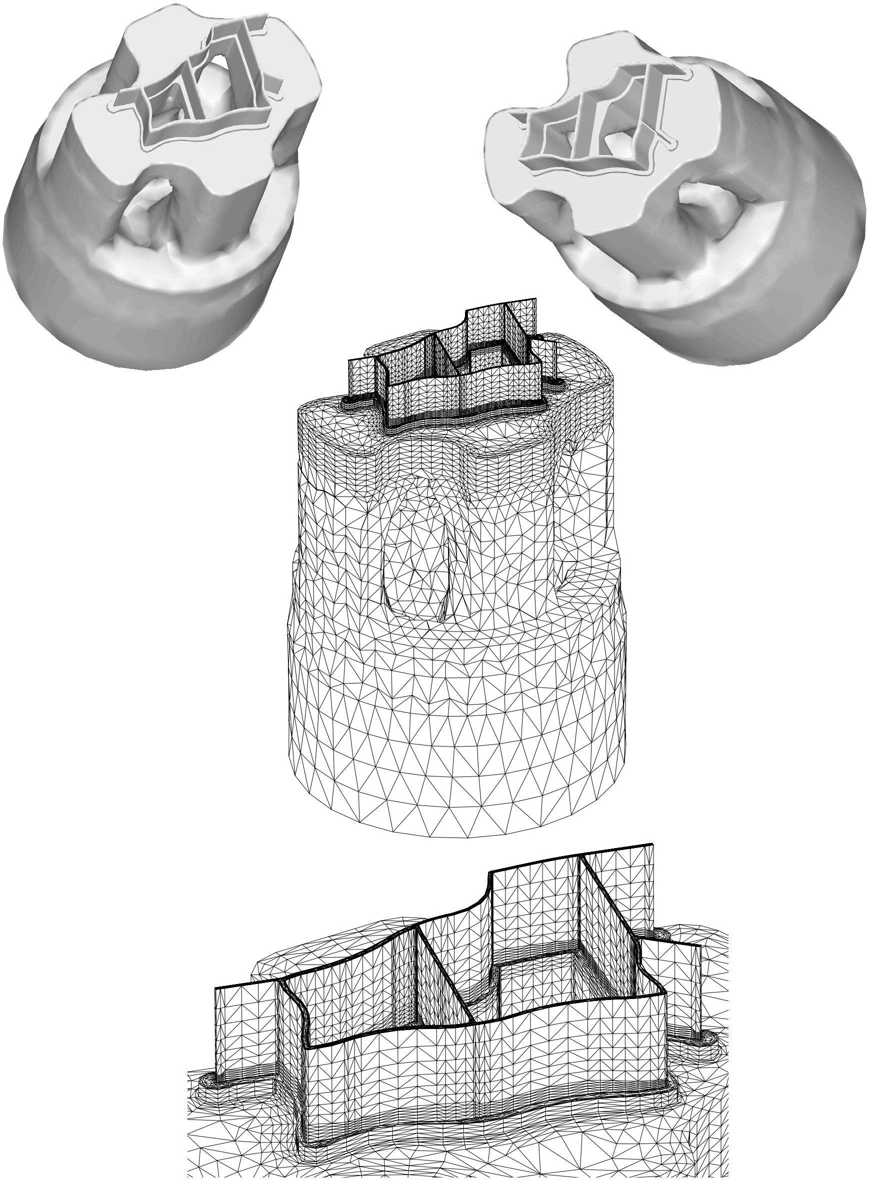


Figure 6.13: Top: The shape of the aluminum domain. Bottom: The final mesh of the aluminum domain. In the enlarged picture the varying bearing length can be observed.

flow in the openings between the legs of the bridge part. This distribution is of practical interest because imbalance in this area of the die can easily lead to an unbalanced exit flow. Also, the flow rate distribution between the legs influences the position in the profile of the

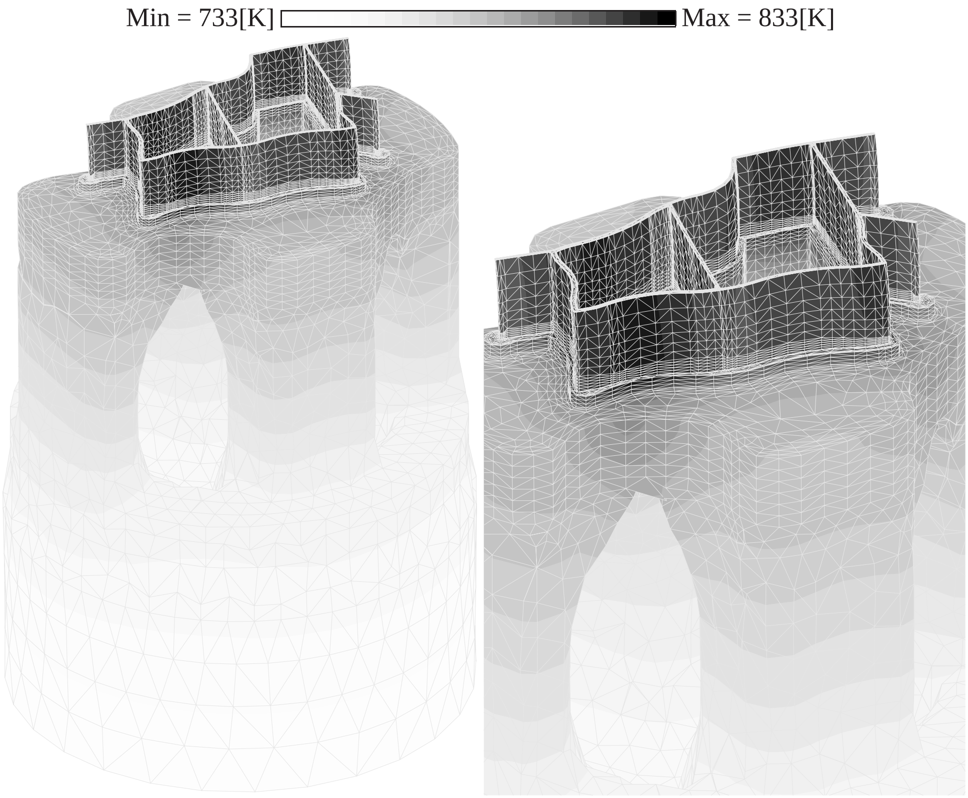


Figure 6.14: Temperature field of the deformed profile. The deformations of the profile are magnified with a factor 10.

surface where the aluminum has welded together again behind the legs. Clearly, the flow can not be measured in situ because of the high pressures and temperatures in these parts of the aluminum domain. However, now this distribution can obviously be determined, using the flow fields from the simulations. In Figure 6.16 the contours of the velocities in extrusion direction are plotted for a number of cross-sections of the aluminum domain. It can be seen that the velocities in each of the legs are different. This implies that the pressure distribution is different in the holes which can cause an undesirable bending of the bridge part.

6.3 Polymer extrusion

In this section the extrusion of a polymer through a fictive die is analyzed. This example is included for a number of reasons. First, it illustrates the versatility of the finite element model that has been presented in this thesis and underlines the potential of the meshing algorithms. Second, this example emphasizes the necessity of modeling the shape of the extrudate. Last, the method that is used to discretize the polymer domain elucidates the advantages of dis-

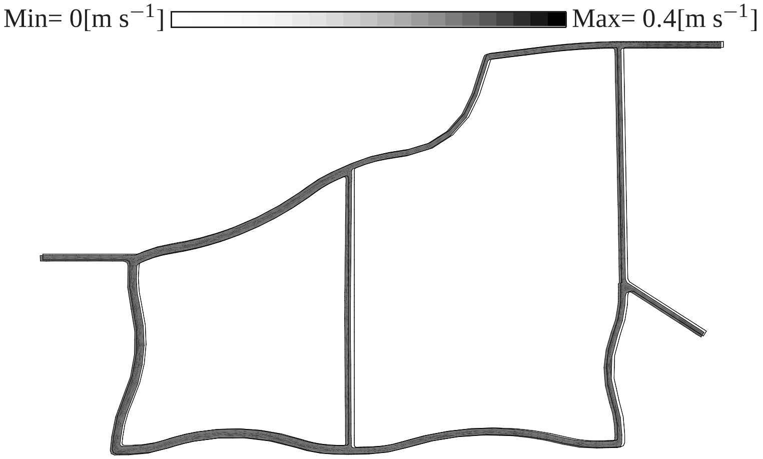


Figure 6.15: Velocity field in a cross-section of the deformed profile. The deformations of the profile are magnified with a factor 10.

