Aspects of Numerical Simulation of the Metal Forming Process

By

R. H. Steventon.

A half dissertation submitted in partial fulfilment of the requirements
for the degree of
Master of Science in Engineering.

Centre for Research in Computational
and Applied Mechanics (CERECAM)
Department of Mechanical Engineering
University of Cape Town

Declaration

This is to certify that the results, calculations and other work presented in this document are essentially my own work, and that no part of it has been submitted to any other university.

Permission is hereby given to the University of Cape Town to reproduce this document in whole or in part.

R. H. Steventon

September 2001

Dedication

I would like to dedicate this thesis to my parents and specifically, to those who have influenced my life to make me who I am.

They have shown me either through expression or action what I can become.

They have given me something to strive towards.

Acknowledgements

I wish to acknowledge the help I have received from the following people:

- Mr H.C Bowles, under whose supervision this thesis was executed and who allowed me the freedom to explore other areas of interest.
- Professor Jack Hu of the University of Michigan for access to experimental facilities.
- Mr Jian Yao at the University of Michigan for operation of the Press in the Experimental Stamping Lab.
- Dr Rob Knutsen of the Centre for Materials Engineering at the University of Cape Town for material testing.
- My peers at the Centre for Research in Computational and Applied Mechanics (CERECAM) for their assistance and friendship and especially, Mr Victor Balden.
- The National Research Foundation (NRF) and CERECAM for financial assistance

Budiansky's dream (Harvard, mid-1970s): *"I imagined a black box – a black computation box that was incredibly powerful and into which we could feed a mathematical description of what the stylists envisioned for a certain sheet metal shape. Then push a button and the computer spits out the die shape, the blank configuration needed, the draw beads and their orientation and configuration. If it is not possible, it tells us that too!"*

Abstract

Metal forming is one of the oldest disciplines and dates back many thousands of years. The pressing of metal into shapes as a manufacturing process is widely distributed and is typically performed in engineering workshops. This process can be simulated to a degree, with a computer which provides many economic advantages.

The metal forming operation was simulated using the general purpose finite element software ABAQUS for a punch and die geometry available at the University of Michigan. The experiments were conducted for a variety of binder pressures, punch stroke speeds and material thickness's.

The material used for the experiments was supplied in two thicknesses. The University of Michigan supplied material properties, but these were unconfirmed. Subsequently, it was discovered that the material thicknesses were incorrect, but a virgin blank was available to conduct material tests for one of the material thicknesses. Once the material was characterised, simulations were conducted with varying values of friction to determine the applicable friction value. The friction value was determined and correlated well with the expected value obtained from literature.

The formed metal blanks were acid etched with a rectangular pattern to allow experimental determination of strain values, but the resultant measured strain values were of the same order as the measurement accuracy, so this avenue was not followed further.

The experiments produced consistent results for repeated experiments and the simulations correlated well for values of punch force and final blank shape when the correct material flow curve and friction value were used. This is important because it shows that a forming operation may be simulated and a Press capacity established which will allow an engineering workshop to determine whether or not they have a Press large enough to manufacture the part.

Table of Contents

Table of Figures

List of Tables

Nomenclature and Symbols

ACIS	Spatial Technology Inc geometry engine.
BDC	Bottom dead centre
Binder	Tool used to hold down metal sheet around the die
Blank	Metal sheet to be formed
CAD	Computer Aided Drawing/Design
CMM	Coordinate Measuring Machine
CSV	Comma Separated Values
Die	Female tool
FEA	Finite element analysis
FLD	Forming Limit Diagram
Psi	pounds per square inch (unit of pressure)
Punch	Male tool
SAT	Spatial Technology Inc. file format for ACIS Engine
SPM	Strokes per minute
Shut height	Over-closure distance between punch and die
TDC	Top dead centre

$\mathbf{B}$	geometry matrix
$\mathbf{M}$	mass matrix
$\mathbf{a}$	acceleration vector
$\dot{d}$	derivative of displacement with respect to time(velocity)

$\ddot{d}$	second derivative of displacement with respect to time (acceleration)
E	Young's Modulus
ε	strain
F^T	tangential friction force
L	characteristic length
σ^o	initial yield stress of material
σ	stress
P	normal pressure at the interface
ρ	density
t	time
Δt	change in time
μ	friction factor
ν	Poisson's ratio
V	Volume

1 Introduction

1.1 Background

For as long as man has used metal based tools, some form of metal forming has been carried out. What started as the beating of copper with a rock has become an industry upon which civilization is based. From copper to bronze to iron to steel, man has always sought to better his life, first by making better materials, and then by making more complex shapes with which to build machines.

Metal forming practice dates back to the traditional blacksmith who forged and formed metal to his will, producing useful implements for the people in the area surrounding the smithery. Today this place is occupied by companies with workshops employing people who mind machines which do the labour. The design of the tooling within the machine which produces the formed item is carried out remotely using either best practice techniques or computer based simulations.

For complex geometry and multi-stage forming, computer based simulations can avoid the need for costly tooling tryouts and multiple attempts at creating the correct tooling for a particular item. Continuous improvements in computer processing power have allowed personal computers to run simulations of greater complexity than a supercomputer could handle 10 years ago. Thus simulation of manufacturing processes has become the norm rather than the exception.

This has had several benefits, including reduced costs, faster time to market and an overall improvement in the quality of the product.

1.2 Objectives

This work correlates the physical process of a metal forming operation with a simulation using the general purpose finite element program ABAQUS.

1.3 Scope

The work includes the experimental forming work undertaken at the University of Michigan and the simulation of the forming process at the University of Cape Town.

The simulation required the generation of the appropriate tooling geometry, and finite element simulation to be performed. A variation of the experimental parameters along with matching of the appropriate simulation parameters is undertaken. Comparison of the reaction forces on the tooling and the final part shape to the simulated part shape are also prepared.

1.4 Plan of development

The literature is surveyed and a brief introduction to the finite element method is given. In addition, issues relating to the successful simulation of the metal forming process are addressed. A typical forming simulation is then shown, and implementation of the corresponding finite element model is discussed.

The Press in the University of Michigan's Stamping Lab is reviewed and the experimental data logging facilities discussed, along with the calibration and processing of the data.

The simulation of the Michigan Press experiments is related, starting with the generation of tooling from Coordinate Measuring Machine data. The ABAQUS model is then described in detail and the various aspects of the experimental results are correlated with simulation for groups of experiments with similar parameters.

Finally, conclusions are drawn relating the relative success of simulating the forming process. The Appendices contain computer program, macros and other information relating to this work.

2 Literature Review

2.1 Factors affecting an analysis and associated issues

In theory, the operation of sheet-metal forming is relatively straightforward. A sheet of metal is plastically deformed into a desired shape. In practice, there are many factors which can influence the successful production of a formed metal part, the majority of which are physical factors such as sufficient lubrication and careful tool design. Other factors that can contribute to the formability of the metal component are machine parameters such as the speed at which the operation is carried out. While these physical factors should be applied to a simulation, realization of an accurate simulation also relies on the correct choice of various other parameters by the analyst. Issues such as how time is modelled, material and friction models and element formulation all play a role in introducing difficulties into the simulation. However, the first issue is almost always the geometric description of the model.

2.2 The finite element method

The mathematical model of a simple problem with discrete boundary conditions is trivial. Thus the capability to extend that model to a geometrically complex structure is greatly desired. With the inclusion of boundary conditions that may vary both in time and space and be functions of other variables, the probability of successfully formulating a set of equations that satisfy the problem description approaches nil.

However, if the problem is broken down into a large number of simple models, and the governing equations for these are solved simultaneously by a computer, then the situation improves considerably. Since the governing equations for each simple model are known, the only physical issues relate to the transfer of boundary conditions between models. Numerical issues such as efficient use of the computer and the simultaneous solution of a large number of equations are dealt with in their own way.

This breaking down of a model into smaller pieces is known as discretisation. Each 'small model' is represented by an 'element' with its own unique geometric formulation and material model. The element's geometry is described by 'nodes' typically at the corners of the element. A four noded shell element with four corner nodes typically used for discretising flat shapes such as those found in metal forming applications is shown below in Figure 2.1

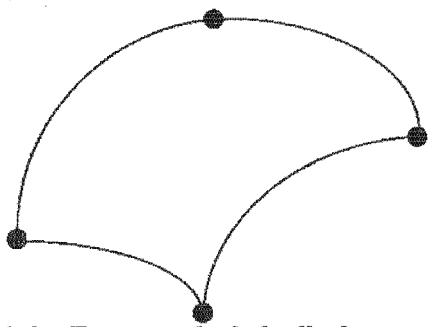

Figure 2.1 : Four noded shell element

The element uses shape functions to approximate the continuous field described by the governing partial differential equations (PDE) and thus may be either linear or quadratic. The shape functions are mapped from real

space (sometimes called material space) to element space where they are numerically integrated using Gauss-Legendre quadrature for computational efficiency.

Each element's contribution is then summed at the nodes to provide a global matrix. The essential boundary conditions are then enforced and the global matrix solved. This process is then repeated for the next time increment.

2.3 Solution methods: Implicit static vs. explicit dynamic formulation

Explicit solution methods are typically used in the modelling of dynamic problems involving contact due to stability and convergence problems when using implicit solution methods. Typically, a forming simulation involves the plastic deformation of material and then the elastic recovery of any residual elastic stresses. This is split into two simulations, the first of which is modelled using an explicit time stepping method, and the second using an implicit time step. The rationale for this is clear, since springback calculations are numerically expensive[1,2] in the explicit time stepping regime.

This can be seen from the derivation of the discrete momentum equations for the implicit vs. the explicit scheme at time step n where $\mathbf{d}$ is the nodal displacement. For a non-linear static analysis, the equilibrium equations are written in terms of a residual vector $\mathbf{r}$.

Static Implicit: $$0 = \mathbf{r}\left(\mathbf{d}^{n+1}, t^{n+1}\right) = \mathbf{f}^{int}\left(\mathbf{d}^{n+1}, t^{n+1}\right) - \mathbf{f}^{ext}\left(\mathbf{d}^{n+1}, t^{n+1}\right) \quad \text{Eqn. 1}$$

And : $$f^{int} = \int \mathbf{B}^T \sigma \, dV$$ Eqn. 2

Where the column matrix $\mathbf{r}\left(\mathbf{d}^{n+1}, t^{n+1}\right)$ is a residual and **B** is a function of the nodal coordinates and the shape functions used, the superscript n+1 refers to the time increment counter and f^{int} is the vector of internal forces. In the non-linear case, **B** is also a function of the displacements (geometric stiffness).

The discrete equations for both the equilibrium and the equations of motion are nonlinear algebraic equations in the nodal displacements and thus have to be implicitly solved.

2.4 Solution methods for static implicit

The solution method for the resultant nonlinear equations is normally attempted using the Newton-Raphson method generalised to an arbitrary number of unknowns. This is shown in the equation which is derived from a truncated Taylor expansion of the equation for the residual **r**:

$$\Delta d = -\left(\frac{\partial r(d_v)}{\partial d}\right)^{-1} r(d_v)$$ Eqn 3

Since the higher order terms have been neglected, the model is said to have been linearised. The solution is obtained by solving over a sequence of linear models with the new value for the unknown in each step being updated using the equation below:

$$d_{v+1} = d_v + \Delta d \qquad \text{Eqn 4}$$

The iterations are continued until the desired level of accuracy has been reached for the solution. However, while this algorithm is generally robust, severe discontinuities such as contact and extreme non-linearity such softening (shown in Figure 2.2 below) may result in convergence problems. This can sometimes be overcome using line-searching or arc-length methods but contact discontinuities are still an issue.

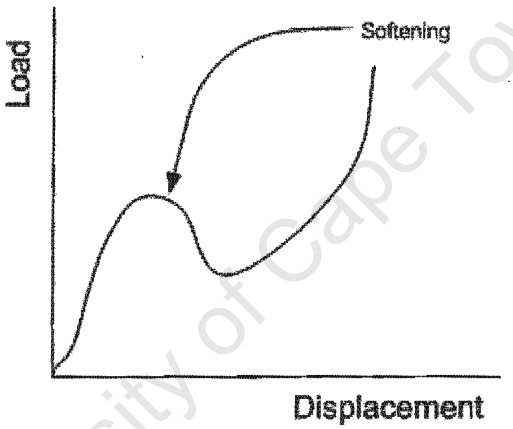

Figure 2.2 : Load vs. Displacement for softening

For these reasons, attempting an explicit solution to a problem may be advantageous over an implicit solution.

2.5 Dynamic explicit formulation

Within the explicit formulation, the dynamic equilibrium equation (Newton's second law) is written as:

Explicit: $$\mathbf{M}\mathbf{a}^n = \mathbf{f}^{ext}\left(\mathbf{d}^n, t^n\right) - \mathbf{f}^{int}\left(\mathbf{d}^n, t^n\right) \qquad \text{Eqn 5}$$

The explicit formulation is an ordinary differential equation which is second order in time. It should be noted that the explicit scheme is subject to boundary conditions which are applied at the nodes.

The overriding point of interest is that in the explicit scheme, the time integration of the discrete momentum equations does not require the solution of any equations, but simply the evaluation of sequential equations. This further allows the exploitation of diagonal mass matrices[3].

2.6 Integrating with respect to time

The most commonly used method of time stepping in explicit formulations is the Central Difference Method. In this method, the time step is variable, and can thus be changed dynamically in response to mesh deformation and changes in wave speed due to stress. The generalised Central Difference Method is obtained using difference equations for the velocity and acceleration in terms of the values at discrete time intervals. These are substituted into Equation 2 to provide an explicit relationship defining new values in terms of the previous values. The time increment is formulated as follows:

Time: $$\Delta t^{n+1/2} = t^{n+1} - t^{n},\ t^{n+1/2} = \frac{1}{2}\left(t^{n+1} + t^{n}\right),\ \Delta t^{n} = t^{n+1/2} - t^{n-1/2} \qquad \text{Eqn 6}$$

And the central difference formulation for the velocity is expressed as:

Velocity: $$\dot{\mathbf{d}}^{n+\frac{1}{2}} \equiv \mathbf{v}^{n+\frac{1}{2}} = \frac{\mathbf{d}^{n+1} - \mathbf{d}^{n}}{t^{n+1} - t^{n}} = \frac{1}{\Delta t^{n+\frac{1}{2}}}\left(\mathbf{d}^{n+1} - \mathbf{d}^{n}\right)$$ Eqn 7

Differentiating and back substituting the velocity and velocities half-step counterpart, the acceleration in terms of displacement and time is obtained:

Acceleration: $$\ddot{\mathbf{d}}^{n} \equiv \mathbf{a}^{n} = \frac{\Delta t^{n-\frac{1}{2}}\left(\mathbf{d}^{n+1} - \mathbf{d}^{n}\right) - \Delta t^{n+\frac{1}{2}}\left(\mathbf{d}^{n} - \mathbf{d}^{n-1}\right)}{\Delta t^{n+\frac{1}{2}}\,\Delta t\,\Delta t^{n-\frac{1}{2}}}$$ Eqn 8

Once this term has been substituted into the Explicit formulation of the momentum equations, the solution is readily obtainable.

2.7 Contact and friction

Contact is defined as the impenetrability of one body with relation to another body. This is normally enforced as a boundary condition at nodes using a search algorithm to prevent the nodes of one body (slave) from penetrating the elements of another body (master) and is shown below in Figure 2.3.

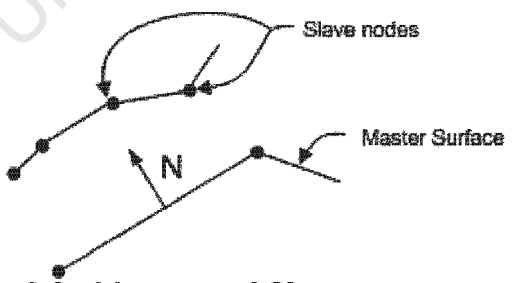

Figure 2.3 : Master and Slave contact surfaces

The search algorithm calculates the normal of the closest master surface facet for a particular slave node. If contact occurs between the slave node

and the master surface facet, this normal is used to produce measures of reaction force on the node and tangential slip.

The interface conditions between the two bodies when contact occurs are modelled as friction. However, one of the key problems with the simulation of large deformation and contact is the nonlinear properties of contact itself. This is because the tangential velocities are discontinuous in time during slip-stick behaviour. As mentioned previously, these discontinuities introduce significant problems in the time integration of an Implicit formulation and also impair the performance of Newton algorithms during the solving procedure. Both these factors contribute to the Explicit method being preferred for contact and large deformation problems.

The simplest friction model is the Coulomb model which relates the Tangential Friction force (F^T) to the pressure at the interface (P) multiplied by a friction factor.

Friction: $$F^T = \mu P$$ Eqn 9

In addition, some sort of capping value may be applied to the maximum friction value (shear capping). This is shown below in Figure 2.4.

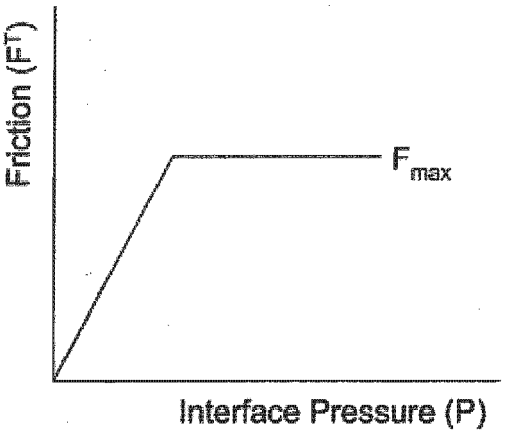

Figure 2.4 : Shear capping of friction

Much work has been done on friction models, and there exist numerous variations. Since the friction may be discontinuous due to slip-stick conditions, the resultant friction force will also contain discontinuities. This may be alleviated to some degree by using plasticity or pressure based friction models which introduce the additional response of asperities on the surfaces. Recent work[4] has also recognised that the quality of the surfaces affects the amount of lubricant retained between the surfaces when contact occurs under an applied load. This has a direct effect on the resultant effective friction coefficient.

When using a simple Coulomb friction model in metal forming, the typical values of the friction co-efficient for a well lubricated surface are expected to be between 0.01 and 0.15

2.8 Material properties

Metal is a complex material, and can react in totally different ways dependent on loading speed and temperature, even for a given set of isotropic material

properties. In addition, the majority of constitutive models are useful only over a limited range. Provided the analyst is aware of the limitations in the model formulation, the available models can still be used to model a material's response with a high degree of accuracy. Typically, the isotropic elastic-plastic model is used for basic metal forming simulations unless more detailed material model information is available regarding anisotropy.

A typical elastic-plastic material curve is shown in Figure 2.5. Initial loading is in the Elastic region. After the yield point (σ_o) is passed, the material deforms plastically. If the material is subsequently unloaded, elastic recovery occurs. Unloading the material from within the Plastic region will result in not all the deformation being elastically recovered (ie: permanent strain).

