TOOLING DESIGN FOR THREE-DIMENSIONAL SHEET METAL FORMING USING FINITE ELEMENT ANALYSIS

by

APOSTOLOS P. KARAFILLIS

SMME, Mechanical Engineering, Massachusetts Institute of Technology
(1992)

Diploma, Mechanical Engineering, National Technical University of Athens
(1989)

Submitted to the Department of
Mechanical Engineering
In Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August, 1994

Signature of Author_____
Department of Mechanical Engineering
August 11, 1994

Certified by_____
Mary C. Boyce
Associate Professor of Mechanical Engineering
Thesis Supervisor

Accepted by_____
Ain A. Sonin, Chairman
Departmental Committee on Graduate Studies

Tooling Design for Three-Dimensional Sheet Metal Forming Using Finite Element Analysis

by

Apostolos P. Karafillis

Submitted to the Department of Mechanical Engineering on August 24, 1994,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Mechanical Engineering

Abstract

Sheet metal forming operations are complicated manufacturing processes which subject the workpiece to various modes of plastic deformation (stretching, bending, in-plane compression) in combination with continuous or intermittent frictional interaction of the workpiece with the forming tools. Mathematical modeling can be of great importance in the design of sheet metal forming processes in order to design defect-free parts within the desired dimensional tolerances. In the present work finite element analysis is combined with a new anisotropic material model, developed as a part of this thesis work, to provide a computational tool capable of numerical simulation of forming of anisotropic thin rolled metal blanks. The developed computational tool was then used to design sheet metal forming processes in which springback occurs in the produced part upon removal of the forming loads. The manufacturing problem of part shape error due to springback was solved by using appropriate designs of the tooling and the binder shape together with a binder constraint. A methodology for tool and binder design based on inverse springback calculations was developed, where the desired part shape is sprung forward to the required die shape. The design procedure was demonstrated for different two-dimensional and three-dimensional geometries with an aluminum alloy used as the workpiece material. The tooling needed to form the desired parts was numerically designed using the proposed algorithm then machined in a CNC machine. The designed tooling was found to produce the desired part shape demonstrating both the accuracy of the numerical simulations and the success of the proposed tooling design algorithm.

Thesis committee: Assoc. Prof. Mary C. Boyce (Chairman)
Prof. David E. Hardt
Prof. David M. Parks

To Mata

Acknowledgments

As I recall my four years at MIT as a graduate student, I feel overwhelmed by a sense of gratitude to more people than I can possibly thank here.

I am particularly grateful to my advisor Professor Mary C. Boyce for her encouragement, precious guidance and continuous support. Professor Boyce's enthusiasm and kind patience made my journey at MIT an invaluable experience.

I would also like to thank Professor David E. Hardt and Professor David M. Parks for the helpful discussions and the considerable time and effort they spent following my research.

I owe gratitude to the Mechanics and Materials community of MIT for contributing to the creation of a pleasant working environment with their humour, comradery and technical assistance. It has been great working with Jian Cao, Clarence Chui, Deborah Demania, Omar Hasan, Simona Socrate, Mario Sunseri, Lisa Tegeler, John Zaroulis, Andrew Zhuk, and all the other members of this group. Also, my collaboration with Dan Walczyk from the Laboratory for Manufacturing and Productivity was positive and fruitful.

Financial support for this research was provided by the National Science Foundation, Division of Manufacturing and Design, under grant NSF-DDM-9202362.

I feel greatful to my parents who instilled on me the drive for higher education.

This work would not have been possible without my wife Mata, for whom I keep the best of my thoughts and feelings.

List of Figures

Fig. 1.1: Process design procedures
Fig. 2.1: The upper bound, the lower bound and the von Mises yield surface in the π-plane.
Fig. 2.2: Different isotropic yield surfaces between the von Mises yield surface and the lower bound (Tresca).
Fig. 2.3: Different isotropic yield surfaces between the von Mises yield surface and the upper bound.
Fig. 2.4: Different isotropic yield surfaces between the lower bound (Tresca) and the upper bound.
Fig. 2.5: The anisotropic material, the Isotropic Plasticity Equivalent (IPE) material and the IPE stress transformation.
Fig. 2.6:(a) Isotropic yield surfaces for FCC (restricted glide) and BCC (pencil glide) materials obtained by popLA.
(b) Isotropic yield surfaces as obtained by using Method I and Method II.
Fig. 2.7: Different isotropic yield surfaces in the $\alpha - \nu$ domain.
Fig. 2.8: Distribution of the uniaxial yield stress in different angles about the 3-axis for a material with cubic symmetry. The parameter β indicates degree of anisotropy.
Fig. 2.9: Distribution of the uniaxial yield stress in different angles about the **p**-axis for a material with cubic symmetry. The parameter β indicates degree of anisotropy.
Fig. 2.10: Distribution of the uniaxial yield stress in different angles about the 3-axis for a material with trigonal symmetry. The parameter ζ indicates degree of anisotropy
Fig. 2.11:(a) The distribution of the R-ratio with respect to the angle with the rolling direction for the H-19 can stock alloy. Experimental results were obtained by MacEwen et. al. [1992]
(b) The distribution of the yield stress in uniaxial tension with respect to the angle with the rolling direction for the H-19 alloy. Experimental results were obtained by MacEwen et. al. [1992]
Fig. 2.12: The yield surface of the H-19 can stock aluminum alloy in the $\sigma_x - \sigma_y$ domain. Experimental results were obtained by MacEwen et al. [1992]
Fig. 2.13:(a) The distribution of the R-ratio with respect to the angle with the rolling direction for the 2008-T4 aluminum alloy. Experimental results by Lege et. al. [1989]. The Bishop-Hill distribution was obtained by Lege et al. [1989]
(b) The distribution of the yield stress in uniaxial tension with respect to the angle with the rolling direction for the 2008-T4 aluminum alloy. The experimental results and the Bishop-Hill distribution were obtained by Lege et. al. [1989]
Fig. 2.14:(a) The distribution of the tensile and compressive yield stress with respect to the angle with the rolling direction for the 2008-T4 aluminum alloy. Experimental

results were obtained by Lege et. al. [1989].
(b) Theoretical and experimental distribution of the R-ratio for the 2008-T4 alloy. The theoretical distribution was obtained with $\mathbf{B} \neq \mathbf{0}$. Experimental results by Lege et al. [1989]
Fig. 2.15: The projection of the yield surface in the $\sigma_x - \sigma_y$ domain for the 2008-T4 material. Experimental results were obtained by Lege et. al. [1989].
Fig. 3.1: Evolution of the material state.
Fig. 4.1: Tooling and undeformed blank for the modeled channel geometry
Fig. 4.2: The yield locus of Karafillis and Boyce [1993] and Hill [1950] for the 2008-T4 aluminum.
Fig. 4.3: The equivalent stress - equivalent strain curve of the examined 2008-T4 alloy.
Fig. 4.4: Side view of successive steps of the channel forming process obtained by simulation (left) and by experiment (right); (a)-(c) loading; (d) unloading.
Fig. 4.5: Stress distribution at the top and the bottom of the blank in the fully loaded stage.
Fig. 4.6: Profile shapes of the produced part upon unloading obtained by theory and experiment.
Fig. 4.7: Displacement history used during the explicit loading simulations.
Fig. 4.8: The ratio of (kinetic energy)/(elastic strain energy) of the blank as a function of the duration of the simulated phenomenon.
Fig 4.9: Stress distribution on the top layer of the blank obtained by implicit static and explicit dynamic finite element analysis.
Fig. 4.10: Stress distribution on the top layer of the blank obtained by explicit dynamic analysis with a large value of r.
Fig. 4.11: Part shape upon springback calculated with the combination of dynamic explicit analysis for loading and static implicit analysis for unloading.
Fig. 4.12: Initial pan design.
Fig. 4.13: Tooling set-up in finite element model.
Fig. 4.14: Deformed part at the fully loaded stage.
Fig. 4.15: Contour of blank thickness on the formed part (1st design)
Fig. 4.16: Altered part design.
Fig. 4.17: Contour of blank thickness on the formed part (2nd design).
Fig. 4.18: Force-displacement curve for the steel pan.
Fig. 4.19: Thinning strain along part diagonal OA on the top of the blank (steel)
Fig. 4.20: Force-displacement curve for the aluminum pan.
Fig. 4.21: Thinning strain along part diagonal OA on the top of the blank (aluminum).
Fig. 4.22: Top view of perimeter (analysis with K-B yield function).

Fig. 4.23: Thinning strain along part diagonal OA on the top of the blank (aluminum), obtained with Hill's quadratic yield function [1950].
Fig. 4.24: Top view of perimeter (analysis with Hill's quadratic yield function [1950]).
Fig. 5.1: Flowchart of the tooling design algorithm.
Fig. 5.2: Schematic graph of the springforward simulations.
Fig. 5.3: Tool shapes used in the different cycles of the tooling design algorithm.
Fig. 5.4: Produced part shapes form the different cycles of the tooling design algorithm.
Fig. 5.5: Part shape error in the vertical direction for the different cycles of the tooling design algorithm.
Fig. 5.6: Side view of successive steps of the channel forming process with the tools obtained from the tooling design algorithm.
Fig. 5.7: RMS of the part shape error for the different cylcles of the tooling design algorithm.
Fig. 5.8: The three-dimensional channel geometry.
Fig. 5.9: Different sections of the three-dimensional geometry.
Fig. 5.10: Finite element model of the part.
Fig. 5.11: Part shapes at section A-B for different cycles of the tooling design algorithm.
Fig. 5.12: Part shapes at section C-D for different cycles of the tooling design algorithm.
Fig. 5.13: Part shapes at section E-F for different cycles of the tooling design algotothm.
Fig. 5.14: Part shapes at section A-C-E and B-D-F for different cycles of the tooling design algorithm.
Fig. 5.15: Die shapes at section A-B for different cycles of the tooling design algorithm.
Fig. 5.16: Die shapes at section C-D for different cycles of the tooling design algorithm.
Fig. 5.17: Die shapes at section E-F for different cycles of the tooling design algorithm.
Fig. 5.18: Die shapes at section A-C-E and B-D-F for different cycles of the tooling design algorithm.
Fig. 5.19: Tooling set-up fot the three-dimensional channel forming.
Fig. AII-1(a): A flow chart of the numerical procedure used to calculate the yield surface when $\mathbf{B} = \mathbf{0}$.
Fig. AII-1(b): A flow chart of the numerical procedure used to calculate the yield surface when $\mathbf{B} \neq \mathbf{0}$.

List of Tables

Table 1.1: Material levels of symmetry and respective number of indepentent elements of **L**.
Table 1.2: Values of k and c obtained with Method I, and Method II.
Table 1.3: Material constants for the H-19 can stock alloy and for the 2008-T4 alloy.

Contents

1 CHAPTER 1: INTRODUCTION 10

2 CHAPTER 2: MODELING THE ANISOTROPIC BEHAVIOR OF POLYCRYSTALLINE MATERIALS 14
- 2.1 Introduction . . . 14
- 2.2 A generic isotropic yield surface . . . 16
- 2.3 An anisotropic yield surface . . . 19
- 2.4 The "backstress" state variable . . . 24
- 2.5 Procedures for the evaluation of the yield surface . . . 24
  - 2.5.1 Isotropic material . . . 25
  - 2.5.2 Anisotropic material . . . 26
- 2.6 Special cases of anisotropy . . . 27
- 2.7 Comparison with experiments . . . 30
- 2.8 Conclusions . . . 33

3 CHAPTER 3: IMPLEMENTATION OF THE MATERIAL MODEL INTO FINITE ELEMENT ANALYSIS 55
- 3.1 Inversion of the flow rule . . . 55
- 3.2 Numerical implementation into ABAQUS Explicit . . . 59

4 CHAPTER 4: SIMULATION OF THREE- DIMENSIONAL SHEET METAL FORMING PROCESSES USING FINITE ELEMENT ANALYSIS 62
- 4.1 Implicit model of loading and unloading . . . 62
- 4.2 Implicit Results vs. Experimental Results . . . 65
- 4.3 Explicit Model of Loading with Implicit Model of Unloading . . . 66
- 4.4 A case study: Forming of a three-dimensional pan . . . 73
- 4.5 Conclusions . . . 76

5 CHAPTER 5: TOOLING AND BINDER DESIGN FOR SHEET METAL PROCESSES COMPENSATING FOR PART SHAPEERROR DUE TO SPRINGBACK 102
- 5.1 Introduction . . . 102
- 5.2 Tooling and binder design by using springback calculations . . . 104
- 5.3 An application of the tooling design algorithm. Forming of a two-dimensional channel . . . 108
- 5.4 Forming of a three-dimensional part . . . 110
- 5.5 Conclusions . . . 112

6 CHAPTER 6: CONCLUSIONS - FUTURE WORK 134

7 References 138

8 Appendix I: Symmetries of the transformation tensor 147

9 Appendix II: Evaluation of the anisotropy in the case of orthotropic symmetry 149

10 Appendix III: Listing of input decks and VUMAT subroutine 182

11 Appendix IV: Software for data transfer from explicit to implicit analysis 192

12 Appendix V: Code for the interpolation and offset of the surface obtained from the Finite Element Analysis 201

1 CHAPTER 1: INTRODUCTION

The design of a sheet metal forming process is a complicated task which may include several trial and error procedures in order to make the process feasible. Such a design process will have to compensate for three potential defects: wrinkling, tearing and shape errors of the produced part.

The different issues that need to be addressed when a sheet metal forming operation is designed, are shown in the flow chart of Fig. 1.1. In this figure, the forming design procedure is viewed as a plant, where the input is the tooling shape, the workpiece material behavior and the frictional characteristics of the contact between the forming tools and the workpiece. The output of the sheet metal forming design procedure is the formed part, which must be defect-free. In order to make the process successful, hardware trials and/or Finite Element Analysis (FEA) simulation trials must be performed. In these operations the material restraining conditions (binder force, drawbead shape) must be selected in order to make the process feasible. Also the tooling shape may be modified in order to reduce the shape errors of the produced part. We suggest that in the first cycles of the algorithm finite element analysis is used to provide a first good and, perhaps, final guess for the different parameters of the forming procedure (restraining conditions, tooling shape) in order to make the process feasible. Then, hardware trial operations can be used if needed to finely tune the process. As will be also shown in Chapter 5 of this thesis the designs obtained by using FEA of the process were adequate to provide the desired part at the first hardware trial for the forming cases examined.

There has been a considerable research effort at MIT to develop closed loop control algorithms in order to reduce the part shape error and/or avoid tearing and wrinkling defects. Webb and Hardt (1991), developed a transfer function to correlate changes in the die shape with changes in the produced part shape. Ousterhout (1991) successfully

implemented this method in a closed loop control scheme and modified the tooling shape in order to reduce the part shape error. The forming press used by Ousterhout offered capabilities for rapid changes of the tooling shape, as it consisted of two reconfigurable matching discrete dies.

Karafillis and Boyce (1992) developed an alternative closed loop control scheme to reduce the part shape error by using finite element analysis. Also, Karafillis (1992) used the method of Webb and Hardt (1991) and simulated the closed loop control experiments of Ousterhout (1992) for an axisymmetric part. Excellent agreement between the simulations and the experiments was found, suggesting the use of finite element analysis to replace initial trials.

Jalkh et al.(1993) showed how forming trials and finite element analysis can be used in order to determine the appropriate binder force conditions when forming a tapered cup and improve the forming characteristics of aluminum alloys. Also, Cao et al. (1993) used finite element analysis to develop the optimum binder force which will increase the forming height when forming a tapered cup. Karafillis and Boyce (1992) developed a contact algorithm used in different sheet metal and bulk forming simulations, where very good agreement was found between existing experimental results and the results of the plane strain finite element analysis.

The results of the studies mentioned above motivated for further use of finite element analysis in a closed loop control framework such as the one of Fig. 1.1. In this research we concentrated on the use of finite element analysis as a part of a sheet metal forming design procedure. Specifically the tasks accomplished were the following:

(i) Provide adequate phenomenological characterization of the material anisotropy prior to the 3-dimensional finite element analysis. This task is developed in detail in Chapter 2.

(ii) Implement the phenomenological material model into finite element analysis codes. This task is developed in Chapter 3.

(iii) Use this material characterization in the finite element analysis of 3D forming processes by using existing finite element methods (explicit and/or implicit) in order to correctly predict different features of sheet metal forming processes such as strain distributions, material draw-in, forming loads, and springback upon unloading and also establish the effect of the material properties on these features. This task is developed in Chapter 4.

(iv) Develop a tooling design algorithm to compensate for part shape error due to springback and integrate the provided analysis capabilities in a closed loop sheet metal forming design system with emphasis on tooling design procedures to compensate shape errors of the produced part due to springback. This task was considered as a part of a complete sheet metal forming system incorporating different prototyping methods for dies as also investigated by Walczyk (1993), see also Chapter 5.

(v) Demonstrate the precision of the finite element simulation and the effectiveness of the tooling design algorithm by considering the forming of two-dimensional and three-dimensional geometry sheet metal parts, see Chapter 5.