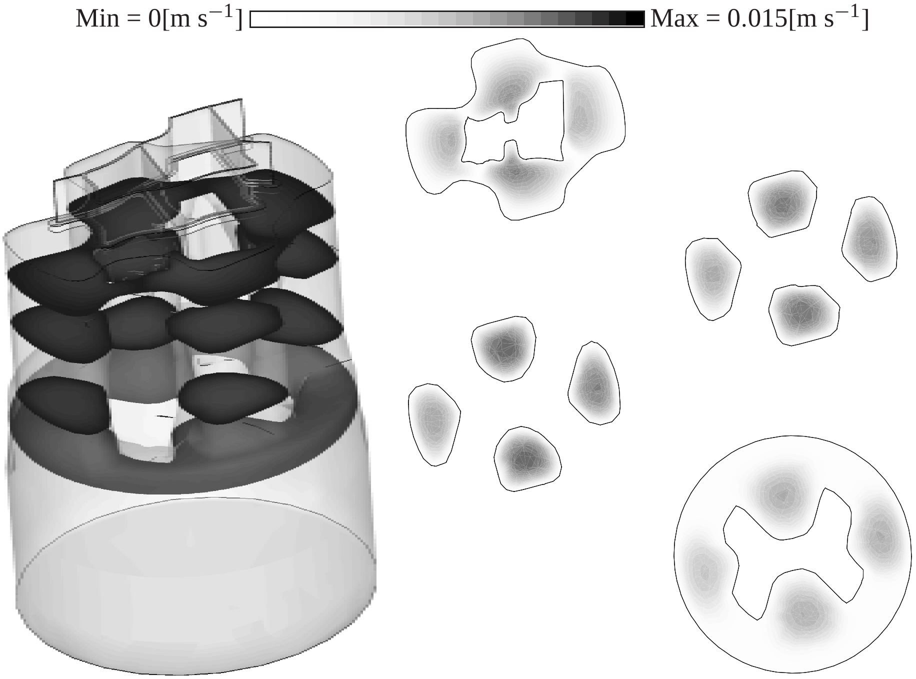


Figure 6.16: Left: Cross-sections in the aluminum domain. Right: Velocities in extrusion direction in the cross-sections.

cretizing the Lagrange multipliers that enforce stick on a surface, through the introduction of control points, as has been discussed in Section 3.1.

In the example the polymer acrylontrile-butadiene-styrene (ABS) is used that shows a strain rate and temperature dependent viscosity. This dependency can be formulated as (v.d. Vegt, 1994; Zoetelief, 1995):

$$\eta = \eta_0 \frac{1 + (\lambda_2 \dot{\gamma})^{-1}}{1 + (\lambda_1 \dot{\gamma})^{n_1}}$$

with

$$\eta_0(T) = H\beta(T), \qquad \beta(T) = e^{\frac{-c1(T-T_0)}{c2+T-T_0}}$$

$$\lambda_1(T) = B_1\beta(T), \qquad \lambda_2(T) = B_2\beta(T)$$

where the definition of $\dot{\gamma}$ has been introduced in Section 2.2. The relevant parameters for ABS are quantified in Table 6.1. The fact that the viscosity of this polymer is dependent on the strain rate and the temperature, as is the case for aluminum, makes the incorporation of the material model into the numerical procedure that has been developed for aluminum extrusion, straightforward.

c_1	7.04	-	c_2	119.7	K
B_1	0.11	s	B_2	3.8	s
n_1	0.79	-	λ	0.17	W $\mathrm{m}^{-1}\mathrm{K}^{-1}$
H	$9.5 \cdot 10^3$	N s m^{-2}	ρ	945	kg m^{-3}
T_0	473	K	c_p	$1.4 \cdot 10^3$	J $\mathrm{kg}^{-1}\mathrm{K}^{-1}$

Table 6.1: Parameters for ABS

The geometry of the extrusion die for the ABS is defined by a combination of cross-sections, as depicted in Figure 6.17. As can be seen, a directional refinement is applied in the surface meshes to obtain thin elements perpendicular to the die wall. This way the high velocity gradients in the polymer close to the die can be captured accurately while keeping the total number of elements relatively low. Since the number of line segments along the contour of a cross-section mesh is not constant for subsequent cross-sections, the expansion mesh generator cannot be applied to generate the envelope meshes. Therefore, an unstructured surface mesh generator has been used for the envelopes. This unstructured surface generator employs Coon's point placing algorithm to generate a smooth surface between the contours (Segal, 1983). The polymer volumes in the die are meshed with tetrahedra using an unstructured generator, whereas the meshes of the polymer volumes in the bearing and the outflow are generated using the expansion generator. Several perspectives of the resulting mesh are depicted in Figure 6.18.

To investigate the effect of the free surface deformation on the exit velocity of the polymer, two computations have been performed. In the first analysis the altering shape of the free surface was not accounted for. In the other analysis the actual geometry of the extrudate, being the shape of the outflow domain, was determined iteratively using the 3-field approach. In both extrusion simulations, the die was assumed to be thermally insulated and the polymer

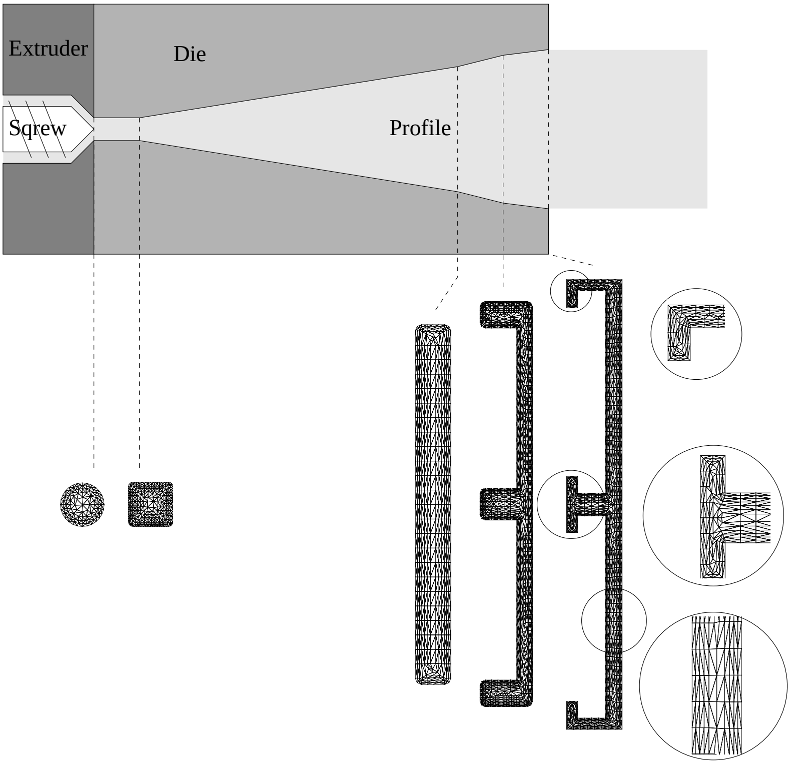


Figure 6.17: Cross sections of the die with corresponding meshes.

was modeled to stick to the surface of the die. The polymer entered the die with a velocity of 0.04[m s^{-1}] and a temperature of 493[K]. In Figure 6.19 the shapes of the extrudate surfaces are plotted for the nominal outflow geometry and the computed outflow geometry. It should be noted that in Figure 6.19(b) no additional magnification of the differences between the computed and the nominal shape has been applied.

In Figure 6.20 it can be seen that taking the geometry of the outflow surface into account has a noticeable effect on the velocity distribution at the outflow. It can be seen that when the geometry of the outflow shape is incorporated in the model, the thin sections of the profile connected to thick parts will be pulled from the die. This results in a reduced cross-sectional thickness and also in a higher exit velocity. This is clearly visible in the left and right legs. Also thicker parts will be slowed down by the thin sections which in turn results in an increased cross-sectional thickness. This effect can be seen in leg of the middle 'T'. Of course, the changes in the velocity field have a noticeable impact on the temperature field as well (van Rens et al., 1999c).

