F.E.M simulation of metal forming processes: development of the rigid plastic simulation program riple
ten Have, D.E.

Published: 01/01/1991

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 author's 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

Citation for published version (APA):
F.E.M. SIMULATION OF METAL FORMING PROCESSES.

Development of the Rigid Plastic Simulation Program Riple

Guestresearch at the Mechanical Engineering Laboratory, Tsukuba, Japan.

Dino E. ten Have November 1991
W.P.A. nr.: 1258
SUMMARY

For 3½ months, I have been doing guestresearch at the Mechanical Engineering Laboratory (M.E.L.) in Tsukuba science city together with a mastercourse student from the Utsunomiya University of Technology.

In the research described in this report, some improvements in a computer simulation program (RIPE) developed at M.E.L. have been investigated. The research has been divided in 3 parts; we looked at the proper use of friction, the implementation of die and punch roundings in the simulation program and we did some experiments to verify the results of RIPE.

In the first part, we changed the way of using the frictional method of Coulomb and we did some simulations of the compression of a cylinder, with Coulomb and Von Mises friction, which we also calculated analytically.

In the second part, we found a method to implement die and punch roundings in the program. This method describes the roundings of the punch and the die mathematically. This method works well, however some contact problems have been encountered.

In the last part, experiments have been carried out to compare the results with the output of RIPE. Here, we saw that the shape of the simulated process resembles the shape of the experiment very well and that predictions can be made about the behaviour of the process, like when and where necking will occur and if fracture will occur.
# CONTENTS

## SUMMARY ........................................................................................................ 1

## CONTENTS ......................................................................................................... 2

## PREFACE ........................................................................................................... 4

1. **GENERAL INFORMATION ABOUT THE RIPLE PROGRAM AND THE SYSTEM BEING USED** ......................................................... 5
   1.1. The RIPLE program ............................................................................. 5
   1.2. The systems being used ................................................................. 5

2. **HOW RIPLE WORKS** .................................................................................... 7
   2.1. The structure of RIPLE ..................................................................... 7
   2.2. Parameter and model definition card .............................................. 8
   2.3. The calculation of the velocity field ............................................... 8

3. **FRICTIONAL DISCOUNT IN RIPLE** ................................................................. 10
   3.1. Introduction ..................................................................................... 10
   3.2. The setup of the model ................................................................. 10
   3.3. Calculating the internal power ..................................................... 11
   3.4. The use of friction ......................................................................... 11
      3.4.1. Changing the value for $\sigma$ used for frictional stress in the Coulomb method ................................................................. 12
   3.5. Compression without friction ....................................................... 14
      3.5.1. A RIPLE simulation without friction ....................................... 14
      3.5.2. An analytical approach without friction ................................... 15
   3.6. Compression with friction ............................................................. 17
      3.6.1. A RIPLE simulation with friction ............................................ 17
      3.6.2. An analytical approach with friction ....................................... 18
   3.7. Conclusions about the results of the frictional discount in RIPLE .. 21

4. **DISCOUNTING DIE AND PUNCH ROUNDINGS FOR DEEPDRAWING IN RIPLE** ................................................................. 23
   4.1. Introduction ..................................................................................... 23
   4.2. Choosing the method for describing roundings ......................... 23
   4.3. Bézier curves .................................................................................. 23
   4.4. The implementation of the roundings .......................................... 24

D.E. ten Have, Tsukuba, 11-1991
PREFACE

Japan has always interested me for its high technology and its unique culture. Therefore, I have been looking for possibilities to spend some time there for my study. My graduate professor, professor Kals has a good relationship with the Mechanical Engineering Laboratory (M.E.L.) in Tsukuba science city, Japan, and he was willing to inform if it was possible for me to go there. Dr. T. Sano, director of the plasticity and forming division, agreed on this and invited me to come to Tsukuba. At that moment, my guestresearch was fixed for 3 months.

This period has been a very instructive one. My theme was "computer simulation of metal forming processes". It consisted of the further development of a F.E.M. simulation program which has been developed in M.E.L. in 1985. More specifically, this meant to look at the proper use of friction and the possibility to use die and punch roundings in the simulation. In this research, I worked together with a Japanese mastercourse student who is graduating at M.E.L. I can say that I learned a lot more about the fundamental theories of metal forming processes, the use of other computers than P.C.'s, programming in FORTRAN and doing experiments. I also experienced things that I would never have experienced if I had stayed in the confident surroundings of my university, like the Japanese way of living, working and thinking.

For the opportunity I have been given and for the great support I have had, I want to thank Dr. T. Sano, my host; Dr. H. Sato, director general of M.E.L.; Dr. K. Matsuno, deputy director general of M.E.L.; T. Shimizu, working on F.E.M. simulation at M.E.L.; Dr. K. Ohuchi, my conductor; E. Sato, the student I worked together with very pleasantly for 3 months and Professor J. Kals.

Tsukuba, November 13, 1991
1. GENERAL INFORMATION ABOUT THE RIPLE PROGRAM AND THE SYSTEM BEING USED

1.1. THE RIPLE PROGRAM

RIPLE stands for RIgid PLastic deformation analysis codE. It is a F.E.M. simulation program developed by Dr. T. Sano and later further developed by T. Shimizu. The program is written in the computer language Fortran.77 and it can be used for all kinds of metal forming processes. For example, it can be applied for bulging, forging and deepdrawing. Because it uses the rigid plastic formula to describe the material flow, the program is especially useful for the analysis of processes where large plastic deformations are involved and where springback does not occur very much. Suitable processes are those that use massive or thickplated steel.

The way that RIPLE uses the finite element method is using a variational principle for rigid plastic deformation. This is done by finding a stationary situation for the energy equation, consisting of the equilibrium of the internal and the external power of a rigid plastic body, deforming totally plastic.

The formula for the rigid plastic behaviour of the material is the following:

\[ \sigma = Y_0 + K \cdot \dot{\varepsilon}^m \cdot \varepsilon^n \]  \hspace{1cm} (1.1)

RIPLE is developed in a 2 dimensional and in a 3 dimensional version. This far, the 2 dimensional version has the possibility to use friction and because the further development of the proper use of friction it can be implemented in the 3 dimensional version.

To visualise the output of simulations done by RIPLE, a program has been written in BASIC. Therefore, the desired output of the simulation is written in a datafile, that can be used as an inputfile for the BASIC program. There are plans to write a program for postprocessing in the computer language "C" because of it's stronger graphic capabilities.

1.2. THE SYSTEMS BEING USED

In Tsukuba, there is a computer centre called the RIPS centre (Research Information Processing System centre). To support research activities, the AIST research institutes in Tsukuba can use the possibilities of the centre freely.
In the RIPS centre, different (super)computersystems are available, of which we mainly used
the cray X-MP/216 supercomputer. From M.E.L., we worked on a SUN spark workstation and
several p.c.'s, connected with the CRAY computer by means of a network called ethernet. The
operating system being applied is UNIX SYSTEM V. In figure 1.1, the relationships are made
clear.

![Diagram](image)

**figure 1.1: Relationships between the computer systems**
2. HOW RIPLE WORKS

2.1. THE STRUCTURE OF RIPLE

To make it clear how the RIPLE program is built, the following flowchart is given, after which explanation of each block will be given.

![Flowchart of RIPLE's structure](image)

**figure 2.1: The structure of Riple**
2.2 Parameter and Model Definition Card

The input for the program consists of a datafile which contains information about the material and the kind of simulation. In this inputfile for the program, two different cards have to be written, the parameter cards and the model definition cards.

In the parameter cards there is information about which options of the program are being used, like the type of simulation. In the two dimensional program there is the possibility to use an axi-symmetric, a plain stress and a plain strain model.

Furthermore it is necessary to give information about the number of stages or the material deformation ratio which is desired, whether friction is being used or not, which information has to be written to the outputfile and how the initial velocity field for the first stage is defined.

To take the condition of incompressibility into account it is possible to use four different methods: using the incompressibility equation of Lagrange, the penalty method and two other methods developed in Japan. For the simulations in this research, the Lagrange method is being used.

In the model definition cards it is necessary to define the model of the workpiece, the proper boundary conditions for the type of process that is being simulated and the distribution of the load on the workpiece or the toolvelocity. The model definition cards also have to contain information about the material property and the frictional factor if friction is being used.

2.3. The Calculation of the Velocity Field

For the rigid plastic deformation problems, the method of minimising the energy equation is being used. The energy equation is written in formula 2.1.

\[ \Phi = \int_{\Omega} \sigma' : \varepsilon \, dV + \int_{\Gamma} \beta \cdot C^T : \varepsilon \, dV - \int_{\Gamma} T^T U \cdot dS \]  

(2.1)

In this formula, the first part on the right side of the equation mark describes the energy dissipated in the material. Here, \( \sigma' \) is the deviatorical stress vector which is derivable from the strain rate vector. The strain rate vector itself is derivable from the velocity field \( U \).
The second part is to take incompressibility into account using the Lagrange method. \( B \) is the mean stress and \( C^r \) is a proper notation of the Kronecker delta in a way that \( B \) multiplied with the strain rate vector gives the incompressibility condition.

The last part of formula 2.1 describes the traction on the surface of the workpiece. \( T \) is the traction vector, specified on the boundary \( S \).

