Simplified modelling of Fixtures in FE Welding Simulation

OLOF WIIPPOLA

MASTER OF SCIENCE PROGRAMME
Engineering Physics

Luleå University of Technology
Department of Mathematics
Simplified modelling of fixtures in FE welding simulation

December 12, 2006

OLOF WIIPPOLA
olof.wiippola@gmail.com

Master Thesis
Institute of mathematics
Luleå University of Technology
Abstract

The goal with this master thesis work, is to find a simplified way of modelling the fixture in FE welding simulation. To take the entire fixture into the welding model, is very CPU-time consuming since the contact between the fixture and the work piece is difficult to simulate. The method used today, is based on fixed degree of freedom boundary condition. This method do not include the stiffness of the fixture, the friction and the thermal effects between the fixture and the work piece. In order to get a more accurate modell, these things can not be neglected. The method that is presented in this master thesis report, uses already existent functions in the FEM program MSC.Marc. The main idea is to use non-linear springs and non-linear elastic foundation to model friction and support. Since the contact force between the fixture and the work piece directly affects the friction force, the distribution functions that describes how a known contact force at one point distributed through the whole fixture must be used. This function is constructed in two ways, the first uses the entire model with the fixture, the second uses Buckingham’s pi theorem. The tests that where carried out shows that the distribution functions gives a good results, but there are several problems that must be solved before the methods can be used in welding simulations. The subroutine used in welding simulation do not support the needed new style table input which is needed in order to use edge foundation as the friction force. MSC.Marc has problem with interaction between the facefilm and the edge foundation. The problem with the facefilm and the edge foundation is most likly a bug, since this problem do not accrue in the new release of MSC.Marc r3.

Keywords: welding simulation, non-linear spring, elastic foundation, edge foundation, face foundation, contact
Preface

This master thesis work has been carried out at Volvo Aero Corporation in Trollhättan. I would like to thank my supervisors, Senior lector Bo Kjellmert at LTU and Torbjörn Kvist at Volvo Aero. A special thanks to Dr. Henrik Alberg for his support and advice during my work at Volvo aero. I would also like to thank the staff of 9634 for there support.

Olof Wiippola
Trollhättan, December 12, 2006
# Contents

1. **Introduction**  
   1

2. **Problem statement**  
   2  
   2.1 Background  
   2  
   2.2 Definition of the problem  
   2  
   2.3 Problem setup  
   3

3. **Methods**  
   4  
   3.1 Definitions of functions in MSC.Marc  
   4  
   3.2 Modelling of contact  
   6  
   3.2.1 Shell contact  
   6  
   3.2.2 Detection of contact  
   6  
   3.2.3 Contact tolerance  
   6  
   3.2.4 Contact between deformable bodies  
   7  
   3.2.5 Penetration  
   7  
   3.2.6 Heat flux due to contact between deformable bodies  
   7  
   3.3 Modelling of friction and support without contact  
   9  
   3.3.1 Springs  
   9  
   3.3.2 Elastic foundation  
   9  
   3.3.3 Face film  
   10

4. **Modelling and solution**  
   11  
   4.1 The models  
   11  
   4.1.1 The reference model  
   11  
   4.1.2 The simplified model  
   11  
   4.2 First step, friction test  
   12  
   4.2.1 Contact force distribution  
   13  
   4.2.2 How to use the distribution function  
   17  
   4.2.3 Implementation and result of the first test using polynomial fitting  
   18  
   4.2.4 Implementation and result of the first test using Buckingham’s Pi theorem  
   20  
   4.2.5 How to use results of the first test  
   21
4.3 Second step .............................................. 22
4.4 Welding simulations ........................................ 24
  4.4.1 Welding simulation setup .............................. 24
  4.4.2 The first welding simulation ......................... 24
  4.4.3 Face film together with elastic foundation ........ 24

5 Conclusions ............................................. 26
  5.1 Errors in the simplified model ......................... 26
  5.2 Recommendations ...................................... 27

A Full size plots ........................................... a

B Element type 75 ......................................... f
  B.1 Properties ............................................. f
  B.2 Bilinear interpolation ................................. f

C Element type 7 ........................................... g

D Matlab code ............................................. h
  D.1 Distribution function construction using Buckingham’s Pr Theorem ........................ h
  D.2 Distribution function construction using numerical polyfit ......................... i
Chapter 1

Introduction

Welding of details to the aero industry is a very complex task. There are no room for errors during the welding process. That is why, it is important to know as much as possible about how the material is going to be deformed during the welding process. The deformation and stress in the material during and after welding are of interest. To make physical tests of deformation and stress during the development step of a new details is both cost and time consuming. That is why, simulation tools are often used. One method that is often used is the Finite Element Method (FEM). FEM is a numerical method of solving mathematical problems. In this Master Thesis, the possibility of an alternative way of fixture representation is investigated.
Chapter 2

Problem statement

This chapter gives the background and definition of the problem on which the master thesis “Simplified modelling of fixtures in FE welding simulation” is based.