From Figure 6.20 it can be concluded that, for this example of polymer extrusion, it is necessary to incorporate the actual shape of the outflowing polymer into the computations in order to obtain accurate predictions of the velocity field. The reason that this effect is so

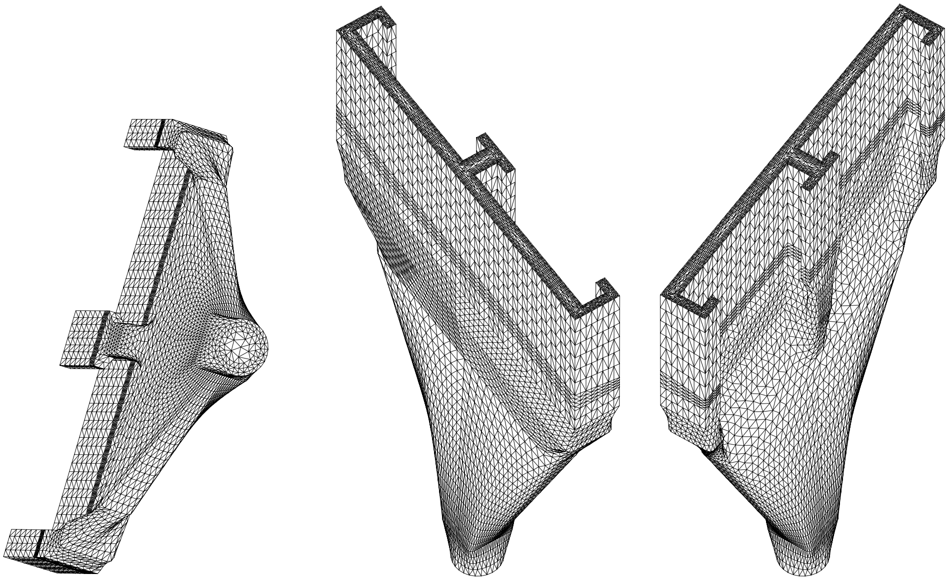
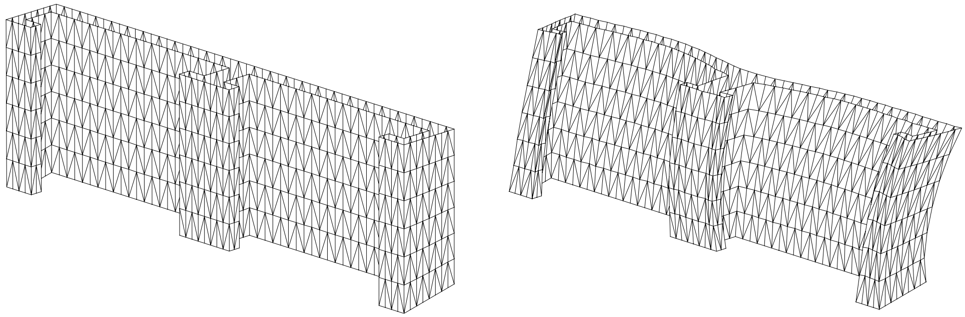


Figure 6.18: Several views of the polymer mesh.



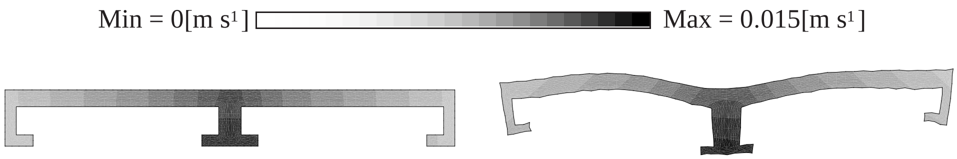
(a) Nominal extrudate shape

(b) Actual extrudate shape

Figure 6.19: Shape of polymer surface.

pronounced for this analysis, is that the computed extrudate shape differs considerably from the nominal shape.

In the analyses presented above use has been made of a fixed inflow velocity and temperature. However, in real polymer extrusion practice the inflow velocity is not prescribed through a ram speed, as it is in aluminum extrusion. In polymer extrusion the material is



(a) Initial extrudate shape

(b) Actual extrudate shape

Figure 6.20: The exit velocity in a cross-section of the extrudate.

forced through the die using a (twin) screw extruder of which the rotational speed is prescribed (see also Figure 6.17). Therefore, the analysis is now extended with the incorporation of a twin screw extruder that forces the polymer through the die.

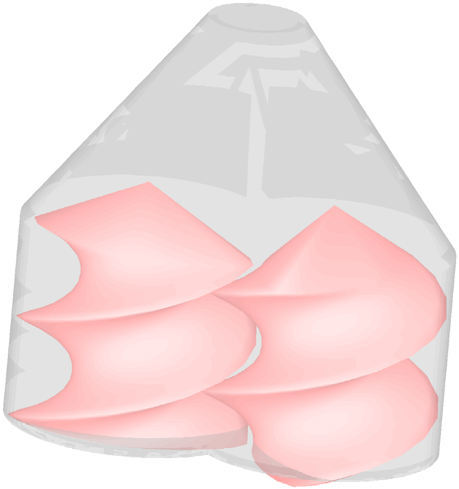


Figure 6.21: Last section of a twin screw extruder.

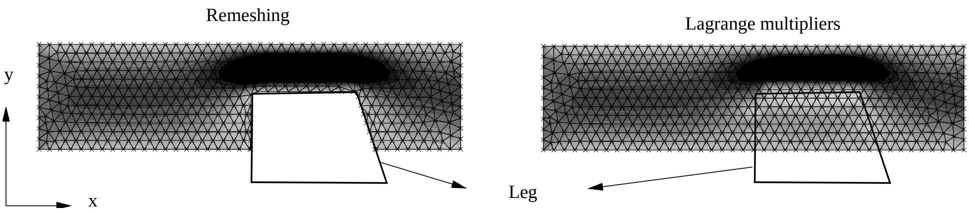


Figure 6.22: Velocity in x-direction for the flow around a simplified leg. The leg geometry is incorporated through remeshing as introduced in Section 4.6 (left) and through Lagrange multipliers (right).

To this end the polymer domain is enlarged such that it covers the last section of the twin screw extruder as well (see Figure 6.21). It is extremely difficult to obtain a suitable mesh for the polymer domain in Figure 6.21 because of the complex shapes of the screws that have very small clearances between the flights. Therefore, initially the polymer domain is meshed, neglecting the presence of the screws. The shape of the domain is then simple enough to be meshed using the methods presented in this work. The effect of the screws on the polymer is then incorporated by employing Lagrange multipliers on the surfaces of the screws. The Lagrange multipliers are comparable to those introduced in Section 3.1.4, and enforce that the polymer sticks to the surface of the screws in a set of control points. The introduction of Lagrange multipliers for the incorporation of complex geometries is different from the method introduced in Section 4.6 which consists of element elimination and node repositioning. However, both methods render the same results as can be seen in Figure 6.22 where the difference between the two methods is illustrated for the simplified leg geometry that was introduced in Section 4.6. For both cases the contours of the velocities in x-direction are very similar even though the computational domain on the right hand side of Figure 6.22 does not account for the leg.

Clearly, as is also the case in the example in Figure 6.22, the screw surfaces do not coincide with nodal points of the polymer mesh. This makes it impossible to choose the control points coincident with the nodes in the mesh, as has been done for the friction surfaces in aluminum extrusion. Therefore, the control points are positioned on the intersections between edges of the polymer mesh and the surfaces of the screws. To ensure that the resulting system of equations remains well posed and does not become overconstrained, the control points are not added on every intersected edge. The edges on which control points are positioned are selected such that at most one of these edges is connected to a node in the mesh.

The solution fields have been computed using a simplified material description: The polymer has been modeled as a Newtonian liquid with constant viscosity. Moreover, the deformation of the outflow surface of the polymer has not been considered. The velocity boundary conditions are defined such that the polymer sticks to the (stationary) walls of the extruder and the die, as well as to the (rotating) walls of the screws. The velocity field that has been obtained in this simulation is used to plot the particle tracks depicted in Figure 6.23.