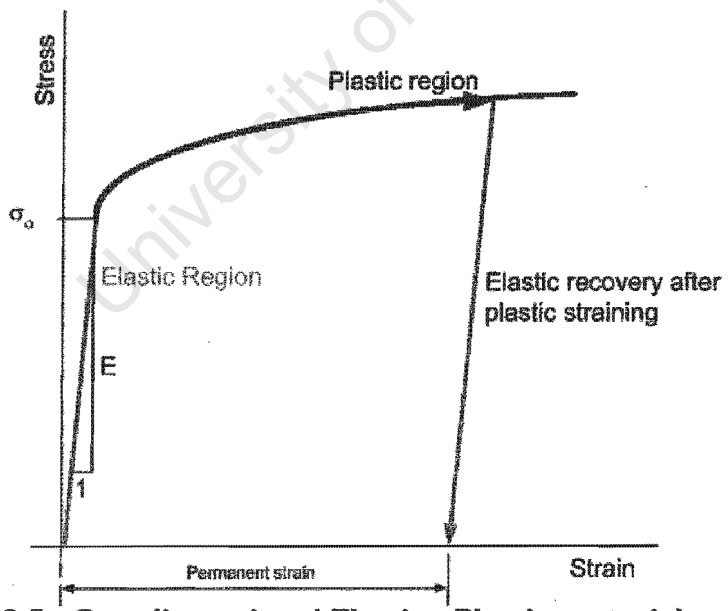

Figure 2.5 : One dimensional Elastic - Plastic material curve

For one dimensional plasticity, the plasticity theory is broken up into the following parts[3]:

1. A decomposition of the strain rate into its elastic and plastic components. This is shown in the Equation below.

$$\dot{\varepsilon} = \dot{\varepsilon}^e + \dot{\varepsilon}^p \qquad \text{Eqn 10}$$

This can be integrated over the time interval to provide $\Delta\varepsilon = \Delta\varepsilon^e + \Delta\varepsilon^p$. The elastic part is recoverable, whereas the plastic part of the strain is not.

2. A yield function which governs the start of (σ^0 shown in Figure 2.5), and continuance of plasticity. The evolution equation for yield of the elastic-plastic surface is typically denoted below for the yield condition:

Yield condition: $f(\sigma, q) = 0$ Eqn 11

Where q is used to denote an internal set of variables to the model and may include items such as temperature dependence etc. The yield function is a mathematical criteria used to define the onset and continuance of plastic flow. This is shown in the Kuhn-Tucker relationships defined below:

$$\begin{aligned} f(\sigma, q) \leq 0 \quad &\Leftrightarrow \quad \dot{\lambda} = 0 \\ f(\sigma, q) = 0 \quad &\Leftrightarrow \quad \dot{\lambda} \geq 0 \\ \dot{\lambda} \cdot f(\sigma, q) &= 0 \end{aligned} \qquad \text{Eqn 12}$$

3. A flow rule which is used to determine the plastic flow increments. In a constitutive model, the strain rate is additively decomposed into its plastic and elastic components, and the rate of plastic flow is given by the equation below.

Flow rule: $\dot{\varepsilon}^p = \dot{\lambda} r(\sigma, q)$ Eqn 13

Where $\dot{\lambda}$ is the plastic rate parameter and r(σ,q) is the plastic flow vector defining the relative magnitudes of the strain components.

4. Evolution equations for internal variables, such as strain hardening relations which would govern the evolution of the yield function. The evolution is defined in terms of the scalar multiplier λ and a hardening function h.

Evolution equations: $\dot{q} = \dot{\lambda} h(\sigma, q)$ Eqn 14

Example: isotropic hardening is often used and is generally a function of the history of plastic deformation.

Plasticity is also path dependent and dissipative and may be associative or non-associative depending on the plastic flow direction. A large part of the energy is used to plastically deform the metal, and this energy is irreversibly converted to other forms of energy in the metal structure and into heat.

The reader is referred to T Belytschko et al [3] for a more comprehensive treatment of computational plasticity.

2.9 Deformation speed

Although the majority of physical metal forming takes place at relatively low punch speeds (in the order of 0.5 m/s), it is not always possible to numerically simulate a forming process at this speed using dynamic explicit

techniques. This is illustrated by the following example. Given the minimum stable time increment for the central difference integration scheme:

Minimum time: $$\Delta t \leq \frac{L}{\sqrt{E/\rho}}$$ Eqn 15

Where L is a characteristic length, E is Young's Modulus, and ρ is the material density. From this it follows that for an element length of 1 mm, and material properties for steel, the minimum time increment is approximately 2.E-9 seconds. For a forming process where the punch speed is 0.5 m/s and the distance travelled is 100 mm, this translates to over one million iterations through a dynamic explicit code. Although this is feasible, it may require several weeks of computing time. In order to make the simulation process commercially viable, one of two types of numerical trickery is employed:

1. The punch speed is artificially increased. If strain rate dependence is included in the model, then the increased punch speed will cause the plasticity calculations to over predict the material yield and flow values.

2. The density of the material is artificially increased. This has the effect of changing the speed of sound in the material, and hence increasing the minimum time by the square root of the density scaling factor.

If the analysis is sped up using one of these schemes, additional artificial body forces are introduced into the model. This has implications if the aim of the

analysis is to correctly predict the reaction force on the tooling, since the artificial body forces will be transferred to the tooling, along with the plastic forming induced loads.

2.10 Punch reaction force

As has been noted above, there are issues involved in modelling the punch reaction force correctly. Mamalis reported[5] good correlations between experimental and numerical results for the forming of deep drawn square cups. However, closer inspection of the presented data shows oscillation spikes in the numerical simulation data. The difference between the simulation and the experimentally recorded data is as much as 20% in some areas of the process in this case. However, other experimental results[6] have been compared to numerical simulation and have produced excellent correlation in punch force versus displacement.

2.11 Wrinkling

The areas where wrinkling will occur are of great interest to the analyst, since the majority of formed metal parts are designed to be smooth. The analyst therefore considers wrinkling to be a defect. Correlation work between experimental and numerical simulations of wrinkling has been done for conical cups[7]. The conclusions reached were that the numerical codes used were sufficient to predict the area of wrinkling, but not the exact number or size of wrinkling.

When attempting to predict wrinkling, it is important to differentiate between implicit and explicit integration FEA codes. Wrinkling is a problem of equilibrium state, and thus it is more difficult[8] for implicit codes to predict wrinkling correctly. Wang[9] indicated that commercial FEA codes using implicit FEA codes will not produce wrinkling, whereas explicit codes tend to produce wrinkles naturally. However, recent work[10] using bifurcation theory has however shown that implicit codes can also be used to model wrinkling successfully.

In addition to these difficulties, various works[7] have shown that the initial mesh has a large effect on the number of wrinkles. This is obvious when one realises that the mesh forms numerical bifurcation points from which buckling initiates. The extent and number of wrinkles is then governed by the available energy and the mesh initiation points.

2.12 Mesh quality

The quality of the mesh used to perform an analysis has an effect on the results. A mesh which contains distorted elements will introduce spurious stiffness in the distorted elements, as will a mesh whose elements are too large, thus not providing sufficient resolution in areas requiring it. However, the degree to which an element can be distorted without introducing additional spurious stiffness is dependent on the element formulation and thus will vary even for two similar elements implemented in two different finite element analysis software programs.

2.13 Choice of element formulation

Typical simulation of metal forming processes is carried out using shell elements in order to reduce the amount of computing time. These elements are either four noded or three noded shell elements. Typically, FEA codes do not implement full integration shell elements into their explicit packages, due to the extra computational expense.

3 Finite element simulation relating to forming

While the theory necessary to complete a finite element solution of the metal forming process is well defined, the mechanics of simulating the process also need to be investigated.

3.1 Press setup

A hydraulic or flywheel press is used to deliver the energy necessary to carry out the metal forming operation. Typically, the tooling is clamped into the Press, and the workpiece or "blank" is placed between the tooling. The tooling clamps the blank, and the punch stroke commences. During this stroke, the punch tooling is used to deform the workpiece into the die tooling.

3.2 Corresponding finite element model

This process is idealised into a three stage process within the finite element model. Stage one is used to set the tooling into the correct positions. During stage two, the blank holder tooling or "binder" clamps the workpiece. The actual forming of the blank (shown in blue in Figure 3.1) is then completed as stage three.

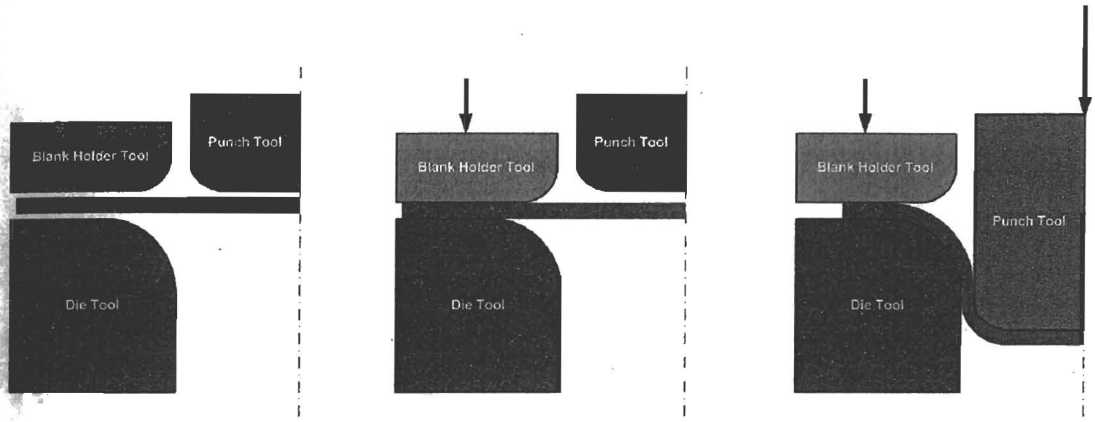

Figure 3.1 : The Three Stages of Simple Two dimensional forming

3.3 Initial state

The initial state is characterised by the initial positions of the various components necessary for a numerical simulation. It may be necessary to apply velocities to the components to move them into the desired positions, necessary for the simulation. It is also during this state that boundary conditions are applied to the tooling. Typical boundary conditions allow for translation in only one direction and possibly also symmetry of the workpiece.

3.4 Contact

Contact is declared between the various surfaces necessary to simulate the physical process. Friction properties for the contact condition are typically assumed values unless accurate friction data is available.

3.5 Movement of tooling

The tools are moved by applying velocity boundary conditions. If the boundary conditions are applied as displacements, then the energy necessary to move the tooling may be applied by the FEA code in a discontinuous

manner. This is evident as acceleration spikes applied to the tooling. If the purpose of the simulation is to acquire punch loading data, then the acceleration spikes will be manifest in this data too, possibly to the exclusion of any useful punch force load information.

3.6 Forming limit diagrams

Once the forming simulation is complete, the results can be examined. Often however, the most important criteria for the designer are the finished part quality and manufacturability. While quality can be measured directly from the simulation, in terms of areas of wrinkling, or thinning, the issue of manufacturability is not so easily measured. It is difficult to tell if a part will tear during the forming process, since finite element failure models are not built into the simulation. In practice, an experimentally determined failure diagram (Forming Limit Diagram or FLD) is created using principle strains until failure, and the theoretical results are overlaid onto this diagram. An example diagram is seen in Figure 3.2.

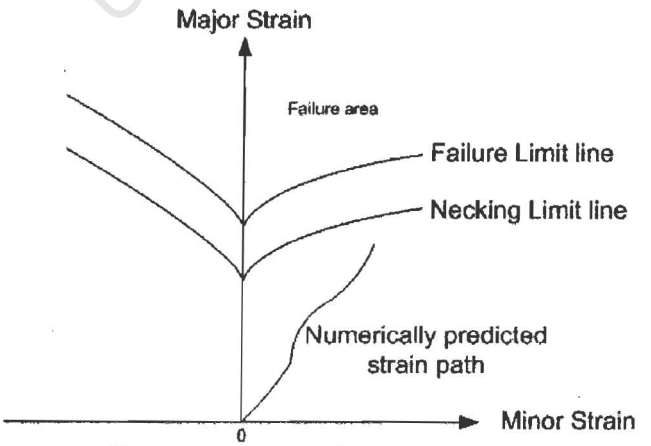

Figure 3.2 : Example Forming Limit Diagram with strain path overlay

If the theoretical results are within the failure area on the diagram, then the part is assumed to fail during manufacture, and the designer must attempt some other method of producing the part, possibly by modifying the tooling geometry.

4 Michigan Experimental Facilities

4.1 Introduction

A calibrated flywheel press was made available at the University of Michigan's Stamping Laboratory by Professor Jack Hu. The press allows for binder force control via a hydraulic pressure monitoring and feedback system. The press operates in one of two speed modes, namely either 12 or 18 strokes per minute. A picture of the press prior to instrumentation is shown below in Figure 4.1.

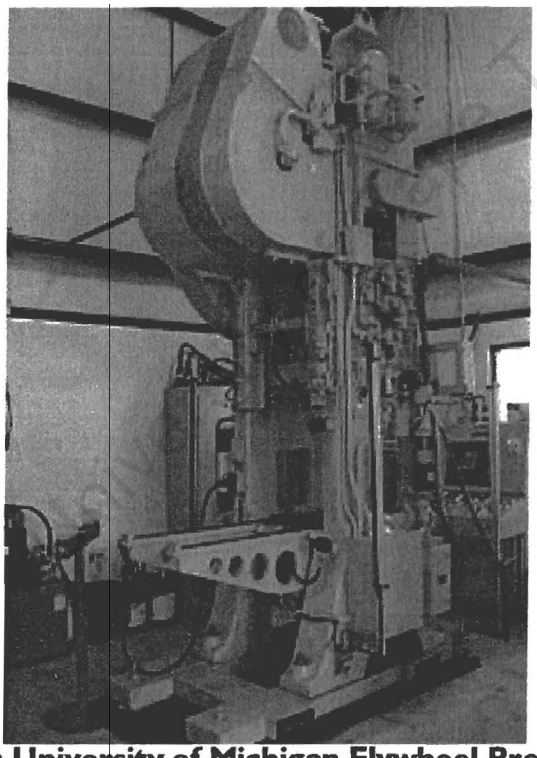

Figure 4.1 : University of Michigan Flywheel Press prior to Instrumentation

It is important to note that the experiments were performed in the United States of America and all results were obtained in imperial units. These, together with the SI equivalent are reflected in this document.

4.2 Press

The press is shown in its fully instrumented state in Figure 4.2. The instrumentation is capable of recording binder pressure, punch force and displacement and the forces on the press frame.

Control of the binder pressure is provided via a feedback mechanism which allows binder pressure control in increments of 500 psi (3.45 MPa). The binder pressure is supplied via 8 hydraulic cylinders each equipped with a load cell to enable the feedback force to be monitored.

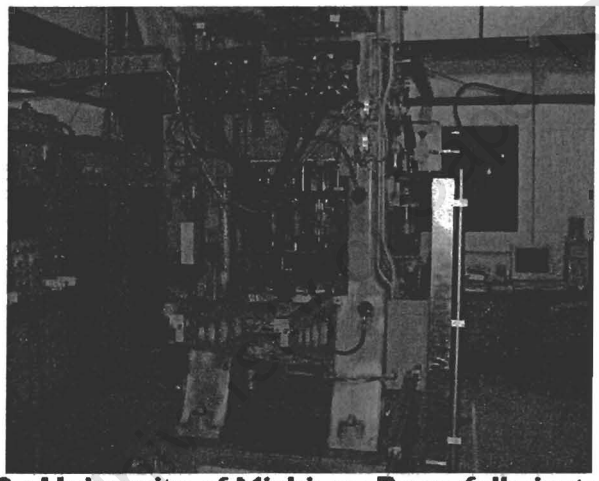

Figure 4.2 : University of Michigan Press fully instrumented

The tooling is placed inside the press, and secured via a bolted arrangement, with the die attached to the lower half of the Press.

4.3 Press safety

The Press is instrumented with a laser curtain around it. This prevents the press from operating if anyone is too close to the press during operation.

The controls also require both hands of the operator to be used to start a press cycle.

4.4 Press forming operation

In order to form a blank, the cycle shown in Figure 4.3 is followed once the press is operational.

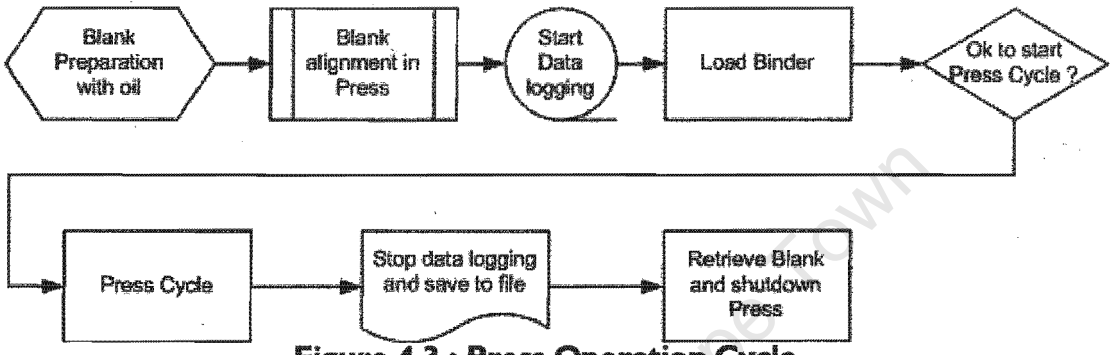

Figure 4.3 : Press Operation Cycle

All the blanks were laser cut to exactly the same shape and were consistently aligned within the press by aligning them against electrical tape placed on the tooling. Since the tape was thinner than the blank, it did not interfere with the forming operation. However, the tape was thick enough to allow blank alignment against its edge, thus ensuring consistent placement against two edges.

4.5 Logging of press data

The data was logged through an Analogue to Digital ISA card plugged into the data-logging PC. This was a 16 channel card, and was set to sample the inputs 100 times per second. The following data was logged:

- Punch position
- Master binder pressure cylinder pressure - required calibration
- Loadcell on binder cylinder N (8 data inputs) - required calibration
- Load on left of press frame
- Load on right of press frame
- Load on punch

The initial phase involved calibration of the press, during which the press was put through a known cycle, to produce voltage calibration values for the cylinder pressure (psi) and the feedback on the binder cylinders (tons).

The punch displacement sensor is a linear induction sensor with a static calibration value of 0.8 volts/inch (0.31 volts/cm).

The master binder cylinder pressure is the actual pressure in the binder cylinders. This value can be checked against the value that the Press was set to deliver for the Press cycle. In addition, this value needed to be calibrated at Press start-up, to provide a conversion factor from volts to psi.

Individual loadcell's on each of the eight binder cylinders which allowed feedback of the loads on the binder. The values also needed to be calibrated at Press start-up to convert the values from volts to tons.