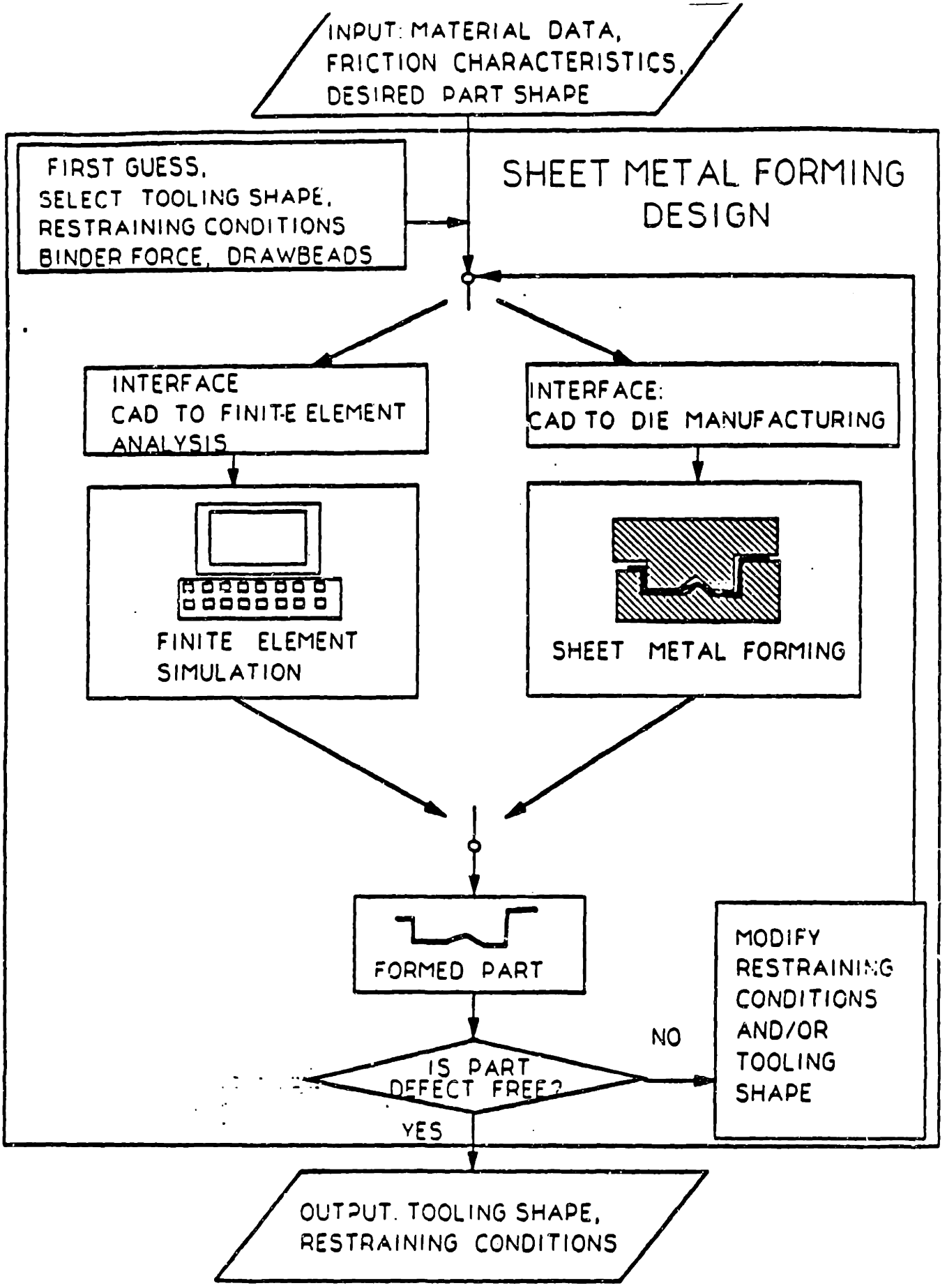


Fig. 1.1 Process design procedures

2 CHAPTER 2: MODELING THE ANISOTROPIC BEHAVIOR OF POLYCRYSTALLINE MATERIALS

2.1 Introduction

The modeling of the plastic behavior of metals is one of the most important aspects in the analysis of metal forming processes. There are primarily two different approaches to follow when describing the plasticity of polycrystalline materials. In the first approach, the Taylor [1938] polycrystalline plasticity model is used to describe both the initial yielding behavior and the subsequent evolution of anisotropy with deformation; Asaro and Needleman [1985], Dawson et. al. [1992], Kalidindi et al.[1992]. However, the use of this formulation is very demanding in computational power when implemented in finite element analysis, see also Dawson et al. [1992]. The Taylor model while reasonably successful in fitting the anisotropic yield surface of FCC and BCC materials, has also been found to greatly overpredict the R-ratio (ratio of transverse to thickness strain in uniaxial tension) for some directions in anisotropic materials, Rollet et al. [1992].

In the second approach, a phenomenological yield function is used to describe the initial yielding behavior, whereas the plastic flow of the material is determined by the flow rule, where the plastic potential function is identical to the yield function. This yield function can be defined appropriately to describe the most important aspects of the plastic behavior of both isotropic and initially anisotropic polycrystalline materials. In this work we will concentrate on the use of phenomenological yield theories which can be more flexible and efficient than polycrystalline plasticity models when implemented in FEA, and which can more accurately predict failure and deformation processing behavior of initially anisotropic sheet metals.

The concepts of the yield surface and the plastic potential in the stress space is

one of the most basic assumptions of the mathematical theory of plasticity. Several representations for the isotropic yield surface of polycrystalline materials have been proposed, including those by Tresca [1864], von Mises [1913], and Hosford [1972]. Anisotropic yield surfaces in the stress space have also been proposed by many authors, the most well known being those of Hill [1950, 1979, 1990], Bassani [1977], Budiansky [1984], and Barlat and coworkers [1989, 1991]. In all of the yield theories listed above an associated flow rule was used, and the yield surfaces were described as convex surfaces in the stress space, as is also required by the maximum dissipation postulate. Also, yield surfaces for polycrystalline materials are generally taken to be pressure independent. In the present work we will be confined to yield surfaces which are homogeneous and convex in the stress space. We will also assume an associated flow rule.

A generic formulation describing anisotropic yield surfaces must be able to describe the isotropic material as a special case. When the isotropic yield surface is recovered as a special case of the anisotropic yield function, it must have the properties of an isotropic function as described, for example, by Gurtin [1981].

Hill's [1950] and Barlat's [1992] anisotropic yield functions include all six components of the stress tensor. In both of these criteria a generalized framework of orthotropic symmetry is used. Hill's [1950] criterion takes the yield surface to be quadratic. However, experimental evidence, Woodthrope and Pearce [1970], Stout et al. [1983], McEwen et al. [1992], and polycrystalline plasticity theories, Bishop and Hill [1951], Hutchinson [1964], indicate that quadratic yield surfaces cannot successfully describe the yield surfaces of polycrystalline materials with an FCC or BCC microstructure. It has been shown by Lian et al. [1989] that the necking behavior in biaxial tension is very sensitive to the shape of the yield locus and therefore a precise description of it is required. Hershey, [1954] and Hosford [1972] have pro-

posed a non-quadratic isotropic yield criterion to more accurately describe the yield surface of polycrystalline materials. The most important capability of this criterion is that it can approximately describe the yield surface of FCC materials as obtained by the Bishop-Hill model. However, this criterion is not flexible enough to describe any desired isotropic yield surface.

Barlat's [1991] criterion is an extension of Hosford's criterion to also include orthotropic anisotropy. However, Barlat's [1991] criterion cannot be used for materials with symmetries other than the orthotropic symmetry (monoclinic, trigonal, asymmetric). Although the assumption of orthotropic symmetry is reasonable for thin rolled sheets, it also possesses a limitation for a generalized yield criterion, as orthotropic anisotropy is associated with only a limited set of strain paths. In general, many deformation histories may yield a totally asymmetric (triclinic) material.

In the present work we develop a new generic isotropic yield criterion for polycrystalline materials. This criterion is flexible and can describe different families of isotropic yield surfaces. An anisotropic yield criterion is presented as an extension of the isotropic criterion. The proposed anisotropic yield criterion uses a set of tensorial state variables to describe the anisotropy and therefore overcomes the deficiency of existing yield criteria which cannot describe materials which do not possess orthotropic symmetry. These tensorial state variables can be incorporated in a state representation of the material which describes both the stress and the anisotropic state and also provides the framework for later including evolution of anisotropy with plastic deformation.

2.2 A generic isotropic yield surface

We consider materials which exhibit pressure independent plasticity and initially consider isotropic materials. Therefore, the yield function is a function of only the principal values of the deviatoric Cauchy stress tensor $\boldsymbol{S}$. Mendelson [1968], has shown

the existence of bounds in an isotropic yield surface of a material with a fixed yield stress in uniaxial tension. These bounds are derived after symmetry and convexity considerations. The lower bound coincides with the limiting maximum shear stress yield surface as described by Tresca [1864], whereas the upper bound corresponds to a limiting value of the sum of the two greater diameters of Mohr's circles as mentioned by Hosford [1972]. These two bounds of the yield surfaces of isotropic materials are depicted in Fig. 2.1. In the deviatoric plane section of the yield surface, the lower bound is inscribed in the von Mises yield circle which in turn is inscribed in the upper bound hexagon, see Fig. 2.1.

Isotropic yield surfaces lying between the bounds defined by the von Mises yield surface and the Tresca yield surface have been mathematically described by Hershey [1954] and Hosford [1972]. Hosford's yield surface is a modification of the von Mises mathematical description of a yield surface, where an exponent different than two is used in the fcllowing yield condition:

$$\Phi_1(\boldsymbol{S}(\boldsymbol{\sigma})) = (S_1 - S_2)^{2k} + (S_2 - S_3)^{2k} + (S_3 - S_1)^{2k} = 2Y^{2k} \tag{2.1}$$

where $S_i = S_i(\boldsymbol{\sigma})$ are the principal values of the deviatoric Cauchy stress tensor, and Y is the yield stress in uniaxial tension (a material property).

In the present work we will confine k to be an integer with values varying from $+1$ to $+\infty$. Thus, the exponent in Eqn. (2.1) is always an even integer, thereby insuring equality of the tensile and compressive yield stress. When $k = 1$, Eqn. (2.1) corresponds to the von Mises yield surface whereas when $k \to \infty$, Eqn. (2.1) corresponds to the Tresca yield surface. Fig. 2.2 shows sections of the yield surface of Eqn. (2.1) in the $\sigma_1 - \sigma_2$ (principal stress) space for different values of k.

In a similar way we can describe yield surfaces lying between the von Mises yield surface and and the upper bound yield surface by using the following yield condition:

$$\Phi_2(\boldsymbol{S}(\boldsymbol{\sigma})) = S_1^{2k} + S_2^{2k} + S_3^{2k} = \frac{2^{2k}+2}{3^{2k}} Y^{2k} \tag{2.2}$$

When $k = 1$ the yield surface of Eqn. (2.2) corresponds to the von Mises yield surface whereas when $k \to \infty$ the upper bound yield surface is recovered. The $\sigma_1 - \sigma_2$ section of the isotropic yield surface of Eqn. (2.2) is shown in Fig. 2.3.

A generic isotropic yield surface should be able to describe all yield surfaces lying between the lower bound and the upper bound yield surface. The most plausible approach to such a generalized mathematical description can be obtained by mathematically mixing Eqns. (2.1) and (2.2). We, therefore, establish the following yield condition:

$$\Phi(\boldsymbol{S}(\boldsymbol{\sigma})) = (1-c)\Phi_1(\boldsymbol{S}(\boldsymbol{\sigma})) + c\frac{3^{2k}}{2^{2k-1}+1}\Phi_2(\boldsymbol{S}(\boldsymbol{\sigma})) = 2Y^{2k} \tag{2.3}$$

where $c \in [0, 1]$. The yield surface of Eqn. (2.3) lies between the bounds defined by Eqn. (2.1) and Eqn. (2.2) and preserves convexity [1]. As we increase the value of k the bounds described by Eqns. (2.1) and (2.2) approach the lower and upper bound respectively. By varying the value of the mixing factor c, we create a family of yield surfaces which is sweeping the space between the two bounds set by the selection of k. Yield surfaces for different values of c and for $k = 15$ are depicted in Fig. 2.4, where we chose $k = 15$ to provide a nearly lower and upper bound representation as Φ_1 and Φ_2.

Equation (2.3) provides a general form for the yield surface spanning from lower to upper bound where the constant Y represents the yield stress in uniaxial tension. When using the isotropic yield function of Eqn. (2.3) we fix the value of k to be high enough, e.g. 15, in order to describe the yield surface bounds, and then we find c simply by obtaining the yield stress in a different stress state, for example in plane strain or in shear (Method I). In an alternative approach, c and k can both be

[1] Convexity of this proposed generic isotropic yield surface in the stress space must be assured. The yield surface is convex if the Hessian matrix Γ of $\Phi(\boldsymbol{S}(\boldsymbol{\sigma}))$ with respect to $\boldsymbol{\sigma}$ is positive semi-definite. The Hessian matrix is derived as: $\Gamma = \frac{\partial^2 \Phi}{\partial \boldsymbol{\sigma}^2}$. Rodin and Parks [1986] have established a criterion for determining the positive semi-definiteness of Γ. It can be shown that Γ is positive semi-definite when $2k > 1$.

considered as material properties (i.e. k is not fixed to be large). In this case, k and c can be calculated to provide the best fit to an isotropic yield surface, as calculated experimentally or with polycrystalline plasticity considerations (Method II).

2.3 An anisotropic yield surface

The yield function of Eqn. (2.3) is an isotropic yield function. It also describes identical behavior in tension and compression. However, we would like to mathematically represent yield surfaces for anisotropic materials. We will again only consider convex and homogeneous yield surfaces.

Barlat et. al. [1991], used a linear transformation of the stress state of the anisotropic material to describe an anisotropic yield surface simply by multiplying all components of the Bishop-Hill stress tensor of the anisotropic material by a different constant. The obtained stress state was then used in Hosford's [1972] yield criterion.

In our development of an anisotropic yield function we will also consider a linear transformation of the actual stress tensor $\boldsymbol{\sigma}$ acting on the anisotropic material. This linear transformation "weights" the different components of the stress tensor of the anisotropic material in order to account for the anisotropy of the material. We call the transformed tensor the "isotropic plasticity equivalent [IPE] deviatoric stress tensor" and will be used as an argument in our isotropic function of Eqn. (2.3). Therefore, for the case of an isotropic material, our linear transformation would simply yield the deviatoric stress tensor of the actual material stress state.

The generalized transformation of the stress state for an anisotropic material is schematically represented in Fig. 2.5. As also shown in Fig. 2.5, the concept of the "IPE material" is introduced. The "IPE stress transformation" introduced here is a transformation of the actual stress state of an anisotropic material to the corresponding stress state of an isotropic material whose yielding behavior is described by Eqn. (2.3). We assign the same equivalent yield strength to both isotropic and anisotropic

materials. We also set the plastic dissipation rate to be equal in both materials at the time of yielding. Note that only the first plastic yield is considered in the developed framework of the transformation, i.e. ongoing evolution in anisotropy with further plastic strain is currently neglected.

The linear transformation of the stress state of the anisotropic material can be effected by multiplicatively operating on the stress tensor σ by a fourth order tensorial operator. Such a transformation is an affine transformation, provided that the fourth order tensorial operator is non-singular and therefore preserves the convexity of the yield surface, Eggleston [1966], as also noted by Barlat et al. [1991] . We will show that the tensorial multiplicative operator used in our "IPE stress transformation" must exhibit some internal symmetries and it must also be traceless. Fourth order tensorial stress concentration operators have also been considered by Dvorak [1982] in conjunction with the von Mises yield criterion for anisotropic composite materials. Also, fourth order tensors have been used in different formulations of quadratic yield criteria by Shih and Lee [1978], Eisenberg and Yen [1984], and Voyadjis and Foroozesh [1990]. As shown by Onat [1984] and developed in detail in Appendix I, fourth order traceless tensors with the internal symmetries required by our transformation are composed of a set of irreducible tensorial variables (one scalar, a second order tensor and a fourth order tensor). With the use of group theory, Backus [1970] it is shown that the fourth order irreducible tensorial component of $\mathbf{L}$ can be associated with material symmetries which range from the lowest level of the triclinic symmetry to the highest level of full symmetry. Therefore the use of the fourth order tensor as a linear multiplicative operator provides a tool for representing different degrees of material symmetry and provides a potential for representation of any anisotropic state of the material.

The "IPE stress transformation" operates as:

$$\tilde{\boldsymbol{S}} = \mathbf{L}\boldsymbol{\sigma} \tag{2.4}$$

where $\mathbf{L}$ is the fourth order tensorial operator, $\tilde{\boldsymbol{S}}$ is the IPE stress tensor, $\boldsymbol{\sigma}$ the actual stress tensor of the anisotropic material. The yield condition of the anisotropic material is:

$$\Phi(\tilde{\mathbf{S}}) = 2Y^{2k} \tag{2.5}$$

where $\tilde{\boldsymbol{S}}$ simply replaces $\mathbf{S}$ in our generic isotropic yield function of Eqn. (2.3) earlier and Y is the average yield stress in uniaxial tension obtained experimentally. In order to estabish the form and the properties of the introduced tensorial operator we consider the stress tensor and the material symmetries. First, by the symmetry of $\boldsymbol{\sigma}$, and $\tilde{\mathbf{S}}$ we have,

$$L_{ijkl} = L_{jikl} = L_{jilk} \tag{2.6}$$

As also discussed in Appendix I, we require $\mathbf{L}$ to also exhibit the symmetry:

$$L_{ijkl} = L_{klij} \tag{2.7}$$

As mentioned above, yielding in the anisotropic material is also pressure independent. Therefore, the values of the components of the "IPE deviatoric stress tensor" should be independent of the hydrostatic pressure component of $\boldsymbol{\sigma}$. We can mathematically express this condition as a property of the fourth order tensorial operator $\mathbf{L}$:

$$L_{ijkk} = 0 \tag{2.8}$$

where the standard summation rules apply.

The symmetries expressed by Eqn. (2.6) and (2.7), together with the constraint of Eqn. (2.8), render $\mathbf{L}$ a traceless partially symmetric fourth order tensor. We

can now derive the population of the independent scalar components of $\mathbf{L}$ after the constraints of Eqns. (2.7-9) for different degrees of material symmetry. In Table 1.1 the rotations which leave $\mathbf{L}$ invariant and the corresponding number of independent components of the tensorial operator are tabulated, where $\boldsymbol{R}_i^{\phi}$ denotes rotation about the i-axis by the angle ϕ. Also, any rotation which is derived as the product of the rotations tabulated in Table 1.1 would leave $\mathbf{L}$ invariant. We see that for isotropic symmetry only one constant can completely describe the tensorial operator whereas in the case of triclinic symmetry 15 components are needed to describe the operator. In Appendix I the three state variables that compose $\mathbf{L}$ are shown: a scalar a, a second order traceless and symmetric tensor $\mathbf{A}$, and a fourth order traceless and symmetric tensor $\mathbf{C}$, as shown in the following equation.

$$\begin{aligned} L_{ijkl} &= a\delta_{ij}\delta_{kl} - \frac{3}{2}a(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \\ &\quad +\delta_{ij}A_{kl} + A_{ij}\delta_{kl} \\ &\quad -\frac{3}{4}(\delta_{ik}A_{jl} + \delta_{il}A_{jk} + \delta_{jk}A_{il} + \delta_{jl}A_{ik}) \\ &\quad +C_{ijkl} \end{aligned} \tag{2.9}$$

These variables in combination with the stress tensor $\boldsymbol{\sigma}$ and the equivalent yield stress Y can be used to represent the state of the material, see Appendix I. The tensorial variables in this representation can be rotated with the tensor rotation rule when the reference frame is rotated and still be compatible with the IPE stress transformation as both the anisotropic and the equivalent isotropic material are attached to the same reference frame. Also evolution equations for the developed tensorial state variables may provide a tool for the description of the evolution of the anisotropy of a material subject to mechanical processing and/or thermal treatment. However, the validity of this hypothesis requires further investigation.