If this energy equation is related to a 4-node element and the requirement of a stationary situation (\( \delta \Phi = 0 \)) is being used for 2 dimensions, we obtain eight equations with nine unknown variables; 8 values for the nodal point velocities and 1 value for the mean stress \( B \), which is assumed uniform over the element. To get a 9th equation, an explicit incompressibility condition is defined.

If the proper notations for the strain rate vector and the deviatorical stress vector are substituted and the requirement of a stationary situation of the functional \( \Phi \) is taken into account, it is possible to write equation 2.1 in nonlinear stiffness relations for every element. The equations for every element can be put together in one matrix equation for all elements of the rigid body. The parts of these equations are calculated in the program as 'stiffness matrix' and 'load vector'. Because it is not the intention of this report to explain this completely and because many papers have been written about it, more explanation can be found in reference 1.

When using small perturbations of the velocity vector for every iteration step, it is possible to solve the matrix equations by means of the Newton Raphson method until the velocity field has reached a satisfactory value (the value \( \Delta v/v \) is smaller than a desired criterium). To make the program converge faster, a factor \( \alpha \) is introduced to multiply with the perturbation \( \Delta v \). This value can be varied during the iteration loop of each stage. The values for \( \alpha \) for which the program converges the fastest depends on the type of simulation and has to be found empirically.

When the velocity field is satisfactory, it is being updated and the coordinate field is also being updated in "preparing for the next stage". After this loop, the next stage can be started.
3. FRICTIONAL DISCOUNT IN RIPLE

3.1. INTRODUCTION

In 1985, the possibility to use friction was built in the 2-D version of RIPLE. The two classical laws of Von Mises and Coulomb were being used in a slightly different way. After that, that option was not being applied so much and there was not much known about how it worked so it was useful to know more about the results. Because of that reason we decided that some simulations with friction of a simple compression model were carried out which could be compared by an analytical calculation.

3.2. THE SETUP OF THE MODEL

For the compression model, a cylinder has been used with a radius of 100 mm. and a height of 200 mm. To reduce the C.P.U. time only a quarter of the model has been simulated. For the "cut off area", the proper boundary conditions were taken into account. A model with 100 elements and 121 nodal points is shown in figure 3.1.

![Figure 3.1: The compression model](image)

The compression has been done with a tool velocity of 4 mm s⁻¹.

The material being simulated was SS41 steel with the following rigid plastic properties (ref. 2):

\[
\begin{align*}
Y_0 &= 200 \text{ [N mm}^2\text{]} \\
K &= 540 \text{ [N mm}^2\text{]} \\
m &= 0.01 \text{ [-]} \\
n &= 0.155 \text{ [-]}
\end{align*}
\]
3.3. CALCULATING THE INTERNAL POWER

To check the solutions of the simulation of the compression model with an analytical model it is very useful to calculate the internal power. This power can also be used to check the results of a simulation with the power output of an experiment.

Also because of the importance of the ability to know the capacity of a machine that has to be used for a certain type of process or to see if the capacity of the intended machine is satisfactory by doing a simulation we decided to build the possibility of calculating the internal power into RIPLE.

The power is derived by using formula 3.1, which is transformed from an integration to a summation to use it in a numerical way.

\[
P = \int \frac{\bar{\sigma} \cdot \dot{\varepsilon}}{N} dV + \int \tau_f \cdot v_i dS = V \cdot \sum_{n=1}^{N} \frac{\bar{\sigma} \cdot \dot{\varepsilon}}{N} + A \cdot \sum_{m=1}^{M} \frac{\tau_{f(\theta)}}{M} |v_{i(\theta)}|
\]  

(3.1)

where \(v_i\) = averaged tangential velocity between two nodal points from the surface where friction occurs

\(N\) = total number of elements

\(M\) = total number of elements on which friction occurs

3.4. THE USE OF FRICTION

RIPLE can use two types of friction, the Coulomb model and the Von Mises model, respectively in formulas 3.2 and 3.3.

\[
\tau_f = \mu \cdot \sigma_n
\]  

(3.2)

\[
\tau_f = \frac{m \cdot \sigma_f}{\sqrt{3}}
\]  

(3.3)

instead of the normal stress in formula 3.2, however, the mean stress was being used. Because there is a difference wether the mean stress or the normal stress is being used, we decided to change the program for the proper use of the Coulomb method. For the flowstress in formula 3.3, RIPLE uses the effective stress, which in fact is the momentary flowstress.
3.4.1. Changing the value for \( \sigma \) used for frictional stress in the Coulomb method

To derive the normal stress we used the deviatorical stress vector \( \sigma' \), which is defined as:

\[
\sigma' = \frac{\sigma}{\dot{\epsilon}} D \dot{\epsilon}
\]  \hspace{1cm} (3.4)

\( D \) is the flow matrix for isotropic material.

The strain rate is obtained in the following way:

\[
\dot{\epsilon} = \begin{pmatrix}
\dot{\epsilon}_r \\
\dot{\epsilon}_\theta \\
\dot{\epsilon}_z \\
\dot{\epsilon}_\phi \\
\end{pmatrix} = \begin{pmatrix}
\frac{\partial}{\partial r} & 0 \\
0 & \frac{\partial}{\partial z} \\
\frac{1}{r} & 0 \\
\frac{\partial}{\partial z} & \frac{\partial}{\partial r}
\end{pmatrix} \begin{pmatrix}
u \\
n
\end{pmatrix} \hspace{1cm} (3.5)
\]

In RIPLE the velocity-strain rate transformation matrix in equation 3.5 is changed from a continuous matrix to a discrete matrix and the coordinate system is transformed to local coordinates after which the strain rate is calculated. The exact way how this is done can be found in reference 1.

For the axi symmetric problem we can write:

\[
\sigma' = \begin{pmatrix}
\sigma'_r \\
\sigma'_\theta \\
\sigma'_z \\
\sigma'_\phi \\
\end{pmatrix} \hspace{1cm} (3.6)
\]

To compose the deviatorical normal stress \( \sigma'_n \), we need the value for \( \sigma'_z \) and \( \sigma'_r \). The deviatorical stress and the global stress have the following relationship (def):

\[
\sigma' = \frac{\sigma}{\dot{\epsilon}} \cdot \frac{2}{3} \cdot \dot{\epsilon} = \sigma - \sigma_m
\]  \hspace{1cm} (3.7)
The sign of the deviatorical stress has to be changed because the stress in RIPLE is positive for compression in opposition to the definition. Therefore, according to the sign of the strain rate which is working the other way as RIPLE’s stress, we get:

\[ \sigma' = \frac{\sigma}{\varepsilon} \cdot \frac{2}{3} \cdot \dot{\varepsilon} \]

Now we can write:

\[ \sigma_z = \sigma_m - \frac{\sigma}{\varepsilon} \cdot \frac{2}{3} \cdot \dot{\varepsilon} \]

\[ \sigma_r = \sigma_m - \frac{\sigma}{\varepsilon} \cdot \frac{2}{3} \cdot \dot{\varepsilon} \]

We can use these equations to calculate the normal stress and use this for the frictional shear stress. For the compression model this will only result in the use of \( \sigma' \), because the angle of contact on the surface is 0° and will stay the same.

In some processes it occurs that the normal stress becomes extremely high. Following Coulombs law for frictional stress, the frictional shear stress would also become very high. It has to be regarded however, that the frictional shear stress can never be higher than the maximum shear stress which is equal to \( \sigma_{\text{en}} / \sqrt{3} \). Therefore we can say that:

\[ \tau_{fr} = \mu \cdot \sigma_n \leq \frac{\sigma}{\sqrt{3}} \]

In RIPLE, we built in the condition that if the shear stress exceeds the maximum shear stress because the normal stress is too high, the shear stress will be equal to the maximum shear stress \( \sigma_{\text{en}} / \sqrt{3} \).
3.5. COMPRESSION WITHOUT FRICTION

3.5.1. A RIPLE simulation without friction

The non-frictional models have all been tested for a period of 1 and 3 seconds.

As expected, the distribution of the stress and strain appeared to be uniform. After a period of 1 second, the mean stress in every element is $1.711 \times 10^8$ N·m$^{-2}$, the effective strain is $4.074 \times 10^{-2}$ and the effective strain rate is $4.149 \times 10^{-2}$ s$^{-1}$. The specific power for a total of 100 elements is $0.2130 \times 10^{10}$ N·s$^{-1}$·m$^{-2}$. Divided by 100 and multiplied by the volume of the cylinder gives a total power of 133.8 kW.

A plot of a compressed model can be seen in figure 3.2.

After a period of 3 seconds the mean stress in all 100 elements is $1.928 \times 10^8$ N·m$^{-2}$, the effective strain is $1.276 \times 10^{-1}$ and the element strain rate is $4.525 \times 10^{-2}$ s$^{-1}$. The specific power for a total of 100 elements is $0.2617 \times 10^{10}$ N·s$^{-1}$·m$^{-2}$. Divided by 100 and multiplied by the volume of the cylinder gives a total power of 164.4 kW.

The calculated values are given in the following table.