The sides of the press separating the die (bottom half) from the binder and punch (top half) are calibrated with a preset sensor which provided the maximum load in tons on each side from a strain gauge for a press cycle. This strain gauge is also connected to the data logging system. The scaling factor for the strain gauges is derived by equating the maximum recorded

load on the preset sensor and the maximum recorded value on the data logger from the strain gauge.

This scaling factor is also applied to the strain gauge data from the punch to get the punch reaction force in tons.

The data was saved in a CSV format file with the number of data points recorded per second as the only header information (100 /sec in all cases).

4.6 Using the load on left / right of press for calibration

The press is instrumented with a data readout that provides that maximum load in tons that occurs on each side of the Press frame separating the die from the punch during a forming stroke.

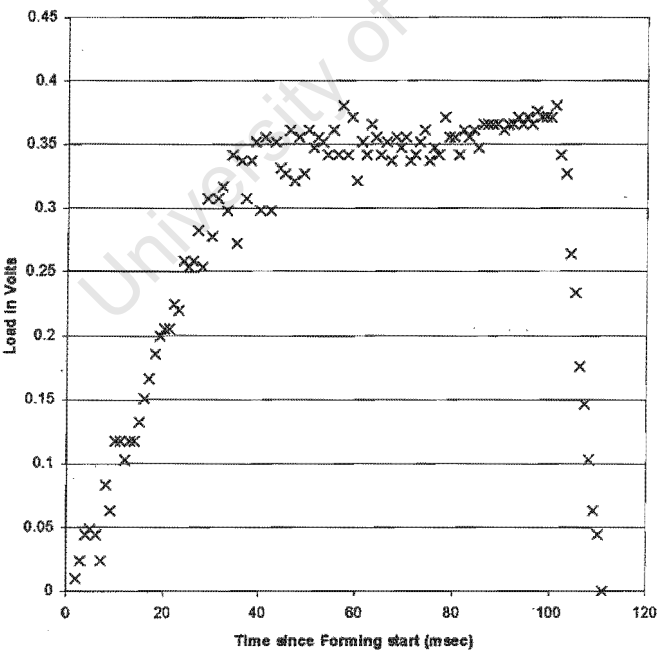

Figure 4.4 : Experiment 2, Load on Left of Press

Figure 4.4 shows the data for the load on the left side of the Press in Volts. The peak value is shown in the figure is 0.381 volts, which corresponds to 3.7 tons. It should be noted that there is a large amount of noise in the voltage signal. This can be seen in Figure 4.4 at the top of the initial voltage climb.

4.7 Matrix of experiments

Both 0.6 mm and 0.8 mm thick laser cut blanks were available, along with binder pressures in increments of 500 psi (3.45MPa), two possible Press cycle speeds, namely 12 and 18 press strokes/minute and a shut height of 2 inches (50.8 mm). This allowed various permutations of press characteristics to be tested. These experiments are shown in Table 4.1 along with the corresponding recorded maximum Press load on the left and right sides of the Press.

Table 4.1 : Matrix of Experiments

Exp. No.	Shut height (inches)	BHF pressure (psi)	Blank Thickness (mm)	Press cycle speed, (spm)	Load Left (tons)	Load Right (tons)
1	2	500	0.6	12	-	-
2	2	1000	0.6	12	3.7	3.5
3	2	1000	0.8	12	5.5	5.0
4	2	1000	0.8	18	5.5	5.0
5	2	1000	0.8	18	5.5	5.0
6	2	1000	0.8	18	5.5	5.0
7	2	1500	0.6	12	4.0	3.7
8	2	1500	0.8	12	5.7	5.2
9	2	2000	0.6	12	4.0	3.7
10	2	2000	0.6	12	4.2	4.0
11	2	2000	0.6	12	4.2	4.0
12	2	2000	0.8	12	6.0	5.5
13	2	2000	0.8	18	6.0	5.5
14	2	2500	0.6	12	4.5	4.2
15	2	2500	0.8	12	6.2	5.5

As can be seen from Table 4.1, various experiments were repeated, sometimes with different results in peak Press load (for example, experiments 9, 10 and 11).

4.8 Macro to process data files

A Visual Basic macro was written to run within the Microsoft Excel VBA runtime environment. This macro imported the raw data file and processed it. The macro determines forming start and end, then splits the data up into individual worksheets for each data logging device while copying only the relevant data within the forming sequence. Once the data is split, each worksheet's data is processed using the calibration data to produce a formatted chart of the data.

The user needs to change only the name of the data file to load, and to set the SPM variable correctly before executing the macro. The SPM variable is used to provide displacement data. An example of the raw data can be found as Appendix B.

4.9 Issues with data

There were various issues with the recorded data, including: number of significant digits recorded for calibration factors and a displacement transducer which was faulty. Each of these is dealt with in turn below.

4.9.1 *Significant digits of the punch calibration factor*

The maximum values are listed in the last two columns of Table 4.1. When these values are correlated with the voltage output of the same strain gauge, a scaling factor for all strain gauges can be obtained.

Ideally, the scaling factor for the left and the right side of the Press should return the same value, however, this is not the case, and so the two values are averaged to provide a global scaling value for the strain gauges. Since the maximum load reading is only provided to two significant digits, there is considerable scope for round-off as illustrated by the example below

Table 4.2 : Effect of significant digits on scaling factors

Experiment	Max (volts)	Max (tons)	Scaling factor
No 2 Left (2 significant digits)	.381	3.7	9.71
No 2 Right (2 significant digits)	.366	3.5	9.56
Test Left (3 significant digits)	.381	3.66	9.61
Test Right (3 significant digits)	.366	3.52	9.62

As can be seen by Table 4.2, if the maximum load reading in tons was increased from two to three significant figures, there could be a better correlation between scaling factors for the left and right side of the Press. Since both these values should be the same, it was decided to average the left and right sides of the press load scaling factor, and use this averaged value as a scaling factor for the punch force.

4.9.2 *Binder load case: 500 psi*

The binder is hydraulically controlled via a feedback loop. This allows a constant pressure in psi to be applied to the hydraulic cylinders controlling the binder. Available pressure settings are from 500 psi (3.45 MPa) to 2500 psi (155 MPa) in increments of 500 psi (3.45 MPa). However, Mr Jian Yao who instrumented and operates the press noted that the 500 psi (3.45 MPa) setting did not return accurate measurements due to insufficient pressure on the blank, which results in the blank being capable of lifting the binder as it wrinkles during draw-in.

4.9.3 *Broken punch displacement transducer*

After the data was logged, it was noticed that the punch displacement sensor was not functioning correctly. Unfortunately, there was insufficient time to order and take delivery of a new sensor. Due to this issue, the displacement data is inferred from previous work by Mr Navin Jahajeeah[11].

These prior experimental results, results obtained from Experiment 2 (typical of displacement data), and a curve fitted trend line are shown in Figure 4.5. This is overlaid with red and green lines showing the start and end of the punch contact with the blank.

The time scale in Figure 4.5 is measured from when timing commenced, and thus the start of timing is arbitrary relative to the point when the forming stroke commences. Data is sampled 100 times per second, and a lot of scatter can be seen prior to the stroke commencement.

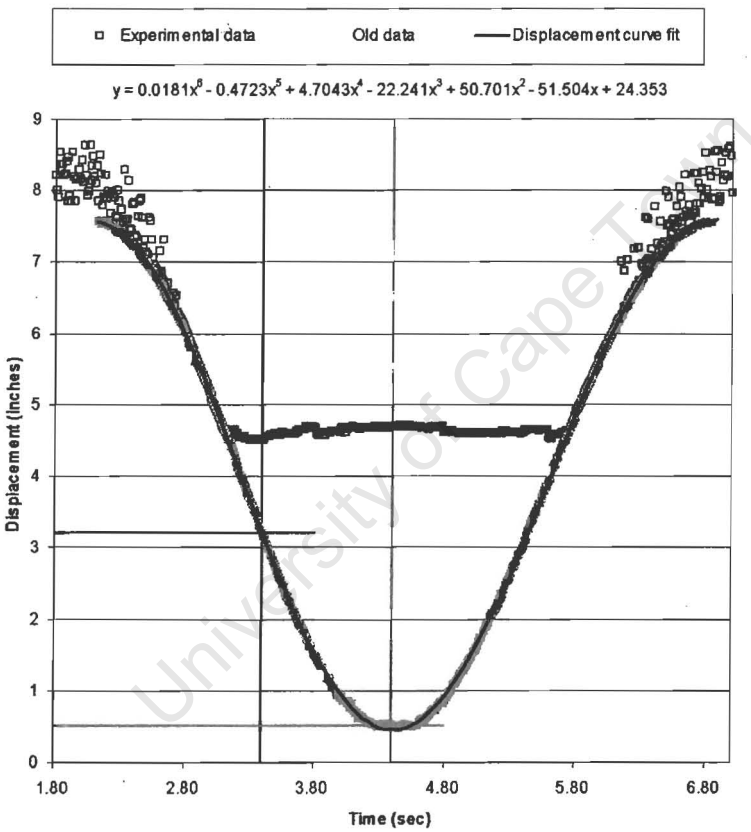

Figure 4.5 : Displacement vs. Time Data for Experiments

At a time point of approximately 3 seconds (corresponding to roughly 3 inches [75 mm] punch travel), the transducer used to measure the stroke stops functioning correctly. From this point on,

in order to follow the punch cycle, one must use the overlaid old data.

At a time of 3.4 seconds, the strain gauges start to output a voltage potential, and it is assumed that this is the point at which contact and forming commences. This point is indicated by the intersection of the two red lines cross on Figure 4.5. After a 2.5 inch (63.5 mm) travel of the punch, the bottom dead centre (BDC) of the stroke is reached (indicated by the intersection of two green lines on Figure 4.5).

After BDC, the strain gauge voltage drops to zero within less than 0.1 seconds (10 data points at 100 readings / second), as the punch leaves the formed blank within the die, and returns to top dead centre (TDC). At some point on the return stroke, the punch picks up the linear displacement transducer, and the readings again assume the rest of the expected curve. Once TDC has been attained, the data continues with a large amount of scatter about a constant position, showing no drift.

The data for the punch displacement was typical for all the experiments. Since the forming procedure takes place past the point where the displacement transducer is broken, no useful displacement data is available for any of the experiments, other than that which can be inferred from prior experiments.

5 Preparation for Numerical Simulation

5.1 Introduction

Before numerical simulation can be undertaken, certain physical information is needed. This includes the geometry of the tooling, and the material data necessary to model the metal to be formed. Other information such as friction and binder pressure is also necessary.

The tooling for the Press consists of various pieces made up by a local manufacturer within Ann Arbour, Michigan. Although the University of Michigan retains original Unigraphics solid model CAD files, the tooling has been modified to the extent that the existing CAD files are now essentially useless. Therefore, it was decided to obtain accurate tooling surface descriptions by using a Co-ordinate Measuring Machine.

5.2 Co-ordinate measuring machine (CMM) data

A co-ordinate measuring machine was used to provide point data on the surface of the tooling. However, the CMM machine was extremely old, and thus only low resolution data was available. In addition to this, the CMM machine's probe was unable to fit within the die.

5.3 Plaster casts for CMM

A plaster-caste was made of the die, and the plaster caste was then scanned with the CMM. The CMM data was provided in a text file of X,Y, Z points and converted to DXF format by a program written in the Perl scripting language. The Perl program is supplied as Appendix A.

5.4 Punch geometry

Figure 5.1 shows the raw CMM data for the punch, along with some curves (in red) used to fit surfaces onto the CMM data.

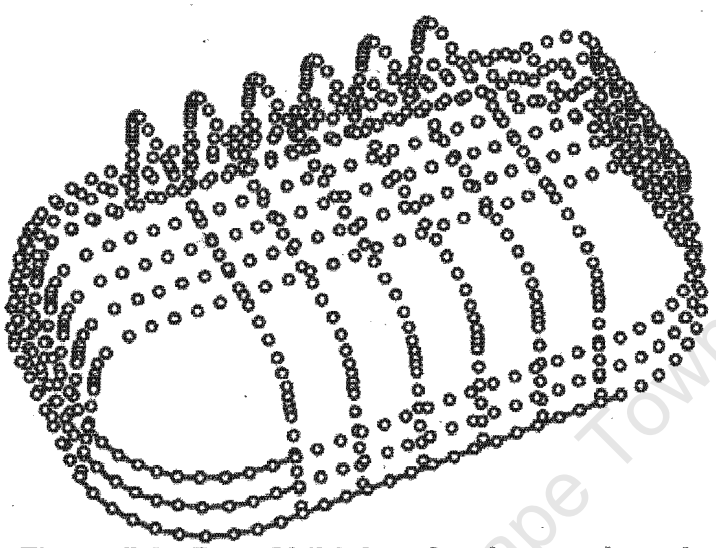

Figure 5.1 : Raw CMM data for the punch tool

Note that the above figure does not contain any CMM data for the corner radii of the punch. These corners were guesses by sweeping the existing edges around the perimeter of the punch.

The punch geometry also needed to be re-orientated so the surface that first contacts the blank was parallel to the blank. This is because the punch was not aligned prior to the original X,Y,Z data scan with the CMM machine.

5.5 Rhino3d surface model

Once the XYZ data was converted to a DXF file, the data was imported into Rhino3D (www.rhino3d.com). The points were then used as the basis for curves which in turn were used to create surfaces. The surfaces were

exported as IGES files and either used directly in the case of the punch, or used to cut a solid in the case of the die.

5.6 Die geometry

Figure 5.2 shows the CMM data for the die which is also incomplete, as only the inner die profile and some of the sidewall cross-sections are defined. However, once the X, Y, Z data points have been translated into a DXF file, the points can be imported into a CAD program and used to construct curves. By lofting the cross-sections curves and using the inner die profile as a guide curve, the die geometry was constructed to within a dimensional tolerance of approximately 1 mm.

Figure 5.2 : Raw CMM data for the Die tool

Alignment of the data to a reference plane again needed to be performed.

5.7 Scan blank shape

The blanks were laser cut steel sheet, either 0.6 or 0.8 mm thick and are shown in Figure 5.3. After the blank's were laser cut, a grid pattern was acid etched onto the steel to enable strain measurements to be taken after forming. The shape of the blank was determined by a prior PhD student using LS-DYNA in an attempt at creating a perfect shaped blank.

The spacing of the rectangular pattern is approximately 9 mm, and the thickness of the lines is approximately 1 mm. This implies that a resolution of the line thickness corresponds to approximately 11% strain.

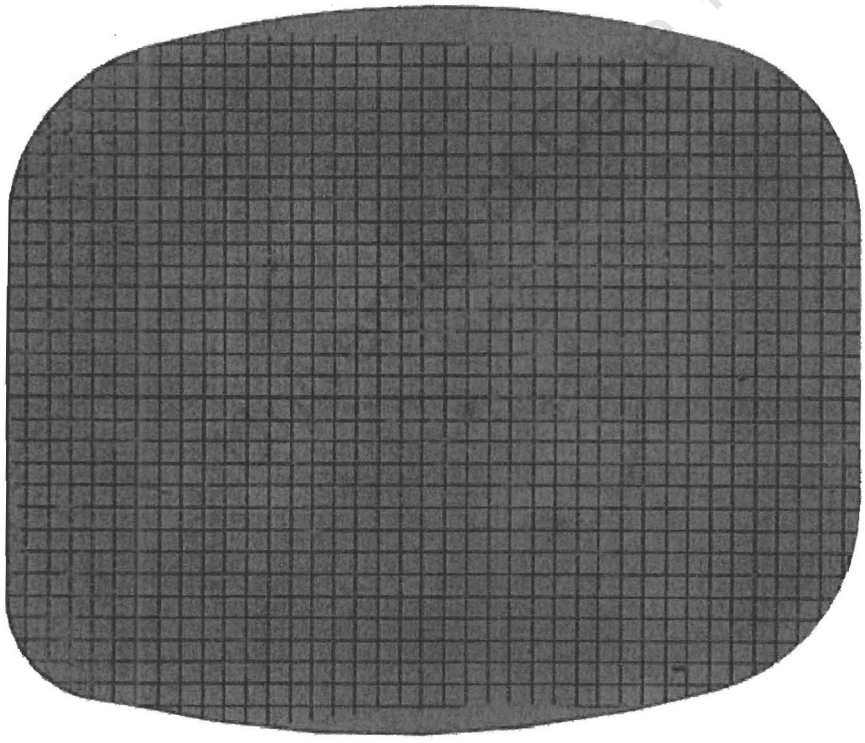

Figure 5.3 : Laser cut blank with acid etched rectangular pattern

For a deep drawn part, with strain's up to 20%, the possible measured results using the resolution of the line thickness are 0%, 10% and 20% strain. This is

not a large variation, and correlation with theoretical results is dubious. For example, a region of the part with theoretical strains of 20 to 25% would correspond to a measured strain of 20%. Indeed, if the majority of the part showed variations in strain of only 10%, the measured strains would be one of two values which bracketed the actual strain value. Although there may be a correlation between experimental and theoretical results on a macro scale, the resolution is too low to draw meaningful conclusions.

5.8 SolidWorks solid model

The final SolidWorks solid geometry for the die tool is shown in Figure 5.4. The inner radius is shown in red. The edge shown in green is the edge used to create the inner limit of the binder geometry. This edge marks the limit of the inside surface of the binder which comes into contact with the blank.

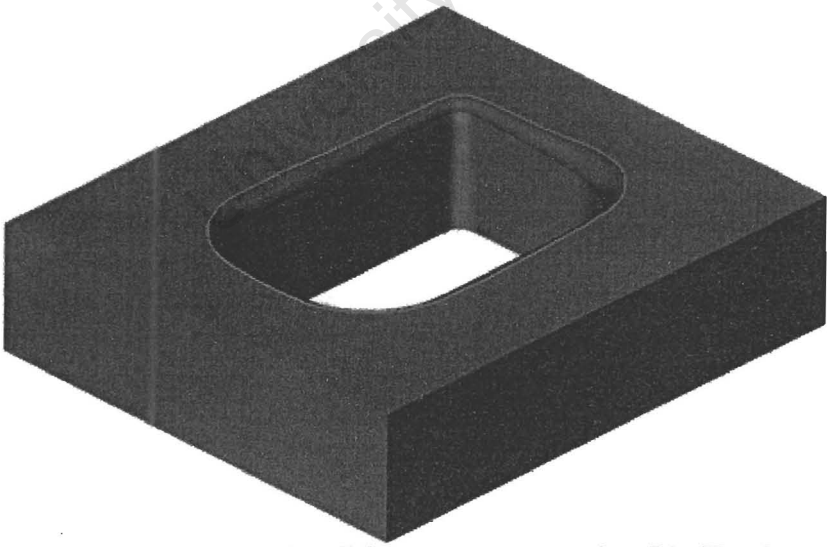

Figure 5.4 : Final solid geometry for the Die Tool

The various components were imported into SolidWorks, and then arranged in an assembly. The assembly was then exported in ACIS format (SAT file), ready for import into ABAQUS/CAE version 6.2-1.