The equality of the plastic dissipation rate of the general anisotropic material and

the equivalent isotropic material can now be considered. It is:

$$\dot{W} = \boldsymbol{\sigma} \cdot \boldsymbol{D}^p = \tilde{\boldsymbol{S}} \cdot \boldsymbol{d}^p = Y\dot{\bar{\epsilon}}^p \tag{2.10}$$

where $\dot{W}$ is the plastic dissipation rate, $\boldsymbol{D}^p$ is the symmetric part of the plastic velocity gradient tensor (plastic stretching) at the moment of the first yield of the anisotropic material, whereas $\boldsymbol{d}^p$ is the work conjugate of the IPE deviatoric stress $\tilde{\boldsymbol{S}}$. We assume an associated flow rule, and therefore the yield function can also be used as a plastic potential. If λ is a scalar such that:

$$\mathbf{d}^p = \lambda \frac{\partial \Phi}{\partial \tilde{\boldsymbol{S}}} \tag{2.11}$$

then by using the homogeneity of Φ, ($\tilde{\mathbf{S}} \cdot \frac{\partial \Phi}{\partial \tilde{\boldsymbol{S}}} = 2k\Phi$), combined with Eqn. (2.10) we obtain:

$$\dot{W} = \lambda \tilde{\boldsymbol{S}} \cdot \frac{\partial \Phi}{\partial \tilde{\boldsymbol{S}}} = 4\lambda k Y^{2k} \tag{2.12}$$

Therefore:

$$\lambda = \frac{\dot{W}}{4kY^{2k}} \tag{2.13}$$

and

$$\mathbf{d}^p = \frac{\dot{W}}{4kY^{2k}} \frac{\partial \Phi}{\partial \tilde{\boldsymbol{S}}} = \frac{\dot{W}}{4kY^{2k}} \mathbf{Q}^T \frac{\partial \Phi}{\partial \boldsymbol{\Lambda}} \mathbf{Q} \tag{2.14}$$

where $\mathbf{Q}$ is the rotation matrix whose rows are the eigenvectors of $\tilde{\boldsymbol{S}}$. $\tilde{\boldsymbol{S}}$, and $\boldsymbol{\Lambda}$ is the principal tensor of $\tilde{\boldsymbol{S}}$ obtained as:

$$\boldsymbol{\Lambda} = \sum_{i=1}^{3} S_i \boldsymbol{v}_i \otimes \boldsymbol{v}_i \tag{2.15}$$

where $\boldsymbol{v}_i$ are the unit vectors.

Also, from the flow normality rule and by using the symmetry of $\mathbf{L}$ expressed by Eqn. (2.8) we obtain:

$$\mathbf{D}^p = \frac{\dot{W}}{4kY^{2k}} \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} = \frac{\dot{W}}{4kY^{2k}} \mathbf{L} \frac{\partial \Phi}{\partial \tilde{\boldsymbol{S}}} = \mathbf{L}\mathbf{d}^p \tag{2.16}$$

Eqn. (2.16) constitutes the flow rule for the anisotropic material as derived from the plastic potential proposed in the present work. Operationally, we will determine the plastic strain ratios or R-values using Eqn. (2.16).

2.4 The "backstress" state variable

The isotropic and anisotropic yield functions presented in the above sections assume that the yield stress in tension is equal to the yield stress in compression in all the directions. However, it has been observed that plastic deformation in uniaxial compression typically initiates at a lower yield stress than in uniaxial tension, if the specimen has been already plastically stretched in uniaxial tension, ("Bauschinger" effect). Also, differences between the tensile and the compressive yield stress can be associated with crystaliographic phenomena (twinning) in HCP materials, see also Backofen [1972]. The proposed yield surface is not capable of describing the "Bauschinger" phenomenon. However, we can further expand Eqn. (2.5) as:

$$\tilde{\boldsymbol{S}} = \mathbf{L}(\boldsymbol{\sigma} - \mathbf{B}) \tag{2.17}$$

where $\boldsymbol{B}$ is an irreducible symmetric and traceless tensorial state variable of second order, which results in different yield stresses in tension and compression. The flow rule in this case remains as given by Eqn. (2.16). The "backstress" $\mathbf{B}$ can be included in the set of irreducible tensorial variables which describe the state of the material, see also Appendix I.

2.5 Procedures for the evaluation of the yield surface

We will concentrate now on the procedures for the estimation of the yield surface of anisotropic materials based on experimental material data.

2.5.1 Isotropic material

We consider now the two different methods for the calculation of an isotropic yield surface. In the first method (Method I) we fix k to a high enough value ($k = 15$) to provide a nearly lower and upper representation of Φ_1 and Φ_2. Then we modify c until a desired value of the yield stress in plane strain tension is reached. This value can be obtained either experimentally or by using polycrystalline plasticity considerations. However, as it is difficult to obtain experimental results for the isotropic state of commercially available thin rolled materials, we used the Bishop-Hill method as it is implemented in the 'popLA' code, developed by Kallend et al. [1989], where a fully symmetric Sample Orientation Distribution Function was used to calculate the average macroscopic plane strain yield stress. Also, the restricted glide assumption was used for FCC materials and the pencil glide assumption was used for BCC materials. The computed Bishop-Hill surfaces are shown in Fig. 2.6(a). The values of c obtained using this method are tabulated in Table 1.2. Also, the sections of the yield surfaces obtained in the $\sigma_1 - \sigma_2$ domain are virtually identical with those obtained with popLA, compare Figs. 2.6(a) and 6(b).

In the second method (Method II) we modified both k and c until we obtained a precise description of the Bishop-Hill yield surface. In order to graphically represent the detailed features of the shape of the yield surface, we depicted it by using Lode's variables [1926], $\nu = 3(D_{\max} + D_{\min})/(D_{\max} - D_{\min})$ and $\alpha = (S_{\max} - S_{\min})/Y$. $D_{\max}$ and $D_{\min}$ are respectively the maximum and minimum principal components of the plastic stretching rate tensor as derived from the flow normality rule whereas $S_{\max}$ and $S_{\min}$ are respectively the maximum and minimum principal components of the deviatoric stress tensor. Correlation between these variables describes completely and in detail all the particular features of the shape of the isotropic yield surface. The exponent k and the mixing coefficient c of the isotropic function of Eqn.

(2.3) were determined to provide a best fit to the $\alpha - \nu$ diagram as derived from polycrystalline plasticity considerations, see Fig. 2.7. The values of these parameters for the different cell-structures considered here are tabulated in Table 1.2. The obtained yield surfaces in the $\sigma_1 - \sigma_2$ domain were almost identical with the ones obtained by popLA and Method I, see also Figs. 2.6 (a) and (b), where the largest differences occur near corner-like regions. However, some differences still exist in both the $\sigma_1 - \sigma_2$ domain and the $\alpha - \nu$ domain, and the yield surface obtained by Method II provides a better description of the Bishop-Hill yield surface. These differences may drastically affect the Forming Limit Diagram (FLD) of the isotropic material, as also mentioned by Lian and Barlat [1989], because it is the shape of the yield surface and its corresponding normal which determines the straining and thining behavior of the sheet. Therefore the appropriate Method and combination of coefficients to describe the yield surface of the isotropic material should be selected by comparison with experimental measurements of the yield surface and/or measurements of the FLD.

Prior to comparing the model predictions and capabilities with experimental data, we first show the general anisotropic features of different symmetry aspects of the model.

2.5.2 Anisotropic material

In order to describe the plasticity of an anisotropic material we must estimate the values of the independent components of the anisotropy tensorial operator by using experimental data. We will concentrate on thin rolled metallic materials which usually exhibit orthotropic symmetry as a result of the geometry of the rolling process and the small thickness of the sheet. In Appendix II we show the form of the tensorial operator when the orthotropic axes of the material are aligned with the reference

system coordinate axes. The IPE function of the IPE material is determined by using Method I, where $k = 15$ and c is determined in order to provide the best fit to the experimental data due to its important effect on the shape of the yield surface and the R-ratio, see also Appendix II. Alternatively we could use Method II and determine both k and c in order to control the shape of the yield surface to a greater extent. However, for the materials examined here we found Method I a very good approximation. Also, the procedure followed to calculate the independent components of the tensorial operator is developed in detail in Appendix II, where three tension tests are required to completely determine $\mathbf{L}$ and c. These tests are uniaxial tests of specimens cut at 0^o, 45^o and 90^o with respect to the rolling direction.

2.6 Special cases of anisotropy

In order to demonstrate the ability of the proposed method to describe the first yield of materials with different symmetries we consider two different cases:

Cubic symmetry: We consider a material possessing properties which remain invariant under rotations of 90^o about all three orthonormal axes. Thus, the IPE transformation tensor $\mathbf{L}$ will have to remain invariant under these rotations. By using this property and the Eqns. (2.6-8), we can derive that the non zero components of $\mathbf{L}$ will follow the relations:

$$\begin{aligned} L_{1111} = L_{2222} &= L_{3333} \\ L_{1122} = L_{1133} &= L_{2233} = -\frac{L_{1111}}{2} \\ L_{1212} = L_{2323} &= L_{3131} \end{aligned} \tag{2.18}$$

It is also required by symmetry considerations that $\mathbf{B} = \mathbf{0}$. Therefore, as also tabulated in Table 1.1, only L_{1111} and L_{1212}, i.e two scalars, are necessary to describe the IPE transformation tensor when the material exhibits cubic symmetry. In Fig. 2.8 we plot the yield stress for different angles of rotation about the 3-axis where

the ratio of $\beta = (\frac{4L_{1212}}{3L_{1111}} - 1)$ was used as an anisotropy parameter. The parameter β provides a measure for the deviation of the material from the isotropic behavior. When $\beta = 0$ the material is isotropic and when the absolute value of β increases, the cubic anisotropy becomes stronger. The yield stress was calculated by using the isotropic yield function of Eqn. (2.3), in conjunction with the IPE transformation as also indicated in Eqn. (2.5). For example purposes, the values of c and k were the ones which correspond to isotropic FCC materials as tabulated in Table 1.2, (Method I). We see that we can successfully describe the cubic symmetry as the yield stress remains identical under rotations of 90^o. We also note the presence of four peaks in the yield stress which corresponds with the presence of four "ears" during the cup forming and/or deep drawing of most aluminum sheets. Also, as shown in Fig. 2.8, the location of the ears can change when the sign of β alternates. Also, in Fig. 2.9 we depict the yield stress for different angles of rotation about the **p**-axis, where $\mathbf{p} = \frac{\mathbf{e_1}+\mathbf{e_2}+\mathbf{e_3}}{\sqrt{3}}$ and $\mathbf{e}_i, i = 1, 2, 3$ are the unit vectors along the $1, 2, 3$ axis respectively. We note that in this case we observe six-peaks in the yield stress variation along a rotation of 180^o around the **p**-axis. This phenomenon indic tes the existence of six-ears in a cup formation of a thin disk made of a material that exhibits cubic symmetry and for which the **p**-axis is normal to the plane of the disk. This behavior cannot be described by quadratic yield surfaces, see also Hill [1950].

We note here that six-ears in cup forming of single crystal sheets were observed by Tucker [1961] for cubic crystals having their **p**-axis normal to the sheet. Therefore, altough the anisotropic yield surface proposed here was not constructed with the intention of describing the plastic behavior of single crystals, we see that it can very successfully describe the way in which the symmetries of the material and associated orientations can affect its plastic behavior.

Trigonal symmetry:In this case, the material properties remain invariant for rotations of 120° about the 3-axis and for a reflection with respect to the $1-2$ plane. Although this case is not the result of a specific forming process, we present it here in order to illustrate how different material symmetries may affect the material's plastic behavior. By using this property we obtain the following relations for the non-zero components of $\mathbf{L}$:

$$\begin{aligned}
L_{1111} &= L_{2222} \\
L_{1122} &= \frac{L_{3333} - 2L_{1111}}{2} \\
L_{1133} &= L_{2233} = -\frac{L_{3333}}{2} \\
L_{1123} &= -L_{2223} \\
L_{2323} &= L_{1313} \\
L_{1213} &= L_{1123} \\
L_{1212} &= \frac{L_{1111} - L_{1122}}{2}
\end{aligned} \tag{2.19}$$

It is also required by symmetry considerations that $\mathbf{B} = \mathbf{0}$. From Eqn. (2.18) it can be seen that only $L_{1111}, L_{1123}, L_{3333}$ and L_{2323} are necessary to fully describe $\mathbf{L}$.

In Fig. 2.10, we depict the variation of the yield stress as a function of the rotation about the 3-axis. For example purposes, the values of c and k in function Φ were obtained to be the ones which correspond to FCC materials as tabulated in Table 1.2 (Method I). We assumed that $L_{1111} = L_{2222} = L_{3333} = \frac{4}{3}L_{1212}$ (same as for the isotropic material), whereas the ratio $\zeta = \frac{L_{1123}}{L_{1111}}$ was used as the trigonal anisotropy parameter, where large values of ζ indicate strong trigonal anisotropy. It can be seen in Fig. 2.10 that peaks in the yield stress are 60° apart, indicating the formation of 6-ears in a cylindrical cup-forming operation.

The case of trigonal anisotropy shows that the introduction of components of $\mathbf{L}$ which destroy its orthotropic symmetry can affect the anisotropic behavior of the

material by changing the amount of ears observed in a cup forming operation.

2.7 Comparison with experiments

We now use the proposed yield surface to describe the anisotropy of thin rolled aluminum alloys, where orthotropic symmetry is assumed. The yield function of the IPE material is determined following Method I, where k is fixed to a high value to approximately describe the upper and lower bound ($k = 15$), whereas the mixing factor c is modified to provide the best agreement with experiments due to its important effect on the shape of the yield surface and the distribution of the R-ratio, see also Appendix II. This method was found very effective for all the examined aluminum alloys. The procedure to calculate the independent parameters of $\mathbf{L}$ and the value of c is developed in Appendix II. In this procedure, only the R-ratio and average yield stress from three tension tests in three different directions ($0^o, 45^o$ and 90^o) are required to determine $\mathbf{L}$. The yield stress distribution and yield surface can then be predicted. The value of the mixing coefficient c of the IPE yield surface can be modified to optimize the prediction of the IPE yield surface. Alternatively, we can use Method II to describe the IPE yield surface and modify both k and c to optimize the predictions of the yield stress distribution and/or the yield surface.

Two different materials were considered: the H19 can stock aluminum and the 2008-T4 aluminum. Results concerning the anisotropic behavior of the H19 aluminum at initial yield were obtained from MacEwen et al. [1992]. For this material we assumed that $\mathbf{B} = \mathbf{0}$. In their work, MacEwen et al. measured the experimental data which are necessary to determine the anisotropy of the material. Therefore we were able to completely determine $\mathbf{L}$, assuming orthotropic anisotropy. This determined IPE transformation tensor $\mathbf{L}$ was used in conjunction with the isotropic function Φ. The values of all of the corresponding coefficients are tabulated in Table 1.3.

The variation of the R-ratio as a function of the angle from the rolling direction is depicted in Fig. 2.11 (a). The experimental values of the R-ratio at 0^o, 45^o and 90^o were the only data used to determine $\mathbf{L}$ and therefore there was very good agreement between theory and experiment, as also shown in Fig. 2.11 (a). We can now *predict* the variation of the tensile yield stress in uniaxial tension with respect to the angle from the rolling direction. All yield stress measurements were obtained at 0.1% offset. Also, error bars for the yield stress measurements were not reported. In Fig. 2.11(b), predictions and experimental values of the yield stress for different angles to the rolling direction are depicted, where in excellent agreement between theory and experiment is observed. All stresses in Fig. 2.12 (b) are normalized with the experimental average tensile yield stress. We see that our anisotropic function which was determined by R-ratio measurements in different stress states can also satisfactorily predict the variation of the yield stress in different directions, thereby indicating the effectiveness of the yield function. Also, in Fig. 2.12 the projection of the yield surface in the $\sigma_x - \sigma_y$ plane is depicted, where x is the rolling direction and y the transverse direction . The stresses in this graph are normalized with the experimental average tensile yield stress. Again, we see the excellent agreement between the prediction with the present theory and the biaxial experimental results as opposed to the predictions with Hill's quadratic (1950) criterion.

Similar results and conclusions are obtained for the 2008-T4 aluminum. For this material, experimental data were obtained by Lege et. al. [1989]. The yield stress was measured in 0.2% offset with a reproducibility of the order of 4%), whereas the reproducibility in the measurements of the R-ratio was of the order of 1%. In a first approach we assume that $\mathbf{B} = \mathbf{0}$ for this material. The strain ratio data in different directions were used in order to calculate $\mathbf{L}$ by following the procedures described in Appendix II. The results are shown in Fig. 2.13(a). We see again that our yield

function successfully describes the variation of the yield stress and the R-ratio in the different directions. The results obtained by using the proposed yield function and the suggested procedures for determining $\mathbf{L}$ can be compared with the results obtained by using Barlat's yield criterion as presented by Barlat et. al. [1991] for the same material, see Fig. 2.13(a). Before any comparisons are made, we note that Barlat's criterion can be recovered as a special case of the proposed criterion, if we set $c = 0$ in the function of Eqn. (2.3). It can also be shown that the coefficients of Barlat's yield function can be obtained as linear combinations of the independent components of $\mathbf{L}$. We see that the proposed yield function and the procedures to estimate its parameters can better describe the variation of the R-ratio, see Fig. 2.13(a), and correctly predict the yield stress distribution, see Fig. 2.13(b). Also in Figs. 2.13 (a) and (b) we show the variation of the R-ratio and and the tensile yield stress respectively as predicted in the work of Lege et al. [1989] by using crystallographic distribution data in combination with the Bishop-Hill method, where we see that the polycrystalline plasticity is not very successful in describing the distribution of the R-ratio and the tensile yield stress as opposed to the proposed yield function. We also emphasize that the ability to accurately predict the R-ratio and its anisotropy is important in the material characterization for forming simulations as this greatly influences the predicted thinning of the sheet during a process.

In Fig. 2.14(a) we show the distribution of the yield stress for the 2008-T4 aluminum alloy when the "backstress" tensorial state variable is also invoked. The "backstress" variable was introduced in order to describe the difference of the tensile and compressive yield stress as obtained in the experiments of Lege et al. [1989]. Orthotropic symmetry implies that all of the non-diagonal components of $\mathbf{B}$ are zero. The components of $\mathbf{B}$ for this material are shown in Table 1.3. The obtained distributions of the yield stress compare well with the experimental data. Also, the

obtained distribution of the R-ratio was again in excellent agreement with the experimental data, see Fig. 2.14 (b). The predicted yield surfaces with and without the "backstress" tensorial variable are shown in Fig. 2.15. The obtained yield surface accounts for the different yield stresses in tension and compression as found by the experimental data. Furthermore, we note the important effect of the introduction of a "small" backstress $\mathbf{B}$ on the shape of the yield surface in order to yield the same distribution of the R-ratio. We observe that by introducing $\mathbf{B}$ we also significantly increase the yield stress in the biaxial tension regime, see Fig. 2.15. This observation suggests that the "anomalous" behavior of aluminum, as described by Woodthrope and Pearce [1970], and Stout et al. [1983] (high biaxial yield stress combined with low R-ratio) could be partially attributed to the existence of Bauschinger phenomena. Hill's [1979] yield function was introduced to represent a yield surface which could describe the anomalous behavior of aluminum but without including any "Bauschinger" phenomena.