2.1 Background

In the aero space industries, it is very important to have good knowledge of all problems that may occur during the manufacturing process. To identify these problems by experiment is both time and cost consuming. That is why, simulation tools are used.

In welding simulations, deformations and stresses are often interesting. The purpose of the welding simulations is to find optimal manufacturing processes and manufacturing concepts. The welding processes are very complicated and CPU-time consuming to model. Efficient methods of fixture simplification does not exist at Volvo Aero Corporation today. If the whole fixture would be a part of the simulation model, the CPU-time that would be required in order to get good simulation results would increase to a level that is impossible to work with. That is why Volvo Aero Corporation today does not include the fixture into the large welding models. Instead fix degree of freedom (D.O.F) at the boundaries are used. These simplifications do not include the effects of the fixture such as the stiffness of the fixture, friction and thermal effects between the fixture and the work piece.

2.2 Definition of the problem

As mention earlier, the friction between the fixture and welding object is today neglected in the welding simulations. The goal of this master thesis work is to develop a method that includes the stiffness of the fixture, friction and thermal effects between the fixture and the work piece into the FE welding simulations, without significantly increasing the CPU-time. The
fixture and plate on which the simulations are based are shown in figure 2.1 and 2.2.

2.3 Problem setup

The fixture in 2.1 is the reference model, on which the simplifications are to be made. Figure 2.2 shows the plate that is to be welded. The thickness of the plate in the figure 2.2 is 1.65mm. As shown in figure 2.1, the left part of the plate is fixed in the clamped area. The right part of the plate, on the other hand, can move up from the fixture. The clamping force of the plate, comes from the bolt force and the weight of the upper part of the fixture.

![Figure 2.1: Fixture setup](image1)

![Figure 2.2: Plate](image2)
Chapter 3

Methods

In this chapter, the methods that are going to be used in order to construct a simplification of the fixture problem in FE welding simulation are going to be presented.

3.1 Definitions of functions in MSC.Marc

- **MSC.Marc**: FEM program that is used in this master thesis.

- **Table driven input**: Almost all inputs into MSC.Marc can be *table driven*. That means that the input value (reference value) is multiplied by the value of the table. The table can be a function of several different variables, such as time, displacement, current coordinates and so on. For example, if a boundary condition of the type *point load* is used with an input value in the $x$–*direction* = 1. That means that the point load is 1 in x-direction. If the table 3.1 is used as an input table, the point load will be the *point load reference value* (which is often set to 1) multiplied by the *multiplication value* at each time step. That means that the point load will change it’s value with time according to table 3.1 and figure 3.1. For example after 1 time unit, the multiplication value is 50 according to table 3.1 and figure 3.1, this multiplied with the reference value 1 gives the point load 50N.
Table 3.1: Example of a table for the table driven input

<table>
<thead>
<tr>
<th>Time</th>
<th>Multiplication value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>150</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
</tr>
<tr>
<td>5</td>
<td>650</td>
</tr>
</tbody>
</table>

Figure 3.1: The figure shows the table plot from the table driven input
Table 3.2: The point load at each time step with the point load reference value set to 1

<table>
<thead>
<tr>
<th>Time</th>
<th>Point load</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>150</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
</tr>
<tr>
<td>5</td>
<td>650</td>
</tr>
</tbody>
</table>

3.2 Modelling of contact

The analyze of contact is complex, due to the fact that the position and the motion of the contact body must be determined with a very high accuracy to avoid that the contact bodies move through each other.

There are two types of contact bodies in MSC.MARC,

- **Deformable contact body** is a set of elements that act like a body. The deformable contact body can be in contact with other deformable bodies or ridged bodies.

- **Ridged contact body** is a set of curves or surfaces that acts as a body. The ridged body doesn’t deform.

3.2.1 Shell contact

Shell elements can be in contact with ridged bodies and deformable bodies. A node of a shell element is said to be in contact with a segment, when the position of the shell element node ± half the thickness of the shell element, normal to the shell are within another segment.