<table>
<thead>
<tr>
<th>Compression time</th>
<th>Mean stress [N·m$^{-2}$]</th>
<th>Strain [-]</th>
<th>Strain rate [s$^{-1}$]</th>
<th>Power [kW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 second</td>
<td>$1.711 \times 10^8$</td>
<td>$4.074 \times 10^{-2}$</td>
<td>$4.149 \times 10^{-2}$</td>
<td>133.8</td>
</tr>
<tr>
<td>3 seconds</td>
<td>$1.928 \times 10^8$</td>
<td>$1.276 \times 10^{-1}$</td>
<td>$4.525 \times 10^{-2}$</td>
<td>164.4</td>
</tr>
</tbody>
</table>

Table 3.1: Results of a simulation without friction

The effect of work hardening is clearly to be seen. The element stress increases as the compression is carried out for a longer period. The effect is also to be seen in the power that
is needed for the compression. For non-hardening, ideal plastic material, the power would theoretically remain the same but for this material, the power that is needed increases as compression is carried out for a longer period.

3.5.2. An analytical approach without friction

The definition for the cylinder in the analytical approach can be seen in figure 3.3.

Because the shape of the cylinder has to be remained, the velocity in r-direction is independent of the z-coordinate. Global volume invariance (volume before compression = volume after compression) gives:

$$\pi r^2 u_z = 2 \pi r h u_r$$  \hspace{1cm} (3.12)

Figure 3.3: The definition of the cylinder

with:  
$u_z$ = velocity in x-direction  
$r = r$-coordinate  
$z = z$-coordinate  
$u = die velocity$  
$h = half of the momentary total height of the workpiece$

from relationship 3.12 follows:

$$u_r=r\frac{u}{2h}$$  \hspace{1cm} (3.13)

Here, $r$ and $h$ are taken in the momentary situation.

Local volume invariance for the axi-symmetric model is given by:

$$\frac{\delta u_z}{u_z} = \frac{\delta (ru_r)}{ru_r} \cdot \frac{1}{r}$$  \hspace{1cm} (3.14)

After integration and with the boundary condition $u_z=0$ for $z=0$ this gives the velocity in z-direction:

$$u_z = -z \frac{u}{h}$$  \hspace{1cm} (3.15)

With the velocities $u_r$ and $u_z$, the strain rates can be calculated from:

$$\dot{\varepsilon}_r = \frac{\delta u_r}{\delta r}, \quad \dot{\varepsilon}_\theta = \frac{u_r}{r}, \quad \dot{\varepsilon}_z = \frac{\delta u_z}{\delta z}, \quad \dot{\varepsilon}_r = \frac{1}{2} \left( \frac{\delta u_z}{\delta r} + \frac{\delta u_r}{\delta z} \right)$$  \hspace{1cm} (3.16)
which gives:
\[ \dot{\varepsilon}_n = \frac{u}{2h}, \quad \dot{\varepsilon}_\phi = \frac{u}{2h}, \quad \dot{\varepsilon}_z = -\frac{u}{h}, \quad \dot{\varepsilon}_\nu = 0 \quad (3.17) \]

and with the formula for the effective strain rate:
\[ \dot{\varepsilon} = \sqrt{\frac{2}{3} \left( \dot{\varepsilon}_y \cdot \dot{\varepsilon}_y \right)} \quad (3.18) \]

we can calculate for the effective strain rate:
\[ \dot{\varepsilon} = \sqrt{\frac{2}{3} \left( \frac{u^2}{4h^2} + \frac{u^2}{h^2} + \frac{u^2}{4h^2} \right)} = \frac{u}{h} \quad (3.19) \]

This means that the effective strain rate for our model (with \( u \) as a constant velocity and with \( h \) calculated with the momentary height) should be \( 0.004/0.096 = 0.042 \text{ s}^{-1} \) for compression of 1 second and \( 0.004/0.088 = 0.045 \text{ s}^{-1} \) for compression of 3 seconds. This is very well in agreement with the results that are derived from the RIPLE simulation. For the effective strain we can deduce equation 3.20.

\[ \bar{\varepsilon} = \int \varepsilon \delta t = \int \frac{u}{h} \delta t = \int \frac{\dot{h}}{h} \delta t = \int \frac{\delta h}{h_0} = \ln \frac{h_0}{h} \quad (3.20) \]

For a period of 1 second, this means that the effective strain would be \( \ln(0.2/(0.2-0.004)) = \ln(0.2/0.192) = 0.041 \) and for a period of 3 seconds the effective strain would be \( \ln(0.2/(0.2-3\cdot2-0.004)) = \ln(0.2/0.176) = 0.128 \). This also matches with the results of RIPLE.

For the calculation of the total power we use the so-called upper bound method. In this case, without friction we only have the factor of the dissipated power, \( P_d \). With the following formula and the formula for the material behaviour we can calculate this power.

\[ P_d = \int \int P_s dV = \int \sigma \cdot \dot{e} dV = \int (Y_0 + K \cdot \dot{e}^m \cdot e^n) \cdot \dot{e} dV = \int (Y_0 \cdot \dot{e} + K \cdot \dot{e}^m \cdot e^n) dV \quad (3.21) \]

If this is being worked out by using the equation for the strain rate we found and formula 3.20, we obtain the following equation:

\[ P_d = 2 \int \int \int (Y_0 \cdot \frac{u}{h} + K \cdot \frac{u^{m+1}}{h} \cdot (\ln \frac{h_0}{h})^n) r dr d\phi dz \quad (3.22) \]

from which follows:
After compression of 1 second we calculated a power of 135.8 kW and after compression of 3 seconds we calculated a power of 165.8 kW. These calculated powers are a little bit higher than in the RIPLE simulation but this is already implied in the name of the method which has been used.

The calculated values are given in the following table.

<table>
<thead>
<tr>
<th>compression</th>
<th>strain</th>
<th>strain rate</th>
<th>power</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>[·]</td>
<td>[s⁻¹]</td>
<td>[kW]</td>
</tr>
<tr>
<td>1 second</td>
<td>4.1⋅10⁻²</td>
<td>4.2⋅10⁻²</td>
<td>135.8</td>
</tr>
<tr>
<td>3 seconds</td>
<td>1.28⋅10⁻¹</td>
<td>4.5⋅10⁻²</td>
<td>165.8</td>
</tr>
</tbody>
</table>

Table 3.2: Results of an analytical approach without friction

### 3.6. COMPRESSION WITH FRICTION

#### 3.6.1. A RIPLE simulation with friction

The same model has been used as for compression without friction. For the frictional factors \( m \) and \( \mu \) we used 0.3, which is a little bit high in reality, but it gives good results to compare. The total compression period was 3 seconds. The result was, as expected, that the deformation is not uniform anymore. A BASIC program has been written to visualize the output. This is necessary because we can no longer speak of stress and strain in general terms.

The dissipated power is calculated for both friction models after 1 and after 3 seconds. The values are 141.5 kW for the Coulomb model and 138.6 kW for the Von Mises model after 1 second. After 3 seconds the powers are 176.7 kW for the Coulomb model and 171.3 kW for the Von Mises model.

The calculated values are given in the following table.
In figures 3.3 and 3.4, simulations of compression for 12 seconds are shown. In figure 3.3, the Von Mises method is used and in figure 3.4, the Coulomb method is used. The definition of the velocity field was in both cases only on the upper surface of the cylinder. It is clearly to be seen that the surface slides less when the Coulomb method is being applied and that the nodes with a lower position are forced up. For the Von Mises method, the surface nodes slide more to the right and therefore the nodes with a lower position stay under the surface nodes.

In figure 3.5 and 3.6, the stress distribution is shown for both compression models.

### 3.6.2. An analytical approach with friction

It is not possible anymore to give the distribution of the stress and strain because the deformation is not uniform anymore. To be able to compare the results that were obtained with the RIPLE simulation we also used the upper bound method as we did in § 3.5.2. We also use a linear velocity field with $u_r$ independent of $z$ and $u_z$ independent of $r$. The part of the calculation of the dissipated power $P_d$ is the same as in § 3.5.2.
Now, we also have a factor for frictional power, $P_f$. The frictional power is only calculated with the Von Mises method because there the stress component is known and constant. $P_f$ can be calculated with formula 3.24.

$$P_f = \frac{2 m \sigma_F}{\sqrt{3}} \int_{A} |u_t| \, dA$$ (3.24)
with $\sigma_p=$ initial yield stress
$A =$ the surface where friction occurs
$u_r =$ the tangential velocity on the surface = $u,$

With $u_r=r \cdot u/2 \cdot h$ (formula 3.13), formula 3.24 can be written as:

$$\int_0^{2\pi} \int_0^{r[u]} r \cdot u \cdot r \cdot dr \cdot d\phi = \frac{2 \cdot Y_0}{\sqrt{3}} \cdot 2\pi \cdot \frac{1}{3} \cdot \frac{R^3 \cdot u}{2 \cdot h}$$ (3.25)

If we substitute the proper values in this formula we derive $P_b = 10.7$ Kw for 1 second of compression and $P_b = 13.3$ kW for 3 seconds of compression. This is about as much as we expected, looking at the difference between the values for the power of the RIPLE simulation with and without friction. If these amounts are added to the analytical solution without friction we get a total of 146.5 kW for 1 second and 179.1 kW for 3 seconds.

The calculated values are given in the following table.