5.9 ABAQUS problem description setup

The ACIS assembly file was imported into ABAQUS, which reads in each part as a separate part object, and retains the assembly position. Once the assembly was imported and the data surfaces were stitched together, no further geometric modelling was necessary.

5.9.1 Tooling

The tooling surfaces were configured to be rigid parts, therefore no deformation was allowed. While this is not strictly true, it is a fair assumption from a simulation viewpoint, as the tool surfaces are many times stiffer than the blank.

5.9.2 ABAQUS material model

The material model for the finite element code was needed to correctly model the material. An elastic-plastic material model with isotropic hardening was used.

The elastic part of the material is defined by the constants of Young's Modulus and Poisson's Ratio, which were 210×10^9 and 0.3 respectively.

The plastic part of the material curve is different for the two different material gauge thicknesses (0.6 mm and 0.8mm) is shown in Figure 5.5. The material curve for the 0.6 mm was obtained from Mr Jahajeeah[11] and has not been re-verified by experimental testing due to numerous assurances that the data was correct. The plastic part of the 0.8 mm curve was obtained from tensile testing done at the Centre for Materials Engineering at the University of Cape Town.

The tensile tests performed were done at various strain rates, with very little difference between results[12]. Therefore it was concluded that the material was fairly strain rate insensitive and there was no strain rate hardening law applied to the material, since this would have interfered with the material stiffness and consequently, the punch reaction force.

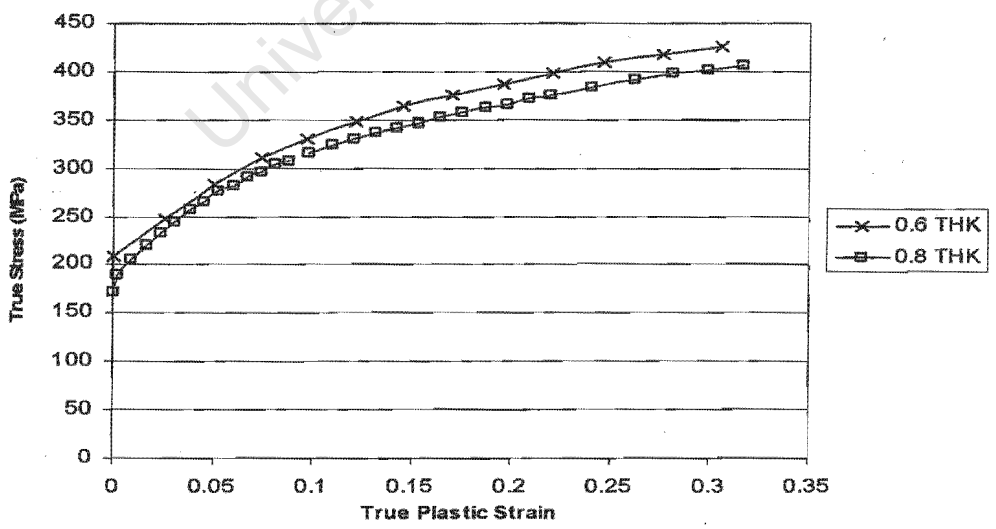

Figure 5.5 : Plastic flow curve for blank material

The forming process takes place within one second, and the maximum strain is 20 percent. The computed maximum strain rate is less than 0.5 per second and therefore the omission of rate dependence in the material model is considered acceptable.

Since the material data is derived from uni-axial tensile tests, there is no anisotropy within the model.

5.9.3 *Friction and contact*

A supplied lubricant resembling light oil was applied to both sides of the blank before the forming operation. However the exact friction value is unknown.

Typical friction values of 0.001 to 0.15 are used in simulations, when the majority of the work is done in plastic forming, and thus friction does not play as large a role as one would expect. However, in this instance, there is draw in of material under the binder, and friction is expected to play a role.

Three sets of contact were setup. Contact was declared between the die and the bottom surface of the blank, and between the top surface of the blank, and the binder and punch, respectively.

5.9.4 *Forming sequence*

Within the simulation the binder and die are brought into contact with the blank. The die is then fixed, and the binder pressure is then applied. The punch is then used to form the blank.

5.9.5 *Binder pressure*

The binder pressure was simulated by a constant pressure load, equivalent to the average feedback load on the binder during the experimental forming operation.

5.9.6 *Punch force*

The punch is used to form the blank by applying a velocity profile to the punch as a boundary condition. The reaction force on the punch is then monitored, and this corresponds to the punch Force obtained from the physical press.

5.9.7 *Monitoring of variables*

In addition to the standard stress and strain output, special outputs were requested for the feedback force on the punch, the energy histories of the blank, and the displacement and velocity curves of the punch.

5.9.8 *Input deck issues*

The file that the ABAQUS solver reads describes the problem and is called an input deck, or input file. An example is shown as Appendix E with the node and element data removed and replaced with

*include statements (shown in red). Currently, ABAQUS/CAE v6.2-1 does not allow the creation of mass elements. These elements are needed to give the rigid tooling elements mass in the finite element simulation. Thus the simulation cannot be run from within ABAQUS CAE, but instead needs to be exported as an input deck and then modified to add the mass elements for the tooling.

Once the input deck was modified, the ABAQUS solver was run manually with the input deck. To help with running several jobs in succession (a process known as batching), a Perl program was written which executed jobs in succession.

5.10 Post processing

ABAQUS macros were written in Python which allowed the punch force history and Model Energy history to be extracted from the data file. This data could then be imported into Microsoft Excel and the values graphed on the same graph as the Experimental values.

In addition to this, ABAQUS/Viewer was used to obtain the final blank shape after forming.

6 Correlation of Simulation with Experiments

6.1 Introduction

The simulation of a metal forming process within an explicit finite element code is relatively straightforward, provided that any numerical trickery employed to speed up the simulation does not have an adverse effect on the results.

In order to ascertain the effect of various parameters, the forming process first needed to be modelled in such a way that it correlated with the (assumed) quasi-static process of metal forming. Thus the energy history of the model is studied to determine that the process is indeed quasi-static.

6.2 Energy histories

Typically, the determination of whether a process is quasi-static or not, is determined by whether dynamic effects play a role. This, if the internal and kinetic energy of the metal blank is studied through the forming process, there should not be an excessive transfer of energy from the punch (which does the "work" on the blank) to the blank in the form of kinetic energy.

Thus the blank should have very little kinetic energy at the end of the forming process, as all the energy will have been used to form the blank, thus increasing its internal energy. This is used as an indication of the validity of the forming simulation.

In metal forming texts[13], a typical value of internal energy vs kinetic energy is quoted as 5 %. A graph of internal energy and kinetic for the metal forming process is shown below for Experiment 9. The initial Internal energy is due to the contact of the binder.

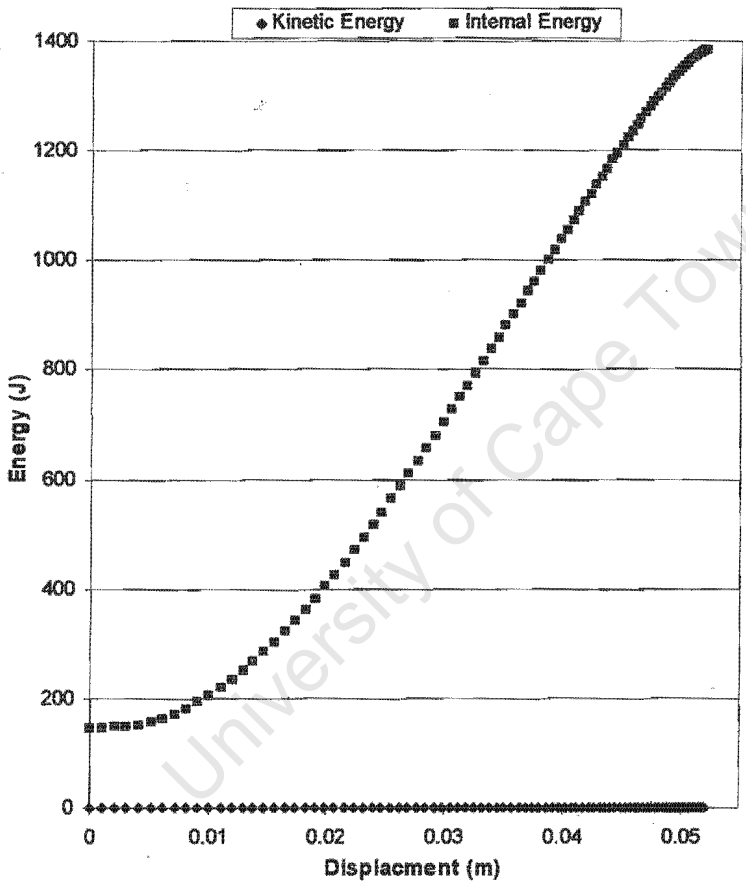

Figure 6.1 : Typical Internal Energy vs Kinetic Energy for Forming Step

Since the energy values are compared on the same scale, no variation in the Kinetic energy can be seen. The Kinetic Energy for the forming step is shown in Figure 6.2

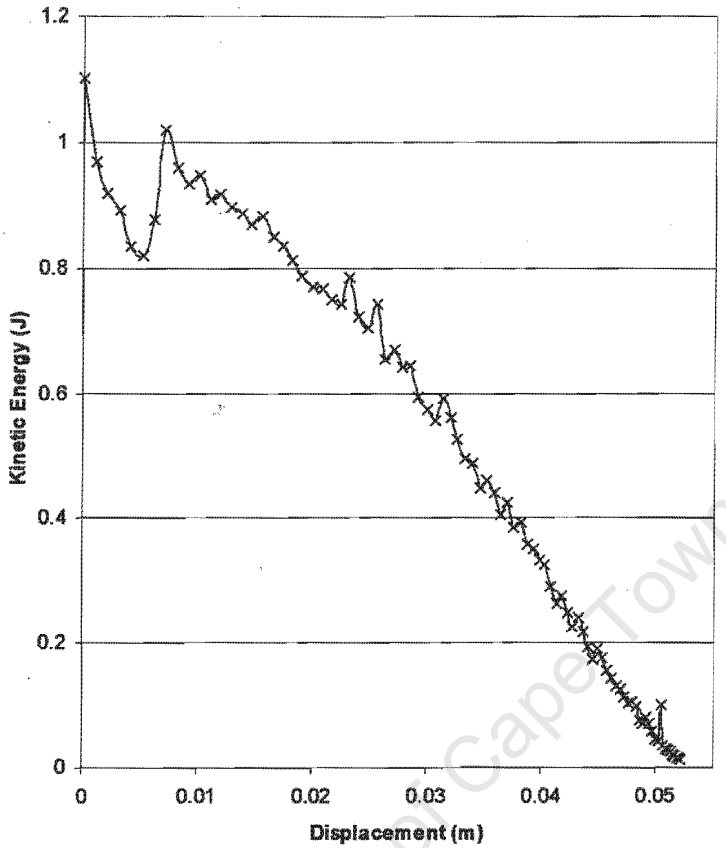

Figure 6.2 : Kinetic Energy for the forming step

The energy is initially high, due to impact with the tooling, but then decreases as the punch slows down to zero velocity at the end of the forming step.

The displacement vs. time graph is second order (constant deceleration of the punch) and this translates to the non-linear horizontal axis seen in the above two graphs. This is evidenced by the increase in data point density towards the end of the step, as the data values are written out at constant time steps, and then graphed vs. displacement.

6.3 Displacements, velocities and accelerations for tooling

The boundary conditions used to drive the simulation are invoked by applying a velocity profile to the tooling. Attempts were made to drive the process using displacement boundary conditions, but due to the algorithm implementation within ABAQUS, the punch force history is incorrect. This is because ABAQUS uses large accelerations to enforce the discrete displacement vs. time profile. Since force = mass x acceleration, these accelerations shows up as large positive and negative forces within the punch force history to the extent that any useful data is obliterated. This is hinted at by the ABAQUS manuals which state that one should use velocity as an applied boundary condition. When velocity is used as an applied boundary condition, the spikes in the data disappear.

6.3.1 Binder and die boundary conditions

The binder and die were moved equally to bring them into contact with the blank in the initial step. The binder was then fixed in space for the second step, and the velocity boundary condition on the binder was replaced with a force. The force equivalent to the binder pressure was calculated using the known binder loadcell histories which were then summed and averaged over the eight binder pistons for the forming operation.

6.3.2 Punch displacement vs. time profile

The actual recorded punch vs. time displacement profile is shown in Figure 4.5. However, since the displacement profile was not

correctly recorded due to a faulty displacement transducer, comparisons against the simulation data cannot be undertaken. However, the inferred displacement vs. time data from prior experiments can be compared.

Since the simulated time used to form the blank during an explicit simulation is very much less than the actual time, it is not useful to graph the displacement histories for both simulation and experiment on the same set of axes.

The simulation displacement vs. time history of the punch for the forming step is shown in Figure 6.3 below.

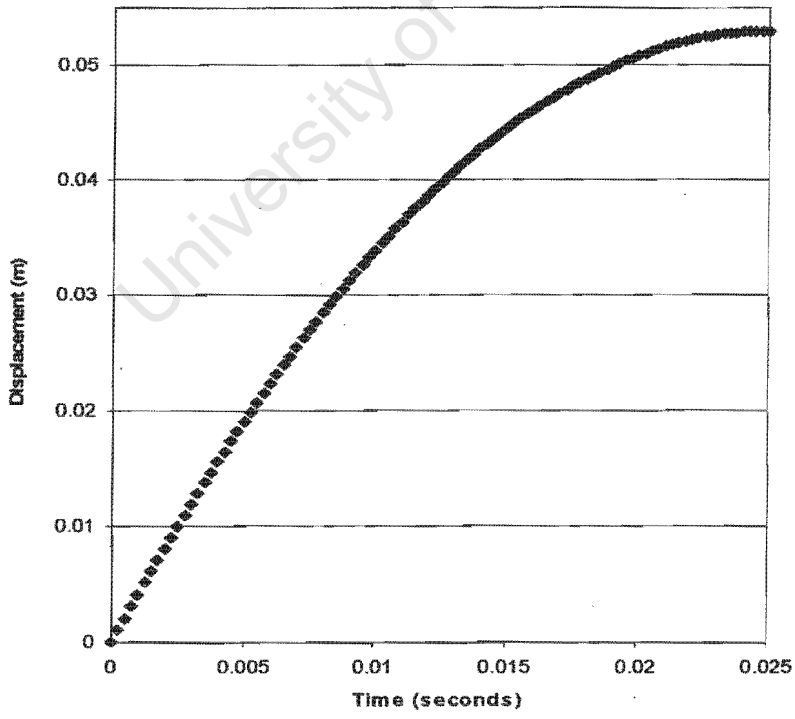

Figure 6.3 : Simulation punch displacement vs. time forming step history

It can be seen from Figure 6.3 that the total punch travel is slightly greater than two inches. This is because the punch is not initially in contact with the blank, and must travel slightly before contact occurs.

6.3.3 *Punch velocity vs. time profile*

As can be seen from the Figure above, the punch displacement is second order, and therefore the velocity of the punch with respect to time is said to be first order or linear.

This can be seen below in Figure 6.4 below, which shows the punch velocity vs. time profile. The initial velocity of the punch is approximately 5 m/s which is consistent[13] with other information on the simulation of metal forming.

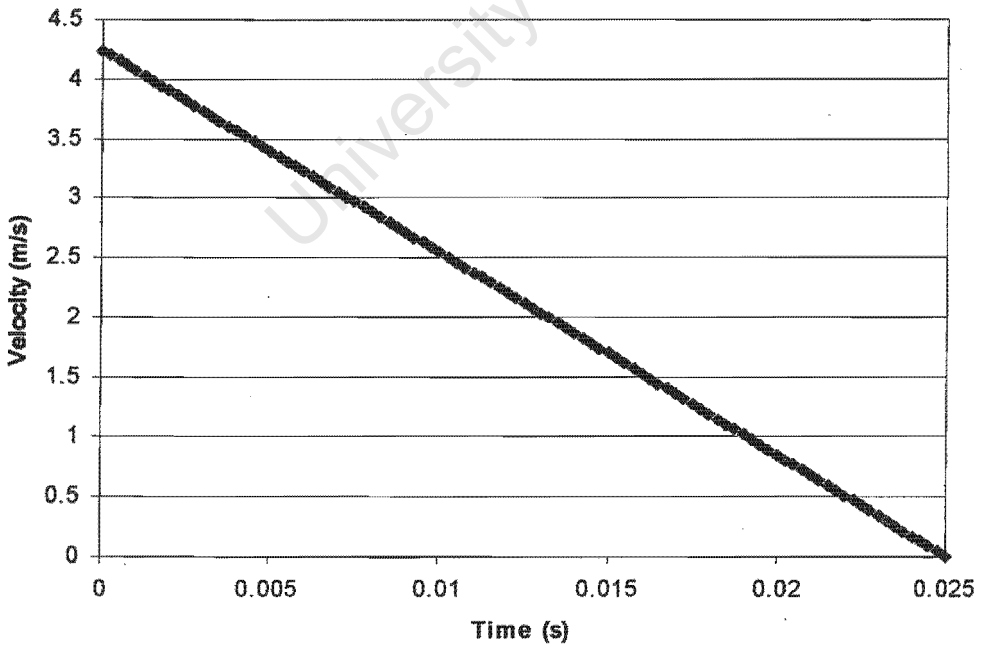

Figure 6.4 : Simulation punch velocity vs. time for forming step

6.3.4 *Punch acceleration vs. time profile*

Given that the punch velocity vs. time profile is linear, the deceleration of the punch will be constant.

6.4 Experimental groupings

The experiments are broken up into natural grouping by material thickness, and then by binder pressure. This is illustrated in the table below by colour groupings.

Table 6.1 : Experimental groupings colour coded

Exp. No.	Depth (in)	Load/Left (tons)	Load/Right (tons)	BHF pressure (psi)	Blank Thick. (mm)	Speed (SPM)
1	2			500	0.6	12
2	2	3.5	3.7	1000	0.6	12
3	2	5.0	5.5	1000	0.8	12
4	2	5.0	5.5	1000	0.8	18
5	2	5.0	5.5	1000	0.8	18
6	2	5.0	5.5	1000	0.8	18
7	2	3.7	4.0	1500	0.6	12
8	2	5.2	5.7	1500	0.8	12
9	2	3.7	4.0	2000	0.6	12
10	2	4.0	4.2	2000	0.6	12
11	2	4.0	4.2	2000	0.6	12
12	2	5.5	6.0	2000	0.8	12
13	2	5.5	6.0	2000	0.8	18
14	2	4.2	4.5	2500	0.6	12
15	2	5.5	6.2	2500	0.8	12

The experiments shown in Table 6.1 correspond to a broad variation of parameters along with repetition of various experiments for experimental result repeatability.