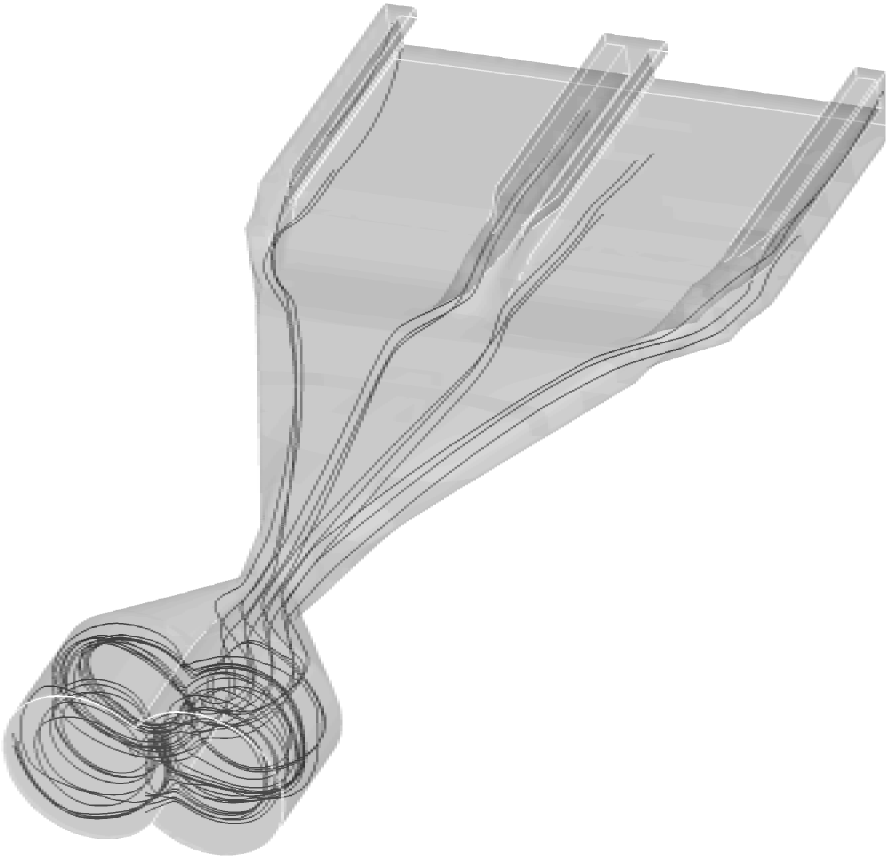


Figure 6.23: Particle tracks for the constant viscosity polymer.

Chapter 7

Conclusions and recommendations

In aluminum profile manufacturing there is a strong desire to reduce costs associated to the design of a die. A considerable portion of these costs is related to the physical trial-pressings that have to be performed before a satisfactory die design is obtained. As a result, the cost reduction can be achieved by replacing these trial-pressings by finite element computations. In these computations a trial-pressing is simulated to assess the success or failure of a given die design, which will generally be cheaper, faster and more informative. In this thesis a finite element model has been presented with which the extrusion of complex (possibly hollow) profiles can be analyzed within acceptable time spans and with a high level of reliability. In the following, first conclusions with respect to the model and its performance will be drawn. Next, recommendations will be made to improve the model, to make better use of the available solution fields, and to obtain these solution fields faster.

7.1 Conclusions

The finite element model that has been presented in this thesis is able to accurately predict the performance of extrusion dies, irrespective of the type (flat or hollow) or the multiplicity (single or multiple opening) of the profile they are designed for. For all the profiles that have been examined in this thesis the numerical predictions concur with the phenomena that are observed in the extrusion practice for these profiles. Moreover, the models are constructed in such a way that these predictions require only a small amount of pre- and post-processing time, and that the computations can be completed within acceptable time spans. In the following, conclusions will be drawn for a number of important aspects of the model.

Based on order of magnitude estimations for the physical entities associated with extrusion, the conservation laws of mass, momentum and energy can be reformulated to arrive at an incompressible Stokes flow problem that is coupled to a convection-diffusion thermal problem. The Eulerian framework that is adopted to describe the flow problem in a finite element context has proven to be suitable to model high deformations of the aluminum while avoiding severe distortions of the mesh. Furthermore, the velocity based formulation that inherently follows from the spatially fixed Eulerian framework can be easily combined with the non-Newtonian fluid model that describes the constitutive behavior of the aluminum. A drawback of a spatially fixed reference frame is that it complicates the incorporation of the moving boundaries that occur at the aluminum outflow and at the interface with the flexible die. This drawback is overcome by employing an iterative three field approach to capture the influence of the changing boundaries on the aluminum flow. This three field approach is found to be stable and to have a satisfactory convergence rate.

The flow problem is discretized through the application of the MINI element which consists of a tetrahedral element with a linear interpolation for the pressure and a linear velocity interpolation which is enriched by a bubble contribution. The shape of this bubble has been taken quadratic to obtain the best results with the highly distorted elements that occur in the discretization of the aluminum domain. The MINI element is of crucial importance for efficient finite element analyses because it enables the use of robust mesh generators as a result of its tetrahedral shape while the piecewise linear interpolation of both the pressure and the velocity ensures that the number of degrees of freedom remains manageable, even for highly complicated meshes.

To model the frictional interaction of the aluminum with the die, the interface surface is discretized and Lagrange multipliers are introduced in the nodes of the surface. These multipliers impose the Coulomb friction at the nodal level. The discretization of the interface surface normal is performed using a new, iterative method that renders more accurate results near sharp edges in coarse grids. Without this new method of defining the normal, the meshes would have to be refined to such a degree near the sharp edges in the die that the analysis of complex profiles would be practically infeasible. Numerical experiments indicate that the coefficient that quantifies the frictional behavior has a considerable influence on the predictions of the model and that accurate results can be obtained by using a value of 0.4 [-] for this coefficient.

The system of equations that describes the entire aluminum extrusion process is split into separate subsystems for the problems associated to the flow, the temperature, the outflow shape, the die deflection and the mesh displacement. These subsystems are solved separately and coupled through a staggered approach. Numerical experiments indicate that each of the problems in the staggered algorithm has a noticeable influence on the outcome of the simulations. However, the effect of the die deflection on the predicted profile shape is the least pronounced. The staggered approach is well stabilized and generally only six to eight iterations are necessary to obtain a converged solution. Moreover, the Picard-iteration process that is applied, is well suited to linearize the strongly non-linear flow and temperature problems that have to be solved within each consecutive iteration.

The dedicated meshing algorithms that have been developed in this work are of paramount importance in generating meshes for the complex domains associated with extrusion. The algorithms generate tetrahedral meshes that are directionally and spatially refined to capture the solution fields accurately while the number of elements is kept to a minimum. This makes it possible to analyze the flow of aluminum through complex dies within reasonable time spans. The mesh generation is split into four steps, being the meshing of the surfaces, the meshing of the volumes, the stretching of the bearing and the incorporation of the leg geometries in case of hollow profiles. This split results in four manageable meshing problems rather than one enormously complex problem. As a result, it is possible to discretize the 3D domains associated with any conceivable die design, while requiring minimal user effort.

Also, the combination of a fast mesh generator and a relatively small number of generated elements allows the analysis of a multitude of die designs within a limited time span.

7.2 Recommendations

For the finite element analyses of extrusion presented in this thesis, three major areas of improvement can be identified. First, systematic, in situ experiments have to be performed to validate the results of the simulations in a structured manner. Second, the solution fields that result from the analysis can be exploited to a larger degree than has been done in this work. A number of possible extensions can be obtained by performing post-processing computations on the solution fields for the flow, the temperature, and the die deflection. Third, the solution time can be reduced drastically by using parallelization techniques. This would make it feasible to perform extrusion computations as a part of the die design process. Each of these areas of improvement will be discussed briefly below.

