Incremental Sheet Forming: Modelling and Path Optimisation

Ankor M. Raithatha MEng (Oxon)

A thesis submitted to complete the
requirements for the degree:
Doctor of Philosophy
in Engineering
Science

Wadham College

University of Oxford

Michaelmas 2008

Abstract

Incremental sheet forming (ISF) is a novel metal shaping technology that is economically viable for low-volume manufacturing, customisation and rapid-prototyping. It uses a small tool that is controlled by a computer-numerically controlled sequence and the path taken by this tool over the sheet defines the product geometry. Little is currently known about how to design the tool-path to minimise geometric errors in the formed part. The work here addresses this problem by developing a model based tool-path optimisation scheme for ISF.

The key issue is how to generate an efficient model for ISF to use within a path optimisation routine, since current simulation methods are too slow. A proportion of this thesis is dedicated to evaluating the applicability of the rigid plastic assumption for this purpose. Three numerical models have been produced: one based on small strain deformation, one based on limit analysis theory and another that approximates the sheet to a network of rods.

All three models are formulated and solved as second-order cone programs (SOCP) and the limit analysis based model is the first demonstration of an upper-bound shell finite element (FE) problem solved as an SOCP. The models are significantly faster than commercially available FE software and simulations are compared with experimental and numerical data, from which it is shown the rigid plastic assumption is suitable for modelling deformation in ISF.

The numerical models are still too slow for the path optimisation scheme, so a novel linearised model based on the concept of spatial impulse responses is also formulated and used in an optimal control based tool-path optimisation scheme for producing axisymmetric products with ISF. Off-line and on-line versions of the scheme are implemented on an ISF machine and it is shown that geometric errors are significantly reduced when using the proposed method. This work provides a new structured framework for tool-path design in ISF and it is also a novel use of feedback to compensate for geometrical errors in ISF.

Statement

This thesis is the result of my own work that was conducted during the research period of this DPhil. No part of this dissertation has been or is currently being submitted for any other qualification at this or any other university.

Ankor Raithatha
Michaelmas 2008

Acknowledgements

Firstly, I am grateful to Stephen Duncan for supervising my project and providing his guidance and encouragement, especially during those periods when persistence was abundant but results were few and far between!

I also thank the members of the Production Processes Group in the engineering department at Cambridge University, for providing useful discussions and access to the single point incremental forming (SPIF) machine. These members include Julian Allwood, Omer Music and Kathryn Jackson.

During my visit to the Research and Advanced Engineering laboratories of Ford Motor Co., Dearborn, USA, I was made very welcome. I thank Carl Johnson at Ford for permitting me to visit his devision. Cedric Xia and Feng Ren of the Manufacturing Processes CAE Research Group conducted several SPIF simulations for me for which I am very grateful. Vijitha Kiridena and Cedric Xia arranged many enlightening discussions, tours and demonstrations during my visit, including a tour of a rapid prototyping plant, a tool manufacturing facility, a robotics manufacturer, and demonstrations of a virtual driving environment and super-plastic forming process.

At Oxford, Antonis Papachristodoulou, Athanasios Makrodimopoulos and Christopher Martin of the Department of Engineering Science provided useful discussions during some periods of this work. I am grateful for the opportunities I had to tutor Oxford undergraduate students at Keble, Wadham, Mansfield, St Hugh's and St Edmund Hall, through which my own understanding of engineering science was solidified.

I also acknowledge the financial support of my Departmental Scholarship, funded by the Engineering and Physical Sciences Research Council (EPSRC), without which this DPhil would not have been feasible. Further to this, the Royal Academy of Engineering (RAE) and my college, Wadham, facilitated a number of conference visits and an academic visit to Detroit, USA.

Last but certainly not least, my family, friends and fiancée Sonia have been invaluable sources of support throughout this work.

Publications

Parts of this thesis have been published or submitted for publication in the following journals and proceedings:

A. Raithatha, S. R. Duncan and J. Allwood. Tool-path optimisation for axisymmetric parts in incremental sheet forming. *IEEE Trans. Automation Science and Engineering*, submitted, 2009.

J. Allwood, O. Music, A. Raithatha and S. R. Duncan. Closed-loop feedback control of product properties in flexible metal forming processes with mobile tools. *CIRP Annals - Manufacturing Technology*, vol. 58, issue 1, pages 287-290, 2009.

A. Raithatha and S. R. Duncan. Rigid plastic model of incremental deformation using second-order cone programming. *Int. J. Numerical Methods in Engineering*, vol. 78, issue 8, pages 955-979, 2009.

A. Raithatha and S. R. Duncan. Deformation model for incremental forming using conic programming. In *Proc. Numisheet 2008*, Interlaken, Switzerland, pages 749-754, 2008.

A. Raithatha and S. R. Duncan. An improved finite element model of incremental forming using conic programming. In *Proc. 2008 American Control Conf.*, Seattle, WA, pages 5109-5114, 2008.

A. Raithatha, S. R. Duncan, K. Jackson and J. Allwood, Second order cone programming in modeling incremental deformation. In *Proc. 2007 American Control Conf.*, New York, NY, pages 4841-4846, 2007.

A. Raithatha, K. Jackson, S. Duncan and J. Allwood. New Method for Modeling Plastic Deformation in Incremental Sheet Forming. In *Proc. IEEE Conf. on Control Applications*, Munich, Germany, pages 1037-1042, 2006.

Contents

List of Figures

List of Tables

Abbreviations & Nomenclature

Δ	Area of a triangular finite element
ϵ, $\dot{\epsilon}$, $\dot{\bar{\epsilon}}$	material strains, strain rate, equivalent strain rate
κ	Curvature
Ω, Ω_e	Sheet volume, element volume
σ, $\bar{\sigma}$	material stresses, effective stress
σ_y/σ_o	yield stress
$\mathbf{T}$	Transformation matrix from global to local coordinate frames in FEM
$\mathbf{t}$, $T(x, y)$	Tool surface vector and tool surface function respectively
A	Area of a sheet
a	Shift variable for measuring relative error in path optimisation
h	Sheet thickness
J	Cost function in an optimal control problem
L	Lagrangian
M, m	Standard and normalised stress resultant moments
N, n	Standard and normalised stress resultant forces
NE	Number of elements
NN	Number of nodes
P	Domain of feasible control inputs
$p, \mathbf{p}$	Control inputs
r	Radius of the tool tip
R_{min}	Minimum radius of curvature in the sheet
s, $\mathbf{s}$	Experimentally obtained impulse response
u	Current deformation field of sheet
u, v	In-plane sheet velocities in limit analysis model
$u_{boundary}$	Subset of deformation field in contact with clamping frame

u_{tool}	Subset of deformation field in the region of the tool
v	Current rate of change of deformation, or sheet velocity, $\dot{u}$
w	Transverse velocity field
X, Y, Z	Coordinates in global cartesian frame in FEM
x, y, z	Coordinates in a cartesian frame - local coordinates in FEM
Abaqus	A commerically available software package for finite element analysis
AISF	Asymmetric incremental sheet forming
CMM	Coordinate measuring machine
CNC	Computer numerically controlled
CPU	A processor, or specifically Central Processing Unit
FD	Finite differences
FDM	Finite difference method
FEA	Finite element analysis
FEM	Finite element method/modelling
IfM	Institute for Manufacturing, Cambridge university
ISF	Incremental sheet forming
KE	Kinetic energy
LIF	Laser incremental forming
LMI	Linear Matrix Inequality
LP	Linear program
LS-Dyna	A commerically available software package for finite element analysis
Mosek	A commerically available software package for solving SOCPs
PC	Personal computer
PE	Potential energy
PI	Proportional and integral control
R&AE	Research and Advanced Engineering, Ford Motor Co., Deaborn, USA
RMS	Root mean square
SDP	Semi-definite program
SDPT3	A free software package for solving SOCPs and SDPs
SeDuMi	A free software package for solving SOCPs

SOCP	Second-order cone program
SPIF	Single-point incremental forming
TPBVP	Two-point boundary value problem
TPIF	Two-point incremental forming
TW	Traction work
Yalmip	A Matlab optimisation problem parser, stands for ‘Yet Another Linear Matrix Inequality Parser’

Chapter 1

Introduction

1.1 Review of ISF Processes

Incremental deformation is a process that applies a series of small deflections to a material that accumulate to produce a large final deformation. The procedure has been employed by sheet metal workers for centuries through traditional manually operated craftsman's tools such as the English Wheel [103]. The modern interest in incremental forming originates from a patent published in 1967 by Leszak [69], which described a technique for automating the process; an excerpt of this is shown in figure 1.1. Modifications to the method over the years have resulted in a range of novel manufacturing techniques that automate the incremental deformation process and all of these are collectively referred to as incremental sheet forming (ISF). Two newer adaptations of the process are described in patents held by Amino Corporation [88] and Luttgeharm [72].

Despite the date of the earliest patent, most published research in ISF is relatively recent and dates back less than two decades. For instance, Powell and Andrew's [95] description of incremental techniques for flanging dates to the early 1990's, as does an early publication by Kitazawa [65] that describes an ISF process using a modified CNC cutting machine. Several reviews of the current state of affairs have been published since then by Shima in 2001 [110], Hagan and Jeswiet in 2003 [45], Jackson in 2005 [57], Jeswiet *et al* also in 2005 [60] and most recently by Cao *et al* [27] in 2008. Of these [60] is the most comprehensive.

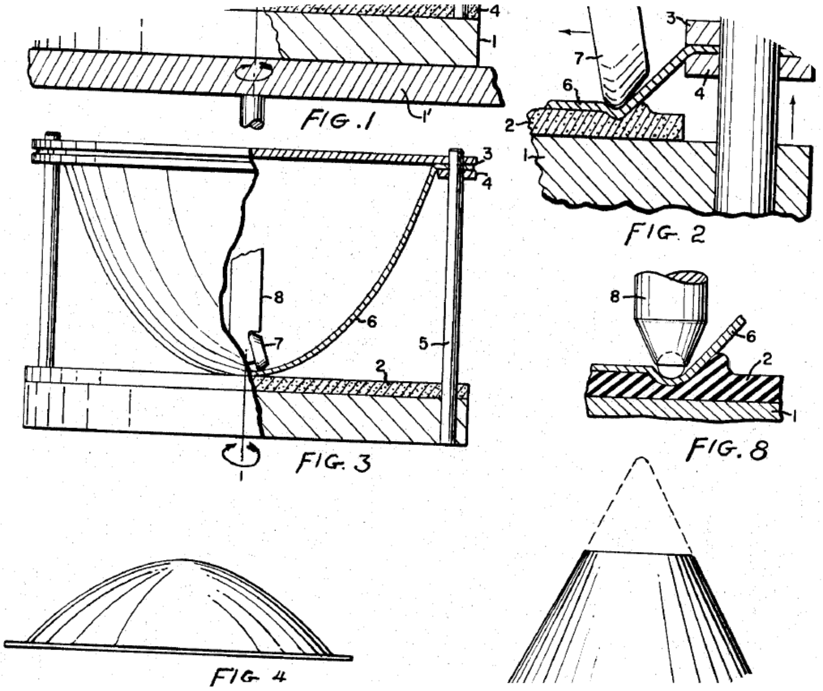


Figure 1.1: Excerpt from ISF patent published by Lezak [69] in 1967. Of particular interest are the alternative tool arrangements in *FIG. 2* and *FIG. 8* and the example product in *FIG. 4*. The tool in *FIG. 8* appears very similar to the one used on the Cambridge ISF machine [5].

1.1.1 Single-point incremental forming and variations

Several configurations of ISF are available that display various differences, for instance in terms of the machinery, the type and number of moving tools used, the number of fixed support tools and the mechanism for holding the work-piece in place. Often the latter is achieved by a blank that holds the edge of the sheet fixed, whereas in traditional manual incremental deformation methods, where the clamping ‘tool’ may even be a hand, the clamping arrangement is flexible and can be moved to allow the sheet to be formed up to the edges. The configuration determines the degrees of freedom available in the process and the complexity of the parts that can be produced. Many of the issues in this thesis apply to the general ISF forming process, but when specific data is required, data from the single point incremental forming (SPIF) machine housed in the Institute for Manufacturing at Cambridge University (IfM) is used [5].

Single point incremental forming

A single point incremental forming (SPIF) machine, such as the one at Cambridge, works by pressing a small indenter into the metal and drawing it over the surface to form a layer

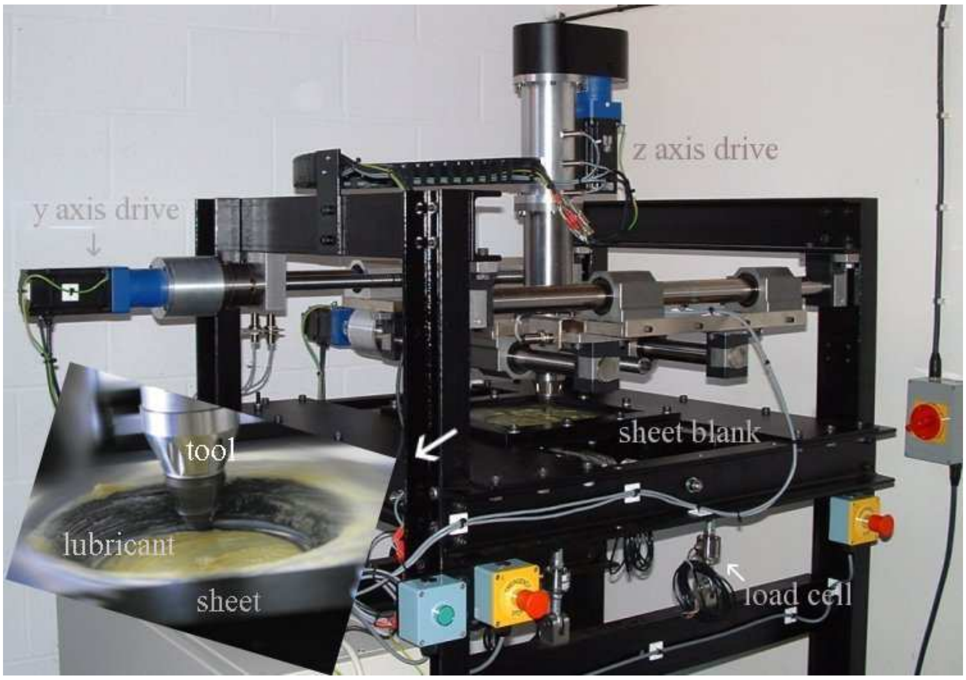


Figure 1.2: Image of the Cambridge ISF machine. Enlarged on the bottom left is an image of the indenting tool in contact with the sheet. The load cell labelled is one of six that are used to measure tool-sheet contact forces.

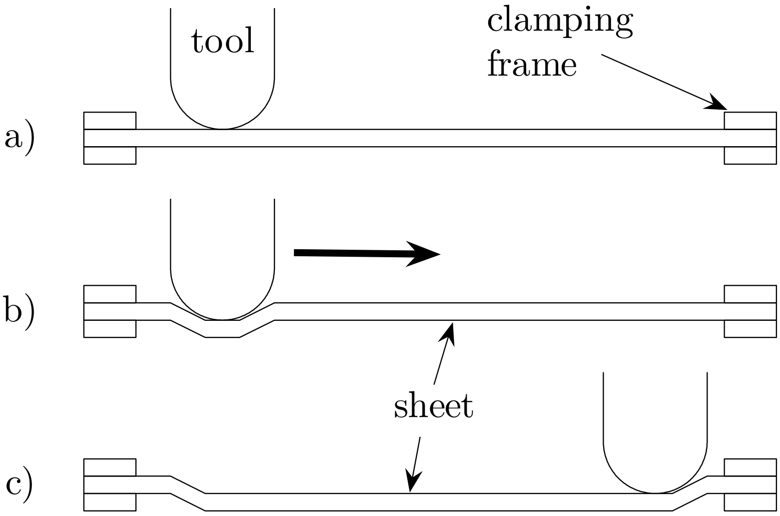


Figure 1.3: Schematic of the SPIF process.

of the desired product. The indenting tool can be seen in the photograph of the machine in figure 1.2. The tool path is applied by a computer numerically controlled (CNC) sequence that often follows the contour of the target product. A simple schematic of SPIF is displayed in figure 1.3 by the sequence a)-c), where the tool indents the sheet by a small amount and tracks along to form one pass over the sheet. After each individual pass, the indenter is pressed in further before tracing the next contour into the metal and so on until the full product is formed. Although SPIF is now the agreed label for this ISF process, it has previously been used interchangeably with the terms 'asymmetric incremental sheet forming' (AISF) and 'CNC incremental [sheet] forming' in literature [45, 11]. SPIF is considered to be the most general ISF technique since it uses no specialised tooling, such as costly dies or casts that would have to be specifically manufactured for a particular product.

There are several techniques available for measuring the deformation of the sheet. For the earlier deformation measurements presented in this project, the dimensions of parts formed by SPIF were laboriously measured using a manually operated coordinate measuring machine (CMM) at the end of the process, after the part was removed from the clamping arrangement. The more recent installation of a stereo-vision camera has greatly facilitated this task and it also allows measurements to be taken 'on-line' during the process. The arrangement with the camera in place is shown in figure 1.4. The camera has to be kept on the opposite side of the sheet to the tool because the surface of the sheet facing the camera currently has to be sprayed with a matt grey paint to avoid reflections and image saturation. Also, since SPIF is a cold forming process, a lubrication gel is applied on the side of the sheet that is worked by the tool to reduce friction induced heating, which is another reason why the bottom surface is imaged.

An example of a part formed on the Cambridge SPIF machine is given in figure 1.5. The product is shown after a post-forming stage where the ceramic based lubricant was removed from the product and the unformed regions of the sheet were cut away by a water jet cutter. It is theoretically possible to produce any convex shaped part by SPIF that can be formed by a conventional die and cast process, although careful design of the tool path is required for complex shapes [60].

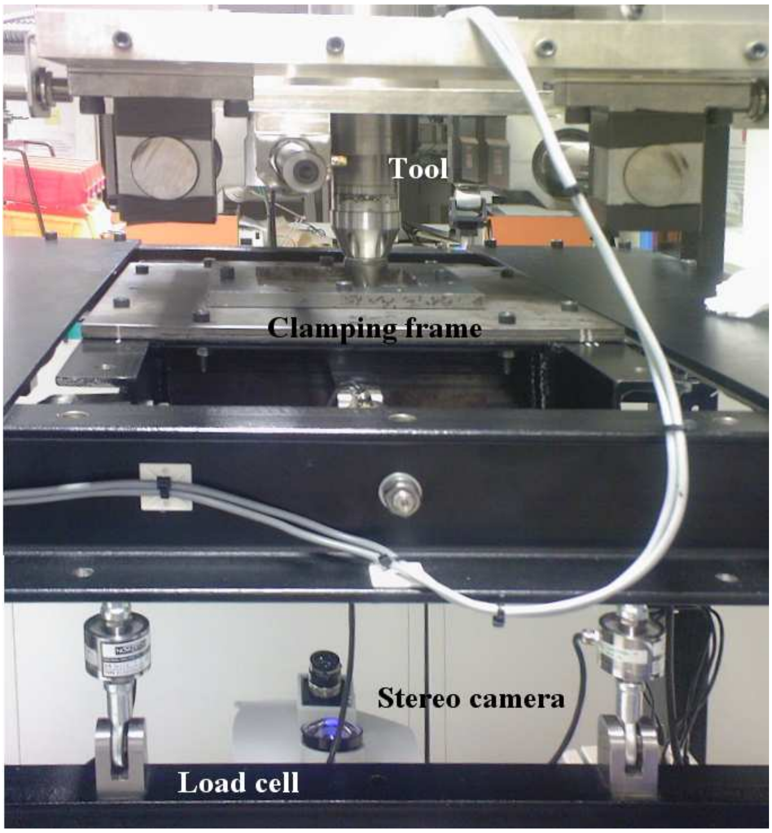


Figure 1.4: Side view of the ISF machine with a stereo vision camera in place.

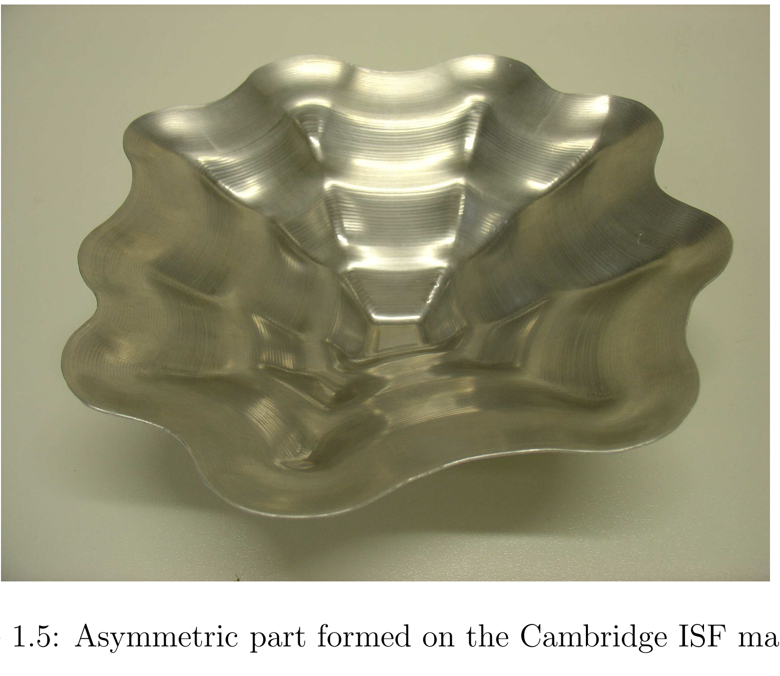


Figure 1.5: Asymmetric part formed on the Cambridge ISF machine.

Figure 1.6: English wheel manufactured by Irvan-Smith, Inc. [1].

Brief descriptions of four other variations of the ISF process are now provided. These processes are not explored in detail in this thesis, but a brief review of them is considered appropriate to provide a better overview of the field and to demonstrate the flexibility of ISF.

Two point incremental forming

In two point incremental forming (TPIF), the sheet metal is supported by a second tool, which can be fixed or moving and operates on the side opposing the primary tool. If the second tool is a moving tool, it is usually identical to the primary tool [77], whereas the fixed tool can come in several guises, such as a male or female die [72], a partial die or a support post. Full dies have been used to make parts with a high degree of geometric accuracy [28], but the production of such backing dies is costly, so more economical polymer dies made by rapid manufacturing processes have also been investigated [48]. However, the use of the specific tool removes a key attraction of ISF, since its flexibility is reduced by the need to manufacture a full die. As a compromise, a partial die that covers only the more complex regions of a product, or a support post that has no specialised shape [75], may perform a similar role for certain shapes.

There are several TPIF processes that are known as incremental curving [112, 119] and they can all be considered as an extension of the traditional ‘English wheel’ (see figure 1.6), or three-roll bending processes [84] for curving sheet metal. The English wheel remains popular with low volume sports car manufacturers that have resisted process automation,

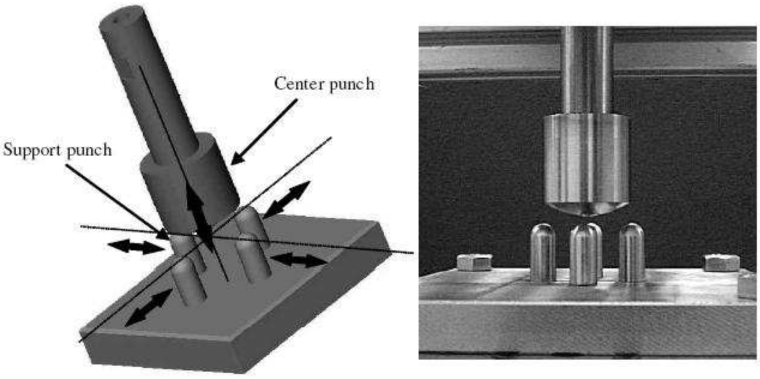


Figure 1.7: Punch set for incremental curving [119].

primarily in order to supply a niche market for unique hand-crafted products. During this process, a sheet of metal is held between the two wheels shown in figure 1.6 and rolled back and forth to incrementally curve the metal sheet. The method is highly flexible and allows the metal to be formed up to the edges. At the IfM in Cambridge, some researchers are analysing the mechanics of the English wheel and hope to either automate it or replicate its features in a new design of an ISF machine.

Other examples of incremental curving include Tanaka *et al*'s [112] process that uses a pair of elastic tools to repeatedly press the blank and Yoon *et al*'s [119] method, which sandwiches the metal between the moving upper tool and four fixed lower tools shown in figure 1.7. To model the process, an empirical relationship for the curvature of the sheet metal was produced by Yoon *et al*, which was used to modify inaccurate finite element simulations, although some circumstances were found where spring-back was too significant for accurate predictions [119]. The ability to form sheet metal up to the edges, which is not possible in SPIF because of the clamping frame, is beneficial in reducing the amount of waste material produced and this is a key attraction of incremental curving, which does not typically restrain the sheet over all the edges.

Incremental forming using water-jets

In order to improve surface finish and to remove the need for lubrication, investigations have been conducted on the application of high speed water-jets in place of a hard indenting tool [55, 62]. The process has been modelled and experimentally evaluated on a water-jet forming machine by Iseki [55]. It was found that in order to control the depth of

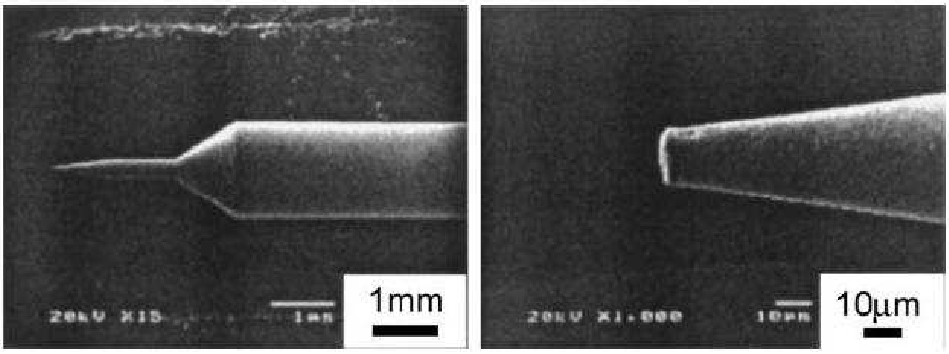


Figure 1.8: Microforming tool [106].

indentation, which is referred to as the bulging depth [55], the process could not be used without a support die on the underside of the tool. One could question the extent to which water-jet ISF differs from the more established hydro-forming [116], but comparatively, this process produces more localised deformation and although it needs a backing support, unlike hydro-forming a full backing die is not necessary.

Incremental hammering

In incremental hammering the moving tool oscillates at a high frequency with a small amplitude, which produces a noisy but effective hammering motion that reduces the tool-sheet reaction forces and places lower tolerances on the machine stiffness. Schäfer and Schraft have investigated its behaviour when the tool is controlled by a robot arm [107], where a high rigidity of the machinery is difficult to achieve. An ISF process that uses a robot arm is attractive because the robot arm is a non-specialised tool that removes the need to custom design an ISF machine. It is also possible to allow space for sensors to measure deformation in real-time and also, there may be ways to utilise the greater degrees of motion that a robot arm provides over basic CNC sequences. Hammering tools have been observed to give more localised deformation than standard tools in ISF, prompting some researchers to demonstrate their use in the incremental 'micro-forming' of foil [106]. The micro-forming tool, shown in figure 1.8, has a tip that is 10 μm in diameter and it is suggested in [106] that it can be used to produce parts for micro-structures or even micro-robots.

Laser incremental forming

Laser incremental forming (LIF) uses a laser to produce deformation through localised heating. Initial investigations into the process have involved an analysis of the effect of the tool path on the final deformed shape [90] and in this respect, LIF differs significantly from other ISF techniques because the relationship between the resulting shape and the tool-path taken is not intuitive. For example, in [90] a grid laser scanning path was found to produce an elliptical shell structure, whereas a radial path produces a spherical structure and a concentric path of uniform intensity was found to produce a saddle like structure, as opposed to the cone shape that would be obtained by other ISF techniques. LIF is also different in the sense that relative to other ISF techniques, the forming depth tends to be small. A laser forming system has recently been developed to produce very shallow parts of forming depths in the range of 1 cm [76] with a view to using it to remove distortion in parts made by other manufacturing processes [38].

More recently, an interesting combination of LIF with SPIF has been described by Duflou *et al* [33, 34], in which a laser is used to heat the region of the sheet near to the SPIF tool. The resulting deformation mechanism is more localised to the tool than for standard SPIF, with less material spring-back.

1.2 Developments and research focus in the ISF community

All of the ISF processes described above are very recent and they offer exciting directions for research. ISF holds promise as an economical method for making some components that are currently formed by existing manufacturing techniques, but it should be noted that ISF may develop to the point where it is able to produce components that are currently unfeasible and extend the scope of the process.

Allwood *et al* [6] state the importance of tracking advances within ISF and assessing applications in a well defined manner, so that important directions for development that would expand the range of applications can be identified. Their particular study is a structured analysis of 28 potential applications, all at UK based companies, where ISF could replace the current manufacturing practice. The paper builds on previous analysis based only on economical markers [46], such as the cost of machinery and tools, by factoring in the effects that switching to ISF could have on productivity, part quality, flexibility and speed. These factors give optimistic results, even more so than those based only on economical factors, and they have provided the main motivation for devoting this thesis, or specifically the period of research that it describes, to ISF.

Much of the existing research in ISF stems from the need for a better understanding of the limits of the process in order to:

1. assess the suitability of ISF for forming novel materials and manufacturing particular parts
2. explore methods for improving the current limitations in ISF

An example of point 1 is the work of Jackson *et al* [59] in assessing the applicability of ISF for forming sandwich panels. This also highlights the statement above regarding extending the scope of ISF, since sandwich panels are difficult to form by many existing forming technologies, yet they have been found to be compliant under ISF [59]. A further example is Houssain and Gao's [51] initial study on the forming of titanium sheets using ISF. For point 2, an interesting example is Allwood and Shouler's [7] new design of a high-frequency rotating 'paddle' shaped tool for ISF that can form sheet metal to greater limits without tearing than standard indenting tools can.

Much of the other current research in ISF focuses on either conducting experimental tests or developing computational models for ISF in order to understand the mechanics of the process better. A review of the performance of commercially available modelling software packages when applied to ISF is provided in §1.5. However, before that the following two sections present some background on the research path taken in this thesis, which is concerned mainly with point 2 above, where the process limitation of interest is the dimensional accuracy of parts formed by ISF.

1.3 Problems with ISF

Currently, the economic advantages and the increased flexibility of ISF are coupled with a compromise in geometric accuracy when compared to processes such as stamping. The removal of geometric errors in ISF is considered to be an important requirement if ISF is to compete effectively in industry against existing higher precision manufacturing techniques. One of the questions that remains to be resolved is:

> *To what extent can the choice of tool-path affect the geometric accuracy of parts manufactured by ISF, and how can the tool-path trajectory be designed to improve the geometric accuracy?*

The design of the tool-path is challenging and little has been done so far on the subject. The current practice is to simply use a 'contour following tool-path' that follows the shape of the target product by tracking out contours of the product layer by layer into the sheet. However, there are several reasons why a contour following tool path may be sub-optimal, such as:

- excessive stretching of the SPIF machine
- elastic spring-back in the sheet metal
- unpredictability of the final plastic deformation due to non-linear and non-localised deformation
- excessive localised stresses that could result in tearing or weak-spots
- unidirectional actuation in the z axis (the vertical axis defined in figure 1.2), assuming there are no tools or support posts on the underside of the sheet to push in the opposite direction to the primary tool

The aim of this project is to improve upon existing knowledge in tool-path design by demonstrating how trajectory optimisation techniques, such as optimal control, can be used to optimise the tool-path. In recent years, some other researchers have also begun to address the issue of tool-path design, but mostly by suggesting trial-and-error based methods. For instance, in [18] Attanasio *et al* tested how the size of the indentation step between layered passes of the tool, using a contour following tool-path, affects the surface quality and part geometry in a TPIF process using a full die. The test product, which was axisymmetric, was formed with a number of different step sizes and the experimental data was used to suggest that the 'optimal' tool-path should take the smallest indentation steps possible.

Mori *et al* [83, 81] were among the first ISF researchers to use numerical optimisation algorithms in tool-path design. In their work, a genetic algorithm was used to optimise the sequence of blows from the hammering tool, but the underlying process model used in the optimisation routine was very unrealistic and the final results were far from convincing, because the method was applied to a tool-path that consisted of merely three blows. More recently, another computer based tool-path optimisation method was described by Bambach *et al* [19], aimed at improving the sheet thickness distribution rather than the part geometry. As with Mori *et al*'s method [83], process modelling for the numerical optimisation routine was a key issue. Finite element (FE) models are too slow to place inside an optimisation routine, so instead Mori *et al* used simple polynomial equations with experimentally measured coefficients to predict the change in curvature over the sheet after a blow from the tool. Bambach *et al*'s method was more effective; they built a database from 162 finite element (FE) simulations and produced a 'metamodel' using linear interpolation between the data. This method was applied to axisymmetric products formed in four layers and it gave encouraging results, suggesting that the same method could be extended to a more realistic product, which is typically formed in over thirty layered passes of the tool. In that case however, it may take a very long time to produce the database for the meta-model. Ambrogio *et al* [11] avoided modelling by simply assuming what the ideal deformation shape would be at intermediate stages of ISF for their test part, a truncated pyramid, but the applicability of their method to general shapes is limited. Filice *et al* [41] have worked on the control of sheet thickness distributions by using tool-force measurements to modify the ISF parameters in real-time, to prevent excessive sheet thinning or tearing.

A review of the literature shows that research into the tool-path trajectory optimisation problem is very underdeveloped, especially for the correction of the product geometry. Furthermore, nobody has published results where their numerically optimised tool-paths have been applied to a full ISF product, preferring to test their methods on smaller simplified shapes instead. The aim in this thesis is to compute an optimal tool-path by solving a path planning algorithm off-line and then to implement it on the Cambridge ISF machine. This method is then combined with the sensing capabilities built into the machine as a feedback signal, thus demonstrating the first use of feedback in regulating geometry in ISF. In this sense, the Cambridge SPIF machine is unique because its design incorporates load cells for measuring tool forces and a stereo-vision camera for measuring sheet deformation, both of which may be used in a feedback loop. SPIF machines created by adapting CNC cutting machines do not have these capabilities, with access to the underside of the sheet for deformation sensors being a particular issue. The following section (§1.4) provides an overview of optimal control theory since, in this thesis, the path-planning problem is addressed under the optimal control framework. Next, an overview of existing techniques for modelling ISF is presented in §1.5, before concluding the chapter with a report on the research contributions made in this thesis (§1.6).

1.4 Path optimisation

There is a wealth of literature on trajectory optimisation methods in the field of control engineering, yet none of these have previously been applied to the tool-path design problem in ISF. The method of interest in this thesis is the optimal control problem, so a summary of optimal control is now provided in this section.

1.4.1 The optimal control problem

Having developed during the 1950s and 1960s, optimal control theory was one of the earliest approaches to modern control. Of the many books written on the subject, two that are notable are by Lee [67], which was published in 1964 and gives an overview of the early development of optimal control and the more comprehensive text by Bryson and Ho [25].

In optimal control, the control objectives are mathematically defined by a function called a performance index [67]. A statement of the optimal control problem is:

> *Given a description of the dynamic system in terms of state variables and control variables, find values of the control variables that minimise the performance index subject to constraints on the state and/or control variables.*

For the general nonlinear case, this minimisation must be performed numerically and this is often not straightforward [22]. The performance index is usually given the symbol J, the state variables are x and the control variables are p. In continuous time, the above statement translates mathematically to

$$\begin{aligned} \min_{p(t)} \; & J = \phi(x(t_f)) + \int_{t_o}^{t_f} L(x,p,t)\mathrm{d}t \\ \text{s.t.} \; & \dot{x} = f(x,p,t) \\ & \boldsymbol{\psi}(x(t_f), p(t_f), t_f) = 0, \end{aligned} \tag{1.1}$$

where only equality constraints on the states and control variables, $\boldsymbol{\psi}(x(t_f), p(t_f), t_f) = 0$, are considered for now. The function $f(x,p,t)$ describes the dynamic system to be controlled, typically in the form of a process model (see §1.5). The cost function J, which is known as the Bolza type, is a trade-off between a terminal cost $\phi(x(t_f))$ that must be minimised in order to achieve a control objective at the final time t_f and a trajectory cost acting over the whole period $t = \{t_o, t_f\}$. There are two approaches to solving this problem, known as the indirect method and the direct method [22]. The direct approach is to solve the problem as it is stated above numerically, whereas the indirect approach solves the equations related to the first order necessary conditions for optimality, but both approaches lead to a *two-point boundary value problem* (TPBVP) [25]. Although the direct method is usually favoured for nonlinear problems [22], the indirect approach is advantageous under certain conditions. For instance, for linear time invariant systems with quadratic cost functions, the indirect approach leads to an optimal feedback control law with a closed form solution. The necessary optimality conditions are presented here using the classical calculus of variations and its extension by the Pontryagin principle. The alternative formulation, which uses a dynamic programming argument that leads to the Hamilton-Jacobi-Bellman equation, will not be discussed but it is covered in [25].

1.4.2 Pontryagin principle

In a similar manner to the Lagrangian function for constrained optimisation problems, an augmented performance index $\hat{J}$ can be defined and minimised instead of equation 1.1,

$$\begin{aligned}\hat{J} &= [\phi + \boldsymbol{\nu}^T \boldsymbol{\psi}]_{t_f} + \int_{t_o}^{t_f} L(x,p,t) - \lambda(t)[\dot{x} - f(x,p,t)]\ \mathrm{d}t,\\ &= \Phi_{t_f} + \int_{t_o}^{t_f} H(t) - \lambda(t)\dot{x}\ \mathrm{d}t,\\ &= \Phi_{t_f} - \lambda(t_f)x(t_f) + \lambda(t_o)x(t_o) + \int_{t_o}^{t_f} H(t) + \dot{\lambda}(t)x\ \mathrm{d}t, \qquad (1.2)\end{aligned}$$

where $H(t)$ is the Hamiltonian. The minimum of the integrand function in (1.2), F, where

$$F = H(t) + \dot{\lambda}(t)x, \qquad (1.3)$$

must satisfy the Euler-Lagrange equations from the calculus of variations,

$$\frac{\partial F}{\partial x} - \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial F}{\partial \dot{x}} = 0, \qquad (1.4)$$

$$\frac{\partial F}{\partial p} - \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial F}{\partial \dot{p}} = 0, \qquad (1.5)$$

$$\frac{\partial F}{\partial \lambda} - \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial F}{\partial \dot{\lambda}} = 0, \qquad (1.6)$$

and the augmented performance index must also satisfy

$$\frac{\partial \hat{J}}{\partial \boldsymbol{\nu}} = 0, \quad \left.\frac{\partial \hat{J}}{\partial x}\right|_{t=t_o} = 0, \quad \left.\frac{\partial \hat{J}}{\partial x}\right|_{t=t_f} = 0 \text{ and } \frac{\partial \hat{J}}{\partial t} = 0, \qquad (1.7)$$

at a minimum. From (1.4) the first necessary condition for optimality is obtained,

$$\dot{\lambda} = -\frac{\partial H}{\partial x}, \qquad (1.8)$$

(1.5) provides the second necessary condition for optimality,

$$\frac{\partial H}{\partial p} = 0, \qquad (1.9)$$

and (1.6) states that the equations of the dynamic system must be satisfied,

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x(t), p(t), t). \tag{1.10}$$

These necessary conditions are subject to boundary conditions at both $t = t_o$ and $t = t_f$ obtained from (1.7),

$$\boldsymbol{\psi} = 0, \quad \lambda(t_o) = 0, \quad \lambda(t_f) = \left.\frac{\partial \Phi}{\partial x}\right|_{t=tf}, \text{ and } \left[\frac{\partial \Phi}{\partial t} + H\right]_{t=tf} = 0. \tag{1.11}$$

This is a TPBVP because the boundary conditions are defined at two different times, t_o and t_f. The Pontryagin principle essentially modifies the results obtained from the calculus of variations, specifically the necessary condition in (1.9), to the case where the control input is bounded so that $p \in P$, where P is the domain of feasible control inputs. Although the theory is quite involved, the result can be stated neatly as a generalisation of (1.9) to

$$\min_{p \in P} H = 0. \tag{1.12}$$

Hence the optimal control law must minimise the associated Hamiltonian. Under certain situations, such as for linear time-invariant systems with a quadratic cost function, this provides both necessary and sufficient conditions for optimal control as explained in [67], but for the more general case this condition is only necessary and the other conditions for optimality must also be considered.

1.5 Numerical modelling of incremental sheet forming

The constraint,

$$\dot{x} = f(x, p, t), \tag{1.13}$$

in the optimal control problem in (1.1) describes the dynamic system that is under control and it predicts how the system will respond to changes in the control input. The function typically takes the form of a numerical model, analytical process equations or a black-box model built from experimental data. For metal forming processes, the deformation behaviour is highly non-linear and the only reliable modelling methods are based on finite

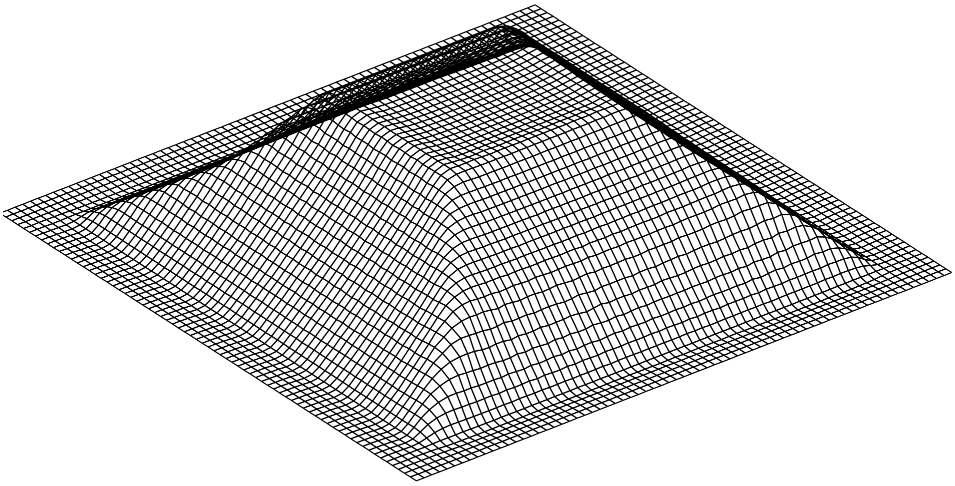


Figure 1.9: An ISF simulation produced in Abaqus using the tool-path in figure 1.10.

element analysis (FEA) software, which are very accurate but so slow that they connot realistically be placed inside an iterative optimiser to solve the optimal control problem. A large portion of this thesis reports the development of simplified models for ISF, which assess whether approximations can be introduced to the material constitutive equations to improve the computational efficiency of ISF models, without significant losses in the accuracy of the predictions. In order to provide a benchmark for the models presented in this thesis, several simulations were produced using FEA software and they are now presented.

1.5.1 Benchmark models for incremental forming

A number of authors have reported that finite element modelling (FEM) of ISF is computationally inefficient. For example, Hirt *et al* [47] observed 16 days for the solution of their finite element model and the model displayed in figure 1.9, which was produced using the commercial FEA package Abaqus [2], took appoximately 24 hours to solve on a desktop computer[1]. The model used the tool-path displayed in figure 1.10 and it was discretised over a uniform mesh of 4900 square elements of which 3600 were free and the remaining were constrained by a backing frame. The target shape in figure 1.9 is described in [21]; it uses wall angles of 35^o and 55^o, has a base of 120×120 mm and is formed with a clamping frame of the same dimensions as the base.

A much larger simulation is shown in figure 1.12, which was produced using the explicit

---

[1] Data courtesy of Markus Bambach, RWTH Aachen University, Germany

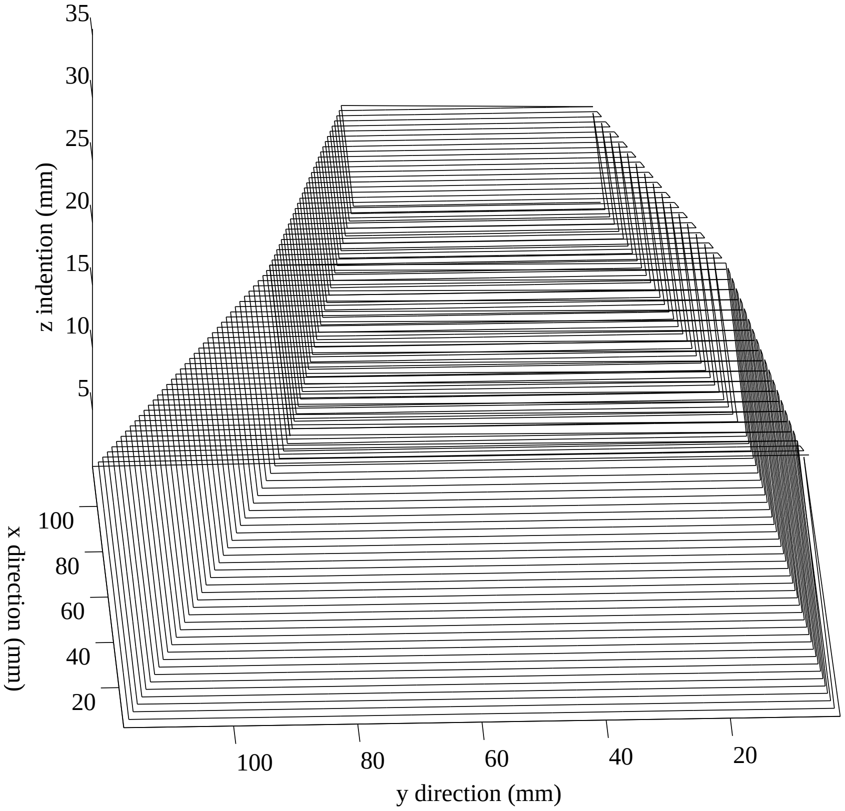


Figure 1.10: Tool path used for simulation in figure 1.9. The scale in the z direction has been stretched for clarity.

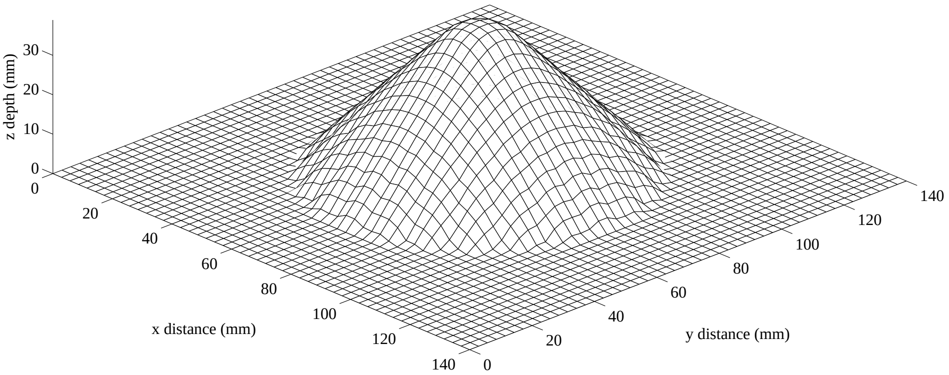


Figure 1.11: Target standard cone shape with 45^o wall angle, 40 mm base radius and 40 mm forming depth.

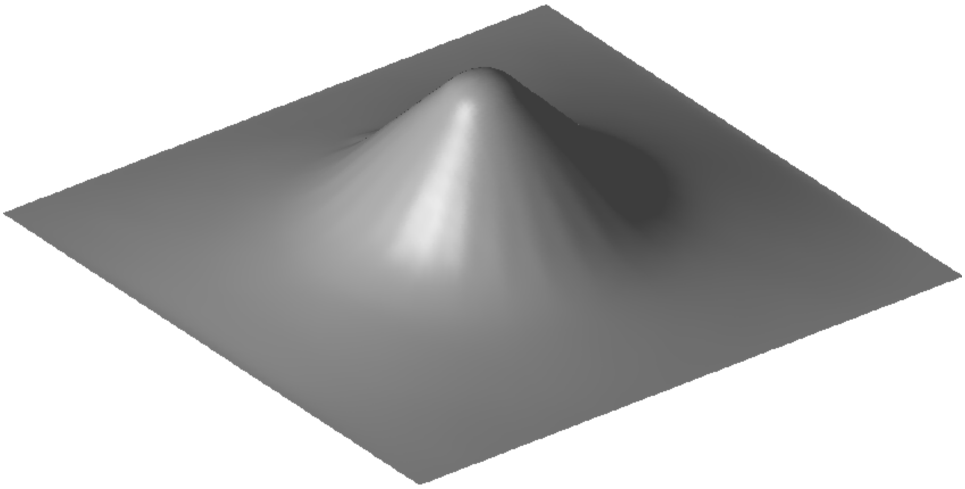


Figure 1.12: Simulation product in figure 1.11 produced using a contour following tool-path with a regular FE mesh in LS-Dyna.

elasto-plastic solver in LS-Dyna, another commercial FEA software package [31]. The target shape for the model is of a cone shaped part with a 45^o wall angle produced using a basic contour following tool-path, shown in figure 1.11. The LS-Dyna model used a fixed regular mesh of 39200 elements and it took 33 hours to complete when run on a parallel processing cluster of 16 Sun Workstations running Linux[2]. Figure 1.13 was also produced in LS-Dyna and it is a simulation of the product shown in figure 1.14, which was formed on the Cambridge ISF machine using 5251-H22 aluminium alloy of 1 mm thickness by the CNC tool-path displayed in figure 1.15, using a hemispherical tool with a radius of 10 mm. The tool was tracked along at 10 mm s^{-1} over a 175 mm×175 mm sheet that was clamped over a 140 mm×140 mm region.

For both of the simulations in figures 1.12 and 1.13 the closest material match to Al 5251-H22 in the LS-Dyna library was used, which was for a 5100 series aluminium alloy. A quadrangular fully integrated shell element with 4 in-plane nodes and 5 through thickness nodes, based on the Belyschko-Tsay element, was used for the simulation. Since the model with the regular mesh was so slow, adaptive re-meshing was applied in figure 1.13, with the smallest allowable element size being 0.7 mm^2 and largest being 5 mm^2. This time, the model took 8 hours to solve when run on the parallel processing cluster of 16 Sun Workstations. It should be noted that the final shape does not replicate the mild contour lines made by the tool on the surface of the sheet, seen in figure 1.14, although further

---

[2] Access kindly provided by Research and Advanced Engineering (RA&E), Ford Motor Co., Dearborn, USA.

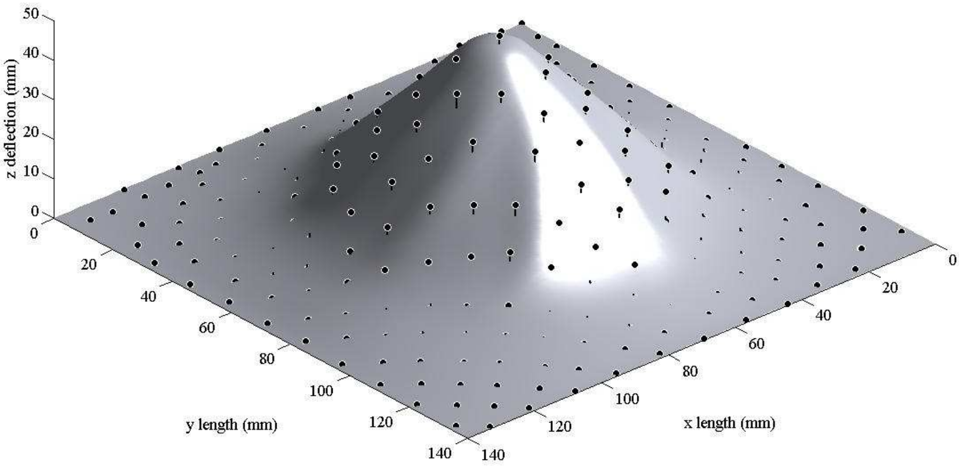


Figure 1.13: Simulation of test product in figure 1.14 produced using adaptive re-meshing in LS-Dyna. The superimposed black dots are measured points from the real product.

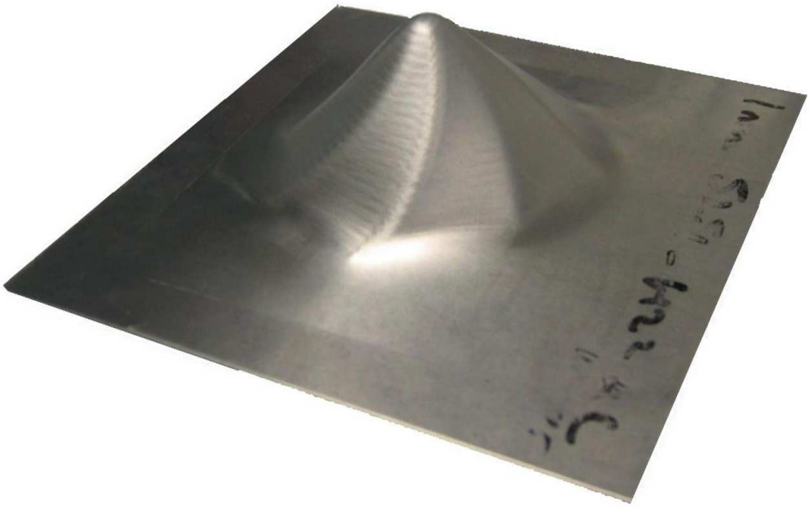


Figure 1.14: Photograph of 5251-H22 Al alloy test product formed on the Cambridge ISF machine.

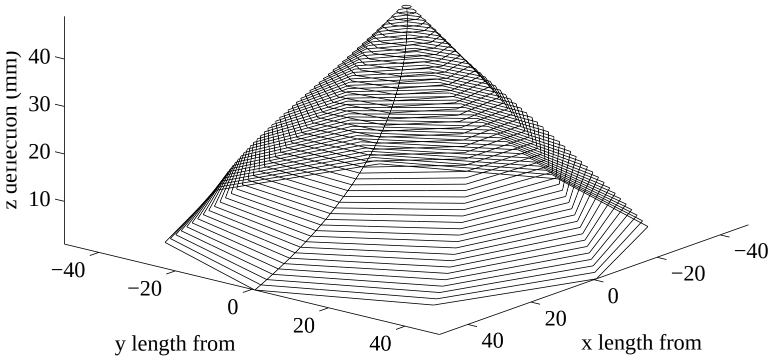


Figure 1.15: The CNC tool path used to produce the product in figure 1.14

.

tests showed that contour lines can be seen in models that were run with larger indentation depths. This demonstrates that there is a limit to the surface resolution that can be replicated by the discrete model. Superimposed upon the simulation in figure 1.13 are a set of 255 measurements taken from the product using a coordinate measuring machine (CMM). The root-mean-square (RMS) vertical difference between these points and the model is 0.998 mm, and the variance is 0.9902 mm^2. Interestingly, the maximum difference between the model and the product at any one point is 3.7 mm, which is over 7% of the maximum forming depth of 50 mm and is arguably quite large. More importantly, the results here provide a benchmark for evaluation purposes later in this thesis.

1.5.2 Second-order cone programming

A definition of a second-order cone program (SOCP) is provided here for future reference because SOCPs are used to solve the models presented in this thesis. An SOCP is an optimisation problem with a linear cost function and quadratic conic constraints, of which linear and quadratic constraints are a subset [71]. A general statement of the SOCP problem is given by

$$\begin{aligned} \min \quad & \mathbf{f}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq 0 \\ & \mathbf{B}\mathbf{x} = 0 \\ & \mathbf{x}_i \in \Theta_i \qquad i = 1, \ldots, NC, \end{aligned} \tag{1.14}$$

where $\mathbf{x} \in \mathbb{R}^n$ are the optimisation variables, $\mathbf{f} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{a \times n}$, $\mathbf{B} \in \mathbb{R}^{b \times n}$, Θ_i are second-order cones, $\mathbf{x}_i$ is a vector containing a subset of elements contained the vector $\mathbf{x}$, and NC is the number of conic constraints. The second-order cone Θ_i is defined as the set of values of $\mathbf{x}_i$ that satisfy

$$\mathbf{x}_i \in \Theta_i \Leftrightarrow \{\mathbf{x}_i \mid \|\mathbf{x}_i^{(2,..n)}\| \leq \mathbf{x}_i^{(1)}\} \tag{1.15}$$

where the superscripts denote elements of $\mathbf{x}_i$.

1.6 Contributions and thesis outline

This section concludes this chapter with a summary of the contributions made during the period of this DPhil with regards to the tool-path planning problem in ISF. There are two areas of contribution in this thesis:

- numerical modelling of ISF
- impulse response based tool-path optimisation for axisymmetric shapes

These are both linked in that they both focus on different aspects of the part accuracy problem in ISF. Automatic tool-path optimisation of part geometry based on feedback has received relatively little attention in the ISF research community and as such, the work here has generated interest at the IfM in Cambridge and it is currently being is being pursued further both there and at Oxford. It hoped that as other researchers begin to address this problem, both of the areas investigated here will be merged into a single product that forms a complete solution for tool-path design in ISF.

The specific contributions made in each area are discussed separately in the subsections below in the order in which they appear in this thesis followed by a summary.

1.6.1 Numerical modelling of ISF

Due to the slow computation times involved in FEA, previous work on path-planning in ISF has either been based on inaccurate simple models for ISF, or on no models at all. The examples of this mentioned in this chapter include Mori *et al*'s [83] simple empirical equations and Bambach *et al*'s [19] metamodel based on a large database of FE simulations. Hirt *et al* [47] suggested a simplified model using spline functions to describe surface geometry in ISF, but the method is limited because it requires prior knowledge of the location of certain points over the surface of the sheet. Attanasio *et al* [18] experimentally evaluated several tool-paths in a two-point ISF process that uses a full product die, but this trial-and-error approach is not appealing because it is time consuming for complex parts. Several approaches for modifying existing elasto-plasitc FE codes to improve efficiency have been reported, for instance by using adaptive remeshing [68], altering the incrementation method for elasto-plastic strain [104] or by using domain decomposition [44], but the speed improvements using these approaches are not sufficient for tool-path optimisation.

Having observed these limitations, it was agreed early on in this DPhil that more work needs to be done if an efficient and accurate method for modelling ISF is to be found. The numerical modelling work conducted here is essentially a feasibility study to evaluate one possible method for producing fast simulations of ISF. The models presented here assess whether firstly, the use of approximations in the constitutive equations describing material deformation and secondly, the use of recent numerical algorithms can result in a fast numerical model for ISF. The key assumption made in the material model is that the sheet behaves in a rigid plastic manner. The motivation for investigating this assumption draws from its successful application in efficiently modelling several other metal forming processes in the late 1980's and 1990's [108]. Under rigid plastic behaviour, it is assumed that there is no elasticity in the deformation of a material, or that elasticity forms a very small component of the overall deformation and it can be ignored. This is illustrated by plot c) of the example tensile test stress-strain curves in figure 1.16, where deformation only occurs if the applied stress, σ, is above a threshold, which is the yield stress. If the yield stress remains constant for any strain as in figure 1.16 d), there is no material work-hardening and the material is referred to as rigid perfectly plastic. For comparison, plots a) and b) show examples of elasto-plastic materials.

Three different approaches to the modelling problem are described in chapters 2, 3 and 4, all of which use the rigid plastic assumption and all of which are previously unexplored methods for modelling ISF. The models exploit recent developments in optimisation software to solve the model equations and where appropriate, the models are compared with the benchmark FEA models described in §1.5.1 and also with parts formed on the Cambridge ISF machine.

In chapter 2, the classical linearised equations of plate and shell theory, comprehensively described in [43], are used to produce a description of the ISF process. These equations are then discretised using the finite difference (FD) method and solved numerically, although some limitations of the linearised equations are observed and the model is found to be memory intensive. Along with the rigid plastic assumption, this model also assumes that only stretching of the sheet and deformation along the primary axis of the tool are

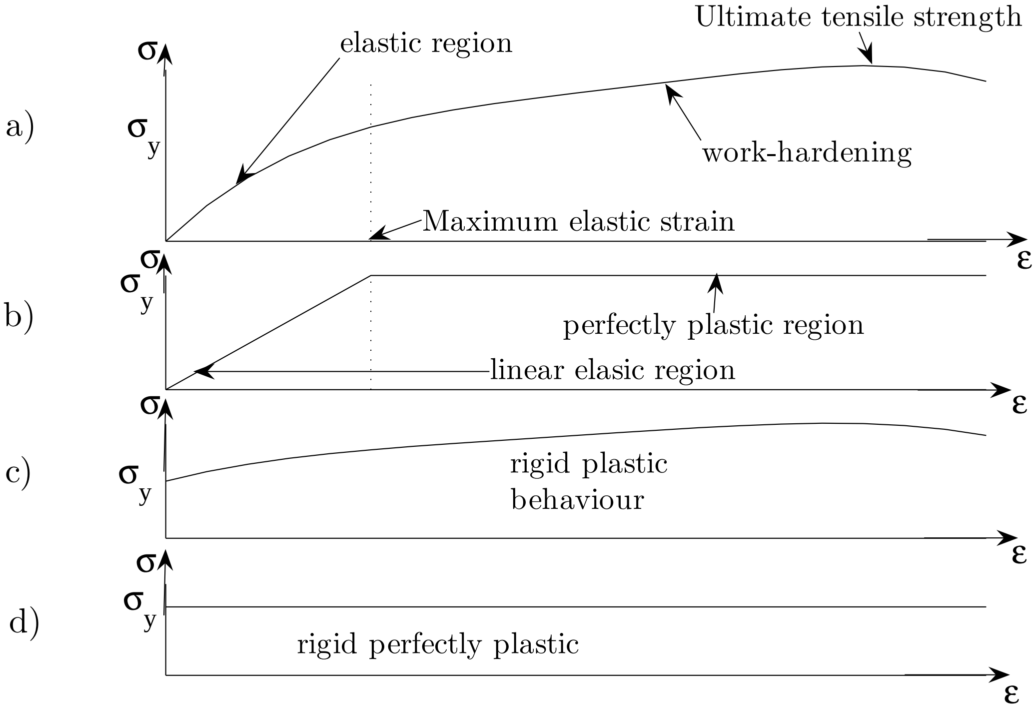


Figure 1.16: Example yield curves for metal with: a) typical elasto-plastic behaviour; b) linear elasticity and perfectly plastic yielding; c) rigid plastic behaviour; d) rigid perfectly plastic behaviour.

significant factors in the model, thus ignoring bending deformation and horizontal motions. A simulation of a tool tracking along a straight line over a sheet is performed using this model [102].