For example; from the table above; it can be seen that Experiments 4, 5 and 6 are all at 1000 psi (6.9 MPa) binder pressure with a 0.8 mm thick blank,

while experiment 3 is a 0.6 mm thick blank at 1000 psi (6.9 MPa) binder pressure.

6.5 Punch force

With the absence of displacement information to point to the start and end of the forming stroke, one must rely on the sudden increase in the reaction force on the punch as an indication of the start of the forming operation. This is not completely correct, as there is a slight delay from the moment of contact, until the reaction force on the punch rises. However, this delay results in a small shift in the data on the horizontal (time) axis, and does not affect the peak punch force, which is normally the object of interest within a simulation.

There is a tendency within the literature to measure punch force vs. displacement, since this measure is then devoid of time, and both experimental and simulation data may be presented on the same set of axes. This approach has been following in this work.

6.5.1 Experimental repeatability

In order to compare the experimental results with simulation, the first concern is that the experimental results are repeatable. In light of this, various experiments were repeated.

Experiments 3, 4, 5 and 6 were compared to each other. This is shown in Figure 6.5 below. As can be seen from the figure, the experimental results are almost exactly the same, even though

experiment 3 was conducted at a press speed of 12 strokes per minute (SPM), and experiments 4, 5, and 6 were conducted at 18 SPM. This indicates that the material is not rate-sensitive over the limited forming speeds available on the Press.

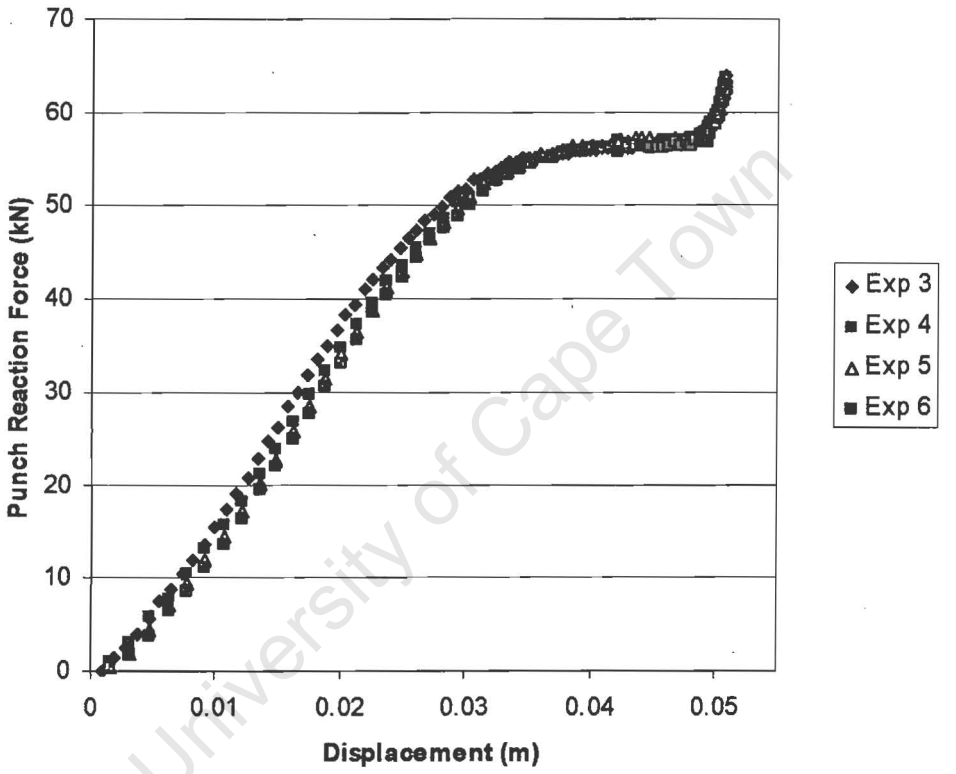

Figure 6.5 : Experimental Repeatability - 1000 psi BHF and 0.6 mm THK

This process was repeated for the 0.6 mm thick blank material, in order to ascertain that the Press could cope with both the thick material, as well as the thinner material in a consistent manner. This is shown in Figure 6.6.

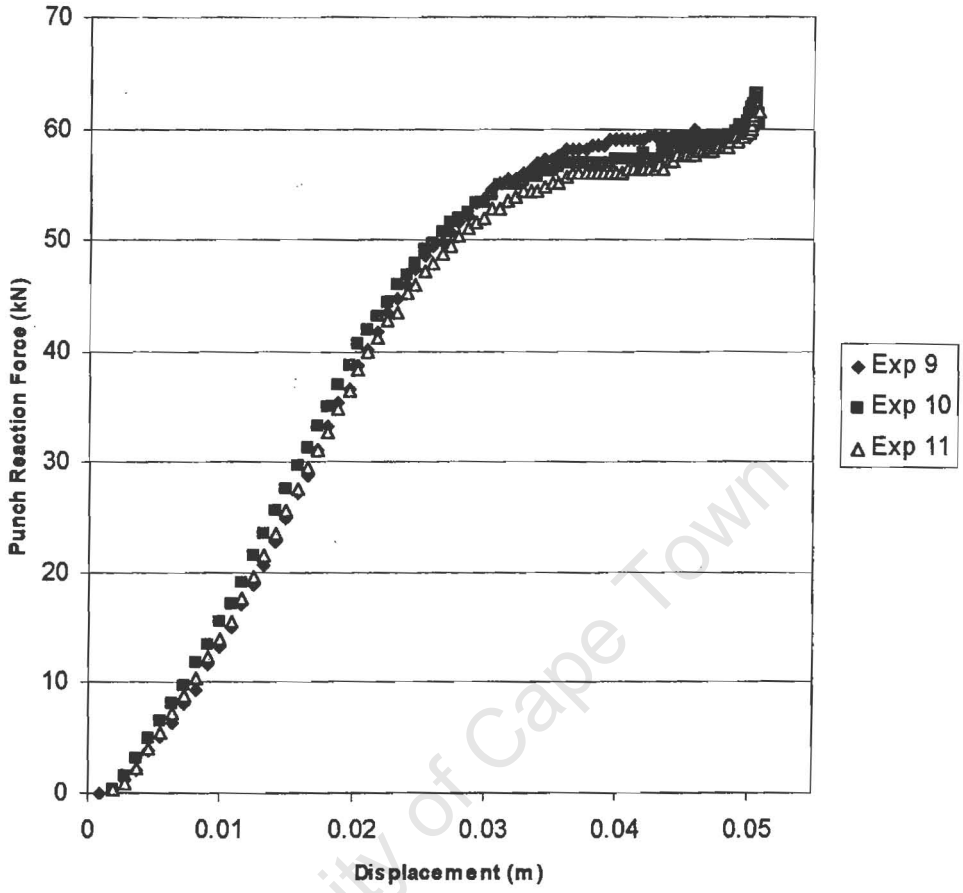

Figure 6.6 : Experimental Repeatability - 2000 psi BHF and 0.8 mm THK

As can be seen from the figures, the experimental results are repeatable with differences in the order of less than 5% in the area of peak punch force.

Once the experimental results have been verified reproducible, the next objective is to determine the friction coefficient.

6.5.2 *Variations in friction coefficient*

The friction model is a simple Coulomb law, and the friction coefficient was varied between 0.001 (0.1 %) and 30 percent. This is seen below in Figure 6.7.

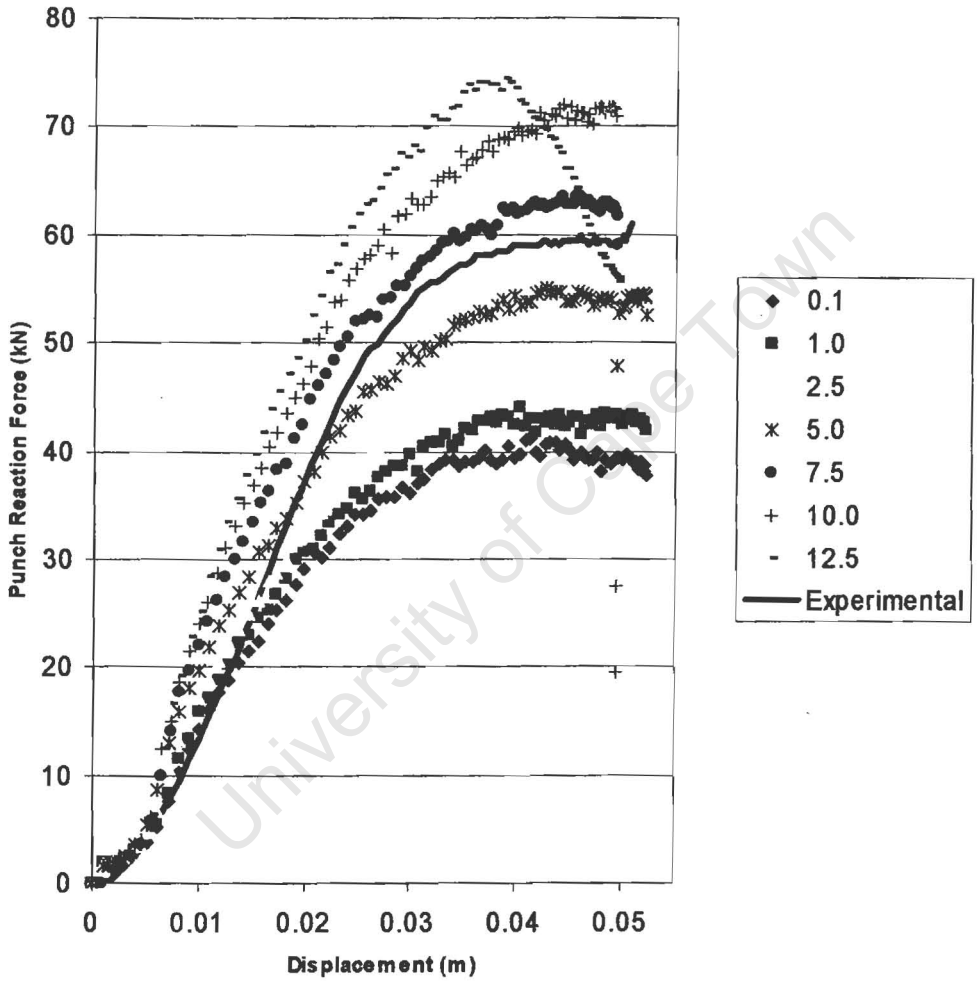

Figure 6.7 : Variations in friction percentage for Experiment 9

A search of the available literature indicated that a well lubricated steel forming process has a friction value of less than 15%. This is evidenced in the above figure, as the friction value appears to lie between 5.0 and 7.5 percent.

This can be easily demonstrated by changing the value of the friction co-efficient to 6% and comparing the ABAQUS results to the experimental results for Experiment 9. This is shown below in Figure 6.8

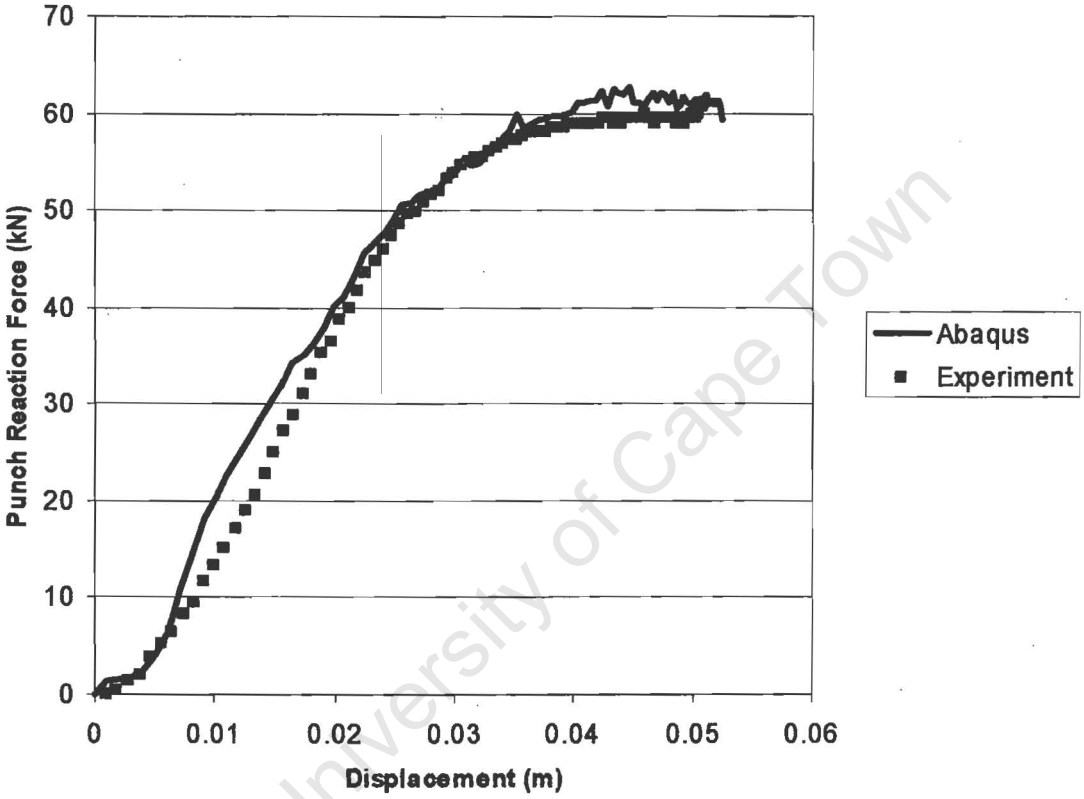

Figure 6.8 : Experiment 9 with friction coefficient set to 6%

The variation in friction values was repeated for Experiment 3 which uses the 0.6mm thick blank.

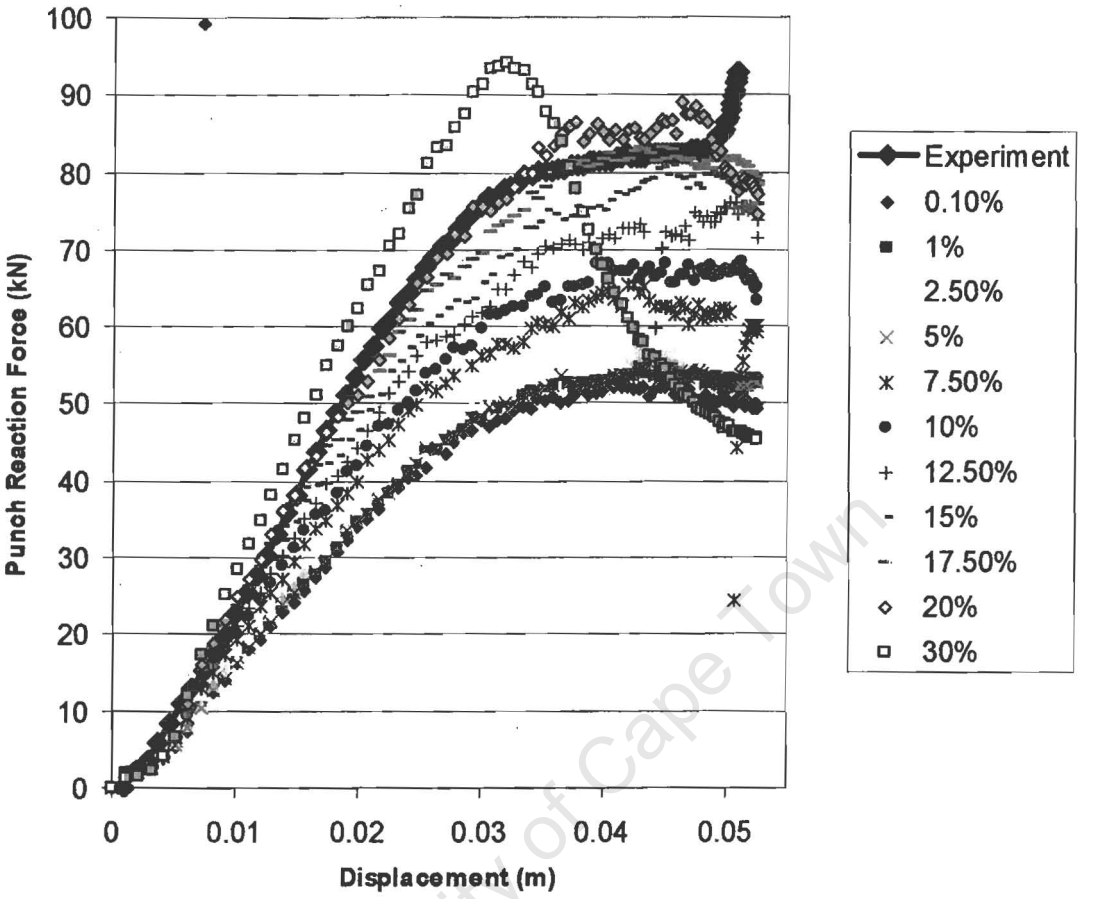

Figure 6.9 : Variation in friction percentage for Experiment 3

However, it is noted that the appropriate friction coefficient is 17.5%. Since it is expected that not only should the friction value be the same for both sets of experiments, but also that the friction value should be under 10%, it was concluded that the supplied material flow curve for 0.6mm material is incorrect.

6.6 Binder load

The binder cylinder pressure along with the feedback load on each of the eight binder cylinders was monitored.

6.6.1 *Binder cylinder pressure*

The cylinder pressure data is converted from volts to psi, but this information cannot be used to determine the binder force without knowledge of the piston geometry. However, it is of interest that the hydraulic cylinder pressure is stable, and this is shown in Figure 6.10 below.

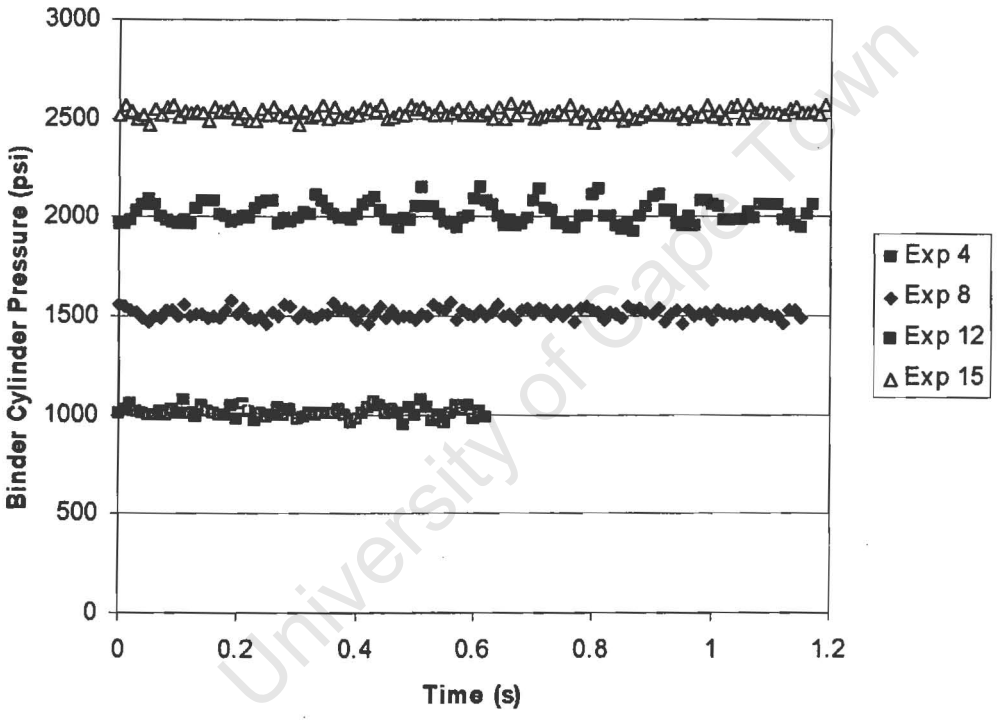

Figure 6.10 : Binder Cylinder Pressure stability

Experiment 4 shows less data points since it corresponds to an 18 SPM cycle, while the other three experiments correspond to 12 SPM cycles. Experiment 12 shows fluctuations in the pressure, and this was evidenced by a large amount of vibration on the Press during the forming process.