3.2.2 Detection of contact

Each potential contact node is checked, if near contact is possible with a segment. In 3-D, faces of elements are possible contact segment. MSC.Marc uses a near contact algorithm to determine if contact nodes are near a contact segment. If a node is in near contact, a more sophisticated algorithm is used to detect contact.

3.2.3 Contact tolerance

It is unlikely, that exact contact occurs, since the position of a node is a numerical value and there are always round-off and truncation errors involved. That is why contact tolerance is needed. Contact tolerance is the measurement of when contact can be said to occur. By default, the contact tolerance
is set to 25% of the shell thickness. If a value of the contact tolerance is too large, it will cause too many nodes to be in contact. If too small values are chosen, it will lead to more computation time.

### 3.2.4 Contact between deformable bodies

When a 3-D deformable contact body is created, the program creates a surface to outline the boundary of the contact body. When a node of a deformable contact body comes in contact with a segment, a tie is created.

### 3.2.5 Penetration

To avoid penetration between elements, MSC.Marc has three procedures to detect and avoid penetration. The default and the used procedure, uses increment splitting. In this procedure, the time step which causes penetration $\Delta t$ is divided into two subincrements. In the first subincrement, a node $n_1$ is not constrained and in the second subincrement the node is constrained, then MSC.Marc has to find the time when contact first occurs, see fig. 3.2. The time step $\Delta t_a$ and $\Delta t_b$ is chosen by linearizing the displacement increment.

$$\Delta t_a = \frac{a}{a + b} \Delta t$$

$$\Delta t_b = \frac{b}{a + b} \Delta t$$

(3.1)

That means that $\Delta t_a$ is the time when contact first occurs.

### 3.2.6 Heat flux due to contact between deformable bodies

![Diagram illustrating increment splitting procedure](image)

Figure 3.2: Illustration that shows how the increment splitting procedure works.
When deformable bodies are created, heat flux films are automatically created at all the boundaries. The fluxes used in MSC.Marc depends on the distance \( d \) between the contact bodies. There are two different distances that are used to calculate the heat flux. The first is the contact distance \( d_{\text{con}} \), when the bodies are said to be in contact according to the contact tolerance. The second is the near contact distance \( d_{\text{near}} \), when the distance between elements is smaller or equal to the smallest element. For distance \( d < d_{\text{con}} \) the heat flux is defined as[4]

\[
q = H_{TC} (T_2 - T_1)
\]

where
- \( q \) = heat flux
- \( H_{TC} \) = film coefficient
- \( T_1 \) = the surface temperature
- \( T_2 \) = the interpolated nodal temperature at contact location on the contacted body.

In the near contact case, the heat flux is defined by

\[
q = H_{CV} (T_2 - T_1) + H_{NC} (T_2 - T_1)^{B_{CN}} + \sigma \epsilon f (T_{A2}^4 - T_{A1}^4) + \left( H_{CT} \left( 1 - \frac{d}{d_{\text{near}}} \right) + H_{BL} \left( \frac{d}{d_{\text{near}}} \right) \right) (T_2 - T_1)
\]

where
- \( q \) = heat flux
- \( H_{CV} \) = convection coefficient for near field behaviour
- \( H_{NC} \) = natural convection for near field behaviour
- \( B_{CN} \) = exponent associated with natural convection
- \( \sigma \) = Stefan-Boltzman constant
- \( \epsilon \) = emissivity
- \( H_{BL} \) = distance dependent heat transfer coefficient.

If the distance \( d > d_{\text{near}} \), the heat convection to the environment is defined by

\[
q = H_{CTVE} (T_2 - T_{\text{SINK}}) + \sigma \epsilon f (T_{A2}^4 - T_{A1}^4)
\]

where
- \( q \) = Heat flux
- \( H_{CTVE} \) = heat transfer coefficient to the environment
- \( T_{\text{SINK}} \) = environment sink temperature.

If the near distance \( d_{\text{near}} \) is not defined in the thermal part of the contact option, and \( d > d_{\text{contact}} \), eq.3.4 is still valid.
3.3 Modelling of friction and support without contact

There are two possible ways to model friction and support without contact. The first is the spring option, the second is the elastic foundation option. Both options can be modelled with both linear and non-linear properties.

3.3.1 Springs

The spring option can be used in both mechanical and thermal analysis. During coupled thermo-mechanical analysis, the thermal part of the spring acts as a link. The thermal link is controlled with a heat transfer coefficient. The springs in MSC.MARC can be defined in three ways. All these spring options need a beginning and an end node. It is important to remember to apply fixed boundary condition to the begin node, in other case the spring will just float in space.