<table>
<thead>
<tr>
<th>compression time</th>
<th>power for Von Mises method [kW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 second</td>
<td>146.5</td>
</tr>
<tr>
<td>3 seconds</td>
<td>179.1</td>
</tr>
</tbody>
</table>

Table 3.4: Results of an analytical approach with friction

An analytical approach with a non linear velocity field is given in appendix A.
3.7. CONCLUSIONS ABOUT THE RESULTS OF THE FRICTIONAL DISCOUNT IN RIPLE

The obtained results are given once again in the following tables:

<table>
<thead>
<tr>
<th>t=1 sec.</th>
<th>without friction [kW]</th>
<th>Coulomb friction [kW]</th>
<th>V. Mises friction [kW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>analytical</td>
<td>135.8</td>
<td>[-]</td>
<td>146.5</td>
</tr>
<tr>
<td>numerical</td>
<td>133.8</td>
<td>141.5</td>
<td>138.6</td>
</tr>
</tbody>
</table>

Table 3.5: Results for 1 second of compression

<table>
<thead>
<tr>
<th>t=3 sec.</th>
<th>without friction [kW]</th>
<th>Coulomb friction [kW]</th>
<th>V. Mises friction [kW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>analytical</td>
<td>165.8</td>
<td>[-]</td>
<td>179.1</td>
</tr>
<tr>
<td>numerical</td>
<td>164.4</td>
<td>176.7</td>
<td>171.3</td>
</tr>
</tbody>
</table>

Table 3.6: Results for 3 seconds of compression

Because the dimension of the workpiece is relatively big, the derived powers are unreasonably high. However, this was useful to compare the results. To give an impression of values which are more real, we also simulated a cylinder with a total height of 20 mm. and a diameter of 10 mm. The velocity is also adjusted to these dimensions and is 0.4 mm s⁻¹.

The obtained results are shown in the following table:

<table>
<thead>
<tr>
<th>t=1 sec.</th>
<th>Power with Coulomb friction model [W]</th>
<th>Power with V. Mises friction model [W]</th>
</tr>
</thead>
<tbody>
<tr>
<td>analytical</td>
<td>[-]</td>
<td>146.5</td>
</tr>
<tr>
<td>numerical</td>
<td>141.1</td>
<td>137.5</td>
</tr>
</tbody>
</table>

Table 3.7: Results for compression of a smaller cylinder for 1 second

<table>
<thead>
<tr>
<th>t=3 sec.</th>
<th>Power with Coulomb friction model [W]</th>
<th>Power with V. Mises friction model [W]</th>
</tr>
</thead>
<tbody>
<tr>
<td>analytical</td>
<td>[-]</td>
<td>179.1</td>
</tr>
<tr>
<td>numerical</td>
<td>176.0</td>
<td>169.9</td>
</tr>
</tbody>
</table>

Table 3.8: Results for compression of a smaller cylinder for 3 seconds
These values are reasonable, and if the power is transformed to machine force by dividing by the velocity, it can be seen that the force can easily be brought in by a universal machine (± 40 tons).

It can generally be seen that the analytically calculated power is higher than the numerical calculated power. There are some reasons that can be mentioned for the difference that occurs between the comparable values.

- For compression with friction, the velocity-field is not linear anymore, which we do assume in the analytical calculation. This leads to differences in the estimated power.

- For the Von Mises friction model, RIPLE uses the effective stress of that moment, while the initial yield stress is used in the analytical calculation. This also leads to different values for the frictional power in the RIPLE simulation.

- As far as the numerical calculation is concerned, we introduce the error that we use the average volume and surface instead of using the real element volume and surface to calculate the power. This has the effect that some values for stress and strain are discounted more than they should be and some other values are discounted less than they should be. This leads to errors in the total power. In the next part of this report, the power is related to the proper volume of every node.

- For the calculation of the power, RIPLE uses the effective stress of the previous stage (the effective stress of the present stage is not known yet when it has to be used for defining the stiffness matrix) and the strain rate of the present stage. Because the stress increases every stage, the power is calculated lower than it should be.

- In general, the upper bound method which we use for the analytical calculation is only an overestimation of the power and can not be equal to the numerical calculated value, which should be more precisely.

Taking all these remarks into account, we can draw the conclusion that the simulation of frictional problems is done in a way that is in accordance with the reality, following the laws of Coulomb and Von Mises. To compare a simulation with an experiment of course, there has to be a good knowledge of the frictional factors.
4. DISCOUNTING DIE AND PUNCH ROUNDINGS FOR DEEPDRAWING IN RIPLE

4.1. INTRODUCTION

It was not possible to use a die or a punch with roundings in the simulation of the deepdrawing process. Only a square punch and die could be used. This gives a lot of problems when the blank has to go around the corner and has a lot of effect on the simulation because in reality, roundings are always being used, otherwise deepdrawing would be too difficult. For this reason, it seemed interesting that this possibility would be built in into RIPLE.

With this possibility it would be very easy to simulate the effects of various punch and die roundings without having to do this by experiment. This can save a lot of time and money.

4.2. CHOOSING THE METHOD FOR DESCRIBING ROUNDINGS

In general there are two ways of describing punch and die shapes (ref. 3). It is (a) possible to use interpolation methods for the elements using splines, Lagrange's and Hermite's formulae and (b) to use approximation methods for the punch and die using basis splines and Bézier curves. For the first method (a) it is necessary to use higher order polynomials to interpolate the element shape. This often results in problems in continuity and more computation time is required. Basis splines and Bézier curves (b) give the advantage in the local control of the curve segment for the modelling of the punch and die shape.

In order to keep the definition of the shape of the elements and because of the advantage of using the second method, we chose to use the method of approximating a quarter of a circle by applying Bézier curves in the RIPLE program.

4.3. BÉZIER CURVES (REF. 3)

Bézier's method uses control vertices to approximate a curve. By multiplying these vertices with a linear equation, the desired shape is obtained.

A Bézier curve is defined in equation 4.1:
Here $Q(u)$ is the curve segment, $V_i$ are the control vertices of the curve segments, $u$ is the parametric variable used to define the linear equations $P_{i,d}$ and therefore describes the curve segment ($0 \leq u < 1$). $P_{i,d}$ are defined by Bernstein polynomials as:

$$P_{i,d}(u) = \binom{d}{i} \cdot u^i \cdot (1-u)^{d-i}$$

Here, $d$ is the number of control vertices minus one (the first control vector is $V_0$) and $i$ represents the present control vector. The factor ($d$ over $i$) gives the number of combinations of $d$ things taken $i$ at a time. For the proper approximation of a rounding that resembles a quarter of a circle, it is useful to use cubic Bézier curves, this means that four control vertices are required, $V_0, V_1, V_2$ and $V_3$. The Bernstein polynomials are then expressed as:

- $P_{0,3} = 1-3u+3u^2-u^3$
- $P_{1,3} = 3u-6u^2+3u^3$
- $P_{2,3} = 3u^2-3u^3$
- $P_{3,3} = u^3$

For every $0 \leq u < 1$, the curve is defined by using equation (4.1). The only things that have to be defined are the positions of the control vertices. To obtain continuity, $V_0$ and $V_3$ must have the same length as the circle radius and have the same direction as respectively the $z$ and the $r$ axis. For the same reason, $V_1$ and $V_2$ must be selected respectively along the opposite horizontal and the opposite vertical line. Figure 4.1 gives an example of this.

To find out the best position of the vertices $V_1$ and $V_2$, a computer program has been written in BASIC and the best values turned out to be $(1.1, -2)$ for $V_1$ and $(2, -1.1)$ for $V_2$ for an approximation of a circle with a radius of 2. This result can be seen in figure 4.2.

As can be seen, the values $V_0(0,-2), V_1=(1.1,-2), V_2=(2,1.1), V_3=(2,0)$ give the best approximation.

4.4. THE IMPLEMENTATION OF THE ROUNDINGS (REF. 3 & 4)

To simulate deepdrawing, the nodes which are in contact with the horizontal part of the punch are given the same velocity as the desired punch velocity as a boundary condition. The contact
The contact problem can be divided in several different parts. For one thing it has to be calculated which nodal point, that is not in contact yet, will touch the rounded surface of the punch or die and after what time that will happen. The nodes that are in contact have to get a velocity vector that is tangential to the tools' surface. Then, the nodal points that are already in contact might lose this if there is tension. This means that the normal stress is lower or equal than 0 (in RIPLE, the stress for compression is positive) and if not, a new position has to be found on the surface according to the tangential movement of the node.

This information is put into the flowchart in figure 4.3 on the next page and can be substituted in "preparing for next stage" in the flowchart of figure 2.1.

4.4.1. Touching of the rounded edges

There are some nodal points that will contact the die or the punch rounding. This depends on the nodal point's velocity and position and this happen after a certain timestep. It is clear that the timestep for the following stage has to be equal to the smallest time after which a node will contact either the die or the punch otherwise the workpiece would penetrate one of the tools. If there is no timestep calculated that is smaller than RIPLE's default timestep (of 0.1 second), then the timestep for the next stage will be equal to the default timestep.

Figure 4.4 gives a schematic plot of the punch, the die and a nodal point.
Assuming that the nodal point \( N \) will touch a rounding we can write the following equation:

\[
Q_s(u) = N_0 + V_r \Delta t
\]  

(4.3)

Here \( Q_s(u) \) is a point on the surface, depending on the value for \( u \) (unknown yet). \( N_0 \) is the initial position of the nodal point, \( V_r \) the nodal point's velocity and \( \Delta t \) the timestep (unknown yet). Equation (4.3) in fact exists of 2 equations, one for the \( r \) direction and one for the \( z \)
direction. Now we have two equations with two unknown variables, \( u \) and \( \Delta t \), and this is to be solved by the Newton-Raphson method with the initial guess of the solution as \( \Delta t = 0 \) and \( u = \frac{|N_0 \cdot V_0|}{(|N_0 \cdot V_0| + |N_0 \cdot V_1|)} \) were \( V \) are the control vertices as can be seen in figure 4.1.