2.8 Conclusions

A generalized yield criterion using bounds and a transformation weighting tensor has been developed. The proposed yield function can describe both isotropic and anisotropic materials. The anisotropy is described by a linear transformation of the stresses. This linear transformation is effected by the use of a set of tensorial variables which can describe both anisotropy and Bauschinger phenomena. The proposed yield criterion is capable of describing different states of material symmetry, including the most generalized case of the totally asymmetric material, in combination with a non-quadratic yield criterion. The yield criterion was found to be in good agreement with experimental results available in the literature for a variety of materials. For the special case of orthotropic materials, and if we neglect Bauschinger effects, only three tension tests are required to fully determine the anisotropy of the material. Also the

use of the tensorial variables in combination with the compact form of the proposed non-quadratic yield criterion render the proposed yield function very convenient when implemented in computational procedures and also provides a framework for later including evolving anisotropy.

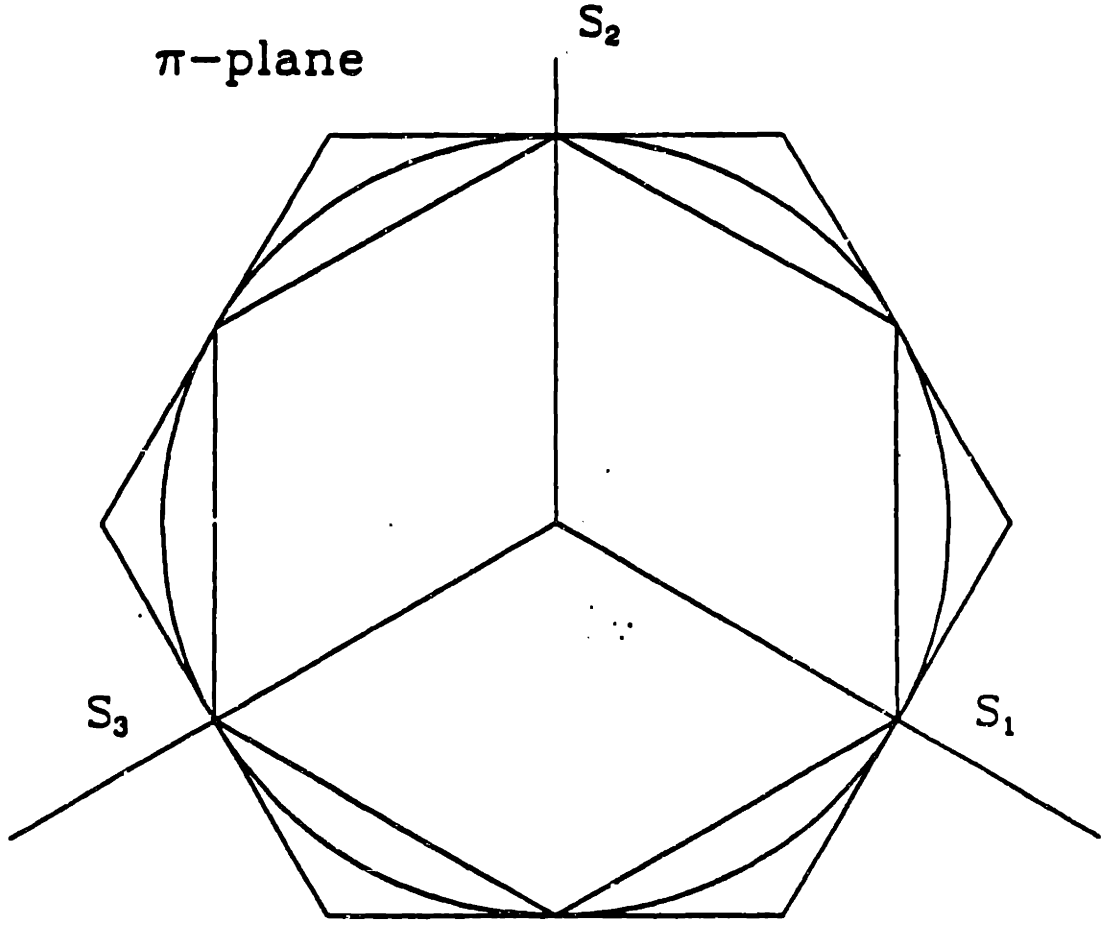


Fig. 2.1: The upper bound, the lower bound and the von Mises yield surface in the π-plane.

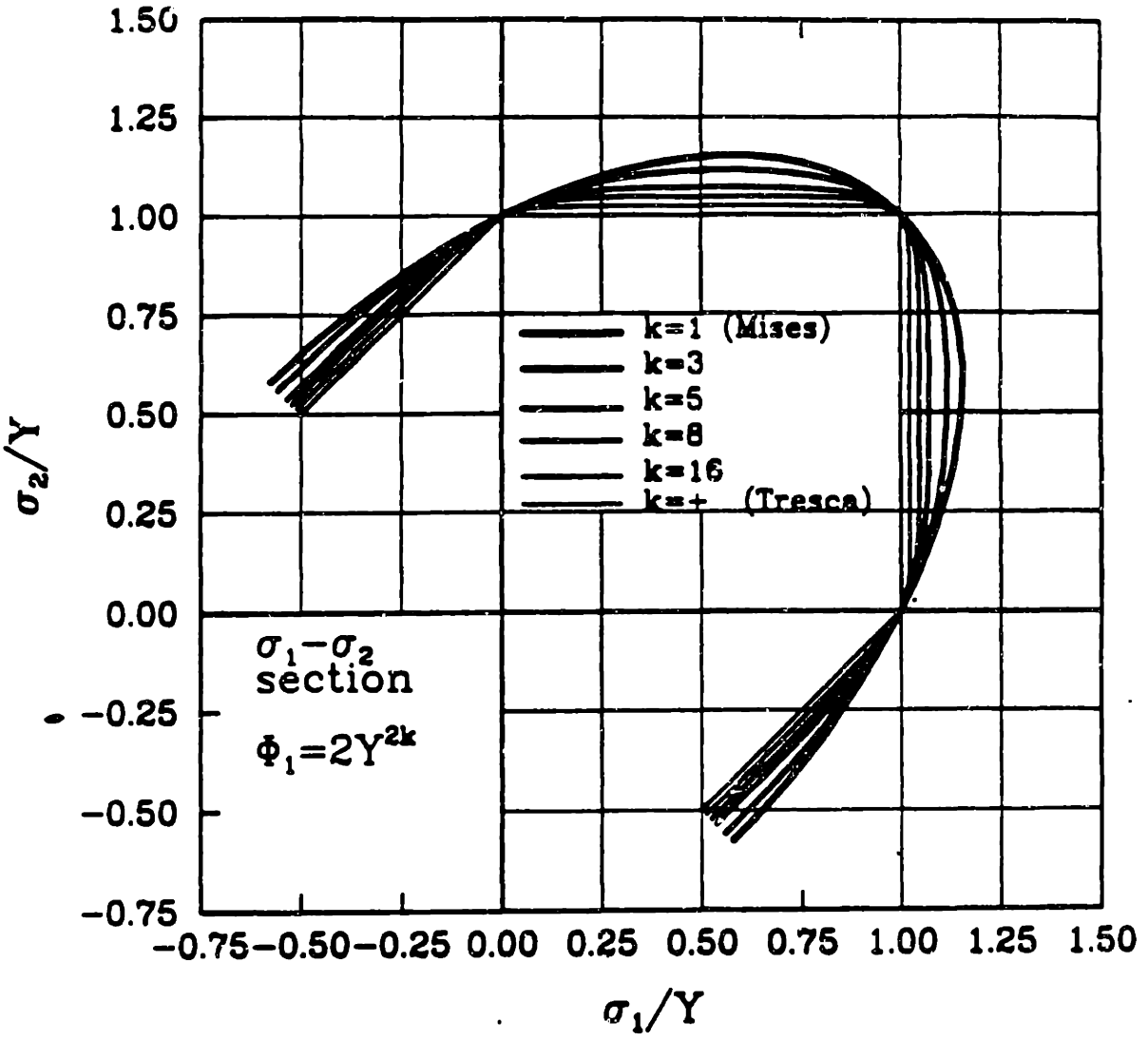


Fig. 2.2 Different isotropic yield surfaces between the von Mises yield surface and the lower bound (Tresca).

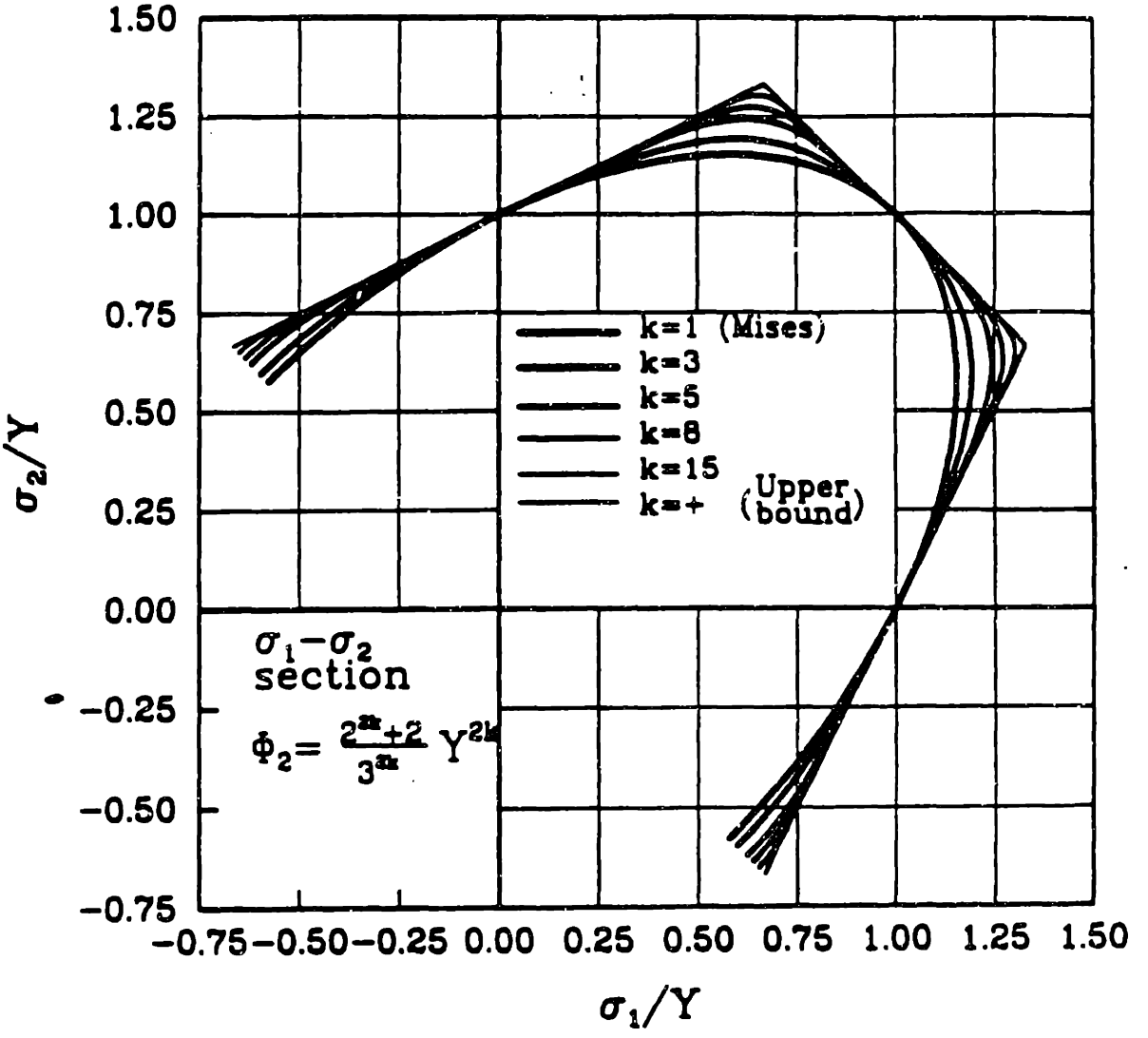


Fig. 2.3 (Different isotropic yield surfaces between the von Mises yield surface and the upper bound.

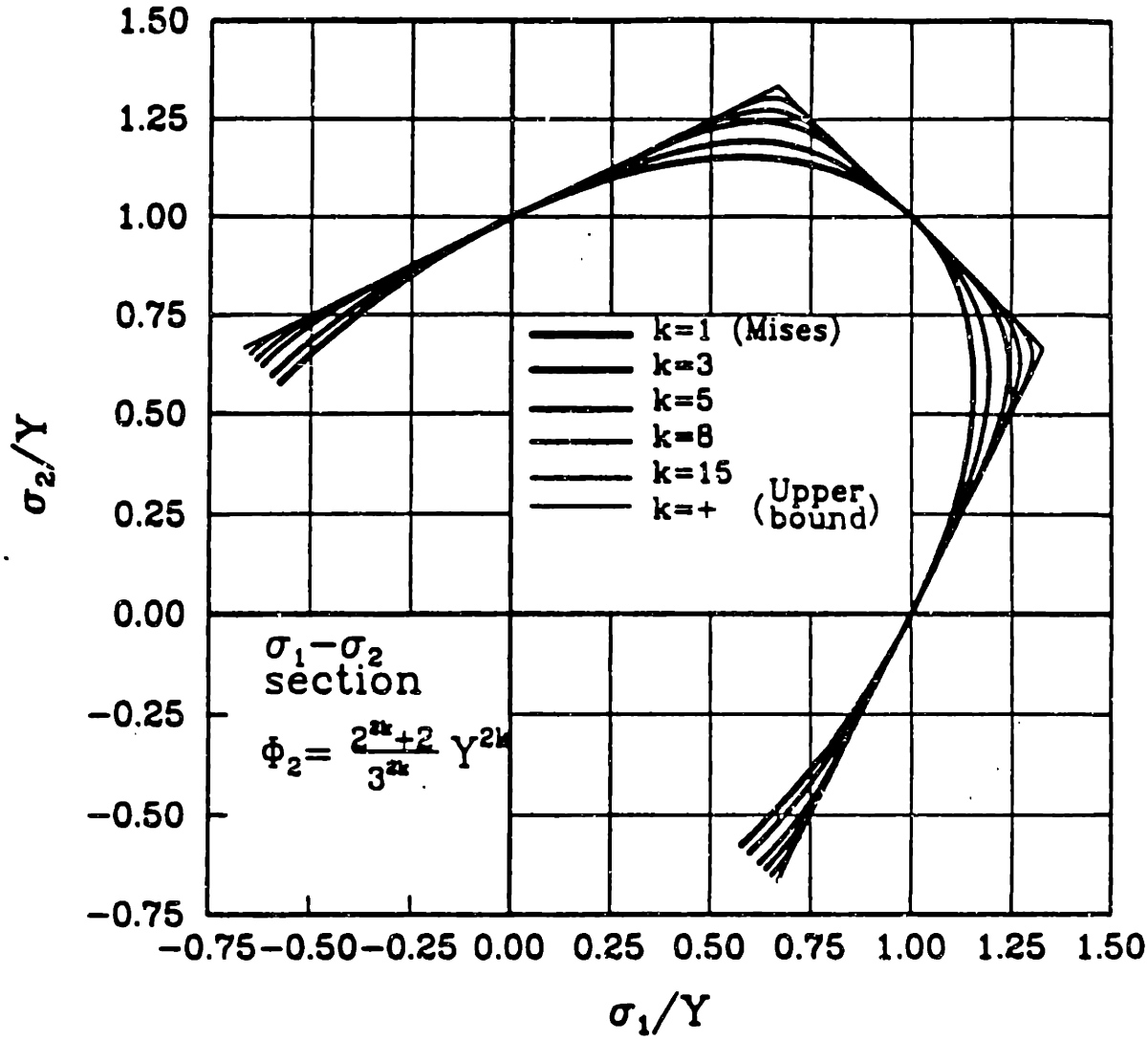


Fig. 2.4 : Different isotropic yield surfaces between the von Mises yield surface and the upper bound.