7.2.1 Experimental validation

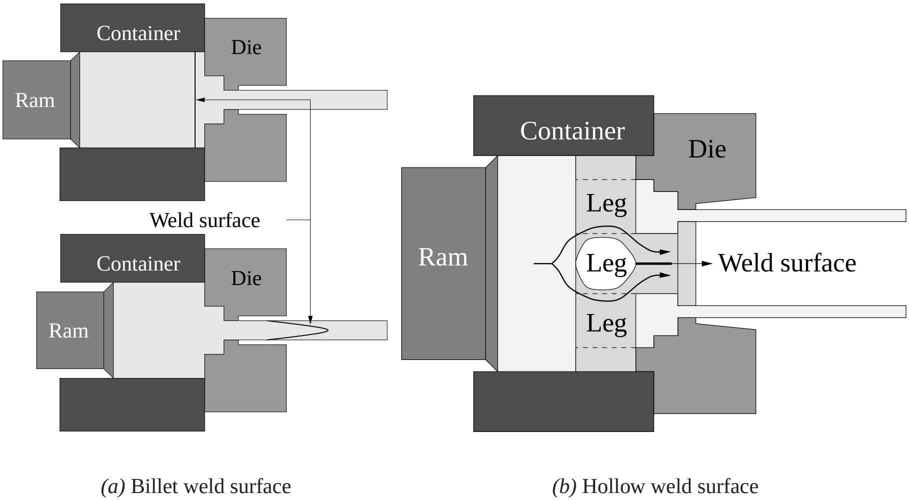
Even though the finite element simulations of extrusion have proven to predict the extrusion process accurately for the die designs considered in this thesis, the model has not been subjected to a sufficiently large amount of experimental validation. A structured comparison between experimental data and the computed flow and temperature fields can be used to assess the correctness of the constitutive models that have been adopted for both the aluminum and the friction. Also, these experiments can be designed to verify the thermal boundary conditions that are imposed on the aluminum as well as the mechanical boundary conditions that are imposed on the die.

Since the in situ measurements of forces, pressures and displacements is complicated as a result of the high pressures and temperatures that are associated to the extrusion process, most variables can only be derived from measurements that are performed on the cooled profile. It is therefore important that, when measuring the (small) effects of a process variable on the geometry of the profile, each profile is extruded and cooled under exactly the same circumstances. This can be achieved by using the multiple opening dies that have been used in this thesis for experimental validation. For such dies the process conditions are the same for each of the profiles that have to be compared, which will improve the reliability of the evaluation.

7.2.2 Exploitation of the solution fields

The solution fields for the velocity, the pressure and the temperature can be used to compute additional, derived quantities. Examples are the shape and position of weld surfaces, the sizes and distribution of aluminum crystals in the profile, and the expected life of a die.

Weld surfaces are formed at those places where two aluminum surfaces come in contact during extrusion. This occurs when a new billet is inserted into the container and this billet is pressed against the remainder of the old billet. This weld surface is initially flat but deforms as it passes through the die and forms a tongue (see also Figure 7.1(a)). The weld surfaces



(a) Billet weld surface *(b)* Hollow weld surface

Figure 7.1: Formation of weld surfaces.

are intrinsically weak parts of the profile and can be visible on the surface of the profile. Therefore, the part of the product that is expected to contain the tongue, is removed from the profile after extrusion. Since in the current extrusion practice the exact position and shape of the tongue is not known, the part that is removed from the profile is chosen considerably larger than the actual size of the tongue. If the shape and position of the tongue would be known, the part that is removed can be decreased, thus increasing the yield of each billet.

Weld surfaces are also formed in dies for hollow profiles when the aluminum separates to flow around the legs of the bridge part and welds together behind these legs in the weld chamber (see also Figure 7.1(b)). For strength and/or esthetic reasons it may be necessary to accurately predict the position of the weld surfaces in hollow profiles and even adapt the die design to change their position. The position, shape and size of the weld surfaces can be computed from the flow field of the aluminum through particle tracking.

Since the crystal size at the surface of the profile influences the appearance of the profile after anodizing or painting, a uniform crystal size distribution is desirable to prevent streaks on the profile. However, in reality it is difficult to predict a priori whether a profile will render uniform crystal size distributions at the surface, particularly near joints and/or jumps in the cross-sectional geometry of the profile. The crystal size distributions within an aluminum profile are determined by the initial crystal size distributions in the billet combined with the strain and temperature history that the aluminum experiences during extrusion and cooling. The velocity and temperature fields from the extrusion analyses can be used to determine the strain and temperature paths during extrusion, which can in turn be used to compute the crystal size distribution before cooling. For a given temperature path during cooling the final crystal size distribution can then be assessed.

From a safety and a commercial perspective it is crucial that a die is strong enough to withstand the enormous pressures that are exerted on it during extrusion. Two types of die failure are common in extrusion practice. The first occurs when the die is filled with aluminum at the start-up of extrusion. During start-up, the die is subjected to peak pressures which may cause the die to fail. The other type of failure is caused by fatigue resulting from the cyclic loading associated with the insertion of a new billet into the container. The die deflection computations render the maximum stresses that occur in the die during normal operation. These maximum stresses are indicative of the peak stresses and the cyclic stresses the die will experience during start-up and billet insertion, respectively, and can thus be used to predict failure of the die.

7.2.3 Parallelization

Ideally, the analyses presented in this work would be applied to automatically design a die through inverse engineering and optimization. Optimization methods start with a given die design and compute the flow field for that design. The quality of the flow field for a given die shape is evaluated through the use of an object function that is minimal for an optimal die design. In extrusion the object function would typically be related to the variation in the exit speed in a cross-section of the profile and the size of the velocity components perpendicular to the extrusion direction. Obviously, the optimization process requires the solution of the flow field for many variations of the die design. Since each permutation in principle implies a completely new extrusion analysis, the optimization approach is only feasible if the solution of one flow field requires a small amount of time. However, the actual analysis of a simple profile requires several hours of CPU time whereas that of a complex profile can consume several days of computer time. This makes it infeasible to pursue these optimization methods at this time.

A less ambitious solution is to compute the flow field as an integrated part of the die design process. While designing the die, the engineer can concurrently perform several analyses to assess the influences of the design choices that were made and change the design in an appropriate manner. To make this approach feasible, however, a considerable reduction of computational time is needed as well.

A reduction in the computational time can be achieved by using parallel solvers, rather than standard iterative solvers, for the solution of the linearized systems of equations. It is beyond the scope of this work to give a detailed description of parallel solvers. Therefore, only a brief overview is given here.

The principle of solving a system of equations in parallel is to distribute the work that needs to be done for the solution of the system over several processors, instead of one. If the work is distributed over n processors the solution time should theoretically be reduced by a factor n. The simplest way of parallelizing an iterative solver is to split the global matrix-vector products that have to be performed within one iteration into n local matrix-vector products that each render a local solution. The global solution is then assembled after the multiplication (Farhat and Roux, 1994). However, each assembly requires mutual communication between the processors which is a comparatively time consuming process and reduces the theoretically possible speed-up of a factor n to a real speed-up of much less than n. Therefore, more sophisticated parallelization algorithms have been developed which require less communication.

These methods are generally known as substructuring methods. The domain is divided into *n* non-overlapping substructures for which the system of equations is a subset of the global system of equations. These subsystems of equations can be solved separately on each domain and therefore in parallel, provided that the correct precautions are taken to ensure that the solution field is compatible across the subdomain boundaries. This is realized by prescribing displacements (Dirichlet preconditioning) or surface tractions (Neumann preconditioning) at the interfaces of the subdomain boundaries (Farhat and Chen, 1995). If these displacements or tractions were known a priori, the system of equations could be solved on each subdomain to render a local solution field. The global solution field could then be assembled using only one communication step. However, in practice the required boundary displacements or tractions are not known and have to be computed. The methods that can be used to compute these boundary surface displacements or tractions can be subdivided into primal- and dual-substructuring methods (Le Tallec, 1994).

The primal-substructuring method consists of formulating a Schur-complement problem that can be solved to render the surface displacements or tractions (Lassignardie et al., 1998). The solution inside the subdomains is then obtained by back-substitution. In the dual-substructuring method additional degrees of freedom λ are introduced in each node on the boundary of a subdomain. These degrees of freedom λ are associated with nodal loads on the boundary. The problem is then rewritten to render a formulation in terms of these degrees of freedom λ. Next, a preconditioned conjugate gradient (PCG) algorithm is applied to solve for λ. After λ is computed the solution field can be obtained for each subdomain. The most commonly used dual-substructuring method is the Finite Element Tearing and Interconnecting (FETI) method (Papadrakakis and Harbis, 1998). The FETI method seems to be the most promising method to speed up the solution of the extrusion analyses. It requires very little communication within each PCG iteration and the number of PCG iterations that is to obtain a converged solution, is, in theory, constant if the number of elements per subdomain is constant. This makes the number of iterations required, and thus the number of communication steps, independent of the problem size as long as a sufficient number of subdomains is defined (Farhat and Roux, 1994).