6.6.2 *Binder cylinder feedback*

Since excessive vibration in the binder cylinder pressure may affect the final shape of the formed part, the feedback on the binder cylinders was examined in more detail. The feedback for Experiment 9 is shown below.

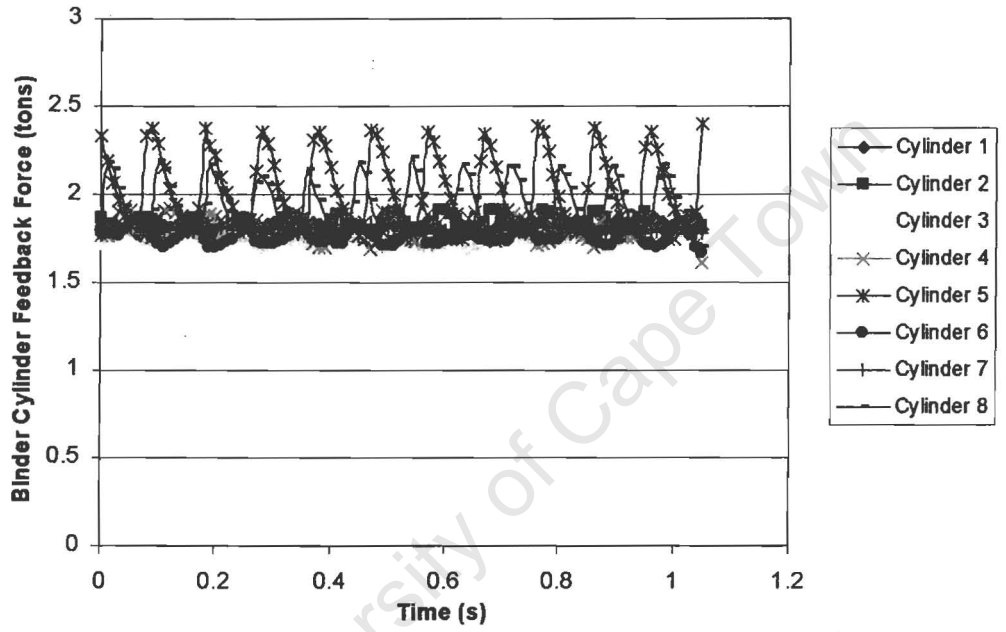

Figure 6.11 : Feedback Force on Binder Cylinders for Experiment 9

The feedback force was also used to determine the load to be applied to the binder in the ABAQUS simulations. This was done by summing the average feedback load over the forming process for all eight cylinders and then converting it from volts to tons using the scaling factors.

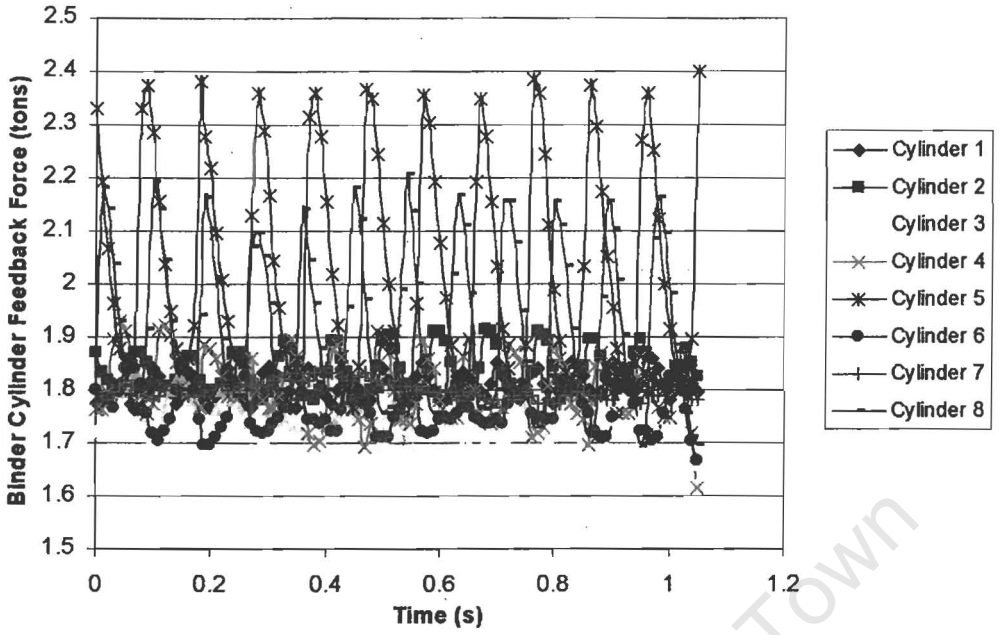

Figure 6.12 : Feedback Force on Binder Cylinders for Experiment 9

The results shown in the above figure are typical. Cylinders 5 and 8 show large fluctuations in load. It is unknown what causes this. This is better evidenced if the vertical axis is cropped to show only the section across which data occurs as is shown below in Figure 6.12.

6.7 Drawn-in shape

The resultant experimental formed pieces were photographed, and their outlines were scaled and superimposed about their centres to determine the difference in final shape for experimental groupings of the same parameters. This is shown in the figure below.

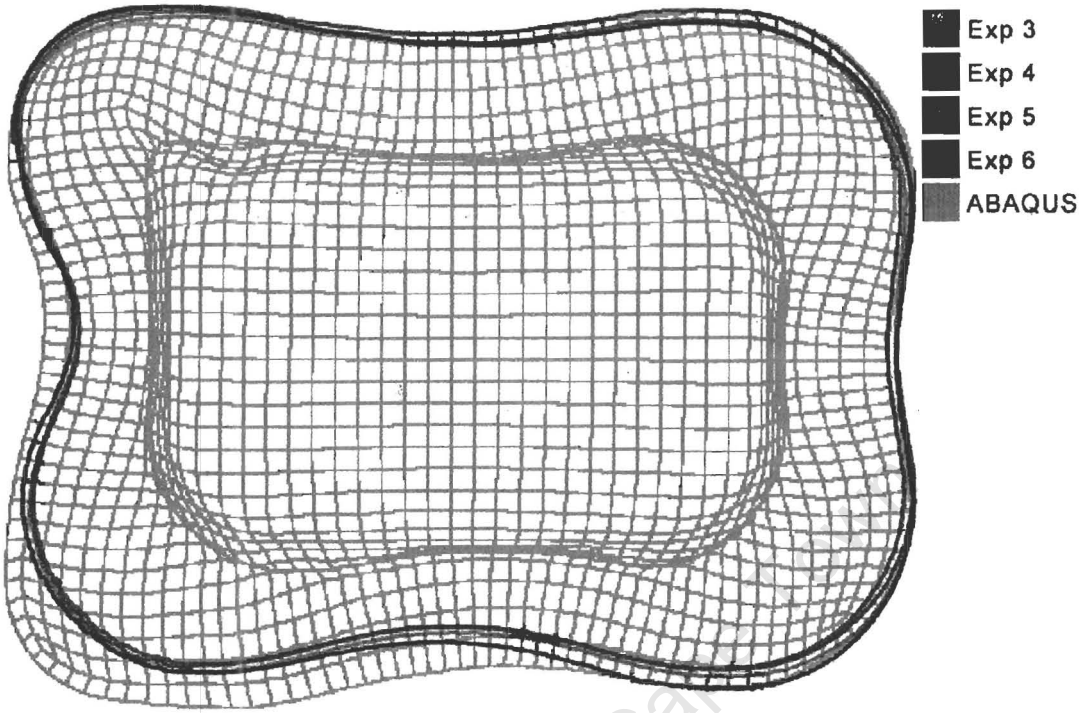

Figure 6.13 : Part outline for Experiments 3, 4, 5 and 6

As can be seen, the outlines correlate well for the experimental results, but the ABAQUS results are different. However it is suspected that this may be due to material anisotropy which was not modeled in the simulation. The Figure above is for Experiments 3, 4, 5, 6 which contains both 12 and 18 SPM forming operations for 0.8 mm thick material. The ABAQUS result shown above was obtained for a friction coefficient of 6%, for which the punch force corresponds well with experiment.

7 Conclusions

The ABAQUS results correlated well with the experimental results for the 0.8 mm thick material and the friction value is within the expected range. The 0.6 mm material does not correlate well, and it is concluded that the material data which was originally supplied by the University of Michigan is incorrect.

The experimental data exhibits good consistency even across experiments where the forming speed was different. The final shapes of the parts are similar, and the ABAQUS simulation produces a part that is reasonably close to the final shape.

The laser cut blank's used for forming were acid etched with a rectangular pattern to allow the experimental determination of the final strain for comparison with the simulation. However, the accuracy of the measured strain is almost the same as the strain itself, so the value of any such strain measurements is controversial.

The reaction force on the punch is compared with the experimentally determined values and correlates well, which would allow the correct selection of a Press capacity.

8 References

1 A. Honecker, K. Mattiasson, *Finite element procedures for 3-D sheet forming simulation*, Numiform, 89, Balkema, Rotterdam, 1989, pp. 456-463. in Narkeeran Narasimhan, Michael Lovell, *Predicting springback in sheet metal forming: an explicit to implicit sequential solution procedure*, Finite Elements in Analysis and Design 33 (1999) 29-42

2 J.C. Nagtegaal, L.M. Taylor, *Comparison of implicit and explicit finite element methods for analysis of sheet forming problems, in: FE-Simulation of 3-D Sheet Metal forming Processes in Automotive Industry*, VDI, Berichte, vol. 894, 1991, pp. 705-725. in Narkeeran Narasimhan, Michael Lovell, *Predicting springback in sheet metal forming: an explicit to implicit sequential solution procedure*, Finite Elements in Analysis and Design 33 (1999) 29-42

3 T Belytschko, W Kam Liu, B Moran, *Nonlinear Finite Elements for Continua and Structures,*

4 Z.M. Hu *, T.A. Dean, *A study of surface topography, friction and lubricants in metalforming*, International Journal of Machine Tools & Manufacture 40 (2000) 1637–1649

5 A.G. Mamalis *, D.E. Manolakos, A.K. Baldoukas, *Simulation of sheet metal forming using explicit finite element techniques: effect of material and forming characteristics Part 2. Deep-drawing of square cups*, Journal of Materials Processing Technology 72 (1997) 110–116

6 Li-Ping Lei, Sang-Moon Hwang, Beom-Soo Kang, *Finite element analysis and design in stainless steel sheet forming and its experimental comparison*, Journal of Materials Processing Technology 110 (2001) 70-77

7 M. Kawkaa, L. Olejnik, A. Rosochowski, H. Sunaga, A. Makinouchi, *Simulation of wrinkling in sheet metal forming*, Journal of Materials Processing Technology 109 (2001) 283 - 289

8 E Docge, T El-Dsoki, D Seibert, *Prediction of necking and wrinkling in sheet metal forming*, Journal of Materials Processing Technology 50 (1995) 197-206

9 Xi Wang, Jian Cao, *On the prediction of side-wall wrinkling in sheet metal forming processes*, International Journal of Mechanical Sciences 42 (2000) 2369}2394

10 J. B,. Kim, J. W. Yoon, D. Y. Yang, F Barlat, *Investigation into wrinkling behaviour in the elliptical cup deep drawing process by finite element analysis*

using bifurcation theory. Journal of Materials Processing Technology 111 (2001) 170-174

11 Jahajeeah, N, *Analysis of drawbeads for sheet metal forming: Numerical and Experimental Aspects*. MSc Dissertation, University of Cape Town 2000.

12 Professor R. Knutsen, Department of Mechanical Engineering, University of Cape Town. Private communications, 2001.

13 Hibbett Karlsson and Sorenson, Inc, *ABAQUS User manuals v6.2 (2001)*.

Appendix A – Perl program to create DXF data file

```
#!/usr/bin/perl
#
# Input : file with x,y,z data points (raw CMM data)
# Output: DXF file with data points

$punchfile='punch_c1.txt';
$writefile='output.dxf';

open (NUMBERS, "<$punchfile") || die "Help, no punchfile!\n";
open (OUTPUT, ">$writefile") || die "Help, cannot write to
file!\n";

print OUTPUT "  0\nSECTION\n  2\nENTITIES\n  0\n";
$i=30;
while (<NUMBERS>) {
      ($x,$y,$z) = split("\t",$_);
      print OUTPUT "POINT\n  8\nDEFAULT\n  5\n";
      printf OUTPUT "%X",$i;
      print OUTPUT "\n 10\n";
      $i = $i+1;
      print OUTPUT $x,"\n 20\n",$y,"\n 30\n",$z,"\n  0\n";
      }
print OUTPUT "ENDSEC\n  0\nEOF";

close NUMBERS;
close OUTPUT;
```

Appendix B – Raw Experimental data sample

Raw data file (first 10 lines)
First line is the number of samples per second (ie: 100/sec).

100.000

8.101	0.879	0.000	0.000	-0.005	0.376	0.322	0.337	0.337	0.459	0.352	0.337	0.396
8.125	0.898	-0.005	0.000	-0.005	0.361	0.308	0.352	0.303	0.459	0.356	0.337	0.386
8.594	0.889	0.000	0.000	-0.005	0.376	0.317	0.317	0.288	0.459	0.356	0.337	0.376
8.564	0.894	-0.005	0.000	-0.005	0.376	0.317	0.303	0.278	0.469	0.361	0.342	0.376
8.569	0.864	-0.005	-0.005	-0.005	0.376	0.317	0.283	0.278	0.474	0.352	0.337	0.376
8.271	0.889	-0.005	-0.005	-0.005	0.376	0.322	0.288	0.293	0.469	0.352	0.337	0.391
7.881	0.884	-0.005	0.000	-0.010	0.381	0.322	0.308	0.347	0.454	0.352	0.342	0.396
8.242	0.869	-0.005	-0.005	-0.005	0.371	0.327	0.347	0.361	0.439	0.352	0.332	0.386
8.076	0.889	-0.005	0.000	-0.005	0.371	0.322	0.342	0.356	0.449	0.352	0.337	0.376
8.101	0.898	0.000	0.000	-0.005	0.371	0.327	0.322	0.322	0.464	0.361	0.337	0.376
8.418	0.879	-0.005	-0.005	-0.005	0.376	0.322	0.303	0.298	0.474	0.361	0.342	0.376

Data headings:

Description	Punch Position	Cylinder Pressure	Load/Left	Load/Right	Load Punch	Load Cylinder 1 .. 8
Units	Volts	Volts	Volts	Volts	Volts	Volts