A different modelling approach is described in chapter 3 that was inspired by more recent developments in solving the non-linear large scale optimisation problems that occur in the field of limit analysis using FEM. This work builds upon a wealth of recent literature in sequential limit analysis and in particular, the use of second-order cone programming (SOCP) to solve problems in limit analysis. Previous implementations of the SOCP method have been for single step 2D problems in limit analysis, such as the initial transverse deformation of a flat sheet in plate theory. In this thesis, an extension of the method to the case of a general 3D shell under a time varying load is described, where sequential limit analysis is used to describe the time varying behaviour. Initially, the plate problem is implemented and a hybrid simulation between this plate model and the FD model in chapter 2 is produced and used to simulate the tool tracking a line along a sheet [98].

The plate model is then extended to the shell problem, where the difficulty of combining the stretching and bending components of the sheet deformation in shells is addressed here by using Ilyushin theory, defined in chapter 3. Two methods for deriving the SOCP problem using Illyushin theory are described, referred to as the 'one cone' and 'two cone' models respectively, and both of them were implemented by simulating the full ISF process and validating the results against experimental data. The tool-sheet contact state can be implemented as a sliding constraint and both material work-hardening and anisotropy can be included without placing any further computational burdens on the model; these methods are described in the chapter.

As well as the derivation of the 'limit analysis FE model', further contributions in this chapter are made by reporting some results from running full process simulations with both the one cone and the two cone models. Validation tests using the benchmark simulations in §1.5.1 show that both of these limit analysis FE models can be used to simulate the final sheet deformation for SPIF to a similar degree of accuracy as LS-Dyna. The models are also found to solve faster than LS-Dyna and the behaviour of the models in terms of accuracy and simulation time for various mesh sizes and types is described, including a remeshing strategy that was tested to reduce the number of model variables. The derivation of the one cone model and its performance in modelling the product in figure 1.14 were described in [97, 99, 100].

The third and final model developed during this DPhil is described in chapter 4, which was produced by combining what were considered to be the most positive aspects of the previous two models, whilst also introducing some alternative concepts to overcome some of their limitations. In particular, this model relates the sheet deformation problem to that of a network of rods; an approach inspired by a review of methods that pre-date finite element analysis [111]. The method also uses the rigid plastic assumption and it is solved as an SOCP, although the derivation of the SOCP is more involved than for the previous cases. The model is found to predict both the final deformation shape and intermediate deformation states during ISF to a good level of accuracy and it is significantly faster than both of the previous models. As well as the benchmark SPIF simulations, further simulations for a TPIF process with two moving tools are performed to demonstrate the capabilities of the model.

1.6.2 Tool-path optimisation using impulse responses

The numerical models produced during this research show improvements in computational speed, but they are still too slow to use in solving the optimal control problem. For this reason, rather than using numerical modelling, a new approach was taken in chapter 5 based on experimentally measured spatial impulse responses, which were used to predict the deformation of the sheet in the path optimisation routine.

Chapter 5 begins with a description of how the tool-path problem would be solved in conjunction with a numerical ISF model and likens the procedure to a discrete multi-stage process optimisation problem. The problem is then solved numerically using the spatial impulse response method and, unlike previous path optimsiation techniques in ISF, the tool-paths here are optimised for a complete product. This chapter is restricted to axisymmetric shaped parts to reduce the number of tool-paths parameters to the radius and depth at each layer of the process. It is shown the optimised tool paths outperform standard contour following tool-paths with respect to geometric accuracy.

The method is very efficient and typically solves for the optimal tool-path in less than a minute, opening up the possibility of implementing a fixed end time on-line tool-path optimisation scheme. An implementation of such a method is described using deformation data from the stereo camera for feedback and it is shown that the use of feedback results in further improvements in the geometry. These outcomes are important because they form the first complete demonstration of how automated tool-path optimisation can reduce the geometric errors in ISF.

1.6.3 Summary of Contributions

With the finite difference model in Chapter 2 the novelty factor is limited to the approach taken in its conception, since the data obtained from implementing the model is limited. This chapter should be regarded as a report of an avenue that was investigated for modelling ISF and was subsequently abandoned after discovering several limitations.

In contrast, Chapter 3 provides a more significant contribution through merging several concepts in the field of limit analysis, such as sequential limit analysis, second order cone programming and upper bound limit load analysis to form a new method for modelling shells. There is currently an active interest in these areas, so this work has been recognised and published in a good journal [100]. Furthermore, this model is used to achieve one of the initial aims of this research, to demonstrate it is possible to model ISF using the rigid plastic assumption. The performance of the model is evaluated against experimental data and elasto-plastic FEA under two criteria, the ability to predict the sheet deformation accurately and the actual computation time involved. Several extensions to the model are also described and evaluated.

Chapter 4 provides a novel approach to modelling ISF that is based on a simple but effective concept. The model is evaluated under the same criteria as that in Chapter 3 and it gives unexpectedly good results. The results provide a strong case for modelling deformation in ISF in this way as an alternative to FEA for some applications. The model also fulfills another initial aim of this research, to gain an idea of the computation speed that can currently be achieved with the rigid plastic assumption.

Chapter 5 completes the third and final aim of this thesis by demonstrating that the tool-path trajectory can automatically be optimised to improve the geometric accuracy of products formed by ISF. The work here shows for the first time how this can be done by using automatic model based control and experimental data is provided to show how effective the method is. The criteria used for evaluating the path optimisation scheme is the reduction in RMS error between the target shape and actual product when compared to an unoptimised tool-path. This path optimisation work is limited to axisymmetric products only so that the experimental impulse response model can be used to predict the sheet behaviour. It provides an effective solution for this class of shapes and has formed the basis of two journal papers [4, 101].

As discussed in the chapter this work should be differentiated from recent work by other researchers that may be considered to use feedback, but which require human input during the process unlike the automatic feedback method developed here. These works are less systematic because they are based on trial and error or intuitive rules and also, the results reported tend to be for one or two shapes. This is awkward in two ways, firstly it is difficult to understand how generally applicable the methods are, which is avoided here by focusing on a particular class of shapes and secondly, it is less straightforward to see how to analyse or extend the work. For instance, it is not possible to analyse the robustness of these methods, whereas the model based method described here would lend itself to established techniques for doing this.

In order to extend the method to a general class of shapes, for which the deformation observed is significantly more non-linear than for axisymmetric shapes, the models in either Chapters 3 or 4 would have to be used in combination with the path optimisation scheme in Chapter 5. Hence although this step has not been performed here, these chapters should be viewed as two components that provide a step towards a complete solution to the part geometry problem in ISF, for any class of shapes.

Chapter 2

Finite difference small strain model

2.1 Finite Difference Model - Membrane Work

During ISF, the rectangular metal sheet is clamped at the edges and deformed by the indentation of a small spherical tool that follows a CNC sequence. The requirements of the deformation model are to predict the geometry, $u(t)$, of a sheet in ISF at time t resulting from a known shape of the sheet, $u(0)$, at an initial time $t = 0$ and a change in the tool path vector over some time period $t \in [0, t_f]$. The model may also predict the force on the tool during this period and must obey displacement constraints due to the boundary conditions imposed by the tool and the clamping frame.

The model described here was developed using Hamilton's principle and in order to do this the Lagrangian, L, in the context of classical mechanics is defined as

$$L(u, \dot{u}, t) = \mathrm{KE} - \mathrm{PE} + \mathrm{TW}, \tag{2.1}$$

where KE is potential energy, PE is potential energy and TW is the non-conservative, or external, work done by the system, such as the work done by traction forces [37]. By assuming that inertia effects are negligible and contact between the tool and the sheet is frictionless, there are no KE or TW terms in the Lagrangian. The PE is the internal work given by the integration of specific work over the volume of the sheet Ω, so that

$$L(u, \dot{u}, t) = \int_{\Omega} \bar{\sigma}(u, \dot{u}, t)\bar{\epsilon}(u, \dot{u}, t) d\Omega, \tag{2.2}$$

where $\bar{\sigma}$ are the material effective stresses, $\bar{\epsilon}$ are equivalent strains and Ω is the volume of the sheet. Hamilton's Principle states that the dynamical path taken by a system is that which leads to a stationary value of a functional known as the *J-integral* [37],

$$\int_0^{t_f} L(u, \dot{u}, t)dt. \tag{2.3}$$

From the calculus of variations, it can be shown that this minimum must satisfy the Euler-Lagrange equation,

$$\frac{\partial L}{\partial u} - \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{u}} = 0. \tag{2.4}$$

The Euler-Lagrange equation is in fact satisfied by all extreme points of a functional, but since the J-integral is derived from an energy based problem it can be assumed that the only extremal point is the minimum. Under the rigid plastic assumption, the internal work is independent of the current deformation state of the sheet, u, because no residual stresses remain in the sheet after loading. Hence $L(u, \dot{u}, t) = L(\dot{u}, t)$, so that

$$\frac{\partial L}{\partial u} = 0. \tag{2.5}$$

Furthermore, if $\dot{\bar{\epsilon}}$ is the effective strain rate, then the Euler-Lagrange equation suggests that the dynamical path of the system is that which minimises the internal work rate, $\dot{W}$, with respect to $\dot{u}$, since the necessary condition for minimising (2.3) reduces to

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{u}} = \frac{\partial \dot{L}}{\partial \dot{u}} = \frac{\partial}{\partial \dot{u}}\int_{\Omega} \bar{\sigma}\dot{\bar{\epsilon}}d\Omega = \frac{\partial \dot{W}}{\partial \dot{u}} = 0, \tag{2.6}$$

which is satisfied when $\dot{W}$ is minimal with respect to $\dot{u}$. The model can therefore be written as an optimisation function that instantaneously minimises the internal work rate with respect to the rate of deformation, usually termed the velocity v, of the sheet,

$$\min_{v(t)} \ \dot{W} = \min_{v(t)} \ \int_{\Omega} \bar{\sigma}\dot{\bar{\epsilon}}d\Omega, \tag{2.7}$$

where

$$\dot{u}(t) = v(t). \tag{2.8}$$

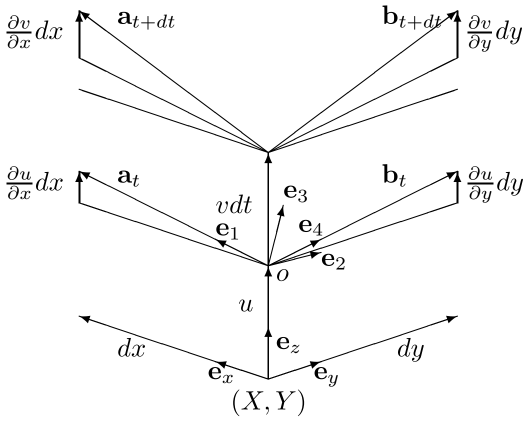


Figure 2.1: Change in orientation of element from time t to time $t + dt$.

By assuming the material is isotropic and pressure-independent, the von Mises criterion can be used to describe the yielding behaviour of the sheet. Furthermore, under the assumption of perfectly rigid plastic behaviour as defined in §1.6.1, the equivalent stress $\bar{\sigma}$ is constant and equal to the yield stress of the material, $\bar{\sigma}(u, \dot{u}, t) = \sigma_y$. The integration over the volume of the sheet in (2.7) becomes

$$\int_\Omega \bar{\sigma}\dot{\bar{\epsilon}} d\Omega = \sigma_y \int_\Omega \dot{\bar{\epsilon}} d\Omega. \tag{2.9}$$

A number of further approximations are made as follows [113, 43, 116]:

- Volume is conserved throughout the deformation, so the sheet is incompressible.
- The sheet has uniform thickness at initial time $t = 0$.
- Vectors normal to the *middle surface* remain normal throughout the deformation, where the middle-surface is defined as the surface half-way through the thickness of the sheet. This is sometimes referred to as Kirchhof's assumption [61].
- The in-plane strain rates are found by considering the stretch and shear of a first-order infinitesimal element from time $t = 0$ to time $t = t_f$, assuming that any point in the sheet displaces vertically only throughout the deformation, if the original configuration of the sheet is horizontal.

The first three of these are standard assumptions that are assumed to hold for all of the models in this thesis. The final assumption is used here to reduce the number of model variables and is considered to be satisfactory because in practice, most of the deformation appears to occur in the vertical direction [57]. The above assumptions are sufficient to obtain equations for the strain rate components that are required to calculate the equivalent strain rate,

$$\dot{\bar{\epsilon}} = \sqrt{\frac{2}{3}(\dot{\epsilon}_{11}^2 + \dot{\epsilon}_{22}^2 + \dot{\epsilon}_{33}^2 + 2\dot{\epsilon}_{12}^2 + 2\dot{\epsilon}_{13}^2 + 2\dot{\epsilon}_{23}^2)}, \tag{2.10}$$

where $\dot{\epsilon}_{11}$, $\dot{\epsilon}_{22}$ and $\dot{\epsilon}_{33}$ are strain rates in mutually perpendicular directions and $\dot{\epsilon}_{12}$ is half the value of the shear strain rate in the plane defined by $\dot{\epsilon}_{11}$ and $\dot{\epsilon}_{22}$. Also, $\dot{\epsilon}_{12} = \dot{\epsilon}_{21}$, $\dot{\epsilon}_{13} = \dot{\epsilon}_{31}$ and $\dot{\epsilon}_{23} = \dot{\epsilon}_{32}$. Note that the benefit in using equivalent strain rate is that it combines six different strain rate terms into a single scalar. Figure 2.1 labels these directions as vectors e_1 for $\dot{\epsilon}_{11}$, e_2 for $\dot{\epsilon}_{22}$, and e_3 for $\dot{\epsilon}_{33}$. The infinitesimal element shown in figure 2.1 is initially orientated in the $(x,\ y)$ plane and by a time t+dt in the future it has deformed into the plane given by e_1 and e_4, which lie in the (x, z) and (y, z) planes respectively. The strain rate values for $\dot{\epsilon}_{13}$, $\dot{\epsilon}_{23}$ and $\dot{\epsilon}_{33}$, which follow from the above assumptions, are

$$\dot{\epsilon}_{13} = 0, \tag{2.11}$$

$$\dot{\epsilon}_{23} = 0, \tag{2.12}$$

and due to conservation of volume,

$$\dot{\epsilon}_{33} = -\dot{\epsilon}_{11} - \dot{\epsilon}_{22}. \tag{2.13}$$

From figure 2.1 the following vectors are obtained:

$$\mathbf{a}_t = \begin{bmatrix} dx \\ 0 \\ \frac{\partial u}{\partial x}dx \end{bmatrix} \tag{2.14}$$

$$\mathbf{b}_t = \begin{bmatrix} 0 \\ dy \\ \frac{\partial u}{\partial y}dy \end{bmatrix} \tag{2.15}$$

$$\mathbf{a}_{t+dt} = \begin{bmatrix} dx \\ 0 \\ \frac{\partial u}{\partial x}dx + \frac{\partial v}{\partial x}dtdx \end{bmatrix} \tag{2.16}$$

$$\mathbf{b}_{t+dt} = \begin{bmatrix} 0 \\ dy \\ \frac{\partial u}{\partial y}dy + \frac{\partial v}{\partial y}dtdy \end{bmatrix} \tag{2.17}$$

The strain rates in the e_1 and e_4 planes are then given by

$$\dot{\epsilon}_{11}\mathrm{d}t = \frac{|\mathbf{a}_{t+dt}| - |\mathbf{a}_t|}{|\mathbf{a}_t|}, \tag{2.18}$$

$$\dot{\epsilon}_{44}\mathrm{d}t = \frac{|\mathbf{b}_{t+dt}| - |\mathbf{b}_t|}{|\mathbf{b}_t|}, \tag{2.19}$$

and

$$\dot{\epsilon}_{14}\mathrm{d}t = \frac{1}{2}(\arg(\mathbf{a}_{t+dt}, \mathbf{b}_{t+dt}) - \arg(\mathbf{a}_t, \mathbf{b}_t)), \tag{2.20}$$

where $\arg(\mathbf{p}, \mathbf{q})$ denotes the angle between any two vectors $\mathbf{p}$ and $\mathbf{q}$. One can define α and β as,

$$\alpha = \sqrt{1 + \frac{\partial u}{\partial x}^2}, \tag{2.21}$$

and

$$\beta = \sqrt{1 + \frac{\partial u}{\partial y}^2}, \tag{2.22}$$

so that from figure 2.1,

$$|\mathbf{a}_t| = \alpha \mathrm{d}x, \qquad |\mathbf{b}_t| = \beta \mathrm{d}y. \tag{2.23}$$

Substituting this into (2.18) gives

$$\begin{aligned} \dot{\epsilon}_{11}\mathrm{d}t &= \frac{\sqrt{(\mathrm{d}x)^2 + \left(\mathrm{d}x\frac{\partial u}{\partial x} + \frac{\partial v}{\partial x}\mathrm{d}x\mathrm{d}t\right)^2} - \alpha \mathrm{d}x}{\alpha \mathrm{d}x}, \\ &= \frac{\alpha \mathrm{d}x\sqrt{1 + \frac{1}{\alpha^2}\left(2\frac{\partial u}{\partial x}\frac{\partial v}{\partial x}\mathrm{dt} + \frac{\partial v^2}{\partial x}\mathrm{d}t^2\right)} - \alpha \mathrm{d}x}{\alpha \mathrm{d}x}, \\ &= \sqrt{1 + \frac{1}{\alpha^2}\left(2\frac{\partial u}{\partial x}\frac{\partial v}{\partial x}\mathrm{dt} + \frac{\partial v^2}{\partial x}\mathrm{d}t^2\right)} - 1. \end{aligned} \tag{2.24}$$

The difficulty with (2.24) is that strain is a non-linear function of the optimisation variable in (2.7), $v(t)$. A linear relationship between strain and sheet velocity would be preferable, firstly because it would simplify the spatial discretisation of the problem and secondly, because it would provide a greater choice of methods for performing the energy minimisation in (2.7). The expression in (2.24) can be approximated by applying a binomial expansion and ignoring terms of the order $\mathrm{d}t^2$ and higher, leading to,

$$\dot{\epsilon}_{11} = \frac{1}{\alpha^2}\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} \tag{2.25}$$

and similarly from (2.19),

$$\dot{\epsilon}_{44} = \frac{1}{\beta^2}\frac{\partial u}{\partial y}\frac{\partial v}{\partial y}. \tag{2.26}$$

These equations confirm those given in the classic text by Flügge [43], where a similar analysis is performed resulting in

$$\dot{\epsilon}_{11} = \frac{\partial v}{\partial x}\sin\xi\cos\xi, \qquad \dot{\epsilon}_{44} = \frac{\partial v}{\partial y}\sin\phi\cos\phi, \tag{2.27}$$

where ξ is $\arg(\mathbf{e}_1,\ \mathbf{e}_x)$ and ϕ is $\arg(\mathbf{e}_4,\ \mathbf{e}_y)$. From figure 2.1, it can be seen that these equations are equivalent to (2.25) and (2.26), since

$$\sin\xi = \frac{1}{\alpha}\frac{\partial u}{\partial x}, \qquad \cos\xi = \frac{1}{\alpha}, \qquad \sin\phi = \frac{1}{\beta}\frac{\partial u}{\partial y}, \qquad \cos\phi = \frac{1}{\beta}. \tag{2.28}$$

The corresponding expression for shear strain also involves the use of first order approximations. Let $\theta = \arg(\mathbf{a}_t, \mathbf{b}_{t+dt})$ and $\theta + \delta\theta = \arg(\mathbf{a}_{t+dt}, \mathbf{b}_{t+dt})$, then

$$\dot{\epsilon}_{14}\mathrm{d}t = \frac{1}{2}\delta\theta = \arg(\mathbf{a}_{t+dt}, \mathbf{b}_{t+dt}) - \arg(\mathbf{a}_t, \mathbf{b}_t). \tag{2.29}$$

Flügge derives the following expression for $\delta\theta$,

$$\delta\theta = \frac{1}{\sin\theta}\left(\frac{\partial v}{\partial x}\cos^3\xi\sin\theta + \frac{\partial v}{\partial y}\cos^3\theta\sin\xi\right), \tag{2.30}$$

where only the $\sin\theta$ term is not known. By taking cross products of the vectors in (2.14) and (2.15), this is

$$\sin\theta = \frac{\zeta}{\alpha\beta}, \tag{2.31}$$

where

$$\zeta = \sqrt{1 + \frac{\partial u}{\partial x}^2 + \frac{\partial u}{\partial y}^2}. \tag{2.32}$$

When combined with equations 2.28, these expressions give

$$\dot{\epsilon}_{14} = \frac{1}{2\zeta}\left(\frac{1}{\alpha^2}\frac{\partial u}{\partial x}\frac{\partial v}{\partial y} - \frac{1}{\beta^2}\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}\right). \tag{2.33}$$

Since e_1 and e_4 are not mutually perpendicular, (2.10) cannot simply be applied to the non-orthogonal strain rates $\dot{\epsilon}_{11}$ and $\dot{\epsilon}_{44}$. A transformation derived from [43] is therefore applied to $\dot{\epsilon}_{44}$ to give $\dot{\epsilon}_{22}$, which does form a perpendicular pair with $\dot{\epsilon}_{11}$,

$$\dot{\epsilon}_{22} = \left(1 - \frac{1}{\alpha\zeta}\frac{\partial u}{\partial x}^2\right)\dot{\epsilon}_{44} + \frac{1}{\beta\zeta}\frac{\partial u}{\partial y}^2\dot{\epsilon}_{11}, \tag{2.34}$$

and a similar transformation is also applied to give $\dot{\epsilon}_{12}$ from $\dot{\epsilon}_{14}$,

$$\dot{\epsilon}_{12} = \frac{(\dot{\epsilon}_{44} - \dot{\epsilon}_{11})}{2\zeta}\frac{\partial u}{\partial x}\frac{\partial u}{\partial y} + \dot{\epsilon}_{14}. \tag{2.35}$$

2.1.1 Boundary conditions

During SPIF, the sheet is clamped by a flat continuous frame around its edges with one perfectly rigid tool moving along a known tool path on the upper surface of the sheet. The shape of the tip of the tool is convex, therefore its surface can be described by a normal unit vector which always points out of the tool. The function $T(x, y, t)$ is used to describe the contact boundary conditions at the tool. The function $T(x, y, t)$ gives the vertical position of the surface of the tool at (x, y) at time t, so that if indentation is in the positive z direction, the constraint for contact at the tool is

$$u(x, y, t) \geq T(x, y, t), \tag{2.36}$$

and if $u(x, y, t) = T(x, y, t)$ at any point (x, y), the tool and sheet are in contact. If it is required to account for the thickness of the sheet, (2.36) must be modified to incorporate the distance between the middle surface and the upper surface of the sheet. If $u_{boundary} \subset u$ and $v_{boundary} \subset v$ describe sheet deformation and velocity in regions (x, y) that are in contact with the clamping frame, the constraints are

$$u_{boundary} = 0, \tag{2.37}$$

$$v_{boundary} = 0. \tag{2.38}$$

2.1.2 Summary

A statement of the problem that needs to be solved in the ISF model is provided in (2.7) and subsequent equations give the components of strain rate that allow the calculation of equivalent strain rate. It is not possible to solve these non-linear PDEs analytically and therefore numerical solution techniques must be explored. The first numerical problem that will be addressed is the discretisation of the continuous strain rate equations, after which some numerical optimisation methods available for performing the minimisation in (2.7) are described.

2.1.3 Spatial discretisation

The finite difference method (FDM) is a natural starting point for discretisation due to its simplicity and, often, computational efficiency over a finite element method (FEM) implementation. The sheet is discretised over a unform grid with regions of size $\Delta X \Delta Y$ where the values of the variables u and v were stored only at the nodes. As shown in figure 2.2, $Region(m, n)$ is defined as the mth region from the global origin in the e_x-direction and the nth region from the global origin in the e_y-direction, where e_x and e_y are perpendicular unit vectors in the plane of the middle-surface of the sheet at time $t = 0$. The gradient of each $Region(m, n)$ in the e_x and e_y directions is uniform and assumed to approximate the gradient of a point in the centre of the region, labelled $Point(m, n)$ in figure 2.2 and hence all components of strain rate across each $Region(m, n)$ are constant. The boundary conditions for the model were discretised in the same way as the sheet. According to these

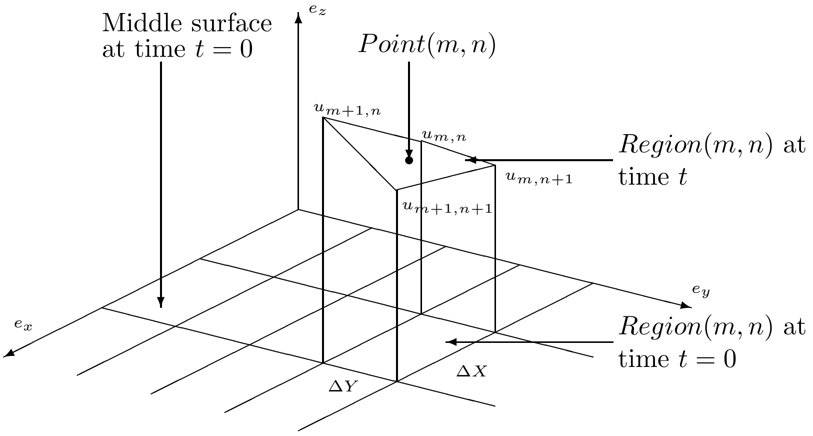


Figure 2.2: Structure of global mesh.

assumptions, (2.7) can be written as the minimisation of the discretised internal work rate

$$\dot{W} = \min_{V_{m,n}} \quad \Omega_e \sigma_y \sum_{m}^{M} \sum_{n}^{N} \dot{\bar{\epsilon}}_{m,n}, \tag{2.39}$$

where the sheet is discretised over a $M \times N$ grid and Ω_e is the volume of an element of area $\Delta X \Delta Y$.

The gradient at each $Point(m,n)$ is assumed to be equal to the gradient at the edges of $Region(m,n)$, which were computed using the following first order forward difference equations:

$$\left.\frac{\partial u}{\partial x}\right|_{m,n} \approx \frac{1}{\Delta X}(u_{m+1,n} - u_{m,n}), \tag{2.40}$$

$$\left.\frac{\partial u}{\partial y}\right|_{m,n} \approx \frac{1}{\Delta Y}(u_{m,n+1} - u_{m,n}), \tag{2.41}$$

$$\left.\frac{\partial v}{\partial x}\right|_{m,n} \approx \frac{1}{\Delta X}(v_{m+1,n} - v_{m,n}), \tag{2.42}$$

$$\left.\frac{\partial v}{\partial y}\right|_{m,n} \approx \frac{1}{\Delta Y}(v_{m,n+1} - v_{m,n}). \tag{2.43}$$

The indicies m and n in (2.43) indicate the nodal position of the deformation variables u and v that are of interest.

2.1.4 Time integration

Discrete time integration was performed using the explicit forward difference method to update the deformation at the ith time step, $u(t_i)$, over a fixed time step Δt as

$$u_{m,n}(t_{i+1}) = u_{m,n}(t_i) + v_{m,n}(t_i)\Delta t. \tag{2.44}$$

This allows a work function that is to be optimised to be written as the change in work from time t_i to t_{i+1} as follows,

$$\Delta W = \min_{v_{m,n}(t_i)} \Omega_e \sigma_y \sum_{m,n}^{M,N} \dot{\bar{\epsilon}}_{m,n}(u_{m,n}(t_i), v_{m,n}(t_i))\Delta t. \tag{2.45}$$

The only unknowns in (2.45) at time t_i are the nodal velocities $v_{m,n}(t_i)$ and the scalar increment of work, ΔW, must be minimised with respect to them. However, the deformed state at time t_{i+1} must continue to satisfy the boundary conditions at the tool as described above, which can be achieved by assuming a contact state at time t_{i+1}, so that for nodes that are destined to be in contact, $U_{m,n}(t_{i+1}) - T_{m,n}(t_{i+1}) = 0$ and the corresponding $V_{m,n}$ are no longer unknown. The minimisation problem in (2.45) is optimised for all remaining nodes and the contact state of the predicted deformed shape at time t_{i+1} is then tested for consistency, by explicitly applying the constraint

$$u_{m,n}(t_{i+1}) - T_{m,n}(t_{i+1}) \geq 0, \tag{2.46}$$

in the optimisation routine.

2.1.5 Implementation of the process model

Equations (2.14-2.20) were combined with the transformations in (2.34) and (2.35) and directly coded into Matlab to compute the change in work in (2.45). This was coupled with the tool and boundary constraints and the change in the deformation field, $\mathbf{v} = \{v_{m,n} \ \forall\ (m,n)\}$, was computed by minimising the change in work using the Matlab Optimisation Toolbox function for constrained optimisation, *fmincon*. It is noted that other solvers for general constrained optimisation problems are available that are considered to be more

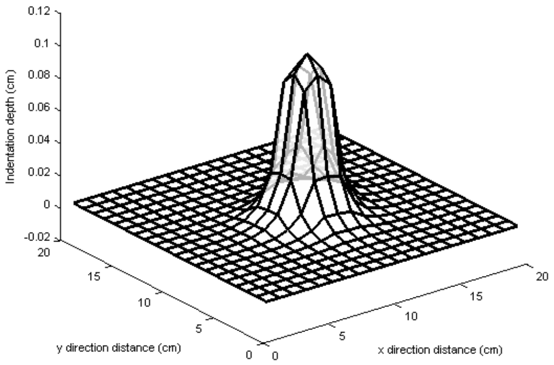


Figure 2.3: Solution: indentation of a single point by a tool with a spherical tip.

reliable. The simulation over a 20 × 20 grid of an indentation by the tool at one point using a large spherical tool-tip is shown in figure 2.3. This program took almost one hour to complete the optimisation on a desktop computer, highlighting the need for a more sophisticated solution method for the process model, especially when considering that the final model would require a spatial resolution equivalent to at least a 50 × 50 grid. In order to demonstrate the evolution of the deformation field, figure 2.4 was produced to show the explicit time stepping of a small indenter, which is in contact with only one node at a time and is tracked along a straight line at a height of 1 mm over a more computationally tractable 10×10 grid.

To improve computational speed, the aim is to form a linear relationship between the optimisation variables and the strain rates by using the linearised approximations of the strain rate equations (2.25), (2.34) and (2.35). It will be demonstrated below that such a relationship allows a more sophisticated second-order cone programming optimisation scheme to be utilised for the energy minimisation.

Before proceeding, a limitation of the linearised strain equations at the first time-step of the ISF process, when the initial orientation of the sheet is flat, must be overcome. In this case, the spatial derivatives of $u_{m,n}(t_0)$ are zero and so (2.25), (2.34) and (2.35) are zero, resulting in a zero objective function, which corresponds to a spurious zero energy mode. This is because the above formulation assumes the initial configuration of the sheet to be non-flat. If one redraws figure 2.1 to analyse the deformation of a point in the metal sheet from an initially flat configuration, the strain rate equations for time $t = 0$ are derived [115]

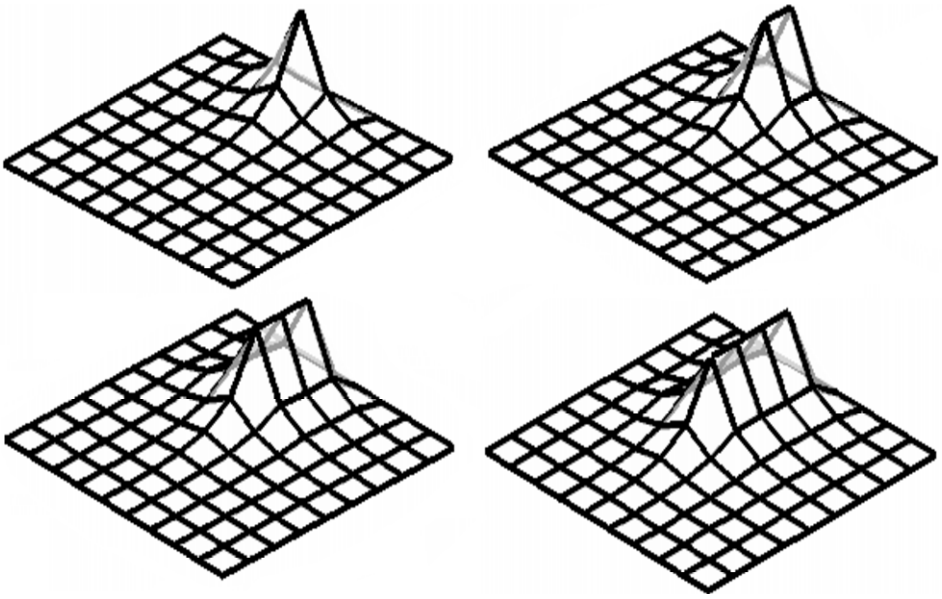


Figure 2.4: Solution: small indenter tacking along a straight line, solved by directly coding (2.14-2.20) and solving using fmincon

as

$$\dot{\epsilon}_{11} = \frac{1}{2}\left(\frac{\partial v}{\partial x}\right)^2, \tag{2.47}$$

$$\dot{\epsilon}_{22} = \frac{1}{2}\left(\frac{\partial v}{\partial y}\right)^2, \tag{2.48}$$

$$\dot{\epsilon}_{12} = \frac{\partial v}{\partial x}\frac{\partial v}{\partial y}, \tag{2.49}$$

which can be used for the first time-step of the ISF process and then replaced by (2.25), (2.34) and (2.35) for subsequent steps. Figure 2.5 displays a solution of (2.47-2.49) for a point load over a 50×50 grid and a further example displaying the time stepping scheme is shown in figure 2.6. These were solved in 1 hr 8 mins and 1 hr 36 mins respectively using *fmincon*, so the need for recasting the optimisation problem for computational speed remains.

2.2 SOCP formulation

In order to improve optimisation speed, the minimisation problem is recast into a second-order cone programme (SOCP), as defined in §1.5.2, which has a convex structure that makes it more efficient to solve than a general non-linear optimisation problem. Let $\Delta v(t)_x$

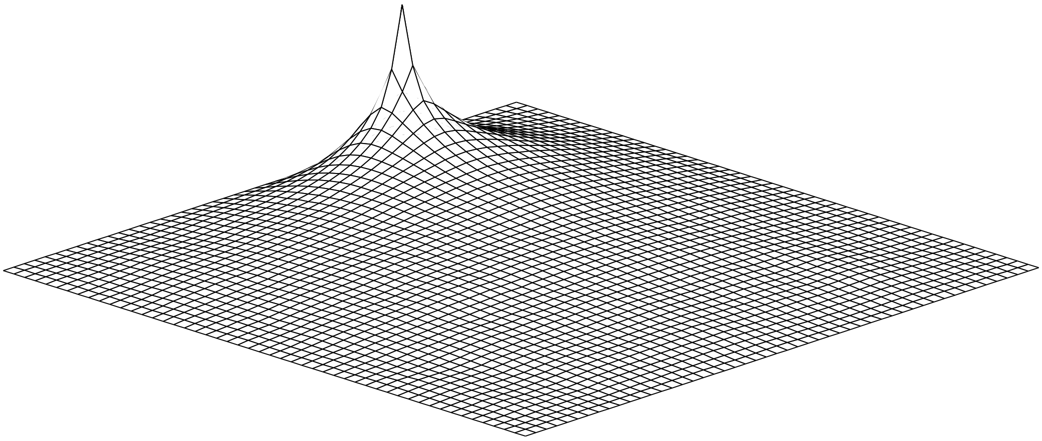


Figure 2.5: Deformation field for a point loaded flat sheet using (2.47-2.49) with a 50×50 mesh. Solved in 1 hr 8 mins on a desktop computer using fmincon.

and $\Delta v(t)_y$ denote the following:

$$\Delta v(t)_x = v_{m+1,n}(t) - v_{m,n}(t) \tag{2.50}$$

$$\Delta v(t)_y = v_{m,n+1}(t) - v_{m,n}(t) \tag{2.51}$$

For each time step Δt, the linearised equations for strain rate can be approximated by strain increments of the form

$$\Delta\epsilon_{11m,n}(t) = P_1\Delta v(t)_x + Q_1\Delta v(t)_y, \tag{2.52}$$

$$\Delta\epsilon_{12m,n}(t) = P_2\Delta v(t)_x + Q_2\Delta v(t)_y, \tag{2.53}$$

$$\Delta\epsilon_{22m,n}(t) = P_3\Delta v(t)_x + Q_3\Delta v(t)_y, \tag{2.54}$$

$$\Delta\epsilon_{33m,n}(t) = P_4\Delta v(t)_x + Q_4\Delta v(t)_y, \tag{2.55}$$

where P_l and Q_l, for $\{l = 1, \ldots, 4\}$, contain all the terms that are independent of the optimisation variables and hence, P_l and Q_l are constant with respect to the optimisation problem. For instance, by discretising (2.25) one can write

$$\Delta\epsilon_{11m,n}(t) = \frac{(u_{m+1,n} - u_{m,n})}{\Delta X^2 + (u_{m+1,n} - u_{m,n})^2}\Delta v(t)_x\Delta t, \tag{2.56}$$

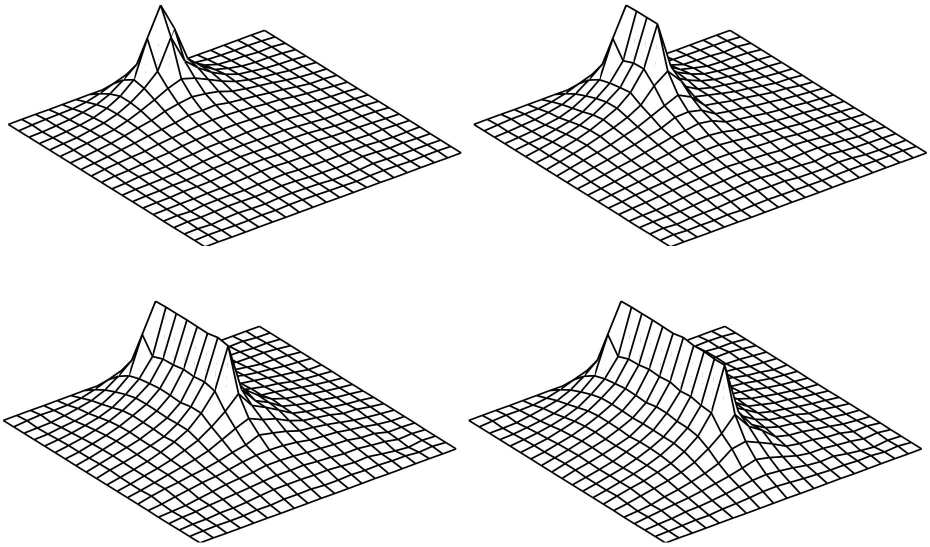


Figure 2.6: Deformation field due to a point load tracking along a straight line. Solved over a 20×20 mesh using (2.47-2.49) for the first time step and (2.25), (2.34) and (2.35) for the remaining time steps. Total solution time 1 hr 36 mins (290 seconds per time step) on a desktop computer using fmincon.

hence by comparing this with (2.52),

$$P_1 = \frac{(u_{m+1,n} - u_{m,n})}{\Delta X^2 + (u_{m+1,n} - u_{m,n})^2}, \qquad Q_1 = 0. \tag{2.57}$$

The other terms P_l and Q_l, for $\{l = 2, \ldots, 4\}$, are obtained in a similar manner and it follows that (2.10) is equivalent to

$$\Delta\bar{\epsilon}_{m,n} = \sqrt{\sum_{l=1}^{4} (P_l \Delta v(t)_x + Q_l \Delta v(t)_y)^2}. \tag{2.58}$$

Note that the equivalent strain rate can be replaced by a strain increment due to the discretisation of time in the problem, so that $\Delta\bar{\epsilon}_{m,n} = \dot{\bar{\epsilon}}_{m,n}\Delta t$.

Nodes at the edge of the sheet are in contact with the clamping frame and are known to have zero deformation and velocity, so they do not contribute to the objective function and can be omitted from the vector of optimization variables. Hence, let $\mathbf{v} \in \mathbb{R}^{M'N'}$, where $M' = M - 2$ and $N' = N - 2$, be a column vector containing all velocity terms for nodes that are not in contact with the frame. Also, let $\mathbf{A} \in \mathbb{R}^{4 \times M'N'}$ be a matrix that comprises of rows containing zero terms except for constants at appropriate locations related to the nodal velocities $V_{m+1,n}(t), V_{m,n+1}(t)$, and $V_{m,n}(t)$. The equivialent strain increment can be written as

$$\Delta\bar{\epsilon}_{m,n} = \left\|\mathbf{A}_k\mathbf{v}\right\|_2, \tag{2.59}$$

from which (2.45) is

$$\Delta W = \min_{v_{m,n}(t_i)} \Omega_e\sigma_y \sum_{m,n}^{M,N} \left\|\mathbf{A}_k\mathbf{v}\right\|_2, \tag{2.60}$$

where $k = 1,..,M'N'$. This is a sum of norms optimisation problem which can be converted into an SOCP [24] by introducing a vector of slack variables $\mathbf{y} \in \mathbb{R}^{M'N'}$ with elements defined such that

$$y_k \geq \Delta\bar{\epsilon}_{m,n}. \tag{2.61}$$

If $\sum y_k$ is now minimized in place of $\sum \Delta\bar{\epsilon}_{m,n}$ whilst subject to the constraint in (2.61), then at the solution point the inequality in (2.61) becomes an equality, allowing (2.60) to be written as the following SOCP,

$$\begin{aligned} \min \quad & [1...1\ \ 0...0]\begin{bmatrix}\mathbf{y}\\ \mathbf{v}\end{bmatrix} \\ \text{subject to} \quad & \left\|[\mathbf{O}\ \mathbf{A}_k]\begin{bmatrix}\mathbf{y}\\ \mathbf{v}\end{bmatrix}\right\|_2 \leq [\mathbf{c}_k^T\ 0...0]\begin{bmatrix}\mathbf{y}\\ \mathbf{v}\end{bmatrix}, \end{aligned} \tag{2.62}$$

where $\mathbf{c}_k \in \mathbb{R}^{M'N'}$ contains zeros except for a 1 in the kth element and $\mathbf{O}$ is a $4 \times M'N'$ zero matrix. The tool boundary conditions can be incorporated as a further linear constraint within the optimization problem without loss of the SOCP structure.

2.2.1 Implementation

The equations for the initial indentation of the sheet in (2.47-2.49) are more difficult to rearrange as an SOCP, because there is a second order relationship between the strain rate and the spatial derivative of sheet velocity in these equations. Hence, to test the SOCP problem in (2.62), the first indentation step of the problem was solved as before using *fmincon* and subsequent steps were solved with the SOCP formulation. Since $u_{m,n}(t_1)$ is non-zero after the first indentation step, the constraint matrices $\mathbf{A}_k$ are non-zero and the problem in (2.62) is solvable. The problem was solved using the open source SOCP

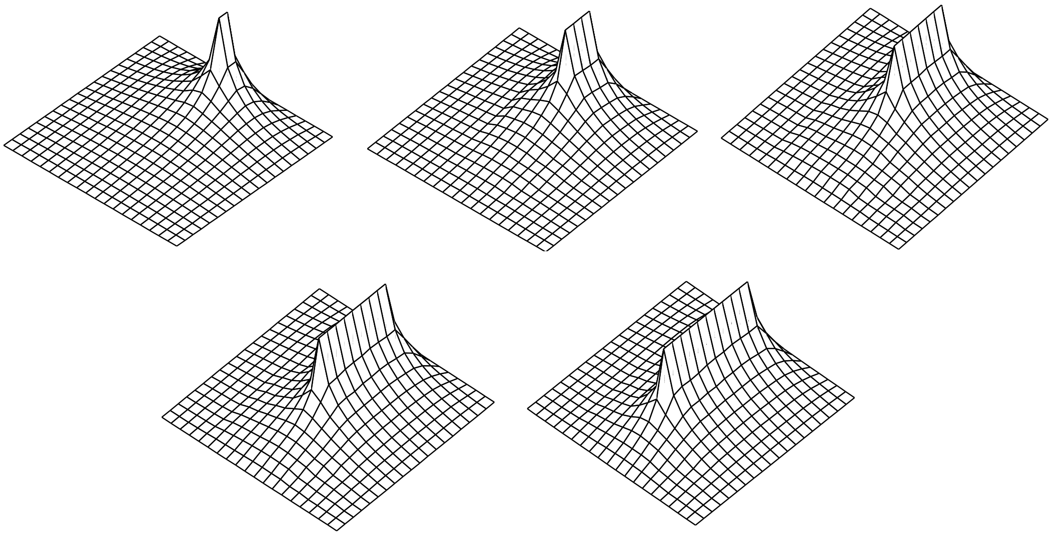


Figure 2.7: Produced by solving (2.47-2.49) for the first time step using fmincon and then solving the SOCP in (2.62) for subsequent steps.

solver SDPT3 [80], which typically required 20 iterations, or 5 seconds, to complete the minimisation at each step. The solver was called by using an optimisation problem parser, YALMIP, which contains a Matlab interface. The SOCP problems in later sections of this thesis were solved using direct calls to the more sophisticated commercial SOCP software package Mosek [80, 14, 85].

2.3 Discussion

In this chapter, a method has been described for producing a finite difference model of the incremental sheet forming process based on classical small strain shell theory. It has been shown that the model can be written as an SOCP and it has been solved for the case of a tool moving along a straight line over the sheet. This result was presented in [102], although the model does also present a number of disadvantages:

- Small time steps are required to avoid instability in the model
- The forward differencing method used for spatial discretisaion is asymmetric
- A different set of strain rate equations are required for the first indentation step, whereas a scheme that is uniform over the whole process would be more appealing.
- Only stretching, or membrane, work is considered, whilst the contribution of bending work to the overall deformation field is ignored

- The problems solved thus far are too small to compare quantitatively against benchmark FEA data in §1.5.1 or real measurements and they can therefore only be evaluated qualitatively.

At this point there are two options: the first is to search for methods to improve the current formulation of the ISF model and the second is to develop an alternative scheme for modelling ISF. The latter approach has the advantage that it provides an opportunity to expand on the depth of literature investigated thus far and furthermore, if successful, it would result in a second model that can be evaluated against the first one. This is therefore the route taken in the following chapter, where a more rigorous FE shell model for ISF based on limit analysis is developed that addresses the above problems. In particular, it results in a model that can be used to simulate the whole ISF process and can therefore be compared against the data in §1.5.1. The model combines both bending and stretching effects, but it maintains the rigid plastic assumption and again, it is formulated as an SOCP.

Chapter 3

Finite element sequential limit analysis models

3.1 Overview

This chapter presents a new method for setting up a shell model for incremental deformation and evaluates it against experimental data and numerical data from commercially available finite element software. The chapter begins with a description of bending energy in the model by formulating the upper bound plate problem as an SOCP, which builds on the work of Hodge and Belytschko [49] and Anderson *et al* [15]. A comparison is made between data from the formulation in [15] and the plate model presented here. Next, the membrane component and coupling between membrane and bending work based on Ilyushin theory [54] is described, which extends the plate problem to the full shell model. Two alternative methods for deriving the SOCP problem for shells are presented. The model is evaluated against products formed by the SPIF process and also compared to the benchmark data provided in §1.5.1 from full FE simulations using commercial software.

3.2 The plate problem

The rigid plastic assumption was often applied to metal stamping models in the 1980s and early 1990s [108], but in recent years, the dramatic improvements in processing power have prompted metal forming researchers to switch to standard elasto-plastic FEA models. However, the rigid plastic deformation assumption is still commonly used in the field of limit analysis in civil engineering and it is there that it has seen the most recent development [29]. Limit analysis is a basis for computing an estimate of the value of a multiplier that must be applied to a load distribution over a solid in order to cause collapse, under the assumption of rigid plastic deformation [15]. Since the exact value of this multiplier is difficult to compute, the upper and lower bound methods are often used to provide the range of values it may take.

For metal forming, the aim is to ensure that the material will collapse and deform, hence it is more appropriate to overestimate the collapse multiplier and guarantee collapse using the upper bound method, rather than underestimate it with the lower bound method, so the upper bound method is applied here. It is noted that unlike most applications in civil engineering, the deformation field, or collapse mechanism, of the material is of a greater interest here, rather than the accurate computation of the value of the load multiplier.

We begin by discussing the plate problem, because this allows the method for computing bending energy in the shell model to be described separately from the rest of the model. More detailed descriptions of the problem can be found in Hodge and Belyscho [49] and in Anderson *et al* [15]. Generally, plate theory is concerned with analysing the behaviour of a thin flat sheet material under transverse loading. Two important assumptions will be made in accordance to Kirchoff thin plate theory: firstly, stresses normal to the surface of the sheet are considered to be negligible; secondly, only the global deformation normal to the surface of the sheet is considered to be significant. The stress states within the plane of the sheet are denoted σ_x, σ_y, τ_{xy}, and the strain rates are $\dot{\epsilon}_x$, $\dot{\epsilon}_y$, and $\dot{\gamma}_{xy}$. The internal energy dissipation within the material is given using these stresses and strains as

$$D_e = \int_\Omega \sigma_x \dot{\epsilon}_x + \sigma_y \dot{\epsilon}_y + 2\tau_{xy}\dot{\gamma}_{xy} \ \mathrm{d}\Omega, \tag{3.1}$$

where Ω is the volume of the sheet. The strain rates are related to u and v, the in-plane velocities in the x and y directions respectively, by

$$\dot{\epsilon}_x = \frac{\partial u}{\partial x}, \quad \dot{\epsilon}_y = \frac{\partial v}{\partial y}, \quad \dot{\gamma}_{xy} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right). \tag{3.2}$$

For a thin sheet of thickness h, the velocities u and v can be approximated in terms of the transverse velocity w only by

$$u = -z\frac{\partial w}{\partial x}, \quad v = -z\frac{\partial w}{\partial y}, \tag{3.3}$$

where $z \in \left[-\frac{1}{2}h, \frac{1}{2}h\right]$. Hence

$$\dot{\epsilon}_x = -z\frac{\partial^2 w}{\partial x^2}, \quad \dot{\epsilon}_y = -z\frac{\partial^2 w}{\partial y^2}, \quad \dot{\gamma}_{xy} = -z\frac{\partial^2 w}{\partial x \partial y}.$$

To avoid through thickness integration, the stress resultant moments M_x, M_y and M_{xy} are defined as

$$M_x = -\int_z z\sigma_x \mathrm{d}z, \quad M_y = -\int_z z\sigma_y \mathrm{d}z, \quad M_{xy} = -\int_z z\sigma_{xy} \mathrm{d}z \tag{3.4}$$

and defining the curvature rates $\dot{\kappa}_x$, $\dot{\kappa}_y$ and $\dot{\kappa}_{xy}$ as work conjugate to the stress resultants,

$$\dot{\kappa}_x = \frac{\partial^2 w}{\partial x^2}, \quad \dot{\kappa}_y = \frac{\partial^2 w}{\partial y^2}, \quad \dot{\kappa}_{xy} = 2\frac{\partial^2 w}{\partial x \partial y}, \tag{3.5}$$

the dissipation may be written as an integration over area,

$$D_e = \int_A M_x\dot{\kappa}_x + M_y\dot{\kappa}_y + M_{xy}\dot{\kappa}_{xy} \ \mathrm{d}A. \tag{3.6}$$

It is assumed that the non-linearity in material behaviour can be described using J_2 plasticity [37], so that the stresses are bound by the von Mises yield surface,

$$f(\sigma_x, \sigma_y, \tau_{xy}) = \sigma_x^2 - \sigma_x\sigma_y + \sigma_y^2 + 3\tau_{xy}^2 - \sigma_o^2 \leq 0, \tag{3.7}$$

where σ_o is the material yield stress and $f(\sigma_x, \sigma_y, \tau_{xy}) = 0$ at yield. If rigid plastic material behaviour is considered, it is possible to replace the above stresses by their corresponding stress resultants, since the stress distributions are rectangular during yield [30]. Hence, the stress resultants take the constant scalar values

$$M_x = \sigma_x\frac{h^2}{4}, \quad M_y = \sigma_y\frac{h^2}{4}, \quad M_{xy} = \tau_{xy}\frac{h^2}{4}, \quad M_0 = \sigma_o\frac{h^2}{4}, \tag{3.8}$$

and

$$f(M_x, M_y, M_{xy}) = M_x^2 - M_xM_y + M_y^2 + 3M_{xy}^2 - M_0^2, \tag{3.9}$$

or by normalising the stress resultants by M_0, so that $m_x = \frac{M_x}{M_0}$ etc,

$$\begin{aligned} f(m_x, m_y, m_{xy}) &= m_x^2 - m_xm_y + m_y^2 + 3m_{xy}^2 - 1 \\ &= \begin{bmatrix} m_x & m_y & m_{xy} \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_{xy} \end{bmatrix} - 1 \\ &= \mathbf{m}^T\mathbf{H}\mathbf{m} - 1. \end{aligned} \tag{3.10}$$

If required, anisotropic material behaviour can be included by modifying $\mathbf{H}$ using Hill anisotropy [120]. In either case, $\mathbf{H}$ is symmetric in structure and for the value of $\mathbf{H}$ in (3.10) the factorisation $\mathbf{H} = \mathbf{C}^{-1}\mathbf{C}^{-T}$ is defined, where for instance

$$\mathbf{C} = \frac{1}{\sqrt{3}} \begin{bmatrix} \sqrt{3} & 0 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.11}$$

From the principle of maximum plastic work the dissipation, D_e, when constrained by the yield function in (3.10), is

$$D_I = M_0 \int_A d_I \mathrm{d}A, \tag{3.12}$$

where

$$\begin{aligned} d_I &= \max_{\mathbf{m}^T\mathbf{H}\mathbf{m}\leq 1} \mathbf{m}^T\boldsymbol{\kappa}, \\ &= \max_{\|\mathbf{C}^{-T}\mathbf{m}\|\leq 1} \mathbf{m}^T\boldsymbol{\kappa}, \end{aligned} \tag{3.13}$$

and the subscript I is used instead of e to emphasise that the allowable stress state is bound by the yield criterion. The dual of (3.13), in terms of dual variables λ, is

$$d_I = \min_{\lambda\geq 0} \quad \frac{1}{4\lambda}\boldsymbol{\kappa}^T\mathbf{H}^{-1}\boldsymbol{\kappa} + \lambda, \tag{3.14}$$

which is minimised by setting the derivative with respect to λ to zero, giving

$$\lambda = \frac{1}{2}(\boldsymbol{\kappa}^T\mathbf{H}^{-1}\boldsymbol{\kappa})^{\frac{1}{2}} = \frac{1}{2}\|\mathbf{C}\boldsymbol{\kappa}\|. \tag{3.15}$$

By back substituting the expression into equation (3.14),

$$\begin{aligned} d_I =& \frac{2}{4}\frac{\boldsymbol{\kappa}^T\mathbf{H}^{-1}\boldsymbol{\kappa}}{\|\mathbf{C}\boldsymbol{\kappa}\|} + \frac{1}{2}\|\mathbf{C}\boldsymbol{\kappa}\|, \\ =& \|\mathbf{C}\boldsymbol{\kappa}\|, \end{aligned} \tag{3.16}$$

where $\mathbf{H}^{-1} = \mathbf{C}^T\mathbf{C}$, so the internal dissipation under von Mises plasticity can be written as the integration over area of the norm of a linear function with respect to curvature,

$$D_I = M_0 \int_A ||\mathbf{C}\boldsymbol{\kappa}||\mathrm{d}A. \tag{3.17}$$

Equation (3.17) was discretised by Anderson *et al* [15] using integration by parts and then using bilinear quadrilateral finite elements to discretise both the stress resultant moments and the transverse velocity field. This leads to a solution that is not strictly an upper bound, since it is possible for the method to underestimate the value of the limit load multiplier [74]. The solution does however tend to the upper bound with increasing mesh density. An alternative solution method is employed here, which is motivated by the need to maintain a strict upper bound scheme and to avoid discretising the stress resultants. The latter is important for the development of the shell problem. The transverse velocity, $w(x, y)$, is discretised by a cubic field in the form of the 9 degrees of freedom triangular, non-conforming BCIZ element, which is described in detail in [122]. There are several other choices of elements that have the required characteristics for discretising the model, although it is not the purpose of this thesis to compare these. The element has nodes only at the vertices, with three variables, w_i, $\frac{\partial w_i}{\partial y}$ and $-\frac{\partial w_i}{\partial x}$, being stored at each node, where i refers to the node number, 1, 2, or 3 and Hermitian interpolation is used to approximate the value of $w(x, y)$ at any point within the element. The interpolation field is cubic with additional bubble functions and is given by the shape functions provided in [40]. Through differentiation of the appropriate shape functions, a linear matrix relationship is established between the curvature and the nodal variables $\mathbf{w}$. For a particular node i where $(i = 1, \ldots, 3)$ within an element (e) where $\{e = 1, \ldots, NE\}$ and NE is the total number of elements, the curvature is given by

$$\boldsymbol{\kappa}_i^{(e)} = \mathbf{B}_{\mathrm{P},i}^{(e)} \mathbf{w}, \tag{3.18}$$

where $\mathbf{B}_{\mathrm{P},i} \in \mathbb{R}^{3 \times NV}$ contains the coefficients of this relationship, NV is the number of variables and $\mathbf{w} \in \mathbb{R}^{NV}$ contains all the degrees of freedom over the sheet. The subscript P is used to emphasise that $\mathbf{B}$ is derived from plate theory and superscripts (e) refer to element number. The expression for curvature in equation (3.18) can be used to calculate the integrand in equation (3.17) at the nodes, since $\|\mathbf{C}\boldsymbol{\kappa}_i\| = \|\mathbf{C}\mathbf{B}_{\mathrm{P},i}\mathbf{w}\|$. An exact integration of $\|\mathbf{C}\boldsymbol{\kappa}\|$ over the area of the element would be costly and is avoided by applying an approximate integration that is calculated using the values of the integrand, $\|\mathbf{C}\boldsymbol{\kappa}\|$, at the nodes and restricting it to vary linearly over the element,

$$\|\mathbf{C}\boldsymbol{\kappa}\| \approx L_1\|\mathbf{C}\boldsymbol{\kappa}_1\| + L_2\|\mathbf{C}\boldsymbol{\kappa}_2\| + L_3\|\mathbf{C}\boldsymbol{\kappa}_3\|, \tag{3.19}$$

where $(L_1,\ L_2,\ L_3)$ are element coordinates using the area coordinate convention for triangular elements [123] defined in (3.22), so that the integration of equation (3.19) over the element becomes

$$\int_A \|\mathbf{C}\boldsymbol{\kappa}\| \mathbf{d}A \approx \frac{\Delta}{3}(\|\mathbf{C}\boldsymbol{\kappa}_1\| + \|\mathbf{C}\boldsymbol{\kappa}_2\| + \|\mathbf{C}\boldsymbol{\kappa}_3\|), \tag{3.20}$$

where Δ is the area of the triangular element, given by

$$\Delta = \frac{1}{2} \det \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}, \tag{3.21}$$

where (x_1, y_1), (x_2, y_2) and (x_3, y_3) are the cartesian nodal coordinates of the element. With reference to figure 3.1, the L_1 coordinate of a point P within an element can be defined as the ratio of the area of the shaded region $areaP23$ to that of the whole element Δ, and similarly for L_2 and L_3, so that

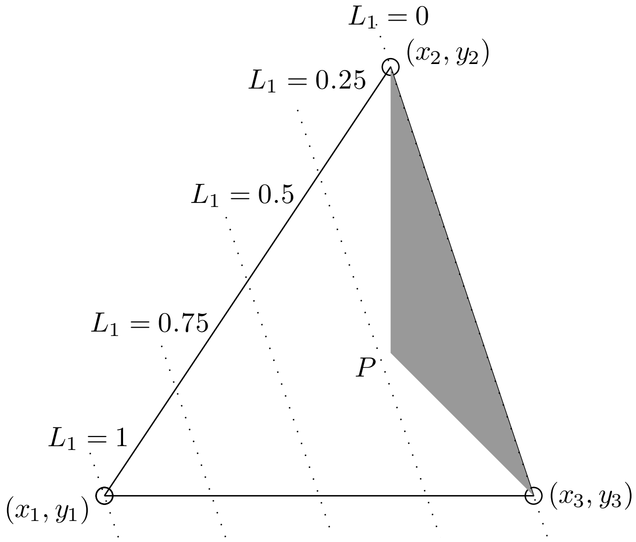


Figure 3.1: Area coordinates [123].

$$L_1 = \frac{areaP23}{\Delta}, \quad L_2 = \frac{areaP31}{\Delta}, \quad L_3 = \frac{areaP12}{\Delta}. \tag{3.22}$$

To visualise this, several loci of constant L_1 have been drawn as dotted lines over the element in figure 3.1.

The assumption in (3.20) is suitable for this formulation since, as has been shown by Makrodimopoulos and Martin [73, 74] in the context of 2D plane strain problems, equation (3.20) always results in a value greater than or equal to the true value of the integration, thus guaranteeing an upper bound solution. The discretised expression for internal dissipation over the sheet is a sum of norms,

$$D_I = \frac{M_0}{3} \sum_{e}^{NE} \sum_{i=1}^{3} \Delta \|\mathbf{C}\mathbf{B}_{\mathrm{P},i}\mathbf{w}\|, \tag{3.23}$$

where NE is the total number of elements in the finite element mesh. By the principle of virtual work, the velocity field that minimises the internal dissipation provides the dynamical path of the system. For the first step of an incremental sheet forming process, the sheet metal is clamped at the edges and a tool of spherical surface indents the sheet at a single point, leading to the following formulation:

$$\begin{aligned}\min_{\mathbf{w}} \;& D_I \\ \text{s.t.} \;& \mathbf{w}_{tool} \geq \mathbf{t} \\ & \mathbf{w}_{boundary} = 0\end{aligned} \tag{3.24}$$

$\mathbf{w}_{boundary}$ contains elements from $\mathbf{w}$ that form the set of variables that are constrained at the boundaries. These can be implemented implicitly by removing the corresponding variables from the optimisation. The tool surface function describes the tool height at the tool position (x, y) and produces a vector $\mathbf{t} \in \mathbb{R}^{NT}$ of heights at positions over the tool surface that coincide with nodes of the finite element mesh. The deformation variables in $\mathbf{w}_{tool}$, which are taken from the vector $\mathbf{w}$, also correspond to these nodes. Strictly, the tool constraint above should be stated as

$$\omega_{tool} + \mathbf{w}_{tool}\mathrm{d}t \geq \mathbf{t}, \tag{3.25}$$

where ω_{tool} is the height of the nodes before the load is applied and dt is the time period over which the tool acts. Taking the plane of the sheet as reference, all points over the sheet are initially at zero height, hence $\omega_{tool} = 0$. Also, in the context of limit analysis the plate problem is static and independent of the period over which the load is applied, as long as the tool-path is discretised by small enough steps to remain sufficiently smooth. Hence, under this condition it will be assumed d$t = 1$ without loss of generality.