As a result of the low amount of communication between the processors that is required in the FETI algorithm, this algorithm is extremely well suited to be used on a distributed parallel computer that is constructed out of a heterogeneous network of computers with limited mutual communication speed. The network of computers that is in use in a standard design environment is a typical example of such a distributed parallel computer. Since these machines are only used during the day, for design purposes, they can be combined at night to form a distributed parallel computer. Furthermore, the FETI algorithm is more efficient than an iterative solver, even if it is invoked sequentially on a single processor computer (Farhat and Roux, 1994). This is due to the fact that the problem that defines the nodal loads λ has a better condition number than the original system of equations. Also, the systems of equations that is defined on each of the subdomains is better conditioned than the original system of equations. This makes the switch to a FETI-based solver worthwhile, even for use on a single processor computer.

It is expected that, due to the apparent drawbacks of physical trial-pressings, the numerical simulation of trial-pressings will become ever more important. Also, because the complexity of the profiles is constantly increasing, more simulations will be required to obtain a satisfactory die design. Moreover, the size of each simulation will become larger as well because the complexity of the domains involved increases. This inevitable growth in the number of simulations that have to be performed and in the size of these simulations will increase the importance of obtaining solutions faster, which is best obtained through parallelization.

Bibliography

S. Abtahi, T. Welo, and S. Støren. Interface mechanisms on the bearing surface in extrusion. In *Extrusion Technology Vol. II*, pages 125–131, 1996. 2.4.1, 2.4.1

L. Anand. A constitutive model for interface friction. *Comput. Mech.*, 12:197–213, 1993. 2.4.1

L. Anand and W. Tong. A constitutive model for friction. *Anals of the CIRP*, 42:361–366, 1993. 2.4.1

F.P.T. Baaijens. *On a Numerical Method to Solve Contact Problems*. PhD thesis, Eindhoven University of Technology, The Netherlands, 1987. 3.1.4

F.P.T. Baaijens, F.E. Veldpaus, and W.A.M. Brekelmans. On the numerical simulation of contact problems in forming processes. In Mattiasson, Samuelsson, Wood, and Zienkiewicz, editors, *Numerical Methods in Industrial Forming Processes*, pages 85–90, 1986. 3.1.4

K.J. Bathe. *The Finite Element Procedures in Engineering Analysis*. Prentice-Hall,inc., New Jersey, USA, 1982. 3.1

F. Bertrand, P.A. Tanguy, and F. Thibault. A three-dimensional fictitious domain method for incompressible fluid flow problems. *Int. J. Numer. Methods Fluids*, 25:719–736, 1997. 3.1.4, 3.1.4

T.D. Blacker and M.B. Stephenson. Paving : A new approach to automated quadrilateral mesh generation. *Int. J. Numer. Methods Engrg.*, 32:811–847, 1991. 4.1, 4.1.1

W.A.M. Brekelmans. *A Simulation Method for Die Compaction of Granular Materials*. PhD thesis, Eindhoven University of Technology, The Netherlands, 1989. 3.1.4, 3.1.4

F. Brezzi, L.P. Franca, T.J.R. Hughes, and A. Russo. Stabilization techniques and subrid scales capturing. *Proc. of the conf. on state of the art in num. anal.*, 1996. 3.1.4

F. Brezzi, L.P. Franca, T.J.R. Hughes, and A. Russo. $b = \int g$. *Comput. Methods Appl. Mech. Engrg.*, 145:329–339, 1997. 3.1.2

A.N. Brooks and T.J.R. Hughes. Streamline Upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32:199–259, 1982. 3.2

M.P. Clode and T. Sheppard. Formation of die lines during extrusion of AA6063. *Materials Science and Technology*, pages 755–763, 1990. 2.4.1

C. Cuvelier, A. Segal, and A.A. van Steenhoven. *Finite Element Methods and Navier-Stokes Equations*. Mathematics and Its Applications. D. Reidel Publishing Company, Dordrecht, 1986. 3.4.2

C. Farhat and P.S. Chen. A scalable Lagrange multiplier based domain decomposition method for time-dependent problems. *Int. J. Numer. Methods Engrg.*, 38:3831–3853, 1995. 7.2.3

C. Farhat, M. Lesoinne, and P. LeTallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Comput. Methods Appl. Mech. Engrg.*, 157:95–114, 1998. 2.7.1

C. Farhat and F.X. Roux. Implicit parallel processing in structural mechanics. *Comput. Mech. Adv.*, 2:1–124, 1994. 7.2.3, 7.2.3, 7.2.3

L. Fourment, K. Mocellin, and J.L. Chenot. An implicit contact algorithm for the 3D simulation of the forging process. In Owen, Oñate, and Hinton, editors, *Proceedings of the 5th International Conference on Computational Plasticity*, pages 873–877, 1997. 3.1.4

L.P. Franca and C. Farhat. Bubble functions prompt unusual stabilized finite element methods. *Comput. Methods Appl. Mech. Engrg.*, 123:299–308, 1995. 3.1.2

L.P. Franca, T.J.R. Hughes, and R. Stenberg. Stabilized finite element methods for the Stokes problem. In *Incompressible Computational Fluid Dynamics - Trends and Advances*, pages 87–108, 1993. 3.1

L.P. Franca and A. Russo. Unlocking with residual-free bubbles. *Comput. Methods Appl. Mech. Engrg.*, 142:361–364, 1997. 3.1.2

F. Garafolo. An empirical realtion defining the stress dependence of minimum creep rate in metals. *Met. Trans.*, 227, 1963. 2.2

J.F. Gobeau, T. Coupez, B. Vergnes, and J.F.A. Agassant. Computation of profile dies for thermoplastic polymers using anisotropic meshing. In *Simulation of Materials Processing: Theory, Methods and Applications*, pages 59–66, 1995. 4.1

L. Hanssen, M. Lefstad, S. Rystad, O. Reiso, and V. Johnsen. Billet surface flow in aluminum extrusion using 'half-moon' dies. In J.L. Chenot, J.F. Agassant, P. Montmitonnet, B. Vergnes, and N. Billon, editors, *Proceedings of the 1st ESAFORM conference on Material Forming*, pages 141–144, 1998. 1.2

A. Heege, P. Alart, and E. Oñate. Numerical modelling and simulation of frictional contact using a generalised Coulomb law. *Engineering Computations*, 12:641–656, 1995. 2.4.1

A. Heege and P.Alart. On an implicit contact-friction algorithm dedicated to 3D sheet forming simulation. In Chenot, Wood, and Zienkiewicz, editors, *Numerical Methods in Industrial Forming Processes*, pages 479–484, 1992. 3.1.4

T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall,inc., New Jersey, USA, 1987. 3.1

T.J.R. Hughes, G. Hauke, K. Jansen, and Z. Johan. Stabilized finite element methods in fluids: Inspirations, origins, status and recent developments. In *A book dedicated to Robert L. Taylor*, pages 272–292, 1994. 3.1

B. Joe. Delaunay versus max-min solid angle triangulations for for three-dimensional mesh generation. *Int. J. Numer. Methods Engrg.*, 31:987–997, 1991. 4.4.4

A.A. Johnson and T.E. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Comput. Methods Appl. Mech. Engrg.*, 119:73–94, 1994. 4.5

J. Kialka and W.Z. Misiolek. Studies of dead metal zone formation in aluminum extrusion. In *Extrusion Technology Vol. II*, pages 107–112, 1996. 2.1

L.A. Lalli and A.J. DeArdo. Experimental assessment of structure and property predictions during hot working. *Metallurgical transactions*, 21A:3101–3114, 1990. 2.2

J.F. Lassignardie, Y. Escaig, and G. Touzot. Direct domain decomposition method on a distributed memory architecture. In B.H.V. Topping, editor, *Advances in Computational Mechanics with High Performance Computing*, pages 65–72, 1998. 7.2.3

P. Le Tallec. Domain decomposition methods in computational mechanics. *Comput. Mech. Adv.*, 1:121–220, 1994. 7.2.3

V. Legat and J.M. Marchal. Die design: An implicit formulation for the inverse problem. *Int. J. Numer. Methods Fluids*, 16:29–42, 1993. 1.2

M. Lesoinne and C. Farhat. Improved staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. In S.R. Idelsohn, E. Oñate, and E.N. Dvorkin, editors, *Computational Mechanics, New Trends and Applications*, 1998. CD-ROM. 2.7.1