- **Fixed degree of freedom** The spring force is only active in one degree of freedom between the beginning and the end node that defines the spring.

- **To ground** The spring force is directed to origo.

- **True direction** The spring force is active in the direction of the line between the two nodes which define the spring.

The spring force $F_s$ is defined by

$$F_s = k(U_1 - U_2)$$

where $k$ is the spring stiffness and $U$ is the displacement.

3.3.2 Elastic foundation

Elastic foundations work like springs, but they don’t operate on nodes like springs.

**Edge foundation**

The edge foundations, as the name says, operate on the edges of the element. The foundation force is defined by

$$F_f = k(U_1 - U_2)$$

where $F_f$ is the foundation force and $k$ is the foundation stiffness. $U_1 - U_2$ is the displacement of the edge.
Face foundation

The face foundation works like the edge foundation, but operates on the faces of the elements instead of on the edges. The foundation force is defined like eq. 3.6, but then $U_1 - U_2$ is the displacement of the faces of the element.

3.3.3 Face film

In order to simulate the heat transfer to the fixture and the environment, the function face film can be used. Face film was presented in section 3.2.6.
Chapter 4

Modelling and solution

In this chapter, the reference and the simplified models are going to be presented.

4.1 The models

The reference model contains both the fixture and the work piece, and gives a guideline of how the simplified model should work. In the simplified model, the fixture is excluded, the fixture is replaced with boundary conditions and non-linear springs.

4.1.1 The reference model

The reference model contains the fixture, modelled with solid elements, and the work piece, modelled with shell elements of type 75, see figure 4.1.

To be able to get the friction forces into the model, contact between the fixture and the work piece is needed. Contact is a very CPU-time consuming operation, due to the fact that in every operation, the contact status in every node must be determined. That is why there is of interest to find an alternative way to model large systems, where friction might occur.

The plate and the fixture are of contact body type deformable, with a friction coefficient of the plate set to be 0.3 and zero for the fixture.

The bolts are modelled as square rods from the upper part of the fixture, penetrating the lower part of the fixture through square holes. The bolt force is modelled as point loads in negative y-direction at the bottom nodes of the bolt.

4.1.2 The simplified model

In the simplified model, the fixture is replaced with springs and elastic foundations. The work piece is, like the reference model, modelled with shell elements of type 75.
4.2 First step, friction test

The first test that was done, was a test of the friction force in the reference model. That was done with a fix displacement in the x-direction, see fig 4.1. The same thing was done to the simplified model, where the friction was modelled with edge foundation\(^1\). To get a similar stress field as in the reference model, two things had to be satisfied.

- The sum of the foundation forces had to be equal to the friction force in the reference model. Eq. 4.1.

- The stiffness of the edge foundation had to have a distribution with the same shape as the contact force of the reference model.

\(^1\)Foundation is presented in 3.3.2
\[
\sum_{i=1}^{n} (F_f)_i = F_{fri}
\]  
(4.1)

where \((F_f)_i\) is the foundation force of the \(i\)th edge, and \(F_{fri}\) is the friction force. The friction force when sliding is defined by

\[
F_{fri} = \mu F_n \frac{2}{\pi} \arctan \left( \frac{v_r}{v_{th}} \right),
\]  
(4.2)

where \(v_r\) is the sliding velocity and \(v_{th}\) is 1-10\% of \(v_r\). \(\mu\) is the friction coefficient and \(F_n\) is the normal force. The force from the foundation is defined like

\[
F_f = kx
\]  
(4.3)

where \(F_f\) is the foundation force, \(k\) is the foundation stiffness, and \(x\) is the displacement. It is obvious that the foundation force will increase with the displacement, and the friction force will be constant. Therefore, the stiffness \(k\) must be a function of the displacement \(x\) i.e. \(k = k(x)\).

\[
kx = c
\]  
(4.4)

\[
\Rightarrow k(x) = k \frac{1}{x}.
\]  
(4.5)

Then the foundation force is constant for all \(x\)

\[
F_f = k(x)x = k \frac{1}{x}x = k.
\]  
(4.6)

The foundation stiffness is distributed like the contact force distribution. The contact force is

\[
F_c = F_a + mg,
\]  
(4.7)

where \(F_c\) is the contact force and \(F_a\) is the applied force to the bolts in the fixture. \(m\) is the mass of the upper part of the fixture, and \(g\) is the gravity. The contact force \(F_c\) and the normal force \(F_n\) is the same, \(F_c = F_n\). So, eq. 4.1 becomes

\[
\sum_{i=1}^{n} k_i = \mu (F_a + mg) \frac{2}{\pi} \arctan \left( \frac{v_r}{v_{th}} \right).
\]  
(4.8)

### 4.2.1 Contact force distribution

If the reference model is known, that means that if simulations are made on a full scale, the distribution function is easy to study. If no reference model is made, one can use Buckingham’s Pi Theorem to get a distribution function.
Distribution function, using polynomial fitting

In this case, the reference model must be known. The distribution function is then constructed using the contact force distribution from the reference model. The method of how to construct the distribution function using polyfit is shown below.