### 4.4.1.1. Touching of the punch

If equation 4.3 is worked out for the contact with the punch, the following relationships can be substituted in this equation.

\[
Q_0(u) = \begin{pmatrix} r \\ z \end{pmatrix} = \begin{pmatrix} r_{qp} \\ z_{qp}(t) \end{pmatrix} + \begin{pmatrix} r(u) \\ z(u) \end{pmatrix} \tag{4.4}
\]

with \( r_{qp} \) and \( z_{qp}(t) \) respectively the \( r \) and \( z \) coordinates of the centre point \( Q \) of the radius. Here, the \( z \) coordinate depends on the punch velocity; \( z_{qp}(t) = z_{qp} + v_{punch} \cdot \Delta t \) and \( v_{punch} \) is negative when moving down. \( R(u) \) and \( z(u) \) give the position on the punch surface relative to \( Q_{punch} \) \((r_{qp}, z_{qp})\).

\[
N_0 = \begin{pmatrix} r_n \\ z_n \end{pmatrix} \tag{4.5}; \quad V_n = \begin{pmatrix} u_n \\ v_n \end{pmatrix} \tag{4.6}
\]

give the nodal point’s position and velocity. See also figure 4.4 for the definition of the punch.

After substitution, equation 4.3 becomes:

\[
\begin{pmatrix} r_q \\ z_q + v_p \cdot \Delta t \end{pmatrix} + \begin{pmatrix} r(u) \\ z(u) \end{pmatrix} - \begin{pmatrix} r_n \\ z_n \end{pmatrix} - \begin{pmatrix} u_n \\ v_n \end{pmatrix} \cdot \Delta t = 0 \tag{4.7}
\]

or

\[
\begin{pmatrix} r_q + r(u) - r_n \\ z_q + z(u) - z_n \end{pmatrix} - \begin{pmatrix} u_n \\ v_n - v_p \end{pmatrix} \cdot \Delta t = 0 \tag{4.8}
\]

For the punch we can substitute the following relationships for \( r(u) \) and \( z(u) \):

\[
\begin{align*}
    r(u) &= -0.7 \cdot k \cdot u^3 - 0.6 \cdot k \cdot u^2 + 3.3 \cdot k \cdot u \\
    z(u) &= -0.7 \cdot k \cdot u^3 + 2.7 \cdot k \cdot u^2 - 2 \cdot k
\end{align*}
\tag{4.9, 4.10}
\]
These relationships are obtained by substituting formula 4.2 in formula 4.1 while using the discrete vertices for a radius of 2. The parameter $k$ is a multiplication factor for the punch radius and is therefore equal to the punch radius divided by 2.

Now it is possible to write $\Delta t$ explicitly from the upper equation of formula 4.8 and we get:

$$\Delta t = (r_{qp} - r_n - k \cdot (0.7 \cdot u^3 + 0.6 \cdot u^2 - 3.3 \cdot u))/u_n$$  \hspace{1cm} (4.11)

If we substitute this relation in the lower equation of formula 4.8 we obtain:

$$(v_n - v_p)/u_n - 1) \cdot 0.7 \cdot k \cdot u^2 + \frac{(v_n - v_p)/u_n + 2.7) \cdot 2 \cdot k \cdot u - 3.3 \cdot (v_n - v_p) \cdot k \cdot u} + z_n - 2 \cdot k \cdot z_n = 0$$  \hspace{1cm} (4.12)

To find the value for $u$ for which this equation is equal to 0 we can use the Newton-Raphson iteration method. This means that we use the following form:

$$u_x = u_{x+1} - f(u_{x+1})/f'(u_{x+1})$$  \hspace{1cm} (4.13)

where $x$ is the number of the present iteration step, the function $f$ is the function that has to be equal to 0 (equation 4.12) and the function $f'$ is the first derivative of this function.

The first derivative of equation 4.12 is:

$$f'(u) = ((v_n - v_p)/u_n - 1) \cdot 2.1 \cdot k \cdot u^2 + (0.6 \cdot (v_n - v_p)/u_n + 2.7) \cdot 2 \cdot k \cdot u - 3.3 \cdot (v_n - v_p) \cdot k \cdot u/n$$  \hspace{1cm} (4.14)

As already mentioned in §4.4.1., for the initial guess we used $u_0 = |N_o - V_0|/(|N_o - V_0| + |N_o - V_1|)$. If this is worked out by substituting the equations for the control vertices and the position for the nodal point, we get:

$$u_0 = \sqrt{(r_n^2 + (z_n + 2 \cdot k)^2))/((r_n^2 + (z_n + 2 \cdot k)^2) + ((r_n - 2 \cdot k)^2 + z_n^2))}$$  \hspace{1cm} (4.15)

To get a quicker convergence, a factor $\beta$ ($\beta < 1$) is introduced so that the Newton-Raphson equation becomes:

$$u_x = u_{x+1} - \beta \cdot f(u_{x+1})/f'(u_{x+1})$$  \hspace{1cm} (4.16)

and the iteration is being performed until the factor $(f(u_{x+1})/f'(u_{x+1}))/u_x$ is small enough (2 x 10^-4).

After a proper value for $u$ is found, it can be substituted in equations 4.9, 4.10 and 4.11 and the timestep and position on the punch are known. It is necessary to check the solutions so that $\Delta t > 0$ and $0 < u < 1$ because values outside these boundaries are useless.
4.4.1.2. Touching of the die

The calculation of the timestep and the possible position on the die is very similar to the one for the punch, only now, because the die has no velocity there will be a timefactor less. For the die, equation 4.3 becomes:

\[
\begin{pmatrix}
  r_{qp} + r(u) - r_n \\
  z_{qp} + z(u) - z_n
\end{pmatrix} - \begin{pmatrix}
  u_n \\
  v_n
\end{pmatrix} \Delta t = 0
\]  

(4.17)

For the definition of the die definition, see figure 4.4.

For the die we can substitute the following relationships for \( r(u) \) and \( z(u) \):

\[
 r(u) = 0.7 \cdot k \cdot u^3 + 0.6 \cdot k \cdot u^2 - 3.3 \cdot k \cdot u \\
 z(u) = 0.7 \cdot k \cdot u^3 - 2.7 \cdot k \cdot u^2 + 2 \cdot k
\]

(4.18) 

(4.19)

These relationships are also obtained by substituting formula 4.2 in formula 4.1, only now multiplied with factor -1 because the dieshape is mirrored in both axes from the punch. Here, also vertices of length 2 were used for these relationships so the value of \( k \) is equal to the diuradius divided by 2.

If we write the value for \( \Delta t \) explicitly from the upper relation in equation 4.17 we get:

\[
\Delta t = (r_{qp} - r_u + k \cdot (0.7 \cdot u^3 + 0.6 \cdot u^2 - 3.3 \cdot u))/u_n
\]

(4.20)

If we substitute this in the lower equation we get:

\[
f(u) = (1 - v_n/u_n) \cdot 0.7 \cdot k \cdot u^3 - (2.7 + 0.6 \cdot v_n/u_n) \cdot k \cdot u^2 + 3.3 \cdot v_n/u_n \cdot k \cdot u + z_q - z_n - (r_{qp} - r_n) \cdot v_n/u_n + 2 \cdot k = 0
\]

(4.21)

To find the proper value for \( u \) we also apply the Newton-Raphson method of the form:

\[
u_{n+1} = u_n - \frac{f(u_n)}{f'(u_n)}
\]

(4.16)

where \( f'(u) = (1 - v_n/u_n) \cdot 2.1 \cdot k \cdot u^2 - (2.7 + 0.6 \cdot v_n/u_n) \cdot 2 \cdot k \cdot u + 3.3 \cdot v_n/u_n \cdot k
\) 

(4.22)

If the factor \( (f(u_n)) / f'(u_n)) \) is satisfactory small enough, the timestep can be calculated and the possible position on the die of the nodal point is known. Here the solution should also be checked on \( \Delta t > 0 \) and \( 0 < u < 1 \).

When all the data are calculated for the nodes which are nominated to touch the punch or the die, the boundary condition of the nodal point with the smallest timestep will change so that that nodal point will be on the surface of the punch respectively of the die in the next stage. The timestep for the next stage is equal to the timestep calculated for that nodal point.
4.4.2. Checking if contact will be continued

For the nodal points which are in contact with the punch or die it has to be checked whether this contact will be continued or not. A way to decide this is to check the normal stress to see if there is tension or compression working on the element.

Because RIPLE uses a positive stress in the case of compression, the contacting nodal points should loose this contact if the normal stress on the element where it belongs to is lower or equal to 0. To calculate the normal stress on the element, it is only necessary to use the perpendicular and the radial stress, in the global system, because the problems concerned are axi-symmetric. The normal stress is related to the angle under which the nodal point is positioned. The relationship is given below:

\[ \sigma_n = \sigma_z \sin(\alpha) + \sigma_r \cos(\alpha) \]  \hspace{1cm} (4.23)

The definition of \( \alpha \) is given in figure 4.5.

To calculate \( \sigma_z \) and \( \sigma_r \), the same method is applied as in §3.4.