The above sum of norms problem can be rearranged into an SOCP. Firstly, (3.24) is restated as

$$\begin{aligned}\min_{\mathbf{w}} \;& f \sum_{e}^{NE} \sum_{i=1}^{3} \Delta \|\mathbf{C}\mathbf{B}_{\mathrm{P},i}\mathbf{w}\| \\ \text{s.t.} \;& \mathbf{E}\mathbf{w} \geq \mathbf{t} \\ & \mathbf{w}_{boundary} = 0,\end{aligned} \tag{3.26}$$

where the clamping constraints are implicit, $f = \frac{M_0}{3}$ and $\mathbf{E}\mathbf{w} = \mathbf{w}_{tool}$. Two vectors of slack variables, $\mathbf{n} \in \mathbb{R}^{9NE}$ and $\mathbf{z} \in \mathbb{R}^{3NE}$ are introduced such that

$$\mathbf{z} \geq \|\mathbf{CB}_{\mathrm{P},i}\mathbf{w}\|, \qquad \mathbf{n}_i = \mathbf{CB}_{\mathrm{P},i}\mathbf{w}, \tag{3.27}$$

where $\mathbf{n}_i = [n_{1+3(i-1)},\ n_{2+3(i-1)},\ n_{3+3(i-1)}]$, n_i are individual elements of their corresponding vectors and the element area Δ in (3.26) has been incorporated into $\mathbf{B}_{\mathrm{P}}$. The optimisation problem in (3.26) can be expressed in terms of these slack variables as

$$\begin{aligned} \min\ & \mathbf{f}^T\mathbf{z} \\ \text{s.t. } & \mathbf{Ew} \geq \mathbf{t} \\ & \mathbf{CB}_{\mathrm{P}}\mathbf{w} = \mathbf{n} \\ & \|\mathbf{n}_i\| \leq z_i \qquad i = 1, \ldots, 3NE, \end{aligned} \tag{3.28}$$

where at the optimum, $z_i = \|\mathbf{n}_i\| = \|\mathbf{CB}_{\mathrm{P},i}\mathbf{w}\|$. In (3.28), the matrices $\mathbf{B}_{\mathbf{p},i}$ have been conveniently stacked into a single stiffness matrix $\mathbf{B}_{\mathbf{p}} \in \mathbb{R}^{9NE \times NV}$ and the vector $\mathbf{f} = [f\ \ f \ldots f]$. The resulting problem is an SOCP with respect to the optimisation variables, $\mathbf{z}$, $\mathbf{n}$ and $\mathbf{w}$, where the third set of constraints in (3.28) are second-order cone constraints, since from equation (1.15),

$$(\mathbf{n}_i, z_i) \in \Theta_i \Leftrightarrow \{(\mathbf{n}_i, z_i) \mid \|\mathbf{n}_i\| \leq z_i\}. \tag{3.29}$$

The model was scripted in Matlab using the optimiser Mosek [85] to solve the SOCP. A solution of the problem when applied to a pinned square sheet of unit dimensions with a central point load is displayed in figure 3.2. When the problem is set up in the form of (3.28), the solution time is approximately 5 minutes using $NE = 5000$ on a standard desktop computer, but this is significantly reduced to just 8 seconds by exploiting the strong duality of the problem and solving the dual optimisation problem,

$$\begin{aligned} D^{opt}_{I,dual} = D^{opt}_{I,primal} = \max \quad & \mathbf{t}^T \boldsymbol{\lambda} \\ \text{s.t.} \quad & \mathbf{E}^T \boldsymbol{\lambda} - \mathbf{B}_{\mathrm{P}}^T \mathbf{C}^T \boldsymbol{\rho} = 0, \\ & \mathbf{f} - \boldsymbol{\mu} = 0, \\ & \boldsymbol{\lambda} \geq 0, \\ & (\boldsymbol{\rho}_i, \mu_i) \in \Theta^*_i, \qquad i = 1, \ldots, 3NE, \end{aligned} \tag{3.30}$$

where $\boldsymbol{\mu}$, $\boldsymbol{\rho}$ and $\boldsymbol{\lambda}$ are the dual optimisation variables. In the dual problem, a general conic set, Θ_i, becomes the conic conjugate, Θ^*_i. However, as a result of the self-duality property of second-order cones the conjugate of a second-order cone is the cone itself, hence $\Theta_i = \Theta^*_i$ where Θ_i is the set defined in equation (3.29). Inspection of equation (3.30) shows that the number of variables in the objective function is equal to the length of the tool constraint vector, NT, which is much lower than the number of variables in the primal problem. Indeed, for the point loaded case with $NE = 5000$, the dual objective function contains only one variable since $NT = 1$ as opposed to $3NE = 15000$ variables in the primal problem, which explains the significant reduction in solution time when solving the problem using the dual.

Figure 3.3 provides a comparison of the limit load multiplier for the problem in figure 3.2 predicted by the model presented in this paper to that of Anderson *et al* [15]. As described in [15], the plate formulation by Anderson *et al* applied integration by parts to the constitutive equation in (3.6). This resulted in first order differential terms with respect to the stress state, **m** and the sheet velocity **w**, both of which were then discretised using linear finite elements, but as a result strict upper boundedness is lost [15]. The formulation in [15] was re-coded in Matlab for the purpose of comparing it with the plate model presented here. In doing so, our implementation differed from the original description in two minor aspects. Firstly, the finite element discretisation was performed using linear simplex, or triangular, elements rather than the original bilinear quadrilateral elements used by Anderson *et al*; an alteration described as insignificant in [15]. Secondly, the original sum-of-norms formulation in [15] was rewritten as an SOCP, so that MOSEK could be used

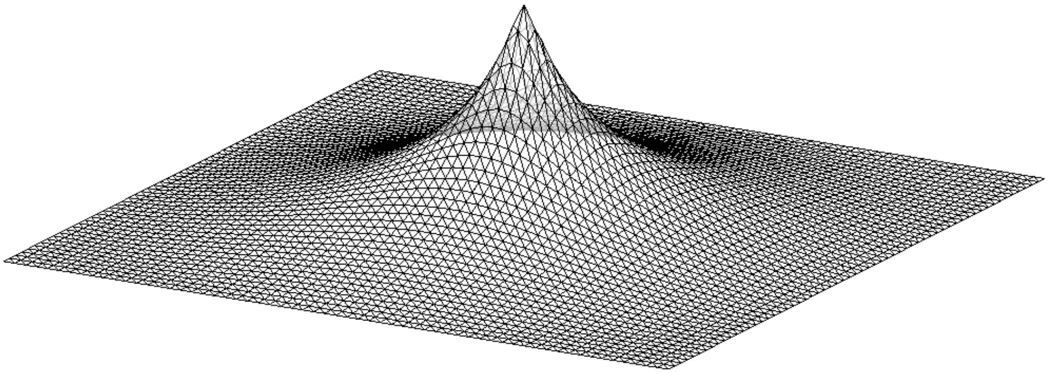


Figure 3.2: Deformation field of point loaded sheet obtained by solving an upper bound plate problem as an SOCP

for solution.

The computations were performed to the capacity of the desktop computer used. As expected, both plots converge towards the same upper bound limit load with increasing mesh density but from different directions. It is apparent that the method presented in this paper always overestimates the limit load and it therefore yields a true upper bound solution. This overestimate reduces with increasing mesh size towards a minimum upper bound.

As an interesting side investigation, it is noted that a better estimate of the minimum upper bound may be obtained by enforcing compatibility along the edge of each finite element as well as at the nodes. This would remove the possibility of solutions with spurious energy modes, which may be difficult to detect if, for instance, they are present in only a few elements. Such modes have not actually been observed here but no guarantee has been provided either here or in [15] that they do not exist. An investigation into locking mechanisms in the finite element mesh may also be beneficial to this extent.

3.3 Some solutions using the plate model

3.3.1 Alternative boundary conditions

Before proceeding to the shell problem, a short diversion is provided here to demonstrate how the plate model behaves under different boundary conditions to those of the basic SPIF process. Nine solutions of the plate problem for different tool and boundary conditions are shown in figure 3.4. These include the conditions for the incremental curving setup in

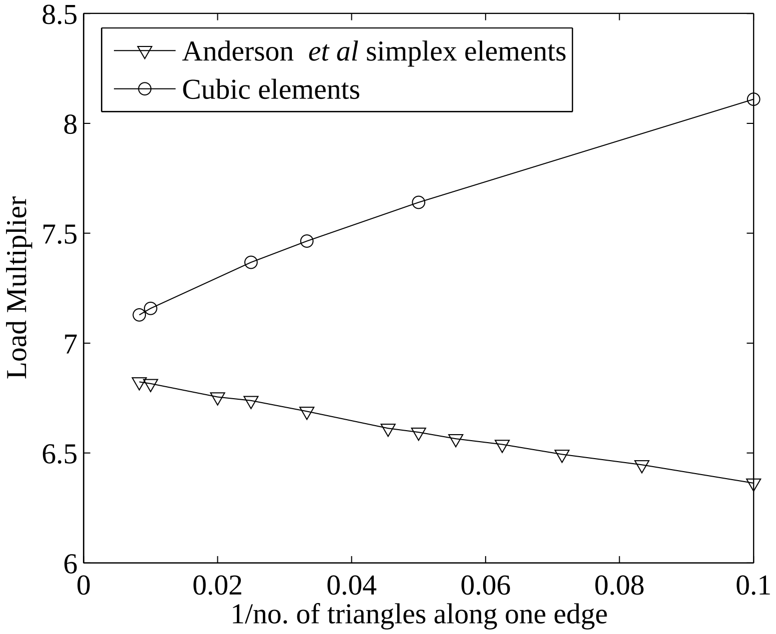


Figure 3.3: Comparison with [15] of limit loads for a point loaded pinned sheet of unit dimensions. Limit loads obtained by an implementation of [15] are labelled ‘Anderson *et al* simplex elements’ and from the model in this thesis, ‘Cubic elements’.

figure 1.7 (top left of figure 3.4) with unconstrained edges, a uniformly loaded clamped plate (bottom left of figure 3.4) and a sheet with two edges rotated (bottom middle of figure 3.4). By using an irregular FE mesh, it is also possible to model different shaped clamping supports. For instance in figure 3.5, the deformation field for a point loaded plate constrained by a circular clamping frame is displayed.

3.3.2 Brief return to the finite difference model

The FD model in §2.1 is also briefly revisited in this section. This model was solved using one set of equations for the initial indentation of the sheet and another set, in the form of an SOCP, for the remaining tool position increments. Here, the solution of the plate bending problem in (3.30) is used to provide the first update $u_{m,n}(t_1)$ in the deformation field before proceeding with the solution of the finite difference problem in (2.62) for subsequent time steps. This appears to introduce an inconsistency because the first indentation step assumes only bending work is contributing to deformation, whereas subsequent steps deform due to pure stretching work. However, since the whole problem has an SOCP structure, this investigation allows (2.62) to be tested on a larger problem with a finer mesh compared to the one previously used.

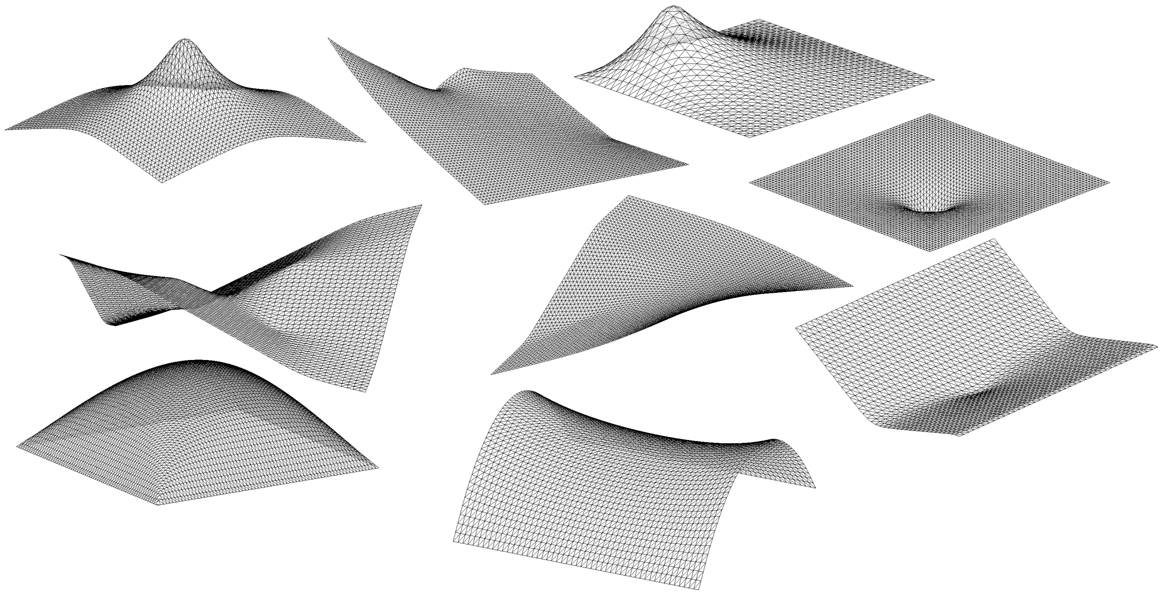


Figure 3.4: Solutions of the plate problem under different boundary conditions ranging from pinned, clamped or free edges to uniform loading and multiple tools.

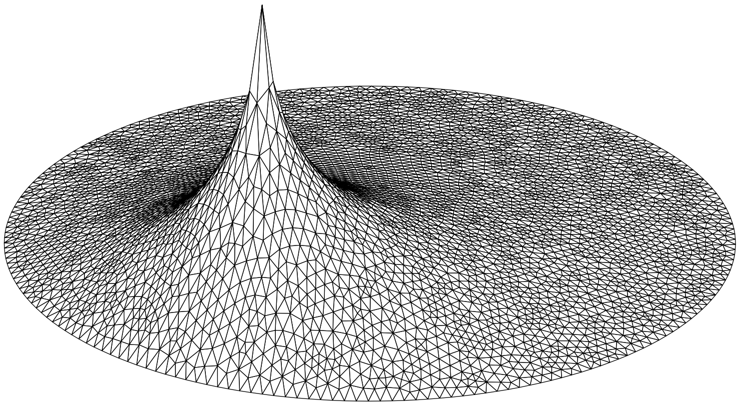


Figure 3.5: Deformation of a point loaded circular plate discretised with approximately 6900 triangluar elements with pinned boundary constraints at the edges.

Figure 3.6 shows an example of the the initial step, for an indentation of a spherical tool into a flat sheet. The dimensions of the sheet are 100 mm×100 mm with a thickness of 1 mm and yield stress of $\sigma_y = 15$ MPa. The number of nodes in the FD model in the x and y directions, M and N, are chosen to be 50 and a time-step of $\Delta t = 0.001$ seconds is used. The spherical tool-tip is 10 mm in radius with an indentation depth of 10 mm. The tool is tracked along in a straight line with a velocity of 0.4 ms^{-1} in the $[-1\ 0]$ direction within the (x,y) plane and the solution of the model after 60 time steps is shown in figure 3.7. Although the general deformation field appears as expected, there are some fluctuations in height along the line of the tool-path, which were not previously observed using the

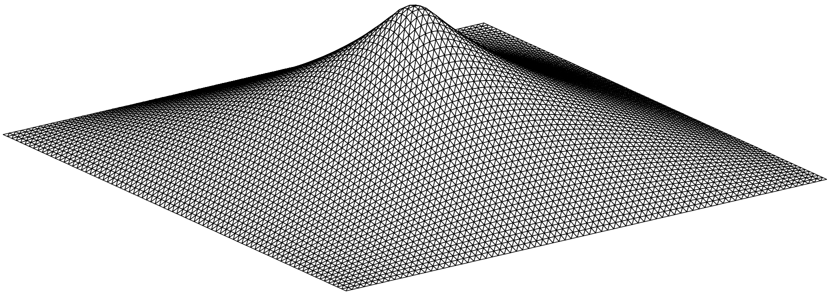


Figure 3.6: Example of the initial deformation field $u_{m,n}(t_1)$ due to indentation by a spherical tool obtained by solving the plate problem.

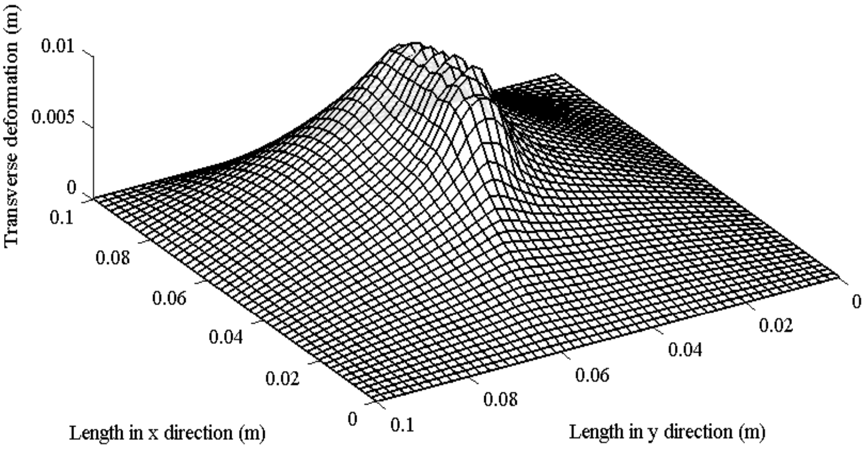


Figure 3.7: First test case: tool path along the x direction solved with 60 time-steps.

coarser mesh. A second test case is also shown using the same parameters as before, but with a tool velocity of 0.45 ms^{-1} in the [1 -2] direction in the (x, y) plane. This test case is of interest because it shows the deformation field that is obtained when the tool-path is not aligned with the grid used for spatial discretisation. The solution, which is shown in figure 3.8 after 100 time increments, provides similar results to figure 3.7, with a smooth deformation field and some fluctuations in height as before.

This fluctuation was found to reduce with decreasing time step size, Δt, suggesting that the ratio between the step size and mesh density is a significant parameter. The likely explanation for this is due to the small angle approximations made in formulating the problem in §2.1, since such approximations only hold for small changes in the deformation field and hence small increments in the tool position. However, reducing Δt is undesirable in

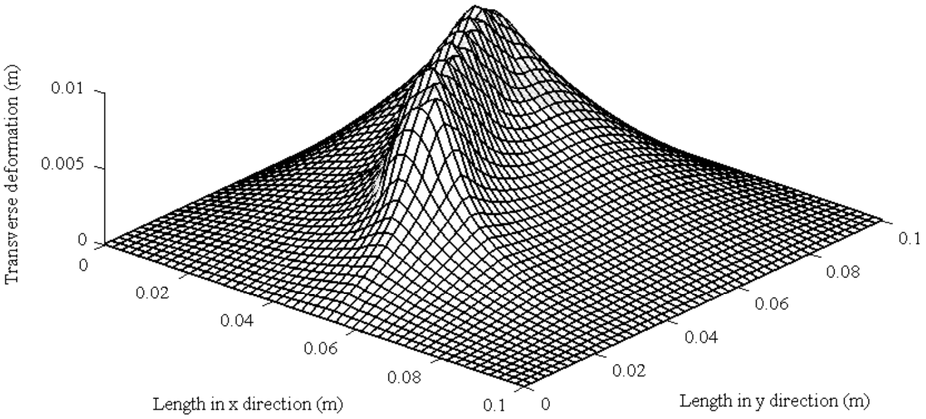


Figure 3.8: Second test case: diagonal tool paths solved with 100 time-steps.

terms of computational efficiency, so a compromise must be made between model accuracy and solution time when selecting a suitable time step. This example of combining the limit analysis model with the FD model was presented in [98].

3.4 Membrane work and coupling

In order to apply the limit analysis approach to a forming process such as ISF, two factors are considered: the dynamic loading problem and the extension from plates to shells. To address the first issue, the single step limit analysis problem is extended using the sequential limit analysis method to handle the effect of dynamic loading caused by a moving tool. This intuitive method involves progressively solving a limit analysis problem for a structure, updating the deformation field and loading conditions by explicit time-stepping, then solving the next limit analysis problem using the updated configuration. Sequential limit analysis has gained interest in recent years as a method for analysing the evolution of the deformation mechanism of a structure after initial collapse and has been discussed by several authors including Yang [118], Hwan [53] and Leu [70]. More recently Kim and Huh [63] have introduced the concept of dynamic limit analysis and applied it to two impact problems. During sequential limit analysis, the non-linear effect of material work hardening can be incorporated in the analysis by updating local yield stress of a finite element in relation to its deformation history [70]. This section presents an investigation of the application of the technique of sequential limit analysis to modelling the incremental deformation of sheet metal.

3.4.1 Transformation between local and global frames

To extend the analysis from plates to shells, it is necessary to allow for the fact that the finite elements that lie within a shell do not all share a common plane, but can instead be oriented arbitrarily in space with nodes at the global coordinates (X_i, Y_i, Z_i), for $i = 1, 2, 3$. The bending work for a particular element can be computed using the plate formulation in the previous section by defining velocities that operate within a local in-plane coordinate frame with respect to the element concerned. A transformation is then performed to align these velocities to the global reference frame. The coordinates in the local frame will be expressed by the lower case equivalents of the global system.

The desired transformation is shown in figure 3.9, where a convention has been chosen whereby the line joining node 1 and 2 is defined as the local x axis and the origin of the local axes is at node 1 [122]. If the vector $\mathbf{V}_{21}$ is defined as

$$\mathbf{V}_{21} = [X_{21} \;\; Y_{21} \;\; Z_{21}]^T, \tag{3.31}$$

and similarly for vector $\mathbf{V}_{31}$ where $X_{ij} = X_i - X_j$ etc, then the unit vector along the x axis is given by

$$\mathbf{v}_x = \frac{\mathbf{V}_{21}}{|\mathbf{V}_{21}|}, \tag{3.32}$$

the unit vector along the z axis is

$$\mathbf{v}_z = \frac{\mathbf{V}_{21} \times \mathbf{V}_{31}}{|\mathbf{V}_{21} \times \mathbf{V}_{31}|} = \frac{\mathbf{V}_{21} \times \mathbf{V}_{31}}{2\Delta}, \tag{3.33}$$

and along the y axis, it is

$$\mathbf{v}_y = \mathbf{v}_z \times \mathbf{v}_x. \tag{3.34}$$

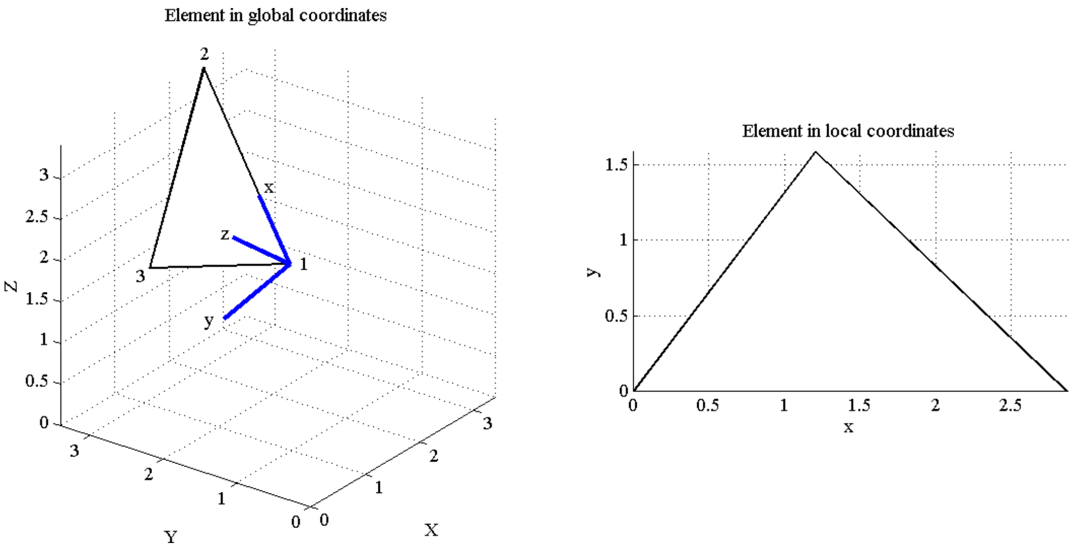


Figure 3.9: An element shown in a global and a local coordinate frame.

The transformation matrix from global coordinates is given by

$$\begin{aligned}
\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} &= \begin{bmatrix} \cos(x,X) & \cos(x,Y) & \cos(x,Z) \\ \cos(y,X) & \cos(y,Y) & \cos(y,Z) \\ \cos(z,X) & \cos(z,Y) & \cos(z,Z) \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}, \\
&= \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \lambda_4 & \lambda_5 & \lambda_6 \\ \lambda_7 & \lambda_8 & \lambda_9 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}, \\
&= \begin{bmatrix} \mathbf{v}_x^T \\ \mathbf{v}_y^T \\ \mathbf{v}_z^T \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}, \\
\mathbf{p}_i &= \mathbf{\Lambda} \mathbf{P}_i, \qquad (3.35)
\end{aligned}$$

where $\cos(x, X)$ denotes the cosine of the angle between the local x axis and global X axis. Since the local axis is in the plane of the element, the local z coordinate at each node is zero, and indeed the only non-zero coordinates are x_2, x_3 and y_3. It is also noted that since $\mathbf{\Lambda}$ is orthogonal, i.e. $\mathbf{\Lambda}^{-1} = \mathbf{\Lambda}^T$, the inverse transformation is straightforward.

3.4.2 Membrane work

The membrane, or in-plane stretching, work component in a shell is considered independently to the plate bending effects. Firstly, three further normalised stress resultants n_x, n_y and n_{xy} are introduced, which are defined as,

$$n_x = \frac{1}{N_o}\int_z \sigma_x dz, \quad n_y = \frac{1}{N_o}\int_z \sigma_y dz, \quad n_{xy} = \frac{2}{N_o}\int_z \sigma_{xy} dz, \tag{3.36}$$

where $N_o = \sigma_o h$ and σ_o is the material yield stress. The dissipation due to in-plane stretching is then given by

$$D_I = N_o \int_A n_x \dot{\epsilon}_x + n_y \dot{\epsilon}_y + n_{xy} \dot{\gamma}_{xy} \ dA, \tag{3.37}$$

where the strain rates are calculated from the in plane velocities u and v using the expressions in equation (3.2). A simple shell finite element is produced by superimposing a bending element, such as the BCIZ element [122], with a membrane element [30]. Note that it has been assumed that the shell geometry can be suitably approximated by flat, rather than curved, triangular elements. In this case, the Allman element [3] is used to discretise the in-plane velocity fields and to compute membrane strains. As with the BCIZ element, this element has 9 degrees of freedom and three nodes, one at each vertex. The in-plane velocities u_i, v_i, and in-plane rotations $\theta_{z,i}$, are stored at each node $i = 1, 2, 3$. Within the local coordinate frame, the in-plane strains are related to these nodal velocities via a linear matrix relationship, so that for a node i of an element e within a finite element mesh,

$$\boldsymbol{\epsilon}_i^{(e)} = \mathbf{B}_{\mathrm{M},i}^{(e)} \mathbf{u}, \tag{3.38}$$

where $\boldsymbol{\epsilon}_i = [\dot{\epsilon}_{x,i} \ \dot{\epsilon}_{y,i} \ 2\dot{\tau}_{xy,i}]$, $\mathbf{B}_{\mathrm{M},i}^{(e)} \in \mathbb{R}^{3\times NV}$ can be obtained from [3], and $\mathbf{u} \in \mathbb{R}^{NV}$ contains all of the in-plane variables for the element. Combining the plate and membrane components of strain [30],

$$\begin{bmatrix} \boldsymbol{\kappa}_i \\ \boldsymbol{\epsilon}_i \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{\mathrm{P},i}^{(e)} & \mathbf{O} \\ \mathbf{O} & \mathbf{B}_{\mathrm{M},i}^{(e)} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix}. \tag{3.39}$$

In order to relate the local variables $\mathbf{w}$ and $\mathbf{u}$ to a global coordinate frame, $\mathbf{W}$ and $\mathbf{U}$, a transformation matrix $\mathbf{T}^{(e)}$ is computed for each element from the terms in $\boldsymbol{\Lambda}$ in (3.35). Equation (3.39) can then be written as

$$\begin{bmatrix} \boldsymbol{\kappa}_i \\ \boldsymbol{\epsilon}_i \end{bmatrix} = \begin{bmatrix} \mathbf{B}^{(e)}_{\mathrm{P},i} & \mathbf{O} \\ \mathbf{O} & \mathbf{B}^{(e)}_{\mathrm{M},i} \end{bmatrix} \mathbf{T} \begin{bmatrix} \mathbf{W} \\ \mathbf{U} \end{bmatrix}. \tag{3.40}$$

For the plate problem, the von Mises criterion was represented by the bending stress resultants only, but now the membrane components must also be incorporated, which can be achieved by using Ilyushin theory [54]. By defining the vectors,

$$\begin{aligned} \mathbf{M} =& M_0[m_x \ m_y \ m_{xy}]^T = [M_x \ M_y \ M_{xy}]^T, \\ \mathbf{N} =& N_0[n_x \ n_y \ n_{xy}]^T = [N_x \ N_y \ N_{xy}]^T, \\ \mathbf{s} =& [\mathbf{M}^T \mathbf{N}^T]^T, \end{aligned}$$

the Ilyushin criterion can be stated in terms of the quadratic stress intensities, Q_P, Q_M and Q_{MP},

$$Q_P = m_x^2 - m_x m_y + m_y^2 + 3m_{xy}^2, \tag{3.41}$$

$$Q_M = n_x^2 - n_x n_y + n_y^2 + 3n_{xy}^2, \tag{3.42}$$

$$Q_{MP} = n_x m_x - \frac{1}{2}(n_x m_y + n_y m_x) + n_y m_y + 3n_{xy} m_{xy}, \tag{3.43}$$

or

$$Q_P = \frac{1}{M_0^2}\mathbf{M}^T\mathbf{H}\mathbf{M}, \quad Q_M = \frac{1}{N_0^2}\mathbf{N}^T\mathbf{H}\mathbf{N}, \quad Q_{MP} = \frac{1}{N_0 M_0}\mathbf{N}^T\mathbf{H}\mathbf{M} = \frac{1}{N_0 M_0}\mathbf{M}^T\mathbf{H}\mathbf{N}, \tag{3.44}$$

where, as with (3.10), for the von Mises criterion,

$$\mathbf{H} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & \sqrt{3} \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & \sqrt{3} \end{bmatrix} = \mathbf{C}^{-1}\mathbf{C}^{-T}. \tag{3.45}$$

The Ilyushin criterion requires that stress resultants must lie within the surface defined by the inequality [26]

$$Q_{MP}^2 \leq Q_P Q_M, \tag{3.46}$$

which becomes an equality at yield and is usually approximated by a linear equation that was also provided by Ilyushin [26],

$$Q_P + \frac{1}{\sqrt{3}}|Q_{MP}| + Q_M \leq 1. \tag{3.47}$$

Equation (3.47) can be written in the form of two intersecting ellipsoids

$$\mathbf{s}^T \mathbf{Q}_1 \mathbf{s} \leq 1, \quad \mathbf{s}^T \mathbf{Q}_2 \mathbf{s} \leq 1, \tag{3.48}$$

where $\mathbf{Q}_i$ is

$$\mathbf{Q}_i = \begin{bmatrix} \frac{\mathbf{H}}{M_0^2} & (-1)^{i-1}\frac{1}{2\sqrt{3}}\frac{\mathbf{H}}{M_0 N_0} \\ (-1)^{i-1}\frac{1}{2\sqrt{3}}\frac{\mathbf{H}}{M_0 N_0} & \frac{\mathbf{H}}{N_0^2} \end{bmatrix}, \qquad i = 1, 2. \tag{3.49}$$

From the principle of maximum plastic work,

$$\begin{aligned} d_I = \max_{\mathbf{s}} \ & \mathbf{s}^T \boldsymbol{\Phi} \\ \text{s.t.} \ & \mathbf{s}^T \mathbf{Q}_1 \mathbf{s} \leq 1 \\ & \mathbf{s}^T \mathbf{Q}_2 \mathbf{s} \leq 1, \end{aligned} \tag{3.50}$$

where $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\kappa}^T & \boldsymbol{\epsilon}^T \end{bmatrix}^T$. The form of equation (3.50) is similar to the equivalent expression for the plate problem in equation (3.13), except for the additional complication of two constraints representing the yield function rather than one. Two alternative derivations of the numerical minimisation problem are now presented, both of which derive from (3.50). The first is described in §3.5, where the SOCP is derived directly from (3.50) without making any changes to the Ilyushin criterion. In §3.6, a relaxation technique is applied to reduce the two constraints to one.

3.5 The ‘two cone’ model: derivation using full linearised Ilyushin criterion

The maximum plastic work expression in (3.50) can be restated as

$$\begin{aligned} d_I = \max_{\mathbf{s}} \ & \mathbf{s}^T \mathbf{\Phi} \\ \text{s.t.} \ & \|\mathbf{P}_1 \mathbf{s}\| \leq 1 \\ & \|\mathbf{P}_2 \mathbf{s}\| \leq 1, \end{aligned} \tag{3.51}$$

where $\mathbf{Q}_i = \mathbf{P}_i^T \mathbf{P}_i$ and $i = 1, 2$. From the Lagrangian of (3.51),

$$\begin{aligned} L &= -\mathbf{s}^T \mathbf{\Phi} - \boldsymbol{\eta}_1^T \mathbf{P}_1 \mathbf{s} - r_1 - \boldsymbol{\eta}_2^T \mathbf{P}_2 \mathbf{s} - r_2, \\ &= -\mathbf{s}^T (\mathbf{\Phi} + \mathbf{P}_1^T \boldsymbol{\eta}_1 + \mathbf{P}_2^T \boldsymbol{\eta}_2) - r_1 - r_2, \end{aligned} \tag{3.52}$$

the dual of (3.51) in terms of the Lagrange multipliers $\boldsymbol{\eta}_1$, $\boldsymbol{\eta}_2$, r_1 and r_2 is formed,

$$\begin{aligned} d_I = \min_{r_{1,2}, \boldsymbol{\eta}_{1,2}} \ & r_1 + r_2 \\ \text{s.t.} \ & \mathbf{\Phi} = -\mathbf{P}_1^T \boldsymbol{\eta}_1 - \mathbf{P}_2^T \boldsymbol{\eta}_2 \\ & \|\boldsymbol{\eta}_1\| \leq r_1, \\ & \|\boldsymbol{\eta}_2\| \leq r_2. \end{aligned} \tag{3.53}$$

For the minimisation in (3.53), the strain resultants in $\mathbf{\Phi}$ are effectively constant, because when considering the overall energy minimisation problem their value is set by the outer minimisation term,

$$\begin{aligned} \min_{\mathbf{\Phi}} \min_{r_{1,2}, \boldsymbol{\eta}_{1,2}} \ & \int r_1 + r_2 \ \mathrm{d}A \\ \text{s.t.} \ & \mathbf{\Phi} = -\mathbf{P}_1^T \boldsymbol{\eta}_1 - \mathbf{P}_2^T \boldsymbol{\eta}_2 \\ & \|\boldsymbol{\eta}_1\| \leq r_1, \\ & \|\boldsymbol{\eta}_2\| \leq r_2, \\ & \text{boundary constraints.} \end{aligned} \tag{3.54}$$

Note that the integration over area has been placed inside both minimisation terms and the inner minimisation is equivalent to the one in (3.53). Since $\mathbf{\Phi}$ is constant within this term, performing the minimisation over $\mathbf{s}$ is equivalent to calculating the dissipation for a given value of $\mathbf{\Phi}$. However, if $\mathbf{\Phi}$ is allowed to vary whilst the inner minimisation is performed, the value of the dissipation d_I computed would be the minimum over all possible values of $\mathbf{\Phi}$. Essentially, this means that there is an independence between the variables $\mathbf{\Phi}$ and $\mathbf{s}$ so that (3.54) can be written as

$$\begin{aligned} \min_{r_{1,2},\boldsymbol{\eta}_{1,2},\mathbf{\Phi}} \quad & \int r_1 + r_2 \, \mathrm{d}A \\ \text{s.t.} \quad & \mathbf{\Phi} = -\mathbf{P}_1^T \boldsymbol{\eta}_1 - \mathbf{P}_2^T \boldsymbol{\eta}_2 \\ & \|\boldsymbol{\eta}_1\| \le r_1, \\ & \|\boldsymbol{\eta}_2\| \le r_2, \\ & \text{boundary constraints.} \end{aligned} \tag{3.55}$$

After spatial discretisation, the integration over area in (3.55) becomes a summation over the nodes i of the finite elements e, so that (3.55) can be expressed as an SOCP,

$$\begin{aligned} \min_{r_{1,2},\boldsymbol{\eta}_{1,2},\mathbf{\Phi}} \quad & \frac{1}{3}\sum_{e=1}^{NE}\sum_{i=1}^{3} \Delta(r_{1,i}^{(e)} + r_{2,i}^{(e)}) \\ \text{s.t.} \quad & \mathbf{Kx} = -\mathbf{P}_1^T \boldsymbol{\eta}_1 - \mathbf{P}_2^T \boldsymbol{\eta}_2, \\ & \mathbf{Ex} \ge \mathbf{t}, \\ & \|\boldsymbol{\eta}_{1,i}^{(e)}\| \le r_{1,i}^{(e)}, \\ & \|\boldsymbol{\eta}_{2,i}^{(e)}\| \le r_{2,i}^{(e)}, \end{aligned} \tag{3.56}$$

where $\boldsymbol{\eta}_{k,i}^{(e)} = [\eta_{1+6(i-1)}, \ldots, \eta_{6i}]$ for $k = 1, 2$, $\mathbf{x} = [\mathbf{W}^T \ \ \mathbf{U}^T]^T$ and $\mathbf{K}$ is

$$\mathbf{K}_i = \begin{bmatrix} \Delta\mathbf{B}_{\mathbf{P},i}^{(e)} & \mathbf{O} \\ \mathbf{O} & \Delta\mathbf{B}_{\mathbf{M},i}^{(e)} \end{bmatrix} \mathbf{T}^{(e)}, \tag{3.57}$$

hence $\mathbf{\Phi} = \mathbf{Kx}$. As with the plate problem, $\mathbf{x}_{tool} = \mathbf{Ex} \subset \mathbf{x}$ and $\mathbf{t}$ is generated by the tool surface function. The boundary variables are set to zero directly and removed from the optimisation. As with the plate problem presented here, the solution in terms of the dual of (3.56) was found to be more efficient than the primal problem.

3.6 The ‘one cone’ model: relaxation to linearised Ilyushin criterion

An alternative method for deriving a minimisation problem is to simplify (3.50) using a relaxation technique to reduce the two constraints to a single one. This single constraint was chosen so that it optimally approximates the solution space given by the original two constraints. This was done by seeking an ellipsoid function of the form $\mathbf{s}^T\hat{\mathbf{Q}}\mathbf{s} \leq 1$ that best fits the intersection of above two ellipsoids. The value of the approximation $\hat{\mathbf{Q}}$ is found by computing the ellipsoid of smallest volume that contains the intersection of the above constraints. If $\hat{e} = \{\mathbf{s}|\mathbf{s}^T\hat{\mathbf{Q}}\mathbf{s} \leq 1\}$ and $e_i = \{\mathbf{s}|\mathbf{s}^T\mathbf{Q}_i\mathbf{s} \leq 1\}$ for $i = 1, 2$, then $\hat{e}$ is the smallest ellipsoid that satisfies,

$$\hat{e} \supseteq \bigcap_{i=1}^{2} e_i. \tag{3.58}$$

In this case, $\hat{e}$ is the minimum outer ellipsoid. An alternative was also implemented, where $\hat{e}$ was approximated as the maximum inner ellipsoid and it will also be discussed below. Both problems were cast as maximum determinant linear matrix inequality (LMI) optimisation problems using the procedure described by Boyd *et al* in [23], and solved for $\hat{\mathbf{Q}}$ using the semidefinite program solver, SeDuMi, details of which can be found in [80]. To describe how this was done, a standard semidefinite program is defined [114] as

$$\begin{aligned} &\min\ \mathbf{c}^T\mathbf{x} \\ &\text{s.t. } \mathbf{F}(\mathbf{x}) \succeq 0, \end{aligned} \tag{3.59}$$

where the objective function is a linear function of the optimisation variables $\mathbf{x}$ and $\mathbf{F}(\mathbf{x})$ is a matrix that must be positive semi-definite under the above constraint, which is known as an LMI. Hence, $\mathbf{F}(\mathbf{x})$ must satisfy $\mathbf{z}^T\mathbf{F}(\mathbf{x})\mathbf{z} \geq 0$ for all vectors $\mathbf{z}$ of appropriate dimensions, or equivalently eig($\mathbf{F}$) ≥ 0. An alternative representation of the ellipsoids defined above can be made as

$$\hat{e} = \left\{ \mathbf{s} | [\mathbf{s}^T \;\; 1] \begin{bmatrix} \hat{\mathbf{Q}} & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ 1 \end{bmatrix} \leq 0 \right\} \tag{3.60}$$

and for $\hat{e}$ to be an outer ellipsoid of the intersection of e_1 and e_2, it is required that

$$\left\{ \boldsymbol{\vartheta}^T \begin{bmatrix} \mathbf{Q}_1 & 0 \\ 0 & -1 \end{bmatrix} \boldsymbol{\vartheta} \leq 0 \;\; \& \;\; \boldsymbol{\vartheta}^T \begin{bmatrix} \mathbf{Q}_2 & 0 \\ 0 & -1 \end{bmatrix} \boldsymbol{\vartheta} \leq 0 \right\} \Rightarrow \boldsymbol{\vartheta}^T \begin{bmatrix} \hat{\mathbf{Q}} & 0 \\ 0 & -1 \end{bmatrix} \boldsymbol{\vartheta} \leq 0, \tag{3.61}$$

where $\boldsymbol{\vartheta}^T = [\mathbf{s}^T \;\; 1]$. This means that if $\mathbf{s}$ satisfies the condition to lie within in the intersection of sets e_1 and e_2, then this implies that the condition for belonging in set $\hat{e}$ is also satisfied. The problem of finding the minimum $\hat{e}$ for which this is true is NP complete [23], but a reasonable suboptimal solution, which in the case of the Illyshin criterion appears to be optimal, can be found by using the S-procedure. As explained in [24], this states that (3.61) is satisfied if there exist positive scalars τ_1 and τ_2 such that

$$\begin{bmatrix} \hat{\mathbf{Q}} & 0 \\ 0 & -1 \end{bmatrix} - \tau_1 \begin{bmatrix} \mathbf{Q}_1 & 0 \\ 0 & -1 \end{bmatrix} - \tau_2 \begin{bmatrix} \mathbf{Q}_2 & 0 \\ 0 & -1 \end{bmatrix} \preceq 0. \tag{3.62}$$

The S-procedure only guarantees optimal solutions for constraints of the form in (3.62) that have only one scalar variable τ, thus the solution to this problem may be conservative [23]. The optimisation problem is formed by minimising the volume of $\hat{e}$ subject to the above condition using the property that the volume of the ellipsoid is proportional to $\sqrt{\det\hat{\mathbf{Q}}^{-1}}$, which gives

$$\begin{aligned} & \min \; \det\hat{\mathbf{Q}}^{-1} \equiv \min \;\; \log\det\hat{\mathbf{Q}}^{-1} \\ & \text{s.t. } \; \hat{\mathbf{Q}} \succ 0, \\ & \qquad \hat{\mathbf{Q}}^T = \hat{\mathbf{Q}}, \\ & \qquad \tau_1 \geq 0, \;\; \tau_2 \geq 0, \\ & \qquad \begin{bmatrix} \hat{\mathbf{Q}} & 0 \\ 0 & -1 \end{bmatrix} - \tau_1 \begin{bmatrix} \mathbf{Q}_1 & 0 \\ 0 & -1 \end{bmatrix} - \tau_2 \begin{bmatrix} \mathbf{Q}_2 & 0 \\ 0 & -1 \end{bmatrix} \preceq 0. \end{aligned} \tag{3.63}$$

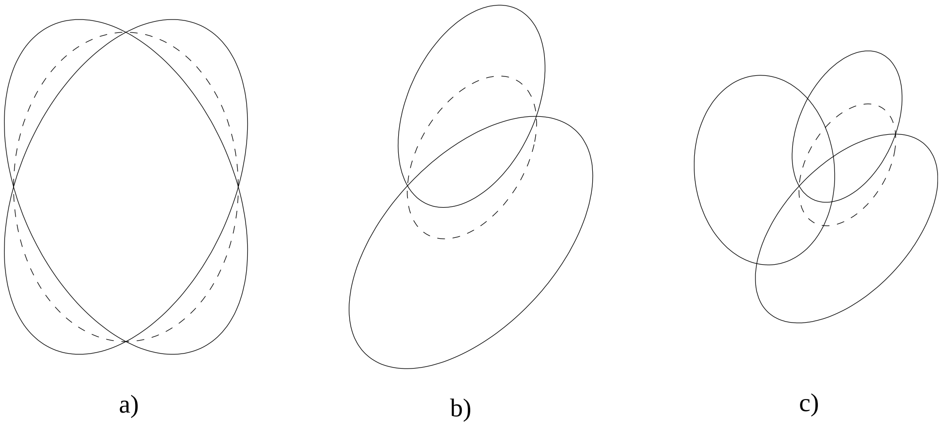


Figure 3.10: Approximate solution for the NP complete problem of finding the smallest ellipsoid $\hat{e}$ (dashed line) enclosing the intersection of other ellipsoids e_1 and e_2 (solid lines) for two dimensions. This was computed by solving an LMI problem using semidefinite programming. For example, in a) the optimal solution was found whereas b) and c) display suboptimal solutions.

This was solved as a semidefinite programme in Matlab using SeDuMi [80] and the resulting solution for the minimum outer ellipsoid is

$$\hat{\mathbf{Q}} = 0.5(\mathbf{Q}_1 + \mathbf{Q}_2). \tag{3.64}$$

This is closely related to approximations to the Ilyushin condition applied by other researchers, as discussed in [120]. $\hat{\mathbf{Q}}$ is six dimensional and therefore it cannot be visualised on a plot, so to demonstrate the way the formulation works, solutions for some two dimensional ellipsoids are shown in figure 3.10. Of the three examples, the underlying ellipsoids in figure 3.10 a) have a structure that is most similar $\mathbf{Q}_1$ and $\mathbf{Q}_2$. They are defined by $e_i = \{\mathbf{s}|\mathbf{s}^T\mathbf{Q}_i^{(2D)}\mathbf{s} \leq 1\}$ for $i = 1, 2$, where

$$\mathbf{Q}_1^{(2D)} = \begin{bmatrix} 16 & 11.1547 \\ 11.1547 & 11 \end{bmatrix}, \tag{3.65}$$

$$\mathbf{Q}_2^{(2D)} = \begin{bmatrix} 16 & -11.1547 \\ -11.1547 & 11 \end{bmatrix}. \tag{3.66}$$

Interestingly, the solution $\hat{e}$ for these ellipsoids is optimal, whereas in figure 3.10 b) and c) the minimum outer ellipsoid is suboptimal. Although this does not guarantee that the

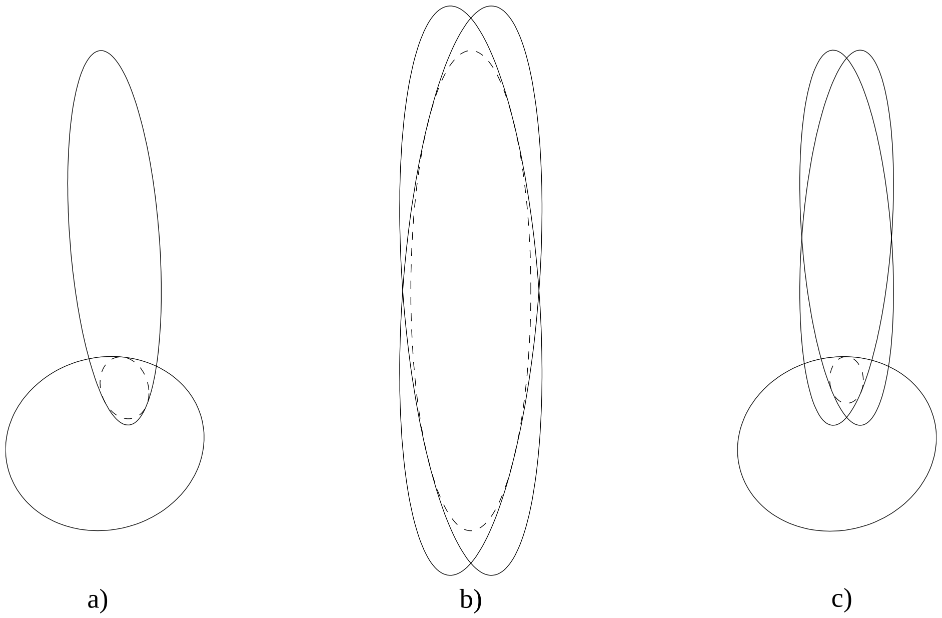


Figure 3.11: Largest inner ellipsoid (dashed line) within the intersection of other ellipsoids (solid line) in two dimensions. This was computed by solving an LMI problem using semidefinite programming.

solution in (3.64) is optimal, it does suggest that it is likely to be optimal due to the symmetry of $\mathbf{Q}_1$ and $\mathbf{Q}_2$.

An alternative approximation to the Ilyushin criterion was also implemented for which an optimal solution is guaranteed. In this case $\hat{e}$ was computed as the maximum inner ellipsoid, where the volume of the ellipsoid was maximised by minimising det$\hat{\mathbf{Q}}$ and from the S-procedure, the equivalent constraints to (3.62) were

$$\begin{bmatrix} \mathbf{Q}_i & 0 \\ 0 & -1 \end{bmatrix} - \tau_i \begin{bmatrix} \hat{\mathbf{Q}} & 0 \\ 0 & -1 \end{bmatrix} \preceq 0, \quad i = 1, 2, \tag{3.67}$$

with $\tau_i \geq 0$. Since each constraint only contains one variable τ_i the solution is optimal, as visualised by example solutions for both 2 dimensional and 3 dimensional ellipsoids in figures 3.11 and 3.12. For the Ilyushin criterion the optimal inner ellipsoid was found to be

$$\hat{\mathbf{Q}} = 1.552(\mathbf{Q}_1 + \mathbf{Q}_2), \tag{3.68}$$

hence no difference in the deformation mechanism would be observed whether (3.64) or (3.68) is used to approximate the yield surface since they are just a scaled forms of each

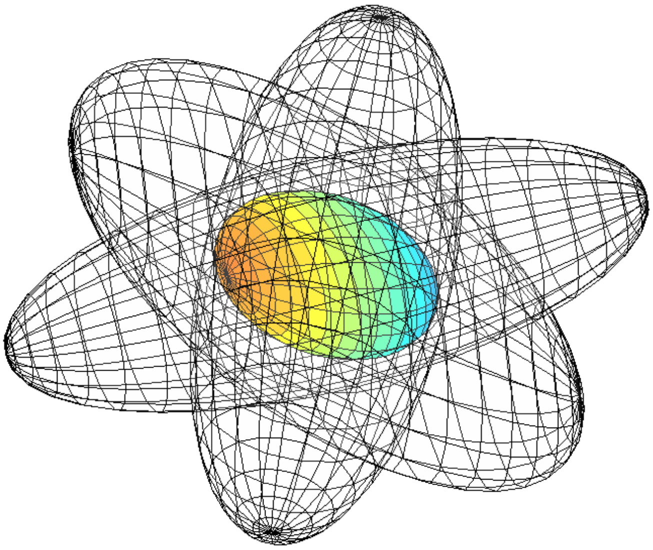


Figure 3.12: Largest inner ellipsoid (coloured) within the intersection of three other ellipsoids in three dimensions. This was computed by solving a linear matrix inequality problem using semidefinite programming.

other.

The relaxed form of the principle of maximum plastic work is

$$\begin{aligned} d_I = \max_{\mathbf{s}} \ & \mathbf{s}^T \mathbf{\Phi} \\ \text{s.t.} \ & \mathbf{s}^T \hat{\mathbf{Q}} \mathbf{s} \leq 1. \end{aligned} \tag{3.69}$$

This can now be treated the same way as the plate problem and the internal dissipation can be cast as the integration over area of a norm by following the steps in equations (3.13-3.16), which can then be formulated as an SOCP after spatial discretisation. The final optimisation problem is

$$\begin{aligned} D_I = \min \ & \sum_{i}^{3NE} \frac{1}{3} z_i \\ \text{s.t.} \ & \mathbf{n} = \mathbf{PKx}, \\ & \|\mathbf{n}_i\| \leq z_i \quad i = 1, .., 3NE, \\ & \mathbf{x}_{tool} \geq \mathbf{t}, \\ & \mathbf{x}_{boundary} = 0. \end{aligned} \tag{3.70}$$

where $\mathbf{n}_i = [n_{1+6(i-1)}, \ldots, n_{6i}]$, $\mathbf{P}^T\mathbf{P} = \hat{\mathbf{Q}}$ and $\mathbf{K}$ and $\mathbf{x}$ were defined in §3.5. As with the plate problem, the edge boundary conditions can be implemented implicitly and the tool constraints can be written as $\mathbf{Ex} \geq \mathbf{t}$. In practice, it is more efficient to solve the dual form of the problem.

3.7 Time integration, tool contact and scaling

The time evolution for both derivations (§3.5 and §3.6) of the model follows a sequential limit analysis scheme [118] that depends only on the change in position of the tool at each time step. This can be considered as explicit time integration using a forward difference scheme to update the deformation field at each time step [98].

When applying the model to SPIF, an assumption about the contact condition between the nodes of the finite element mesh and the tool is made for the $t+1$ time step. Specifically, if r is the radius of the spherical tool-tip, only nodes that fall within the region of the tool at time step t, defined as the projection of a circular area onto the sheet of radius r centred at the (x, y) position of the tool tip belong to the subset $\mathbf{x}_{tool}$, which is subject to the tool constraints until time step $t + 1$.

For now, these nodes will be constrained to only move vertically, but in §3.10.1 a method for relaxing this constraint whilst maintaining the SOCP structure is described. This permits sliding between the tool and the sheet, although as before, only the subset of nodes in $\mathbf{x}_{tool}$ are constrained by the tool surface, so it is possible during a time step for a node that falls just outside of the constrained region to slide into the tool region but not be constrained by the tool. In practice, the consequence of this is negligible because little sideways motion of the nodes was observed in comparison to vertical motion, both in these models and in the benchmarks in §1.5.1. Frictionless contact states are considered here when sliding is permitted, but it is possible to include friction effects by applying a traction force term in the objective function. One paper that discusses friction within the context of sequential limit analysis is [52].

The initial simulations of the SPIF process provided in §3.8 use the perfectly plastic assumption, although it is straightforward to modify this to include work-hardening. To incorporate non-linear work-hardening in the model, the cost coefficients f associated with an element should be made variable with respect to time and should be increased at time step $t+1$ appropriately in relation to the degree of equivalent strain the element underwent at time step t. This is discussed in further detail in §3.11.

Given that the stress resultant yield constants M_0 and N_0 have been defined as

$$M_0 = \sigma_o \frac{h^2}{4}, \quad N_0 = \sigma_o h, \tag{3.71}$$

where σ_o is the material yield stress and h is the sheet thickness, for a general choice of units for length measurement, M_o differs to N_o by several orders of magnitude due to the extra thickness multiplier, leading to ill-conditioning of the problem in equation (3.70). Indeed, the problem is only stable if mm are used, since $h \approx 1$ mm. In order remove this dependency on length scale, the stress resultant moments were redefined in a generalised form [120] as

$$\tilde{m}_i = -\frac{1}{M_0 h^2} \int_z z\sigma_i \mathrm{d}z \qquad i = x,\ y,\ xy \tag{3.72}$$

and similarly

$$\tilde{n}_i = \frac{1}{N_0 h} \int_z \sigma_i \mathrm{d}z \qquad i = x,\ y,\ xy, \tag{3.73}$$

so that $M_0 = \frac{\sigma_o}{4}$, $N_0 = \sigma_o$. A corresponding generalised strain vector was also defined that is work conjugate to the redefined stress resultants,

$$\tilde{\mathbf{\Phi}}_e = \begin{bmatrix} \tilde{\boldsymbol{\kappa}}_i \\ \tilde{\boldsymbol{\epsilon}}_i \end{bmatrix} = \begin{bmatrix} h\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\kappa}_i \\ \boldsymbol{\epsilon}_i \end{bmatrix} = \mathbf{Z}\mathbf{\Phi}_e, \tag{3.74}$$

and the optimisation problem was modified appropriately. The resulting optimisation problem was found to be stable for a large range of length units.

3.8 Results

In this section the two cone and one cone models were applied to two cases: The first simulates a series of point loads applied along a straight line to an initially flat, clamped sheet; the second test simulates a full ISF product and compares the predicted deformation to data from LS-Dyna and measurements from the actual part. As with the plate problem,

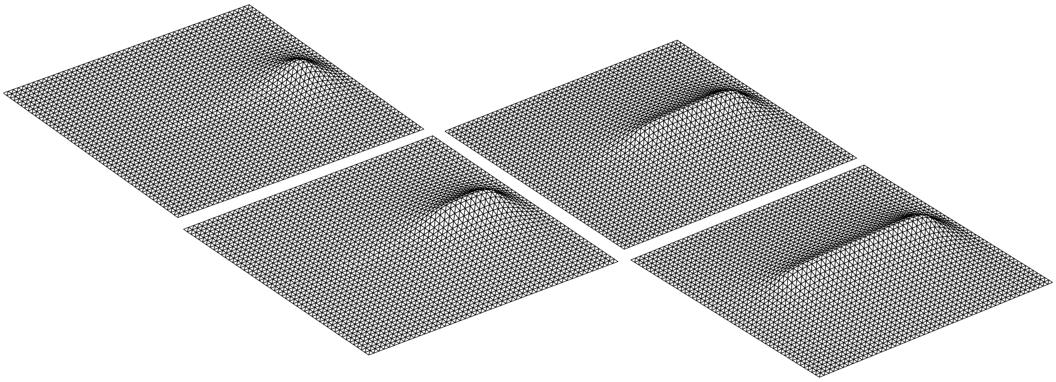


Figure 3.13: Deformation of one cone model for tool sequentially applying point loads along a straight line.

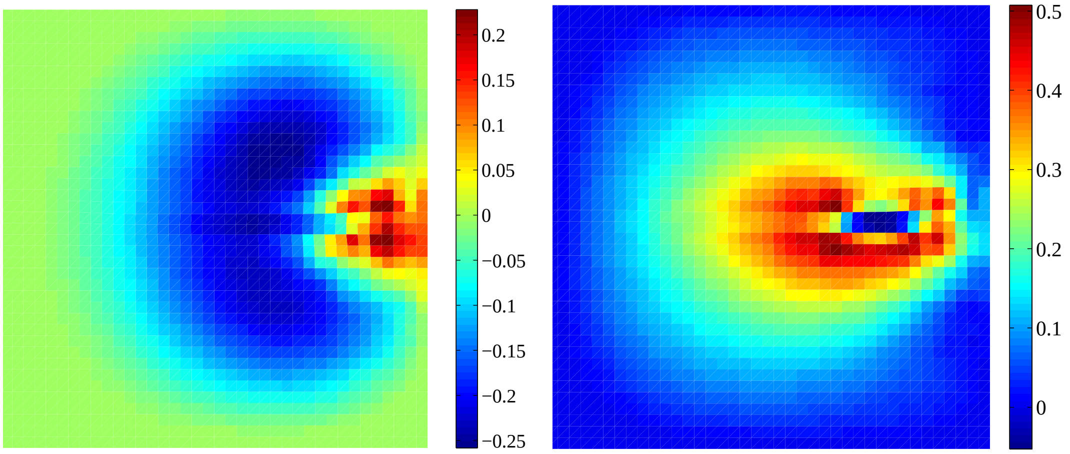
the models were solved using the SOCP software package Mosek [85] on a standard desktop computer running MS Windows.

3.8.1 Plate subjected to a sequence of loads

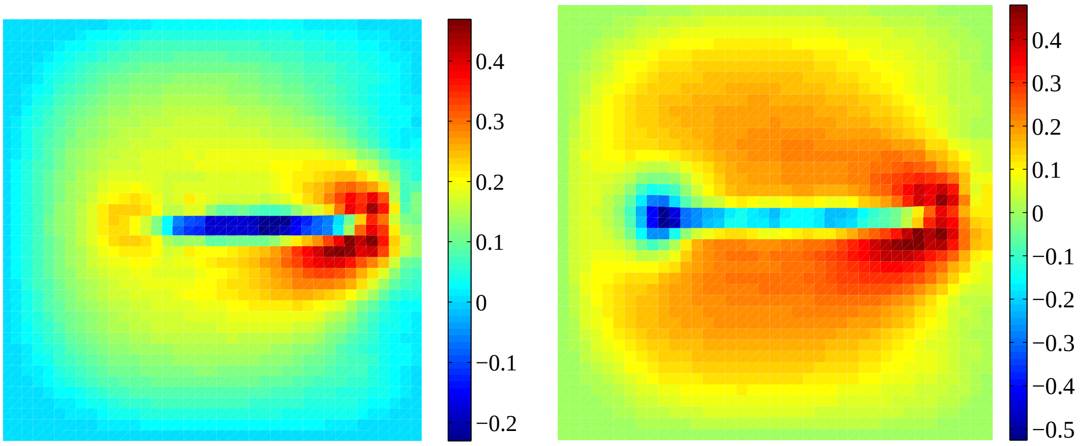
The one cone and two cone models were used to simulate the deformation of an initially flat sheet subjected to a load that was applied sequentially at a series of equally spaced points along a straight line. Since this line test is fairly simple, it is commonly used in ISF research. For instance, in [68] it is used to evaluate a re-meshing strategy for an elasto-plastic FEA model for ISF and in [86] it is used to experimentally compare ISF with several other metal forming processes.