- The numerical values of the contact force are exported to Matlab.
- The values are fitted to a polynomial.
- The polynomial is evaluated at the points where the edge foundations are to be applied.
- The sum of foundation stiffness must be equal to the friction force.

An example of a Matlab code that constructs the distribution function using polyfit is presented in appendix D.
Buckingham’s Pi Theorem

As a start of the dimension analysis, the quantities that the physical law depends on must be identified. The law must depend on the force $F$, the distance from the middle of the plate $d$ and Young’s modulus $E$. Then, if there exist a physical law, that depends on these quantities, i.e. $f(F, d, E) = 0$, then Buckingham’s Pi theorem states that there is an equivalent law. That is expressed by dimensionless quantities, like

$$f \left( F^\alpha E^\beta d^\gamma \right) = 0$$  \hspace{1cm} (4.9)

$$f \left( \frac{ML}{T^2} \right)^\alpha \left( \frac{M}{LT^2} \right)^\beta [L]^\gamma = 0.$$  \hspace{1cm} (4.10)

where the dimensions of the quantities are

$$F = \left[ \frac{ML}{T^2} \right]$$

$$E = \left[ \frac{M}{LT^2} \right]$$

$$d = [L].$$

(4.11)

If the law exists, then the dimensions on each fundamental unit must be the same. From eq.4.9 it is easy to see that

$$f \left( F^\alpha E^\beta d^\gamma \right) = 0 \Leftrightarrow$$  \hspace{1cm} (4.12)

$$f \left( \left[ \frac{ML}{T^2} \right]^\alpha \left[ \frac{M}{LT^2} \right]^\beta [L]^\gamma \right) = 0.$$  \hspace{1cm} (4.13)

(4.14)

Solving for each fundamental unit $M, L, T$ gives

$$\alpha + \beta = 0$$

$$\alpha - \beta + \gamma = 0$$

$$-2\alpha - 2\beta = 0,$$

then with $\gamma = 1$

$$\alpha + \beta = 0$$

$$\alpha - \beta + 1 = 0$$

$$-2\alpha - 2\beta = 0,$$
\[
\alpha = -\frac{1}{2} \\
\beta = \frac{1}{2} \\
\gamma = 1.
\]

Hence eq. 4.9 gives
\[
f\left(\sqrt{\frac{F}{E}d}\right) = 0. 
\]

(4.18)

Then
\[
f\left(\sqrt{\frac{F}{E}d}\right) = 0 
\]

(4.19)

It can be seen that \(\sqrt{\frac{F}{E}d} = C\) where C is a constant

(4.20)

or, \(F = \frac{Ed^2}{C_1}\)

(4.21)

The constant \(C_1\) is easy to determine, using the fact that \(F\) must be equal to the applied bolt force at a distance \(d = D\), where \(D\) is the distance from the middle of the plate to the bolt, see figure 4.2.

![Diagram](image)

**Figure 4.2: Fixture seen from above**

The constant \(C_1\) is determined by \(D(=125 \text{ mm in fig. 4.2)}\) and \(F_{\text{bolt}}\). See eq. 4.22.
4.2.2 How to use the distribution function

- Evaluate the function at all the nodes of the boundary where the edge foundation is to be applied.

- Weight the sum of all foundations, so that eq. 4.1 is satisfied.

A Matlab script that constructs the distribution function, using Buckingham’s Pi theorem is presented in appendix D.

Figure 4.3: Edge foundation, acting like friction force
4.2.3 Implementation and result of the first test using polynomial fitting

As shown in figure 4.4, the shape of the curves are very similar. The max and min values are presented in table 4.1

![Graph showing stress plots](image)

Figure 4.4: The graphs shows the stress plots of the reference- and foundation model of the 20N test. The x-axis of the plot shows the coordinate 0, 1, 2, . . ., 190mm along the weld path according to fig. 4.2. The y-axis of the plot shows the stress.