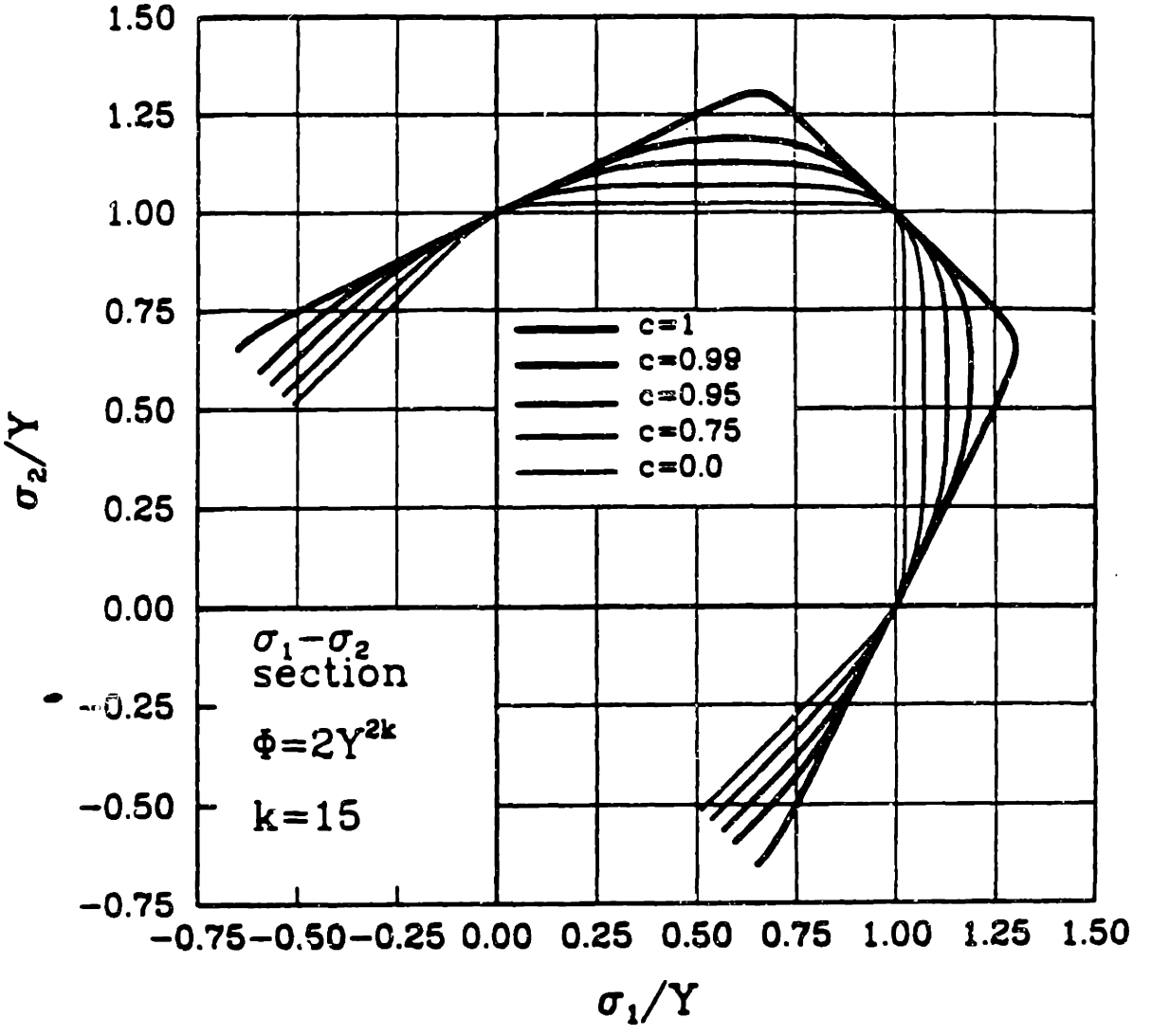
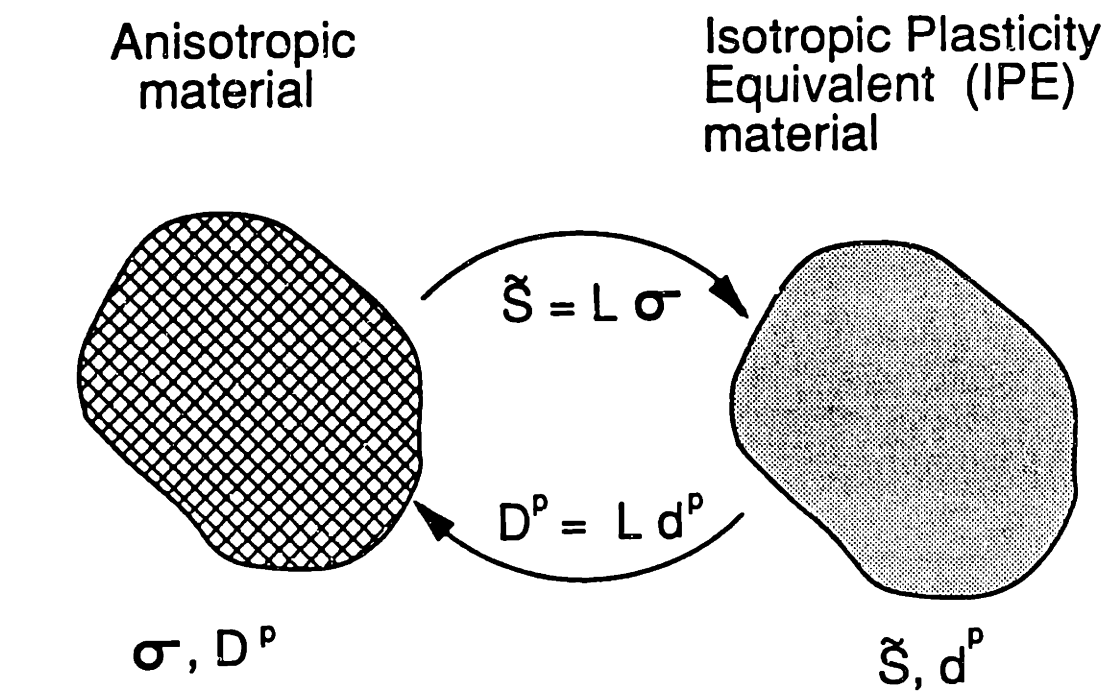


Fig. 2.5 : Different isotropic yield surfaces between the lower bound (Tresca) and the upper bound.



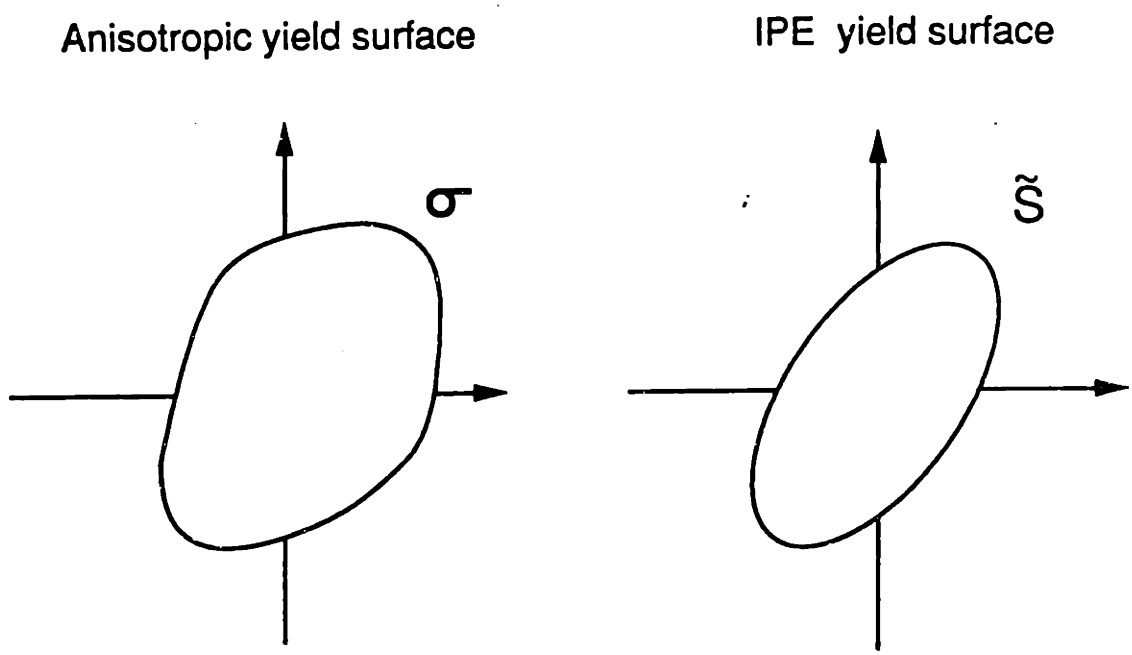


Fig. 2.6: The anisotropic material, the Isotropic Plasticity Equivalent (IPE) material and the IPE stress transformation.

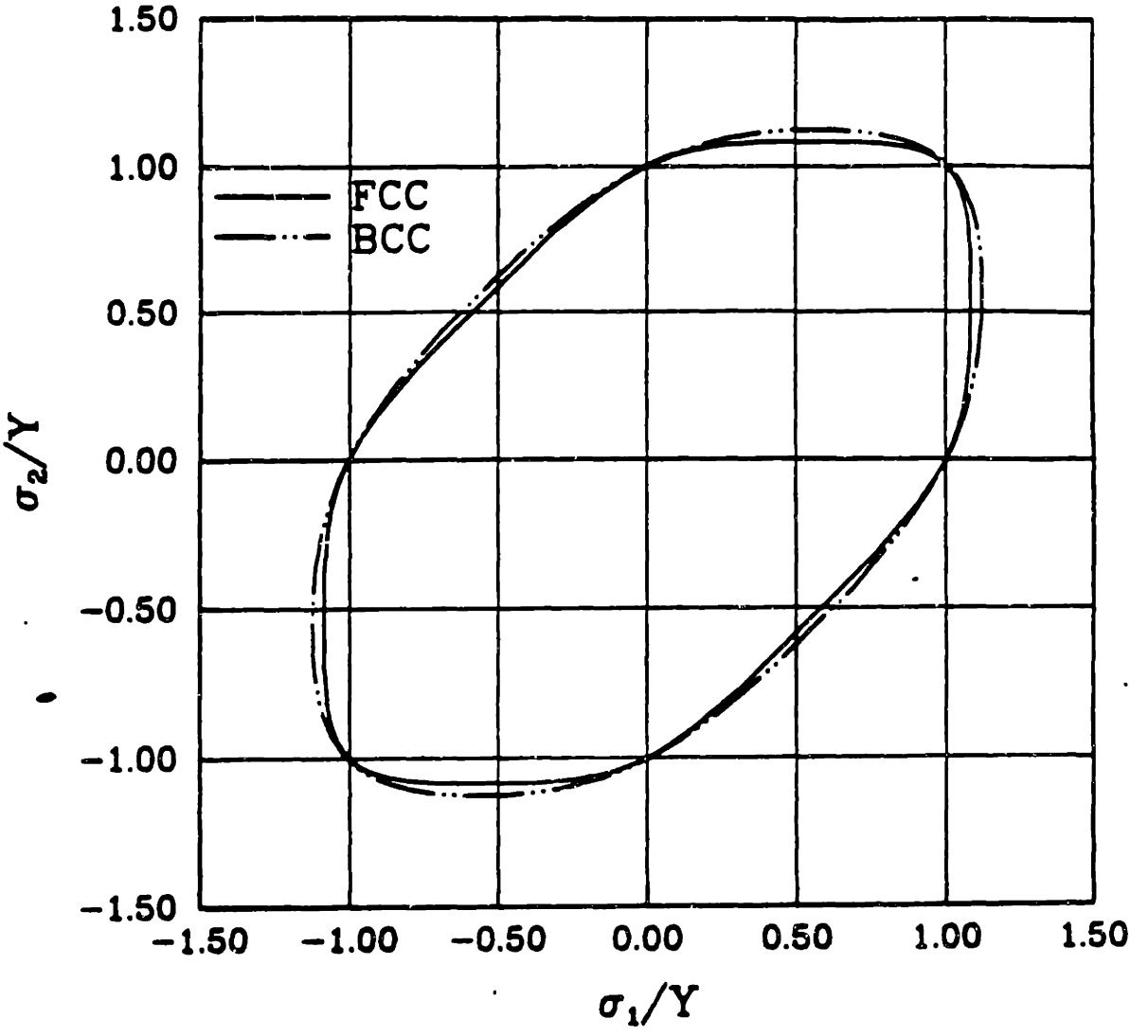
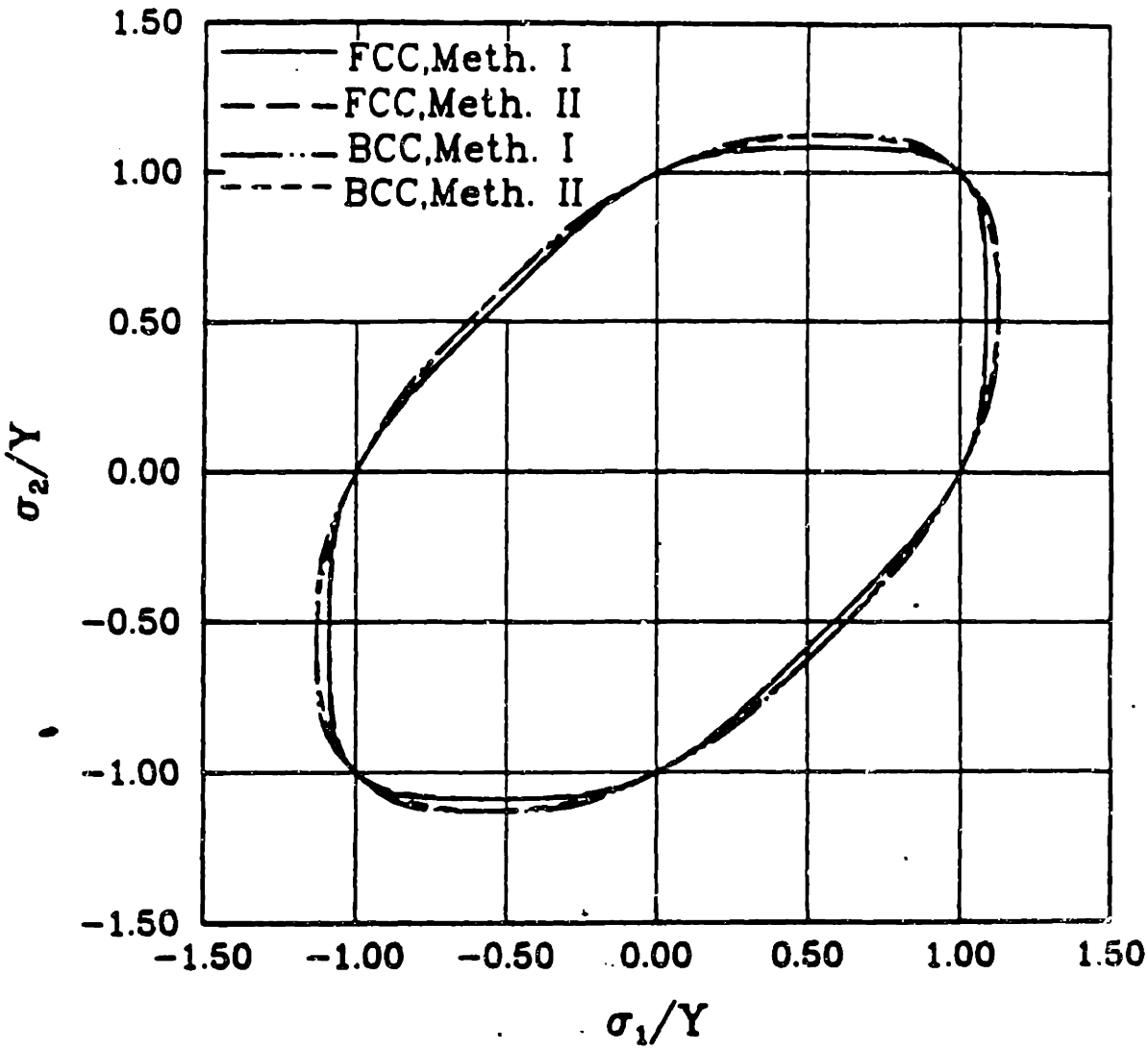


Fig. 2.7 :(a) Isotropic yield surfaces for FCC (restricted glide) and BCC (pencil glide) materials obtained by popLA.



(b) Isotropic yield surfaces as obtained by using Method I and Method II.

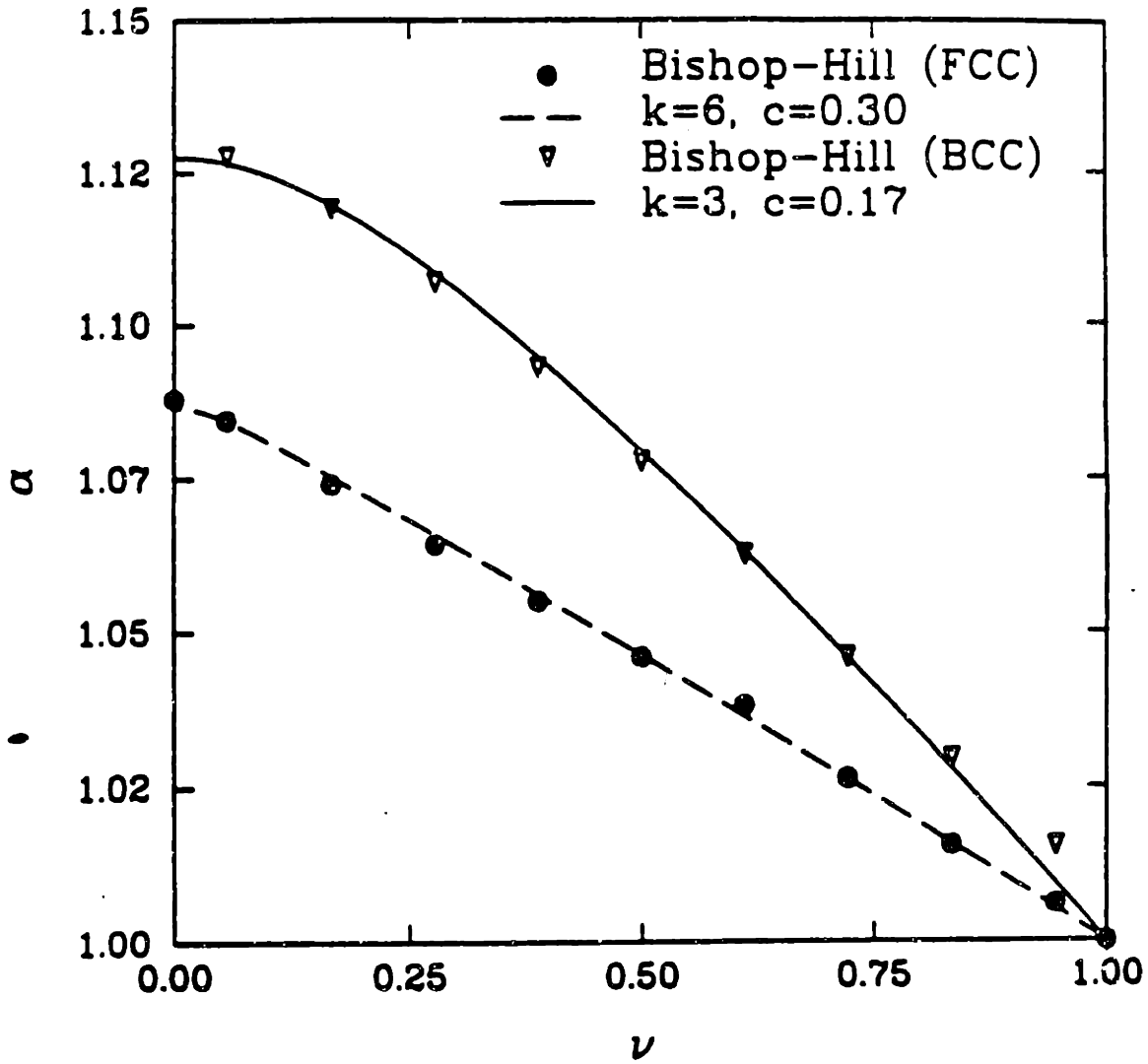


Fig.2.8 Different isotropic yield surfaces in the $\alpha - \nu$ domain.