The deformation field for both the one cone and two cone simulations are very similar, so only the results from the one cone model are displayed in figure 3.13 at four equally spaced periods of the process. The indentation depth of the tool was 10 mm and the one cone model used $NE = 4802$ whereas the two cone model used $NE = 3042$, since for this value of NE, the two cone model was close to the memory capacity of the computer used, because of the need to define two second-order cone constraints per node rather than one. In terms of solution time, the two cone model took an average of 38.84 sec and the one cone model took an average of 33.4 sec to solve the limit analysis problem at each increment of the tool position, including both the set-up time in Matlab and the solver time. In total this amounts to 53 mins and 45 mins 37 sec respectively, hence the two cone model was slower than the one cone model despite having a coarser mesh.

Figure 3.14 shows the actual difference between the two models at the four stages in



(a) RMS: 0.0639mm; ν: 0.0072mm^2 (b) RMS: 0.0115mm; ν: 0.0137mm^2



(c) RMS: 0.0993mm; ν: 0.0085mm^2 (d) RMS: 0.0972mm; ν: 0.012mm^2

Figure 3.14: Comparison of vertical differences in mm between the one cone and two cone models. RMS = root mean square difference and ν = variance.

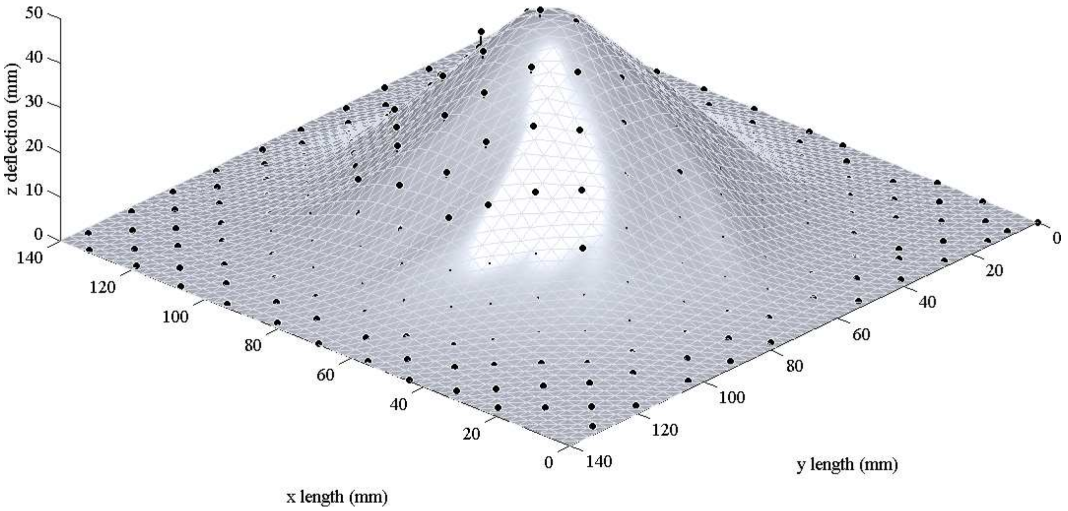


Figure 3.15: Simulation of test product in figure 1.14 produced by solving the rigid plastic shell problem using SOCP for a regular finite element mesh. The superimposed black dots are measured points from the real product.

figure 3.13. These differences are mostly within 1% of the the overall indentation depth, suggesting that the consequence of the approximation to the Ilyushin criterion made in the one cone model is minor. Due to the similarity in the deformation fields for the two cone and the one cone models, the results in the following sections are presented using only the one cone model.

3.8.2 Simulation of ISF part

In this section the one cone model was verified against a part produced on the Cambridge SPIF machine [5]. The boundary conditions matched those above and the solution was compared with the CMM measurements from the real product. These were superimposed over the model and an interpolation field within each element was used to find the predicted deflection directly below, or above, each of the measured points.

The result for the one cone model using $NE = 5000$ is shown in figure 3.15, where the RMS vertical difference between measured and simulated data is 0.88 mm, with a variance of 0.69 mm^2 and maximum difference of 3.17 mm. The simulation took approximately 18 hours, or 39 seconds per time step, with 60% of the time spent in the SOCP solver. The remaining period was spent in Matlab and could be reduced by using a compiled script. The solution times for other mesh sizes and types are displayed in figure 3.21, where a distinction is made between the solver time and assembly time.

Mesh size	**Error measures**		
No. of elements, NE	**RMS** (mm)	**Var.** (mm^2)	**Max. diff.** (mm)
800	1.95	2.39	5.21
1250	1.48	1.57	4.20
1800	1.16	1.05	3.22
2450	1.03	0.88	2.98
3200	0.95	0.77	3.09
4050	0.91	0.73	3.15
5000	0.88	0.69	3.17

Table 3.1: Error measures displayed against mesh density.

In table 3.1, a comparison of RMS error, variance and maximum difference between the rigid plastic finite element model and measured data is displayed for increasing values of NE, the number of elements. All tests used the same regular triangular mesh and the RMS error and variances both decreased as the mesh density increased. The errors display a common trend, which is easier to see when plotted, as done in figure 3.16, where the error decreases towards a plateau beyond which increasing the mesh density has no effect on the accuracy of the solution. The plot for maximum difference is different because it appears to increase after $NE = 2500$, although very slightly. This is likely to be because the plateau for maximum deflection has been reached at a value of about 3 mm, in which case the solutions are simply oscillating about this value. However, although the maximum error is useful because it provides the bounded domain within which all measured errors lie, it is less important than the RMS error and variance because the latter two provide information about the distribution of errors, rather than a value from one single point on the sheet.

As well as measured data, the model has also been compared with data computed from commercial finite element software. Figure 3.17 shows the differences over the sheet between the current model for $NE = 5000$ and the LS-Dyna models in figures 1.12 and 1.13. The simulation time per time-step for the standard cone was the same as for the twisted cone shape. For both cases the error measures between the two models displayed in table 3.2 appear satisfactory.

The rigid plastic models and LS-Dyna appear to disagree most at the base of the product, as seen the by the red regions in figure 3.17. However, when comparing against the real product for the twisted cone shape, both models show greater discrepancies in

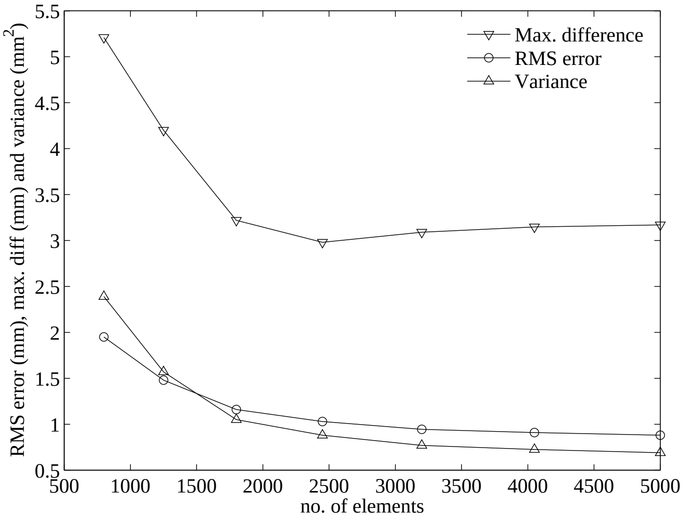


Figure 3.16: A comparison of selected error measures between the rigid plastic finite element model and measured data for different numbers of elements (from the data in table 3.1).

	Error measures		
Shape	**RMS** (mm)	**Var.** (mm^2)	**Max. diff.** (mm)
strandard cone	0.39	0.13	1.38
twisted cone	0.64	0.36	1.77

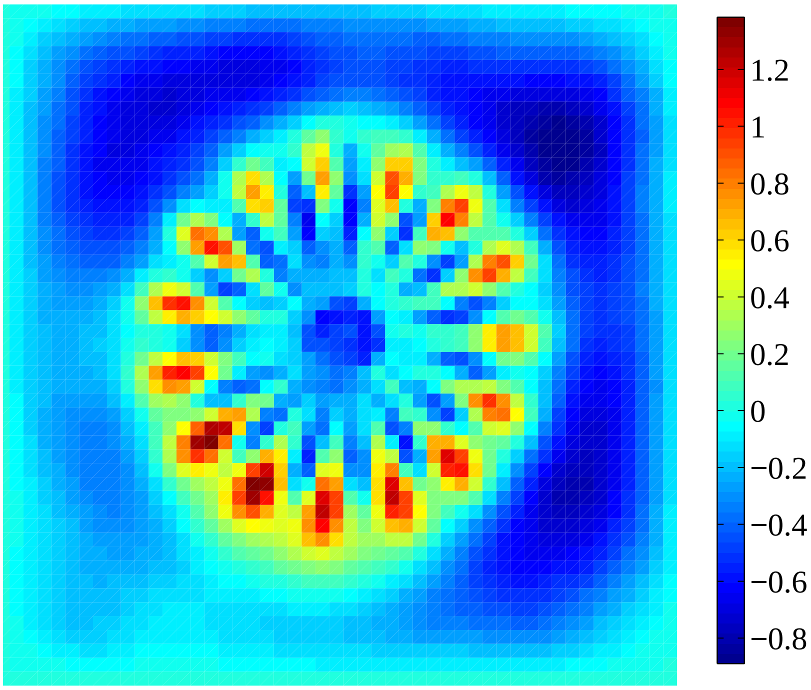
Table 3.2: Error measures of difference between one cone model and LS-Dyna from the plots in figure 3.17.

the midrange between the base and the maximum depth. Since the twisted cone product was removed from its clamping frame so that it could be measured on the CMM, it was expected that elastic recovery, which is not accounted for in the rigid plastic model and is also not computed in the explicit solver in LS-Dyna, would result in larger differences at base regions of the product yet this appears not to be the case.

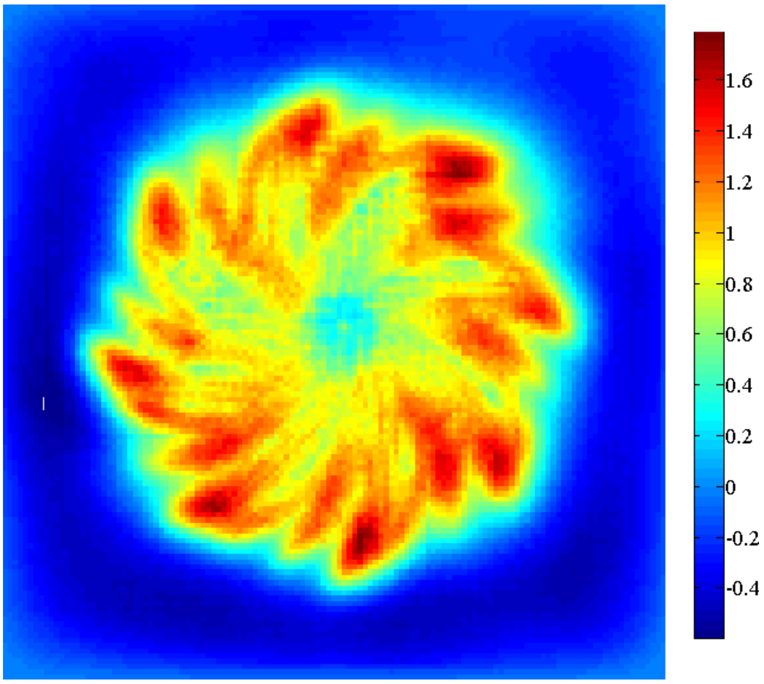
When it comes to addressing the control problem of reducing errors between the model and the target product, which are different to the errors used to verify the model above, the greatest differences between the two lie at the base region of the product. Since LS-Dyna and the current model appear to predict the deformation in this region equally well, either of the two models ought to perform similarly if used for this purpose.

3.8.3 Irregular meshes and mesh refinement

The results provided so far are the first evaluation of the rigid plastic assumption for modelling ISF and the use of recent SOCP methods in solving a shell problem. However, it



(a) Standard cone product



(b) Twisted cone product

Figure 3.17: Difference in mm between deformation field predicted by the one-cone model and LS-Dyna models (standard cone in figure 1.12 and twisted cone in figure 1.13).

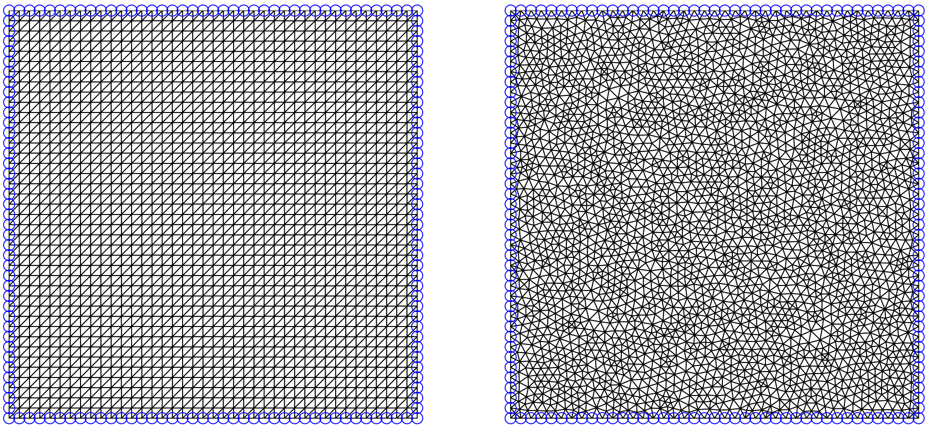
Mesh size		**Error measures**		
Hmax (mm)	**No. of elements,** NE	**RMS** (mm)	**Var.** (mm^2)	**Max. diff.** (mm)
7	1208	1.24	1.23	3.60
5.6	1954	0.94	0.82	2.94
4.67	2832	0.83	0.65	3.26
4	3854	0.83	0.65	3.28
3.5	5226	0.79	0.61	3.32
3.11	6586	0.79	0.60	3.40
2.8	8412	0.77	0.57	3.4

Table 3.3: Mesh sizes and corresponding simulation error measures for irregular meshes produced using *initmesh*.

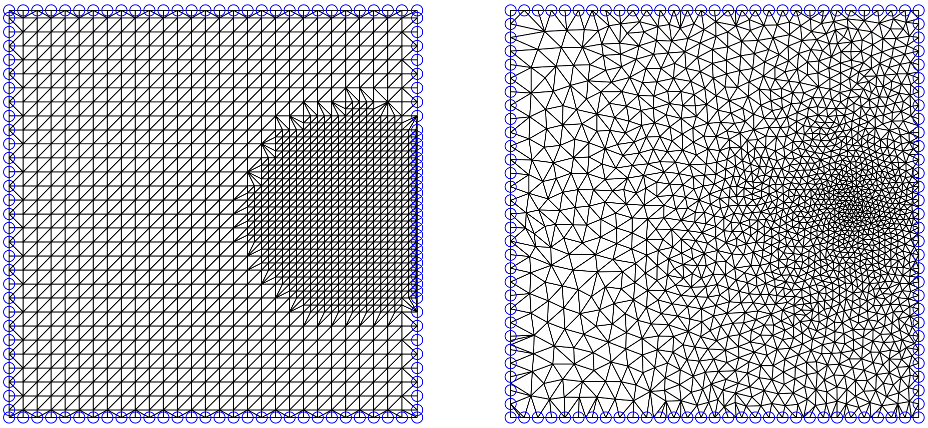
is believed that further work could be done to maximise the benefits that can be obtained from applying the rigid plastic assumption in terms of computation time. One possibility is to use adaptive re-meshing, whereby the overall number of simulation variables is reduced by using small elements only in regions of the sheet that are of interest. The four meshes in figure 3.18 were tested using the rigid plastic finite element model for ISF and two of these use mesh refinement. Figure 3.18(a) shows the regular mesh used in the previous section. This mesh can be created by writing a simple script or by using the *poimesh* function in the Matlab PDE toolbox. The *initmesh* function from the same toolbox is used to create the mesh in figure 3.18(b) and was also used for the one in figure 3.5. A result using the irregular mesh is shown in figure 3.19, for which the RMS difference between the model and the simulation was 0.79 mm, the variance was 0.60 mm^2 and the maximum difference was 3.40 mm.

As with the regular mesh, a test was performed to see how the simulation errors varied with mesh size. The number of elements NE cannot be directly controlled in *initmesh*, so this was manipulated indirectly by attempting to limit the maximum size of the triangles generated, or specifically the maximum edge length of any element by using the built-in property *Hmax*. As shown in table 3.3, the resulting NE do not vary linearly with this parameter because there is no lower size limit on the triangular element, but the range of NE covered is satisfactory. The corresponding error measures are also shown in the table 3.3.

So that the regular and irregular meshes can be compared to each other, a parameter must chosen against which the prediction errors for the two can be plotted. The natural candidates for this are either the maximum edge length, *Hmax*, or the number of elements,



(a) Regular mesh (b) Irregular mesh



(c) Regular mesh with refinement around tool (d) Irregular mesh with refinement around tool

Figure 3.18: The four different meshes tested with the rigid plastic finite element model.

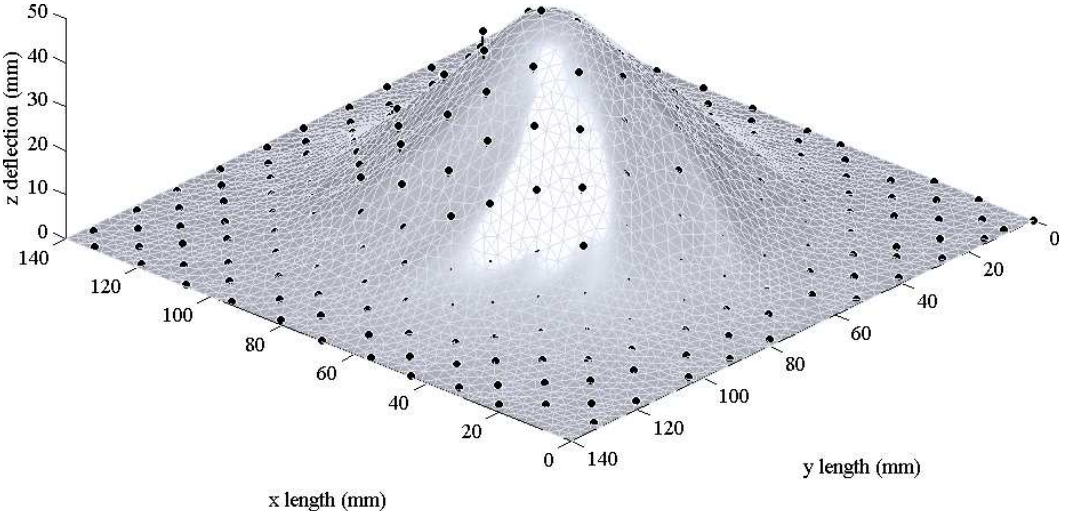


Figure 3.19: Simulation of test product in figure 1.14 produced by solving the rigid plastic shell problem using SOCP for an irregular mesh of 6586 elements. The superimposed black dots are measured points from the real product.

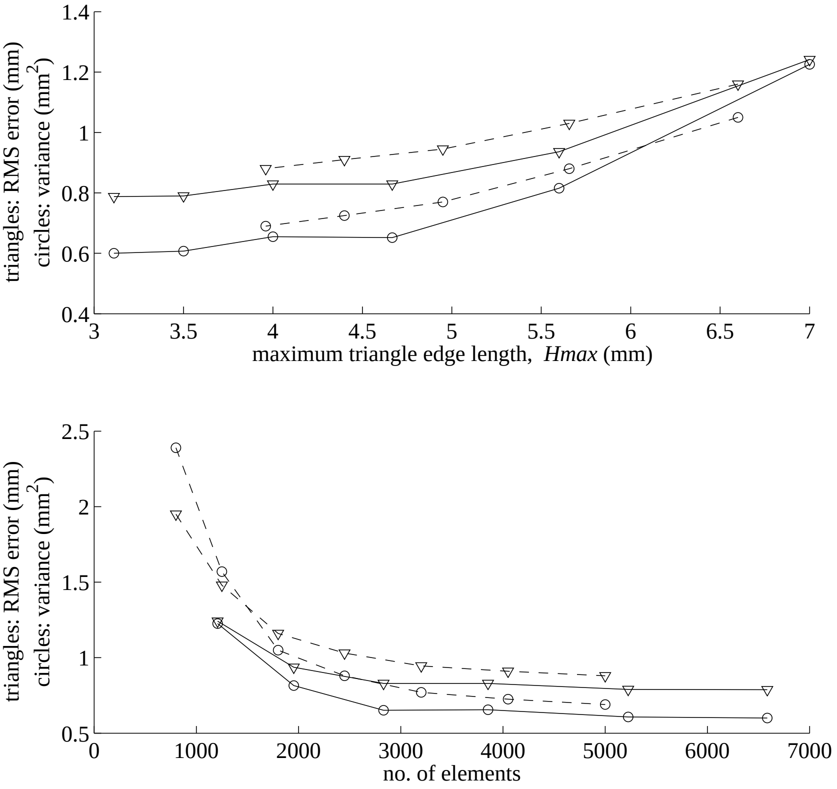


Figure 3.20: A comparison of RMS error and variance between the rigid plastic finite element model and measured data for regular meshes (dashed) and irregular meshes (solid).

NE. In figure 3.20 the RMS error and variances from tables 3.1 and 3.3 are compared against both of these measures over a suitable range of values. It is interesting that for a given value of $Hmax$ or NE, the irregular mesh outperforms the regular mesh. This is especially clear in the plot of errors against NE.

It has been observed that most of the material deformation appears to occur within a reasonably local region surrounding the tool. Therefore, for the refined regular mesh in figure 3.18 c), the idea is to use a coarse mesh that was refined within a region localised to the tool. As the tool moves the region of refinement is moved with it. The underlying mesh was formed from a regular mesh with one degree of mesh refinement computed by using the quadtree algorithm, which is implemented in the open source mesh generator QMG [79]. It is possible to produce further degrees of mesh refinement but as discussed below this provides little further benefit. For figure 3.18 c) the model used a base mesh that was generated over a 30×30 grid from which the localised region was refined to the equivalent of a 60×60 mesh. In figure 3.18 d) the open source Matlab script DistMesh [93]

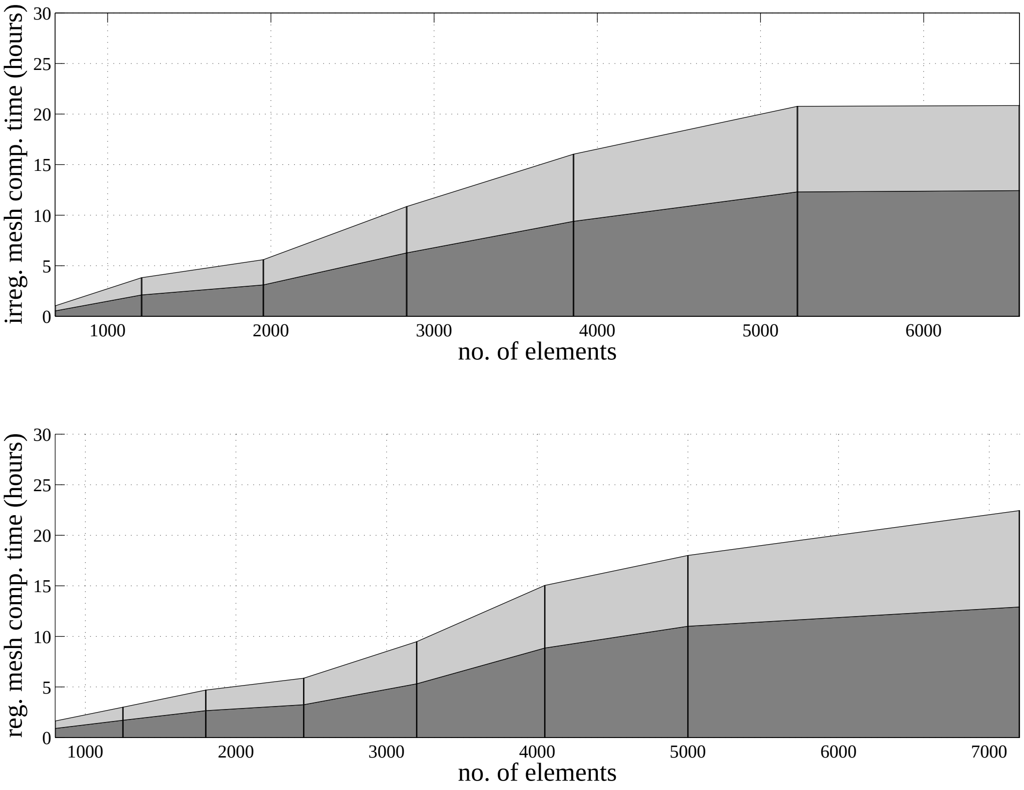


Figure 3.21: Comparison of computing times for simulations of the product in figure 1.14 against mesh size, for irregular (top) and regular meshes (bottom). Total time is broken into solver time (dark grey) and set-up time (light grey). The average time for each SOCP solved is obtained by dividing the above times by 1664, the number discrete tool position increments.

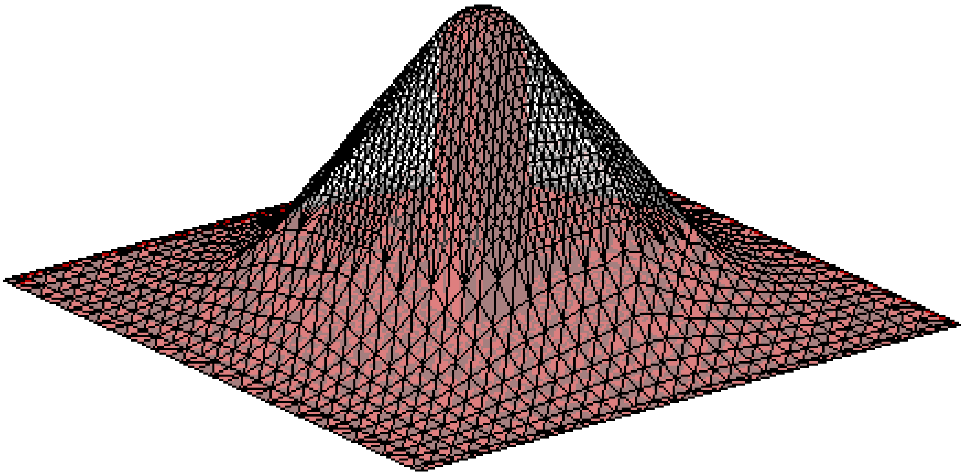


Figure 3.22: Simulation of test product in figure 1.14 produced by solving the rigid plastic shell problem using SOCP and mesh refinement around the tool. The tool is shown under the sheet in its final position.

was used to refine the irregular mesh using the same idea of producing smaller elements in the region surrounding the tool.

By using the mesh in figure 3.18 c) the solution time for the model was reduced by 50% by halving the number of elements to $NE = 2528$ when compared to $NE = 5000$ for a regular 50×50 mesh, without affecting the predicted geometry. Specifically, a RMS vertical difference of 0.81 mm was observed with a variance of 0.65 mm^2 and maximum deflection of 3.23 mm when compared to the measured data. The deformation field is shown in figure 3.22, where the tool is displayed underneath the mesh in its final position. The mesh displays poor element quality at the edge of the refinement region, but it was found to be better at maintaining the shape of the product during remeshing when compared to the irregular refined mesh. For instance, the corners in the product became smoother with each refinement step using an irregular mesh. It was found that finer degrees of mesh refinement could be used for the regular mesh but occassionally this also led to difficulties in maintaining the product shape when updating the mesh at each time step. For ease of comparison, the error measures for the remeshing strategy are shown in table 3.4 along with the corresponding results for the 50×50 regular mesh and LS-Dyna model.

3.8.4 Strain ('stretch') comparison

It was outlined previously that the main purpose of the rigid plastic model was to predict deformation and so accuracy of stress and strain prediction was not of great concern.

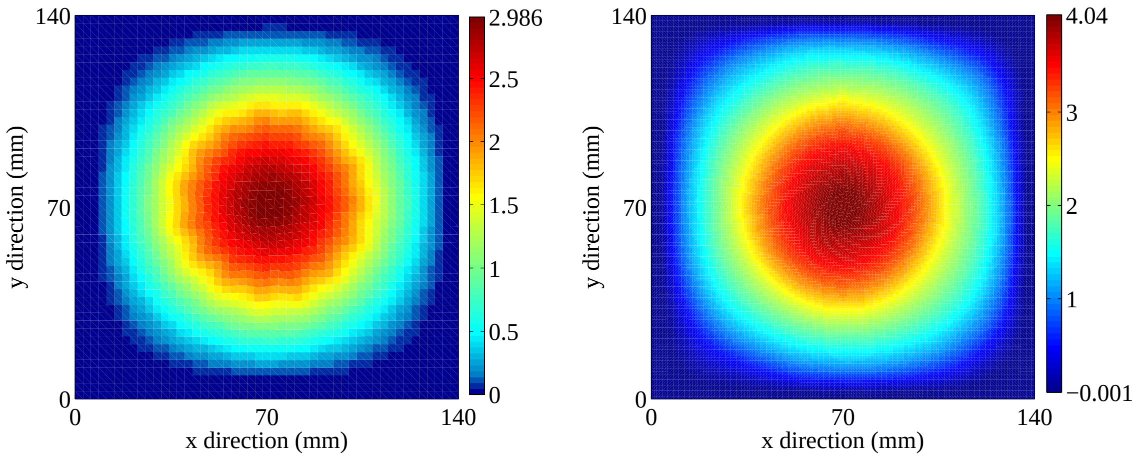
Model	**Cone** shape	**RMS** (mm)	**Var.** (mm^2)	**Max. diff.** (mm)	**Comp. time** (hrs)
		Comparison with measured data			
reg. mesh	twisted	0.88	0.69	3.17	11 (1 PC)
remeshing	twisted	0.81	0.65	3.23	5.5 (1 PC)
LS-Dyna	twisted	0.998	0.99	3.7	8 (16 CPUs)
		Comparison with LS-Dyna			
reg. mesh	twisted	0.64	0.36	1.77	11 (1 PC)
reg. mesh	standard	0.39	0.13	1.38	6.7 (1 PC)

Table 3.4: Summary of error measures of models compared against measured data and LS-Dyna for regular meshes.

However, the strain distribution in the one cone model and LS-Dyna for the benchmark shape of figure 1.12 is briefly studied here. In order to do this the major and minor logarithmic principle stretches [86] were computed for the two models and displayed in figures 3.23 and 3.24 respectively. Logarithmic principle stretches were chosen because they provide an estimate of the strain distribution using only the initial and final states of the sheet and they were also used for the analysis of ISF experiments and Abaqus simulations by Music and Allwood [86]. One limitation of this approach is that the strain path taken is not known, but due to its simplicity the technique is commonly used in many areas of metal forming and indeed, it is the standard calculation for producing forming limit diagrams [66]. In ISF in particular, the technique has been used experimentally by a number of other researchers [65, 56, 109, 64, 42].

The experimental method typically involves imprinting a grid onto the sheet on the opposite side to the one worked by the tool, using an etching or printing method and comparing the final distorted grid to the initial one. With the numerical models here a regular FE mesh was used, so the nodal coordinates readily provided the grid arrangement of the data required for the calculation. The algorithm used to compute the stretches was the same one used in [86] and the distribution of stretches was plotted in figures 3.23 and 3.24.

The major stretches for both models in figure 3.23 follow a similar pattern of increasing in value with forming depth. This trend has been observed experimentally and it is discussed in [58], where it is suggested that the reason why the strains are not constant, as might be expected for a uniform tool path, is that as the process progresses the deformation transforms from being a widely distributed deformation across the sheet to being localised



(a) one cone model (b) LS-Dyna model

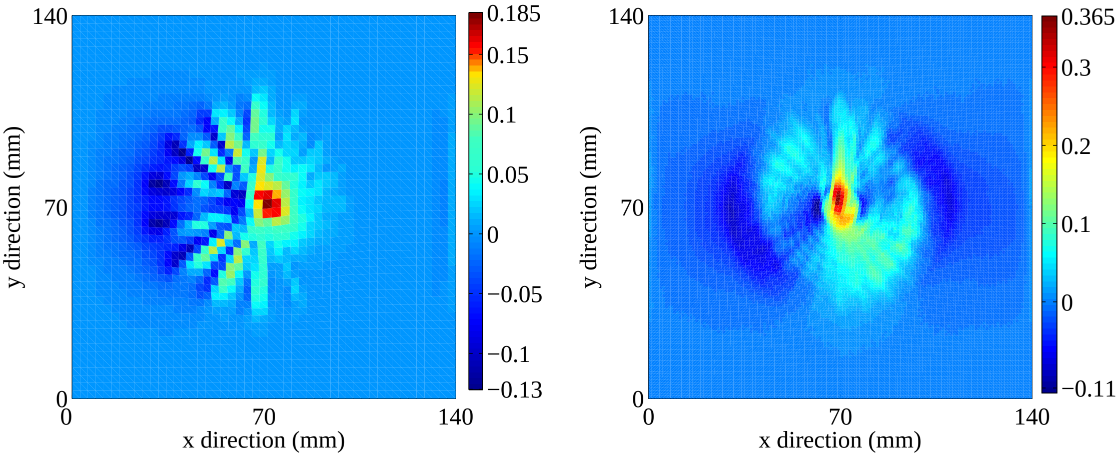
Figure 3.23: Distribution of major stretches over the sheet

to the contact area of the tool. This is because as the sheet assumes a 3D shape it becomes more rigid and hence resists widely spread deformation, therefore restricting the deforming area and increasing the strains in the regions that do deform.

The trend in the strain distributions is the main factor to observe, but it should be noted in figures 3.23 a) and 3.23 b) the maximum major stretch value differs by about one. This could be due a number of factors, such as the finer grid available for calculating the stretch in the LS-Dyna simulation, or the parameters used for the material properties and tool-sheet contact state. The minor stretches in figure 3.24 are also similar in the sense that they are typically much smaller than the major stretches. This is more apparent in figure 3.25, which is a plot of major stretches against the minor stretches for each element in the grid. Hence, the darker regions in the plot represent the more frequently occurring major-minor stretch combinations over the sheet. This plot is referred to as a stretch spectrogram and as with the ones produced in [86], the distributions here are both in the direction of the major axis. This agrees with the statements in [55, 91, 86] that the deformation mechanism during ISF is largely that of plain strain stretching.

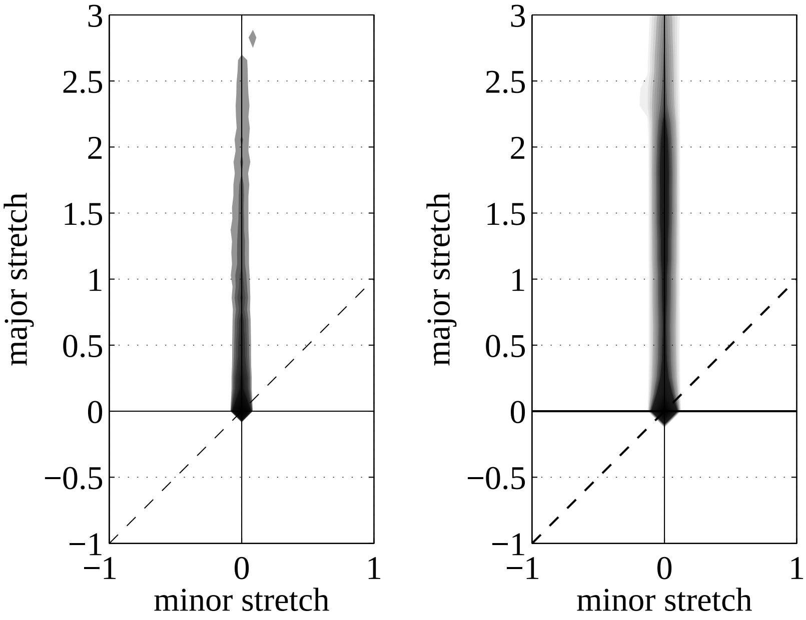
3.8.5 Approaches taken by other researchers

Since ISF presents several challenges in terms of numerical modelling, it is the subject of extensive work in a number of ISF research groups around the world. Most of this work is concerned with reducing simulation times, so to help put this work into context it is



(a) one cone model (b) LS-Dyna model

Figure 3.24: Distribution of minor stretches over the sheet



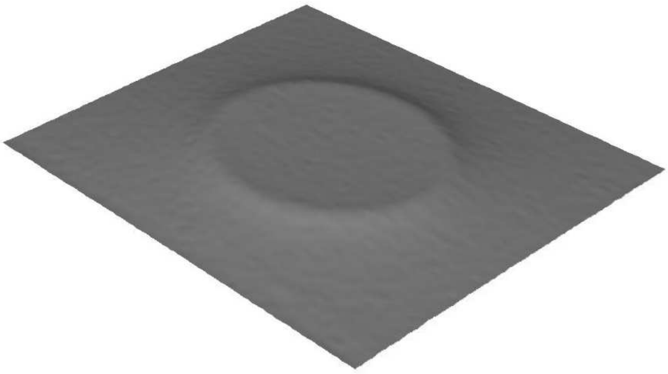
(a) one cone model (b) LS-Dyna model

Figure 3.25: Stretch spectrograms for two models

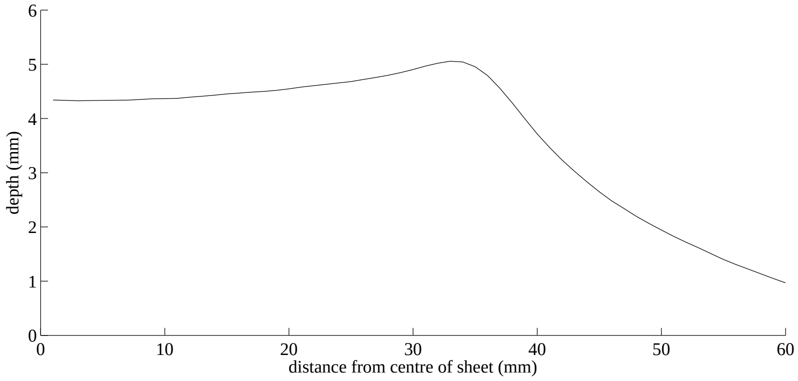
worth considering some of the other approaches that are being taken before continuing with further analysis of the ISF model presented here.

Bambach *et al*'s [19] metamodel based on a database of FE simulations and Hirt *et al*'s [47] geometric model for ISF have already been discussed in §1.6. More recently, Lequesne *et al* [68] have reported that they are developing a method for adaptive mesh refinement and de-refinement in elasto-plastic FE simulations of ISF. The main issue here is how to maintain accuracy in the strain histories during the de-refinement step, but this work is still in need of further development. In [68] only some results for a line test with and without the adaptive meshing scheme are presented, but no indications of the computation times involved have been given yet. Robert *el al* [104] also recently produced a simplified elasto-plastic scheme based on incremental deformation theory. This requires fewer iterations than the standard flow rule method used in elasto-plastic FEA, but in practice an improvement in computation time of only 4.2% was observed over Abaqus Explicit when run on a single CPU.

Hadoush and van den Boogaard [44] have developed a more successful domain decomposition strategy for modelling ISF, which is based on recent work in modelling conventional spinning [96] and extrusion [92] processes. The idea is to separate the sheet into an 'elasto-plastic domain' localised to the tool and an 'elastic domain' that covers the rest of the sheet. A full elasto-plastic implicit FE model is solved in the elasto-plastic domain and it is matched to the elastic domain by appropriate boundary conditions. A quicker elastic simulation is performed in the elastic domain, which provides the extra computational benefit. In [44] a mesh of 1600 finite elements is used to simulate a flattened pyramid shaped product formed by 24 layered passes of the tool. The domain decomposition method took just over six hours to solve for this, as opposed to 12 hours using a standard implicit simulation, hence an improvement of almost 50% was achieved. In the model presented in this chapter, an emphasis is placed on predicting deformation, but the domain decomposition approach allows for elasticity in the model, which makes it more accurate for predicting forces and internal stresses. However, if the rigid plasitc model is solved under the same conditions as [44] with the same number of elements, it would be expected to take only about 30 minutes.



(a) 3D image of product taken on a stereo camera



(b) Profile of (a) after post-processing

Figure 3.26: Data from a shallow product formed on the Cambridge ISF machine using 140×140×1 mm Al 5251-H22 sheet metal and a spherical tipped tool with a 7.5mm radius.

The ISF model presented here is therefore much faster than the above models, and indeed than most of the other known models used in ISF, but because of the approximations it makes it is also only suitable for a certain range of applications. For a number of research areas, speed is less of an issue and for these researchers continue to use commercial explicit FEA codes, such as LS-Dyna by Dejardin *et al* [32] and Abaqus by Bambach *et al* [21], although often mass scaling is applied and implicit codes are still avoided to prevent excessive computation times. An attempt to analyse the use of commercial FEA for ISF has been made in [20] and also, in [21], it is explained that a careful input of the tool trajectory in Abaqus is required to avoid unrealistic dynamics over the sheet.

3.9 Intermediate stages of the ISF process: the ‘dip’

In the simulation results presented so far, only the final deformation shape of the ISF products were considered, but here some observations are made of the deformation behaviour at intermediate stages of the forming process. Figure 3.26 a) displays a 3D image taken from

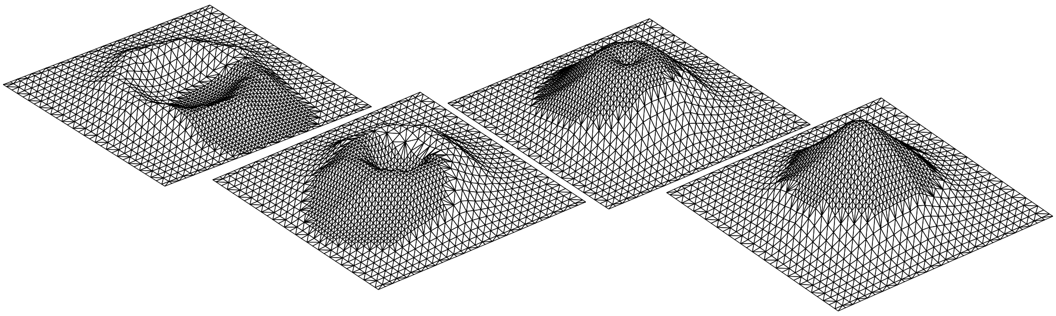


Figure 3.27: Simulation of test product in figure 1.14 using the QMG re-meshing strategy shown at different stages of the process. The tool position is at the centre of the fined region of the mesh. Left to right: 25%, 50%, 75% and 100% through the process.

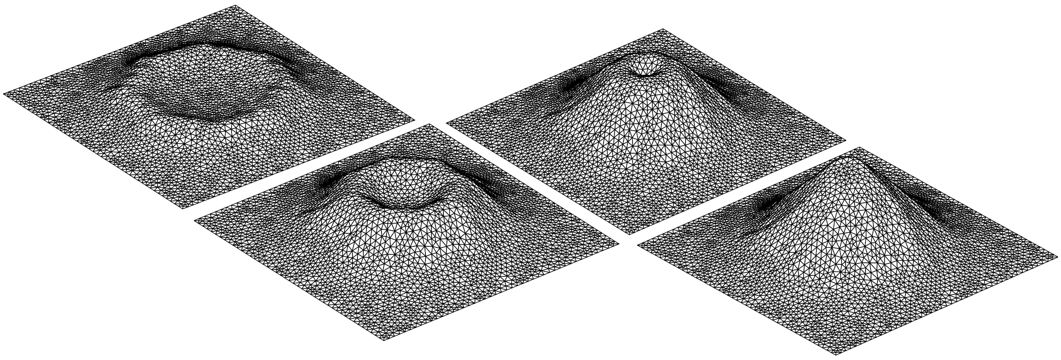


Figure 3.28: Simulation of test product in figure 1.14 using the irregular mesh shown at different stages of the process. Left to right: 25%, 50%, 75% and 100% through the process.

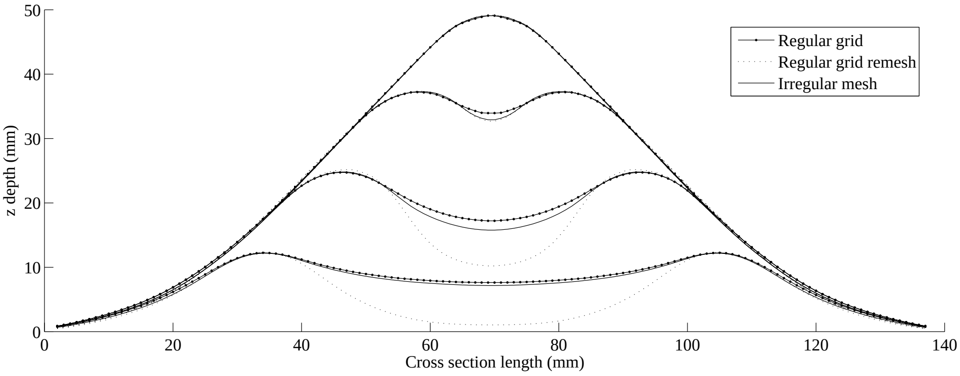


Figure 3.29: Average forming depth from the centre to the edge of the sheet predicted by the limit analysis model at four stages (25%, 50%, 75% & 100%) of the forming process for different types of mesh.

the Cambridge SPIF machine after tracking several circular passes over the sheet, each 1 mm below the previous one. The image was taken using a stereo camera just after the tool completed a pass at a depth of 5 mm. A Matlab script was written to process the point cloud data, the details of which are described further in §5.2 and it was used to produce the plot in figure 3.26 b), which shows the average depth over the sheet from the centre out to 60 mm. A ‘dip’ is observed over the central unformed region of the sheet such that the centre of the sheet is ≈0.75 mm below the maximum forming depth.

This ‘dip’ also occurs in numerical simulations, where it is more pronounced as observed in the snap-shots in figures 3.27 and 3.28, which were taken at four stages of a simulation and are identical except for the meshing strategy used. In addition to these 3D images, the meshing strategies can be compared directly in the 2D plot in figure 3.29, where the average depth is plotted at four stages of the simulation process. The plot shows that the observed ‘dip’ appears to vary with the type of meshing strategy used, although for a similar test comparing mesh densities, it was found the level of ‘dip’ remains constant provided that the mesh is reasonably dense. For example, figure 3.30 is a plot of the ‘dip’ simulated by using regular meshes with densities of 800, 1800, 3200 and 5000 elements, showing that for the three denser meshes the solutions are similar.

The one cone and two cone models were also compared in terms of the ‘dip’ they predict and they were both found to be similar, as displayed in figure 3.31. The agreement between the two plots provide further justification for the relaxation applied to the Ilyushin function in the one cone model.

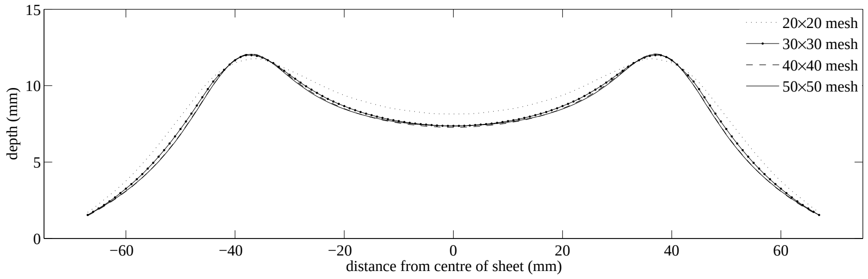


Figure 3.30: Level of dip predicted by the one cone model after completing 25% of the forming process for a twisted cone shaped part. Regular meshes of four different densities are compared, showing that the solution settles down above 1800 elements (30×30 mesh).

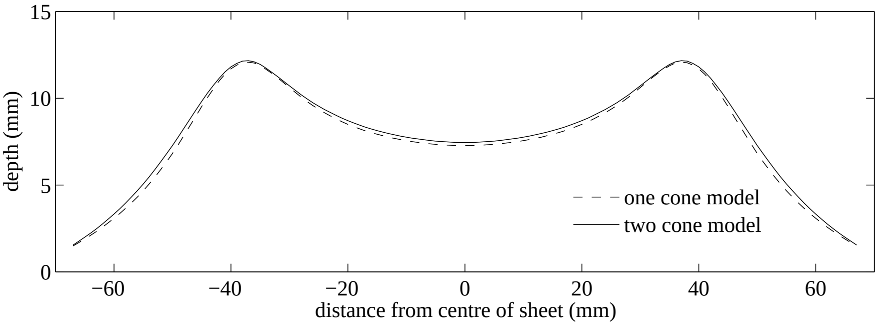


Figure 3.31: Level of dip predicted after completing 25% of the forming process for a twisted cone shaped part. The one cone and two cone models are compared for a 40×40 mesh.

A mismatch in the ‘dip’ between simulation and experiment is not an issue if only final deformation shape is of interest, particularly if the tool-path covers the whole region of the target product. However, when simulating products with tool paths that do not cover the whole surface of the target shape, such as the one in figure 1.10, large differences might be observed between the simulated product and actual product. It is therefore important to isolate any conditions that can reduce the ‘dip’ in simulations, to ensure that the model matches the observed behaviour accurately. From the plot in figure 3.29, the first observation is that a regular mesh should be used, since it gives the smallest ‘dip’ out of the meshing strategies applied here. Further observations can be made by comparing the model with an elasto-plastic explicit model that was produced in Abaqus using a frictionless tool-sheet contact condition.

Two Abaqus simulations were produced for this comparison and the results are plotted in figure 3.32, which was created in the same way as figure 3.29. Both of the Abaqus models used the same tool path, but with different indentation step sizes, and they took approximately 15 hrs 30 mins on a Linux CPU to reach the stage in the simulation displayed

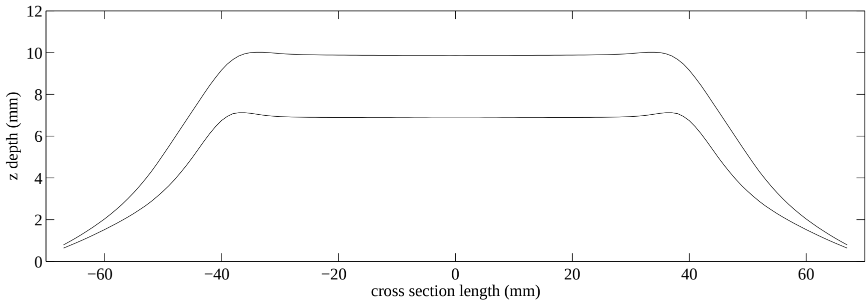


Figure 3.32: Average forming depth in Abaqus simulation using frictionless contact conditions. The two plots used the same tool-path but with different indentation depths between each pass of the tool.

in figure 3.32. The models used 4500 triangular shell elements with five through-thickness nodes per element.

There are several methods for reducing computation times or altering the material behaviour in Abaqus, such as mass scaling, but these were not explored in detail here. Indeed, although it was not observed in figure 3.32 it is possible to simulate a large 'dip' in Abaqus under certain conditions, as shown in figure 3.33. This was taken from a simulation produced in Abaqus 6.5 by Otegi *et al* [89], but the details of this simulation were not provided so it was not possible to compare them with the Abaqus simulations in figure 3.32. The difference between figures 3.32 and 3.33 serves as yet further evidence of the degree of variability that can exist in metal forming simulations, even when the same software package is used. For this reason, the metal forming research community is annually challenged to simulate a set of benchmark processes or shapes and the results are published as a supplement to the proceedings of the Numisheet[1] conferences.

Otegi *et al* refer to the 'dip' as a 'pillow effect' and interestingly, they argue that this effect causes geometric inaccuracies in their TPIF process, which uses a negative die and one moving tool. They believe that the 'dip' is caused by the accumulation of material in the centre of the sheet and proceed to design a multi-pass tool-path that attempts to spread material from the centre of the sheet outwards. This results in an improvement in geometrical accuracy, suggesting that their theory could be correct. Aside from this, it is also interesting that Otegi *et al* approximate the tool shape to a rigid body sphere.

Despite the above suggestion, it is unlikely that material accumulation is the cause

---

[1] The full title for Numisheet is The International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes. The latest benchmarks are published in [50].

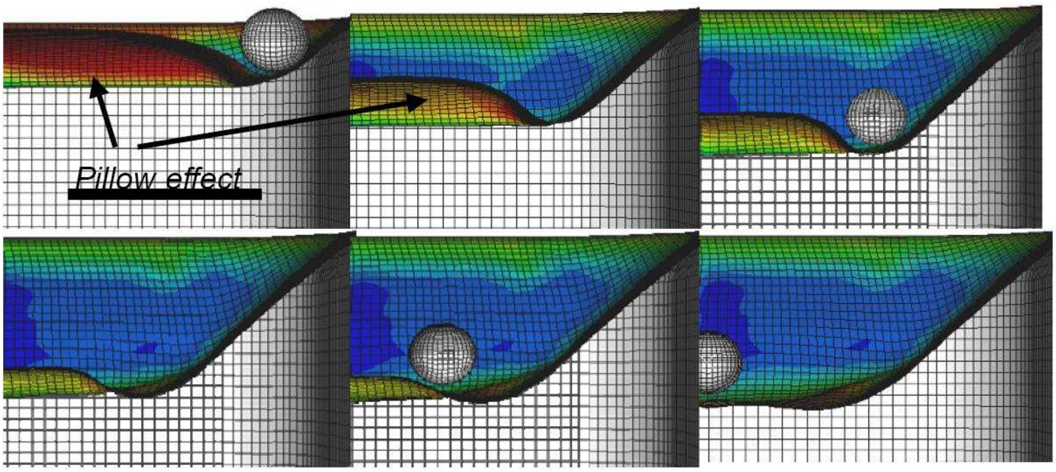


Figure 3.33: The pronounced ‘dip’ observed in Abaqus simulations by Otegi *et al* [89], where it is called the ‘pillow effect’.

of the ‘dip’ in either the rigid plastic model or Abaqus because to simulate material flow accurately, solid element models should be used rather than the shell approximations used here. The causes of the pronounced ‘dip’ in the rigid plastic limit analysis model, or the lack of it in our Abaqus simulations, are likely to be due to a number of interacting factors, several of which are now considered. The main approach in this discussion is to break down any assumptions that were made in the new model that were not present in the Abaqus model and to assess whether any of these could have an effect in forming the ‘dip’. Furthermore, the level of the ‘dip’ is assumed to be directly related to the localisation of deformation around the tool. The assumptions in the model made fall into the following categories:

- The material description
- The material geometry
- The tool-sheet contact state

Material description

Several materials have been observed that give a more pronounced ‘dip’ than the aluminium 5251-H22 alloy sheet metal used in the experiments. Ambrogio *et al* [9] and Micari *et al* [78] observed this phenomenon, which they also refer to the as a ‘pillow effect’, in their experiments with the annealed aluminium alloy AA1050-O and so did Otegi *et al* [89], with low-carbon steel FeP04 sheets. Figures 3.34 and 3.35 display examples of two other such

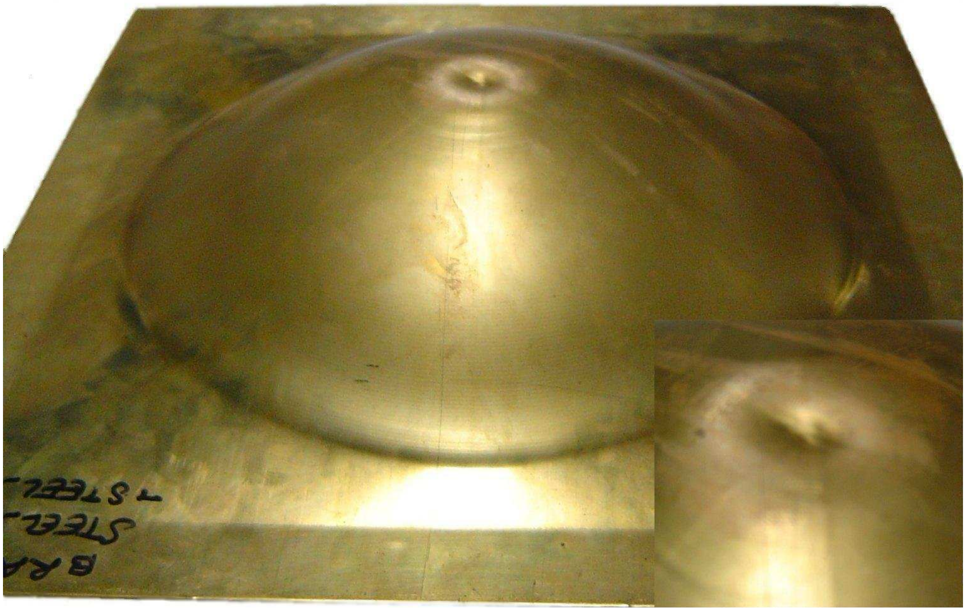


Figure 3.34: Brass musical drum produced on the Cambridge ISF shown prior to trimming. The 'dip' in the unformed central point of the sheet is enlarged on the bottom left of the image.

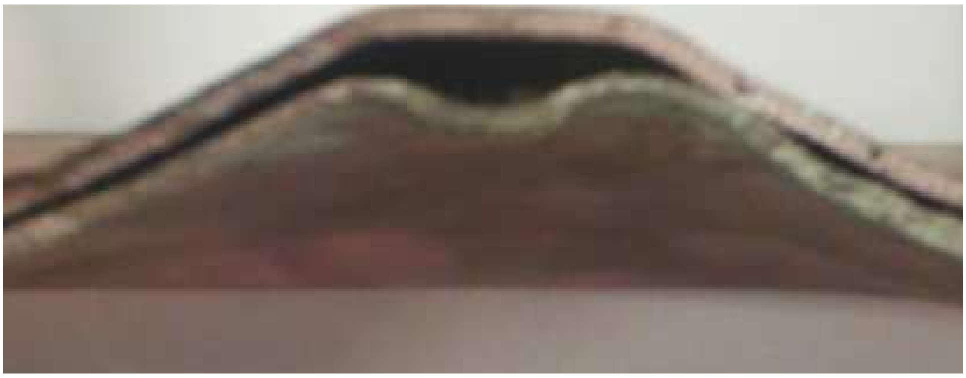


Figure 3.35: Cross section through incrementally formed pure copper (top) and annealed copper (bottom). Both specimens are 0.25 in thick and were subjected to the same toolpath.

materials that were formed on the Cambridge ISF machine for some experiments described in [58]. The first part is a brass sheet that has been formed over the whole surface except for the centre and the second shows two small thick sheets of copper; one untreated and one annealed. The annealed copper displays a 'dip' during forming, whereas in its pure form it remains flat because annealed cooper, like brass and indeed annealed aluminium, are softer materials that deform relatively more locally to the tool.

As an aside, one could question why Al 5251-H22 was used for the previous model validation tests instead of brass, which displays a significant 'dip' and so can be considered to be described more accurately in the numerical model. This is because it is the standard sheet metal used on the Cambridge ISF machine and also, other than the brass product in figure 3.34, which was produced at the IfM during an investigation into methods for manufacturing steel drums with improved musical qualities, the scope of applications for incrementally formed brass is limited when compared to aluminium alloys and in particular Al 5251-H22, which is designed for sheet metal working. Steel is also commonly used and it would also be a good test material, although it requires relatively higher forming loads than aluminium.

The aluminium alloy Al 5251-H22 undergoes a large degree of work-hardening compared to the softer materials described above, but this factor has been neglected in the material description so far. In particular, one effect of work hardening could be to increase the area of deformation caused by the tool at later stages of the process, thus reducing the localisation of deformation. This is because previously work-hardened regions of the sheet near the tool may be more resistant to deformation when compared to non work-hardened regions further away from the tool. A method for incorporating this effect is described in §3.11 and the effect this has on the ‘dip’ is also discussed.

Another factor in the material description, which is the key difference between Abaqus and the formulation here, is the rigid-plastic assumption. The inclusion of elasticity in Abaqus has the effect of providing an ‘energy-free’ mechanism for the shrinkage of previously formed regions of the sheet. This is considered to be a factor that could reduce the ‘dip’ because a shallow region over a sheet occupies less surface area than a region with a large ‘dip’ and during elastic spring-back, the surface area of the sheet reduces. To test this concept using the current formulation, the rigid plastic assumption would have to be converted into an elasto-plastic one, which would be inefficient to solve because it cannot be cast as an SOCP. However, it would be expected that the simulation times would be similar to standard elasto-plastic solvers such as Abaqus.

In the 1980s and early 1990s, when there was more interest in the rigid plastic assumption, some work was done on combining simplified elasticity descriptions with the rigid plastic method to avoid the computing times needed for a full elasto-plastic scheme. Elastic-springback occurs when the residual stress that builds up in a material during loading is released, but such residual stresses do not develop when using the rigid plastic

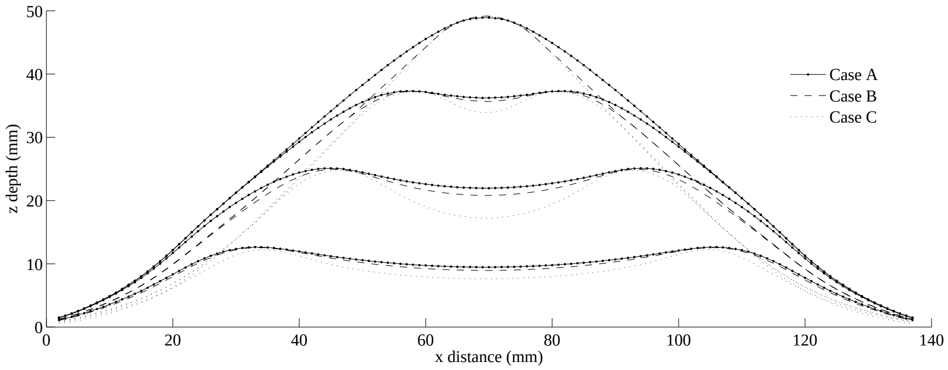


Figure 3.36: Variation of ‘dip’ using a regular mesh with increasing thickness from case C to case A. By increasing thickness, greater emphasis is placed on bending energy, explaining the reduction in curvature over the sheet with increasing thickness.

assumption. In [87], as a rough predictor for spring back, it is suggested that the stress field computed at the final time-step of a rigid plastic simulation can be transferred to an elasto-plastic solver and used to compute the change in deformation that would result from releasing this stress. In [82], an alternative method is described that can adaptively switch between rigid plastic and elasto-plastc elements so that the elasto-plastic formulation is only used on the relevant elements. Those elements that are expected to undergo large strains are converted to rigid plastic elements and the greater the proportion of rigid plastic elements, the faster the simulation. For future research, it may prove to be productive to investigate incorporating either of these two methods into the model developed here.