<table>
<thead>
<tr>
<th>Method</th>
<th>Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>0.1893Mpa</td>
<td>0.0404Mpa</td>
</tr>
<tr>
<td>Foundation</td>
<td>0.1847Mpa</td>
<td>0.0437Mpa</td>
</tr>
</tbody>
</table>

Table 4.1: Max and min values of the 20N test with polyfit method

\(^{2}\)Full-size plots in appendix A
The shape of the graphs of fig. 4.5 are also similar, but there are a larger difference between the max and min values. The values are presented in table 4.2

![Graph showing stress plots with x-axis coordinate range 0 to 190mm and y-axis values]

Figure 4.5: The graphs shows the stress plots of the reference and foundation model of the 60N test. The x-axis of the plot shows the coordinate 0,1,2,...,190mm along the weld path according to fig. 4.2. The y-axis of the plot shows the stress.

Table 4.2: Max and min values of the 60N test with polyfit method

<table>
<thead>
<tr>
<th>Method</th>
<th>Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>0.2525Mpa</td>
<td>0.0722Mpa</td>
</tr>
<tr>
<td>Foundation</td>
<td>0.2968Mpa</td>
<td>0.0703Mpa</td>
</tr>
</tbody>
</table>
4.2.4 Implementation and result of the first test using Buckingham’s Pi theorem

The shapes of the curves in fig. 4.6 are similar. The min and max values are presented in table 4.3. The curves of the 60N test are shown in fig. 4.7, and the max and min values are presented in table 4.4

![Graph](image)

Figure 4.6: The graphs shows the stress plots of the reference and foundation model of the 20N test. The x-axis of the plot shows the coordinate 0, 1, 2, . . . , 190mm along the weld path according to fig. 4.2. The y-axis of the plot shows the stress. The distribution function of the foundation model is constructed using Buckingham’s Pi Theorem.

<table>
<thead>
<tr>
<th>Method</th>
<th>Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>0.1893Mpa</td>
<td>0.0404Mpa</td>
</tr>
<tr>
<td>Foundation</td>
<td>0.1909Mpa</td>
<td>0.0467Mpa</td>
</tr>
</tbody>
</table>
Figure 4.7: The graphs show the stress plots of the reference and foundation model of the 60N test. The x-axis of the plot shows the coordinate 0, 1, 2, ..., 190mm along the weld path according to fig. 4.2. The y-axis of the plot shows the stress. The distribution function of the foundation model is constructed using Buckingham’s Pi Theorem.

Table 4.4: Max and min values of the 60N test with Buckingham’s pi theorem

<table>
<thead>
<tr>
<th>Method</th>
<th>Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>0.2525Mpa</td>
<td>0.0722Mpa</td>
</tr>
<tr>
<td>Foundation</td>
<td>0.3086Mpa</td>
<td>0.074Mpa</td>
</tr>
</tbody>
</table>

4.2.5 How to use results of the first test

The stiffness of the foundations in the first test, was controlled as a function of time (table driven input). This can not be done during a welding simulation, due to the fact that the displacement is not known at the start of the simulation. To be able to use the results from the first test, the stiffness of the foundation must be controlled as a function of the displacement. MSC MARC offers several different table types, which can be used to control the displacement, for example as a function of the x-coordinate. However, when this report was written, the function x-coordinate did not make any difference between -x and x. This means that MARC did not make any difference between displacement in positive and negative x-direction. In this case, it makes no difference since the friction is the same in all directions.
4.3 Second step

There are two ways of how the lower part of the fixture can be represented. The first way is to replace the fixture with face foundations. From the conclusion from the first test, foundation can’t be used since it is impossible to control the stiffness of the foundation as a function of displacement. The second way is to use springs.

The spring alternative (see figure 4.8) on the other hand, can be controlled as a function of general displacement, which makes difference between positive and negative displacement. The table that controls the spring stiffness is just a table that is zero if the displacement is negative and $10^9$ if the displacement is positive. The sign convention in MARC seems to be reversed in this case. Since the work piece should not be able to move in negative y-direction, i.e. with a logical argument, the table should be reversed.
Springs with fix degree of freedom

Plate

Springs with fix degree of freedom

Figure 4.8: The support from the fixture modelled with spring
4.4 Welding simulations

4.4.1 Welding simulation setup

First step

The first step is to construct the welding model. That is done by separating
the two pieces $\approx 0.1mm$ apart. Then the weld-path is created between the
two plate half’s. For the welding setup see fig. 4.10

Tack weld

In the same way as in real welding, tack welds are used in these welding sim-
ulations. The tack welds are modelled by fixed temperature above melting
temperature at the nodes where the tack are to be applied. The tack welds
are shown in fig. 4.9.