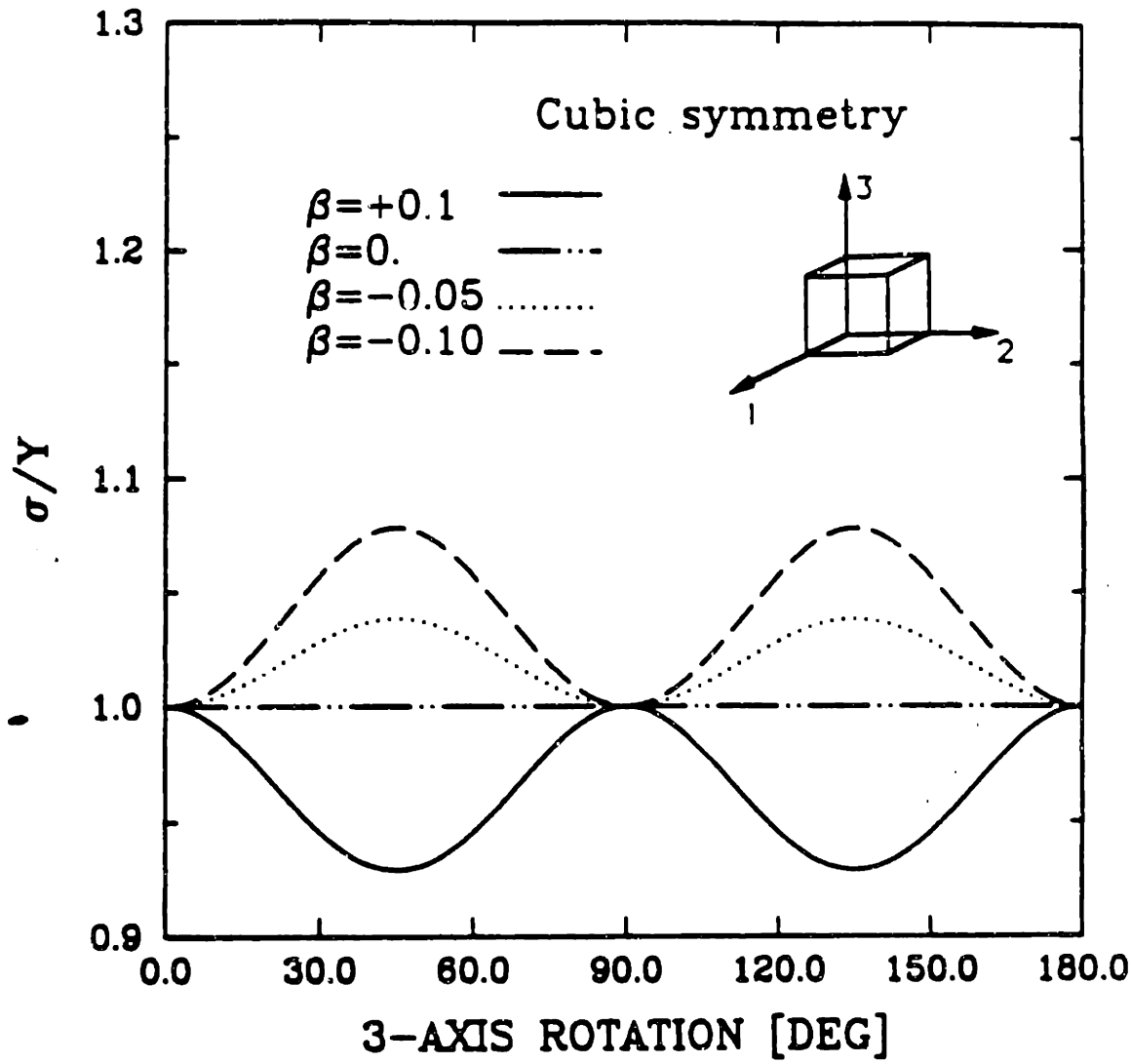


Fig. 2.9: Distribution of the uniaxial yield stress in different angles about the 3-axis for a material with cubic symmetry. The parameter β indicates degree of anisotropy.

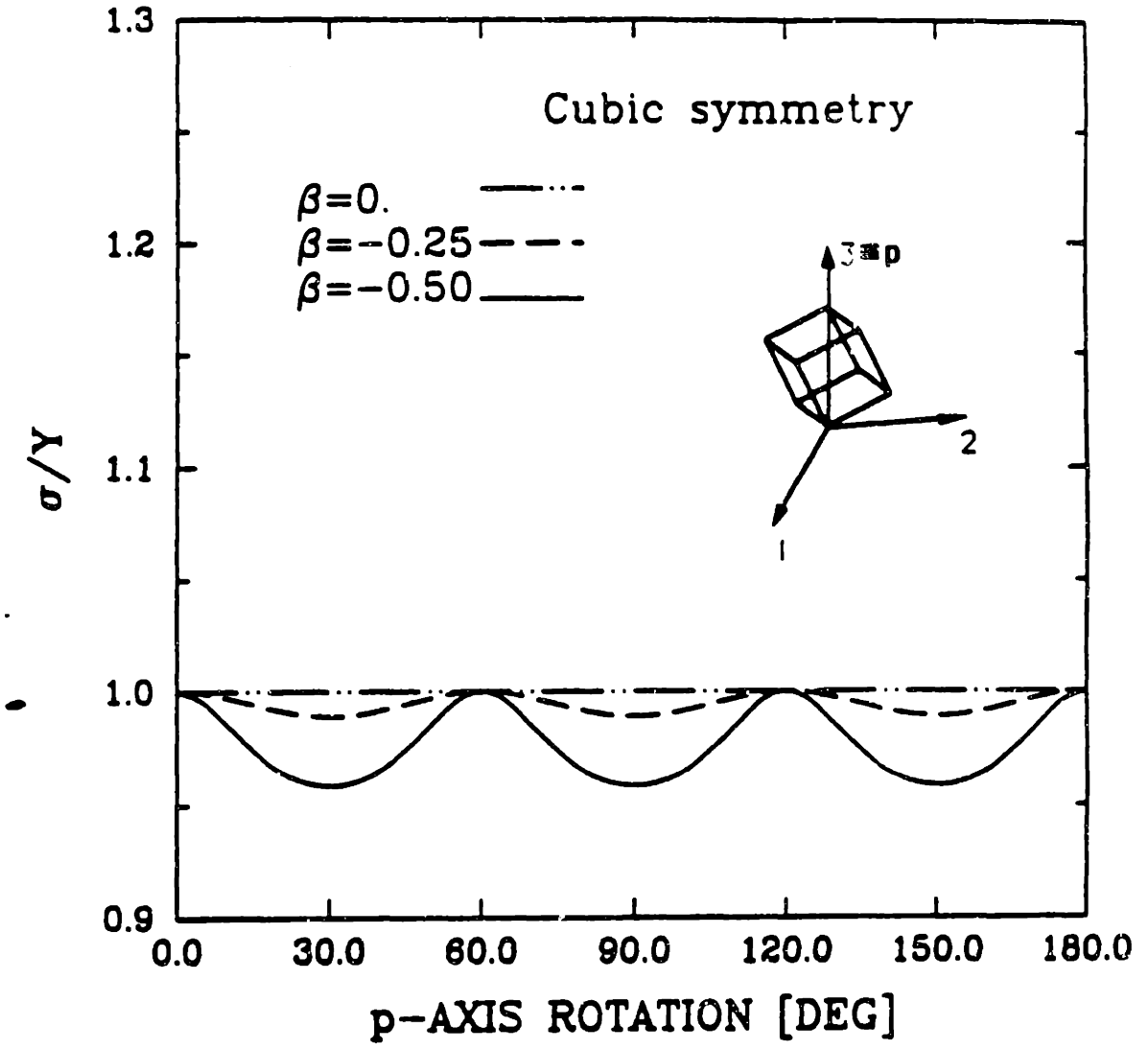


Fig. 2.10: Distribution of the uniaxial yield stress in different angles about the p-axis for a material with cubic symmetry. The parameter β indicates degree of anisotropy.

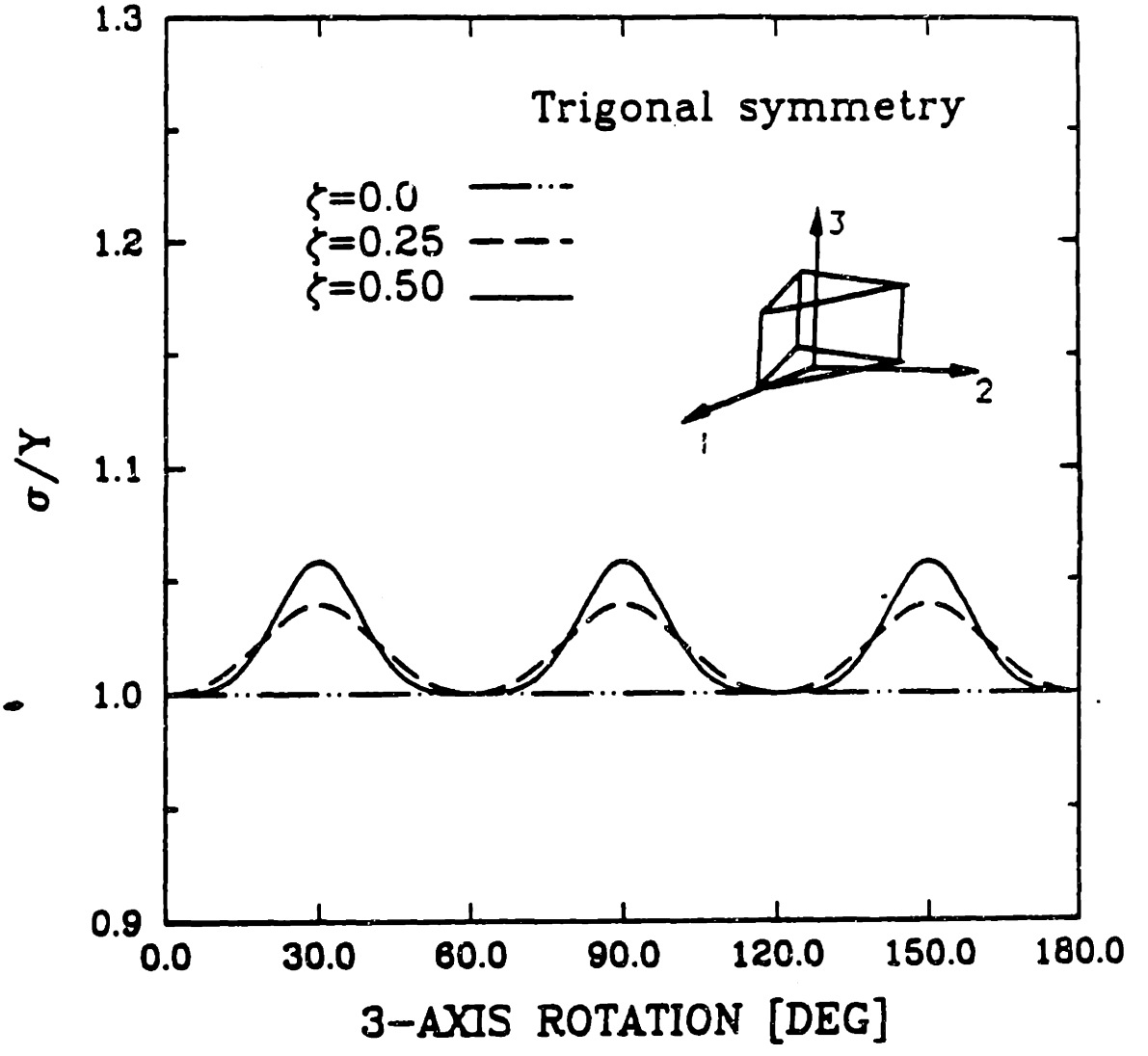


Fig. 2.11: Distribution of the uniaxial yield stress in different angles about the 3-axis for a material with trigonal symmetry. The parameter ζ indicates degree of anisotropy

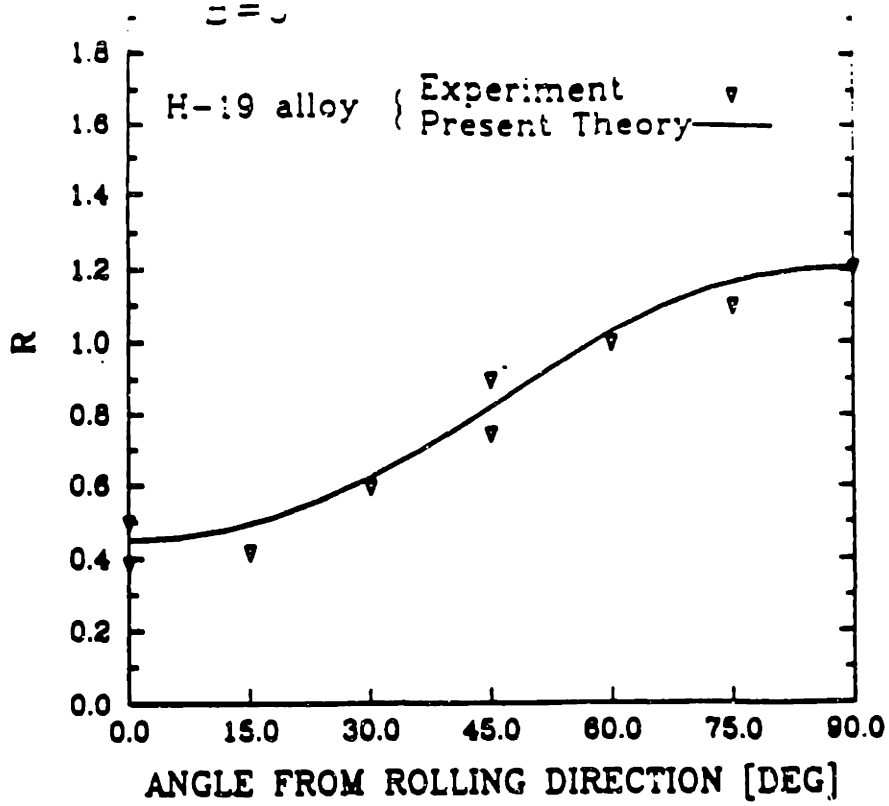
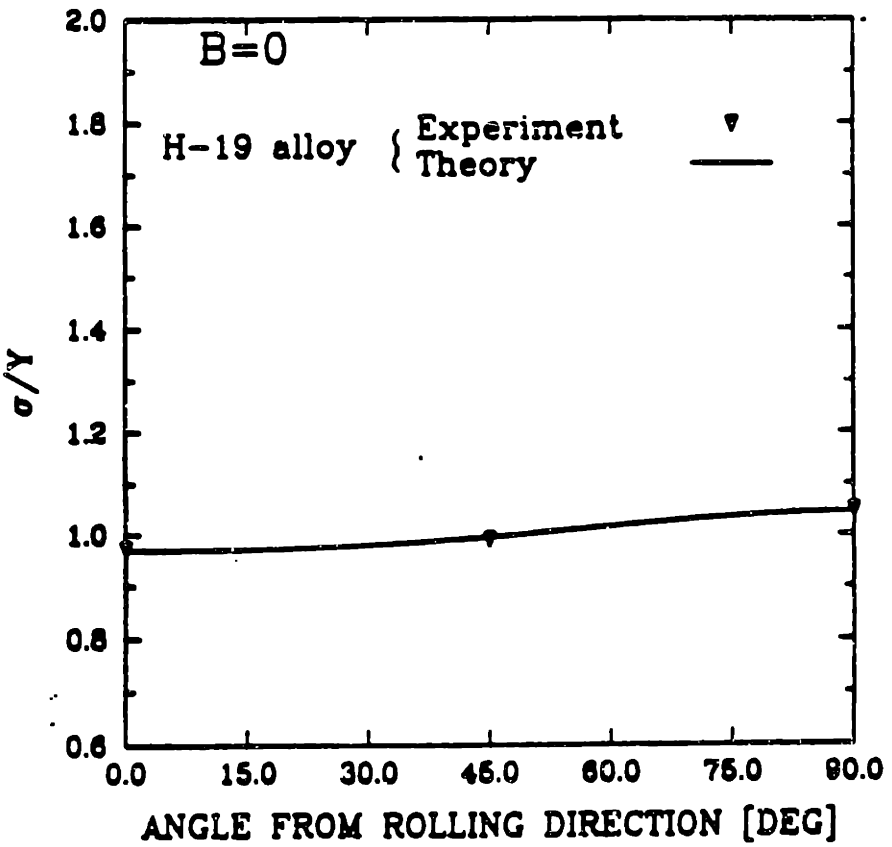


Fig. 2.11 :(a) The distribution of the R-ratio with respect to the angle with the rolling direction for the H-19 can stock alloy. Experimental results were obtained by MacEwen et. al. [1992]



(b) The distribution of the yield stress in uniaxial tension with respect to the angle with the rolling direction for the H-19 alloy. Experimental results were obtained by MacEwen et. al. [1992]

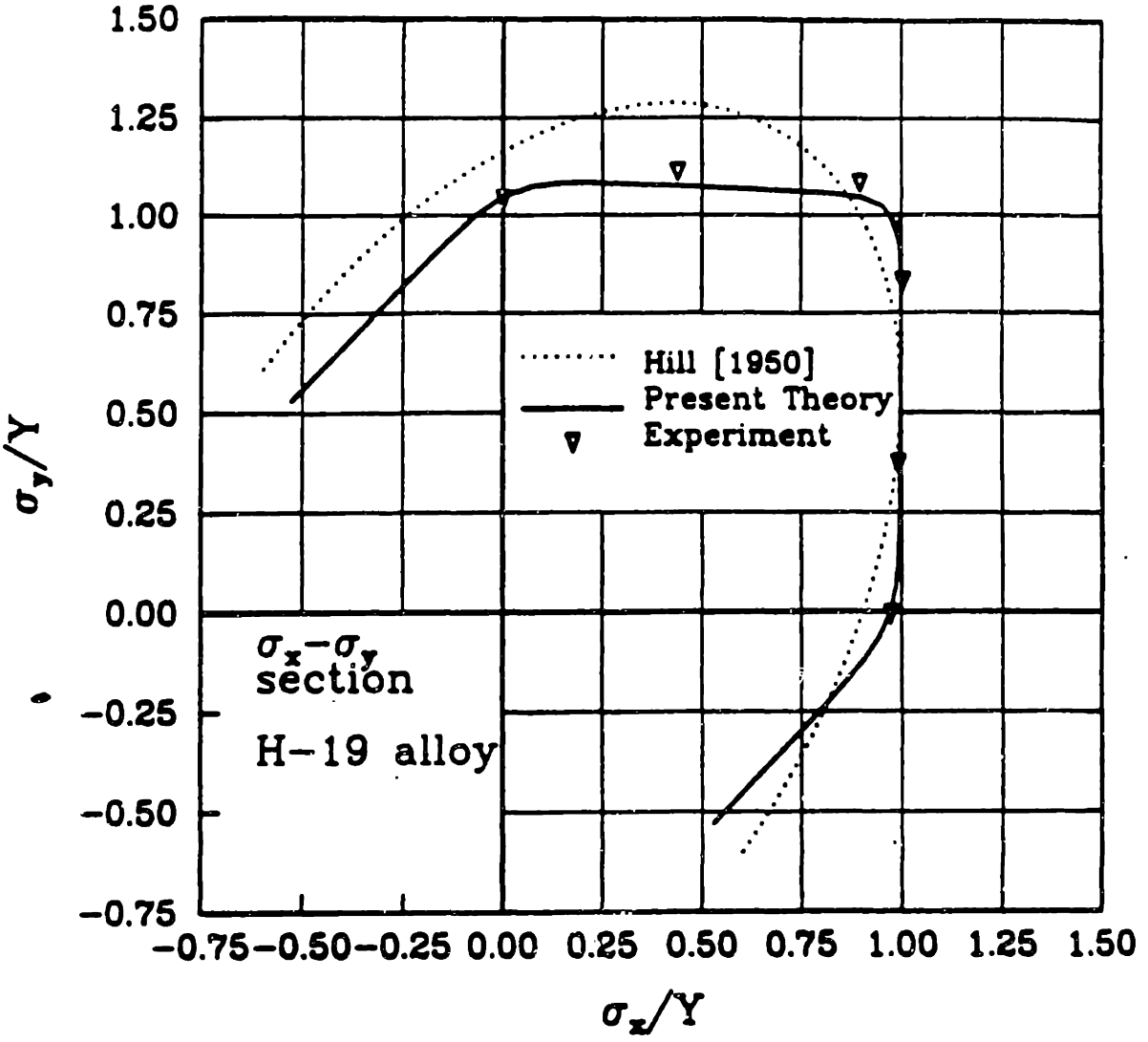


Fig. 2.12: The yield surface of the H-19 can stock aluminum alloy in the $\sigma_x - \sigma_y$ domain. Experimental results were obtained by MacEwen et. al. [1992]