Conversion factors are applied by the macro in Appendix C

Appendix C – Visual Basic Macro to Process Data

```
Sub ProcessData()
' ---------------------------------------------------------------------------
-
' Macro to process raw data files.
' ---------------------------------------------------------------------------
-' variables
experi = 6       ' experiment number
speed$ = "18"    ' SPM  <-- CAVEAT EMPTOR: make this 12 or 18
' ---------------------------------------------------------------------------
-filen$ = "rich0" & Trim(Str(experi))
outputfile$ = "exp" & Trim(Str(experi)) & ".xls"

' filtering vars
before = 4
after = 8

'open file
    ChDir "D:\umich\data\raw"
    file$ = filen$ + ".TXT"
    Sheetname$ = Left$(file$, (Len(file$) - 4))
    tmp$ = "D:\umich\data\raw\" & file$
    Workbooks.OpenText Filename:=tmp$, Origin:= _
        xlWindows, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
        xlDoubleQuote, ConsecutiveDelimiter:=True, Tab:=True, Semicolon:=True, _
        Comma:=True, Space:=True, Other:=False, FieldInfo:=Array(Array(1, 1), _
        Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7,
1), Array(8, 1), _
        Array(9, 1), Array(10, 1), Array(11, 1), Array(12, 1), Array(13, 1)), _
        TrailingMinusNumbers:=True
    Selection.ClearContents
    Windows(file$).Activate

' debug stuff
'   MsgBox (Sheetname$)
' ---------------------------------------------------------------------------
-' Remove extra data
' ---------------------------------------------------------------------------
-' calc end first
    duh$ = Selection.SpecialCells(xlCellTypeLastCell).Address(False, False)
    x = Val(Right(duh$, Len(duh$) - 1))
' check for it going above 0.02
' this is the start of the stroke
    maxi = 0.02
    For y = 1 To x
        curr = Cells(y, 5).Value
        If (curr > maxi) Then
            ymax = y
            Exit For
        End If
    Next y
 startpt = ymax - before
' now check for the end
' we want the maximum point + a bit
   mini = 0
   For y = ymax To x
        curr = Cells(y, 5).Value
        If (curr < mini) Then
            ymax = y + 1
            Exit For
        End If
   Next y
 endpt = ymax - after
' select and delete before and after
' last part first
z$ = Trim(Str(endpt)) & ":" & Trim(Str(x))
Rows(z$).Delete Shift:=xlUp
z$ = "2:" & Trim(Str(startpt))
Rows(z$).Delete Shift:=xlUp
  ---------------------------------------------------------------------------
-' Create Headings
' ---------------------------------------------------------------------------
-    Range("A1").FormulaR1C1 = "Punch Position"
    Range("B1").FormulaR1C1 = "Cylinder Pressure"
```

```
    Range("C1").FormulaR1C1 = "Load/Left"
    Range("D1").FormulaR1C1 = "Load/Right"
    Range("E1").FormulaR1C1 = "Load/Punch"
    Range("F1").FormulaR1C1 = "Load/Cylinder1"
    Range("G1").FormulaR1C1 = "Load/Cylinder2"
    Range("H1").FormulaR1C1 = "Load/Cylinder3"
    Range("I1").FormulaR1C1 = "Load/Cylinder4"
    Range("J1").FormulaR1C1 = "Load/Cylinder5"
    Range("K1").FormulaR1C1 = "Load/Cylinder6"
    Range("L1").FormulaR1C1 = "Load/Cylinder7"
    Range("M1").FormulaR1C1 = "Load/Cylinder8"
' ------------------------------------------------------------------------------
'
' Create and order sheets
' ------------------------------------------------------------------------------
'
    Sheets(Sheetname$).Name = "Raw Data"
    Sheets.Add
    Sheets("Sheet1").Name = "Displacement"
    Sheets.Add
    Sheets("Sheet2").Name = "Punch Force"
    Sheets.Add
    Sheets("Sheet3").Name = "Binder Pressure"
    Sheets.Add
    Sheets("Sheet4").Name = "Binder Load Cells"
    Sheets.Add
    Sheets("Sheet5").Name = "Load Left"
    Sheets.Add
    Sheets("Sheet6").Name = "Load Right"
    Sheets("Punch Force").Move before:=Sheets(1)
    Sheets("Raw Data").Move before:=Sheets(1)
    Sheets("Load Right").Move before:=Sheets(5)
' ------------------------------------------------------------------------------
'
' Copy data to sheets
' ------------------------------------------------------------------------------
'
    Sheets("Raw Data").Select
    Columns("E:E").Select
    Selection.Copy
    Sheets("Punch Force").Select
    Range("C1").Select
    ActiveSheet.Paste
    Call writedisp(speed$)
    Sheets("Raw Data").Select
    Columns("C:C").Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Load Left").Select
    ActiveSheet.Paste
    Sheets("Raw Data").Select
    Columns("D:D").Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Load Right").Select
    ActiveSheet.Paste
    Sheets("Raw Data").Select
    Columns("B:B").Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Binder Pressure").Select
    Application.CutCopyMode = False
    Sheets("Binder Pressure").Move before:=Sheets(5)
    Sheets("Raw Data").Select
    Selection.Copy
    Sheets("Binder Pressure").Select
    ActiveSheet.Paste
    Sheets("Raw Data").Select
    Range("A1").Select
    Application.CutCopyMode = False
    Sheets("Punch Force").Select

' ------------------------------------------------------------------------------
'
' Punch force sheet
' ------------------------------------------------------------------------------
'
```

```
    Range("E1").FormulaR1C1 = "slope"
    Range("F1").Value = 9.638554217              ' pre-calc'ed average value
    Range("D2").Select
    ActiveCell.FormulaR1C1 = "=RC[-1]*R1C6*9810"
    Range("D2").Select
    Selection.Copy
    tmp$ = "D3:D" & Trim(Str(endpt - startpt))
    Range(tmp$).Select
    ActiveSheet.Paste
    Range("A1").Select
    Application.CutCopyMode = False
' calc range size
    If (speed$ = 12) Then
        t = 106
    Else
        t = 63
    End If
' add chart
    Charts.Add
    ActiveChart.ChartType = xlXYScatterSmooth
    ActiveChart.SetSourceData Source:=Sheets("Punch Force").Range("A1:F122"), _
        PlotBy:=xlColumns
    ActiveChart.SeriesCollection(1).Delete
    ActiveChart.SeriesCollection(1).Delete
    ActiveChart.SeriesCollection(1).Delete
    ActiveChart.SeriesCollection(1).Delete
    ActiveChart.SeriesCollection(1).Delete
    ActiveChart.SeriesCollection(1).Delete
    ActiveChart.SeriesCollection.NewSeries
    tmp$ = "B2:B" + Trim(Str(t + 1))
    ActiveChart.SeriesCollection(1).XValues = Worksheets("Punch
Force").Range(tmp$)
    tmp$ = "D2:D" + Trim(Str(t + 1))
    ActiveChart.SeriesCollection(1).Values = Worksheets("Punch Force").Range(tmp$)
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Punch Force"
    With ActiveChart
        .HasTitle = False
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Displacement
(m)"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Punch Load (N)"
    End With
    ActiveChart.HasLegend = False
    ActiveSheet.Shapes("Chart 1").IncrementLeft 18#
    ActiveSheet.Shapes("Chart 1").IncrementTop -954.75
    ActiveSheet.Shapes("Chart 1").ScaleWidth 1.02, msoFalse, msoScaleFromTopLeft
    ActiveSheet.Shapes("Chart 1").ScaleHeight 1.91, msoFalse, msoScaleFromTopLeft
    ActiveChart.PlotArea.Select
    With Selection.Border
        .Weight = xlThin
        .LineStyle = xlNone
    End With
    Selection.Interior.ColorIndex = xlNone
    ActiveChart.ChartArea.Select
    ActiveWindow.Visible = False
    Windows(file$).Activate
    Range("E13").Select
    ActiveSheet.ChartObjects("Chart 1").Activate
    ActiveChart.ChartArea.Select
    With Selection.Border
        .Weight = 2
        .LineStyle = 0
    End With
    Selection.Interior.ColorIndex = xlAutomatic
    Sheets("Punch Force").DrawingObjects("Chart 1").RoundedCorners = False
    Sheets("Punch Force").DrawingObjects("Chart 1").Shadow = False
    ActiveChart.PlotArea.Select
    With Selection.Border
        .Weight = xlThin
        .LineStyle = xlAutomatic
    End With
    Selection.Interior.ColorIndex = xlNone
    ActiveChart.ChartArea.Select
    ActiveChart.Axes(xlValue).Select
    With ActiveChart.Axes(xlValue)
        .MinimumScaleIsAuto = True
```

```
        .MaximumScaleIsAuto = True
        .MinorUnitIsAuto = True
        .MajorUnitIsAuto = True
        .Crosses = xlAutomatic
        .ReversePlotOrder = False
        .ScaleType = xlLinear
        .DisplayUnit = xlThousands
        .HasDisplayUnitLabel = False
    End With
    ActiveChart.Axes(xlValue).AxisTitle.Select
    Selection.Characters.Text = "Punch Load (kN)"
    Selection.AutoScaleFont = False
    With Selection.Characters(Start:=1, Length:=15).Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 10.25
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
    End With
    ActiveChart.Axes(xlValue).Select
    ActiveChart.Axes(xlCategory).Select
    With ActiveChart.Axes(xlCategory)
        .MinimumScaleIsAuto = True
        .MaximumScale = 0.052
        .MinorUnitIsAuto = True
        .MajorUnitIsAuto = True
        .Crosses = xlAutomatic
        .ReversePlotOrder = False
        .ScaleType = xlLinear
        .DisplayUnit = xlNone
    End With
    With ActiveChart.Axes(xlCategory)
        .MinimumScale = 0
        .MaximumScale = 0.055
        .MinorUnitIsAuto = True
        .MajorUnitIsAuto = True
        .Crosses = xlAutomatic
        .ReversePlotOrder = False
        .ScaleType = xlLinear
        .DisplayUnit = xlNone
    End With
' move chart down a bit
    ActiveSheet.ChartObjects("Chart 1").Activate
    ActiveChart.ChartArea.Select
    ActiveSheet.Shapes("Chart 1").IncrementTop 13.5
    ActiveWindow.Visible = False
    Windows(file$).Activate
    Range("A1").Select

' ----------------------------------------------------------------------------
'
' Punch scaling numbers
' ----------------------------------------------------------------------------
'
    Height$ = Trim(Str(endpt - startpt))
    Sheets("Load Left").Select
    Range("C2").Select
    tmp$ = "=MAX(RC[-2]:R[" & Height$ & "]C[-2])"
    ActiveCell.FormulaR1C1 = tmp$
    Range("B2").Select
    ActiveCell.FormulaR1C1 = "max"
    Range("B2:C2").Select
    Selection.Cut
    Range("C2").Select
    ActiveSheet.Paste
    ActiveCell.FormulaR1C1 = "max left"
    Range("C3").Select
    Sheets("Load Right").Select
    Range("C2").Select
    ActiveCell.FormulaR1C1 = "max right"
    Range("D2").Select
    tmp$ = "=MAX(RC[-3]:R[" & Height$ & "]C[-3])"
```

```
    ActiveCell.FormulaR1C1 = tmp$
    Range("C2:D2").Select
    Selection.Copy
    Sheets("Load Left").Select
    ActiveSheet.Paste
    Range("C3").Select
'    ActiveWindow.ActivateNext
    Sheets("Load Left").Select
    Range("D3").Select
    Sheets("Load Right").Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Load Left").Select
    Range("C3:D3").Select
    ActiveSheet.Paste Link:=True
    Range("C5").Select
    Application.CutCopyMode = False
    ActiveCell.FormulaR1C1 = "real max left"
    Range("C6").Select
    ActiveCell.FormulaR1C1 = "real max right"
    Range("C7").Select
    Columns("C:C").ColumnWidth = 11.86
    Range("D5").Select
    ActiveCell.FormulaR1C1 = "3.6"
    Range("D6").Select
    ActiveCell.FormulaR1C1 = "3.6"
    Range("C8").Select
    ActiveCell.FormulaR1C1 = "scale left"
    Range("C9").Select
    ActiveCell.FormulaR1C1 = "scale right"
    Range("D8").Select
    ActiveCell.FormulaR1C1 = "=R[-3]C/R[-6]C"
    Range("D9").Select
    ActiveCell.FormulaR1C1 = "=R[-3]C/R[-6]C"
    Range("C11").Select
    ActiveCell.FormulaR1C1 = "average"
    Range("D11").Select
    ActiveCell.FormulaR1C1 = "=(R[-3]C+R[-2]C)/2"
    Range("D11").Select
    Selection.Copy
    Sheets("Punch Force").Select
    Range("F1").Select
    ActiveSheet.Paste Link:=True
    Range("D15").Select
    ActiveSheet.ChartObjects("Chart 1").Activate
    ActiveChart.Axes(xlValue).Select
    ActiveWindow.Visible = False
    Windows(file$).Activate
    Sheets("Load Left").Select
    Range("D5").Select
    Application.CutCopyMode = False

' ------------------------------------------------------------------------------
'
' Load left graph
' ------------------------------------------------------------------------------
'
    Range("A1").Select
    Range(Selection, Selection.End(xlDown)).Select
    ActiveWindow.LargeScroll Down:=-2
    Charts.Add
    ActiveChart.ChartType = xlXYScatter
    tmp$ = "A1:A" & Trim(Str(endpt - startpt))
    ActiveChart.SetSourceData Source:=Sheets("Load Left").Range(tmp$), _
        PlotBy:=xlColumns
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Load Left"
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = "Load/Left"
        .Axes(xlCategory, xlPrimary).HasTitle = False
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Load in Volts"
    End With
    ActiveChart.PlotArea.Select
    With Selection.Border
        .ColorIndex = 16
        .Weight = xlThin
```

```
        .LineStyle = xlContinuous
    End With
    Selection.Interior.ColorIndex = xlNone
    ActiveChart.ChartArea.Select
    ActiveSheet.Shapes("Chart 1").ScaleHeight 1.48, msoFalse, _
        msoScaleFromBottomRight
    ActiveWindow.Visible = False
    Windows(file$).Activate
    Range("C1").Select
' ------------------------------------------------------------------------------
-
' Load right graph
' ------------------------------------------------------------------------------
-
    Sheets("Load Right").Select
    Range("A1").Select
    Range(Selection, Selection.End(xlDown)).Select
    ActiveWindow.LargeScroll Down:=-2
    Charts.Add
    ActiveChart.ChartType = xlXYScatter
    ActiveChart.SetSourceData Source:=Sheets("Load Right").Range(tmp$), _
        PlotBy:=xlColumns
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Load Right"
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = "Load/Right"
        .Axes(xlCategory, xlPrimary).HasTitle = False
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Load in Volts"
    End With
    ActiveSheet.Shapes("Chart 1").ScaleHeight 1.48, msoFalse, _
        msoScaleFromBottomRight
    ActiveChart.PlotArea.Select
    With Selection.Border
        .ColorIndex = 16
        .Weight = xlThin
        .LineStyle = xlContinuous
    End With
    Selection.Interior.ColorIndex = xlNone
    ActiveWindow.Visible = False
    Windows(file$).Activate
    Range("B1").Select
' ------------------------------------------------------------------------------
-
' Binder Pressure values
' ------------------------------------------------------------------------------
-
    Sheets("Binder Pressure").Select
    Range("E2").Select
    ActiveCell.FormulaR1C1 = "Intercept"
    Range(Selection, Cells(ActiveCell.Row, 1)).Select
    Range("E2").Select
    ActiveCell.FormulaR1C1 = "Slope"
    Range("E3").Select
    ActiveCell.FormulaR1C1 = "Intercept"
    Range("F2").Select
    ActiveCell.FormulaR1C1 = "1271.750527"
    Range("F3").Select
    ActiveCell.FormulaR1C1 = "-621.7885791"
    Range("E5").Select
    ActiveCell.FormulaR1C1 = "Average"
    Range("F5").Select
    tmp$ = "B2:B" & Trim(Str(endpt - startpt))
    ActiveCell.Value = "=AVERAGE(" & tmp$ & ")"

    Range("B2").Select
    ActiveCell.FormulaR1C1 = "=RC[-1]*R2C6+R3C6"
    Range("B2").Select
    Selection.Copy

    tmp$ = "B2:B" & Trim(Str(endpt - startpt))
    Range(tmp$).Select
    ActiveSheet.Paste

    Range("A1").Select
    Application.CutCopyMode = False
    Selection.Copy
```

```
    Range("B1").Select
    ActiveSheet.Paste
    Range("B1").Select
    Application.CutCopyMode = False
    Range("B1:B2").Select
    Range(Selection, Selection.End(xlDown)).Select
' ------------------------------------------------------------------------------

' Binder Pressure graph
' ------------------------------------------------------------------------------

    ActiveWindow.LargeScroll Down:=-2
    Charts.Add
    ActiveChart.ChartType = xlXYScatter
    ActiveChart.SetSourceData Source:=Sheets("Binder Pressure").Range(tmp$) _
        , PlotBy:=xlColumns
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Binder Pressure"
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = "Cylinder Pressure"
        .Axes(xlCategory, xlPrimary).HasTitle = False
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Pressure (PSI)"
    End With
    ActiveChart.PlotArea.Select
    With Selection.Border
        .ColorIndex = 16
        .Weight = xlThin
        .LineStyle = xlContinuous
    End With
    Selection.Interior.ColorIndex = xlNone
    ActiveChart.Axes(xlValue).Select
    With ActiveChart.Axes(xlValue)
        .MinimumScale = 0
        .MaximumScale = 3500
        .MinorUnitIsAuto = True
        .MajorUnitIsAuto = True
        .Crosses = xlAutomatic
        .ReversePlotOrder = False
        .ScaleType = xlLinear
        .DisplayUnit = xlNone
    End With
    ActiveWindow.Visible = False
    Windows(file$).Activate
    Sheets("Binder Pressure").Select
    ActiveSheet.ChartObjects("Chart 1").Activate
    ActiveChart.ChartArea.Select
    ActiveSheet.Shapes("Chart 1").IncrementLeft 74.25
    ActiveSheet.Shapes("Chart 1").IncrementTop -139.5
    ActiveSheet.Shapes("Chart 1").ScaleHeight 1.74, msoFalse, msoScaleFromTopLeft
' ------------------------------------------------------------------------------

' Binder Pressure
' ------------------------------------------------------------------------------

    Sheets("Raw Data").Select
    Range("F1:M1").Select
    Range(Selection, Selection.End(xlDown)).Select
    Selection.Copy
    Sheets("Binder Load Cells").Select
    ActiveSheet.Paste
    Range("L2").Select
    Application.CutCopyMode = False
    ActiveCell.FormulaR1C1 = "Slope"
    Range("L3").Select
    ActiveCell.FormulaR1C1 = "Intercept"
    Range("M3").Select
    ActiveCell.FormulaR1C1 = "-0.034913793"
    Range("M2").Select
    ActiveCell.FormulaR1C1 = "1.318965517"
    Range("L5").Select
    ActiveCell.FormulaR1C1 = "Averages"
    Range("L6").Select
    ActiveCell.FormulaR1C1 = "=AVERAGE(R[-1]C[-11]:R[108]C[-11])"
    Range("L6").Select
    Selection.Copy
    Range("M6:S6").Select
```

```
    ActiveSheet.Paste
    Range("L2:S6").Select
    Application.CutCopyMode = False
    Selection.Cut
    Range("J2").Select
    ActiveSheet.Paste
    Range("J8").Select
    ActiveCell.FormulaR1C1 = "Sum"
    Range("J9").Select
    ActiveCell.FormulaR1C1 = "=SUM(R[-3]C:R[-3]C[7])"
    Range("J11").Select
    ActiveCell.FormulaR1C1 = "Convert to tons"
    Range("J12").Select
    ActiveCell.FormulaR1C1 = "=R[-3]C*R[-10]C[1]+R[-9]C[1]"
    Range("J14").Select
    ActiveCell.FormulaR1C1 = "Convert to Newtons"
    Range("J15").Select
    ActiveCell.FormulaR1C1 = "=R[-3]C*9810"

' ------------------------------------------------------------------------------

' Binder Pressure Cylinders Graph
' ------------------------------------------------------------------------------

    Charts.Add
    ActiveChart.ChartType = xlXYScatterSmooth
    ActiveChart.SetSourceData Source:=Sheets("Binder Load Cells").Range("J14:J15"

        ), PlotBy:=xlColumns
    ActiveChart.SeriesCollection(1).Delete
    ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection.NewSeries
' Cylinder 1
    tmp$ = "='Binder Load Cells'!R2C1:R" & Trim(Str(endpt - startpt)) & "C1"
    ActiveChart.SeriesCollection(1).Values = tmp$
    ActiveChart.SeriesCollection(1).Name = "Cylinder 1"
' Cylinder 2
    tmp$ = "='Binder Load Cells'!R2C2:R" & Trim(Str(endpt - startpt)) & "C2"
    ActiveChart.SeriesCollection(2).Values = tmp$
    ActiveChart.SeriesCollection(2).Name = "Cylinder 2"
' Cylinder 3
    tmp$ = "='Binder Load Cells'!R2C3:R" & Trim(Str(endpt - startpt)) & "C3"
    ActiveChart.SeriesCollection(3).Values = tmp$
    ActiveChart.SeriesCollection(3).Name = "Cylinder 3"
' Cylinder 4
    tmp$ = "='Binder Load Cells'!R2C4:R" & Trim(Str(endpt - startpt)) & "C4"
    ActiveChart.SeriesCollection(4).Values = tmp$
    ActiveChart.SeriesCollection(4).Name = "Cylinder 4"
' Cylinder 5
    tmp$ = "='Binder Load Cells'!R2C5:R" & Trim(Str(endpt - startpt)) & "C5"
    ActiveChart.SeriesCollection(5).Values = tmp$
    ActiveChart.SeriesCollection(5).Name = "Cylinder 5"
' Cylinder 6
    tmp$ = "='Binder Load Cells'!R2C6:R" & Trim(Str(endpt - startpt)) & "C6"
    ActiveChart.SeriesCollection(6).Values = tmp$
    ActiveChart.SeriesCollection(6).Name = "Cylinder 6"
' Cylinder 7
    tmp$ = "='Binder Load Cells'!R2C7:R" & Trim(Str(endpt - startpt)) & "C7"
    ActiveChart.SeriesCollection(7).Values = tmp$
    ActiveChart.SeriesCollection(7).Name = "Cylinder 7"
' Cylinder 8
    tmp$ = "='Binder Load Cells'!R2C8:R" & Trim(Str(endpt - startpt)) & "C8"
    ActiveChart.SeriesCollection(8).Values = tmp$
    ActiveChart.SeriesCollection(8).Name = "Cylinder 8"

    ActiveChart.Location Where:=xlLocationAsObject, Name:="Binder Load Cells"
    With ActiveChart
        .HasTitle = False
        .Axes(xlCategory, xlPrimary).HasTitle = False
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Presure (Volts)"
```

```
    End With
    ActiveSheet.Shapes("Chart 1").IncrementLeft 173.25
    ActiveSheet.Shapes("Chart 1").IncrementTop 92.25
    ActiveSheet.Shapes("Chart 1").IncrementLeft 38.25
    ActiveSheet.Shapes("Chart 1").IncrementTop -47.25
    ActiveSheet.Shapes("Chart 1").ScaleHeight 1.29, msoFalse, msoScaleFromTopLeft
    ActiveChart.PlotArea.Select
    With Selection.Border
        .ColorIndex = 16
        .Weight = xlThin
        .LineStyle = xlContinuous
    End With
    Selection.Interior.ColorIndex = xlNone
    ActiveWindow.Visible = False
    Windows(file$).Activate
    Range("K15").Select
' ------------------------------------------------------------------------------
-
' Cleanup
' ------------------------------------------------------------------------------
-
    Sheets("Raw Data").Select
    Sheets("Raw Data").Move after:=Sheets(7)
    Range("A1").Select
    Sheets("Binder Pressure").Select
    ActiveSheet.ChartObjects("Chart 1").Activate
    ActiveWindow.Visible = False
    Windows(file$).Activate
    Range("A1").Select
    Sheets("Load Right").Select
    Range("A1").Select
    Sheets("Load Left").Select
    Range("D5").Select
    Sheets("Punch Force").Select
    Range("A1").Select
    Sheets.Add
    Sheets("Sheet7").Select
    Sheets("Sheet7").Name = "Abaqus"
' ------------------------------------------------------------------------------
-
' Save File
' ------------------------------------------------------------------------------
-
    Path$ = "D:\umich\data\processed"
    ChDir Path$
    tmp$ = Path$ & "\" & outputfile$
    ActiveWorkbook.SaveAs Filename:=tmp$, _
        FileFormat:=xlNormal, Password:="", WriteResPassword:="", _
        ReadOnlyRecommended:=False, CreateBackup:=False

End Sub
' ------------------------------------------------------------------------------
-
' Subroutine for Speed selection to create relative displacement data
' ------------------------------------------------------------------------------
-
Sub writedisp(speed$)
    Range("A1").Select
    ActiveCell.Value = "Time"
    ActiveCell.Offset(0, 1).Value = "Displacement"
    ActiveCell.Offset(1, 0).Activate
' if speed$ = 12 spm, then process occurs in 1.06 seconds
' if speed$ = 18 spm, then process occurs in 0.63 seconds
    If (speed$ = 12) Then
        t = 1.06
    Else
        t = 0.63
    End If
    s = 2
    s = s * 25.4 / 1000
    u = 2 * s / t           ' initial velocity
    a = -u * u / (2 * s)
    For tt = 1 To t * 100
        ActiveCell.Value = tt / 100
        ActiveCell.Offset(0, 1).Value = u * tt / 100 + 0.5 * a * (tt / 100) ^ 2
        ActiveCell.Offset(1, 0).Activate
    Next
```

```
    Range("A1").Select
End Sub
```