Figure 4.9: Tack welds, modelled with fixed temperature at the nodes shown
in the figure

4.4.2 The first welding simulation

During the first welding simulation a problem with the welding source code
was detected. The code does unfortunately not support “new style table
input”, which must be used in order to use foundation in the same way as
in the first test. During the rest of the simulations, the foundation was not
controlled by any function. That is not correct, but it was the only way to
continue.

4.4.3 Face film together with elastic foundation

In order to simulate radiation behaviour, face film can be used. But Face
film and elastic foundation can not be used together. The heat transfer to
the fixture can still be simulated, using the thermal part of the springs that
is used as support of the plate. If the begin node of the springs has a fixed
Figure 4.10: The edge foundation in the figure, represents the friction force temperature, like a sink point, then some of the heat transfer to the fixture is taken care of. The reason why Face film and elastic foundation do not work together, is most likely that the MSC.Marc source code has a bug. In the new release of MSC.Marc, r3, the problem does not occur. However, MSC.Marc r3 requires New style table input to be used in the face film option. This is not supported by the welding code developed at Volvo Aero Corporation.

The test that was done, to check the new release, did not include any welding simulation. Only the tack welding simulation were done (the welding subroutine are not needed during tack welding).
Chapter 5

Conclusions

The first test shows that foundations can be used to model the friction force instead of modelling with contact. Both distribution function methods show good results. If the fixture is of a complex geometrical, then it is most likely easier to construct the entire fixture in the model and calculate the contact force between the fixture and the work piece. If the geometry is simple, then Buckingham’s Pi Theorem is a better choice. However, there are several problems that must be solved before this can be used in welding simulations. The first thing that must be solved, is that the edge- and the face foundation must be controlled as a function of displacement (table driven input). In the friction case, the displacement must be in the sliding direction and in the support case it must be in the direction normal to the contact plane.

In order to model fixture support with face foundation, the foundation stiffness must be controlled as a function of the displacement (table driven input) in order to get different stiffness in positive and negative direction. MSC.Marc does not distinguish between positive and negative direction. That means that MSC.Marc states that $\delta = -\delta$ instead of $\delta \neq -\delta$.

The second thing that must be solved, is to make it possible to enter "new style table input" into the welding subroutine. The "new style table input" is not supported by the welding subroutine today.

The third thing, is the Face film and the Elastic foundation interaction. This problem seems to be solved in the new release of MSC.Marc r3. But the problem of New style table input remains, since r3 demands that New style table input is used.

If these things are solved, welding simulations with foundations acting as friction force and fixture support are possible to make.

5.1 Errors in the simplified model

The stiffness of the springs that represents the fixture is not as stiff as the true fixture. It is not likely that this causes any larger errors in the model.
The distribution function of the edge foundation shows some error at higher bolt forces, this is most likely caused by the fact that contact- and friction modelling is more complicated then the distribution model can handle at this time.

5.2 Recommendations

My recommendations are that Volvo Aero Corporation continues to work with the following things:

- The most important thing is to solve the *New style table input* problem. In order to model friction with edge foundations, the foundation stiffness must be controlled by a table (*table driven input*) since the stiffness must be constant and therefore independent of the displacement. The table type that is needed requires *New style table input*, which does not work with the welding subroutine. The *Face film* option also requires the *New style table input* in the new release of MSC.Marc r3.

- The problem with the lack of sign convention in the table styles $x$-, $y$- and $z$ coordinate in MSC.Marc. When face foundations are used to model fixture support, MSC.Marc does not make any difference between positiv and negativ displacement. This means that the face foundation have the same siffness in both directions and it is not possible to use the face foundation in the setup shown in fig. 2.1, since the work piece is only fixed at one side and can move upwards on the other side.