A. Lishnij, N. Biba, and A. Milenin. Three-dimensional simulation of extrusion process on PC. In J.L. Chenot, J.F. Agassant, P. Montmitonnet, B. Vergnes, and N. Billon, editors, *Proceedings of the 1st ESAFORM conference on Material Forming*, pages 137–140, 1998. 1.2

R. Löhner. Matching semi-structured and unstructered grids for Navier-Stokes calculations. In *AIAA 93-3348-CP*, pages 555–564, 1993. 4.3.2

A. Mizukami. An implementation of the Streamline Upwind/Petrov-Galerkin for linear triangular elements. *Comput. Methods Appl. Mech. Engrg.*, 49:357–364, 1985. 3.2.1

A. Moal and E. Massoni. Finite element simulation of the inertia welding of two similar parts. *Engrg. Comput.*, 12:497–512, 1995. 2.4.1

H.G. Mooi. *Finite Element Simulations of Aluminum Extrusion*. PhD thesis, University of Twente, The Netherlands, 1996. 1.2, 2.4.1, 2.6

M. Papadrakakis and D.C. Harbis. Domain decomposition solution techniques on workstation clusters. In B.H.V. Topping, editor, *Advances in Computational Mechanics with High Performance Computing*, pages 143–153, 1998. 7.2.3

R. Pierre. Optimal selection of the bubble function in the stabilization of the P1-P1 element for the Stokes problem. *SIAM J. Numer. Analysis*, 32(4):1210–1224, 1995. 3.1, 3.1.2, 3.1.2, 3.1.2, 3.1.2

A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*, volume 23 of *Springers series in Computational Mathematics*. Springer Verlag, 1994. 3.4.2

A. Ramage and A.J. Wathen. Iterative solution techniques for the Stokes and Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 19:67–83, 1994. 3.4.2

A. Russo. Bubble stabilization of finite element methtods for the linearized incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 132:335–343, 1996. 3.1

Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing company, Boston, USA, 1996. 3.4.2, 3.4.2, 3.4.2

V.M. Sample and L.A. Lalli. Effects of thermomechanical history on hardness of aluminum. *Mater. Sci. and Techn.*, 3:28–35, 1987. 2.2

A. Segal. *SEPRAN user manual*. Ingenieursburo SEPRA, Leidschendam, the Netherlands, 1983. 6.3

A. Segal. Finite element methods for advection-diffusion equations. In C.B. Vreugdenhil and B. Koren, editors, *Notes on Numerical Fluid Mechanics*, volume 45, pages 195–214. Vieweg and Sohn Verlagsgesellschaft, Wiesbaden, Germany, 1993. 3.2.1, 3.2.1

T. Shimizu and T. Sano. Development of a penalty method contact algorithm and its application to sheet forming. In Shen and Dawson, editors, *Numerical Methods in Industrial Forming Processes*, pages 489–494, 1995. 3.1.4

D. Silvester and J. Atanga. Iterative methods for stabilized mixed velocity-pressure finite elements. *International Journal for Numerical Methods in Fluids*, 14:71–81, 1992. 3.4.2

N. Solomon, M. Teodorescu, and I. Solomon. The effect of material flowing during deformatin on the extrusion product quality. In J.L. Chenot, J.F. Agassant, P. Montmitonnet, B. Vergnes, and N. Billon, editors, *Proceedings of the 1st ESAFORM conference on Material Forming*, pages 165–168, 1998. 1.2

S. Støren. Theory of extrusion-advances and challenges. *Int. J. Mech. Sci.*, 35:1007–1020, 1993. 1.2, 3

T.E. Tezduyar, M. Behr, S. Mittal, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces. *Comput. Methods Appl. Mech. Engrg.*, 94: 353–371, 1992. 4.5

H. Valberg, F.P. Coenen, and R. Kopp. Metal flow in two-hole extrusion. In *Extrusion Technology Vol. II*, pages 113–124, 1996. 2.1

B.J.E. van Rens, W.A.M. Brekelmans, and F.P.T. Baaijens. Modeling the elastoplastic behavior of perforated plates. In D.R.J. Owen, E. Oñate, and E. Hinton, editors, *Proceedings of the 5th International Conference on Computational Plasticity*, pages 1998–2004, 1997.

B.J.E. van Rens, W.A.M. Brekelmans, and F.P.T. Baaijens. Influence of boundary conditions on the results of 3D numerical simulation of aluminum extrusion. In J.A. Covas, editor, *Proceedings of the 2nd ESAFORM conference on Material Forming*, pages 69–72, 1998a. 2.7.1

B.J.E. van Rens, W.A.M. Brekelmans, and F.P.T. Baaijens. Modeling friction near sharp edges when using an Eulerian approach. In S.R. Idelsohn, E. Oñate, and E.N. Dvorkin, editors, *Proceedings of the 4th World Conference on Computational Mechanics, New Trends and Applications*, 1998b. CD-ROM. 3.1.4, 3.1.4

B.J.E. van Rens, W.A.M. Brekelmans, and F.P.T. Baaijens. A semi-structured mesh generator applied to extrusion. In J. Huetink and F.P.T. Baaijens, editors, *Simulation of Materials Processing: Theory, Methods and Applications*, pages 621–626, 1998c. 4.1, 4.3.2

B.J.E. van Rens, W.A.M. Brekelmans, and F.P.T. Baaijens. Steady, three dimensional flow calculation of aluminum extrusion with complicated die geometries. In J.L. Chenot, J.F. Agassant, P. Montmitonnet, B. Vergnes, and N. Billon, editors, *Proceedings of the 1st ESAFORM conference on Material Forming*, pages 495–498, 1998d. 4.5

B.J.E. van Rens, W.A.M. Brekelmans, and F.P.T. Baaijens. Three dimensional finite element analysis of aluminum extrusion. In B.H.V. Topping, editor, *Advances in Computational Mechanics with High Performance Computing*, pages 25–32, 1998e. 4.5

B.J.E. van Rens, W.A.M. Brekelmans, and F.P.T. Baaijens. Boundary conditions for the 3D numerical simulation of extrusion. In *Proceedings of the European Conference on Computational Mechanics*, 1999a. In press. 5.1

B.J.E. van Rens, W.A.M. Brekelmans, and F.P.T. Baaijens. A priori prediction of the performance of aluminum extrusion dies through 3D numerical simulation. In *Proceedings of the 6th International Conference on Technology of Plasticity*, 1999b. In press.

B.J.E. van Rens, W.A.M. Brekelmans, and F.P.T. Baaijens. Shape prediction for complex polymer extrudates through 3D numerical simulation. In F.N. v.d. Vosse and P.G.M. Kruijt, editors, *Proceedings of the 15th PPS conference*, 1999c. CD-ROM. 2.7.2, 6.3

B.J.E. van Rens, W.A.M. Brekelmans, J.J. Rusch, and F.P.T. Baaijens. Homogenization of the elastoplastic behavior of perforated plates. *Comp. and Struct.*, 69:537–545, 1998f.

B.J.E. van Rens, D. Brokken, W.A.M. Brekelmans, and F.P.T. Baaijens. A two-dimensional paving mesh generator for triangles with controllable aspect ratio and quadrilaterals with high quality. *Engrg. with Computers*, 14:248–259, 1998g. 4.1

A.K. v.d. Vegt. *Polymeren van keten tot kunstof*. Delftse Uitgevers Maatschappij, Delft, The Netherlands, 1994. 6.3

T. Welo, S. Abtahi, and I. Skauvik. An experimental and numerical investigation of the thermo-mechanical conditions on the bearing surfce of extrusion dies. In *Extrusion Technology Vol. II*, pages 101–106, 1996. 2.4.1, 2.4.1, 2.4.1

S.O. Wille. A preconditioned alternating inner-outer iterative solution method for the mixed finite element formulation of the navier-stokes equations. *International Journal for Numerical Methods in Fluids*, 18:1135–1151, 1994. 3.4.2

R.N. Wright, G.G. Lea, and F.F. Kraft. Constitutive equations and flow stress characterization concepts for aluminum extrusion. In *Extrusion Technology Vol. I*, pages 259–268, 1996. 2.2

D.Y. Yang. Investigaton into non-steady-state three-dimensional extrusion of a trocoidal helical gear by the rigid-plastic finite element method. *Anals of the CIRP*, 43:229–233, 1994. 2.4.1

Y. Zheng, R.W. Lewis, and D.T. Gethin. Three-dimensional unstructured mesh generation : Part 1. Fundamental aspects of triangulation and points creation. *Comput. Methods Appl. Mech. Engrg.*, 134:249–268, 1996. 4.1, 4.4.4