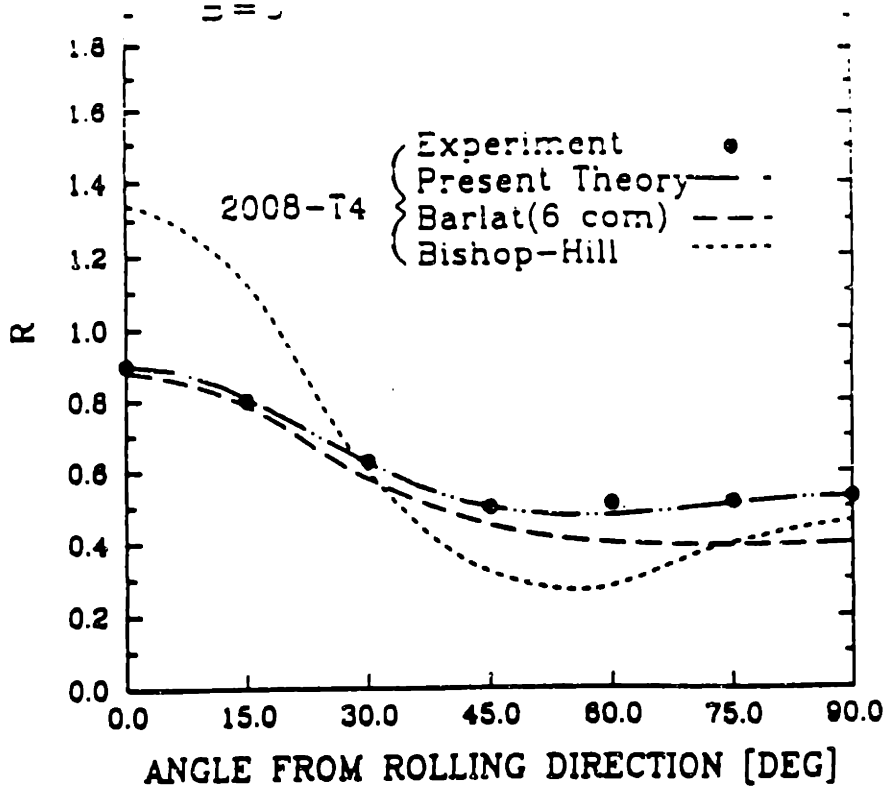
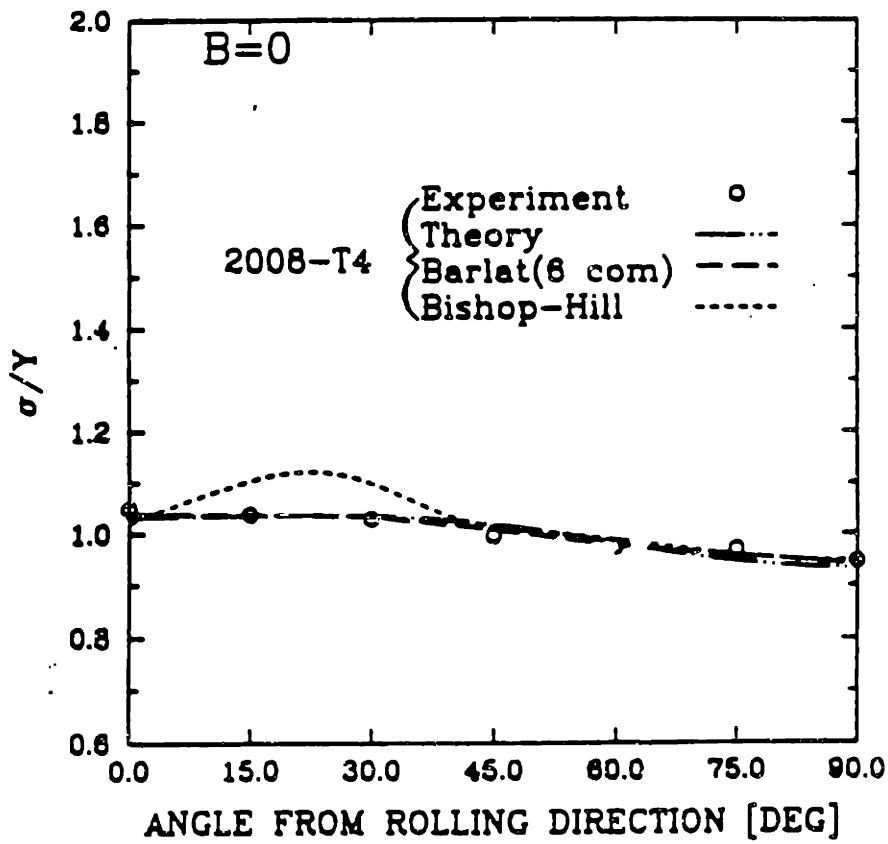


Fig. 2.13.:(a) The distribution of the R-ratio with respect to the angle with the rolling direction for the 2008-T4 aluminum alloy. Experimental results were obtained by Lege et. al. [1989] . The Bishop-Hill distribution was obtained by Lege et al. [1989]



(b) The distribution of the yield stress in uniaxial tension with respect to the angle with the rolling direction for the 2008-T4 aluminum alloy. The experimental results and the Bishop-Hill distribution were obtained by Lege et. al. [1989]

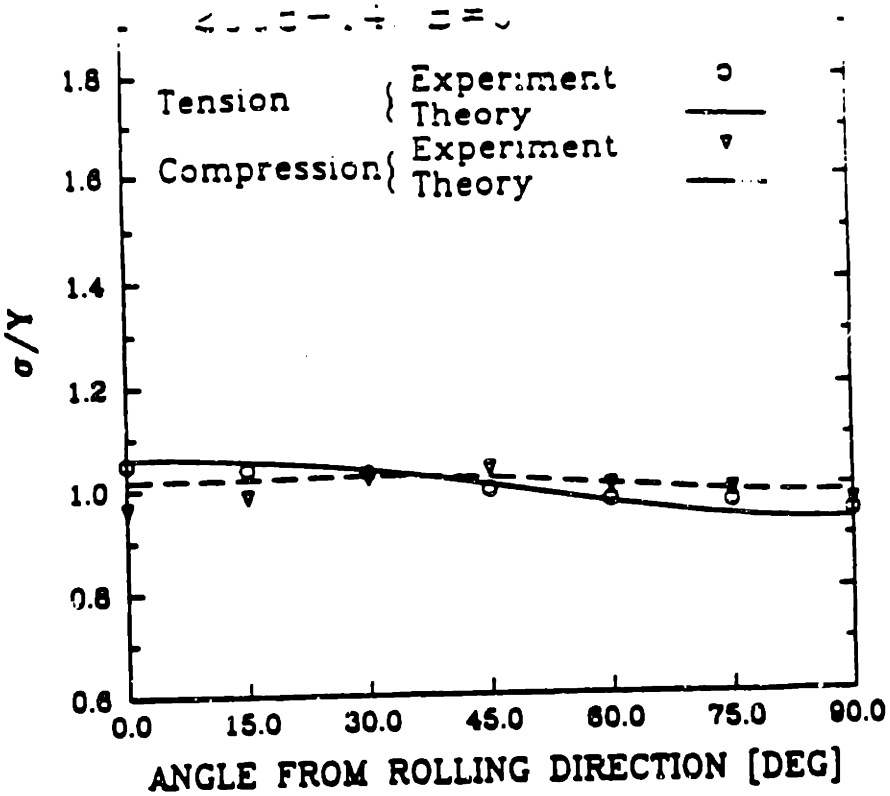
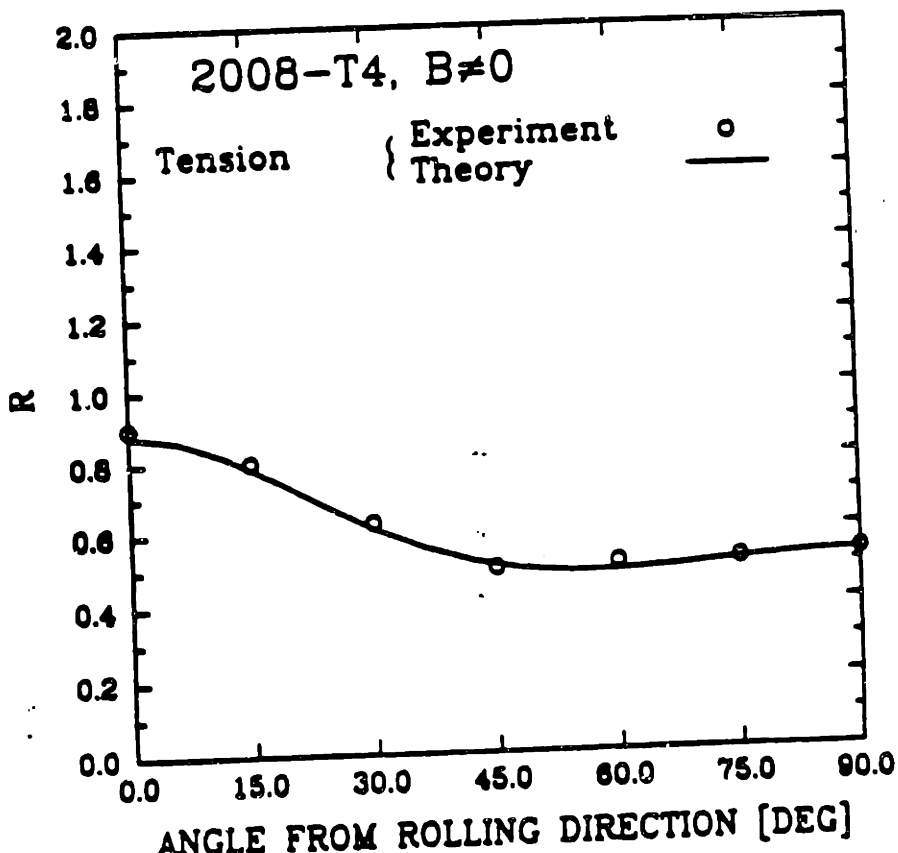


Fig. 2.14:(a) The distribution of the tensile and compressive yield stress with respect to the angle with the rolling direction for the 2008-T4 aluminum alloy. Experimental results were obtained by Lege et. al. [1989].



(b) Theoretical and experimental distribution of the R-ratio for the 2008-T4 alloy. The theoretical distribution was obtained with $B \neq 0$. Experimental results by Lege et al. [1989]

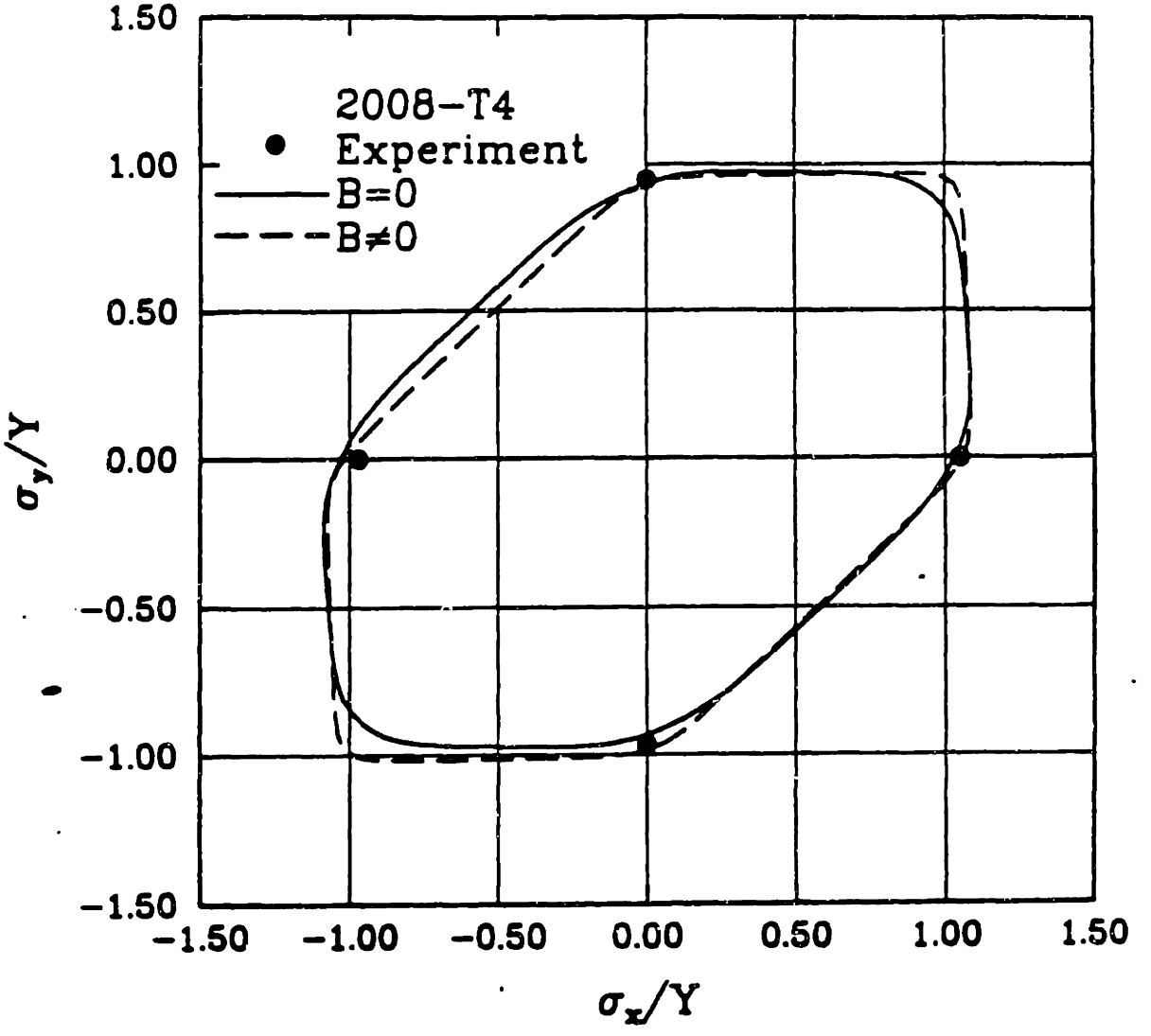


Fig. 2.15 : The projection of the yield surface in the $\sigma_x - \sigma_y$ domain for the 2008-T4 material. Experimental results were obtained by Lege et. al. [1989].

Material symmetry	Rotations which leave **L** invariant	Number of independent elements of **L**
Triclinic	none rotation	15
Monoclinic	R_2^x	8
Orthotropic	R_1^x, R_2^x	6
Trigonal	$R_3^{\frac{2\pi}{3}}, R_1^x$	4
Tetragonal	$R_3^{\frac{\pi}{2}}, R_1^x$	4
Transversely isotropic	all R_3^ϕ, R_1^x	3
Cubic	$R_1^{\frac{\pi}{2}}, R_2^{\frac{\pi}{2}}, R_3^{\frac{\pi}{2}}$	2
Isotropic	all rotations	1

Table 1.1: Material levels of symmetry and respective number of indepentent elements of **L**.

Method I

Slip system	k	c
FCC (restricted glide)	15	0.835
BCC (pencil glide)	15	0.948

Method II

Slip system	k	c
FCC (restricted glide)	6	0.3
BCC (pencil glide)	3	0.17

Table 1.2: Values of k and c obtained with Method I, and Method II.

Material	k	c	C	α_1	α_2	γ_3	$\mathbf{B}$
H-19 can stock	15	0.36	0.644	0.931	0.989	0.728	0
2008-T4 $\mathbf{B} = \mathbf{0}$	15	0.81	0.559	1.103	1.120	0.730	0
2008-T4 $\mathbf{B} \neq \mathbf{0}$	15	0.3	0.575	1.083	1.057	0.762	$B_{11} = 0.012Y$ $B_{22} = -0.018Y$ $B_{33} = 0.006Y$

Table 1.3: Material constants for the H-19 can stock alloy and for the 2008-T4 alloy.

3 CHAPTER 3: IMPLEMENTATION OF THE MATERIAL MODEL INTO FINITE ELEMENT ANALYSIS

The numerical implementation of the proposed material model into a finite element analysis was effected by developing a user material subroutine (VUMAT) for a commercial finite element analysis (ABAQUS 5.2). The constitutive hypotheses implemented in the material subroutine are presented in this section.