Material geometry

In terms of material geometry, the thickness of the sheet, h, is of particular interest because it can alter the ratio of stretching work against bending work in the model. As sheet thickness increases the bending work becomes more significant, because the plastic bending work increases as h^2, whereas stretching work varies linearly with thickness. The plot in figure 3.36 labelled Case A uses h=8 mm, Case B uses h=4 mm and Case C uses h=1 mm. The figure shows that the effect of increasing the thickness and therefore the bending energy is to reduce the amount of curvature over the sheet. This in turn reduces the ‘dip’, but also causes more splaying in the outer regions of the sheet, especially for h=8 mm. A

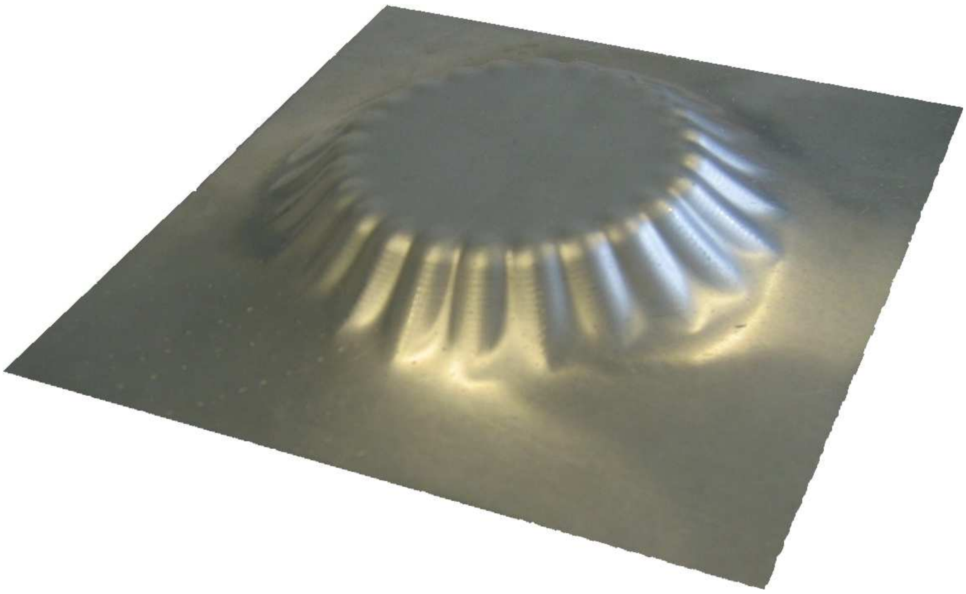


Figure 3.37: Product formed by using a 'discrete' tool-path on the Cambridge ISF machine.

sheet of 8 mm thickness would actually be too thick to form on the Cambridge ISF machine because of the greater forming forces that would be needed, but if it were possible, a solid element model would probably be more appropriate than the shell formulation used here. There is no definitive method for knowing when a solid or a shell model is more suitable, although a ratio defined in [30] can be used to help make the decision. If the smallest radius of curvature over the sheet is defined as R_{min} and the sheet thickness is h, the ratio states that ideally,

$$\frac{R_{min}}{h} > 20 \tag{3.75}$$

should be satisfied when using a shell model. This is described as a heuristic condition in [30] so it is not strictly necessary, but it is almost always satisfied for the thin sheets used in ISF.

So far, the sheet thickness has been kept constant during the process, whereas in reality sheet-thinning will occur as process progresses. Owing to the observed effect in figure 3.36 that altering the sheet thickness can have on the model, it is important to adaptively update the thickness of each element in the model at each time-step. A simple method for doing this is described and implemented in §3.11.

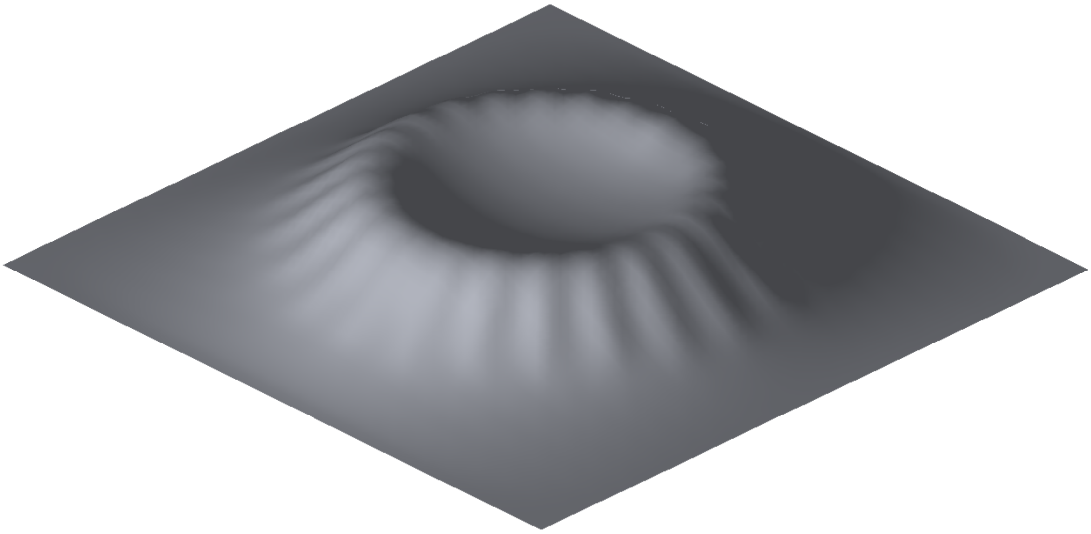


Figure 3.38: Simulation of test product in figure 3.37 using the one cone rigid plastic FE model, with a ‘discrete’ tool path.

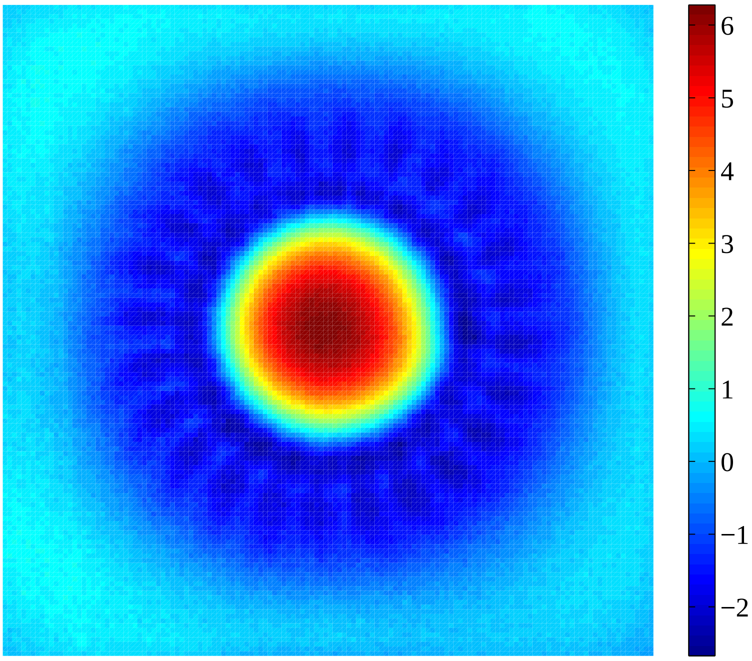


Figure 3.39: Difference in mm between the product in figure 3.37 and the one cone simulation, showing how large the ‘dip’ in the simulation is compared to the actual part. RMS=1.510 mm, Variance=2.248 mm^2, Max. difference=6.278 mm.

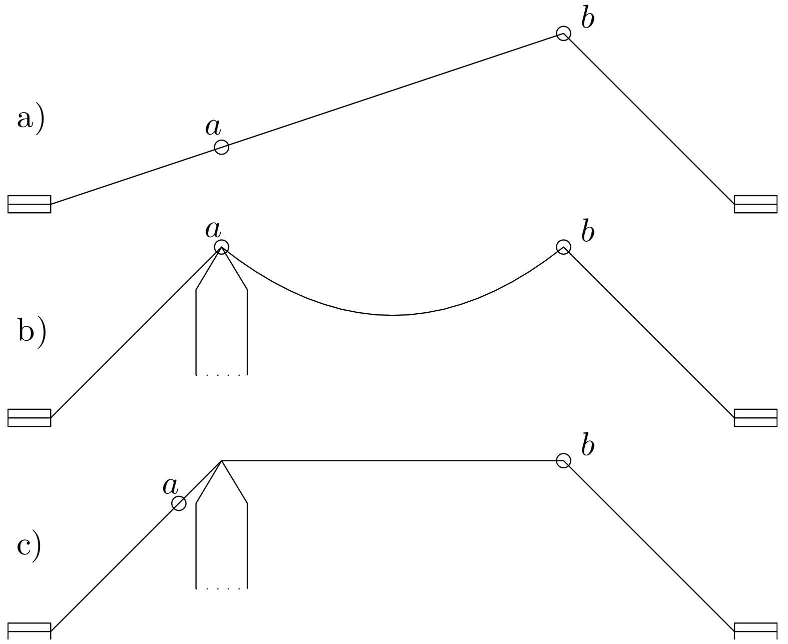


Figure 3.40: The possible effect of a rough contact state on the 'dip'.

Tool-sheet contact state

The final factor discussed in this section is the tool-sheet contact condition because in some respects, this is currently crudely implemented. Since the tool path is discretised in the limit analysis model, the continuous tool path is represented by a series of point loads. This is like applying the tool vertically during ISF and then lifting it away from the sheet before moving it to the next point in the tool-path and then repeating the process. The product in figure 3.37 was produced using a more representative 'discrete' tool-path of this form. However, even with this tool-path the simulation of the same product using the one cone model, as shown in figure 3.38, displays a much larger 'dip' than the actual product does. The large difference between the actual part and the simulation is shown in figure 3.39. This could be because of a more significant underlying issue with the tool constraints, such as for instance the fact that the nodes in the region of the tool are only allowed to move vertically, which is a restriction that invokes a full friction, or rough, contact state because no sliding over the tool is permitted.

The reason why a rough contact state may further increase the 'dip' is given by considering the 2D example of a rigid plastic cord, such as the one drawn in figure 3.40 a), where point b on the cord has been displaced upwards by a tool. The tool is then moved to point a where it indents the cord again. If point a is not permitted to slide over the tool, the distance between a and b is reduced as shown in figure 3.40 b). For a rigid plastic material, the only way that the element can expend no work under loading is if it does not deform, either by stretching or bending, so in terms of the energy minimisation problem under the rigid plastic assumption, it would not be cost effective for the cord between a and b to completely shorten, resulting in a 'dip' in the cord. In an elasto-plastic model, some shortening between a and b would occur from elastic recovery but also, with the contact state, if a was allowed to slide way from the point of contact, as shown in 3.40 c), a minimal energy configuration might be found that displays a reduced dip. An ISF model in Abaqus was run with a 'discrete' tool path under both frictionless and rough contact conditions. The result, shown in figure 3.41, confirms that point loading can cause a 'dip' and that it increases for a rough contact state, but only if the indentation depth is large.

The contact state is therefore unlikely to be a large factor causing the 'dip', but for completeness the current description of the tool-sheet contact state is improved in the following section (§3.10) to permit sliding. As expected, it is found that this improvement does not significantly reduce the 'dip' produced during a simulation of the part in figure 3.37, but in §3.11 it is then shown that thickness changes and work-hardening are the more significant factors and by including their effects in the model the 'dip' is significantly reduced. Furthermore, having considered the key mechanisms that could affect the 'dip', it is concluded that the small deficit that remains in the 'dip' between some of the elasto-plastic simulations and this model, after including work hardening and thickness effects, is due to the lack of elasticity.

3.10 Improved tool constraint description

In previous treatments of the tool constraint in the rigid plastic FEM model, nodes within the vertical projection of the tool onto the sheet were constrained such that they could only move vertically. The location of these nodes were stored in the vectors $\mathbf{x}_{tool}$, $\mathbf{y}_{tool}$ and $\mathbf{z}_{tool}$ and their velocities were $\mathbf{u}_{tool}$, $\mathbf{v}_{tool}$ and $\mathbf{w}_{tool}$ respectively. The tool constraints in terms of

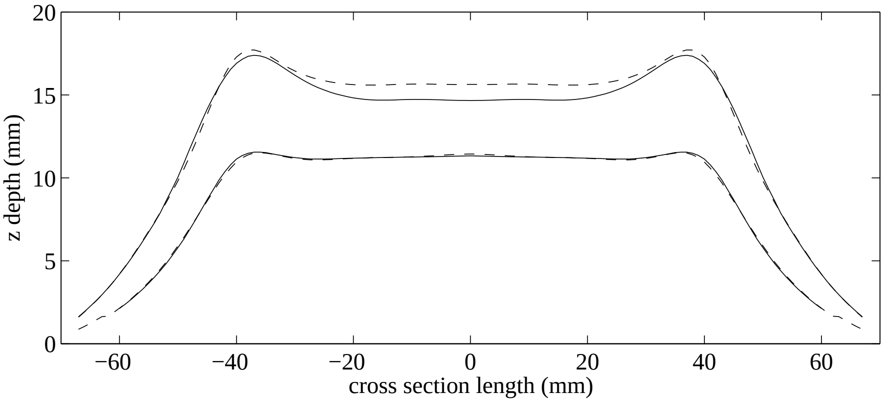


Figure 3.41: Average forming depth in a simulation of a tool path defined as a sequence of point loads. The deeper shape used the same tool-path as the shallower shape, but with larger indentation steps between each pass of the tool. The dashed lines used a frictionless contact condition and the solid lines a rough contact state.

these variables were

$$\mathbf{u}_{tool} = 0, \tag{3.76}$$

$$\mathbf{v}_{tool} = 0, \tag{3.77}$$

$$\mathbf{w}_{tool}\mathrm{d}t \geq \mathbf{t} - \mathbf{z}_{tool}, \tag{3.78}$$

where $\mathbf{t}$ was the tool surface function and $\mathrm{d}t$ could be set to 1.

In practice, the velocities $\mathbf{u}_{tool}$ and $\mathbf{v}_{tool}$ are non-zero but they were small enough to be considered insignificant in the model. However, it would be preferable to relax this restriction and to allow the sheet to slide over the tool. A method for doing this is described here that does not compromise the structure of the model or add any significant burden on the computational demands of the model.

The initial approach one might take is to exploit the spherical shape of the tool surface, which can be described as a norm, by attempting to write the contact state as a second-order cone constraint. For a point (x, y, z) on the surface of the tool the equality,

$$\|[x - x_p, \quad y - y_p, \quad z - z_p]\| = t_r, \tag{3.79}$$

is satisfied, where t_r is the tool radius and (x_p, y_p, z_p) is the position of the centre of the tool. Hence, for a node within the region of the tool, the tool constraint in terms of (3.79)

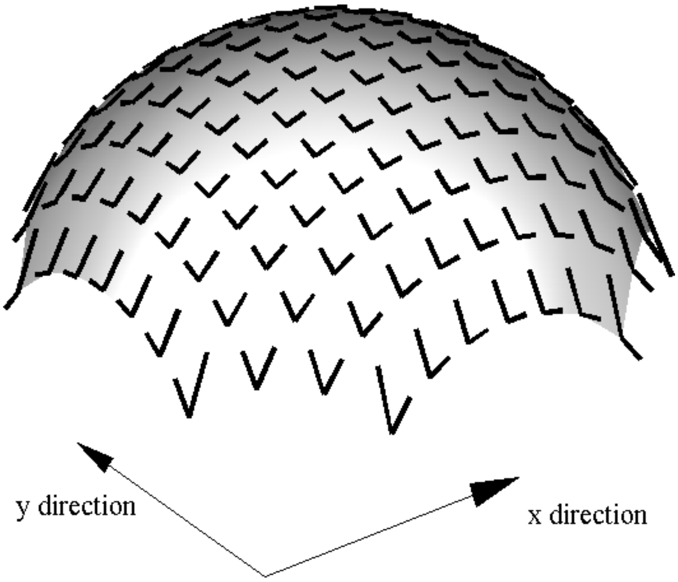


Figure 3.42: Gradients at each node.

is

$$\left\| \begin{bmatrix} \mathbf{x}_{tool} + \mathbf{u}_{tool}\mathrm{d}t - x_p \\ \mathbf{y}_{tool} + \mathbf{v}_{tool}\mathrm{d}t - y_p \\ \mathbf{z}_{tool} + \mathbf{w}_{tool}\mathrm{d}t - z_p \end{bmatrix} \right\| \geq t_r. \tag{3.80}$$

Sliding is permitted by allowing $\mathbf{u}_{tool}$ and $\mathbf{v}_{tool}$ to vary, but due to the direction of the inequality in (3.80), the constraint is not a second-order cone constraint; it is also non-convex and cannot easily be implemented, so further thought is required to construct a feasible sliding constraint.

3.10.1 Sliding contact using linear constraints

The approach taken here is to linearise the tool surface about each node position in ($\mathbf{x}_{tool}$, $\mathbf{y}_{tool}$) and to define linear tool constraints for each node. By rearranging (3.79) to,

$$z = z_p + \sqrt{t_r^2 - (x - x_p)^2 - (y - y_p)^2}, \tag{3.81}$$

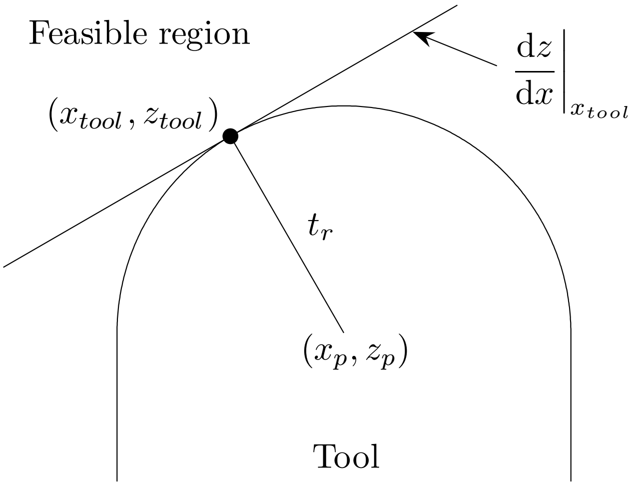


Figure 3.43: Example of linear sliding constraint in 2D with only one node within the tool region at (x_{tool}, z_{tool}).

the derivatives of the tool surface in the x and y directions are obtained,

$$\frac{\partial z}{\partial x} = \frac{-(x-x_p)}{\sqrt{t_r^2-(x-x_p)^2-(y-y_p)^2}}, \tag{3.82}$$

$$\frac{\partial z}{\partial y} = \frac{-(y-y_p)}{\sqrt{t_r^2-(x-x_p)^2-(y-y_p)^2}}. \tag{3.83}$$

In the figure 3.42, these derivatives are evaluated and displayed as tangential lines at a set of regularly spaced points over the tool surface. If these points refer to nodal positions, the tool constraint is defined such that the nodes are constrained by the plane defined by the two tangents (3.82) and (3.83) at each node. This tool constraint can be written in the linear form

$$\mathbf{z}_{tool} + \mathbf{w}_{tool}\mathrm{d}t \geq \mathbf{t} + \left.\frac{\partial z}{\partial x}\right|_{x_{tool}} \mathbf{u}_{tool}\mathrm{d}t + \left.\frac{\partial z}{\partial y}\right|_{y_{tool}} \mathbf{v}_{tool}\mathrm{d}t, \tag{3.84}$$

or, by placing all the variables on the left hand side,

$$\mathrm{d}t \begin{bmatrix} -\left.\frac{\partial z}{\partial x}\right|_{x_{tool}} & -\left.\frac{\partial z}{\partial y}\right|_{y_{tool}} & 1 \end{bmatrix} \begin{bmatrix} u_{tool} \\ v_{tool} \\ w_{tool} \end{bmatrix} \geq t - z_{tool}, \qquad \begin{array}{l} \forall \;\; x_{tool}, y_{tool}, \in \mathbf{x}_{tool}, \mathbf{y}_{tool}, \\ z_{tool}, u_{tool}, v_{tool} \in \mathbf{z}_{tool}, \mathbf{u}_{tool}, \mathbf{v}_{tool}, \\ w_{tool}, t \in \mathbf{w}_{tool}, \mathbf{t}. \end{array} \tag{3.85}$$

As long as u_{tool} and v_{tool} remain small, this constraint is similar to approximating the tool surface by a set of linear facets. The diagram in figure 3.43 provides a further visual representation of the tool constraint, where it is reduced to 2D for simplicity and only one node at (x_{tool}, z_{tool}) is displayed. The 2D equivalent of the tool constraint in (3.85) is

$$\mathrm{d}t \begin{bmatrix} -\left.\dfrac{\mathrm{d}z}{\mathrm{d}x}\right|_{x_{tool}} & 1 \end{bmatrix} \begin{bmatrix} u_{tool} \\ w_{tool} \end{bmatrix} \geq \mathbf{t} - z_{tool}, \tag{3.86}$$

which permits the variables u_{tool} and w_{tool} to take values that move the node anywhere within the area labelled 'feasible region' in the figure.

3.10.2 Results

The one cone model was modified to incorporate the new sliding constraints and used to simulate the product in figure 3.37. A comparison between the simulation and the actual part was made in the same way as figure 3.39, as shown in figure 3.44, from which it appears that the sliding contact state has made very little difference to the solution. The error measures for this simulation are in fact slightly worse than for figure 3.39.

For the simulation of a SPIF process, one can conclude that the previous description of the contact state without sliding was adequate. However, there may be other situations a sliding contact state is more significant, so it is still useful to have a prescribed method such as the one described here for doing this.

3.11 Incorporating work-hardening and thickness changes

The effects of work-hardening and thickness changes in the sheet were briefly touched on in §3.9 and they are now addressed in detail here. Some modifications to the ISF model are made to incorporate these effects and some simulation results are presented. Both modifications can be implemented within an existing loop in the Matlab code so they do not place any extra demands on the processor and the computing time is not compromised.

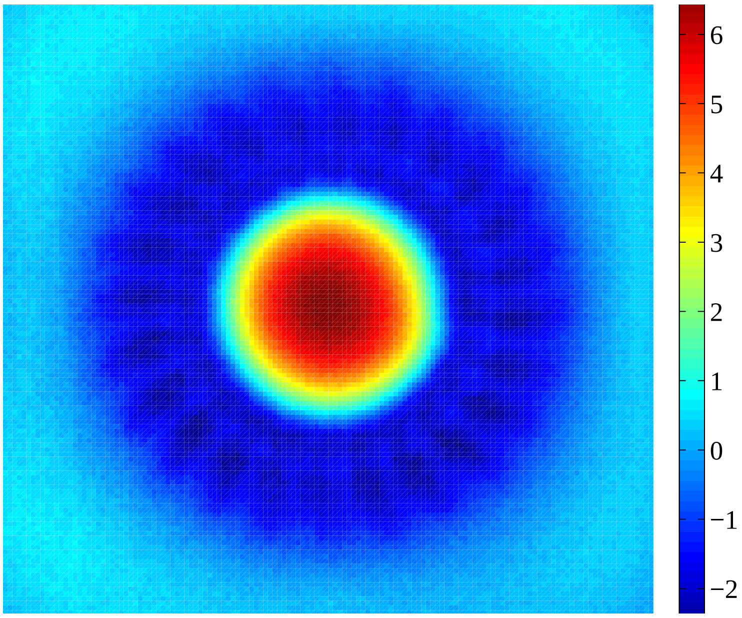


Figure 3.44: Difference in mm between the product in figure 3.37 and the one cone simulation with a sliding contact state, showing how large the 'dip' in the simulation is compared to the actual part. RMS=1.630 mm, Variance=2.632 mm^2, Max. difference=6.748 mm.

3.11.1 Thickness changes

The sheet thickness h is currently treated as a constant that is defined at the beginning of the simulation and then used in (3.71) and (3.74), but a more realistic description of the state of the sheet would be to allow the thicknesses to vary. A simple method for doing this without affecting numerical efficiency is to replace h in (3.71) and (3.74) by a variable $h^{(e)}$ that is updated for each element e at each time-step i.

Assuming the material is incompressible and using the principle of the conservation of volume, if $A_o^{(e)}$ is the initial area of an element and h is the initial thickness of the sheet, which is assumed to be uniform, then at some time-step i, the ratio $(hA_o^{(e)} : h_i^{(e)}A_i^{(e)})$ is preserved, where $A_i^{(e)}$ is the area of element e at the current time-step. The value of $A_i^{(e)}$ is not known at the beginning of time-step i since its value is computed over the time-step, so instead the area $A_{i-1}^{(e)}$ and the corresponding thickness value, $h_{i-1}^{(e)}$, where

$$h_{i-1}^{(e)} \approx \frac{hA_o^{(e)}}{A_{i-1}^{(e)}} \tag{3.87}$$

is used to calculate $\tilde{\mathbf{\Phi}}_e$ from (3.71) and to update M_0 and N_0 from (3.71) at the beginning of step i. These values are then kept constant until the end of the time-step when they are recalculated using the new area $A_i^{(e)}$.

	Si	**Fe**	**Cu**	**Mn**	**Mg**	**Cr**	**Zn**	**Ti**	**Others**	
Min				0.10	1.70				Each	Total
Max	0.40	0.50	0.15	0.50	2.40	0.15	0.15	0.15	0.03	0.15

Table 3.5: Percentage of alloy present in Al 5251-H22.

3.11.2 Work-hardening

The basic composition of Al 5251-H22 is listed in table 3.5 and it shows that the quantities of the impurities added to form the alloy are very small, yet it displays very different properties to pure aluminium during yielding. For instance, at $\sigma_y = 130$ MPa the yield stress of the alloy is higher than pure aluminum and it exhibits significant non-linear work-hardening up to a final tensile strength of 200-240 MPa. This work-hardening effect has not been represented in the model so far, but it is possible to incorporate it by modifying the cost function at each time step. Specifically, the cost coefficient for the energy associated with a particular element can be increased according to the amount of strain it underwent in the previous time step. This amounts to modifying the cost function in (3.70) to

$$\sum_{i}^{3NE} \frac{1}{3} f(\epsilon_{t-1,i}) z_i, \tag{3.88}$$

where, given the strain state $\epsilon_{t-1,i}$ of an element associated with node i, the function $f(\epsilon_{t-1,i})$ computes the increase in the yield stress due to work hardening relative to the initial yield stress. The strain measure $\epsilon_{t-1,i}$ could be chosen to be the equivalent strain $\bar{\epsilon}$ so that with reference to figure 3.45, $f(\epsilon_{t-1,i})$ returns a scalar corresponding to the vertical axis in the plot, which displays the ratio of equivalent stress $\bar{\sigma}$ against the yield stress σ_y. Under the rigid perfectly plastic assumption the equivalent stress was set to a constant value of σ_y, as shown by the straight line in the figure whereas the curve displays an example of how $\bar{\sigma}$ could vary non-linearly with $\bar{\epsilon}$ in a modified work hardening model.

3.11.3 Results

The version of the one cone model that was used in §3.10, with the sliding contact state, was modified again to include the thickness variations as described in §3.11.1. As before, the product in figure 3.37 was simulated using a regular mesh with $NE = 4802$ and the difference between the simulation and the actual product was computed. The result in

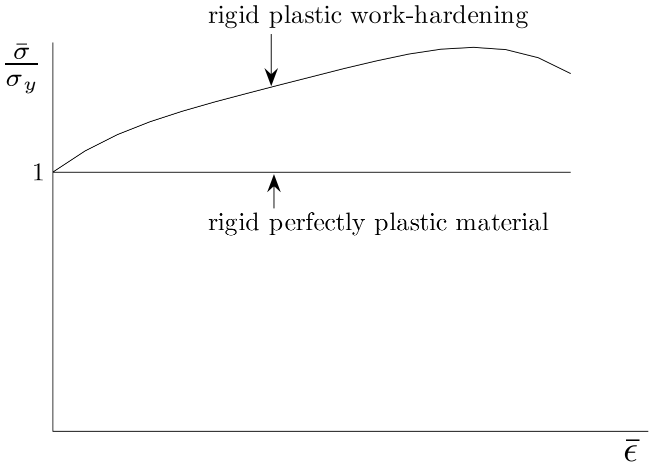


Figure 3.45: Perfectly plastic stress-strain yield curve against one displaying working hardening.

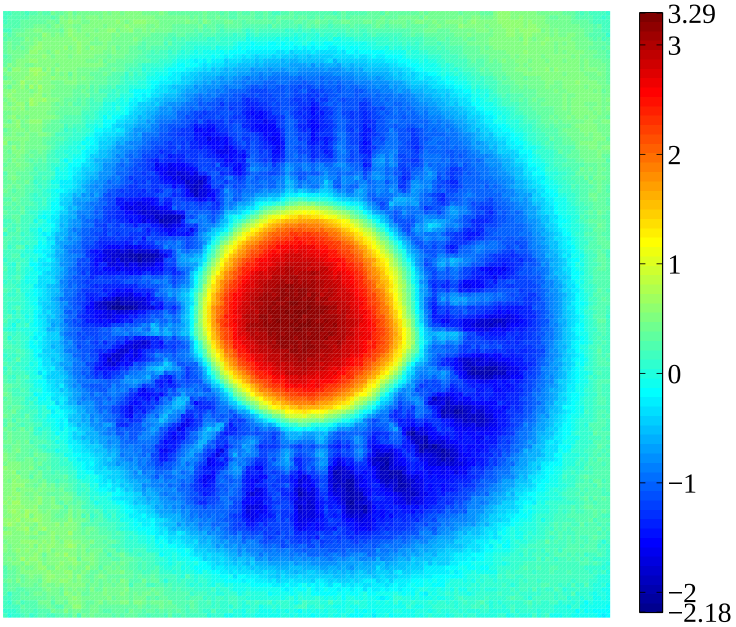


Figure 3.46: Difference in mm between the product in figure 3.37 and the one cone simulation with thickness variation and a sliding contact state, showing a reduced ‘dip’ compared to figure 3.44. RMS=1.0742 mm, Variance=1.1028 mm^2, Max. difference=3.2960 mm.

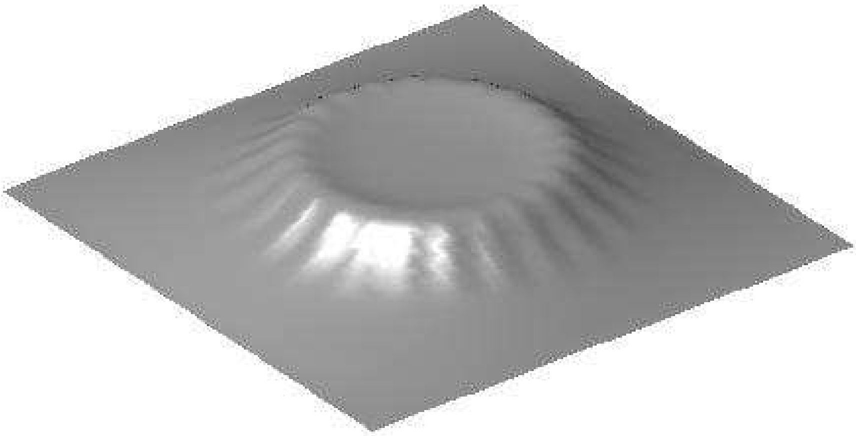


Figure 3.47: Simulation of the product in figure 3.37 using the one cone model modified to include thickness variation, a sliding contact state and linear work hardening.

figure 3.46 shows that the 'dip' has been almost halved from over 6 mm to 3.3 mm, so one may conclude that thickness variation is one factor that must be included in the model in order to describe the intermediate stages of the ISF process appropriately.

The model was then modified further to include work hardening as outlined in §3.11.2. The major stretches over the sheet were used as the strain measure for the work hardening function and, as a simple demonstration of the method, a linear model was chosen where the yield stress over the sheet was increased by a one-to-one ratio with the magnitude change in the major stretches from one time-step to the next. For comparison with figure 3.38 an image of the resulting simulation is shown in figure 3.47 and the difference between the two is displayed in figure 3.48. The 'dip' in this model has been reduced to just 1.81 mm, although to the slight detriment of the fit over the rest of the shape, which is likely to be because the the work-hardening model chosen is not entirely appropriate for this material. The key conclusion drawn from this result is that as well as thickness variations, work-hardening can also significantly affect the 'dip'. From this investigation it appears that these two factors along with elasticity, which has been purposefully omitted here, explain why such a pronounced 'dip' was observed in the initial formulation of the rigid plastic limit analysis FE model. All the error measures for the simulation of the product in figure 3.37 in this chapter have been included in table 4.2 (§4.4.1) with further results from the next chapter.

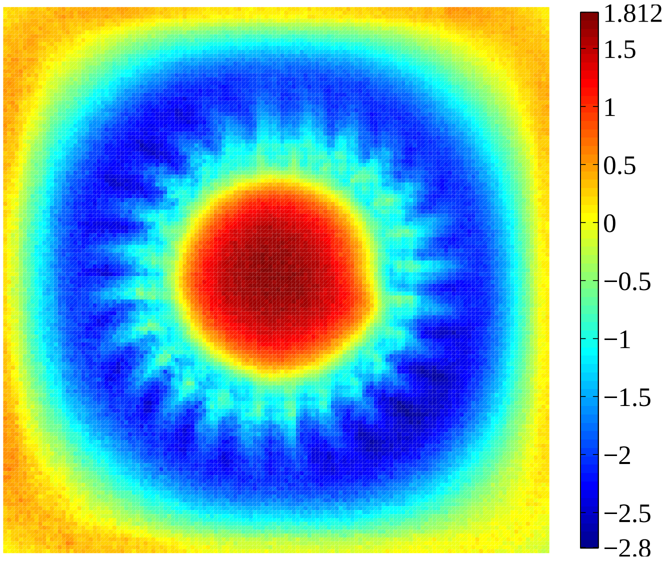


Figure 3.48: Difference in mm between the product in figure 3.37 and the one cone simulation with thickness variation, a sliding contact state and work hardening, showing a reduced 'dip' compared to the simulation in figure 3.44. RMS=1.3503 mm, Variance=1.0990 mm^2, Max. difference=2.8056 mm.

3.11.4 Failure of the sheet

When simulating the standard cone (figure 1.12) and twisted cone (figure 1.13) benchmark shapes, the thickness at some points approached zero indicating that the sheet had torn. It would of course be expected that the sheet should tear after significant straining, but this failure occurs earlier in the simulation than it does in practice. Only the results for simulating the standard cone shape are presented here, since they were similar for the twisted cone shape. Figure 3.49 displays the deformation field along with a colour plot representing the thicknesses over the sheet for a simulation that used no work-hardening, but a similar tear also occurred in a simulation that included work-hardening. The simulation was continued to the end of the process even though the tear occurred earlier, when the tool was at about the depth of the failed region of the sheet represented by the hole in figure 3.49.

As an indication of the behaviour at intermediate stages of the process, the deformation field along the black line drawn on figure 3.49 is plotted in figure 3.50 at four equally spaced periods of the process. Similar plots for the original model with sliding contact, the simulation that also included work-hardening and the LS-Dyna standard cone model (figure 1.12) are also shown in figure 3.50. A comparison with the LS-Dyna data demonstrates that the deficit in the 'dip' has been significantly reduced but not completely eliminated and this, as discussed previously, is likely to be due to the lack of elasticity in the one

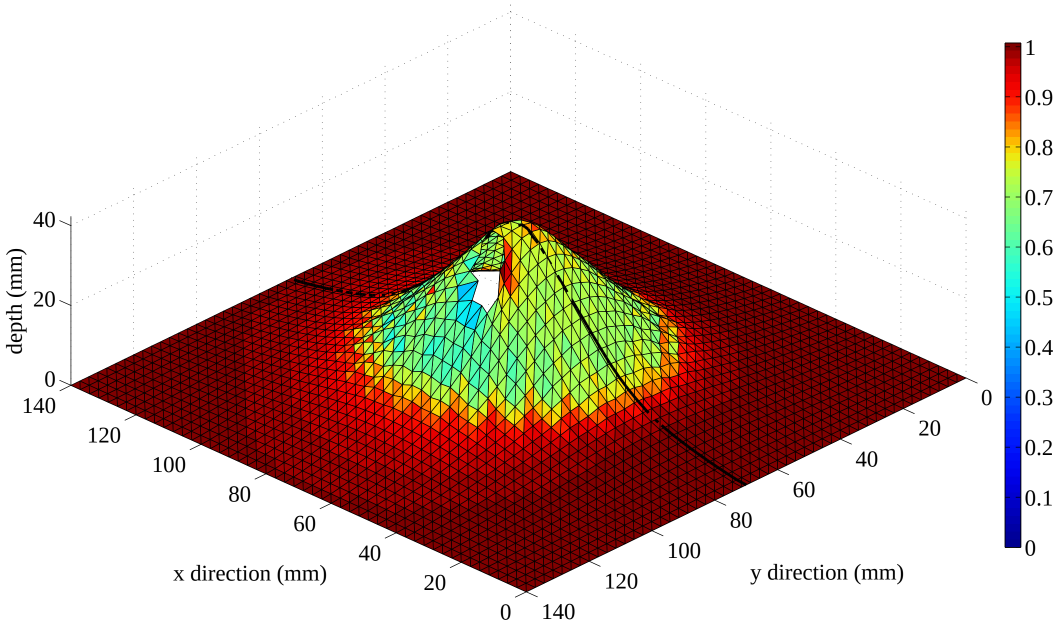


Figure 3.49: Simulation of cone shape with element thicknesses in mm indicated by the color plot. The failed region is displayed as a hole.

cone model. It should be noted that for the work-hardening model, to prevent excessive work-hardening, the stress factor in this model was increased by one tenth of the increase in stretch, rather than the one-to-one ratio used in the previous example in figure 3.47.

The reason why the sheet fails for the deeper standard and twisted cone shapes is not fully understood yet, but because the real product does not fail under these conditions, further work should be conducted to address this problem. One possibility is that the method used here for calculating thickness needs to be changed. For instance, a number of authors have suggested the ‘sine law’ [116] for shear deformation can be used as a measure for the thickness change in ISF and more recently, a detailed study in [58] shows that the actual thinning behaviour is a combination of shear and stretching. Other options that may be investigated are to avoid keeping the thickness constant over each time-step, to check for numerical instabilities, to test different mesh types and densities and to try different methods for computing strain in the sheet. Further studies of these possibilities are kept for future research.

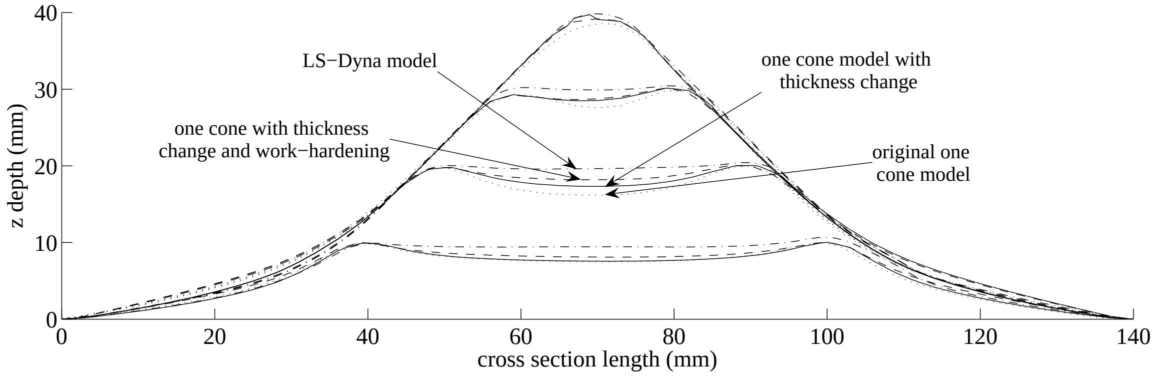


Figure 3.50: Deformation field along the black line in figure 3.49 at four stages of the process. The four models compared are: the original model with sliding contact (dotted); with thickness change added (solid); with work-hardening too (dashed); the LS-Dyna model (dash-dot).

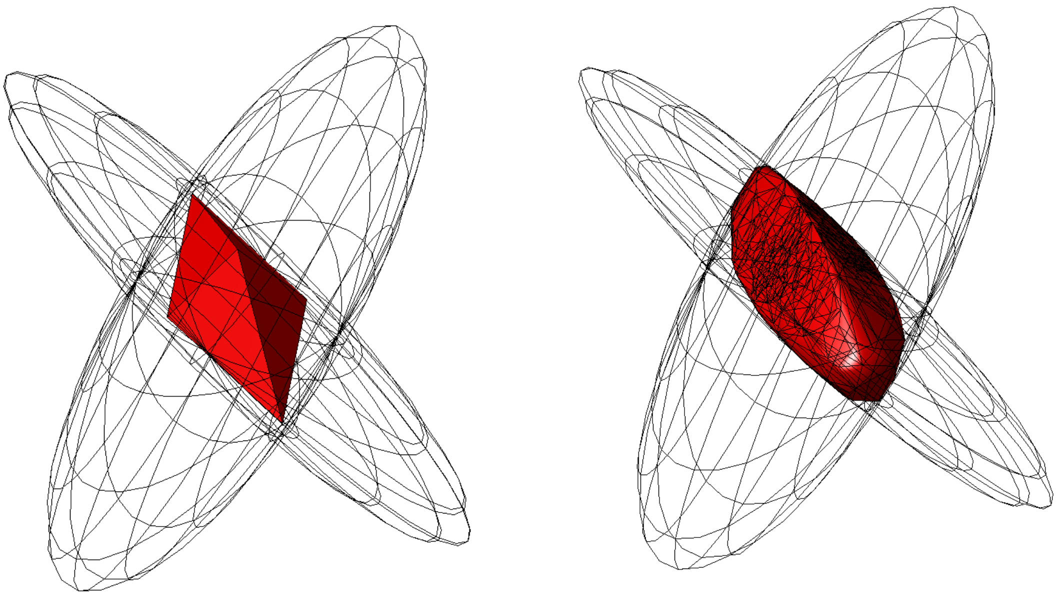
3.12 Linear programming vs SOCP

In general, a linear program (LP) is simpler and faster to solve than an SOCP problem with a similar number of optimisation constraints and variables [24]. It is therefore worth considering whether an LP version of the sequential limit analysis model can be formulated. The idea of using linear programming in FEA has been around for sometime [12] and the typical approaches for doing this are to use a linear yield criterion or to approximate the yield surface by a set of linear facets. The latter is now considered here, where the constraint space defined by the intersection of the ellipsoids $\mathbf{Q}_1$ and $\mathbf{Q}_2$ in the SOCP model in (3.50) is replaced by a set of linear facets.

In order to produce the approximation to the yield surface, a method for generating facets from point cloud data must be described. Given N points located in J dimensional space, the coordinates $(x_{p,1}, \ldots, x_{p,J})$ of each point can be stacked into a matrix,

$$\mathbf{X}_p = \begin{bmatrix} x_{p,1}^1 & \cdots & x_{p,J}^1 \\ \vdots & \ddots & \vdots \\ x_{p,1}^N & \cdots & x_{p,J}^N \end{bmatrix}. \tag{3.89}$$

If $N = J$, a unique linear equation $f(\mathbf{x}) = \mathbf{a}_Q\mathbf{x}$ can be defined that passes through all N points. In 3D, this is the equation of a planar facet that intersects with three points in the space and for higher dimensions, the facet is a hyperplane. In general, for any J the



(a) (b)

Figure 3.51: Intersection of two 3D ellipsoids approximated in (a) by 8 facets with corners that lie on one of the three principle axes and in (b) by 1872 randomly placed facets.

equation of the hyperplane is given by

$$f(\mathbf{x}) = \mathbf{a}_Q\mathbf{x} = a_{Q,1}x_1+,\ldots,+a_{Q,J}x_J = b_Q, \tag{3.90}$$

where $b_Q = \det(\mathbf{X}_p)$ and if $f(\mathbf{x}) < b_Q$, the point $\mathbf{x}$ lies on one side of the hyperplane. The coefficients in (3.90) are given by $a_{Q,k} = \det(\hat{\mathbf{X}}_{p,k})$, where $\hat{\mathbf{X}}_{p,k}$ is a square matrix containing elements $\hat{x}^n_{p,j} = x^n_{p,j}$ (with reference to (3.89)) if $k \neq j$, and $\hat{x}^n_{p,j} = 1$ when $k = j$, where $j, n, k = 1, \ldots, J$.

By representing the yield surface by a point cloud, a facet can be defined for every set of J points in the point cloud. This results in a set of vectors $\mathbf{a}_Q$ and scalars b_Q that are joined up to form a matrix $\mathbf{A}_Q$ and vector $\mathbf{b}_Q$, defined as

$$\mathbf{A}_Q = \begin{bmatrix} \mathbf{a}_Q^{(1)} \\ \vdots \\ \mathbf{a}_Q^{(p)} \end{bmatrix}, \qquad \mathbf{b}_Q = \begin{bmatrix} b_Q^{(1)} \\ \vdots \\ b_Q^{(p)} \end{bmatrix}, \tag{3.91}$$

where p is the total number of facets. By appending the appropriate ± sign to each element in $\mathbf{A}_Q$ and $\mathbf{b}_Q$, the inequality

$$\mathbf{A}_Q\mathbf{s} \leq \mathbf{b}_Q \tag{3.92}$$

can describe any point within the space enclosed by the facets.

The Ilyushin function is six dimensional and cannot be plotted, so in figure 3.51, two 3D ellipsoids are used to demonstrate the above method. In figure 3.51 a), the smallest possible number of facets p of 8 were used, resulting in a poor approximation to the intersection of the two ellipsoids. For the six dimensional case of the Ilyushin function the minimum possible value of p is $2^6 = 64$. A much more accurate description of the intersection of the two 3D ellipsoids is provided in figure 3.51 b), where $p = 1872$.

Using (3.92), the equation for the principle of maximum plastic work in (3.50) becomes

$$\begin{aligned} d_I = \max_{\mathbf{s}} \ & \mathbf{s}^T\mathbf{\Phi} \\ \text{s.t.} \ & \mathbf{A}_Q\mathbf{s} \leq \mathbf{b}_Q. \end{aligned} \tag{3.93}$$

The Lagrangian of (3.93),

$$L = \mathbf{s}^T(\mathbf{\Phi} + \mathbf{A}_Q^T\boldsymbol{\lambda}) - \mathbf{b}_Q^T\boldsymbol{\lambda}, \tag{3.94}$$

is minimised with respect to the primal variables $\mathbf{s}$ to give the dual of (3.93),

$$\begin{aligned} d_I^* = \min_{\boldsymbol{\lambda}} \ & \mathbf{b}_Q^T\boldsymbol{\lambda} \\ \text{s.t.} \ & -\mathbf{A}_Q^T\boldsymbol{\lambda} = \mathbf{\Phi}, \\ & \boldsymbol{\lambda} \geq 0, \end{aligned} \tag{3.95}$$

in terms of the dual variables $\boldsymbol{\lambda}$. The overall energy minimisation problem is

$$\begin{aligned} D_I = \min_{\mathbf{\Phi}} \ & \int_A \min_{\boldsymbol{\lambda}} \mathbf{b}_Q^T\boldsymbol{\lambda} \ \mathrm{d}A \\ \text{s.t.} \ & -\mathbf{A}_Q^T\boldsymbol{\lambda} = \mathbf{\Phi}, \\ & \boldsymbol{\lambda} \geq 0, \end{aligned} \tag{3.96}$$

or after discretisation using BCIZ and Allman elements as previously,

$$\begin{aligned} D_I = \min_{\mathbf{w}} \quad & \left[\sum_{e=1}^{NE} \sum_{i=1}^{3} \min_{\boldsymbol{\lambda}_i^{(e)}} \mathbf{b}_Q^T \boldsymbol{\lambda}_i^{(e)} \right] = \left[\min_{\boldsymbol{\lambda}} \sum_{e=1}^{NE} \sum_{i=1}^{3} \mathbf{b}_Q^T \boldsymbol{\lambda}_i^{(e)} \right] \\ \text{s.t.} \quad & -\mathbf{A}_Q^T \boldsymbol{\lambda}_i^{(e)} = \mathbf{K}\mathbf{w}, \\ & \boldsymbol{\lambda}_i^{(e)} \geq 0, \\ & \mathbf{E}\mathbf{w} \geq \mathbf{t}, \end{aligned} \tag{3.97}$$

where, as previously, $\boldsymbol{\Phi} = \mathbf{K}\mathbf{w}$ and $\mathbf{E}$ and $\mathbf{t}$ describe the tool boundary conditions. Due to the independence between λ and $\mathbf{w}$,

$$\begin{aligned} D_I = \min_{\mathbf{w}} \quad & \sum_{e=1}^{NE} \sum_{i=1}^{3} \mathbf{b}_Q^T \boldsymbol{\lambda} \\ \text{s.t.} \quad & -\mathbf{A}_Q^T \boldsymbol{\lambda}_k = \mathbf{K}\mathbf{w}, \\ & \boldsymbol{\lambda}_k \geq 0, \qquad k = 1, \ldots, 3NE, \\ & \mathbf{E}\mathbf{w} \geq \mathbf{t}. \end{aligned} \tag{3.98}$$

The problem in (3.98) was coded in Matlab and parsed into Mosek, because as well as the SOCP solver, it has an algorithm for large scale LPs that is more efficient than the Matlab Optimisation Toolbox function *linprog*. However, memory problems were encountered using meshes with $NE > 800$ because the yield surface matrix $\mathbf{A}_Q$ is repeated $3NE$ times in the optimisation constraint matrix in (3.98) and the solver was very slow even for smaller problems. The size of $\mathbf{A}_Q$ depends on the number of facets p that are used to approximate the Ilyushin yield surface, hence in theory p should be chosen to be large enough to represent the yield surface sufficiently well without making $\mathbf{A}_Q$ too large. In practice, even when the smallest possible value of $p = 64$ was used the size of the constraint matrix was too large. From this work, one can conclude that too many constraints are required in the LP problem to provide any benefit over the SOCP problem, because only one or two SOCP constraints are required for the one cone and two cone models respectively, to perform that same job as $p \geq 64$ LP constraints. A further advantage with the SOCP is that the yield function does not have to be approximated if the two cone model in §3.5 is used.

Chapter 4

Networked rod structure model

4.1 Grid model

The rigid plastic finite element model in chapter 3 requires a very large number of variables to be defined at each node in the discretised mesh. This is one of the reasons why the full potential increase in computation speed by using the rigid plastic assumption has not been observed. In order to reduce the number of model variables, a third model is presented here, which is developed using some ideas from both chapters 2 and 3, whilst also introducing some new ones. As with (2.7), the formulation begins with the minimisation of internal power dissipation,

$$\dot{W} = \min_{v(t)} \int_{\Omega} \bar{\sigma}\dot{\bar{\epsilon}} d\Omega, \tag{4.1}$$

where $v(t)$ is the rate of change of deformation, $\dot{u}$, $\bar{\sigma} = \sigma_y$ is the material yield stress, $\dot{\bar{\epsilon}}$ is material strain rate and Ω is the volume of the sheet. Considering only the stretching work, the equivalent strain rate is given by

$$\dot{\bar{\epsilon}} = \frac{2}{3} \| [\dot{\epsilon}_x \ \ \dot{\epsilon}_y \ \ 2\dot{\epsilon}_{xy}] \|_2, \tag{4.2}$$

and so under the rigid plastic assumption,

$$\dot{W} = \min_{v(t)} \ \sigma_y \frac{2}{3} \int_{\Omega} \| [\dot{\epsilon}_x \ \ \dot{\epsilon}_y \ \ 2\dot{\epsilon}_{xy}] \|_2 d\Omega. \tag{4.3}$$

The next step is to compute the strain rates in (4.3), which is done at both an infinitesimal

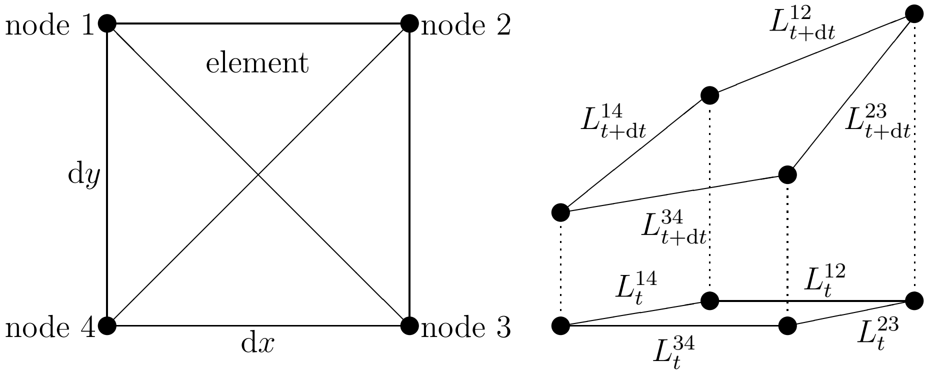


Figure 4.1: Labelling convention for a square region in the plane of the sheet. The nomenclature L_t^{ij} denotes the length of the connection from node i to j at time t.

scale and at a finite scale when the model is discretised, by considering the sheet metal as being built up of square elements as shown in figure 4.1. To reduce the number of variables used in the model, each node is assumed to only move in the vertical direction as in §2.1, with a velocity v_i, where i is the node number.

It is postulated that the in plane strain rates in the x and y direction, $\dot{\epsilon}_x$ and $\dot{\epsilon}_y$, and the shear strain $\dot{\epsilon}_{xy}$ can be approximated by the deformation of a set of six bars that connect nodes 1-2, 2-3, 3-4, 1-4, 1-3 and 2-4 together in each element. The energy related to these strains is equated to the membrane stretching work that is minimised subject to the tool and clamping boundary constraints, resulting in the complete model that will now be referred to as the 'grid model'. The approach taken here was inspired by a discussion in [111], where it is noted that the use of a network of bars and indeed the finite difference method in material deformation, were both featured prominently in academic literature prior to the developments in computational power that lead to the widespread uptake of FEA packages. It is further noted in [111] that such methods have now become sidelined and should receive more attention, especially in applications where FEA can be an over complication. Recent evidence of the effective use of bar networks is in the physical modelling of cloth deformation [121], where although the analysis is predominantly performed in the elastic domain, material non-linearity is also present due to differing behaviour under tension and compression. An application of such cloth models is for the prediction of regions of compression and subsequent wrinkling that lead to weak-spots in the engineering structures that the cloths are used in.

Using a networked bar approach, $\dot{\epsilon}_x$ is given by

$$\dot{\epsilon}_x dt = \frac{1}{2}\left(\frac{L^{12}_{t+dt} - L^{12}_t}{L^{12}_t} + \frac{L^{34}_{t+dt} - L^{34}_t}{L^{34}_t}\right), \tag{4.4}$$

where the lengths of the bars ij at time $t + \mathrm{d}t$, $L^{ij}_{t+\mathrm{d}t}$ and time t, L^{ij}_t, as shown in figure 4.1, are calculated by

$$L^{ij}_{t+dt} = \left\| \begin{bmatrix} \mathrm{d}x \\ (u_j + v_j\mathrm{d}t) - (u_i + v_i\mathrm{d}t) \end{bmatrix} \right\|_2, \tag{4.5}$$

$$L^{ij}_t = \left\| \begin{bmatrix} \mathrm{d}x \\ u_j - u_i \end{bmatrix} \right\|_2. \tag{4.6}$$

Similarly to (4.4), if the transformation that would need to be applied to make $\dot{\epsilon}_x$ and $\dot{\epsilon}_y$ perpendicular to each other is neglected, then

$$\dot{\epsilon}_y dt = \frac{1}{2}\left(\frac{L^{14}_{t+dt} - L^{14}_t}{L^{14}_t} + \frac{L^{23}_{t+dt} - L^{23}_t}{L^{23}_t}\right). \tag{4.7}$$

The work done due to in-plane shear strain is the energy required to distort, or skew, the shape of the element, so the shear strain rate has been approximated by the changes in length of the diagonal bars 13 and 24 in the element, hence

$$2\dot{\epsilon}_{xy} dt = \frac{1}{2}\left(\frac{L^{13}_{t+dt} - L^{13}_t}{L^{13}_t} + \frac{L^{24}_{t+dt} - L^{24}_t}{L^{24}_t}\right). \tag{4.8}$$

After spatial discretisation, $\mathrm{d}x \rightarrow \Delta x$ and $\mathrm{d}y \rightarrow \Delta y$, and the integration over volume, (4.3) becomes a summation over the square elements in figure 4.1,

$$\dot{W} = \min_{\mathbf{v}} \quad \sigma_y \Omega_e \frac{2}{3} \sum_{k=1}^{NE} \|[\dot{\epsilon}^{(k)}_x \;\; \dot{\epsilon}^{(k)}_y \;\; 2\dot{\epsilon}^{(k)}_{xy}]\|_2, \tag{4.9}$$

where $\mathbf{v}$ is a vector that contains the vertical velocities of each node, Ω_e is the volume of an element and NE is the total number of elements. After temporal discretisation, $\mathrm{d}t \rightarrow \Delta t$ and the above expressions for strain rates become strain increments because $\dot{\epsilon}_{xy}\Delta t \approx \Delta\epsilon_{xy}$, so (4.9) can be modified to compute the change in work from time t to time $t + \Delta t$,

$$\Delta W = \min_{\mathbf{v}} \quad \sigma_y \Omega_e \frac{2}{3} \sum_{k=1}^{NE} \| [\Delta\epsilon_x^{(k)} \;\; \Delta\epsilon_y^{(k)} \;\; 2\Delta\epsilon_{xy}^{(k)}] \|_2, \tag{4.10}$$

which can then be solved for the nodal velocities $\mathbf{v}$ using an appropriate optimisation routine subject to tool and boundary constraints. As with the previous models, the deformation field is updated at the end of each time step using an explicit forward differencing scheme,

$$\mathbf{u}_{t+\Delta t} = \mathbf{u}_t + \mathbf{v}\Delta t. \tag{4.11}$$

The sheet movement is not dependent on time, in the sense that there are no time varying components such as vibration in the model. The only dynamic component is therefore the tool position as each increment, so the variable Δt above can be set to 1 without loss of generality. This is equivalent to scaling the tool velocity such that each position increment in the discretised tool-path occurs within a time period of $\Delta t = 1$.

4.2 SOCP formulation

The problem was written as an SOCP by introducing slack variables that, as with the previous models, are defined as inequality constraints that should become equalities at the optimal point. However, as will be seen, the required equality does not always occur and an extra penalty term is added to the energy term in the cost function to help overcome the effect this has. The first set of slack variables are stored in a vector $\mathbf{y} \in \mathbb{R}^{NE}$, which is defined such that

$$\| [\Delta\epsilon_x^{(k)} \;\; \Delta\epsilon_y^{(k)} \;\; 2\Delta\epsilon_{xy}^{(k)}] \|_2 \leq y_k \;\; k = 1, \ldots, NE, \tag{4.12}$$

and a further set of variables $\mathbf{m} \in \mathbb{R}^{6NE}$ are defined that are bounded by the constraints

$$\left\| \begin{bmatrix} \mathrm{d}x \\ u_j + v_j - u_i - v_i \end{bmatrix} \right\|_2 = L_{t+dt}^{ij} \leq m_l, \qquad l = 1, \ldots, 4NE + 2\sqrt{NE}. \tag{4.13}$$

Each pair of node numbers, $ij \in \{12, 23, 34, 14, 13, 24\}$ associated with an element k is allocated by a number l, where $l \in \{6(1-k)+1, \ldots, 6k)\}$. If $\mathbf{v} \in \mathbb{R}^{NN}$, where NN is the total number of nodes, the above can be written as

$$\|\mathbf{G}^{(l)}\mathbf{v} + \mathbf{g}^{(l)}\|_2 \leq m_l, \qquad l = 1, \ldots, 4NE + 2\sqrt{NE}, \tag{4.14}$$

where $\mathbf{G}^{(l)}$ picks the relevant variables v_j and v_i from $\mathbf{v}$ and

$$\mathbf{g}^{(l)} = \begin{bmatrix} \mathrm{d}x \\ u_j - u_i \end{bmatrix}. \tag{4.15}$$

The change in strain terms in (4.12) can be written as linear functions of the variables $\mathbf{m} \in \{m_1, \ldots, m_{4NE+2\sqrt{NE}}\}$. For instance, for a particular element with index value $k = 1$, the strains are computed by

$$\boldsymbol{\epsilon}^{(k)} = \begin{bmatrix} \Delta\epsilon_x^{(1)} \\ \Delta\epsilon_y^{(1)} \\ 2\Delta\epsilon_{xy}^{(1)} \end{bmatrix} = \begin{bmatrix} \frac{1}{2L_t^{12}} & \frac{1}{2L_t^{34}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2L_t^{14}} & \frac{1}{2L_t^{23}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2L_t^{13}} & \frac{1}{2L_t^{24}} \end{bmatrix} \begin{bmatrix} m_1 \\ \vdots \\ m_6 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \tag{4.16}$$

where L_t^{ij} is known and is constant for the current time step. Overall,

$$\boldsymbol{\epsilon} = \mathbf{A}\mathbf{m} - \mathbf{b}, \tag{4.17}$$

and the complete optimisation problem is

$$\begin{aligned} \min_{\mathbf{v}} \quad & \sigma_y \frac{2}{3} \sum_{k=1}^{NE} \Omega_e^k y_k, \\ \text{s.t.} \quad & \boldsymbol{\epsilon} = \mathbf{A}\mathbf{m} - \mathbf{b}, \\ & \|\boldsymbol{\epsilon}^{(k)}\|_2 \leq y_k, \qquad k = 1, \ldots, NE, \\ & \|[\mathbf{G}^{(l)}\mathbf{v} + \mathbf{g}^{(l)}]\|_2 \leq m_l, \qquad l = 1, \ldots, 4NE + 2\sqrt{NE}, \\ & \mathbf{v}_{tool} \geq \mathbf{t}, \\ & \mathbf{v}_{boundary} \geq 0, \end{aligned} \tag{4.18}$$

where two sets of second-order cone constraints are defined and $\mathbf{t}$ describes the tool surface as a function of the tool position. The variables $\mathbf{v}_{boundary} \subset \mathbf{v}$ are in contact with the

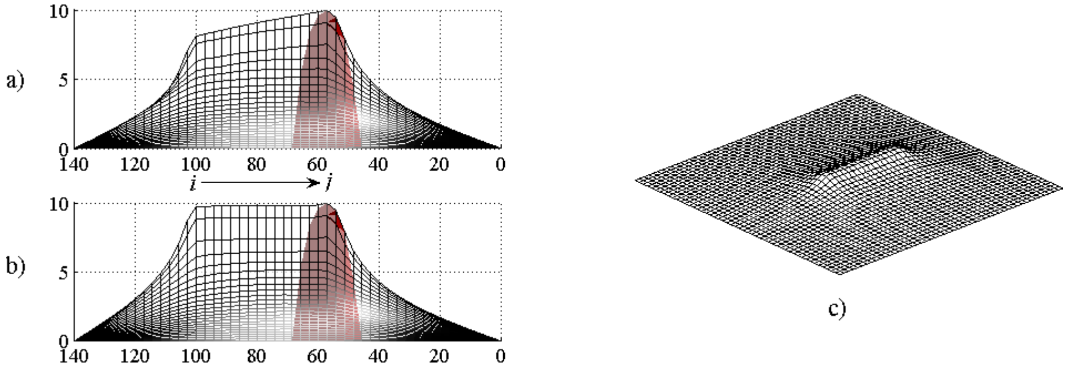


Figure 4.2: An example of shortening from the initial indentation depth of 10 mm as the tool tracked along from i to j (plot a)) and the same test when the penalty function was applied with $\rho = 0.01$ (in b)), which does not display shortening. The units used in the axes are mm.

clamping frame and are set to zero. Since there is no bending component here, it is not neccessary to enforce a zero gradient condition at the frame. Note that the strain variables in one set of second-order cone (SOC) constraints are a function of the variables $\mathbf{m}$, which are defined by the second set of SOC constraints. The inequality terms in both sets of cones must become equalities to give the correct solution, but the equality is only enforced for the SOC constraints associated with the variables y_k above. For m_l, which should be equal to the length of the rod numbered l, if the equality does not hold, the length of the rod will be less than the value of m_l and therefore shorter than it ought to be for the minimum energy cost. This would, and indeed does, manifest itself during simulation through a shortening of certain bars as the tool progresses over the grid network.