4.4.3. Finding a new position on the surface

If the contact of a nodal point with the tool will be continued, it is necessary to find a new point on the surface of the tool. This can be done by finding the minimum distance between the nodal point and the tool surface. The definition of the punch and the die is given once again in figure 4.4. At this moment, the position of the nodal point is already updated so we can write the following equation:

\[ I(u) = |N_1 - Q_1(u)| \]  \hspace{1cm} (4.24)

Here, \( I(u) \) represents the distance between the nodal point and the tool surface which has to be minimized. \( N_1 \) is the position that the nodal point would get if contact would be lost, after being updated and \( Q_1(u) \) is the new unknown position on the tool that has to be found.
If this equation is written out, we have one equation to be solved with one unknown variable. The solution can be found by applying the Newton-Raphson method again.

### 4.4.3.1. Finding the closest point on the punch

If equation 4.24 is written out, we can write for the distance of the nodal point and the punch:

\[
IP^2 = ((r_n) - (r(u) + r_{qp}))^2 + ((z_n) - (z(u) + z_{qp}))^2
\]  

(4.25)

where:
- \( r_n \) = nodal point \( r \) position
- \( z_n \) = nodal point \( z \) position
- \( r(u) \) = \( r \) position on punch surface (relative to \( Q_p \) punch)
- \( r_{qp} \) = \( r \) coordinate \( Q_p \) punch (centre of punchrounding)
- \( z(u) \) = \( z \) position on punch surface (relative to \( Q_p \) punch)
- \( z_{qp} \) = \( z \) coordinate \( Q_p \) punch (centre of punchrounding)

We don't have to take care of the velocity of the punch, because at the moment when this calculation takes place, the position of the punch has already been updated so it is just like a momentary situation.

With the equations for \( r(u) \) and \( z(u) \) (in which \( k \) is a multiplication factor for the punchradius, equal to half the punchradius, see also § 4.4.1.1.):

\[
r(u) = -0.7 \cdot k \cdot u^3 - 0.6 \cdot k \cdot u^2 + 3.3 \cdot k \cdot u
\]

(4.9)

\[
z(u) = -0.7 \cdot k \cdot u^3 + 2.7 \cdot k \cdot u^2 - 2 \cdot k
\]

(4.10)

we can write for the square of the distance:

\[
IP^2 = ((r_n) + (0.7 \cdot k \cdot u^3 + 0.6 \cdot k \cdot u^2 - 3.3 \cdot k \cdot u - r_{qp}))^2 \quad \text{←A}
\]

\[
+ ((z_n) + (0.7 \cdot k \cdot u^3 - 2.7 \cdot k \cdot u^2 + 2 \cdot k - z_{qp}))^2 \quad \text{←B}
\]

(4.26)

With \( (2 \cdot k \cdot z_{qp}) = c_z \):

\[
A := 0.49 \cdot k^2 \cdot u^4 + 0.84 \cdot k^2 \cdot u^3 + (0.36 \cdot k^2 + 4.62 \cdot k^2) \cdot u^2 + (1.4 \cdot k \cdot r_n - 1.4 \cdot k \cdot r_{qp} + 3.96 \cdot k^2) \cdot u^3 +
\]

\[-(1.2 \cdot k \cdot r_n \cdot 1.2 \cdot k \cdot r_{qp} + 10.89 \cdot k^2) \cdot u^2 + (6.6 \cdot k \cdot r_{qp} - 6.6 \cdot k \cdot r_n) \cdot u - 2 \cdot r_n \cdot r_{qp} + r_{qp}^2 + r_n^2
\]

(4.27)

\[
B := 0.49 \cdot k^2 \cdot u^4 - 3.78 \cdot k^2 \cdot u^3 + 7.29 \cdot k^2 \cdot u^2 + (1.4 \cdot k \cdot z_n + 1.4 \cdot k \cdot c_i) \cdot u^3 +
\]

\[-(5.4 \cdot k \cdot z_n + 5.4 \cdot k \cdot c_i) \cdot u^2 + z_n^2 + 2 \cdot z_n \cdot c_i + c_i^2
\]

(4.28)
\[ IP^2 = A + B = 0.98k^2u^4 - 2.94k^2u^5 + 3.03k^2u^4 + (1.4k\tau_u + 1.4kz_u + \\
+ 1.4kc_i - 1.4k\tau_{qp} - 3.96k^2)u^3 + (1.2k\tau_u - 5.4kz_u + \\
- 5.4kc_i - 1.2k\tau_{qp} + 10.89k^2)u^2 + (6.6k\tau_{qp} - 6.6k\tau_u)u + \\
+ z_{\alpha}^2 + 2z_{\alpha}c_i + c_i^2 - 2\tau_u\tau_{qp} + \tau_{qp}^2 + \tau_{\alpha}^2 \] (4.29)

This equation has to reach a minimum value, this means that we have to find the appropriate value for \( u \) for which the first derivative of the equation is equal to 0. This can be done by using \( u_0 = u_{s,1} - \beta f'(u_{s,1})/f''(u_{s,1}). \) For the initial guess of \( u \) we use \( u_0 = \left| N_1-V_s \right|/\left( \left| N_1-V_s \right| + \\
+ \left| N_1-V_s \right| \right) \)

The first derivative of equation 4.29 is \( \delta IP^2/\delta u = \)

\[ IP^2 = 5.88k^2u^4 - 14.7k^2u^5 + 12.12k^2u^4 + (4.2k\tau_u + 4.2kz_u + 4.2kc_i + \\
- 4.2k\tau_{qp} + 11.88k^2)u^2 + (2.4k\tau_u - 10.8kz_u - 10.8kc_i - 2.4k\tau_{qp} + 21.78k^2)u + \\
+ 6.6k\tau_{qp} - 6.6k\tau_u \] (4.30)

The second derivative of equation 4.29 is \( \delta^2 IP^2/\delta u^2 = \)

\[ IP^2 = 29.4k^2u^4 - 58.8k^2u^5 + 36.36k^2u^4 + (8.4k\tau_u + 8.4kz_u + 8.4kc_i + \\
- 8.4k\tau_{qp} - 22.16k^2)u^2 + 2.4k\tau_u - 10.8kz_u - 10.8kc_i - 2.4k\tau_{qp} + 21.78k^2 \] (4.31)

The iteration is carried out until the factor \( (IP^2/IP^2)/u \) is small enough. The factor \( \beta \ (<1) \) is used again to make the convergence go faster.

4.4.3.2. Finding the closest point on the die

To find the closest point on the surface of the die, is similar to the way this is done for the punch. For the distance of the nodal point and the die we can write:

\[ ID^2 = ((r_u)-(r(u) + r_{qp}))^2 + ((z_u) - (z(u) + z_{qp}))^2 \] (4.32)

\[ r_u = \text{nodal point } r \text{ position} \]
\[ z_u = \text{nodal point } z \text{ position} \]
\[ r(u) = \text{r position on die surface (relative to Q die)} \]
\[ r_{qp} = \text{r coordinate Q die (centre of the rounding)} \]
\[ z(u) = \text{z position on die surface (relative to Q die)} \]
\[ z_{qp} = \text{z coordinate Q die (centre of the rounding)} \]
With the equations for \( r(u) \) and \( z(u) \):

\[
\begin{align*}
    r(u) &= 0.7 \cdot k \cdot u^3 + 0.6 \cdot k \cdot u^2 - 3.3 \cdot k \cdot u \\
    z(u) &= 0.7 \cdot k \cdot u^3 - 2.7 \cdot k \cdot u^2 + 2 \cdot k
\end{align*}
\]

(4.18) (4.19)

For the square of the distance we can write:

\[
\begin{align*}
    \mathbf{D}^2 &= ((r_m) - (0.7 \cdot k \cdot u^3 + 0.6 \cdot k \cdot u^2 - 3.3 \cdot k \cdot u + r_{qd}))^2 + \\
    &+ ((z_m) - (0.7 \cdot k \cdot u^3 - 2.7 \cdot k \cdot u^2 + 2 \cdot k + z_{qd}))^2
\end{align*}
\]

\[\begin{align*}
    A &= 0.49 \cdot k^2 \cdot u^6 + 0.84 \cdot k^2 \cdot u^5 + (0.36 \cdot k^2 - 4.62 \cdot k^2) \cdot u^4 + \\
    &+ (1.4 \cdot k \cdot r_{qd} - 1.4 \cdot k \cdot r_{nt} - 3.96 \cdot k^2) \cdot u^3 + \\
    &+ (1.2 \cdot k \cdot r_{qd} - 1.2 \cdot k \cdot r_{nt} + 10.89 \cdot k^2) \cdot u^2 + (6.6 \cdot k \cdot r_{nt} - 6.6 \cdot k \cdot r_{qd}) \cdot u + r_{nt}^2 + r_{qd}^2
\end{align*}
\]

(4.33)

\[\begin{align*}
    B &= 0.49 \cdot k^2 \cdot u^6 - 3.78 \cdot k^2 \cdot u^5 + 7.29 \cdot k^2 \cdot u^4 + (1.4 \cdot k \cdot z_{qd} - 1.4 \cdot k \cdot z_{nt} + 2.8 \cdot k^2) \cdot u^3 + \\
    &+ (5.4 \cdot k \cdot z_{nt} - 5.4 \cdot k \cdot z_{qd} - 10.8 \cdot k^2) \cdot u^2 + z_{qd}^2 - 2 \cdot z_{nt} \cdot z_{qd} + 4 \cdot z_{qd} \cdot k + 4 \cdot k^2 - 4 \cdot k \cdot z_{nt} + z_{nt}^2
\end{align*}
\]