Appendix D – Python Macro's to Extract ODB Data

Recovery of Reaction force: rf.py

```
#!/usr/local/bin/python
# Run using:
#       abaqus python rf.py filename.odb
#
# Outputs the Punch reaction force to filename.csv for Step 3
#
from odbAccess import *
import sys, getopt

arg = sys.argv[1:]
odbPath = arg[0]
if not odbPath:
      print "Please specify a ODB file"
rf_filename = odbPath[:-4] + ".csv"

rf_file = open(rf_filename, 'w')

odb = openOdb(path=odbPath)
assembly =odb.rootAssembly
ref_node = assembly.instance['PUNCH-1'].nodeSet['"PUNCH REFPT"']

step3 = odb.step['"Form Blank"']
for frame in step3.frame:
      rf = frame.fieldOutput['RF'].getSubset(region=ref_node)
      u = frame.fieldOutput['U'].getSubset(region=ref_node)
      val2 = `-1 * rf.value[0].data[1]` + "\n"
      val = `-1 * u.value[0].data[1]` + ", " + val2
      rf_file.write(val)
rf_file.close()
```

Recovery of Displacement vs Time: dt.py

```
#!/usr/local/bin/python
# Run using:
#       abaqus python dt.py filename.odb
#
# Outputs the displacement vs time to filename.csv for Step 3
#
from odbAccess import *
import sys, getopt

arg = sys.argv[1:]
odbPath = arg[0]
if not odbPath:
      print "Please specify a ODB file"
dt_filename = odbPath[:-4] + ".csv"

dt_file = open(dt_filename, 'w')

odb = openOdb(path=odbPath)
assembly =odb.rootAssembly
ref_node = assembly.instance['PUNCH-1'].nodeSet['"PUNCH REFPT"']

step3 = odb.step['"Form Blank"']


for frame in step3.frame:
      u = frame.fieldOutput['U'].getSubset(region=ref_node)
      t = frame.frameValue
      val = `t` + ", " + `-1 * u.value[0].data[1]` + "\n"
      dt_file.write(val)
dt_file.close()
```

Recovery of Velocity vs. time: vt.py

```
#!/usr/local/bin/python
#
# Run using:
#       abaqus python vt.py filename.odb
#
# Outputs the displacement vs time to filename.csv
#  for Step 3
#
from odbAccess import *
import sys, getopt

arg = sys.argv[1:]
odbPath = arg[0]
if not odbPath:
      print "Please specify a ODB file"
vt_filename = odbPath[:-4] + ".csv"

vt_file = open(vt_filename, 'w')

odb = openOdb(path=odbPath)
assembly =odb.rootAssembly
ref_node = assembly.instance['PUNCH-1'].nodeSet['"PUNCH REFPT"']

step3 = odb.step['"Form Blank"']


for frame in step3.frame:
      v = frame.fieldOutput['V'].getSubset(region=ref_node)
      t = frame.frameValue
      val = `t` + ", " + `-1 * v.value[0].data[1]` + "\n"
      vt_file.write(val)
vt_file.close()
```

Recovery of Internal and Kinetic Energy vs. displacement: ieke.py

```
#!/usr/local/bin/python
#
# Run using:
#       abaqus python energy.py filename.odb
#
# Outputs the time, KE, IE to filename.ieke.csv
#  for Step 3
#
from odbAccess import *
import sys, getopt

arg = sys.argv[1:]
odbPath = arg[0]
if not odbPath:
      print "Please specify a ODB file"
rf_filename = odbPath[:-4] + ".ieke.csv"

odb = openOdb(path=odbPath)
step3 = odb.step['"Form Blank"']
assembly =odb.rootAssembly
ref_node = assembly.instance['PUNCH-1'].nodeSet['"PUNCH REFPT"']

# Displacement values
u = []
for frame in step3.frame:
      u_raw  = frame.fieldOutput['U'].getSubset(region=ref_node)
      u_filt = -1 * u_raw.value[0].data[1]
      u.append(u_filt)

rf_file = open(rf_filename, 'w')
ke = step3.historyRegion['Assembly
ASSEMBLY'].historyOutput['ALLKE']
ie = step3.historyRegion['Assembly
ASSEMBLY'].historyOutput['ALLIE']

size = len(ke.data)/2
for x in range(size):
      val = `u[x]`+", "+`ke.data[x*2][1]`+",
"+`ie.data[x*2][1]`+"\n"
      rf_file.write(val)
rf_file.close()
```

Appendix E – Sample ABAQUS Input Deck

```
*Heading
 Michigan Press Metal Forming
** Job name: 70kN_06mm thick Model name: Model-1
**
** PARTS
**
*Part, name=Blank
*End Part
*Part, name=Die
*End Part
*Part, name=Holder
*End Part
*Part, name=Punch
*End Part
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=Blank-1, part=Blank
0., 0., 0.
0., 0., 0., 1., 0., 0., 180.
*Include blank.inp
**
** Region: (Steel:Picked)
*Elset, elset=_I1, internal, generate
    1,  1800,     1
** Section: Steel
*Shell Section, elset=_I1, material=Steel
0.0008, 11
*End Instance
**
*Instance, name=Die-1, part=Die
*Include die.inp
**
*Node
   3536,           0.,       -0.005,           0.
*Nset, nset=Die-1-RefPt_, internal
3536,
*element, type=mass, elset=tools
3536,3536
*mass, elset=tools
.1
*Elset, elset=Die-1, generate
    1,  3384,     1
*End Instance
**
*Instance, name=Holder-1, part=Holder
*Node
*Include holder.inp
**
*Node
   2543,           0.,        0.005,           0.
*Nset, nset=Holder-1-RefPt_, internal
2543,
*element, type=mass, elset=tools
2543, 2543
*mass, elset=tools
```

```
.1
*Elset, elset=Holder-1, generate
    1,  2403,     1
*End Instance
**
*Instance, name=Punch-1, part=Punch
*Include punch.inp
**
*Node
   1552,           0.,        0.005,           0.
*Nset, nset=Punch-1-RefPt_, internal
1552,
*element, type=mass, elset=tools
1552, 1552
*mass, elset=tools
.1
*Nset, nset="Punch Refpt"
 1552,
*Elset, elset=Punch-1, generate
    1,  1508,     1
*End Instance
*Nset, nset=_G16, internal, instance=Die-1
 3536,
*Nset, nset=_G17, internal, instance=Holder-1
 2543,
*Nset, nset=_G18, internal, instance=Punch-1
 1552,
*Nset, nset=_G22, internal, instance=Die-1
 3536,
*Nset, nset=_G23, internal, instance=Holder-1
 2543,
*Nset, nset=_G24, internal, instance=Punch-1
 1552,
*Nset, nset=_G25, internal, instance=Punch-1
 1552,
*Nset, nset=_G26, internal, instance=Holder-1
 2543,
*Nset, nset=_G82, internal, instance=Die-1
 3536,
*Elset, elset=__G11_SNEG, internal, instance=Die-1, generate
    1,  3384,     1
*Surface, type=ELEMENT, name=_G11, internal
__G11_SNEG, SNEG
*Elset, elset=__G13_SNEG, internal, instance=Holder-1, generate
    1,  2403,     1
*Surface, type=ELEMENT, name=_G13, internal
__G13_SNEG, SNEG
*Elset, elset=__G15_SPOS, internal, instance=Punch-1, generate
    1,  1508,     1
*Surface, type=ELEMENT, name=_G15, internal
__G15_SPOS, SPOS
*Elset, elset=__G83_SNEG, internal, instance=Blank-1, generate
    1,  1800,     1
*Surface, type=ELEMENT, name=_G83, internal
__G83_SNEG, SNEG
*Elset, elset=__G84_SNEG, internal, instance=Blank-1, generate
    1,  1800,     1
*Surface, type=ELEMENT, name=_G84, internal
__G84_SNEG, SNEG
*Elset, elset=__G85_SNEG, internal, instance=Blank-1, generate
    1,  1800,     1
```

```
*Surface, type=ELEMENT, name=_G85, internal
__G85_SNEG, SNEG
*Rigid Body, ref node=Die-1.Die-1-RefPt_, elset=Die-1.Die-1
*Rigid Body, ref node=Holder-1.Holder-1-RefPt_, elset=Holder-
1.Holder-1
*Rigid Body, ref node=Punch-1.Punch-1-RefPt_, elset=Punch-1.Punch-
1
*End Assembly
*Amplitude, name=Forming, time=TOTAL TIME, smooth=0.25,
value=absolute
0.002, -4.24, 0.027, 0.
*Amplitude, name=Smooth, definition=SMOOTH STEP
0., 0., 1., 1.
**
** MATERIALS
**
*Material, name=Steel
*Density
7800.,
*Elastic
 2.1e+11, 0.3
*Plastic
1.72E+08, 0.00000
1.90E+08, 0.00164
2.06E+08, 0.00907
2.20E+08, 0.01646
2.33E+08, 0.02376
2.45E+08, 0.03103
2.57E+08, 0.03829
2.65E+08, 0.04556
2.76E+08, 0.05281
2.82E+08, 0.06006
2.91E+08, 0.06712
2.96E+08, 0.07423
3.04E+08, 0.08142
3.08E+08, 0.08832
3.13E+08, 0.09550
3.16E+08, 0.09836
3.23E+08, 0.10940
3.30E+08, 0.12040
3.36E+08, 0.13150
3.42E+08, 0.14240
3.47E+08, 0.15336
3.53E+08, 0.16438
3.57E+08, 0.17545
3.63E+08, 0.18645
3.66E+08, 0.19743
3.72E+08, 0.20842
3.76E+08, 0.21931
3.84E+08, 0.24050
3.92E+08, 0.26174
3.98E+08, 0.28110
4.01E+08, 0.29815
4.06E+08, 0.31641
4.12E+08, 0.33484
4.12E+08, 0.35369
**
** BOUNDARY CONDITIONS
**
** Name: BC Die Type: Displacement/Rotation
*Boundary
```

```
_G16, 1, 1
_G16, 3, 3
_G16, 4, 4
_G16, 5, 5
_G16, 6, 6
** Name: BC Holder Type: Displacement/Rotation
*Boundary
_G17, 1, 1
_G17, 3, 3
_G17, 4, 4
_G17, 5, 5
_G17, 6, 6
** Name: BC Punch Type: Displacement/Rotation
*Boundary
_G18, 1, 1
_G18, 3, 3
_G18, 4, 4
_G18, 5, 5
_G18, 6, 6
** ----------------------------------------------------------------
**
** STEP: Clamp Blank
**
*Step, name="Clamp Blank"
Close blank holder and Die
*Dynamic, Explicit
, 0.001
*Bulk Viscosity
0.06, 1.2
**
** BOUNDARY CONDITIONS
**
** Name: Die Closeup Type: Displacement/Rotation
*Boundary, amplitude=Smooth
_G22, 2, 2, 4.6
** Name: Holder Closeup Type: Displacement/Rotation
*Boundary, amplitude=Smooth
_G23, 2, 2, -4.6
**
** INTERACTION PROPERTIES
**
*Surface Interaction, name="Coulomb Friction"
*Friction, taumax=8e+07
 0.06,
**
** INTERACTIONS
**
** Interaction: Blank to die
*Contact Pair, interaction="Coulomb Friction", mechanical
constraint=KINEMATIC
_G11, _G83
** Interaction: Blank to holder
*Contact Pair, interaction="Coulomb Friction", mechanical
constraint=KINEMATIC
_G13, _G84
**
** OUTPUT REQUESTS
**
*Restart, write, number interval=1, time marks=NO
**
** FIELD OUTPUT: F-Output-1
```

```
**
*Output, field, variable=PRESELECT
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT, time interval=5e-06
*End Step
** ----------------------------------------------------------------
**
** STEP: Blank Holder Force
**
*Step, name="Blank Holder Force"
*Dynamic, Explicit
, 0.001
*Bulk Viscosity
0.06, 1.2
**
** BOUNDARY CONDITIONS
**
** Name: BC Die Type: Displacement/Rotation
*Boundary, op=NEW
_G16, 1, 1
_G16, 3, 3
_G16, 4, 4
_G16, 5, 5
_G16, 6, 6
** Name: BC Holder Type: Displacement/Rotation
*Boundary, op=NEW
_G17, 1, 1
_G17, 3, 3
_G17, 4, 4
_G17, 5, 5
_G17, 6, 6
** Name: BC Punch Type: Displacement/Rotation
*Boundary, op=NEW
_G18, 1, 1
_G18, 3, 3
_G18, 4, 4
_G18, 5, 5
_G18, 6, 6
** Name: Die Closeup Type: Displacement/Rotation
*Boundary, op=NEW
** Name: Die Fix Type: Displacement/Rotation
*Boundary, op=NEW
_G82, 1, 1
_G82, 2, 2
_G82, 3, 3
_G82, 4, 4
_G82, 5, 5
_G82, 6, 6
** Name: Holder Closeup Type: Displacement/Rotation
*Boundary, op=NEW
** Name: Punch Closeup Type: Velocity/Angular velocity
*Boundary, op=NEW, amplitude=Smooth, type=VELOCITY
_G24, 2, 2, -4.24
**
** LOADS
**
** Name: Blank Holder Force   Type: Concentrated force
*Cload
_G26, 2, -70000.
```

```
**
** OUTPUT REQUESTS
**
*Restart, write, number interval=1, time marks=NO
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT, time interval=5e-06
*End Step
** ----------------------------------------------------------------
**
** STEP: Form Blank
**
*Step, name="Form Blank"
*Dynamic, Explicit
, 0.025
*Bulk Viscosity
0.06, 1.2
**
** BOUNDARY CONDITIONS
**
** Name: BC Die Type: Displacement/Rotation
*Boundary, op=NEW
_G16, 1, 1
_G16, 3, 3
_G16, 4, 4
_G16, 5, 5
_G16, 6, 6
** Name: BC Holder Type: Displacement/Rotation
*Boundary, op=NEW
_G17, 1, 1
_G17, 3, 3
_G17, 4, 4
_G17, 5, 5
_G17, 6, 6
** Name: BC Punch Type: Displacement/Rotation
*Boundary, op=NEW
_G18, 1, 1
_G18, 3, 3
_G18, 4, 4
_G18, 5, 5
_G18, 6, 6
** Name: Die Fix Type: Displacement/Rotation
*Boundary, op=NEW
_G82, 1, 1
_G82, 2, 2
_G82, 3, 3
_G82, 4, 4
_G82, 5, 5
_G82, 6, 6
** Name: Punch Closeup Type: Velocity/Angular velocity
*Boundary, op=NEW
** Name: Punch Forming Type: Velocity/Angular velocity
*Boundary, op=NEW, amplitude=Forming, type=VELOCITY
_G25, 2, 2, -4.24
**
** INTERACTIONS
```

```
**
** Interaction: Blank to punch
*Contact Pair, interaction="Coulomb Friction", mechanical
constraint=KINEMATIC
_G15, _G85
**
** OUTPUT REQUESTS
**
*Restart, write, number interval=1, time marks=NO
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT
**
** FIELD OUTPUT: Punch Feedback force
**
*Output, field, number intervals=100, time marks=YES
*Node Output
U, V, RF
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT, time interval=0.000125
*End Step
```