- Continue the work with the distribution function. Develop a more sophisticated distribution function, to make it possible to handle more complex geometries of the fixture.
Appendix A

Full size plots

Here the full size plots of the first test is shown.
Figure A.1: The graphs shows the stress plots of the reference- and foundation model of the 20N test. The x-axis of the plot shows the coordinate 0, 1, 2, . . . , 190mm along the weld path according to fig. 4.2. The y-axis of the plot shows the stress.
Figure A.2: The graphs shows the stress plots of the reference- and foundation model of the 60N test. The x-axis of the plot shows the coordinate 0,1,2,...,190mm along the weld path according to fig. 4.2. The y-axis of the plot shows the stress.
Figure A.3: The graphs show the stress plots of the reference- and foundation model of the 60N test. The x-axis of the plot shows the coordinate 0, 1, 2, …, 190mm along the weld path according to fig. 4.2. The y-axis of the plot shows the stress. The distribution function of the foundation model is constructed using Buckingham’s Pi Theorem.
Figure A.4: The graphs shows the stress plots of the reference- and foundation model of the 60N test. The x-axis of the plot shows the coordinate 0,1,2,...,190mm along the weld path according to fig. 4.2. The y-axis of the plot shows the stress. The distribution function of the foundation model is constructed using Buckingham’s Pi Theorem.
Appendix B

Element type 75

B.1 Properties
Element of type 75 [5] is a thick-shell element with four nodes, with displacement and rotation as degrees of freedom. Elements 75 uses bilinear interpolation.

B.2 Bilinear interpolation
Bilinear interpolation is a linear interpolation of a function of two variables.
Appendix C

Element type 7

Elements of type 7 [5] is a eight node isoparametric hexahedral. Element 7 uses trilinear interpolation. This element is preferred to higher order elements in contact analysis.
Appendix D

Matlab code

D.1 Distribution function construction using Buckingham's Pi Theorem

function b = f(x,n)
format long
E=2e5; %youngs moduluse
x=0;
x0=0;
fs=20; %applied force at the bulits
m=9.348; %mass
g=9.82; %gravation
const=(E*(15625))/fs; %constant, determine using
%boundary condition f=fs@d=d_bult=125mm

ft=0.3; %friction number
v=5; %displacement velocity mm/s
nk=(m*g)+2*fs; %normal force
fg=(ft*nk*2/pi)*atan(v/(0.01*v)); %friction force, when gilding
f=((E*x*x)/const); % dist. function
vec_x=zeros(95,1);
vec_f=zeros(95,1);

% foundation stiffness
for n=0:95
    x0=x0+n;
    x=x0;
    f=((E*x*x)/const)+const_2;
    vec_f(96-n,1)=f;
    vec_x(96-n,1)=n;
end

h
%% End foundation stiffness
vec_x;
sum(vec_f)
vec_x=vec_x*m*g
fg
vec_f;
vec_f=vec_f*(fg/sum(vec_f))
sum(vec_f)
plot(vec_x,vec_f);
vec_f=vec_f/2;

D.2 Distribution function construction using numerical polyfit

function foundationny = fc(x,y,n)
fs=60; %force at the bolts
m=9.348; %mass
g=9.82; %gravity
ft=0.3; %friction coefficient
v=5; %sliding velocity
nk=(m*fs+nk*2/2); %normal force
fg=(ft*nk/2)*atan(v/(0.01*v)); %friction force when sliding
fg
x=[1.25000000000000e+01
 2.50000000000000e+01
 3.75000000000000e+01
 5.00000000000000e+01
 6.25000000000000e+01
 7.50000000000000e+01
 8.75000000000000e+01
 1.00000000000000e+02
 1.12500000000000e+02
 1.25000000000000e+02
 1.37500000000000e+02
 1.50000000000000e+02
 1.62500000000000e+02
 1.75000000000000e+02
 1.87500000000000e+02]; %coord. of normal force

y=[1.932257771492e+00
 8.071396946907e-01
 2.853528559208e-01
 7.140973955393e-02

```
0.0000000000000e+00  
0.0000000000000e+00  
0.0000000000000e+00  
0.0000000000000e+00  
0.0000000000000e+00  
0.0000000000000e+00  
7.140976190567e-02  
2.85352859208e-01  
8.071396350861e-01  
1.932257890701e+00];
%normal force, from ref. mod.  
x1=zeros(191,1);  
a=0;  
for i=0:190  
%coord of foundation  
    x1(i+1,1)=a;  
    a=a+1;  
end  
format long  
con=polyfit(x,y,3);  
%fitting to poly of degree 3  
f = polyval(con,x1);  
a=zeros(95,1);  
null=0;  
for n=1:length(f)-1  
    if f(n)<=0 && f(n+1)>0  
        null=f(n+1);  
    end  
end  
for n=1:length(x1)  
    if f(n)<=0  
        f(n)=null;  
    end  
end  
sum(f');  
length(f);  
b=zeros(95,1);  
b(1:95)=f(1:95);  
div=fsum(b');  
b=b*div;  
%weight to have same sum as friction force  
b=b/2;  
%foundation are applied at both sides->must divide by 2.
Bibliography


[]