(4.35)

\[\begin{align*}
    \mathbf{D}^2 &= A + B = 0.98 \cdot k^2 \cdot u^6 - 2.94 \cdot k^2 \cdot u^5 + 3.03 \cdot k^2 \cdot u^4 + (1.4 \cdot k \cdot r_{nt} + 1.4 \cdot k \cdot z_{qd} + \\
    &- 1.4 \cdot k \cdot z_{nt} - 1.4 \cdot k \cdot r_{nt} - 1.16 \cdot k^2) \cdot u^3 + (1.2 \cdot k \cdot r_{qd} + 5.4 \cdot k \cdot z_{nt} + \\
    &- 5.4 \cdot k \cdot z_{qd} - 1.2 \cdot k \cdot r_{nt} + 0.09 \cdot k^2) \cdot u^2 + (6.6 \cdot k \cdot r_{nt} - 6.6 \cdot k \cdot r_{qd}) \cdot u + z_{qd}^2 + \\
    &- 2 \cdot z_{nt} \cdot z_{qd} + 4 \cdot z_{qd} \cdot k + 4 \cdot k^2 - 4 \cdot k \cdot z_{nt} + z_{nt}^2 - 2 \cdot r_{nt} \cdot r_{qd} + r_{nt}^2 + r_{qd}^2
\end{align*}
\]

(4.36)

The first derivative of this equation is:

\[
\begin{align*}
    \frac{\delta \mathbf{D}^2}{\delta u} &= \mathbf{D}^{\prime\prime} = 5.88 \cdot k^2 \cdot u^4 - 14.7 \cdot k^2 \cdot u^3 + 12.12 \cdot k^2 \cdot u^2 + (4.2 \cdot k \cdot r_{qd} + 4.2 \cdot k \cdot z_{qd} + \\
    &- 4.2 \cdot k \cdot z_{nt} - 4.2 \cdot k \cdot r_{nt} - 3.48 \cdot k^2) \cdot u + (2.4 \cdot k \cdot r_{qd} + 10.8 \cdot k \cdot z_{nt} + \\
    &- 10.8 \cdot k \cdot z_{qd} - 2.4 \cdot k \cdot r_{nt} + 0.18 \cdot k^2) \cdot u + 6.6 \cdot k \cdot r_{nt} - 6.6 \cdot k \cdot r_{qd}
\end{align*}
\]

(4.37)

and the second derivative is:

\[
\begin{align*}
    \frac{\delta^2 \mathbf{D}^2}{\delta u^2} &= \mathbf{D}^{\prime\prime\prime} = 29.4 \cdot k^2 \cdot u^4 - 58.8 \cdot k^2 \cdot u^3 + 36.36 \cdot k^2 \cdot u^2 + (8.4 \cdot k \cdot r_{qd} + 8.4 \cdot k \cdot z_{qd} + \\
    &- 8.4 \cdot k \cdot z_{nt} - 8.4 \cdot k \cdot r_{nt} - 6.96 \cdot k^2) \cdot u + 2.4 \cdot k \cdot r_{qd} + 10.8 \cdot k \cdot z_{nt} - 10.8 \cdot k \cdot z_{qd} + \\
    &- 2.4 \cdot k \cdot r_{nt} + 0.18 \cdot k^2
\end{align*}
\]

(4.38)

With these equations and the same initial guess as for the punch, the solution for \( u \) for which we get the smallest distance can be found.
4.4. Tangential Velocity Field

If a nodal point touches one of the toolroundings, the velocity field has to become tangential to the tool. Because the die is fixed, the velocity in the z and in the r direction depend on each other in the following way:

\[
\begin{bmatrix}
V_n \\
V_t
\end{bmatrix} =
\begin{bmatrix}
0 \\
V_r
\end{bmatrix} =
\begin{bmatrix}
V_z \sin(\alpha) - V_r \cos(\alpha) \\
V_z \cos(\alpha) + V_r \sin(\alpha)
\end{bmatrix}
\]

(4.39)

Here, \(V_r\) and \(V_z\) are respectively the nodal point velocity in \(r\) and \(z\) direction. The derivation of this relation can be seen from figure 4.6.

Now we have these relationships, we can use the upper one to write \(V_z\) as:

\[
V_z = \frac{V_r \cos(\alpha)}{\sin(\alpha)}
\]

(4.40)

and use this relationship when the velocity for the nodal point on the die surface is calculated.

For the punch, we have to take into account that it has a certain velocity in \(z\) direction. If this is done following figure 4.7, we get the relationship for the tangential velocity:

\[
\begin{bmatrix}
V_n \\
V_t
\end{bmatrix} =
\begin{bmatrix}
0 \\
V_r
\end{bmatrix} =
\begin{bmatrix}
-V_p \sin(\alpha) \\
-V_p \cos(\alpha)
\end{bmatrix} =
\begin{bmatrix}
V_z \sin(\alpha) - V_r \cos(\alpha) \\
V_z \cos(\alpha) + V_r \sin(\alpha)
\end{bmatrix}
\]

(4.41)

Here, \(V_r\) and \(V_z\) are the nodal point velocity in \(r\) and \(z\) direction again and \(V_p\) is the punch velocity.

We can write the upper relationship to:

\[
V_z = \frac{V_p + V_r \cos(\alpha)}{\sin(\alpha)}
\]

(4.42)

and use this relationship when the velocity for the nodal point on the punch surface is calculated.
4.5. Simulation of the Deepdrawing Process While Discounting Die and Punch Roundings

When working on the program, a lot of simulations of deepdrawing and parts of these simulations have been carried out to check the results of the implemented parts.

It became clear that using the normal stress to check the contact did not work out as it should. At first, when the contact between a node and a tool would be lost the node was defined free in the next stage. In the next stage, the new velocity for that node was being calculated with no restrictions for the velocity field. It turned out that the velocity calculated for the node sometimes made the node go into the tool. This is an unexpected effect, because it is logical that if it has been decided to loose contact, the velocity of the node concerned would be calculated in the opposite direction of the tool.

The appearance of this effect makes it look like the criterium to use the normal stress is too rough to judge whether contact will be continued or not. This can be understood because the normal stress used in the 2D program in fact is an averaged value calculated from the strain rate between the four nodes. Especially in the simulation for deepdrawing where relatively thin blanks are used with only a few layers in the z-direction, there can be a large variation in the normal stress. Therefore, it is not unreasonable to suppose that the normal stress on the surface is different from the average normal stress of the concerned surface element.

To prevent the situation mentioned above, it was decided that the checking whether contact would be lost or not would be done before calculating the smallest timestep so that the node that loses contact has the possibility to contact the tool again. As expected in the situation described above, where the node would go inside the tools, the smallest timestep was calculated for that node because of the close distance to the tool and the tangential velocity. This was the reason that an effect was created in which one node keeps on switching between getting contact because of the smallest timestep and losing contact because of the tensile stress. For this node, the timestep being calculated was extremely small so that the total time is being updated with a very small value every stage. This results in an almost stagnating situation.

Because it was not possible to judge on the contact, we chose to let the blank follow the die and punch roundings. This resulted in complete deepdrawing simulations.

While using friction in the simulations, we encountered problems with the convergence. It was not possible to get a complete drawn cup. Because of this reason we decided not to use friction in the program to verify the results.
The setup of the model is done as in figure 4.8. The way of numbering the nodes is very important, because of the bandwidth of the stiffness matrix. Here, the numbering is done from the bottom to the top. This makes the bandwidth small. If the numbering is done from the left to the right, the bandwidth will become larger because the difference between the node number and the element number is bigger.

![fig 4.8: The deepdrawing model](image)

4.5.1. Simulations with several punch and die configurations

To check the results of the simulation, we chose to use different punch and die configurations from which we would use one configuration for experimental verification. We only simulated half of the process to save computing time. This is possible because the model is symmetric in the z-axis.

In the following part, results are being shown of simulation with a configuration of punches with diameters of 71 and 80 mm. and a rounding of 8 mm. The die being used has a diameter of 90 mm. and a rounding of 12 mm.

The material properties used for the simulation was SS41 steel with the following rigid plastic properties (ref. 2):

\[
\begin{align*}
Y &= 200 \text{ [N mm}^2\text{]} \\
K &= 540 \text{ [N mm}^2\text{]} \\
m &= 0.01 \\
n &= 0.155 \\
\end{align*}
\]

D.E. ten Have, Tsukuba, 11-1991
4.5.1.1. Simulation with punch diameter 80 mm. and die diameter 90 mm.

As mentioned above, the configuration consisted of a punch with diameter of 80 mm. and a radius of 8 mm. and a die with diameter 90 mm. and a radius of 12 mm. The blank has a diameter of 155 mm. and a thickness of 3.2 mm.

In the figures 4.9, 4.10 and 4.11, the simulation is shown in different steps until a complete cup has been drawn. The effect of a blankholder is approximated by giving the nodes on the horizontal part of the die a restriction in the z direction.

---

D.E. ten Have, Tsukuba, 11-1991
4.5.1.2. Simulation with punch diameter 71 mm. and die diameter 90 mm.