For example, in figure 4.2 a), the tool was tracked along as a constant height of 10 mm from point i to point j, yet a shortening of the bars near point i occurs resulting in a reduction in indentation depth as the tool progresses. One may think that this can be corrected by simply constraining the sheet velocities to be positive only but firstly, this would not be physically correct and secondly, the modification would only work for the current ISF configuration. For instance, the simulation of a two point forming process whereby two tools work both above and below the sheet would not be possible. A better approach is to try to enforce the equality in the cone constraints and this has been done by introducing a penalty term in the objective function in (4.18), which becomes

$$\sigma_y \frac{2}{3} \sum_{k=1}^{NE} \Omega_e^k y_k + \rho \sum_{l=1}^{NN} n_l, \tag{4.19}$$

in which the variables n_l are associated with an extra set of cone constraints, $n_l \geq \|v_l\|_2$, where $v_l \in \mathbf{v}$ and $l = 1, \ldots, NN$. Minimising n_l is equivalent to minimising the magnitude of the velocity at a node l, so the new cost function is a trade-off between the original energy minimisation and an inertia-like term that essentially resists sheet velocity, although the constant ρ has a small value typically in region of $\rho = 0.01$. Since ρ is small, the inertia term has a negligible effect on areas of the sheet that undergo significant straining and therefore have a large work component. However, for regions where the movement of nodes is purely due to the equality not holding in the conic constraints in (4.18), there is no strain energy term and only the second term in the objective function in (4.19) becomes significant, countering the non-work related movement of the relevant nodes. Figure 4.2(b) displays the result of using this scheme in the line tracking simulation, where the spurious shortening observed in figure 4.2(a) has been eliminated. Although it is not attempted here, it is noted that instead of using the inertia term it may also be possible to achieve a similar result by placing a penalty on **m** instead.

Examining the deformed regions in figures 4.2 a) and b) that have not yet been formed by the tool, such as the region from 0-40 mm, there is no difference between the plots, providing confidence that adding the extra term in (4.19) has had a negligible effect. Of course, if the value of ρ were to be increased this would not be the case, so care must be taken in choosing its value. Figure 4.3 shows this by displaying how the deformation field varies with ρ for a single indentation of the tool at the centre of the sheet. This highlights a further benefit in using ρ because its value can be tuned against experimental data to fit a particular material, as will be demonstrated later.

4.3 Incorporation of bending work

For completeness, it is important to briefly consider how one can incorporate bending work into the grid modelling scheme. One possible way of doing this would be to superimpose

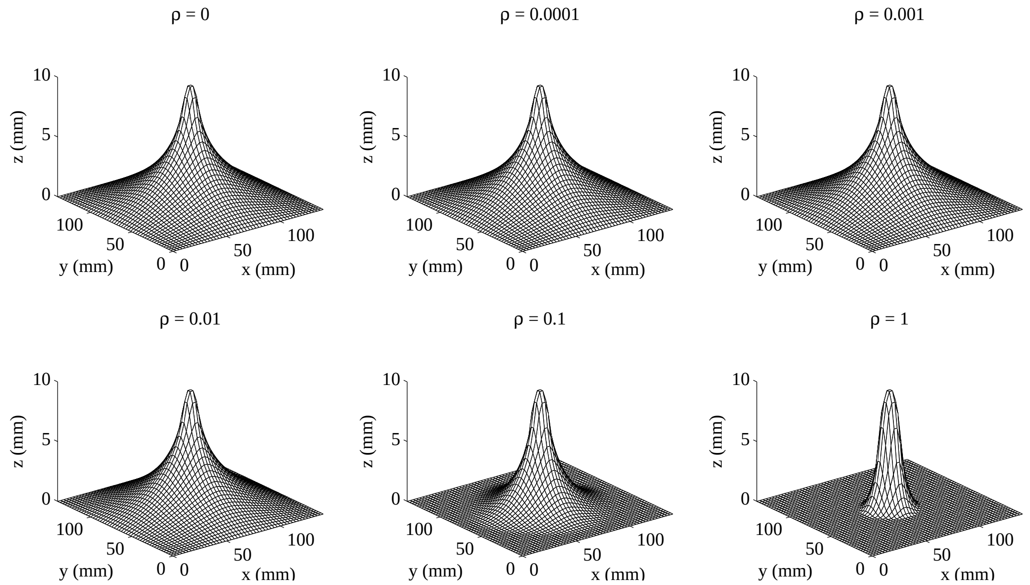


Figure 4.3: If ρ is too large it has a significant and undesireable effect on the deformation field by preventing the internal energy from being correctly minimised. The effect of varying ρ for the indentation of a single point by a spherical tool is shown. The solution with $\rho = 0$ shows the desired deformation field.

the plate bending model from §3.2 over the square element in figure 4.1. The nodes of the square element can also be used to define two triangular finite elements (1,2,3) and (1,3,4) over which the plate bending work component can be computed using exactly the same finite element formulation discussed in §3.2. The bending contribution can then be added to the stretching work due to the deformation of bars over the nodes (1,2,3,4) as described by the grid model, resulting in a hybrid FE-grid model. If $\dot{W}$ describes the total rate of energy dissipation for an FE-grid element e, then

$$\dot{W}^{(e)} = \chi_s \dot{W}_s^{(e)} + \chi_b \sum_{i=1}^{2} \dot{W}_b^{(e_i)}, \tag{4.20}$$

where χ_s and χ_b are weighting factors, e_1 and e_2 are the triangular subsections of e and the subscripts s and b refer to the stretching and bending components respectively. The first query arising from (4.20) is to ask how $\dot{W}_b$ is modified if e_i, where $i = \{1, 2\}$, is not flat. This is answered by considering a horizontal element and a vertical element in a velocity field constrained to the vertical direction. The bending work is assumed to be related to the sheet velocities that are perpendicular to the plane of the sheet. For a horizontal sheet the work contribution due to bending is at a maximum whereas in the case of a vertical sheet under vertical loading, all the deformation is due to stretching. A modification to

(4.20) to describe this behaviour naturally follows as

$$\dot{W}^{(e)} = \chi_s \dot{W}_s^{(e)} + \chi_b \sum_{i=1}^{2} \dot{W}_b^{(e_i)} \cos\theta_i, \tag{4.21}$$

where θ_i is the angle between the plane of element e_i and the horizontal. Hence for a flat horizontal sheet when $\theta_i = 0$ and $\cos\theta_i = 1$ the bending component is at its maximum and when the sheet is vertical, $\theta_i = 0.5\pi$, so $\cos\theta_i = 0$ and the bending work component is removed.

A second concern also arises from (4.20) in how one might determine the value of the factors χ_s and χ_b. In the rigid plastic sequential limit analysis model, this problem was avoided by using the Ilyushin condition, which provided a natural method for combining the bending and membrane components of work in the model. The approximate formulation of the model provided here is less rigorous than the sequential limit analysis model and it has no framework in which to easily incorporate a similar yield condition. The best approach here may be to use a set of systematic trials against experimentation to determine suitable values for the weighting factors. This could be a significant undertaking and is not considered worthwhile unless it is absolutely necessary to compute bending work in the gridwork model. In this thesis, the results obtained using the gridwork model were considered to be suitable for an approximate model for ISF, so the incorporation of bending work is not considered sufficiently important to explore further here.

4.4 Results

The optimisation problem in (4.18) was solved using the Mosek toolbox with $NE = 4900$, which corresponds to a 70×70 square grid, using the twisted cone tool-path shown in figure 1.15 and the resulting deformation field is displayed in figure 4.4. The set-up time was significantly faster than for the finite element (FE) model due to the reduced number of variables, corresponding to significantly reduced matrix sizes. The dual SOCP problem was briefly investigated but unlike for the rigid plastic limit anlysis FE model, it appeared to give no benefit regarding the efficiency of the solution and so it was not explored in further detail. The model was also solved for other values of NE using the same tool-path and

Mesh size	**Error measures**		
No. elements, NE	**RMS** (mm)	**Var.** (mm^2)	**Max. diff.** (mm)
841	0.9237	0.8055	3.5912
1521	0.9132	0.7731	3.5935
2401	0.9101	0.7655	3.7454
3481	0.8986	0.7439	3.6586
4761	0.9064	0.7506	3.6465
6241	0.9076	0.7526	3.6561
7921	0.9031	0.7430	3.6675

Table 4.1: Error measures displayed against mesh density.

the resulting RMS vertical differences, variances and maximum deflections are displayed in table 4.1 for different mesh sizes. These values are also plotted in figure 4.5 and the corresponding solution times are shown in figure 4.6.

The images in figure 4.7 were taken at different stages of the forming process and they show that this model is better at simulating unformed regions when compared to the FE model, since the dip that was observed in the FE model is not present here. The reason why this ‘dip’ does not occur here remains unclear, but it is likely to be a consequence of shortening in some of bars despite the presence of the penalty function in the objective function. The lack of ‘dip’ allows this model to be used to simulate the product shown in figure 1.9, for which the tool-path does not cover the whole region of the sheet leaving an unformed square region in the centre of the sheet. Hence, a further simulation was produced using the tool path in figure 1.10 and the resulting deformation field is shown in figure 4.8 along with a comparison between the Abaqus model and the grid method model in figure 4.9. The greatest differences between the two models are along the steepest edge of the product, where the maximum vertical difference is 5.08 mm. The fit over the rest of the product is much better, with an overall RMS error of 1.59 mm and variance of 0.55 mm^2.

From the fit between the simulations in figures 4.4 and 4.9, one can conclude that the more significant contribution to deformation in ISF originates from stretching work rather than bending work, which has been neglected here. This is related to the principle of conservation of volume and is not unexpected, considering that the material is tightly clamped at the edges so that any deformation must a involve a large degree of stretching. As an aside on the subject of stretching, a forming process that undergoes stretching can

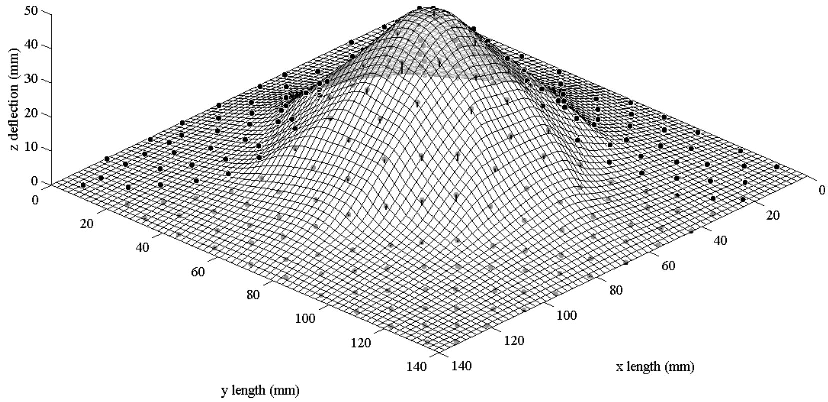


Figure 4.4: Simulation of test product in figure 1.14 produced by solving the rigid plastic shell problem using the grid model for a regular square mesh. The superimposed black dots are measured points from the real product.

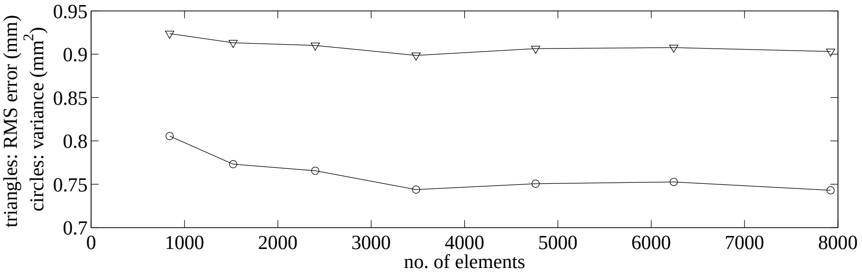


Figure 4.5: Plot of RMS errors (triangles) and variances (cirlces) between the grid model and CMM measured data for the product in figure 1.14. The data for these plots is given in table 4.1.

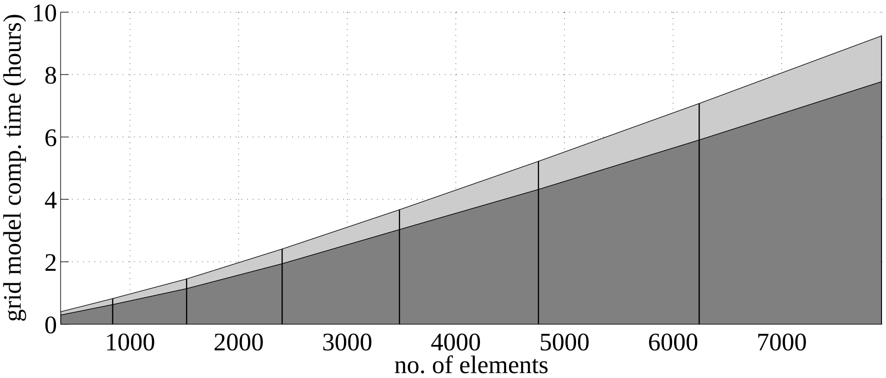


Figure 4.6: Plot of the computation times related to figure 4.5. The total time is broken into solver time (dark grey) and set-up time (light grey).

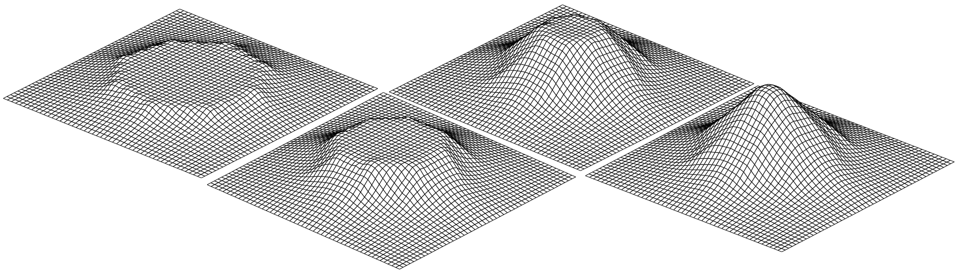


Figure 4.7: Simulation of test product in figure 1.14 at different stages of the process. Left to right: 25%, 50%, 75% and 100% through the process.

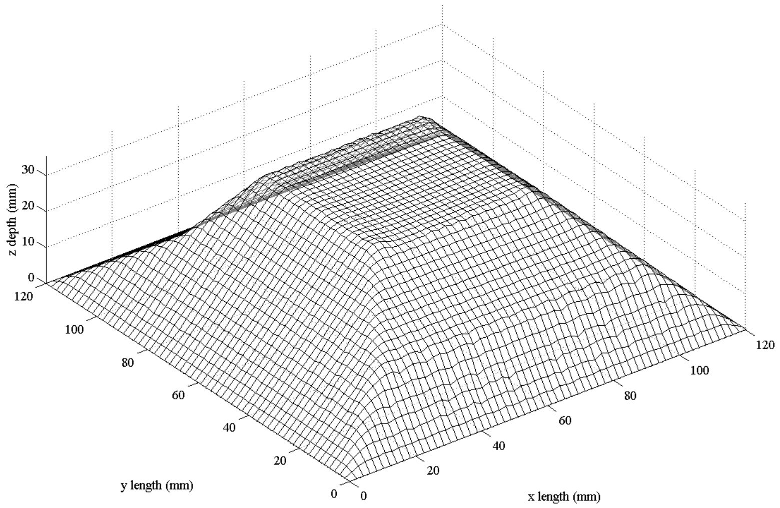


Figure 4.8: Simulation of test product in figure 1.9 produced by solving the rigid plastic shell problem using the grid model for a regular square mesh.

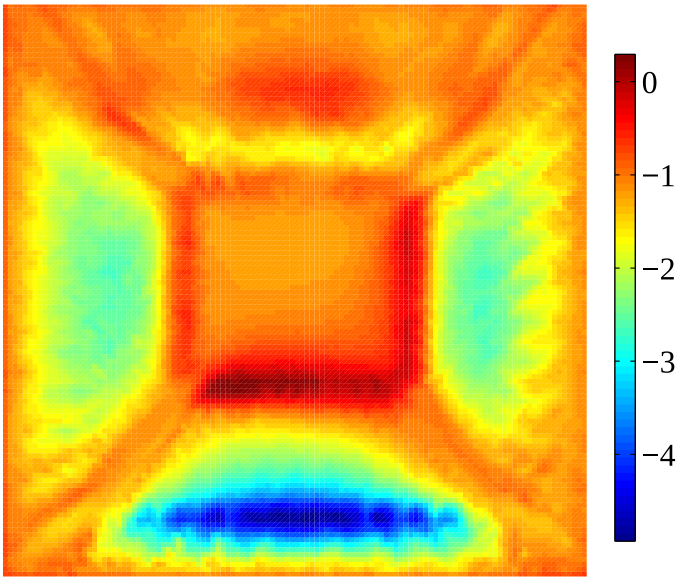


Figure 4.9: Difference in mm between the simulation in figure 4.8 and the Abaqus model in figure 1.9. The origin when compared to figure 4.8 is at the bottom left corner. The greatest discrepancies between the two simulations are observed on one face of the product.

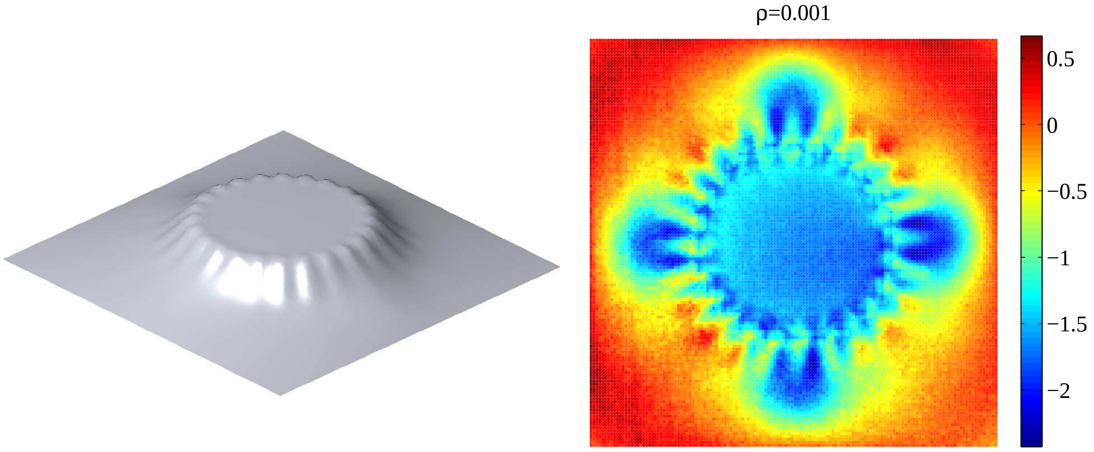
be prone to excessive thinning which can then lead to tearing; the forming limit is defined by the amount of deformation that can take place before this occurs. To prevent tearing and therefore to improve the forming limit, processes such as deep drawing use draw beads within the blanks which allow some material to be drawn in during the forming process [116]. If excessive material flow occurs wrinkling can take place which sets a limit to the amount of draw permitted. Such methods have not yet been investigated in detail for ISF, but it has been observed that even under tight clamping ISF can lead to greater forming limits than deep drawing [109, 8]. This is attributed to the incremental deformation mechanism in ISF, which results in a more uniform distribution of sheet thickness when compared to drawing, as well as an increased level of through thickness shear forces.

4.4.1 Effect of varying the parameter ρ

Another product that can be simulated more effectively using the grid model is the one in figure 3.37, so this was done using the same tool-path that was used to produce the original part with NE=4900. This time the model was run with three different values of ρ in order to see what the effect this had on the deformation field. An image of the simulated shape for $\rho = 0.01$ is shown in figure 4.10(a) and it looks similar for the other two simulations. The difference between the model and the actual part, along with error measures, are given in figures 4.10(b), 4.10(c) and 4.10(d), showing that if the RMS error is the error measure of interest, then a value of $\rho = 0.1$ is most appropriate, whereas if variance is considered to be of greater importance then it is better to use $\rho = 0.01$. This procedure of testing ρ is simple and it can be repeated for any new materials that are used on the ISF machine in order to tune ρ such that it improves the effectiveness of the model. The error measures obtained with the grid model are repeated in table 4.2 along with those from the FE model so that they can be easily compared.

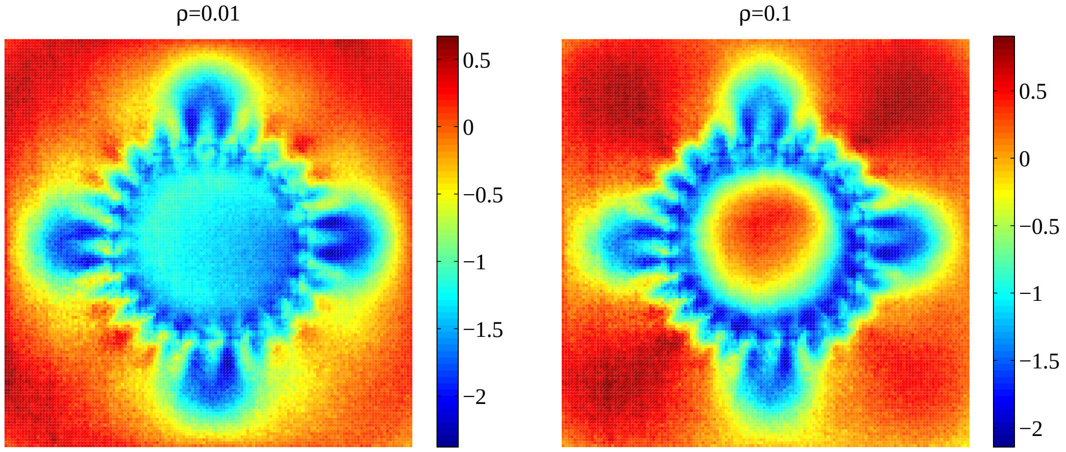
4.4.2 Two point incremental forming simulations

Since the grid model does not display a ‘dip’, it gives a more convincing deformation field when compared to the finite element model in chapter 3 at intermediate stages of the process, as well as with certain shapes like the one in §4.4.1. As such, it can be used to model



(a) Simulation of product in figure 3.37 using $\rho = .001$.

(b) RMS=0.945 mm, var.= 0.495 mm^2, max. diff.=2.427 mm.



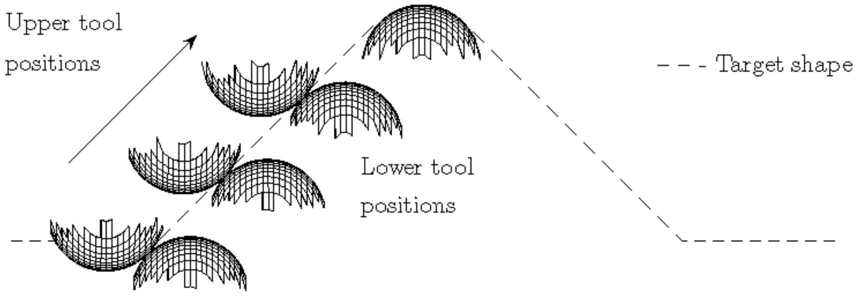
(c) RMS=0.892 mm, var.= 0.471 mm^2, max. diff.=2.378 mm.

(d) RMS=0.740 mm, var.= 0.517 mm^2, max. diff.=2.142 mm.

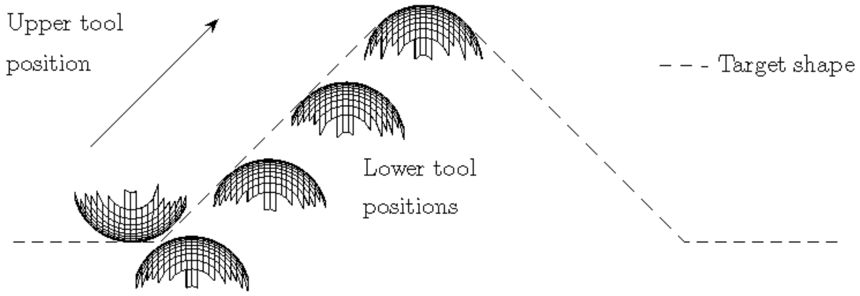
Figure 4.10: Difference between the actual product and gridwork model in a) for $\rho = 0.001$, $\rho = 0.01$ and $\rho = 0.1$.

	Mesh size	**Error measures**		
Model used	*NE*	**RMS** (mm)	**Var.** (mm^2)	**Max. diff.** (mm)
one cone (O)	4802	1.510	2.248	6.278
one cone (S)	4802	1.630	2.632	6.748
one cone (S & TC)	4802	1.074	1.103	3.296
one cone (S, TC & WH)	4802	1.350	1.099	2.806
grid, $\rho = 0.001$	4900	0.945	0.495	2.427
grid, $\rho = 0.01$	4900	0.892	0.471	2.378
grid, $\rho = 0.1$	4900	0.740	0.517	2.142

Table 4.2: Error measures obtained when simulating the product in figure 3.37 using the grid model and one cone model (from §3.9, §3.10 and §3.11.3). O=original, S=sliding contact state, TC=thickness variation and WH = linear work hardening.



(a) The upper tool follows lower one while maintaining a contact angle of 45^o. The resulting deformation field is shown in figure 4.12.



(b) The upper tool continually follows the base contour of the target product whilst the lower tool follows same path as a). The resulting deformation field is shown in figure 4.13.

Figure 4.11: Two different tool paths for TPIF for the target shape in figure 1.11 shown at four stages of the process.

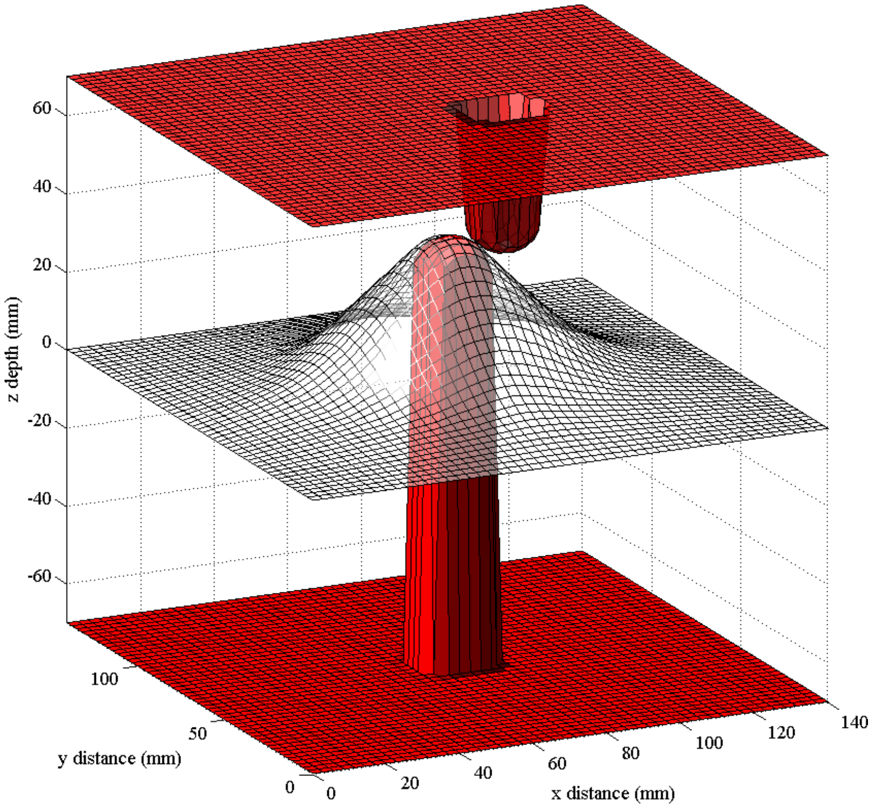


Figure 4.12: Simulated deformation field resulting from the tool path in figure 4.11 a) for a 50×50 grid.

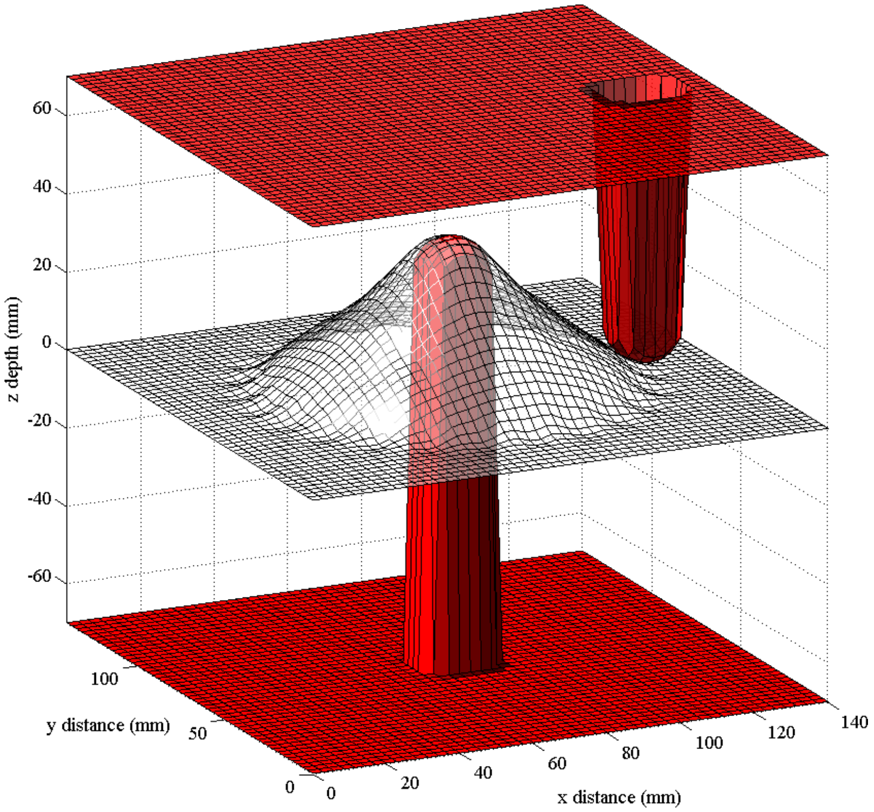
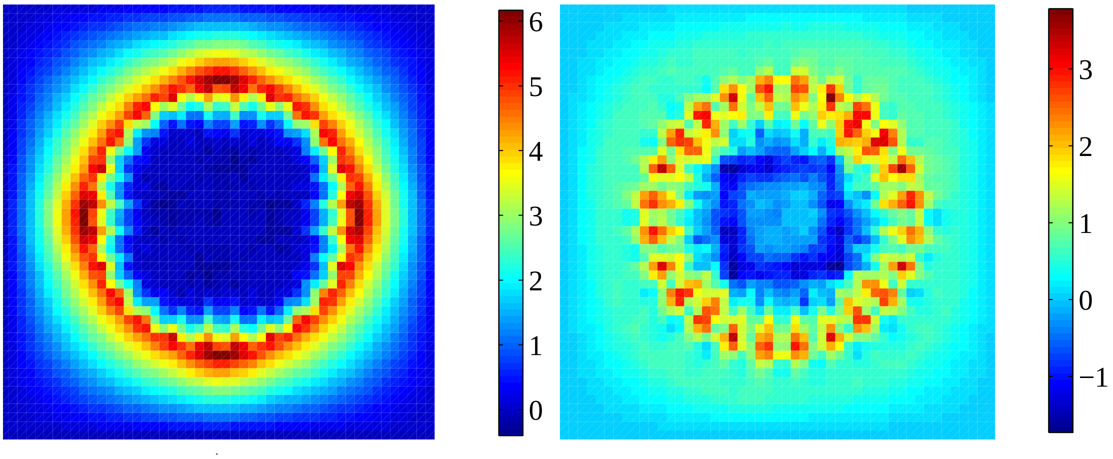


Figure 4.13: Simulated deformation field resulting from the tool path in figure 4.11 b) for a 50×50 grid.

alternative configurations of the ISF process rather than just the single point incremental forming (SPIF) process used so far. One possibility is to model a two point incremental forming (TPIF) process that consists of two identical tools that work on opposite sides of the sheet. The main advantage of this configuration is that it allows actuation in two directions, but this configuration complicates the path planning problem because two optimised tool-paths have to be constructed. TPIF was not adopted on the Cambridge ISF machine because the space that the second tool would occupy was needed for a stereo vision camera. However, during a visit to Ford Motor Company's Research and Advanced Engineering facilities in Dearborn, Detroit, the early development of a TPIF process was observed and it was suggested at that time that an effective method for modelling TPIF would be useful. For this reason, an example of the application of the grid model in modelling a TPIF process is now provided.

The target shape chosen for this simulation was the benchmark cone shaped product displayed in figure 1.11. Since it is not known how to design the two tool-paths for the part, the heuristic of following the contour of the part was used for the lower tool. Two different paths were used for the upper tool, firstly where it followed the lower tool whilst maintaining an angle of 45^o at point of closest contact between the two tools and secondly where it continually circled the base of the part to restrain any movement in that region. These tool-paths are displayed in figure 4.11 a) and 4.11 b) respectively. It can be seen from 4.11 a) that the upper tool indented the sheet in the reverse direction for the first few layers of the process. This has little effect on the final deformation field, which is very similar to the one that would be obtained if only the lower tool was used. Figure 4.12 displays the simulated deformation field resulting from this tool-path and figure 4.14 a) displays the vertical difference between the simulation and the target shape, from which an RMS difference of 2.30 mm, a variance of 2.65 mm^2 and a maximum error of 6.17 mm^2 were calculated.

The aim of the tool-path figure 4.11 b) is to try to reduce the maximum error that occurs in the flanging around base region of the product when using the tool-path in figure 4.11 a). The aim of upper new tool-path is to continually push back any deformation that occurs in the based region of the product while the lower tool-path produces the shape. The resulting simulated deformation is displayed in figure 4.13 and the difference between this and the target shape is displayed in figure 4.14 b). This example shows a clear improvement compared to the first tool-path, with an RMS error of 0.89 mm, a variance of 0.55 mm^2 and a maximum deflection of 3.78 mm. Overall there were 975 tool position increments in the discretised tool-path and the model took 4.42 hours, which is an average of about 16 seconds solution time for each step. Large position increments were used for the upper tool because the velocity of the upper tool was increased so that it completed each circuit in the same time as the lower tool despite covering a greater distance. However, these large steps appear to have caused dents in the upper surface of the sheet, which were removed by reducing the speed of the tool as shown in figure 4.15, by using twice the number of position increments. To the eye, the deformation in this more accurate simulation appears be a much better fit to the target shape in figure 1.11, with the only discrepency being a slight bulge along the cone faces closest to the corners. From figure 4.16, it can be seen that the greatest errors are observed on these four faces and the resulting error measures are: an RMS difference of 0.96 mm, a variance of 0.62 mm^2 and a maximum deflection of



(a) (b)

Figure 4.14: Difference in mm between a) figure 4.12 and the target shape in figure 1.11, b) figure 4.13 and figure 1.11.

4.08 mm.

4.4.3 Simulation of a hood part

The products simulated so far have all been rather simple shapes that are only used in academic research, so a brief demonstration is provided here of the application of the grid-work model in simulating a more realistic hood part for a car. The tool-path was a contour following tool-path provided by Ford R&AE that used a combination of TPIF with two moving tools and SPIF. Figure 4.17 shows the target shape and the deformation field predicted by the grid model and the difference between the two is displayed in figure 4.18. No measurement data is available from the actual process, so no further analysis of the part is performed, except for the qualitative observation that realistic parts can be simulated by the gridwork model.

4.4.4 Summary

In this section, some results that were obtained using the grid model described in this chapter were presented, from which at least three conclusions can be drawn. Firstly, the current model appears to give reasonable results when simulating the SPIF process used on the Cambridge ISF machine and in particular, it fits the experimental data better than the FE limit analysis model for tool-paths in which a ‘dip’ is observed in the latter model.

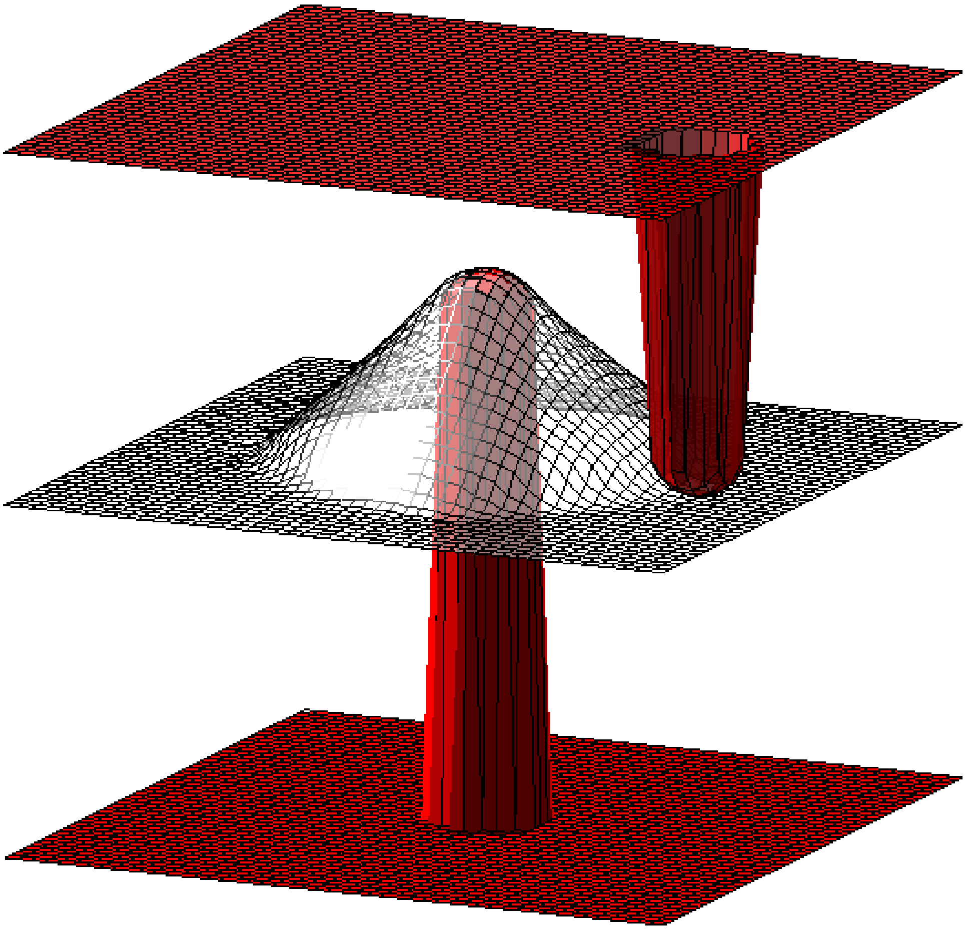


Figure 4.15: Simulated deformation field resulting from the tool path in figure 4.11 b) for a 50×50 grid using smaller time steps compared to figure 4.13.

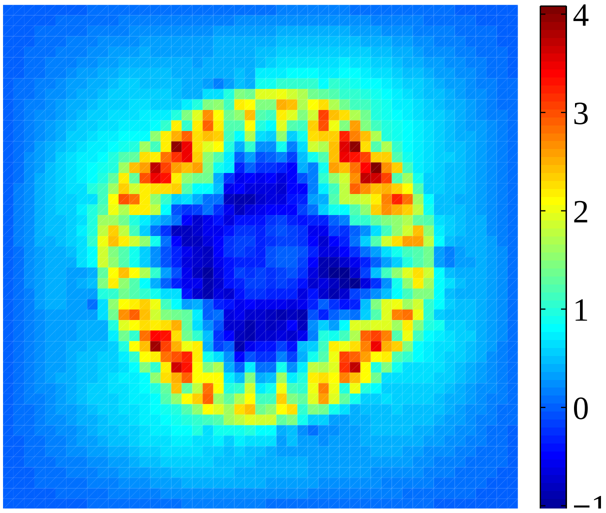
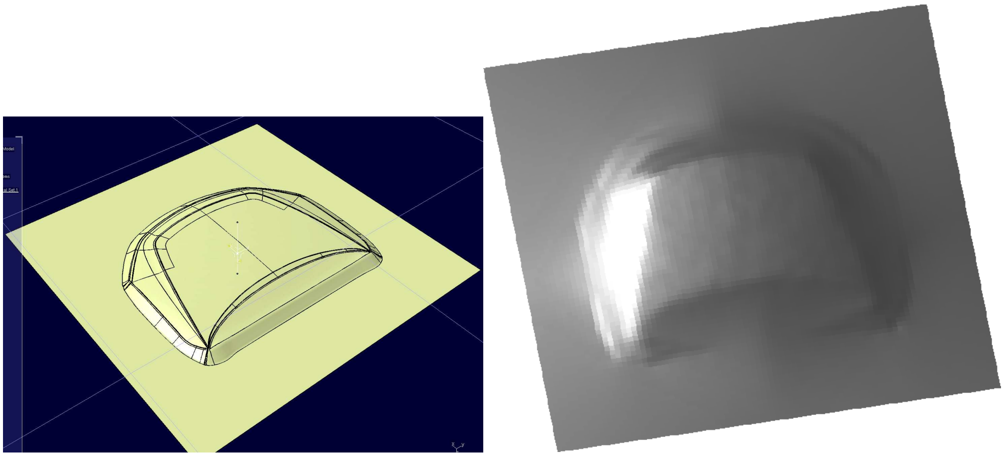


Figure 4.16: Difference in mm between figure 4.15 and figure 1.11 using shorter time steps than figure 4.14 b).



(a) traget shape

(b) simulation

Figure 4.17: Simulation of hood part using gridwork model.

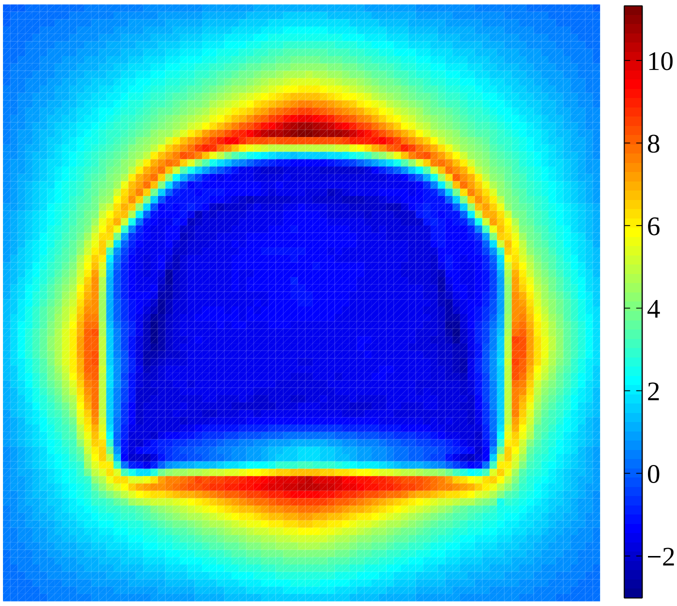


Figure 4.18: Difference between simulation and target shape.

A simulation of the two-point incremental forming (TPIF) process was also successfully performed and the resulting deformation field appears sensible. This leads to the second conclusion: the extra actuation offered by the second tool allows an intuitive tool-path to be designed that produces, in simulation at least, a far closer match to the target product when compared to SPIF. This is because the second tool can be used to resist deformation in the regions of the sheet where the greatest errors are observed in the SPIF process, which is around the base region of the product. This evidence should be used to open up a debate about the requirements of a control system for ISF; for instance, whether a second tool is necessary to achieve suitable controllability in the process, or more generally whether more work should be done in optimising the mechanical design of the ISF process, so as to improve the predictability of the process and to ease the task of the controller. The third and final conclusion is that, although the second tool or some other improvement to the process is likely to remove much of the burden on the control system, some errors will inevitably remain between target product and the finished product, so there is still a need for a controller that removes these errors by optimising the tool-path(s).

Chapter 5

Tool-path optimisation

The majority of this thesis has been concerned with modelling ISF with the intention of producing a fast approximate model to use for tool-path optimisation. The models produced have been found to predict the deformation behaviour adequately and they are faster to solve than the available commercial FEA packages. The gridwork model in particular, being the fastest, also holds some potential in becoming used for the purpose of path optimisation in the future.

However, none of the models are currently fast enough to apply in the optimal control problem, especially without access to powerful processors. To give an idea of the computation time required for the model, one may estimate that the path-optimisation routine would require at least one hundred iterations, or one hundred runs of the model. If one day is permitted for computing an optimal tool-path, the model would have to simulate the complete process in approximately six minutes, which allows 0.3 seconds per time-step. Hence, an order of magnitude improvement in computation time is required over what has been reported here, but it is noted that the results provided here have been obtained on a desktop computer and with the current rate of increase in computational speeds it is reasonably likely that even with desktop PCs, the limitation on computation speed will not remain for long. Another one possibility that has not been explored here is to exploit the parallel processing options available in Mosek for solving the SOCP problem in the model. In [13], the developers of Mosek described how the interior point algorithm was extended to exploit parallel computing. Although this paper is now slightly dated, it reports significant improvements in computation time for a number of test cases, which were solved on a parallel cluster of just four inexpensive PCs.

If successful, the approach of using a model based path optimisation scheme to reduce geometric errors would be a significant contribution to the ISF community. It is therefore important to provide some discussion about how this would be performed once suitable computing resources for the models become available.

The optimal control problem is outlined in this chapter and a numerical method that could be used for solving it is discussed. Some optimal tool paths are then generated by introducing a novel impulse response based method for predicting the process behaviour because numerical modelling is not currently feasible for this purpose. Experimental results are provided here that show that this impulse response based method is highly effective in the optimisation of the tool-path for axisymmetric parts, opening up the possibility of extending the method for asymmetric parts as future research. The approach is very promising and with further development, it may also by-pass the need for numerical process modelling in the path optimisation problem.

5.1 Formulation of optimal path problem for ISF

The path optimisation problem is treated as a multi-stage discrete time process control problem [25] by defining each 'stage' as the application of the tool-path at a constant depth, or layer, before proceeding to the next stage, or layer, of the ISF process. A set of control variables are subsequently chosen from the set of parameters that define the tool-path at each stage.

For instance, considering an axisymmetric product that is produced in M layers, at each layer the tool moves along a circular tool-path of radius r_i at a height $\Delta z_i = z_i - z_{i-1}$ above the previous layer, where $\{i = 1, \ldots, M\}$, as displayed in figure 5.1. Here, Δz_i are kept constant and the control inputs at each stage are r_i and the control variables over all the stages are $\mathbf{p} = [r_1, r_2, \ldots, r_M]^T$. These inputs are chosen to satisfy a multi-stage

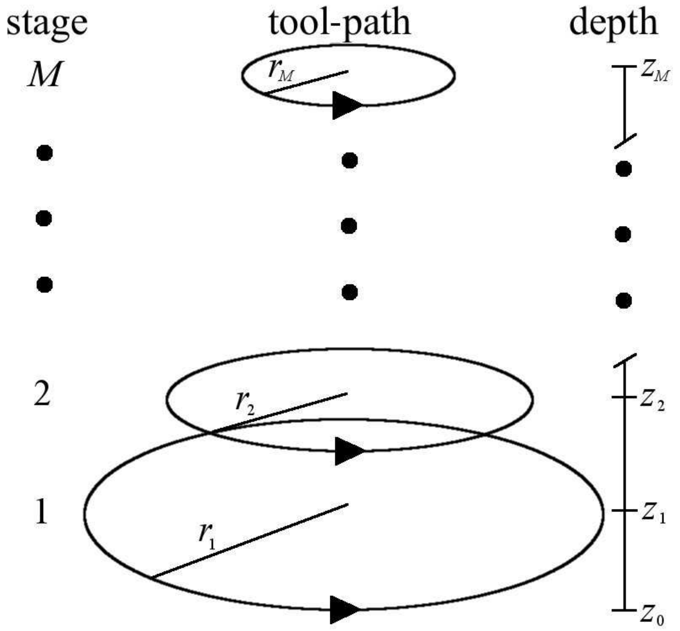


Figure 5.1: Typical tool-path for an axisymmetric product. Each layer of the tool-path is assigned a stage number $i = 1, \ldots, M$. The tool-path at layer i is characterised by the forming depth, z_i and the radius, r_i, of the circular path that the tool follows at that layer.

optimal control problem of the form,

$$\begin{aligned} &\min_{\mathbf{p}} \; J = \Phi(\mathbf{x}_M) + \rho \sum_{i=1}^{M} L_i(\mathbf{x}_i, r_i) \\ &\text{s.t. } \mathbf{x}_{i+1} = \mathbf{f}_i(\mathbf{x}_i, r_{i+1}) = \mathbf{x}_i + \mathbf{w}_{i+1}(\mathbf{x}_i, r_{i+1}), \quad i = 0, \ldots, M-1, \end{aligned} \tag{5.1}$$

where $\mathbf{x}_i$ is the deformation field and $\mathbf{w}_i$ is the output of a model of the process with a known initial state $\mathbf{x}_0 = \mathbf{c}_0$. Generally, the terms in $\mathbf{c}_0$ are the initial (x, y) locations of nodes over a flat sheet with the z coordinates approximately equal to zero. For any of the ISF models described in this thesis, $\mathbf{w}_i$ would be the change in deformation in the simulation from stage $i-1$ to stage i of the forming process.

The first term in the cost function in (5.1) is defined as

$$\Phi(\mathbf{x}_M) = \|\mathbf{x}_M - \hat{\mathbf{x}}\|_2, \tag{5.2}$$

which is a measure of the error between the final deformation field of the system $\mathbf{x}_M$ and the target shape $\hat{\mathbf{x}}$. A trade-off in the control objective is posed by the second term, $L_i(\mathbf{x}_i, r_i)$, which places a penalty over the whole period of the process and is scaled by a factor, ρ, that is defined by the designer. This can be chosen to penalise the length of the tool-path by, for instance, minimising the actual length of the tool-path, in which case $L(r_i) = \|[\Delta z_i \;\; \Delta r_i]^T\|_2 + 2\pi r_i$. In practice, a jittery tool-path can cause more problems than a long tool path by causing the sheet to tear, so here the performance index

$$L(r_i) = \|[\Delta z_i \;\; \Delta r_i]^T\|_2 \tag{5.3}$$

is chosen to smooth the tool-path by preventing excessive variations in the input between two adjacent layers.

The augmented performance index for (5.1) is

$$\hat{J} = \Phi(\mathbf{x}_M) + \sum_{i=0}^{M-1} \left[(\rho L_{i+1} + \boldsymbol{\lambda}_{i+1}^T \mathbf{f}_i) - \boldsymbol{\lambda}_{i+1}^T \mathbf{x}_{i+1} \right], \tag{5.4}$$

from which the discrete Hamiltonian is

$$H_i = \rho L_{i+1} + \boldsymbol{\lambda}_{i+1}^T \mathbf{f}_i, \quad i = 0, \ldots, M-1 \tag{5.5}$$

and by changing the summation in (5.4),

$$\hat{J} = \Phi(\mathbf{x}_M) - \boldsymbol{\lambda}_M^T \mathbf{x}_M + H_0 + \sum_{i=1}^{M-1} \left[H_i - \boldsymbol{\lambda}_i^T \mathbf{x}_i \right]. \tag{5.6}$$

Similarly to the continuous time case, the variation of $\hat{J}$,

$$\mathrm{d}\hat{J} = \left[\frac{\partial \Phi(\mathbf{x}_M)}{\partial \mathbf{x}_M} - \boldsymbol{\lambda}_M^T \right] \mathrm{d}\mathbf{x}_M + \frac{\partial H_0}{\partial \mathbf{x}_0} \mathrm{d}\mathbf{x}_0 + \frac{\partial H_0}{\partial r_1} \mathrm{d}r_1 + \sum_{i=1}^{M-1} \left\{ \left[\frac{\partial H_i}{\partial \mathbf{x}_i} - \boldsymbol{\lambda}_i^T \right] \mathrm{d}\mathbf{x}_i + \frac{\partial H_i}{\partial r_{i+1}} \mathrm{d}r_{i+1} \right\}, \tag{5.7}$$

is set to zero when the cost is minimised. Defining the value of the Lagrange multipliers $\boldsymbol{\lambda}_i$ so that

$$\boldsymbol{\lambda}_i = \frac{\partial H_i}{\partial \mathbf{x}_i} = \rho \frac{\partial L_{i+1}}{\partial \mathbf{x}_i} + \boldsymbol{\lambda}_{i+1}^T \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i}, \quad i = 0, \ldots, M-1, \tag{5.8}$$

with the boundary condition

$$\boldsymbol{\lambda}_M = \frac{\partial \Phi(\mathbf{x}_M)}{\partial \mathbf{x}_M}, \tag{5.9}$$

and substituting these values into (5.7), gives

$$\mathrm{d}\hat{J} = \boldsymbol{\lambda}_1^T \frac{\partial \mathbf{f}_0}{\partial \mathbf{x}_0} \mathrm{d}\mathbf{x}_0 + \sum_{i=0}^{M-1} \frac{\partial H_i}{\partial r_{i+1}} \mathrm{d}r_{i+1}. \tag{5.10}$$

Since $\mathbf{x}_0 = \mathbf{c}_0$ is constant, $\mathrm{d}\mathbf{x}_0 = 0$ and so at an extremum of $\mathrm{d}\hat{J}$ for arbitrary $\mathrm{d}r_{i+1}$, it is required that

$$\frac{\partial H_i}{\partial r_{i+1}} = \rho \frac{\partial L_{i+1}}{\partial r_{i+1}} + \boldsymbol{\lambda}_{i+1}^T \frac{\partial \mathbf{f}_i}{\partial r_{i+1}} = 0, \quad i = 0, \ldots, M-1, \tag{5.11}$$

given that (5.8), (5.9) and the equations for the dynamic model are satisfied. This leads to the optimisation problem,

$$\begin{aligned} \min_{\mathbf{p}} \ & H_i \\ \text{s.t.} \ & \mathbf{x}_{i+1} = \mathbf{f}_i = \mathbf{x}_i + \mathbf{w}_{i+1}(r_{i+1}), \\ & \boldsymbol{\lambda}_i = \rho \frac{\partial L_{i+1}}{\partial \mathbf{x}_i} + \boldsymbol{\lambda}_{i+1}^T \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i}, \\ & \mathbf{x}_0 = \mathbf{c}_0 \\ & \boldsymbol{\lambda}_M = \frac{\partial \Phi(\mathbf{x}_M)}{\partial \mathbf{x}_M} \end{aligned} \tag{5.12}$$

where $\mathbf{p} = [r_1, r_2, \ldots, r_M]^T$, $i = 0, \ldots, M-1$, and the boundary conditions $\mathbf{x}_0 = \mathbf{c}_0$ and $\boldsymbol{\lambda}_M = \frac{\partial \Phi(\mathbf{x}_M)}{\partial \mathbf{x}_M}$ are two point boundary conditions.

5.1.1 Indirect method

The indirect method is to solve (5.12) for the optimal control law. If the cost function L_i is chosen such that it is a function of the control variables $\mathbf{p}$ only, the problem becomes

$$\begin{aligned} \min_{\mathbf{p}} \ & H_i \\ \text{s.t.} \ & \mathbf{x}_{i+1} = \mathbf{f}_i = \mathbf{x}_i + \mathbf{w}_{i+1}(r_{i+1}), \\ & \boldsymbol{\lambda}_i = \boldsymbol{\lambda}_{i+1}^T \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i}, \\ & \mathbf{x}_0 = \mathbf{c}_0 \\ & \boldsymbol{\lambda}_M = \frac{\partial \Phi(\mathbf{x}_M)}{\partial \mathbf{x}_M} \end{aligned} \tag{5.13}$$

where

$$\frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i} = 1 + \frac{\partial \mathbf{w}_{i+1}}{\partial \mathbf{x}_i}. \tag{5.14}$$

Hence, the partial differential of $\mathbf{f}_i$ depends on how the rate of deformation $\mathbf{w}_i$ varies with respect to perturbations in the current deformation state of the system. Considering the large number of state variables, this would be very costly to compute unless some structure in the ISF model can be utilised.

It is not fully understood what form this may take, but one idea that may prove to work is to exploit the information carried by the Lagrange multipliers used in the energy minimisation problem in the ISF models presented previously. The Lagrange multipliers calculated from the dual optimisation problem give an idea of the sensitivity of the system [24], because their magnitude at the optimal point can indicate the degree of variation of the objective function at the optimal point with respect to perturbations in the constraints corresponding to the multiplier. It is thought that it may be possible to exploit this property of the Lagrange multipliers to find the partial derivative $\frac{\partial \mathbf{w}_i}{\partial \mathbf{x}_i}$ and subsequently $\frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i}$, but this has not been attempted here. If it is possible to compute the partial derivative in this way, it would be obtained as a by-product of the model that would have to be solved for $\mathbf{w}_i$ anyway, so the values for both constraints in (5.13) would be calculated in one step. It is suggested that the viability of this method should be established in a future research exercise, because if computation times become feasible, then it would be an appealing reason to choose the indirect method for solving the optimal control equations.

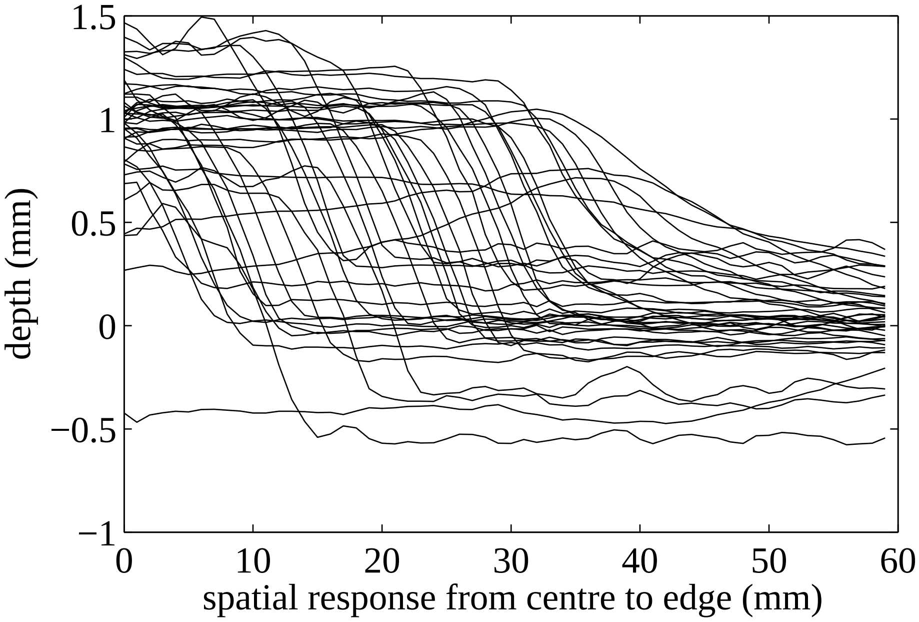
For the spatial impulse response method used in this chapter, it will be seen that the model consists of a summation of experimentally obtained impulse response shapes, for which the computation is very quick, so rather than using the indirect method it is straightforward to solve the optimal control problem via a direct method.