W.F. Zoetelief. *Multi-Component Injection Moulding*. PhD thesis, Eindhoven University of Technology, The Netherlands, 1995. 6.3

Samenvatting

Aluminiumextrusie is een productieproces waarin een blok heet aluminium in een recipient wordt geplaatst en vervolgens door een matrijs wordt geperst om een profiel te verkrijgen. De matrijs speelt een cruciale rol in dit proces omdat deze zorgt voor de balancering van de aluminium stroom. Een gebalanceerde aluminium stroom is nodig opdat het resulterende profiel aan de eisen van de klant voldoet wat betreft, bijvoorbeeld, de vorm van de dwarsdoorsnede en de rechtheid. In de praktijk wordt voor het ontwerp van de matrijs een trial-and-error strategie gevolgd waarin proefpersingen een belangrijke rol spelen. Deze proefpersingen, waarmee de toepasbaarheid van een gegeven matrijs ontwerp wordt vastgesteld, zijn tijdrovend en duur. Daarnaast is het moeilijk om afwijkingen in het geëxtrudeerde profiel te vertalen naar aanpassingen in de matrijs en er is veel ontwerpervaring vereist om deze vertaalslag te kunnen maken. Het doel van dit proefschrift is het ontwikkelen van een eindige elementen model dat de proefpersingen, in ieder geval ten dele, kan vervangen door numerieke simulaties.

Het extrusieproces van aluminium is gemodelleerd als een Stokes stromingsprobleem dat is gekoppeld aan een thermisch convectie-diffusie probleem. Het stromingsprobleem is geformuleerd met een Euler beschrijvingswijze, waarbij de ruimtelijk gefixeerde discretisatie geschikt is om processen te modelleren waarin grote deformaties plaatsvinden, zoals extrusie. Deze beschrijvingswijze vraagt echter wel extra aandacht als het ruimtelijke domein van het materiaal a priori (gedeeltelijk) onbekend is. Deze complicatie doet zich voor na de uitstroomopening van de matrijs omdat daar het aluminium niet meer omsloten is door het gereedschap en dus kan vervormen als gevolg van een onbalans in de stroming. Ook op het scheidingsvlak tussen matrijs en aluminium kan dit probleem zich manifesteren wanneer de matrijs significant deformeert door de extrusiedruk. Het reconstrueren van de vorm van het aluminium domein wordt bereikt met behulp van de zogenaamde drie-velden benadering. In deze benadering worden de vorm van het profiel nadat het de matrijs heeft verlaten, en de vorm van de matrijs, afzonderlijk berekend. Het effect van de geometrische veranderingen van het aluminium domein wordt vervolgens verdisconteerd door de (aluminium) mesh aan te passen.

Het consitutieve gedrag van het aluminium is gemodelleerd als een niet-Newtonse vloeistof waarvan de viscositeit afhangt van de temperatuur en de effectieve afschuifsnelheid. Daarnaast is voor de interactie tussen het aluminium en het gereedschap Coulombse wrijving verondersteld.

Het stromingsprobleem is gediscretiseerd door gebruik te maken van het MINI element. Dit element maakt het mogelijk om krachtige tetraheder meshgeneratoren toe te passen, terwijl er per element een minimaal aantal graden van vrijheid wordt gegenereerd. Tevens zijn de Lagrange multiplicatoren, die het effect van de wrijving op de aluminium stroming verdisconteren, gediscretiseerd volgens een speciaal daartoe ontwikkelde methode die het mogelijk maakt om nauwkeurige resultaten te verkrijgen op relatief grove meshes. Tenslotte, is het thermische probleem gediscretiseerd met behulp van lineaire tetraheder elementen die gestabiliseerd zijn op basis van de SUPG (Streamline Upwinding / Petrov-Galerkin) methode.

De ruimtelijke domeinen die behoren bij het aluminium en de matrijs zijn complex. Daardoor poneert het genereren van een eindige elementen verdeling een ernstig probleem. Dit probleem is opgelost door nieuwe meshgeneratie algoritmes te ontwikkelen. Deze algoritmes creëren meshes met richtingsafhankelijke verfijning, zodanig dat de gradienten in de oplossingsvelden kunnen worden berekend met gebruik van een beperkt aantal elementen. Tevens bevatten deze algoritmes een aantal nabewerkingsstappen om de variërende lengte van de matrijscylinders en de complexe vormen van holle matrijzen in de discretisatie op te nemen. De nieuwe methode van meshgeneratie maakt het mogelijk om, met een minimale inspanning van de gebruiker, meshes te genereren voor elk realistisch matrijsontwerp.

Gebaseerd op de vergelijking van verschillende numerieke experimenten kan worden geconcludeerd dat het belangrijk is om de temperatuurafhankelijkheid van de viscositeit, de vorm van het uitstromende profiel, en soms de deformatie van de matrijs te verwerken in de berekeningen. Ook kan worden geobserveerd dat de waarde die wordt toegekend aan de (Coulombse) wrijvingscoëfficiënt een aanzienlijke invloed heeft op zowel de snelheid- als de temperatuurvelden van het aluminium. Vergelijking met experimentele gegevens uit de industrie laten zien dat het model de stromingsvelden en de vorm van het resulterende profiel goed voorspelt, zelfs voor een complexe profielgeometrie.

Concluderend kan worden gesteld dat het model dat is beschreven in dit proefschrift in staat is om de merites van complexe matrijzen op accurate wijze te voorspellen, in een tijd die aanzienlijk korter is dan de tijd die noodzakelijk is om een proefpersing uit te voeren.

Dankwoord

Dit proefschrift is mede tot stand gekomen dankzij de bijdragen van een groot aantal mensen, die ik bij deze dan ook wil bedanken. Enkelen van hen wil ik met name bedanken.

Mijn grote dank gaat uit naar Marcel Brekelmans en Frank Baaijens wiens enthousiasme, hulpvaardigheid en belangstelling ik enorm heb gewaardeerd. De vele inhoudelijke discussies met Marcel en diens consciëntieuze werkwijze zijn van grote waarde geweest voor dit onderzoek, niet in de laatste plaats bij het schrijven van dit proefschrift. De vrijheid die Frank me heeft gegeven om de nodige zijpaden te bewandelen, alsook de gedachtenwisselingen over problemen van numerieke aard, zijn zeer stimulerend geweest. Dank ook aan Bas van Hal voor zijn bijdrage aan dit werk in de vorm zijn afstuderen. Ron Peerlings dank ik voor het lezen en corrigeren van één van de laatste versies van het proefschrift.

Met plezier heb ik samengewerkt met de mensen van Alcoa Europe, location Drunen. Speciaal wil ik Henk Zwikker en Johan Zentjens bedanken voor het gestelde vertrouwen, en Peter den Dikken voor de belangrijke rol die hij heeft gespeeld bij het opzetten en uitvoeren van de experimenten.

Ook wil ik de collega's van de Materials Technology groep bedanken voor de prettige werksfeer, in het bijzonder de oud-kamergenoten Dirk Brokken, Robert Smit en Marcel Meuwissen en mede-MAIO Michael van der AA.

Tenslotte wil ik mijn ouders en broer danken voor hun niet aflatende steun, begrip en geduld, niet alleen gedurende deze promotie. Als allerlaatste en allerbelangrijkste gaat mijn grote bewondering en dankbaarheid uit naar Anne-Marie; haar steun en vertrouwen zijn cruciaal geweest gedurende deze promotie. Anne, je hebt de komst van onze kleine helemaal alleen moeten voorbereiden, maar van nu af aan staan we er weer samen voor.

Bas van Rens
Eindhoven, 19 september 1999

Curriculum vitae

6 oktober 1971	Geboren te Eindhoven
sep. 1983 - jun. 1989	Ongedeeld Gymnasium, Stedelijk Gymnasium Nijmegen
sep. 1989 - jun. 1990	School of Engineering, University of Kansas, Lawrence, KS, Verenigde Staten
sep. 1990 - dec. 1995	Werktuigbouwkundige Medische Technologie, Faculteit Werktuigbouwkunde, Technische Universiteit Eindhoven
jan. 1996 - nov. 1999	Assistent in opleiding, sectie Materials Technology, Faculteit Werktuigbouwkunde, Technische Universiteit Eindhoven