In the simulations used for experimental verification, we wanted to look at the effect of the blankthickness and the blankradius. For this reason and because of the availability of certain blanks for later experimental comparison, we chose to simulate the following situations:

<table>
<thead>
<tr>
<th>simulationnumber</th>
<th>Blankdiameter [mm.]:</th>
<th>Blankthickness [mm.]:</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>130</td>
<td>4.2</td>
</tr>
<tr>
<td>V2</td>
<td>155</td>
<td>3.2</td>
</tr>
<tr>
<td>V3</td>
<td>162</td>
<td>3.2</td>
</tr>
</tbody>
</table>

Table 4.1: Three different simulations

The toolconfiguration consisted of a punch with diameter of 71 mm. and a radius of 8 mm. and a die with diameter 90 mm. and a radius of 12 mm.
4.5.1.2.1. Simulation V1

In figure 4.12 and 4.13, the stress and strain distribution of the completely drawn cup are given.

4.5.1.2.2. Simulation V2

Of the simulation with the blank with a diameter of 155 mm. and a thickness of 3.2 mm., the stress and strain distribution are given in the figures 4.14 and 4.15. Here, it is also to be seen that the cup has been drawn completely.
4.5.1.2.3. Simulation V3

In figures 4.16 and 4.17 the simulation with a blank of 162x3.2 is shown for 2 different stages. It is clearly to be seen that necking occurs. After 119 stages, the program does not converge anymore and the stress where necking occurs becomes very high. This can be an indication that in reality, fracture will occur.

fig 4.16: Simulation V3 for 50 stages

fig 4.17: Simulation V3 for 119 stages
5. EXPERIMENTAL VERIFICATION

Only one series of experiments have been done because of a shortage of time. It would be very interesting to do more experiments with which RIPLE could be checked on more points.

For the experiments we used the material from which the properties were used in the simulations, SS41 steel. In table 5.1, some results are given of tensile tests, done to measure the properties of SS41 steel (ref.2).

<table>
<thead>
<tr>
<th>Tensile direction to R.D.</th>
<th>0(^\circ)</th>
<th>45(^\circ)</th>
<th>90(^\circ)</th>
<th>Ave.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(m)-value</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>(n)-value</td>
<td>0.15</td>
<td>0.21</td>
<td>0.16</td>
<td>0.17</td>
</tr>
<tr>
<td>(r)-value</td>
<td>0.79</td>
<td>0.76</td>
<td>0.98</td>
<td>0.84</td>
</tr>
<tr>
<td>Yielding point, MPa</td>
<td>272</td>
<td>278</td>
<td>265</td>
<td>272</td>
</tr>
<tr>
<td>Tensile strength, MPa</td>
<td>525</td>
<td>600</td>
<td>530</td>
<td>563</td>
</tr>
</tbody>
</table>

Table 5.1: Material properties of SS41 steel

The press being used was an AIDA multi axial press. This press is equipped with sidetools so that the blank can also be pushed from the sides. For this verification, we didn't use this option. Figure 5.1 shows the press.

Figure 5.2 shows a deepdrawn cup, made of a blank with a thickness of 3.2 mm. and a diameter of 155 mm. A clearance has been used between the blank and the blankholder. The effect of this experiment is that the sides of the cup are wrinkled.
5.1. Verification of the simulations

To minimise the effect of friction, because we did not do the simulations with friction, we used MoS₂ for lubrication between blank and die. As we did in the simulation, there was no clearance between blank and blankholder.

Figures 5.3, 5.4 and 5.5 show cups of respectively the simulations V1, V2, V3 (see table 4.1 in §4.5.1.2)

In figures 5.3 and 5.4, it can be seen that the cups are completely drawn. In the cup with a diameter of 162 mm. however, fracture occurred and it was not possible to draw a complete cup. This is what we already expected after having done the simulations.

Of the cups, parts were cut so that they could be compared with the simulations. In figures 5.6, 5.7 and 5.8, the cups are shown with the outline of the cut parts on the left and the simulated part of the cups on the right side. In figure 5.8, it can be seen that at the moment that fracture occurred in the experiment, much necking occurred in the simulation. At that moment, the program did not converge anymore.
EXPERIMENTAL VERIFICATION

fig 5.4: A cup with $D=155$ mm. and $T=3.2$ mm.

fig 5.5: A cup with $D=162$ mm. and $T=3.2$ mm.

figure 5.6: A complete cup, assembled from an experiment and a simulation. $D_{\text{blank}} = 130$ mm., $T_{\text{blank}} = 4.2$ mm.

figure 5.7: A complete cup, assembled from an experiment and a simulation. $D_{\text{blank}} = 155$ mm., $T_{\text{blank}} = 3.2$ mm.
5.2. LOAD OUTPUT

Because of the usefulness to estimate the load-punch stroke curve in drawing, we wrote the calculated load, taking each element's proper volume into account, to an output file so that it can be compared with the load-punch stroke curve of the experiments. This calculated load can also be used to see the effect of, for example, deepdrawing with different clearances between punch and die.

In figure 5.9, the load curve is given for simulation V2 and for the appropriate experiment. It can be seen that the maximum of the load for the simulation is much lower than the maximum for the experiment. Probably the main reason is that the simulation is done without friction and in reality, the frictional force is relatively high. This can also be the reason that the maximum is reached sooner for the simulation.

In figure 5.10, we can also see that the maximum load is lower for simulation V3 than for the experiment. Probably the reason for this is the same as the one mentioned above. In this simulation, the load does not go back to 0, because the program stops converging at a certain
moment when there is too much necking in the cup and no solution for the velocity field can be found.
6. GENERAL CONCLUSION

In the first part of this research, it could be seen that the use of friction is good and according to the reality. The only problem, as could be seen later, is that for some processes like deepdrawing, the program does not converge when using friction. It might be useful to pay attention to this problem in the future.

In the second part, the use of toolroundings has been implemented and some deepdrawing simulations have been carried out. Here, we have found a method with which it is possible to discount the effect of die and punch roundings in deepdrawing. In this part, some contact problems were encountered. It seemed not to be possible to use the normal stress as a criterium to judge on the contact between the tool and the workpiece. When this criterium has not been used, it was possible to simulate a complete deepdrawing process that is very well to be compared with the reality. Simulations are a good method of predicting the progress of a process and to predict when fracture will occur.

The load curve of the simulations is too low compared with the load output of the press, probably because friction has not been used in the simulation of the deepdrawing process, and therefore it can not be used to predict the maximum load. The load curve can be used however, to compare simulations with each other to find the optimal blankthickness or tool configuration, because of the real shape of the curve.
REFERENCES


AN ANALYTICAL APPROACH WITH ANOTHER VELOCITY FIELD

APPENDIX A.
AN ANALYTICAL APPROACH WITH ANOTHER VELOCITY FIELD

To be able to give a better estimation of the power a velocity field with which bulging of the cylinder is possible is chosen.

For the velocity in \( r \) direction we now choose a linear velocity field that is dependent of \( r \) as well as \( z \) (equation (A.1)).

\[
 u_r = \left( \frac{\alpha - \frac{z}{h/2}}{\beta} \right) \frac{r}{R} \tag{A.1}
\]

The principle of global volume invariance leads to the solution for \( \beta \) and with local volume invariance it is possible to calculate the velocity in \( z \)-direction.

From the velocity field we can derive the effective strain rate (equation (A.2)).

\[
 \dot{\varepsilon} = \frac{1}{2} \left( \frac{4z}{h} - 1 \right) \cdot \left( \alpha + \frac{R}{h} \right)^2 + \frac{4}{3} \cdot \left( \frac{\alpha r}{h} \right)^2 \cdot \left( \frac{u}{R} \right)^{\frac{1}{2}} \tag{A.2}
\]

The dissipated power in the compression process is described in equation (A.3)

\[
P_d = \int \int \int \sigma \cdot \dot{e} \, dV = \int \int \int (Y_0 + K \dot{e}^m \dot{\varepsilon}^p) \dot{e} \, dV = \int \int \int (Y_0 \dot{e}^m + K \dot{e}^{m+1} \dot{\varepsilon}^p) \, dV \tag{A.3}
\]

equation (A.2) substituted in equation (A.3) leads to the following equation (A.4):

\[
\begin{align*}
\int \int \int & (Y_0 \cdot \left( \left( \frac{4z}{h} - 1 \right) \cdot \alpha + \frac{R^2}{h} + \frac{4}{3} \cdot \left( \frac{\alpha r}{h} \right)^2 \cdot \left( \frac{u}{R} \right)^{\frac{1}{2}} \right) r \, dr \, d\phi \, dz + \\
\int \int \int & (K \cdot \left( \left( \frac{4z}{h} - 1 \right) \cdot \alpha + \frac{R^2}{h} + \frac{4}{3} \cdot \left( \frac{\alpha r}{h} \right)^2 \cdot \left( \frac{u}{R} \right)^{\frac{1}{2} (m+1)} \right) \dot{\varepsilon}^p ) r \, dr \, d\phi \, dz
\end{align*}
\tag{A.4}
\]

With the following equation we can eliminate \( \varepsilon \) from formula (A.4) (this is also possible using formula (3.20) in the text, which should give the same result):
It is possible to substitute equation (A.5) in equation (A.4) because there are no time dependent factors in equation (A.4). If we do this we obtain equation (A.6).

\[
\bar{\delta}e = \int_0^t \delta t = t \bar{e}
\]  \hspace{1cm} (A.5)

Now this equation has to be solved and differentiated to \(\alpha\) which leads to the minimum value for \(\alpha\).