5.1.2 Direct method

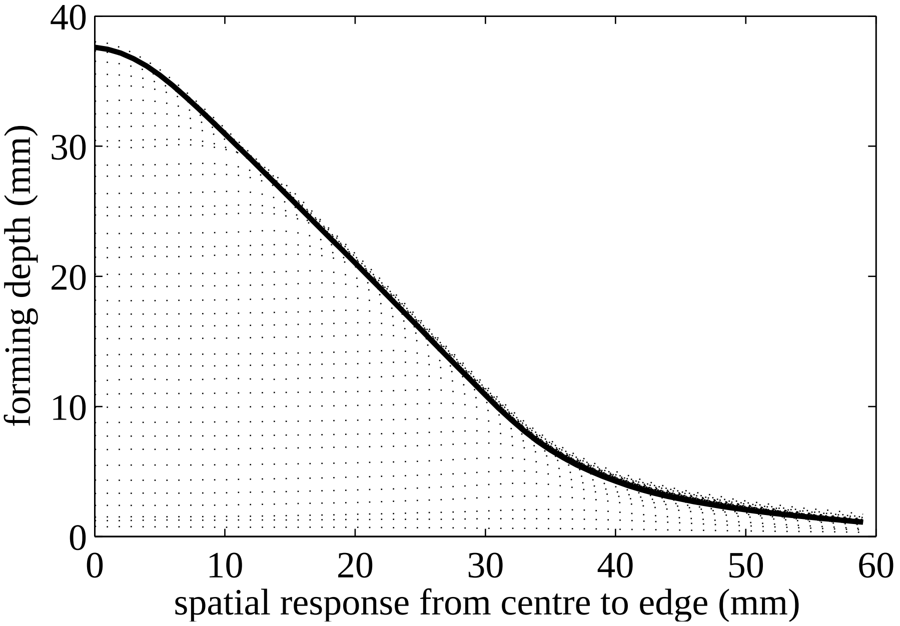
The direct approach is to solve (5.1) as it is stated, using a numerical technique such as the method of shooting [105, 22], which is used for two-point boundary value problems (TPBVP). A TPBVP requires the solution of a set of differential equations that satisfy constraints defined at two different times; usually at the initial and final states of the system. Using the method of shooting, any boundary conditions defined at the initial time t_o of the system are treated as optimisation constraints and a performance index is written for the constraints at time t_f, which must be minimised for them to be satisfied. In the case of the optimal control problem, the constraint at time t_o is an equality $\mathbf{x}_0 = \mathbf{c}_0$, which can be set explicitly by substituting in the known values $\mathbf{c}_0$ for $\mathbf{x}_0$, and the performance index J to be minimised is already defined in (5.1).

For each iteration in the minimisation routine, the system is integrated from t_o to t_f with the current guess for the control variables, $\mathbf{p}$, to predict the final deformation field, $\mathbf{x}_M$, which is then used to compute the value of the cost, J. This integration from t_o to t_f is equivalent to solving one of the numerical ISF models or summing up a set of spatial impulse response shapes, as will be seen in the following section. After the cost function is computed the control variables $\mathbf{p}$ are modified using, for instance, a gradient based optimisation algorithm to compute a new value of $\mathbf{p}$ that reduces the cost further. The procedure is repeated until a termination condition for the minimum cost is satisfied.

It should be noted that the cost function is not necessarily convex and so the minimal cost J^*, and the corresponding inputs $\mathbf{p}^*$, may not be unique. The solution as computed by a gradient based algorithm may indeed correspond to a local minimum that could be a long way from the global minimum, in which case an alternative so-called intelligent optimisation method such a simulated annealing algorithm may perform better [94]. However, in practice the new tool-paths $\mathbf{p}^*$ computed by the gradient based algorithms in the Matlab Optimisation Toolbox have been found to give improvements in the product geometry such that the products fall within the error tolerances specified by the user. Hence, only gradient based algorithms are used here, but the importance of making a good initial guess for the inputs $\mathbf{p}$ is stressed and for this, it is found that a tool-path that follows the contour of the target shape is often a good choice. The contour following tool-path is the best heuristic tool-path for a target shape, so it is also used as the 'benchmark' against which the optimal tool-paths are measured.



(a) Measured impulse responses from each layer of the forming process.



(b) The final deformation, given by the summation of the impulse responses in (a). Intermediate steps displayed as dotted lines.

Figure 5.2: Impulse responses and deformation field for a cone shaped part produced by a contour following tool-path.

5.2 Open loop path planning using impulse responses

To predict the final deformation field of the product, a simulation can be performed using a computational ISF model with a given tool-path. However, because numerical modelling has been found to be slow, an alternative ‘spatial impulse response’ model has been investigated and the results are provided in this section.

The idea behind the spatial impulse response method is analogous to the time domain approach used in control engineering, where a system is characterised by its response to a set of unit impulses applied sequentially to each of its input variables, with zero initial conditions. In [35] and [36], Duncan *et al* describe two methods for thickness control in the metal processing industry based on continuous time impulse responses, in strip metal rolling and spray coating respectively. Examples of recent applications of control in metal forming are Arai’s [16, 17] simple yet effective use of proportional and integral (PI) control to regulate thickness in conventional spinning, and Endelt and Danckert’s [39] proportional feedback control scheme to minimise wrinkling in deep-drawing. In Endelt and Danckert’s work the feedback gain matrix was computed by solving an optimal control problem using LS-Dyna for the underlying model. This is one of the few examples of FE modelling in the control of metal deformation and it is feasible because the model is only used in a single off-line computation, since the gain matrix is fixed during the actual process.

For SPIF, the spatial impulse response is defined as the change in sheet geometry created by the shortest *appropriate application* of the controlled tool, where an appropriate application can be chosen to best suit the target product geometry. For instance, in a recent example where spatial impulse responses were used to analyse various metal forming processes [86], the impulses were defined as the change in deformation from tracking the tool a small distance along a straight line over a flat sheet. This was the convenient choice in [86] because the required tool motion could be replicated in each of the forming processes tested.

Since the path optimisation problem here is concerned with the production of axisymmetric shaped parts, an appropriate application of the tool at layer i is to track a radius r_i

and indentation depth Δz_i with the tool, so that the change in deformation,

$$\mathbf{s}_i = \mathbf{x}_i - \mathbf{x}_{i-1}, \tag{5.15}$$

is the spatial impulse response of the sheet for stage i. For a given target shape the impulses can be generated from an experiment on a SPIF machine using a predefined tool-path with Δz_i set to 1 mm and $r_i = r_i^{(c)}$, where $r_i^{(c)}$ are the contour following tool-path radii for the part.

This was done for a cone shaped part with a 40 mm base radius and 45^o wall angle by conducting an experiment on the Cambridge ISF machine using the aluminium alloy H22-5251 of 1 mm thickness, with a 140×140 mm clamping frame and a tool-tip radius of 7.5 mm. These conditions were used for all of the experiments in this section. Figure 5.2 a) shows the impulse responses obtained with this tool-path on the Cambridge ISF machine using 40 layers, each 1 mm below the previous one. The stereo camera was used to image the part after each pass of the tool and since the product is axisymmetric, rather than displaying 3D images, the plots in figure 5.2 show the average deformation over the sheet at radial points from 0 to 60 mm from the centre of the sheet. The sum of the impulse responses in figure 5.2 a) gives the final deformation field that was obtained by the contour following inputs, as displayed in figure 5.2 b).

The collection of the deformation data was made feasible by the installation of a 3D camera on the Cambridge ISF machine. The camera is shown in figure 1.4 and it takes deformation images during the process in the form of point clouds after each layer of a product in formed. This is a significant improvement over the CMM measurement method used in the earlier model validation tests in §3.8 for the twisted cone shaped product. In order to create the plots in figure 5.2, a Matlab script was written to post-process the point cloud data from the stereo cameras. This performs the following steps:

- imports and samples data to reduce excessively large point cloud arrays
- automatically searches for common markers in each image and uses them to rotate and align successive images, with the convention that the base of the sheet lies on the xy plane and the edges are aligned to the x and y axes

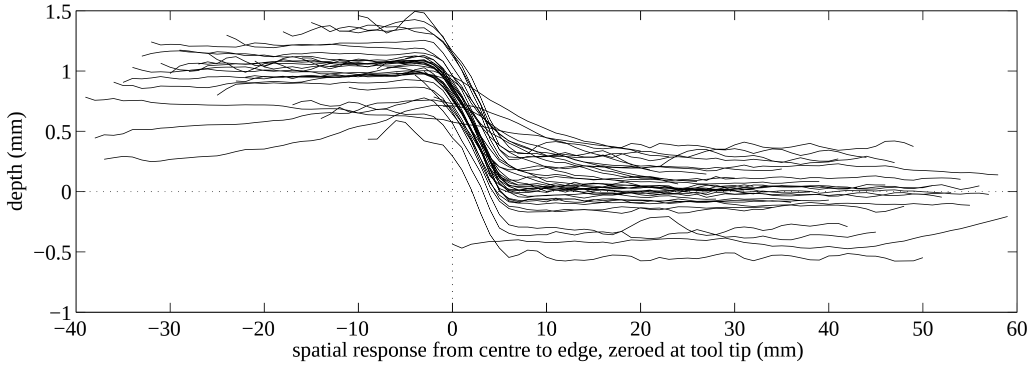


Figure 5.3: The impulse responses in figure 5.2 a) shifted so that the origin coincides with the radius of the circular tool path that produced the impulse.

- projects images onto a common set of xy coordinates
- computes the mean height, i.e. the z value, at radial points from the centre to the edge of the sheet, using linear interpolation between data in the point cloud where necessary
- computes and plots the impulse responses

The impulse response plot, $\mathbf{s}_i$, obtained at stage i is a discrete set of K values representing heights at radial points x from zero to 60 mm in figure 5.2 a). However, by interpolating between sample points this plot can be thought of as a continuous impulse described by the function, $s_i(x)$, that returns a value for any radial point x. The relationship between this function and the original data is

$$\mathbf{s}_i = [s_i(1\Delta x),\ \ s_i(2\Delta x), \ldots, s_i(K\Delta x)], \tag{5.16}$$

where Δx is the spacing between sample points. The impulse was generated by the contour following input $r_i^{(c)}$ and the spatial impulse response model assumes that the deformation that results from small deviations from the contour following input can be predicted by shifting the impulses s_i horizontally according to the alternative inputs r_i, so that

$$w_i(x) = s_i(x + r_i^{(c)} - r_i), \tag{5.17}$$

which is sampled to give the predicted impulse $\mathbf{w}_i$ due to the input r_i,

$$\mathbf{w}_i = [w_i(1\Delta x),\ \ w_i(2\Delta x), \ldots, w_i(K\Delta x)]. \tag{5.18}$$

Hence $\mathbf{x}_M$ in (5.1) is given by

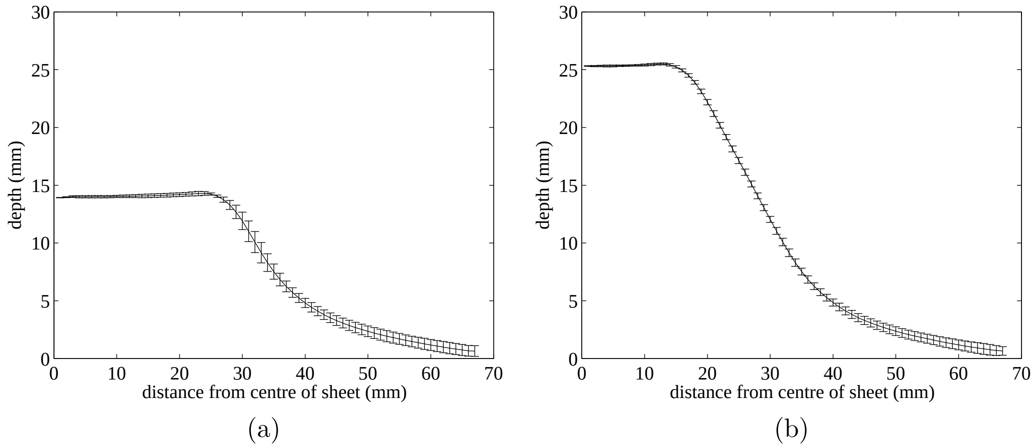
$$\mathbf{x}_M = \sum_{i=1}^{M} \mathbf{w}_i + \mathbf{s}_f, \tag{5.19}$$

where $\mathbf{s}_f$ is the effect of impulses to the sheet that are not controlled by the tool-path radii. For instance, the change in deformation due to springback when contact between the tool and sheet is removed at the end of the process.

The model can therefore be considered as a linearisation of the deformation behaviour about the contour following tool-path. In figure 5.3, the horizontal coordinates were shifted by plotting s_i against $(x + r_i^{(c)})$ instead of s_i against x as shown in figure 5.2 a), for $\{i = 1, \ldots, M\}$. This is equivalent to plotting $w_i(x)$ with all $r_i = 0$ in (5.17), and it aligns the impulse responses making them easier to compare. It appears that most of the impulses are similar in shape but they vary in height significantly. Some impulses have a large element of negative deformation, which is due to material spring back, stretching of the ISF machine and small movements relative to the camera mounting. If the spatial impulse responses were more uniform, it would be possible to parameterise the response shape and define a generating function that could produce all of the impulse response shapes. However, here it is more straightforward to store the impulse response plots and use them directly.

5.2.1 Measurement errors

Each point in an impulse response plot, at some distance r along the horizontal axis, is the mean value out of 72 measurements taken over the locus of a circle with a radius r centred in the middle of the sheet, referred to as the ‘measurement loci’. The 72 values could vary significantly if, for instance, there are inherent errors in the imaging hardware, if there is misalignment between the image data and the centre of the sheet, or despite the symmetry of the product, there are significant variations in the sheet deformation over a particular



(a) (b)

Figure 5.4: Two example plots computed from 3D deformation images where each point represents the mean value at radial points from the centre to the edge of the sheet. Error bars are plotted at each point to mark out ±2×(standard deviation) of the data.

measurement locus. Hence, before using the impulse responses, an analysis of the errors in the image plots was conducted by measuring the standard deviations in the data for each point in the image plots.

It was found that the typical standard deviation between the 72 measurements at each point was less than 0.6 mm. The standard deviation tends to increase at large values of r, as some points on the measurement locus approach the edge of the sheet. At these distances, the deformation at points on the measurement loci towards the corners of the sheet tend to be significantly higher than at other points. This is because of the mismatch between the square clamping frame used and the circular tool-path applied in the experiments.

Figure 5.4 provides two examples of these observations and displays two processed deformation images taken when forming the cone shaped part in figure 5.2 b), after the tool had completed a pass at 15 mm and 27 mm depth respectively. Error bars are displayed on these plots to mark out the distance two standard deviations above and below each point. They show that the deviations are small relative to the mean depths and they tend to increase as the distance from the centre of the sheet increases towards the edge of the sheet. The maximum standard deviations were 0.457 mm and 0.230 mm for the 15 mm and 27 mm plots respectively and overall the errors in the image data plots were considered small enough to use in the path planning problem.

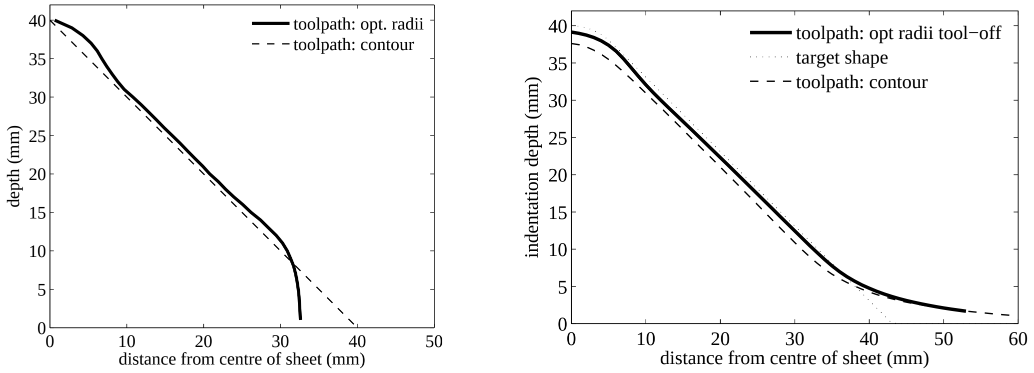
5.2.2 Implementation

It has been noted that the summation of the original impulse responses $\mathbf{s}_i$ gives the sheet deformation shown in figure 5.2 b), so an alternative summation of the impulses $\mathbf{w}_i$ might be found that brings the final product closer to the target shape. A scheme based on this principle was created in the form of the discrete optimal control problem in (5.1), which was then coded as a constrained optimisation problem in Matlab and solved using the gradient based optimisation tool-box algorithm *fmincon*.

As noted in §5.1.2, the solution space is not convex, as observed by testing the optimiser with several initial states, from which the routine exits with different, local, minimal costs. With the tests conducted here, when the contour following tool-path was used as the initial guess, it often appeared to give the lowest final cost and it usually reached a solution within a minute. A 'very close optimal' solution obtained from a good initial guess is likely to perform just as well as 'the optimal' solution, so the solutions obtained by *fmincon* are considered acceptable in practice.

For the first term in the cost function, $\Phi(\mathbf{x}_M) = \|\mathbf{x}_M - \hat{\mathbf{x}}\|_2$, the target shape $\hat{\mathbf{x}}$ was set to the cone shaped part that was the basis for the contour following tool-path from which the impulses $\mathbf{s}_i$ were generated. The optimisation problem was run using a scaling factor of $\rho = 1$ with $\mathbf{p}^{(c)} = [r_1^{(c)}, \ldots, r_M^{(c)}]$ as the initial guess for the algorithm. The resulting set of optimal tool-path radii are plotted in figure 5.5 a), along with the original contour following tool-path for comparison. Initially, an off-line scheme was tested where the optimised r_i were applied directly to the ISF machine without using feedback to modify the tool-path on-line.

The measured deformation resulting from both the contour and optimal tool-paths are displayed with the target shape in figure 5.5 b), showing that the optimised tool-path does improve the fit to the target shape. Since there is no resistance on the side of the sheet opposing the tool, it is not possible to avoid the 'flanging', or raised edges, at the base of the product by optimising the tool-path alone. However, as discussed in §5.2.5, this flanging can be removed by simply allowing the flange to form outside of the target region of the product, or by using a backing plate. By measuring points over the sheet from the centre out to 40 mm, the root-mean square (RMS) error between the target shape and actual



(a) heuristic tool-path and tool-path with optimised radii (b) deformation resulting from both tool-paths

Figure 5.5: Optimisation of cone shaped product.

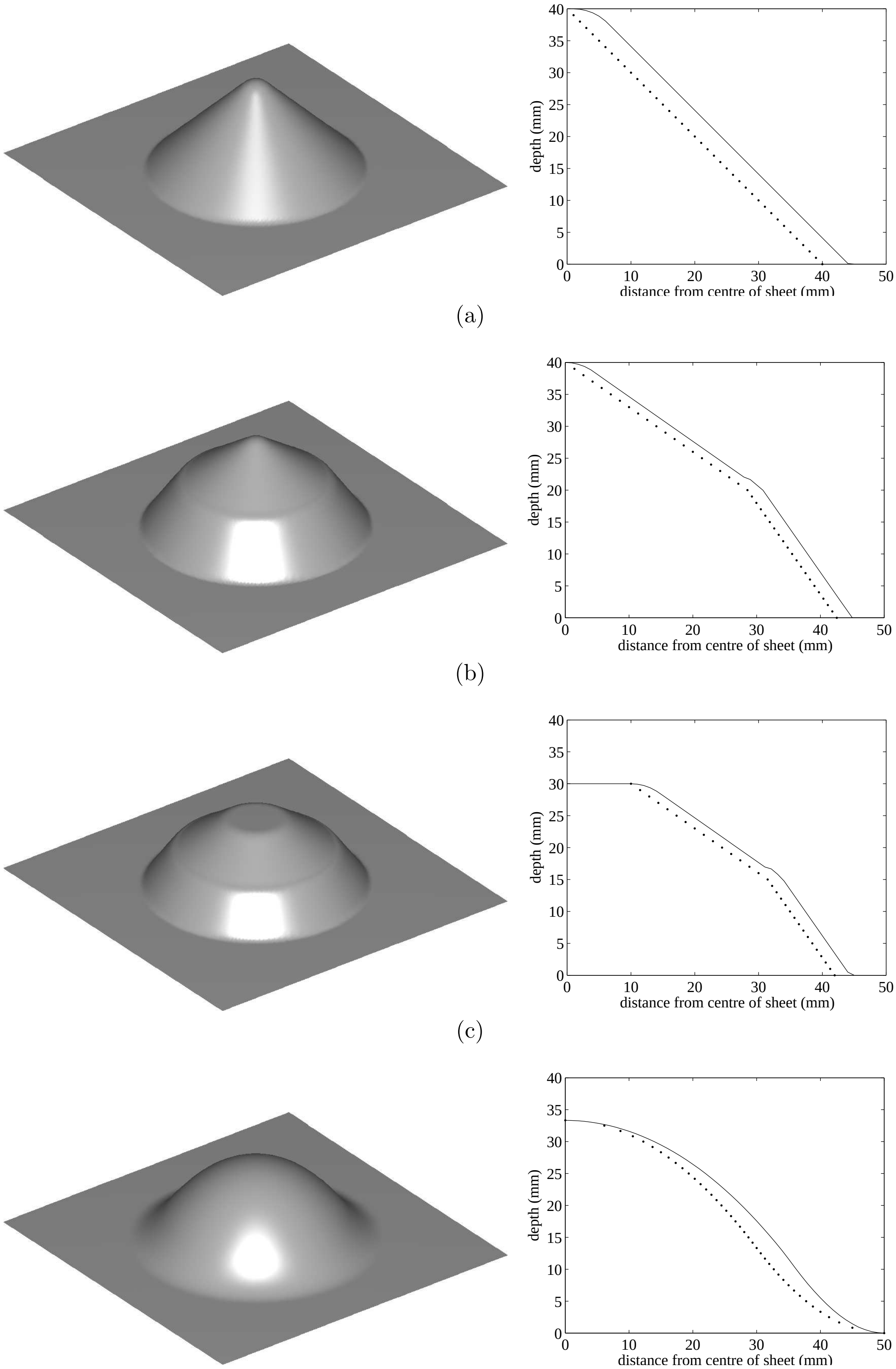
Toolpath	**RMS** (mm) **0-40 mm**	**Max. diff.** (mm)	Benefit
Contour	2.10	2.53	-
Optimised r_i	0.80	1.14	62.0%

Table 5.1: Error measures between target and actual product. Further data for this part is given in table 5.8 under shape a). Benefit = improvement in RMS error over the contour following tool-path.

shape was computed. For the contour following tool-path, the RMS error as displayed in table 5.1 is 2.10 mm, over which the optimised tool-path gives a 62% improvement, with an RMS error of 0.80 mm.

5.2.3 Application to different test shapes

The above test shows that the experimental impulse responses can be used as a suitable predictor for generating an optimal tool-path trajectory off-line, and therefore to improve the shape of the product that generated the impulse responses. It is now possible to ask an interesting question: can the same impulse response plots be used to predict the optimal tool-path required for a different axisymmetric shape? In order to test this, a set of alternative target shapes, $\hat{\mathbf{x}}$, were chosen such that each introduces one level of complexity over the previous product. The chosen target shapes are shown in figure 5.6: image a) is the original cone shaped product; b) is a cone shape with two wall angles; c) introduces a flat-topped region; d) introduces a 2nd order profile rather than the linear profiles used in the previous three shapes. The contour tool-paths associated with each part are also shown as dotted lines in the figure.



(a)

(b)

(c)

(d)

Figure 5.6: Target shapes (3D image & solid line) with contour tool-path (dotted line).

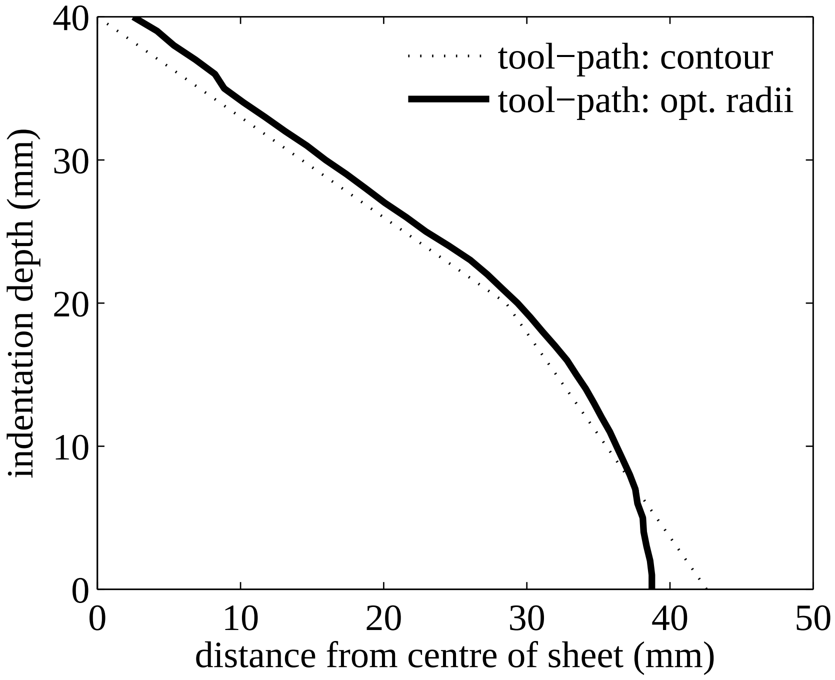


Figure 5.7: Comparison of standard contour toolpath with open loop optimised tool-paths.

Product with two wall angles

The first new target shape tested, shape b), has two wall angles, 55^o from the base up to 20 mm of the forming height and 35^o for the remaining 20 mm up to the final depth of 40 mm. As previously, the heuristic tool-path follows the contour of the product in 40 equal layers of 1 mm indentation steps. Using this tool-path as the initial guess for the optimisation algorithm, figure 5.7 shows the optimal tool-path obtained as well as the contour following tool-path for comparison. As with the cone shaped product, the optimal tool-path here pushes the sheet beyond the perimeter of the target shape to account for springback. In figure 5.8, the deformation from each tool-path in figure 5.7 is displayed.

When measured out to 45 mm, the contour following tool-path gives an RMS error of 1.72 mm whereas the optimised trajectory gives 1.27 mm. In the most controllable region of the sheet, up to 35 mm from the centre, the optimised tool-path gave a 76% reduction in the RMS error between the target shape and actual product, from 1.58 mm to 0.37 mm. These results are also displayed in table 5.4. The region beyond 35 mm is 'uncontrollable' in the sense that the flanging that develops at the base of the product cannot significantly be prevented by changing the input tool-path, $\mathbf{p}$.

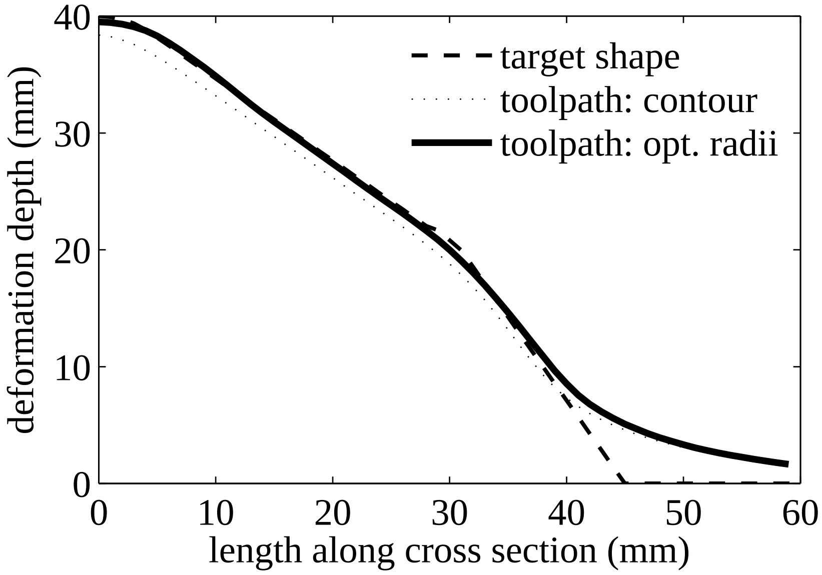
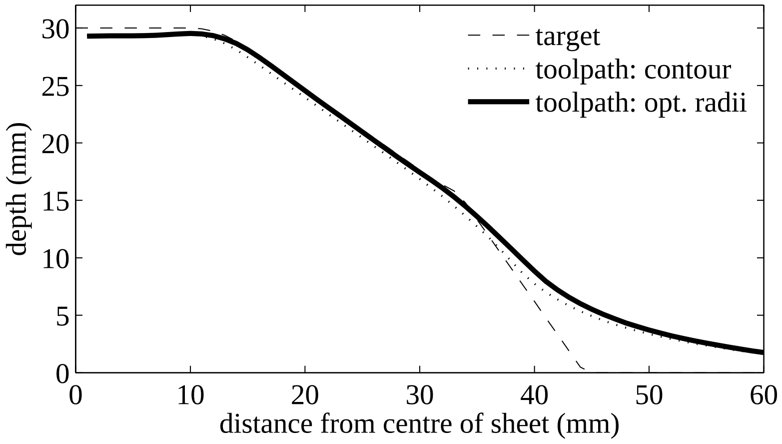


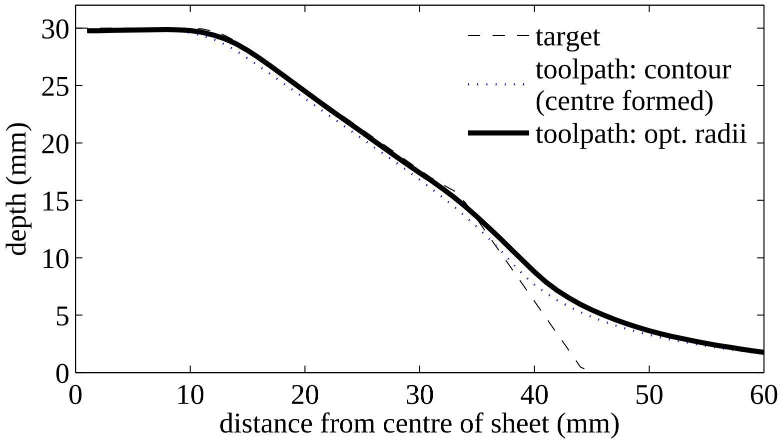
Figure 5.8: Shape of cone with 55^o wall angle for the first 20mm and 35^o for final 20mm.

Product with a flat top

The next product tested is shape c) in figure 5.6, which is similar to shape b) except for the flat region at the centre of the product. There are two options for forming the flat region of the sheet. For the first case, the contour following tool-path left this region unformed by the tool, although this does result in a small ‘dip’. Only the first 30 impulses in figure 5.3 were used because the product depth was 30 mm. The final shape resulting from the optimised tool-path is displayed in figure 5.9 a), showing a good fit between 10 mm and 35 mm from the centre of the sheet. The region beyond 35 mm is not expected to fit well because of the flanging observed in previous parts, but a better fit would be expected over the flat region of the sheet if the ‘dip’ can be removed. This was done in the second case, where a simple modification was made to both the countour tool-path and the optimised tool-path, by forming over the flat region of the product using a tool-path that spiralled inwards from 10 mm to 0 mm at a depth of 30 mm. The effect of this final step was treated as an extra impulse to the sheet that could not be altered by the control inputs and therefore, it was added to $\mathbf{s}_f$ in the process model in (5.19). The resulting fit to the target shape is shown in figure 5.9 b) and the RMS errors for both plots in figure 5.9 can be compared in table 5.2.



(a)



(b)

Figure 5.9: Flat topped shape. a) No central forming, b) with central forming.

Product with second order wall profile

The final test shape is the one shown in figure 5.6 d), which has a forming depth of 33 mm and therefore requires only the first 33 impulse response plots in figure 5.3 for the model. The optimal set of r_i were computed in exactly the same way as for the previous test products and similar deformation plots were produced for comparison, as shown in figure 5.10. Since the shape is more complex than the previous examples, the fit between the actual and target product is not expected to be as good as in the previous examples. The RMS errors shown in table 5.3 provide a quantitative measure for this, which shows that when measuring out to 45 mm there is little difference between the two tool-paths, but the optimised tool-path gives an improvement of 58% over the first 35 mm of the product.

	up to 45 mm			
Toolpath	**RMS** (mm)	**Var** (mm^2)	**Max. diff**	**Benefit**
Contour	1.49	2.28	4.92	-
Optimised r_i	1.70	2.75	5.53	-13.9%
Contour with centre formed	1.46	2.18	4.83	-
Optimised r_i with centre formed	1.65	2.50	5.45	-12.8
	up to 35 mm			
Toolpath	**RMS** (mm)	**Var** (mm^2)	**Max. diff**	**Benefit**
Contour	0.73	0.03	1.32	-
Optimised r_i	0.39	0.06	0.70	46.3%
Contour with centre formed	0.73	0.11	1.39	-
Optimised r_i with centre formed	0.24	0.01	0.57	67.2%

Table 5.2: Error measures from the plots in figure 5.9.

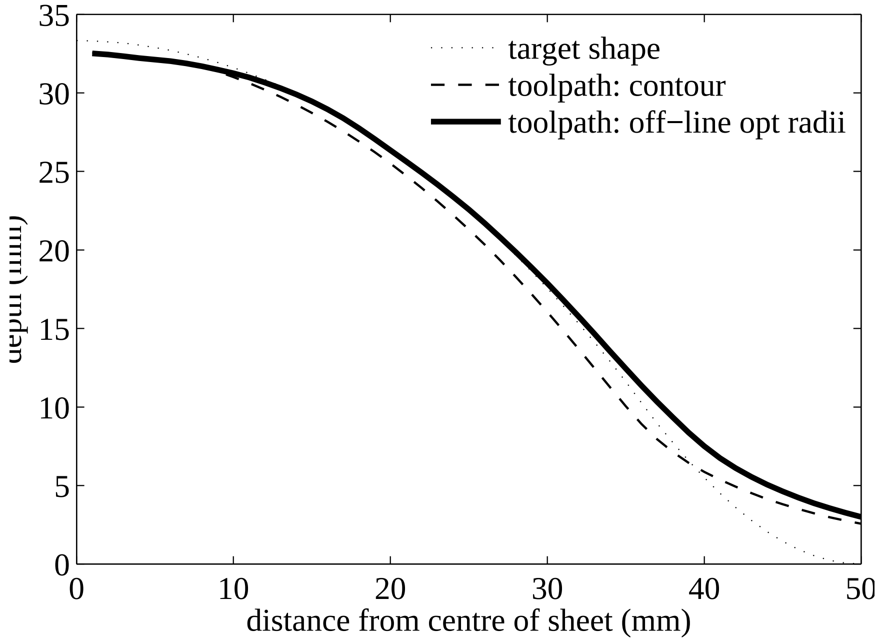


Figure 5.10: Deformation resulting from the contour and optimised tool-path for the target shape in figure 5.6 d).

	up to 45 mm			
Toolpath	**RMS** (mm)	**Var** (mm^2)	**Max. diff**	**Benefit**
Contour	1.13	0.89	2.36	-
Optimised r_i	1.13	1.15	3.18	0.1%
	up to 35 mm			
Toolpath	**RMS** (mm)	**Var** (mm^2)	**Max. diff**	**Benefit**
Contour	1.06	0.13	1.66	-
Optimised r_i	0.44	0.19	0.84	58.2%

Table 5.3: Error measures from the plots in figure 5.10.

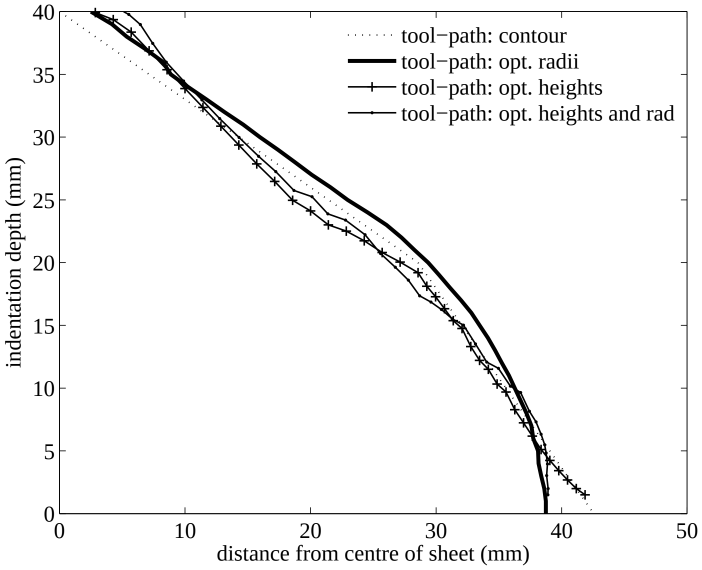


Figure 5.11: Comparison of standard contour toolpath with open loop optimised tool-paths.

	up to 45 mm			
Toolpath	**RMS** (mm)	**Var** (mm^2)	**Max. diff**	**Benefit**
Contour	1.72	2.02	4.58	-
Optimised r_i	1.27	1.11	5.09	26.2 %
Optimised Δz_i	1.77	0.94	4.41	-2.7 %
Optimised Δz_i & r_i	1.76	3.04	4.86	-2.3 %
	up to 35 mm			
Toolpath	**RMS** (mm)	**Var** (mm^2)	**Max. diff**	**Benefit**
Contour	1.58	0.04	2.11	-
Optimised r_i	0.37	0.10	0.91	76.4%
Optimised Δz_i	1.68	1.90	2.59	-6.0%
Optimised Δz_i & r_i	1.54	2.43	2.39	3.0%

Table 5.4: Error measures between target and actual product.

5.2.4 Indentation depth as a variable

Having shown the effectiveness of the tool-path radii $\mathbf{p} = [r_1, \ldots, r_M]$ as control inputs, one may now consider allowing the indentation depths to vary too, so that $\mathbf{p} = [r_1, \ldots, r_M,\ \Delta z_i, \ldots, \Delta z_M]$. This requires (5.17) in the process model to be modified to

$$w_i(x) = \frac{\Delta z_i}{\Delta z_i^{(c)}} s_i(x + r_i^{(c)} - r_i), \tag{5.20}$$

where $\Delta z_i^{(c)}$ were the original indentation depths used to generate s_i, which were all 1 mm. Two versions of the optimal control routine were solved using this modification, firstly where both r_i and Δz_i were allowed to vary and secondly where r_i were fixed to the

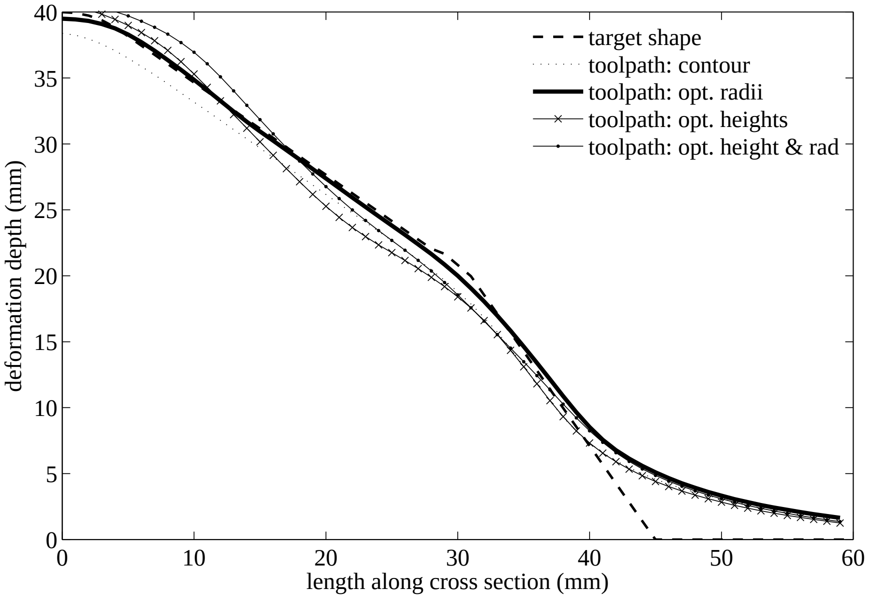


Figure 5.12: Shape of cone with 55^o wall angle for the first 20 mm and 35^o for final 20 mm. The fit is poor when deformation depth is allowed to vary.

contour following radii. To restrict excessive variations in Δz_i, the constraints $0.9\Delta z_i^{(c)} \leq \Delta z_i \leq 1.1\Delta z_i^{(c)}$, where $\{i = 1, \dots, M\}$, were also applied. Figure 5.11 shows the optimal trajectories when using shape b) as the target shape and figure 5.12 shows the resulting deformation.

The fit to the target shape is very poor, suggesting that it is not possible to use the heights Δz_i as control variables, or specifically that, unlike radii, the spatial impulse response model is not suitable for predicting the change in process behaviour with respect to changes in depths. For this reason only r_i should be controlled with the impulse responses defined here, but it may be possible to define an alternative set of impulse response shapes that can be used for depth optimisation.

5.2.5 Removal of flanging at the base

Due to the lack of restraint on the underside of the sheet, it is not possible to prevent the flaring in the base regions of the sheet seen in, for instance, figures 5.2 b), 5.5 b) and 5.8. Hence, a sharp corner cannot be obtained at the base of the target shape by controlling the tool-path alone unless, for example, a second restraining tool is available as with the TPIF process described in §4.4.2. For the Cambridge ISF machine, two alternative solutions are available that do not require a second tool:

1. to treat the base region as a ‘pre-form’ beyond the range of the target shape.
2. to use a backing plate to restrain the sheet (figure 5.14).

Both of these suggestions are now explored below, where in order to demonstrate the methods, they are used to optimise the product geometry for the target shape in figure 5.6 b).

Relative error measures

The product error measure minimised so far is the absolute RMS error, but in practice there is no reason why the product should not be free to shift vertically to improve the fit, by minimising the relative error between the two shapes. In conjunction with this, the target shape can be defined so as to allow for flanging to occur outside of the target region and reduce errors at the base of the product. It is shown here that both of these techniques can be used effectively along with the path optimisation scheme. This ‘adaptive shifting’ approach is attractive because it requires the no modification to the ISF equipment.

The example part used here is shape b), but this time with a target base radius of 30 mm. The product was formed in 40 layers of 1mm depth, of which the initial layers were present only to produce a flange that would subsequently be cut away. All the tool-path radii r_i, for $i = 1, \ldots, 40$, including those at the base, remained as variables in the tool-path optimisation routine.

In order to allow the target shape to shift, a further optimisation variable a was introduced that was added to the depth coordinates of the target shape in the cost function $\Phi(\mathbf{x}_M)$, which was modified to

$$\Phi(\mathbf{x}_M, a) = \|\mathbf{x}_M - \hat{\mathbf{x}} + a\|_2. \tag{5.21}$$

This effectively allows the target shape to shift vertically by a mm when computing the optimal tool-path radii. The resulting radii were run on the Cambridge ISF machine and the final product was imaged and compared to the target product as previously.

The target shape is shown as the black dotted lines in figure 5.13, with the contour and optimised tool-paths shown respectively as dashed and solid lines, and the associated

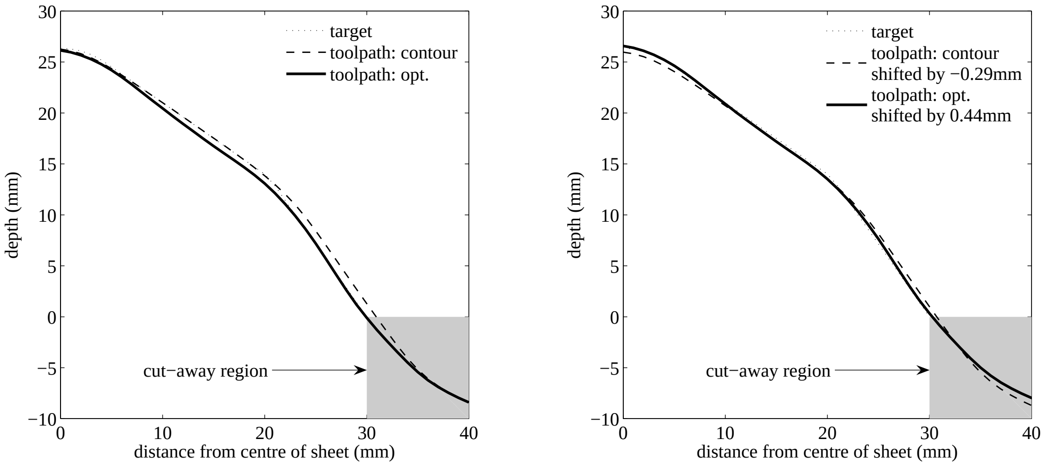


Figure 5.13: Deformation field when allowing flange to develop (left) and shifting the target shape vertically (right) for the best fit. The area of the sheet to be cut away is shaded in gray.

error measures displayed in table 5.5. The left hand plot shows the final deformation as it was obtained from the camera, whereas the right hand plot displays the deformation after the plots were shifted vertically so as to minimise the RMS error. As seen in table 5.5, the RMS error over the region is reduced by the optimised tool-path even if a shift is not applied, but this is reduced further to 0.29 mm when shifted to $a = 0.44$ mm, which is a 54% improvement over the contour following tool-path. It should be noted that the tool-path optimisation routine predicted that at a vertical shift of $a = 1.1$ mm would be needed, whereas with the actual product, the best fit was obtained when $a = 0.44$ mm. It is also interesting to note from figure 5.13 that the reason why the optimised shape gives a lower RMS error is that it follows the target profile better than the product formed by the contour following tool-path, which is too high at the base of the target product and too low at the peak.

The conclusion that may be drawn from this test is that in order to overcome the issue of the flanging, one option is to accept that the flange will form and simply allow for it by ensuring that it will fall outside of the region of the target product. Also, it is beneficial to allow the target shape to shift vertically, although the actual amount of shift predicted may differ in the real process.

Toolpath	**RMS** (mm)	**Benefit**
Contour	0.65	-
Optimised r_i	0.51	21.8%
Contour shifted	0.58	-
Optimised r_i & shifted	0.27	54.0%

Table 5.5: Error measures from the plots in figure 5.13.

Using a backing structure

Here, it is demonstrated that the path optimisation scheme can be used in conjunction with a backing plate that restrains the underside sheet and reduces flanging at the base of the product. For the axisymmetric products considered here, the backing plate is simply a frame with a circular cut-out, as shown in figure 5.14, with a radius equal to that of the base of the product. The test here was conducted on the optimisation of a cone shaped product with two wall angles and a 50 mm base radius, for which an 18 mm thick wooden backing frame was readily available. The shape of the impulse responses were expected to be quite different because of the new restraint, so a new set of impulse response plots had to be generated, which was done experimentally by using a contour following tool-path. Figure 5.15 shows the impulse response plot, where the increased uniformity when compared to figure 5.3 is noticeable.

In figure 5.16, the contour and optimal tool-paths are compared, showing very little difference between the two tool-paths. The main difference is that the optimised tool-path pushes a little further into the sheet and tapers outwards at the tip in order increase the depth of the product, which was about 2.5 mm below the target shape when using the contour tool-path. The deformation fields resulting from the two tool-paths are shown in figure 5.17. The optimised tool-path showed a smaller improvement against the contour toolpath than before, which is not unexpected because of the presence of a backing restraint. However, it is interesting to observe that this path optimisation method still gives some improvement, even though the configuration of the machinery is altered, demonstrating the versatility of the method.

Figure 5.18 shows the deformation obtained using the tool-paths in figure 5.16 when a further error reducing measure was applied to the sheet, by cutting slots into the sheet around the base of the product before forming with the backing plate. The idea came from

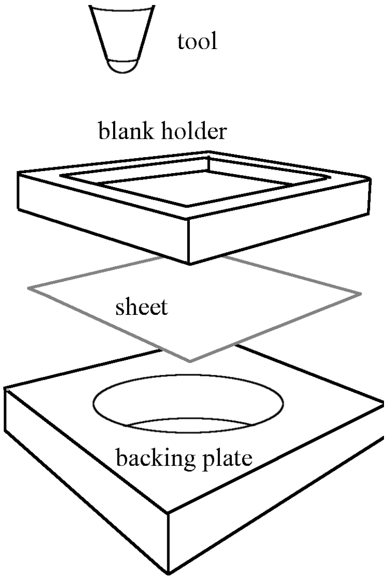


Figure 5.14: Expanded view of the basic ISF set-up with a backing plate.

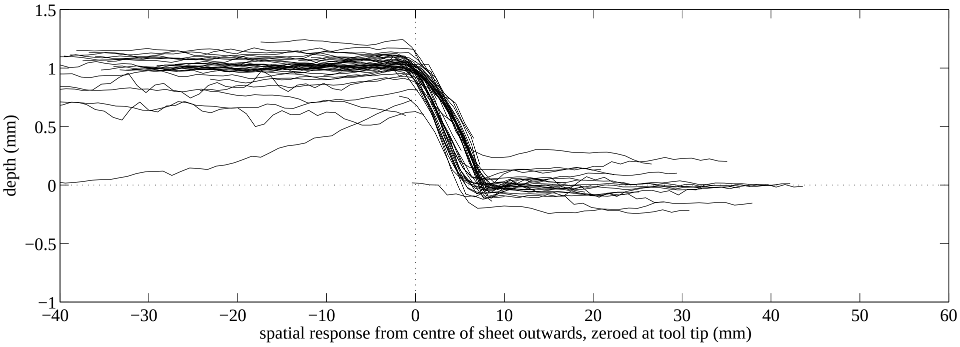


Figure 5.15: The impulse responses obtained when a backing plate is used. The origin for each impulse is defined such that it coincides with the radius of the tool-path that produced the impulse.

a visiting masters student's work at the IfM in Cambridge[1] on tests to see if reducing the bending stiffness by cutting slots at the base of the sheet would improve the deformation field. In this case, the reduction in error is significant, as observed in table 5.6, where the error measures for the tests in figures 5.17 and 5.18 are displayed.

5.2.6 Multi-pass forming

In §5.2.3, an application of the impulse response path optimisation method was demonstrated that produced a product that fitted the target shape better than the basic contour tool-path did. An interesting result was demonstrated; that the impulse responses measured from a part that was different to the target product could be used to optimise the

---

[1] Daniel Braun, masters student at Technische Universität Dortmund, visitor summer 2008.

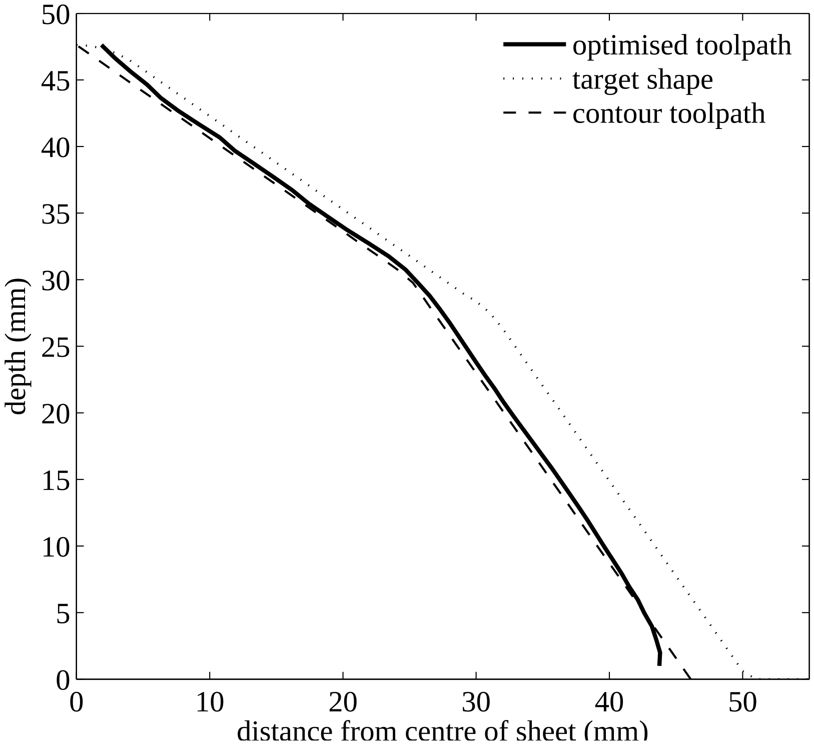


Figure 5.16: Tool-path for test with backing plate.

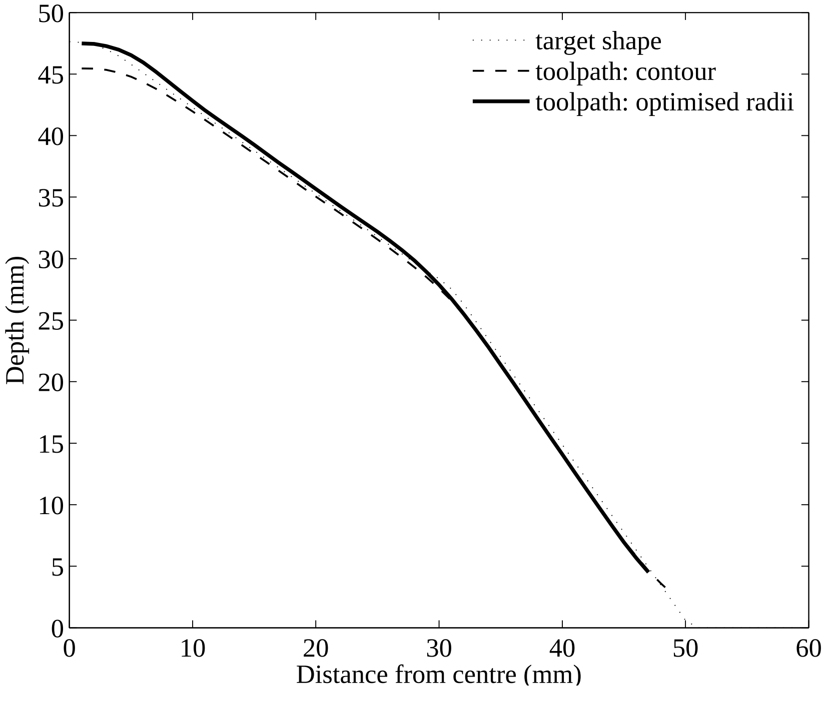


Figure 5.17: Deformation with backing.

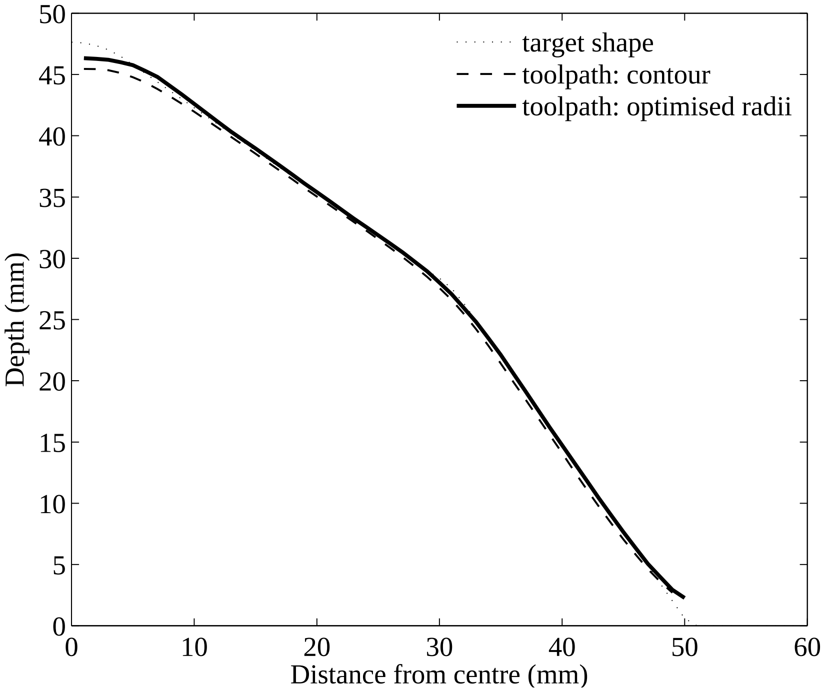


Figure 5.18: Deformation with backing and slots.

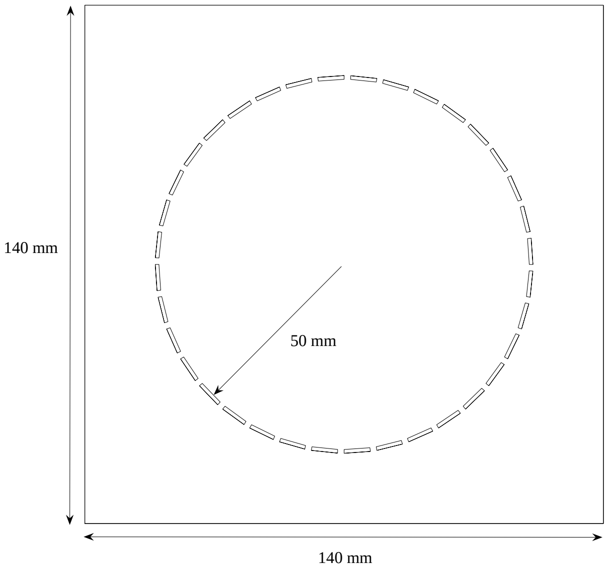


Figure 5.19: View from above of slots cut into the sheet for a second test using the backing plate.

Toolpath	**RMS** (mm)	**Benefit**
Contour with backing	0.75	-
Optimised r_i with backing	0.58	22.4%
Optimised r_i & with backing and slots	0.33	55.4%

Table 5.6: Error measures for cone shape with 2 wall angles when using a backing structure and slots.

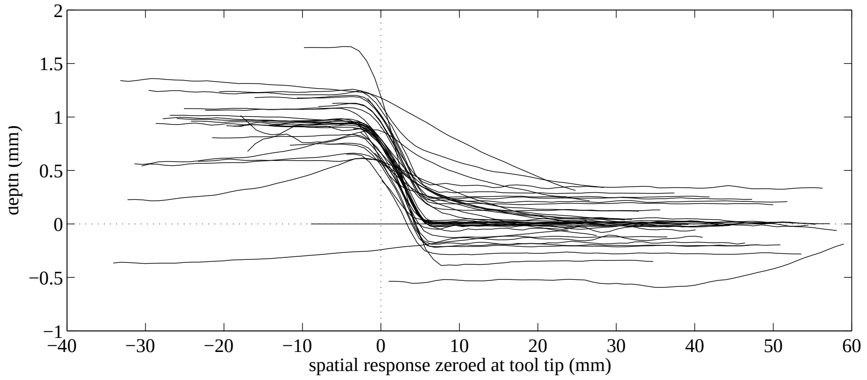


Figure 5.20: Impulse responses for inner shape in multiform test.

tool-path, hence showing that the impulse responses are to some extent transferable. In particular, it was shown that the impulse responses for a specified tool-path can be used to optimise the tool-path for several different target shapes.

The scheme described needs the impulse response data to be generated at some point prior to forming the part, wasting a sheet of metal in the process. This waste is not an issue for many applications, but for others, for instance where the material is particularly costly, it could be disadvantageous. It is worth considering whether it is possible to extend the method and produce a product in, for instance, two passes over the same sheet. The first pass would be to generate a set of impulses responses, which can then be used produce an optimal set of radii r_i to apply over the second pass.

As an initial test of this strategy, a small cone shape was produced on a sheet with a 45^o wall angle and 36 mm radius using 0.9 mm indentations. This produced the 40 impulse responses in figure 5.20, which were all scaled in height by a factor of $\frac{1}{.9}$ for the 1 mm steps used in the second pass. The impulses were then used to predict the optimal tool-path for the product in figure 5.8. It is noted that due to the preformed inner shape, the impulse responses that would be observed over the second pass are different to those in figure 5.20, but a more detailed study than the one conducted here should be pursued to find a method for generating more suitable impulse response shapes for this multi-pass strategy.

Figure 5.21 shows that the predicted tool path is actually very similar to the one in figure 5.7, which is repeated in figure 5.21 (labelled ‘normal’) for convenience. The product shape resulting from the multiform strategy is shown in figure 5.22. Interestingly, the part does fit the target shape better than the one produced by the contour following tool-path,

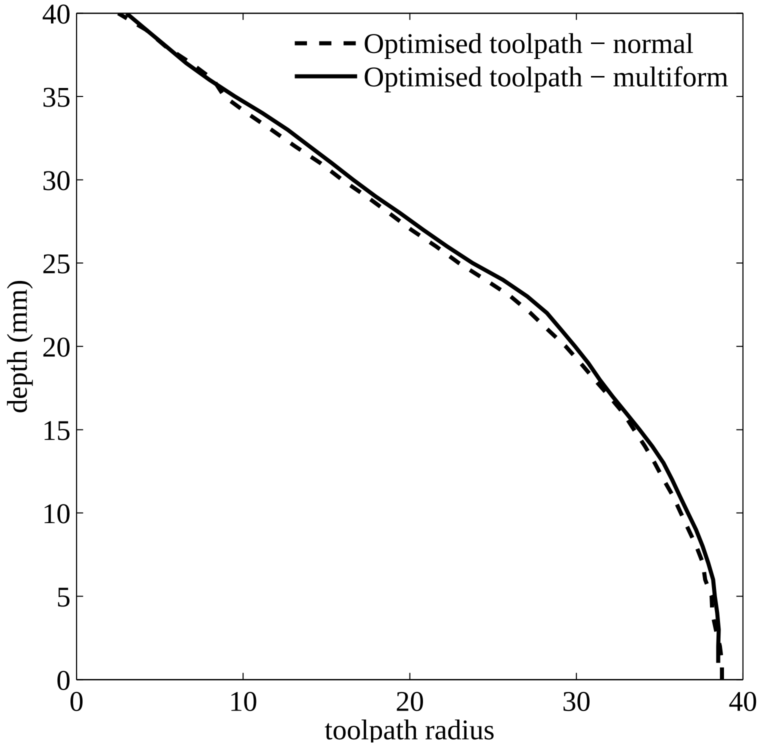


Figure 5.21: Comparison of tool-path generated using impulse responses in §5.2.3 and that produced by using the multi-pass strategy, for the target product in figure 5.8.

although of course not as well as the r_i optimised tool-path in figure 5.8. The results are summarised in table 5.7 by displaying the error measures between the target and actual product along with the corresponding test results from §5.2.3. Some of the error in this method is due to the presence of the inner shape, which is not accounted for in the impulse responses, so by compensating for the inner shape a further reduction in the errors should be observed.

	up to 45 mm	
Toolpath	**RMS** (mm)	**Benefit**
Contour	1.72	-
Optimised: ‘normal’	1.27	26.2%
Optimised: ‘multi-pass’	1.69	1.9%
	up to 35 mm	
Toolpath	**RMS** (mm)	**Benefit**
Contour	1.58	-
Optimised: ‘normal’	0.37	76.4%
Optimised: ‘multi-pass’	0.69	56.5%

Table 5.7: Error measures between target and actual product.

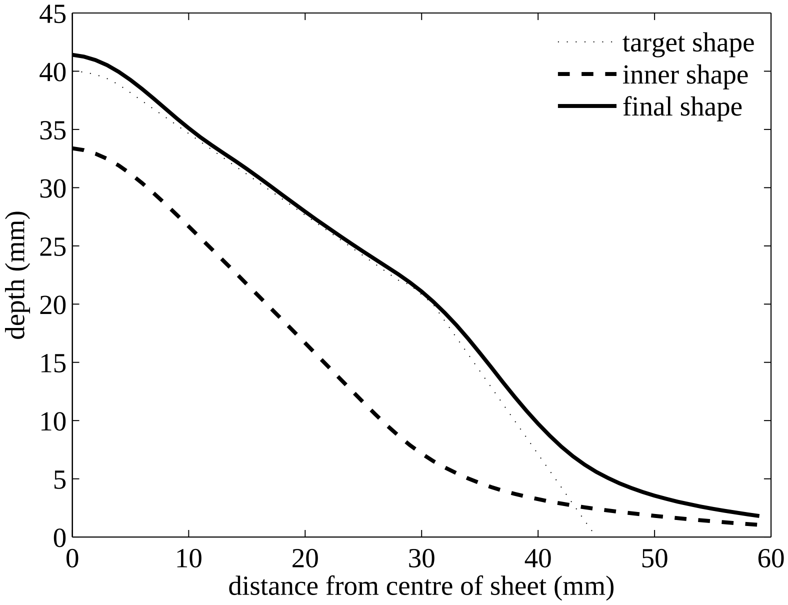


Figure 5.22: Shape of cone product with 55^o wall angle for the first 20mm and 35^o for final 20mm.

5.3 On-line path planning using impulse responses

The model of ISF for axisymmetric parts based on experimental impulse response shapes clearly has its drawbacks, since it assumes that a shifted summation of impulse responses is sufficient to describe the deformation due to small changes in the tool-path radii. However, the model is very quick, which is why it can be used to solve the open-loop tool-path optimisation problem, as was demonstrated in §5.2.

Despite the rather inaccurate underlying model, the off-line optimised tool-path radii produce significantly improved part geometries when compared to the contour following tool-path. This demonstrates that the model is sufficient for path optimisation and also that the contour following tool-path is a good initial guess for the input to the numerical optimisation routine. In this section, it is demonstrated that feedback can be used to reduce the dependence on the impulse response model and as a result, further improvements in part geometry are observed. This is particularly interesting because there is little evidence in metal forming literature of the use of feedback to regulate part geometry, even though it has been used to regulate sheet thicknesses and to control tool forces to prevent tearing, as cited in §1.3. Even without considering other metal forming techniques, the work described here is the first successful application of feedback for the control of geometry in an ISF process.

5.3.1 Implementation

The method employed is to apply the first input from an off-line optimised tool-path to the ISF machine, which is computed by using the same routine as in §5.2, and then to measure the resulting deformation using the stereo camera system. This data is used to replace the impulse response data up to the current layer and, along with the remaining impulse response shapes, it is used as a new process model in the path optimisation routine, which is re-run to re-optimise the tool-path from the current position. The first input from the newly optimised tool-path is applied as the next input to the process and the procedure is repeated until the part has been formed. The optimisation procedure at some stage $i = I$ is,

$$\begin{aligned} &\min_{\mathbf{p}} \; J = \Phi(\mathbf{x}_M) + \rho \sum_{i=I}^{M} L_i(\mathbf{x}_i, r_i) \\ &\text{s.t. } \mathbf{x}_i = \mathbf{m}_{i-1} + \mathbf{w}_i(r_i), \quad i = I, \dots, M, \end{aligned} \tag{5.22}$$

where $\mathbf{m}_i$ is the measured deformation after stage i of the process. Hence, the tool-path is re-optimised M times during the process according to (5.22). Since the optimised tool-path at each layer is shorter by one input compared to the previous layer, this method will be referred to as a fixed horizon model predictive control scheme, although some may prefer the term ‘shrinking horizon’ or ‘fixed end-time’. A schematic of the procedure is displayed in figure 5.23. To perform the feedback at each layer i, the image data is generated using the camera software and transferred to Matlab, where it is processed to generate $\mathbf{m}_i$ and passed through the path optimisation routine. The first output of this routine is then transferred to the tool motion software and applied to the machine. In order to demonstrate the method, this procedure is currently performed manually and it takes between four to six minutes during which time the machine is at rest. This pause does not effect the deformation behaviour, so it is not a significant issue, but for an industrial implementation it would be straightforward to fully automate and quicken the feedback process.

In general, the previous optimal tool-path is used as the initial guess in the optimiser, although to avoid getting stuck in local minima several other initial guesses were tried at some points and the solution with the lowest cost was used. This was a suitable approach for demonstrating the concept of the proposed on-line optimisation scheme, but it is suggested that in the future several initial guesses should be used at *every* step. A systematic method for generating these initial guesses would need to be developed in order to do this, especially if the procedure is to be automated.

Four test parts were produced based on the ones in figure 5.6, the exact dimensions of which are shown in figure 5.24. As before, the impulses used in the model were those displayed in figure 5.3. The first target shape to be optimised, shape a) in figure 5.24, is the same as the one that the impulses were generated from. The subsequent shapes from b)-d) were chosen to analyse how the on-line path optimistion scheme performed as the target shape deviated from the shape that the impulse responses were generated from, since each one is considered to add an extra level of deviation over the preceding shape. For instance, shape b) maintains the profile of shape a) but it is scaled down by a factor of $\frac{5}{6}$ in both height and base radius, shape c) then adds the complexity of forming two flat wall profiles of 55^o and 35^o and finally, target shape d) deviates further with a more complex profile defined by two quadratic equations.

All of the parts were formed in 40 layers, so to model shapes b)-d), the spatial impulse responses were scaled by modifying (5.17) to

$$w_i(x) = \frac{5}{6} s_i(x + r_i^{(c)} - r_i). \tag{5.23}$$

This may appear to contradict the result in §5.2.4, where it was shown that the spatial impulse responses cannot individually be scaled by different amounts to predict changes in depth. In (5.23), however, a uniform scaling of the height of all of the response plots is applied and, as shown below, the resulting model is found to perform suitably.

5.3.2 Results

The final profiles resulting from the new method for the four test shapes a)-d) in figure 5.24 are plotted in figure 5.25 and for comparison, the profiles resulting from the contour following tool-path and the off-line optimised tool-path are also displayed. The corresponding

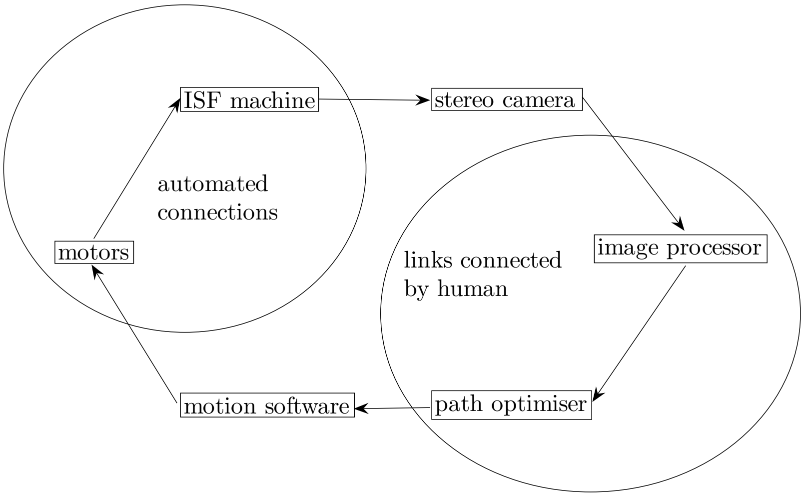


Figure 5.23: Schematic of the feedback loop used in the on-line path optimisation scheme.

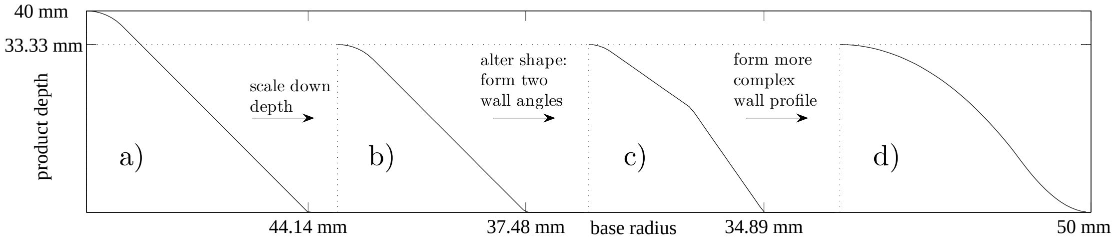
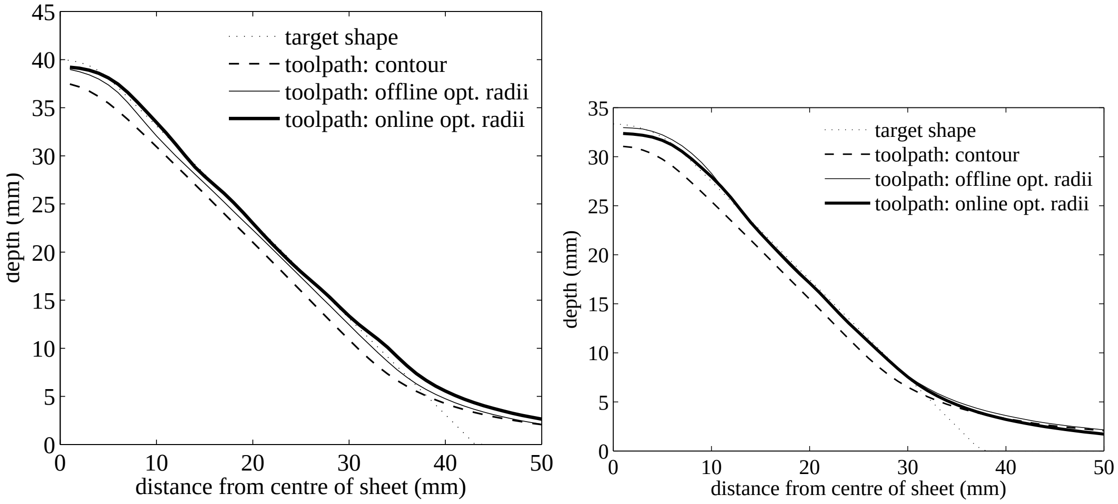


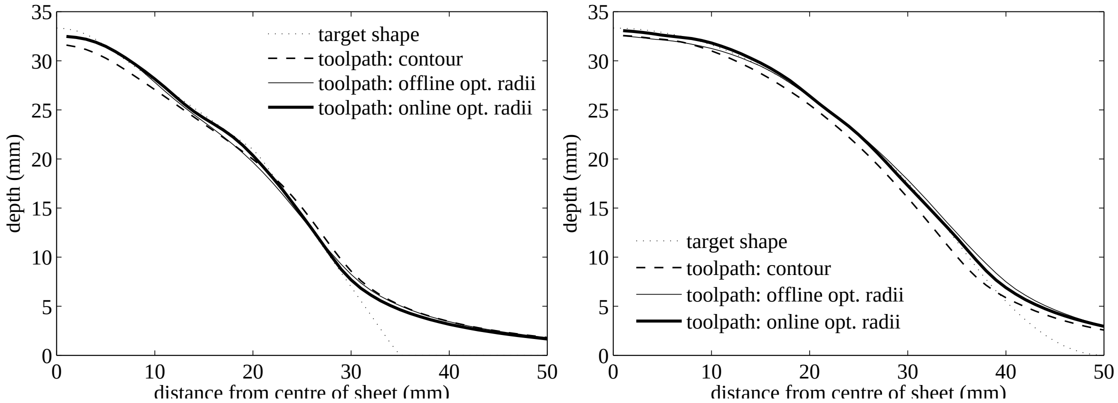
Figure 5.24: The test shapes used for path optimistion with feedback.

tool-paths are displayed in figure 5.26. The actual tool-paths followed during the forming process are plotted in black and the gray plots in figure 5.26 show the individual optimised tool-paths for each layer. For the off-line path optimisation scheme, the tool-path appears very smooth because the smoothness factor ρ in the control objective function in (5.1) was set relatively high, typically to one or two. When lower values of ρ were tested the resulting jittery tool-paths occasionally caused material failure but for the online path optimisation scheme, factors of $\rho = 0.1$ and lower were used for all of the optimisation steps in all four test parts and no tearing was observed.

The error measures corresponding to the plots in figure 5.25 are displayed in table 5.8, where the percentage benefit is particularly of interest because it shows clearly that for all four shapes, the use of feedback in the path optimisation scheme significantly reduced the RMS error when measured out to 45 mm, 40 mm, 35 mm, and 30 mm from the centre of the sheet. In some cases, the benefit from the on-line scheme is about 80% over the contour following tool-path and over 60% if compared to the off-line optimised tool-path.



(a) For shape a) in figure 5.24 (b) For shape b) in figure 5.24



(c) For shape c) in figure 5.24 (d) For shape d) in figure 5.24

Figure 5.25: Deformation resulting from the contour tool-paths, the off-line tool-paths and the on-line optimised tool-paths.

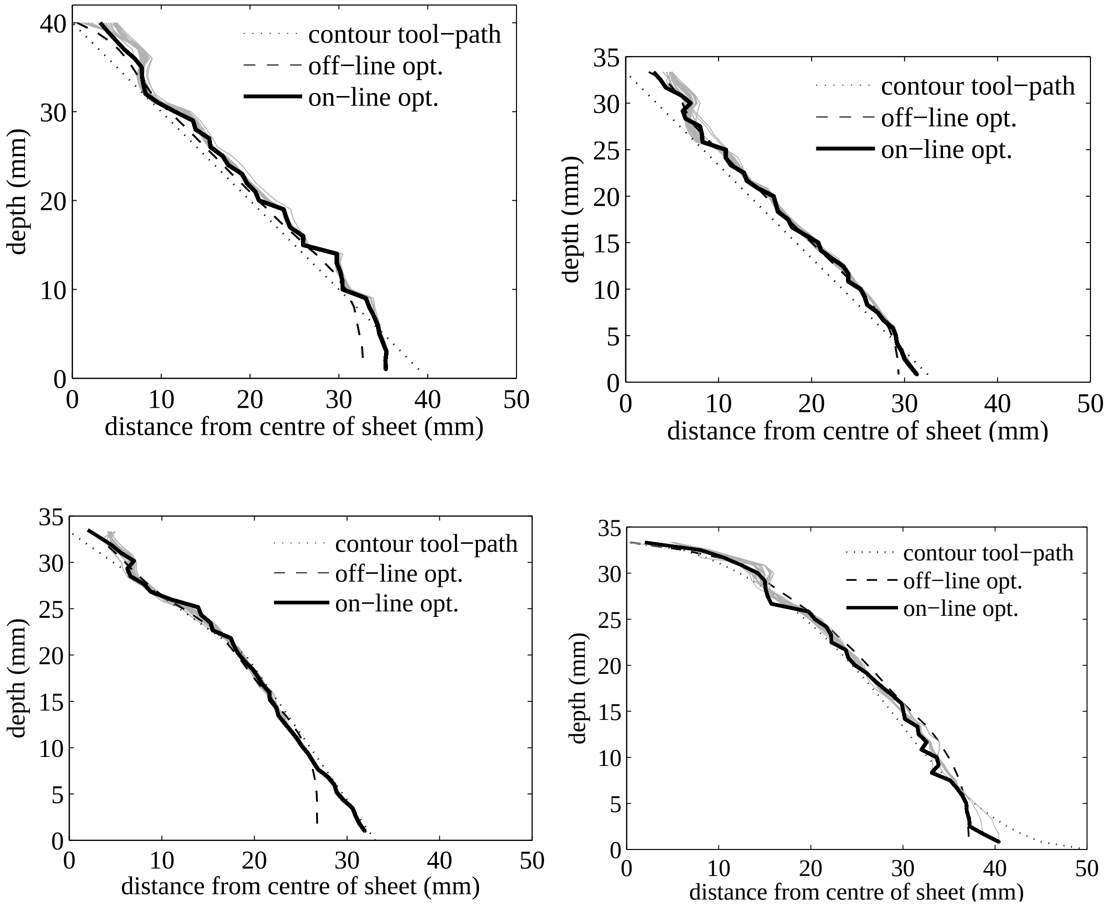


Figure 5.26: The tool-paths used to generate the profiles in figure 5.25. The gray plots show the re-optimised toolpaths at each step.

	Shape	**Toolpath**	**RMS** (mm)	**Var** (mm^2)	**Max. diff**	**Benefit**
up to 45 mm	a)	Contour	2.1442	2.3172	3.1965	-
		Off-line	1.20	1.38	3.52	44.1%
		On-line	1.29	1.40	4.21	39.6%
	b)	Contour	2.23	4.73	3.68	-
		Off-line	1.73	2.18	4.08	22.4%
		On-line	1.54	1.98	3.71	37.5%
	c)	Contour	2.2360	4.4028	5.0987	-
		Off-line	2.10	3.65	5.05	6.1%
		On-line	1.86	2.61	4.64	16.6%
	d)	Contour	1.13	0.89	2.36	-
		Off-line	1.13	1.15	3.18	0.1%
		On-line	0.88	0.66	2.91	22.3%
up to 40 mm	a)	Contour	2.1010	0.3894	2.5290	-
		Off-line	0.80	0.18	1.14	62.0%
		On-line	0.60	0.33	1.93	71.6%
	b)	Contour	2.14	3.67	3.68	-
		Off-line	1.48	1.80	4.08	30.8%
		On-line	1.34	1.66	3.71	37.5%
	c)	Contour	2.2360	4.4028	5.0987	-
		Off-line	2.10	3.65	5.05	7.3
		On-line	1.86	2.61	4.64	18.6
	d)	Contour	1.03	0.19	1.66	-
		Off-line	0.70	0.50	2.00	31.9%
		On-line	0.40	0.15	1.39	61.5%
up to 35 mm	a)	Contour	2.2206	0.0200	2.5291	-
		Off-line	0.82	0.03	1.14	62.9%
		On-line	0.36	0.14	0.96	83.6%
	b)	Contour	1.91	1.08	2.27	-
		Off-line	0.71	0.47	2.55	62.5%
		On-line	0.65	0.43	2.23	66.1%
	c)	Contour	1.7143	3.0162	5.0987	-
		Off-line	1.49	2.24	5.05	13.0%
		On-line	1.24	1.46	4.64	27.7%
	d)	Contour	1.06	0.13	1.66	-
		Off-line	0.44	0.19	0.84	58.2%
		On-line	0.21	0.05	0.35	79.9%
up to 30 mm	a)	Contour	2.2388	0.0160	2.5291	-
		Off-line	0.85	0.025	1.14	61.9%
		On-line	0.29	0.085	0.69	86.9%
	b	Contour	2.01	0.07	2.27	-
		Off-line	0.38	0.15	0.94	80.9%
		On-line	0.41	0.13	0.92	79.4%
	c)	Contour	1.0797	0.9705	1.6633	-
		Off-line	0.68	0.37	1.32	37.3%
		On-line	0.35	0.12	0.78	67.1%
	d)	Contour	0.93	0.077	1.51	-
		Off-line	0.41	0.13	0.84	55.6%
		On-line	0.21	0.05	0.31	77.1%

Table 5.8: Error measures for four test shapes of increasing complexity. Benefit is defined as the improvement in RMS error over the contour tool-path.

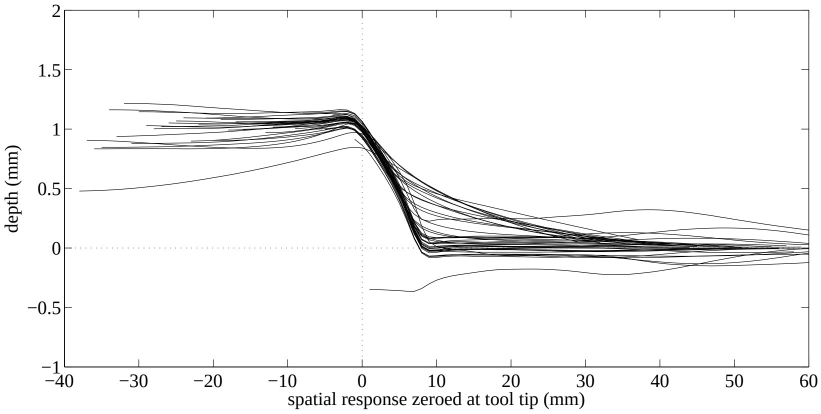
5.3.3 Summary, review and future experiments

A new method for tool-path planning in ISF based on optimal control has been proposed and it has successfully been demonstrated to improve the fit between the part geometry and target geometry for axisymmetric products. All together, four products were produced using the new method and they all used the same impulse response data for the model in the optimal control routine. As such, the results confirm that the method remains robust as the target geometry deviates from the original shape that the impulse responses were generated from. The impulse response model for prediction is particularly novel since it avoids the need for solving a highly non-linear material deformation model.

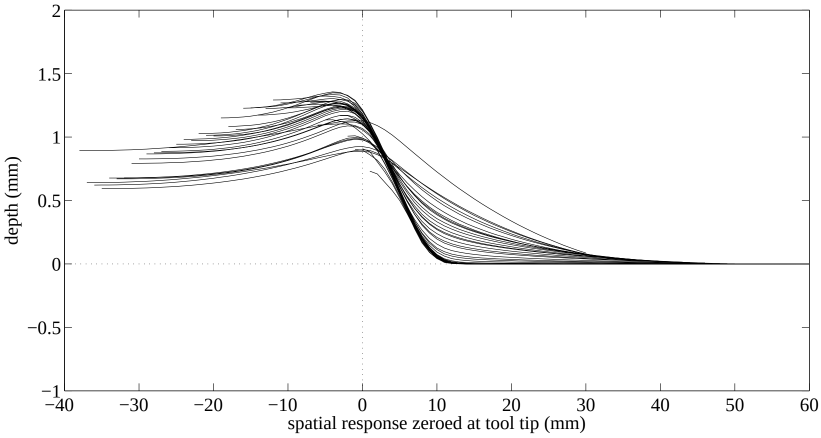
Further tests using the method described here are now underway in Cambridge to establish a clear relationship for how the degree of compensation varies as the impulse response data degrades. This study would be of interest to practitioners because low quality data is generally quick and easy to generate, so an indicator of the sacrifice in geometry that results from using such data in the on-line optimal path planning scheme is useful.

One example of data degradation has already been tested here by using the impulses generated for a cone to produce several alternative geometries, since better quality impulse response data would have been obtained if they were generated directly for the alternative geometries in question, but to do this would have been more time consuming. However, such tests based on better quality data are currently underway as well as another set of tests based on impulse response data generated from a single LS-Dyna FE simulation and from the rigid plastic model in §3.6. For the latter two, the impulse response data would be expected to be of a lower quality than experimental data but it is also more convenient to generate. The simulations for a cone shape part have been generated and the related impulse responses have been produced, as displayed in figure 5.27. The impulse responses from the LS-Dyna plot show more variation than the ones from the one cone model, although both of them are much more uniform than the original experimental impulse responses in figure 5.3, so it will be interesting to see how they perform.

A number of ISF researchers have been developing methods for improving geometric accuracy in ISF, but this result is particularly important because it is the first complete



(a) Impulse responses generated from LS-Dyna simulation of a cone shaped part formed by SPIF with a 45^o wall angle.



(b) Impulse responses generated using the one cone rigid plastic limit analysis model, with no thickness variation or work hardening.

Figure 5.27: Impulse response shapes obtained from simulation

framework for path planning based on on-line feedback. The approaches taken by other researchers include Otegi *et al*'s [89] method introduced in §3.9 for reducing the 'dip' during ISF and a novel laser assisted SPIF technique developed by Duflou *el al* [33, 34], which results in deformation that is more localised to the tool and in turn, improved part geometries. The laser assisted method adds more complexity to the equipment design, but the results are promising when evaluated for part geometries similar to those used here. A patent for the process is being filed by Duflou *el al.* One of the strengths of the impulse response method is that it can be used in conjunction with the above techniques, simply by generating an appropriate set of impulse response shapes.

The work of Ambrogio *et al* described in [11] was published in 2005 but it was only discovered by the author recently, after the work in this chapter was carried out. It is worth discussing because, interestingly, it displays some similarity to the work described here, although there are significant differences. In [11], a truncated pyramid is produced according to a set of inputs, which would be defined as $\mathbf{p}_i = [r_i \;\; \Delta z_i]$ using the notation of this chapter, where r_i is the edge-length of a square tool-path at stage i and Δz_i is the change in depth from the previous stage. These inputs follow the contour of a target shape and they are variable, because this target shape is adaptively modified at each layer according to a simple correction rule based on the assumed deformation coordinates $(x_i^{\mathrm{i}},\;z_i^{\mathrm{i}})$ of the ideal intermediate shape of the sheet at stage i, defined along a line drawn from the centre to the edge of a the sheet. For some stage i, the target modification scheme also depends on the actual coordinates of the shape obtained at the previous stage, $(x_{i-1}^{\mathrm{a}},\;z_{i-1}^{\mathrm{a}})$, which are measured at the beginning of stage i using a manually operated probe. The correction rules are

$$\begin{aligned} x_i^* &= x_i^{\mathrm{i}} + \left\| \begin{bmatrix} x_{i-1}^{\mathrm{i}} - x_{i-1}^{\mathrm{a}} \\ z_{i-1}^{\mathrm{i}} - z_{i-1}^{\mathrm{a}} \end{bmatrix} \right\|_2 , \\ z_i^* &= z_i^{\mathrm{i}} + (z_{i-1}^{\mathrm{i}} - z_{i-1}^{\mathrm{a}}), \qquad\qquad i = 1, \ldots, M, \end{aligned} \tag{5.24}$$

from which the next input $\mathbf{p}_i$ is defined from the contour following tool-path related to the modified target shape $(x_i^*,\;z_i^*)$. This scheme was repeated until all layers were formed and the final shape was qualitatively compared to that of the contour following tool-path. For the truncated pyramid test part used, some improvement in geometry was observed, although as explained in [11] the method is limited because:

1. it only uses information from the previous stage to modify the current one
2. the rules in (5.24) are intuitive and should be refined
3. the general applicability of the method has not been demonstrated

Due to these problems, Ambrogio *et al* appear to have changed their focus to preventing part failure using online force measurements [10] and applying statistical analysis to experimental data from SPIF [9]. The aim of the latter is to produce a design tool that can be used to set some parameters related to the tool trajectory so as to minimise the 'dip' and springback when forming truncated pyramid shaped parts. The impulse response method described here improves upon point 1 above by using information for the complete process at each optimisation stage. The main limitation with points 2 and 3 is the need to set the value of $(x_i^{\mathrm{i}},\ z_i^{\mathrm{i}})$, which appears to be done by educated guesswork and is therefore only likely to work for simple geometries. Point 2 is addressed by this work, since the optimal control based path optimisation rule is a refinement over using the simple update rule used by Ambrogio *et al.* The issue of the applicability of this method, with regards to point 3, is also avoided by verifying the method for a number of shapes and by using quantitative error measures rather than the qualitative comparisons made in [11].

Chapter 6

Conclusions

6.1 Concluding remarks

This thesis reports the work conducted during a research project aimed at reducing geometrical errors in incremental sheet forming (ISF). In chapter 1 several variants of the process were introduced including single point incremental forming (SPIF), which was the main process of interest here. This was followed by a general summary of the current research focus in the ISF community and of the problems addressed in this thesis. In particular, two problems were identified for which it was considered that further research would benefit the ISF research community: the numerical modelling of deformation and the improvement of part geometry in SPIF. The contributions made towards solving these problems in this thesis were:

- an investigation of the applicability of the rigid plastic assumption in modelling ISF
- a structured method for on-line and off-line tool-path design for axisymmetric parts so as to reduce geometrical errors

These aspects of the work are summarized in the subsections below before discussing the link between them.

6.1.1 Numerical modelling summary

In terms of numerical modelling, a number of researchers have taken the approach of modifying existing elasto-plastic FE codes to improve efficiency, for instance by using adaptive remeshing [68], altering the incrementation method for elasto-plastic strain [104] or by using domain decomposition [44]. However, the speed improvements obtained using these approaches are not sufficient for the tool-path optimisation problem, which is why the more drastic approximation of rigid plastic material deformation was investigated here. Three models that use this assumption were described in chapters 2, 3 and 4 respectively.

In chapter 2, the first attempt was made at using the rigid plastic assumption by developing a membrane model based on small strain deformation theory. The model was discretised using finite differences (FDs) and was based on a constrained energy minimisation problem that was initially solved using an algorithm from the Matlab Optimisation Toolbox. This was found to be very inefficient, so the model was then rearranged into a second-order cone program (SOCP), which was quicker to solve. Simulations of a 'line test' were performed using this model, where the tool tracks a straight line over the sheet at a constant depth.

The FD model presented a number of disadvantages, such as the need for small time increments to avoid numerical instability and for an alternative set of equations to describe the initial indentation of the sheet. A new model was developed in chapter 3 that removed these disadvantages. The model drew on a number of recent developments in the field of limit analysis, where the rigid plastic assumption is commonly used. It was found that finite element (FE) implementations of one step problems in limit analysis such as the initial deformation of a flat sheet subjected to a point load, referred to as the plate problem, have previously been solved as SOCPs. A complete shell model for ISF was produced here by extending current one-step SOCP problems to time evolving problems by using sequential limit analysis theory. Ilyushin theory was used to combine the bending energy and membrane energy over the sheet to form a shell problem. This was then arranged as an SOCP in two different ways, which were referred to as the 'two cone' (§3.5) and the 'one cone' (§3.6) models respectively. The two cone model exactly describes the Ilyushin yield surface whereas the one cone model approximates it to a single 6D ellipsoid. By comparing line tests from both models (§3.8.1) it was shown that this approximation makes little difference in terms of predicting deformation.

The 'one cone' model is more efficient than the 'two cone' model because it uses fewer conic constraints in the SOCP problem, so it was used for a number of further validation tests against the benchmark experimental and LS-Dyna data in §3.8. The computation times and errors for a number of mesh densities were reported and generally a good fit between the data and the model was observed for regular meshes, irregular meshes and also an adaptive meshing strategy. A stretch spectrogram of the model was compared to one from LS-Dyna, from which it was confirmed that the deformation mechanism in both models is that of plane strain. Although the rigid plastic assumption is typically used only in large deformation processes, ISF actually involves the accumulation of a large number of small deformation steps, yet the results here show that the rigid plastic assumption is suitable for an approximate model of ISF.

By analysing intermediate stages of the process, a pronounced 'dip' was observed in central unformed regions of the sheet that was far shallower in the actual process. This led to a review of existing literature on the 'dip' and also of the assumptions made in the model in terms of the tool-sheet contact state, geometry and material description that could cause the 'dip'. For each of these respectively, an improvement to the model was made by incorporating a sliding tool-sheet contact state (§3.10), in-process thickness variations (§3.11.1) and work hardening (§3.11.2). The latter two modifications to the model significantly reduced the 'dip' and so were identified as the most important factors.

The rigid plastic limit analysis model in chapter 3 is significantly faster than existing elasto-plastic models for ISF, but at 20-30 seconds per time step it is still not fast enough to use within a tool-path optimisation routine. However, because it uses a large number of variables, a general conclusion on the speed improvements that can be obtained by using the rigid plastic assumption cannot be made from this model. In order to do this, the rigid plastic assumption had to be applied to a model with far fewer variables and this was done in chapter 4, where a reduced 'grid' model was developed that approximated the sheet to a network of rods. Despite the lack of thickness variation or work-hardening, no 'dip' was observed in the grid model due to the side-effect of an approximation used to formulate the SOCP problem, where inequality constraints were used for some constraints that strictly should have been equalities. An inertia-like term was introduced to reduce the effect of this approximation and a good fit was observed between the model and the benchmark data.

In terms of computation speed, it can be concluded from this model that for reasonable mesh sizes, a solution time of approximately 5-10 seconds per time-step can be achieved on standard desktop PC using the rigid plastic assumption. Due to its speed, this model was used to simulate a two point incremental forming process (TPIF) for an industrial automotive product.

A complete tool-path can often take more than one thousand time-steps and since the model would have to be run many times in a path optimistion routine, it is concluded that the rigid plastic assumption cannot currently be used for the purposes of path optimisation. There have been suggestions of other applications where such models can be useful, for instance for a rapid analysis of tool-paths where detailed elasto-plastic simulations can be an over-kill. The main outcomes of the numerical modelling component of this research can be summarised as follows:

- the rigid plastic assumption has been evaluated in modelling ISF
- an approximate numerical model for ISF based on this assumption can be used to predict deformation
- a new application of SOCP in solving shell problems has been demonstrated using sequential limit anlaysis
- thickness variations and work-hardening must be accounted for to accurately predict deformation at intermediate periods of the process and reduce the ‘dip’.
- solutions times within 10 s per time-step can be achieved on current desktop PCs

6.1.2 Path optimisation summary

Since there are no alternative efficient numerical models for the path optimisation problem, in chapter 5 an experimental model was developed for axisymmetric products. The model was generated as a shifted summation of experimental spatial impulse responses and then used in a path optimisation problem based on optimal control. In doing so, a new method for reducing geometric errors in SPIF was demonstrated and a number of results, as well as some interesting points, were discovered along the way.

The path optimisation scheme was used to compute off-line optimal trajectories for tool-path radii and these radii performed well in reducing geometric errors for several test shapes. The optimsation routine was set up as a non-convex optimisation problem that was solved using a gradient based algorithm with a good initial guess, which was typically chosen as the contour following tool-path. It was shown that the impulse response model used here is not suitable for optimising the indentation depth at each layer in the tool-path, so this was fixed. A number of extensions to the scheme were tested, by using a backing plate, a multipass strategy and adaptive shifting of the target shape and in all cases, the off-line optimal tool-path performed better than the contour following tool-path. Furthermore, by using impulse response data generated by forming a simple cone shaped part, it was shown that impulse responses generated from a single shape could be used to optimise the tool-path for several alternative target shapes.

An on-line scheme that uses feedback from a stereo camera was also implemented and again the same impulse response data was used to optimise the tool-path for several shapes. The on-line scheme was found to perform better than both the contour following tool-path and the off-line implementation and in some cases, the RMS error was reduced by up to 80% over the contour following tool-path. Since very little is currently known in terms of tool-path design in ISF, this work is particularly useful and a number of ongoing tests using the proposed method are currently underway since, unlike trial and error or heuristic rule based methods, this is the first structured use of automatic feedback control for improving geometry in ISF.

6.1.3 Discussion

The initial period of work for this DPhil was devoted to numerical modelling because it was believed at the time that a simpler model would not sufficiently be able to predict the complex behaviour of the sheet in ISF to demonstrate the benefit of path-optimisation. However, later on the concept of an experimental impulse response model came about by focusing on axisymmetric shapes and with this, it was proved that an automatic control scheme for path design in ISF can be implemented.

The consequent separation between the numerical modelling and control aspects of the work described in this thesis could give the impression that they are distinct and unrelated pieces of work. It is therefore emphasised here that they are linked by the fact that they are the two key components required for a solution to a single problem; the geometric accuracy problem in general ISF. The first component shows that it is possible to create simplified numerical models that describe the deformation behaviour in ISF, whereas with the latter a method for tool-path optimisation is developed, which provides results that show for the first time that automatic feedback control can be used to reduce geometric errors in ISF. In the long term when the scope of this work extends to the production of geometrically accurate general shapes, these two components will converge to provide a useful solution for practitioners. Specifically, as and when computation speeds improve to a rate that permits the online execution of the numerical models, the initial step in the extension to general shapes will be to replace the impulse response model in the path optimisation algorithm with the numerical model in either chapter 3 or 4, hence merging this apparently distinct component with chapter 5 to form a single product.

Finally, it is also noted that in the short term a more immediate link can be formed that connects these pieces of work by verifying the experimental impulse responses against those generated from the numerical models. As suggested in chapter 5 this should be done as future work since this would also provide an additional result of avoiding the need for an experimental model, therefore demonstrating an extension to the path design scheme for axisymmetric shapes that is based only on first principle physical modelling of ISF.

6.2 Future directions for research

Several areas of this thesis may be extended in future projects and possible areas for further research have been noted periodically in the thesis. The limit analysis model in chapter 3 has only been tested with one type of shell finite element, but many elements exist in literature that could be implemented in the model and their performance analysed. It was noted in §3.9 that several methods exist for combining simplified elasticity descriptions with the rigid plastic assumption, which could be incorporated with the limit analysis based model described here. Further to the linear work-hardening simulations conducted here, non-linear work-hardening material descriptions should be tested. Also, an explanation is needed for why the simulations with thickness variation in chapter 3 predict sheet failure earlier than found in practice. It would then be possible to compare the final sheet thicknesses with LS-Dyna or experimental data. The TPIF simulations performed with the grid model in chapter 4 should be compared to experimental data. It is worth considering relaxing some of the constraints in the grid model, so long as doing so does not cause further computational burden. For instance, the nodes in the grid model are currently restricted to move vertically, so one could consider reformulating the model to allow horizontal movement too.

The SOCP solver used for the numerical models is based on an interior-point algorithm that, unlike gradient based algorithms, does not permit initial guesses for the value of the optimisation variables. Efficient SOCP algorithms that exploit the ability to use ‘warm starts’ are in development [117], but they are not currently available in commercial codes. For the ISF models described in this thesis the deformation field changes very little from one time-step to the next, so the previous deformation could be used as an initial guess for the current deformation. If the appropriate SOCP algorithms become available it would be interesting to test their performance on the models in this thesis. Similarly, the problems in this thesis were solved on a single PC, but in [13] the parallel processing capability of interior-point algorithms is discussed and could be tested on the ISF models described here. Also, rather than using Matlab a compiled script such as C would be more efficient.

There are many possibilities for further work on the path optimisation problem in chapter 5. The impulse response shapes used here were generated from experiments, so this can be compared to impulses generated from numerical simulations. Indeed, the numerical models in this thesis can be used to verify the impulse responses and to generate new definitions of process impulse responses. For instance, with the impulse responses defined here it was found that the radii of the tool-path at each layer could be optimised, but with further work it is likely that it will be possible to produce a set of impulse responses that can be used to optimise the tool application depth at each layer too. It is suggested that the initial step towards this should be to conduct a study of the variation of impulse response shapes to small changes in the application of the tool, with reference to a control shape, and this can be done either experimentally or perhaps more efficiently and less wastefully, by using the models in this thesis.

In the on-line scheme, the first control input was applied at each step before re-optimising the tool-path, but with further tests it may be found that adequate compensation is achieved by re-optimisating the tool-path less frequently. As an analysis of the robustness of the scheme, a relationship for the part accuracy against frequency of tool-path re-optimisation can be established. A related robustness test suggested in §5.3.3 is to establish a relationship for the degree of compensation against the quality and type of impulse response data generated. Such experimental robustness studies would be neatly complimented with theoretical studies using a host of tools used in control engineering for robust control. These methods show how to systematically introduce errors into the model that can be parametrised so that their effect on the tool-path can be analysed. For instance, with the model in (5.19) one could study the effect of a bounded error $\mathbf{e}_i$ that is parametrised by imposing the constraint $\|\mathbf{e}_i\| \leq e_{max}$, where e_{max} can be varied. The model in (5.19) would become

$$\mathbf{x}_M = \sum_{i=1}^{M} (\mathbf{w}_i + \mathbf{e}_i) + \mathbf{s}_f, \tag{6.1}$$

which could be used to set up the min-max problem of finding the tool-path that minimises the control objective whilst maximising the constrained model error $\mathbf{e}_i$. Alternative optimisation algorithms can be explored for solving this and indeed the path optimisation problem itself that suit the structure of these problems.

Also, since only axisymmetric products were considered here this research should be extended to asymmetric parts too and as stated previously, in order to do this it is expected that the models described in chapters 3 or 4 will be combined with the path optimisation routine as future work. Another option that could be studied is to attempt to develop a systematic method for defining the ‘spatial impulse response’ for arbitrary shapes, which can be used to predict the deformation behaviour in a suitable manner for more general classes of shapes than axisymmetric ones.

Many further possibilities exist for following on from the work in this thesis and it will be interesting to see how these develop over time, which approaches succeed or fail and what other surprises appear from further research.

Bibliography

[1] Irvan Smith Inc. http://www.irvansmith.com/.

[2] Abaqus. *Abaqus/CAE Version 6.7 User's Manual.* Dassault Systemes, 2007.

[3] D. J. Allman. A compatible triangular element including vertex rotations for plane elasticity analysis. *Computers and Structures*, 19:1–8, 1984.

[4] J. Allwood, O. Music, A. Raithatha, and S. R. Duncan. Closed-loop feedback control of product properties in flexible metal forming processes with mobile tools. *CIRP Annals - Manufacturing Technology*, 58(1):287–290, 2009.

[5] J. M. Allwood, N. E. Houghton, and K. P. Jackson. The design of an incremental sheet forming machine. In *Proc. SheMet '05 International Conference on Sheet Metal*, pages 471–478, Erlangen, Germany, 2005.

[6] J. M. Allwood, G. P. F. King, and J. Duflou. A structured search for applications of the incremental sheet-forming process by product segmentation. *Proc. IMechE Part B: J. Engineering Manufacture*, 219:239–244, 2005.

[7] J. M. Allwood and D. R. Shouler. Paddle forming: a novel class of sheet metal forming processes. *CIRP Annals - Manufacturing Technology*, 56(1):257–260, 2007.

[8] J. M. Allwood, D. R. Shouler, and A. E. Tekkaya. The increased forming limits of incremental sheet forming processes. In *Proc. SheMet '07 International Conference on Sheet Metal*, Palermo, Italy, 2007.

[9] G. Ambrigio, V. Cozza, L. Filice, and F. Micari. An analytical model for improving precision single point incremental forming. *J. Materials Processing Technology*, 191:92–95, 2007.

[10] G. Ambrigio, L. Filice, and F. Micari. A force measuring based strategy for failure prevention in incremental forming. *J. Materials Processing Technology*, 177:413–416, 2006.

[11] G. Ambrogio, L. Filice, L. De Napoli, and M. Muzzupappa. A simple approach for reducing profile diverting in a single point incremental forming process. *Proc. IMechE Part B: J. Engineering Manufacture*, 219:823–830, 2005.

[12] E. Anderheggen and H. Knopfel. Finite element analysis using linear programming. *Int. J. Solids and Structures*, 8:1413–1431, 1972.

[13] E. Andersen and K. Andersen. Exploiting parallel hardware in solving optimization problems. *SIAM News*, 32(4), 1999.

[14] E. D. Anderson and K. D. Anderson. *The MOSEK interior point optimizer for linear programming: an implementation of the homogenous algorithm*, chapter 8, pages 197–232. Kulwer Academic Publishers, 2000.

[15] K. D. Anderson, E. Christiansen, and M. Overton. Computing limit loads by minimising the sum of norms. *SIAM J. Scientific Computing*, 19:3, 1998.

[16] H. Arai. Robotic metal spinning - shear spinning using force feedback control. In *Proc. of 2003 IEEE Int. Conf. Robotics & Automation*, pages 3977–3983, Taipei, Taiwan, 2003.

[17] H. Arai. Robotic metal spinning - forming non-axisymmetric products using force control. In *Proc. of 2005 IEEE Int. Conf. Robotics & Automation*, pages 2691–2696, Barcelona, Spain, 2005.

[18] A. Attanasio, E. Ceretti, and C. Giardini. Optimization of tool path in two points incremental forming. *J. Materials Processing Technology*, 177:409–412, 2006.

[19] M. Bambach, M. Cannamela, M. Azaouzi, G. Hirt, and J. L. Batoz. Computer-aided tool path optimization for single point incremental sheet forming. *Advanced Methods in Material Forming*, pages 233–250, 2007.

[20] M. Bambach, G. Hirt, and J. Ames. Quantitative validation of FEM simulations for incremental sheet forming using optical deformation measurements. In *Proceeding of the 11th International Conference on Sheet Metal Erlangen, Germany*, 2005.

[21] M. Bambach, B. Taleb-Araghi, G. Hirt, and W. Reimers. Error control in explicit finite element simulations of incremental sheet metal forming. In Pavel Hora, editor, *Numisheet 2008 - Proc. 7th International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes*, pages 715–720, Interlaken, Switzerland, 2008.

[22] J. T. Betts. *Practial Methods for Optimal Control using Nonlinear Programming.* SIAM, Philadelphia, PA, 2001.

[23] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM, Philadelphia, PA, 1994.

[24] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, UK, 2004.

[25] A. E. Bryson and Y-C Ho. *Applied Optimal Control.* Taylor and Francis, Levittown, PA, 1975.

[26] C. J. Burgoyne and M. G. Brennan. Exact Ilyushin yield surface. *Int. J. Solids and Structures*, 30:1113–1131, 1993.

[27] J. Cao, Y. Huang, N V. Reddy, R. Malhotra, and Y. Wang. Incremental sheet metal forming: Advances and challenges. In *Proceeding of 9th Int. Conf. Technology of Plasticity*, pages 1967–1982, Seoul, Korea, 2008.

[28] E Ceretti, C. Giardini, and A. Attanasio. Experimental and simulative results in sheet incremental forming on CNC machines. *J. Materials Processing Technology*, pages 176–184, 2004.

[29] E. Christiansen. Limit analysis for plastic plates. *Siam. J. Mathematical Analysis*, 11:514–522, 1980.

[30] R. D Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt. *Concepts and Applications of Finite Element Analysis, 4th edition.* John Wiley & Sons, 2002.

[31] Livermore Software Technology Corp. *LS-Dyna Theory Manual, ISBN 0-9788540-0-0.* 2006.

[32] S. Dejardin, J.C. Gelin, and S. Thibaud. A fe analysis of springback for improving precision in single point incremental sheet forming process. In Pavel Hora, editor, *Numisheet 2008 - Proc. 7th International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes*, pages 721–725, Interlaken, Switzerland, 2008.

[33] J. R. Duflou, B. Callebaut, J. Verbert, and H. De Baerdemaeker. Laser assisted incremental forming: Formability and accuracy improvement. *CIRP Annals - Manufacturing Technology*, 56(1):273–276, 2007.

[34] J. R. Duflou, B. Callebaut, J. Verbert, and H. De Baerdemaeker. Improved SPIF performance through dynamic local heating. *Int. J. Machine Tools and Manufacture*, 48(5):543–549, 2008.

[35] S. R. Duncan, J. M. Allwood, and S.S. Garimella. The analysis and design of spatial control systems in strip metal rolling. *IEEE Trans. Control Systems Technology*, 6(2):220–232, 1998.

[36] S. R. Duncan, P. Jones, and P. Wellstead. A frequency-domain approach to determining the path separation for spray coating. *IEEE Trans. Automation Science and Engineering*, 2(3):233–239, 2005.

[37] F. Dunne and N. Petrinic. *Introduction to Computational Plasticity.* Oxford University Press, 2005.

[38] S. P. Edwardson, E. Abed, P. French, G. Dearden, K.G. Watkins, A.J. Moore, R. McBride, D.P. Hand, and J.D.C. Jones. Iterative 3d laser forming of continuous surfaces. In *Proc. 23rd Int. Congress on Applications of Lasers and Electro-Optics, San Francisco, USA*, 2004.

[39] B. Endelt and J. Dankert. Optimal draw-in and feedback control using distributed blank-holder force. In Pavel Hora, editor, *Numisheet 2008 - Proc. 7th International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes*, pages 877–882, Interlaken, Switzerland, 2008.

[40] C. A. Felippa. Web-posted lectures on advanced finite element methods, at http://caswww.colorado.edu/courses.d/afem.d/home.html.

[41] L. Filice, G. Ambrogio, and F. Micari. On-line control of single point incremental forming operations through punch force monitoring. *CIRP Annals - Manufacturing Technology*, 55:245–248, 2006.

[42] L. Filice, L. Fratini, and F. Micari. Analysis of material formability in incremental forming. *CIRP Annals - Manufacturing Technology*, 51:199–202, 2002.

[43] W. Flügge. *Stresses in Shells. 2nd ed.* Springer-Verlag, Berlin, Germany, 1973.

[44] A. Hadoush and A. H. van den Boogaard. Time reduction in implicit single point incremental forming simulation by domain decomposition. In Pavel Hora, editor, *Numisheet 2008 - Proc. 7th International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes*, pages 411–414, Interlaken, Switzerland, 2008.

[45] E. Hagan and J. Jeswiet. A review of conventional and modern single-point sheet metal forming methods. *Proc. IMechE Part B: J. Engineering Manufacture*, 217:213–225, 2003.

[46] G. Hirt, J. Ames, and M. Bambach. Economical and ecological benefits of CNC incremental sheet forming (ISF). In *Proc. ICME*, Miskolc, Hungary, 2003.

[47] G. Hirt and M. Bambach. Modelling incremental sheet forming using a meshless surface representation based on radial basis functions. In *Proc. 8th ICTP*, pages 261–262, Verona, Italy, 2005.

[48] G. Hirt, S. Junk, M. Bambach, and I. Chouvalova. Process limits and material behaviour in incremental sheet forming with CNC-tools. *Materials Science Forum*, 426–432(5):3825–3830, 2003.

[49] P. G. Hodge and T. Belytschko. Numerical methods for the limit analysis of plates. *J. Applied Mechanics, Transactions of the ASME*, pages 796–802, 1968.

[50] P. Hora, W. Volk, K. Roll, B. Griesbach, L. Kessler, and W. Holtz, editors. *Numisheet 2008 Part B: The Numisheet 2008 Benchmark Study.* ETH Zurich, 2008.

[51] G. Houssain and L. Gao. Fundamental studies on incremental forming of titanium sheet-metal. In *Proc. 2006 ASME Int. Conf. on Manufacturing Science and Engineering (MSEC2006)*, number 10, Ypsilanti, MI, 2006.

[52] C.-L. Hwan. Plane strain extrusion by sequential limit analysis. *Int. J. Mechanical Sciences*, 39(7):807–817, 1997.

[53] C. L. Hwan. An upper bound finite element procedure for solving large plane strain deformation. *Int. J. Numerical Methods in Engineering*, 40:1909–1922, 1997.

[54] A. A. Ilyushin. Plasticité. *(in French) Editions Eyrolles, Paris*, 1956.

[55] H. Iseki. An approximate deformation analysis and FEM analysis for the incremental bulging of sheet metal using a spherical roller. *J. Materials Processing Technology*, 111:150–154, 2001.

[56] H. Iseki. Flexible and incremental bulging of sheet metal using high-speed water jet. *JSME International Journal, Series C*, 44(2):486–493, 2001.

[57] K. Jackson. 1st year PhD report: Asymetric incremental sheet forming. Master's thesis, University of Cambridge, Cambridge, UK, 2005.

[58] K. Jackson. *The Mechanics of Incremental Sheet Forming.* PhD thesis, University of Cambridge, 2008.

[59] K. P. Jackson, J. M. Allwood, and M. Landert. Incremental forming of sandwich panels. In *SheMet'07 Int. Conf. on Sheet Metal*, pages 591–598, Palermo, Sicily, 2007.

[60] J. Jeswiet, F. Micari, G. Hirt, A. Bramley, J. Duflou, and J. Allwood. Asymmetric single point incremental forming of sheet metal. *CIRP Annals - Manufacturing Technology*, 54(2):623–650, 2005.

[61] M. Jirásek and Z. P. Bažant. *Inelastic Analysis of Structures.* John Wiley and Sons, 2002.

[62] M. Junkar, K. C. Heiniger, and B. Jurisevic. The application of water-jet technology for incremental sheet-metal forming. *Strojniski-Vestnik*, 50(12):613–622, 2004.

[63] K. P. Kim and H. Huh. Dynamic limit analysis formulation for impact simulation of structural members. *Int. J. Solids and Structures*, 43:6488–6501, 2006.

[64] Y. H. Kim and J. J. Park. Effect of process parameters on formability in incremental forming of sheet metal. *J. Materials Processing Technology*, 130-1:42–46, 2002.

[65] K. Kitazawa. Incremental sheet metal stretch-expanding with CNC machine tools. In *Proc. 4th Int. Conf. in Technology of Plasticity*, pages 1899–1904, Beijing, China, 1993.

[66] K. Lange. *Handbook of Metal Forming.* SME Publications, 1985.

[67] R. C. K. Lee. *Optimal Estimation, Identification and Control.* The M.I.T. Press, Cambridge, MA, 1964.

[68] C. Lequesne, C. Henrard, C. Bouffioux, J. R. Duflou, and A. M. Habraken. Adaptive remeshing for incremental forming simulation. In Pavel Hora, editor, *Numisheet 2008 - Proc. 7th International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes*, pages 399–403, Interlaken, Switzerland, 2008.

[69] E. Leszak. Apparatus and process for incremental dieless forming. *United States Patent 3342051.* 1967.

[70] S.-Y. Leu. Convergence analysis and validation of sequential limit analysis of plane-strain problems of the von mises model with non-linear isotropic hardening. *Int. J. Numerical Methods in Engineering*, 64:322–334, 2005.

[71] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.

[72] C. A. Luttgeharm. Numerically controlled forming method. *United States Patent 6532786.* 2003.

[73] A. Makrodimopoulos and C. M. Martin. Lower bound limit analysis of cohesive-frictional materials using second-order cone programming. *Int. J. Numerical Methods in Engineering*, 66(4):604–634, 2005.

[74] A. Makrodimopoulos and C. M. Martin. Upper bound limit analysis using simplex strain elements and second-order cone programming. *Int. J. Numerical and Analytical Methods in Geomechanics*, 31:835–865, 2007.

[75] S. Matsubara. A computer numerically controlled dieless incremental forming of a sheet metal. *Proc. IMechE Part B: J. Engineering Manufacture*, 215, 2001.

[76] R. McBride, M. Gross, A. J. Moore, D. P. Hand, and J.D.C. Jones. Calibration of bending and membrane strains for iterative laser forming of non-developable surfaces. In *Proceedings of the 23rd International Congress on Applications of Lasers and Electro-Optics*, pages 71–80, San Francisco, CA, 2004.

[77] H. Meier, V. Smukala, O. Dewald, and J. Zhang. Two point incremental forming with two moving forming tools. *Key Engineering Materials*, 344:599–605, 2007.

[78] F. Micari, G. Ambrogio, and L. Filice. Shape and dimensional accuracy in single point incremental forming: State of the art and future trends. *J. Materials Processing Technology*, 191:390–395, 2007.

[79] S. A. Mitchell and S. Vavasis. Quality mesh generation in higher dimensions. *SIAM J. Computing*, 29:1334–1370, 2000.

[80] H. D. Mittelmann. An independent benchmarking of SDP & SOCP solvers. *Mathematical Programming*, 95:407–430, 2003.

[81] K. Mori, M. Otsu, N. Fujiwara, and K. Osakada. Incremental hammering forming of sheet metal using CCD camera and database. In *Proc. NAMRX XXVI*, pages 49–54, Atlanta, GA, 1998.

[82] K. Mori, C. C. Wang, and K. Osakada. Inclusion of elastic deformation in rigid-plastic finite element analysis. *Int. J. Mech. Sci.*, 38:621–631, 1996.

[83] K. Mori, M. Yamamoto, and K. Osakada. Determination of hammering sequence in incremental sheet metal forming using a genetic algorithm. *J. Materials Processing Technology*, (60):463–468, 1996.

[84] K. I. Mori, G. Yang, and K. Osakada. Determination of optimal motion of tools in metal forming processes by controlled FEM simulation. *Int. J. Machine Tools and Manufacture*, 35(6):851–859, 1995.

[85] MOSEKApS. *The MOSEK optimization toolbox for MATLAB version 3.2 (Revision 8). User's guide and reference manual.* MOSEK ApS. Denmark., 2002.

[86] O. Music and J. M. Allwood. A method for evaluation of flexible forming processes. In *Proceedings of 9th Int. Conf. Technology of Plasticity*, pages 1771–1776, Seoul, Korea, 2008.

[87] S. I. Oh and S. Kobayashi. Finite element analysis of plane-strain sheet bending. *Int. J. Mech. Sci.*, 22:583–594, 1979.

[88] N. Okada, G. Ro, and Y. Suzuki. Method and apparatus for incremental forming. *United States Patent 6971256.* 2005.

[89] N. Otegi, L. Galdos, A. Sukia, and A. Mentxaka. Optimization of geometrical accuracy of an industrial shape in single point incremental forming. In *Proceeding of 9th Int. Conf. Technology of Plasticity*, pages 1983–1989, Seoul, Korea, 2008.

[90] M. Otsu, M. Fuji, and Kozo Osakada. Three-dimensional laser bending of sheet metal. In *Proc. of the 6th ICTP*, pages 1025–1030, Erlangen, Germany, 1999.

[91] Jong-Jin Park and Yung Ho Kim. Fundamental studies on the incremental sheet metal forming technique. *J. Materials Processing Technology*, 140:447–453, 2003.

[92] K. Park and D.Y. Yang. Mismatching refinement with domain decomposition for the analysis of steadystate metal forming process. *Int. J. for Numerical Methods in Engineering*, 48:1089–1106, 2000.

[93] P.-O. Persson and G. Strang. A simple mesh generator in matlab. *SIAM Review*, 46(2):329–345, 2004.

[94] D. T. Pham and D. Karaboga. *Intelligent Optimisation Techniques.* Springer-Verlag, 2000.

[95] N. N. Powell and C. Andrew. Incremental forming of flanged sheet metal components without dedicated dies. *Proc. IMechE Part B: J. Engineering Manufacture*, 206:41–47, 1992.

[96] E. Quigley and J.Monagan. The finite element modelling of conventional spinning using multidomain models. *J. Materials Processing Technology*, 124:360–365, 2002.

[97] A. Raithatha and S. Duncan. An improved finite element model of incremental forming using conic programming. In *Proc. 2008 ACC*, pages 5109–5114, Seattle, WA, 2008.

[98] A. Raithatha, S. Duncan, K. Jackson, and Julian Allwood. Second order cone programming in modeling incremental deformation. In *Proc. 2007 American Control Conf.*, pages 4841–4846, New York, NY, 2007.

[99] A. Raithatha and S. R. Duncan. Deformation model for incremental forming using conic programming. In Pavel Hora, editor, *Numisheet 2008 - Proc. 7th International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes*, pages 749–754, Interlaken, Switzerland, 2008.

[100] A. Raithatha and S. R. Duncan. Rigid plastic model of incremental deformation using second-order cone programming. *Int. J. Numerical Methods in Engineering*, 78(8):955–979, 2009.

[101] A. Raithatha, S. R. Duncan, and J. Allwood. Tool-path optimisation for axisymmetric parts in incremental sheet forming. *IEEE Trans. Automation Science and Engineering*, page submitted, 2009.

[102] A. Raithatha, K. Jackson, S. Duncan, and J. Allwood. New method for modeling plastic deformation in incremental sheet forming. In *Proc. IEEE CCA*, pages 1037–1042, Munich, Germany, 2006.

[103] T. Remus. *Ultimate Sheet Metal Fabrication Book.* Wolfgang Publications, Inc., Scandia, MN, 1999.

[104] C. Robert, P. Dal Santo, A. Delamzire, A. Potiron, and J.-L. Batoz. On some computational aspects for incremental sheet metal forming simulations. *Int. J. Material Forming*, pages 1–4, 2008.

[105] S. Roberts and J. S. Shipman. *Two-Point Boundary Value Problems: Shooting Methods.* Elsevier, New York, NY, 1972.

[106] Y. Saotome and T. Okamoto. An in-situ incremental microforming system for three-dimensional shell structures of foil materials. *J. Materials Processing Technology*, 113:636–640, 2001.

[107] T. Schafer and R. D. Schraft. Incremental sheet metal forming by industrial robots. *Rapid Prototyping Journal*, 11(5):278–286, 2005.

[108] M. J. Sharan. Comparison of elastic-plastic and rigid-plastic implicit FEM simulations in sheet forming applications. *J. Materials Processing Technology*, 27:279–300, 1991.

[109] M-S Shim and J-J Park. The formability of aluminium sheet in incremental forming. *J. Materials Processing Technology*, 113:654–658, 2001.

[110] S. Shima. State of the Art - Incremental Forming. In *Proc. 3rd IPMM*, pages CD–ROM, Vancouver, BC, Canada, 2001.

[111] R. Szilard. *Theories and Applications of Plate Analysis.* John Wiley and Sons, 2004.

[112] S. Tanaka, T. Nakamura, and K. Hayakawa. Incremental sheet metal forming using elastic tools. In *Proc. of the 6th ICTP*, pages 1477–1482, Erlangen, Germany, 1999.

[113] S.P. Timoshenko and S. Woinowsky-Kreiger. *Theory of Plates and Shells. International Student ed.* McGraw-Hill International Book Company, 1959.

[114] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[115] E. Ventsel and T. Krauthammer. *Thin Plates And Shells: Theory Analysis And Applications.* Marcel Dekker Inc, 2001.

[116] R. H. Wagoner and J.-L. Chenot. *Fundamentals of Metal Forming.* John Wiley & Sons, Inc., New York, NY, 1997.

[117] Yu Xia. A Newton's method for perturbed second-order cone programs. *Computational Optimization and Applications*, 37:371–408, 2007.

[118] W. H. Yang. Large deformation of structures by squential limit analysis. *Int. J. Solids and Structures*, 30(7):1001–1013, 1993.

[119] S. J. Yoon and D. Y. Yang. Investigation into a new incremental forming process using an adjustable punch set for the manufacture of a doubly curved sheet metal. *Proc. IMechE Part B: J. Engineering Manufacture*, 215, 2001.

[120] Q. Zeng, A. Combescure, and F. Arnaudeau. An efficient plasticity algorithm for shell elements application to metal forming simulation. *Computers and Stuctures*, 79:1525–1540, 2001.

[121] H. W. Zhang and M. Wang. An improved mathematical programming method for wrinkling prediction of cloths. *Int. J. Clothing Science and Technology*, 17:13–28, 2005.

[122] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method for Solid and Structural Mechanics, 5th edition.* Butterworth-Heinemann, 2000.

[123] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method Volume 1: The Basis, 5th edition.* Butterworth-Heinemann, Oxford, UK, 2000.