3.1 Inversion of the flow rule

The flow rule obtained from the proposed yield function is mathematically represented by Eqn. (2.16). We now consider a strain-rate independent material for which:

$$W = f(Y) \tag{3.1}$$

If also we consider elastic-plastic behavior where C is the stiffness matrix of the material then we assume the following constitutive response:

$$\dot{\boldsymbol{\sigma}}^{o} = \boldsymbol{C}(\boldsymbol{D} - \boldsymbol{D}^{p}) \tag{3.2}$$

where, $\dot{\boldsymbol{\sigma}}^{o}$ is an objective stress rate of the Cauchy stress, $\boldsymbol{D}$ is the symmetric part of the velocity gradient tensor of the deformed material and $\boldsymbol{D}^{p}$ is the plastic part of $\boldsymbol{D}$. While $\boldsymbol{D}$ and $\boldsymbol{D}_{p}$ act on the current configuration, all tensors in Eqn. (3.2) are expressed in the original coordinate system. We now consider a class of objective stress rates generated by the skew-symmetric matrix $\boldsymbol{\omega}$:

$$\dot{\boldsymbol{\sigma}}^{o} = \dot{\boldsymbol{\sigma}} + \boldsymbol{\sigma}\dot{\boldsymbol{\omega}} - \boldsymbol{\omega}\boldsymbol{\sigma} \tag{3.3}$$

If

$$\boldsymbol{\omega} = \dot{\boldsymbol{R}}\boldsymbol{R}^{T} \tag{3.4}$$

where,

$$\boldsymbol{R} = \boldsymbol{V}^{-1}\boldsymbol{F} = \boldsymbol{F}\boldsymbol{U}^{-1} \tag{3.5}$$

and $\boldsymbol{F}, \boldsymbol{V}, \boldsymbol{U}$ are the deformation gradient tensor, the left stretching tensor and the right stretching tensor respectively, then $\dot{\boldsymbol{\sigma}}^o$ is the Green-Naghdi rate, see also Green and Naghdi [1965]. If we now consider a rotation of the material point by $\boldsymbol{R}^T$, (or equivalently a rotation of the coordinate system by $\boldsymbol{R}$) then $\boldsymbol{\sigma}$ will be transformed to $\boldsymbol{\sigma}_R$ as follows:

$$\boldsymbol{\sigma}_R = \boldsymbol{R}^T\boldsymbol{\sigma}\boldsymbol{R} \tag{3.6}$$

Also, from Eqns. (3.3) and (3.6) we will have, see also Hughes [1983]

$$\dot{\boldsymbol{\sigma}}_R = \boldsymbol{R}^T\dot{\boldsymbol{\sigma}}^o\boldsymbol{R} \tag{3.7}$$

Therefore, by combining Eqns. (3.2) and (3.7) we have:

$$\dot{\boldsymbol{\sigma}}_R = \boldsymbol{C}_R(\boldsymbol{D}_R - \boldsymbol{D}_R^p) \tag{3.8}$$

Thus, Eqn. (3.8) provides a simple expression for the stress rate, where the actual Cauchy stress $\boldsymbol{\sigma}_R$, the stress rate $\dot{\boldsymbol{\sigma}}_R$ and the strain rates $\boldsymbol{D}_R, \boldsymbol{D}_R^p$ are expressed in the "corotational coordinate system", i.e. the coordinate system obtained if we rotate the original coordinate system by $\boldsymbol{R}$. We note that this is the coordinate system at which the stress tensor and the strain increment tensor are expressed in the VUMAT interface of ABAQUS Explicit [1993]. Therefore, by using the Green-Naghdi rate we can use Eqn. (3.8) for the constitutive response of the material. Equation (3.8) was used in the present implementation to the VUMAT subroutine.

If a different stress rate were to be considered (i.e. $\boldsymbol{\omega}$ not necessarily equal to $\dot{\boldsymbol{R}}\boldsymbol{R}^T$), then Eqn. (3.8) will be modified as follows:

$$\dot{\boldsymbol{\sigma}}_R = \boldsymbol{C}_R(\boldsymbol{D}_R - \boldsymbol{D}_R^p) - \boldsymbol{\sigma}_R\boldsymbol{q}_R + \boldsymbol{q}_R\boldsymbol{\sigma}_R \tag{3.9}$$

where,

$$\boldsymbol{q}_R = \boldsymbol{R}^T(\boldsymbol{\omega} - \dot{\boldsymbol{R}}\boldsymbol{R}^T)\boldsymbol{R} \tag{3.10}$$

For example in the case of the Jaummann stress rate it will be:

$$\boldsymbol{q}_R = \boldsymbol{R}^T(\boldsymbol{w} - \dot{\boldsymbol{R}}\boldsymbol{R}^T)\boldsymbol{R} \tag{3.11}$$

where $\boldsymbol{w}$ is the antisymmetric part of the velocity gradient tensor.

We note here that in the case of a material with an isotropic stiffness tensor, the stiffness tensor $\boldsymbol{C}$ is rotation independent and therefore $\boldsymbol{C}_R = \boldsymbol{C}$.

We would like now to invert Eqn. (2.16) in order to obtain an expression for the stress rate as a function of the the strain rate $\boldsymbol{D}$ and the current state of the material, defined by the Cauchy stress $\boldsymbol{\sigma}$ and the equivalent yield stress Y. In order to invert Eqn. (2.16) we operate in both terms of Eqn. (2.16) by the objective stress rate tensor $\dot{\boldsymbol{\sigma}}$:

$$\boldsymbol{D}_R^p \cdot \dot{\boldsymbol{\sigma}}_R = \frac{\dot{W}}{4kY^{2k}} \frac{\partial \Phi}{\partial \boldsymbol{\sigma}_R} \cdot \dot{\boldsymbol{\sigma}}_R = \frac{\dot{W}}{4kY^{2k}} 2kY^{2k-1}\dot{Y} \tag{3.12}$$

Also,

$$\boldsymbol{D}_R^p \cdot \dot{\boldsymbol{\sigma}}_R = \frac{\dot{W}}{4kY^{2k}} \frac{\partial \Phi}{\partial \boldsymbol{\sigma}_R} \cdot \boldsymbol{C}_R(\boldsymbol{D}_R - \boldsymbol{D}_R^p) \tag{3.13}$$

The purpose of inverting the flow rule is to obtain an expression for the stress rate given the value of strain rate $\boldsymbol{D}_R$ imposed on the material, see also Fig. 3.1. In order to achieve that we combine Eqns. (3.12) and (3.13) and we obtain the yield stress rate as:

$$\dot{Y} = \frac{\frac{\partial \Phi}{\partial \boldsymbol{\sigma}_R} \cdot \boldsymbol{C}_R \boldsymbol{D}_R}{2kY^{2k-1} + \zeta(Y) \frac{\partial \Phi}{\partial \boldsymbol{\sigma}_R} \cdot \boldsymbol{C}_R \frac{\partial \Phi}{\partial \boldsymbol{\sigma}_R}} \tag{3.14}$$

where

$$\zeta(Y) = \frac{\dot{W}}{4kY^{2k}\dot{Y}} \tag{3.15}$$

The function $\zeta(Y)$ can be numerically calculated once the functional depedence of W on Y, see Eqn. (3.1), is known.

Equation (3.8) can be used now in the calculation of the stress rate, where $\boldsymbol{D}_R^p$ is calculated by Eqn. (2.16) once $\dot{W} = \zeta(Y)4kY^{2k}\dot{Y}$ is known. Therefore, the explicit expression for $\dot{\boldsymbol{\sigma}}_R$ is:

$$\dot{\boldsymbol{\sigma}}_R = \boldsymbol{C}_R(\boldsymbol{D}_R - \frac{\zeta(Y)\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R}\cdot\boldsymbol{C}_R\boldsymbol{D}_R}{2kY^{2k-1}+\zeta(Y)\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R}\cdot\boldsymbol{C}_R\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R}}\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R}) \tag{3.16}$$

The constitutive Jacobian $\frac{\partial\dot{\boldsymbol{\sigma}}_R}{\partial\boldsymbol{D}_R}$ can be now calculated from Eqn. (3.16):

$$\frac{\partial\dot{\boldsymbol{\sigma}}_R}{\partial\boldsymbol{D}_R} = \boldsymbol{C}_R - \frac{\zeta(Y)(\boldsymbol{C}_R\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R})\otimes(\boldsymbol{C}_R\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R})}{2kY^{2k-1}+\zeta(Y)\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R}\cdot\boldsymbol{C}_R\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R}} \tag{3.17}$$

For the case of plane stress we have an additional constraint in the stress-rate equations:

$$\dot{\sigma}_{R_{33}} = 0 \tag{3.18}$$

whereas $D_{R_{33}}$ will have to be calculated in order to yield the desired constraint. In order to calculate $D_{R_{33}}$ we will rewrite Eqn. (3.16) as follows:

$$\dot{\boldsymbol{\sigma}}_R = \boldsymbol{C}_R(\boldsymbol{D}_R - (A + D_{R_{33}}B)\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R}) \tag{3.19}$$

where:

$$\begin{aligned} A \;=\; & \frac{\zeta(Y)}{2kY^{2k-1}+\zeta(Y)\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R}\cdot\boldsymbol{C}_R\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R}}((C_{R_{1111}}D_{R_{11}} + C_{R_{1122}}D_{R_{22}})\frac{\partial\Phi}{\partial\sigma_{R_{11}}} \\ & +(C_{R_{2222}}D_{R_{22}} + C_{R_{1122}}D_{R_{11}})\frac{\partial\Phi}{\partial\sigma_{R_{22}}} \\ & +(C_{R_{1133}}D_{R_{11}} + C_{R_{2233}}D_{R_{22}})\frac{\partial\Phi}{\partial\sigma_{R_{33}}} \\ & +2C_{R_{1212}}D_{R_{12}}\frac{\partial\Phi}{\partial\sigma_{R_{12}}} + 2C_{R_{2323}}D_{R_{23}}\frac{\partial\Phi}{\partial\sigma_{R_{23}}} + 2C_{R_{1313}}D_{R_{13}}\frac{\partial\Phi}{\partial\sigma_{R_{13}}}) \end{aligned} \tag{3.20}$$

and

$$B \;=\; \frac{\zeta(Y)}{2kY^{2k-1}+\zeta(Y)\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R}\cdot\boldsymbol{C}\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R}}(C_{R_{1133}}\frac{\partial\Phi}{\partial\sigma_{R_{11}}} \tag{3.21}$$

$$+C_{R_{2233}}\frac{\partial\Phi}{\partial\sigma_{R_{22}}}$$
$$+C_{R_{3333}}\frac{\partial\Phi}{\partial\sigma_{R_{33}}})$$

If we replace $\dot{\sigma}_{R_{33}}$ in Eqn. (3.18) with the algebraic expression obtained for $\dot{\sigma}_{R_{33}}$ from Eqn. (3.19) then we obtain a first order equation where the only unknown is $D_{R_{33}}$. The solution of this equation is:

$$D_{R_{33}} = \frac{A(C_{R_{1133}}\frac{\partial\Phi}{\partial\sigma_{R_{11}}} + C_{R_{2233}}\frac{\partial\Phi}{\partial\sigma_{R_{22}}} + C_{R_{3333}}\frac{\partial\Phi}{\partial\sigma_{R_{33}}}) - C_{R_{1133}}D_{R_{11}} - C_{R_{2233}}D_{R_{22}}}{C_{R_{3333}} - B(C_{R_{1133}}\frac{\partial\Phi}{\partial\sigma_{R_{11}}} - C_{R_{2233}}\frac{\partial\Phi}{\partial\sigma_{R_{22}}} - C_{R_{3333}}\frac{\partial\Phi}{\partial\sigma_{R_{33}}})} \quad (3.22)$$

3.2 Numerical implementation into ABAQUS Explicit

The stress rate-equations, as developed in section 3.1 were implemented into ABAQUS explicit with the aid of a user material (VUMAT). The steps of the integration algorithm are outlined below:

Step 1: First calculate a trial elastic stress:

$$\boldsymbol{\sigma}_{R,tr}^{t+\Delta t} = \boldsymbol{\sigma}_R^t + \boldsymbol{C}_R\boldsymbol{D}_R\Delta t \quad (3.23)$$

If plane stress theory is used then by assuming elastic response on that step we have:

$$D_{R_{33}} = \frac{-C_{R_{1133}}D_{R_{11}} - C_{R_{2233}}D_{R_{22}}}{C_{R_{3333}}} = D_{R_{33},tr} \quad (3.24)$$

Step 2: Calculate the equivalent yield stress Y^* according to Eqn. (2.6).

Step 3: We now introduce a flag variable ω for which:

$$\omega = 0 \quad \text{if} \quad Y^* \leq Y \quad (3.25)$$

$$\omega = 1 \quad \text{if} \quad Y^* > Y \quad (3.26)$$

Step 4: Calulate $\frac{\partial\Phi}{\partial\boldsymbol{\sigma}_R}$ at $\boldsymbol{\sigma}_R = \boldsymbol{\sigma}_{R,tr}$

Step 5: Calculate $D_{R_{33}}$ according to either of Eqns. (3.17, 3.19).

$$D_{R_{33}} = (1-\omega)D_{R_{33},tr} \quad (3.27)$$

$$+ \quad \omega \frac{A(C_{R_{1133}} \frac{\partial \Phi}{\partial \sigma_{R_{11}}} + C_{R_{2233}} \frac{\partial \Phi}{\partial \sigma_{R_{22}}} + C_{R_{3333}} \frac{\partial \Phi}{\partial \sigma_{R_{33}}}) - C_{R_{1133}} D_{R_{11}} - C_{R_{2233}} D_{R_{22}}}{C_{R_{3333}} - B(C_{R_{1133}} \frac{\partial \Phi}{\partial \sigma_{R_{11}}} - C_{R_{2233}} \frac{\partial \Phi}{\partial \sigma_{R_{22}}} - C_{R_{3333}} \frac{\partial \Phi}{\partial \sigma_{R_{33}}})}$$

Step 6: Calculate the yield stress increase from Eqn. (3.14). Then multiply the yield stress increase with ω and update yield stress as:

$$Y^{t+\Delta t} = Y^t + \omega \dot{Y} \Delta t \tag{3.28}$$

Step 7: Calculate the plastic strain increment as dictated from Eqn. 2.16.

Step 8: Update the Cauchy stress:

$$\boldsymbol{\sigma}_R^{t+\Delta t} = \boldsymbol{\sigma}_R^t + \dot{\boldsymbol{\sigma}}_R \Delta t \tag{3.29}$$

where $\dot{\boldsymbol{\sigma}}_R$ is calculated from Eqn. (3.16).

The FORTRAN code for the user material subroutine used in ABAQUS explicit [1993] in conjunction with a driver input deck can be found in Appendix III.

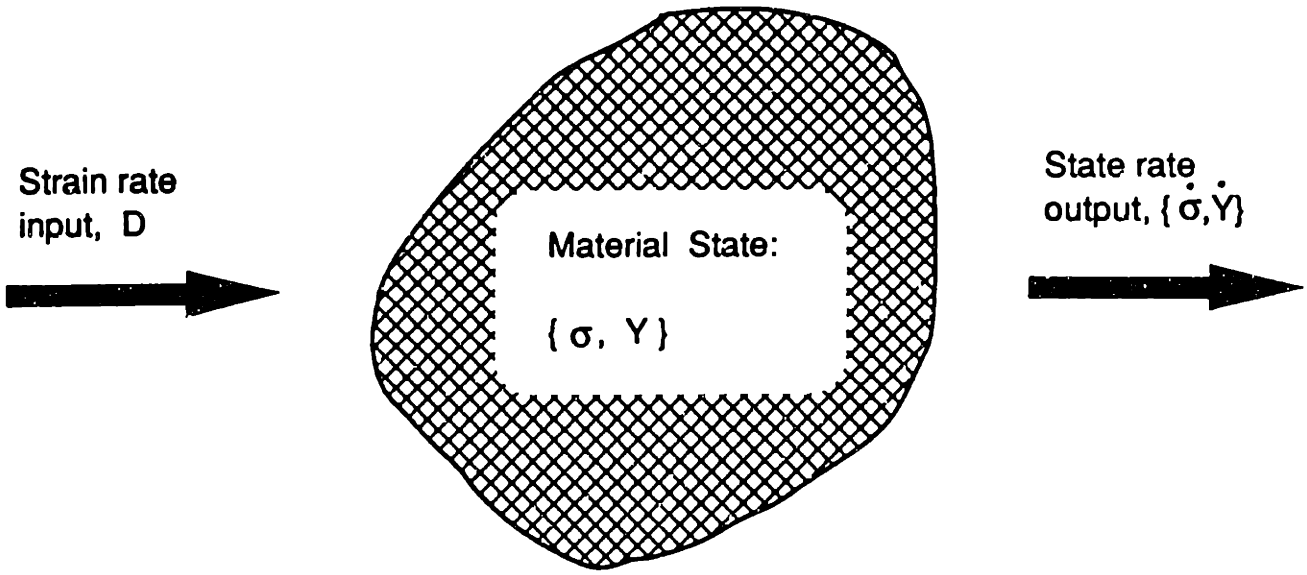


Fig. 3.1 Evolution of the material state

4 CHAPTER 4: SIMULATION OF THREE-DIMENSIONAL SHEET METAL FORMING PROCESSES USING FINITE ELEMENT ANALYSIS

In this chapter we will examine different finite element techniques in order to simulate three-dimensional sheet metal forming operations. We will present and compare different available computational techniques and we will assess their effectiveness in correctly predicting forming loads, springback, strain distributions and stress distributions when forming sheet metal parts.

4.1 Implicit model of loading and unloading

In the implicit finite element analysis used in our simulations, the static equilibrium equations are solved by using the Newton-Raphson iteration method. The Newton-Raphson iteration technique operates as follows, ABAQUS [1993]:

$$\Delta \boldsymbol{u}^{i+1} = \Delta \boldsymbol{u}^{i} + \boldsymbol{K}_t^{-1}(\boldsymbol{F}^{i} - \boldsymbol{I}^{i}) \tag{4.1}$$

where $\boldsymbol{u}$ is the nodal displacement vector, $\boldsymbol{K}_t$ is the current tangent stiffness matrix, $\boldsymbol{F}$ is the external load vector, and $\boldsymbol{I}$ is the internal force vector. The internal force vector is calculated by summing the internal force vectors calculated from every element:

$$\boldsymbol{I} = \sum_k \boldsymbol{I}_k \tag{4.2}$$

whereas the $\boldsymbol{I}_k$ is calculated as:

$$\boldsymbol{I}_k = \int \boldsymbol{B}_k^T \boldsymbol{\sigma}_k dV_k \tag{4.3}$$

where $\boldsymbol{B}_k$ is the stress-displacement matrix for the kth element, $\boldsymbol{\sigma}_k$ is the stress tensor in vector form, and V_k is the element volume.

When modeling forming processes of thin blanks, shell elements can be used to model the formed material, see also Honecker and Matiasson [1989], Rebelo et

al. [1992], and Taylor et al. [1993]. Shell elements can capture the combination of stretching and bending as opposed to membrane elements which do not possess bending modeling capabilities. On the other hand, shell elements cannot describe the development of normal stresses through the thickness of the blank, as opposed to continuum elements which offer this capability. However, in sheet metal forming processes, the values of the normal stresses developed in the blank are usually one or two orders of magnitude smaller than the in-plane stresses. Therefore, neglecting the normal stresses when considering the material's constitutive response can be considered as a reasonable assumption. The thickness change of the shell elements is computed as a Poisson effect from the in-plane stretching which, when plastic, assumes plastic incompressibility.

Due to the geometry of the tooling (see Fig. 4.1) and the width of the blank, straining conditions almost identical to plane strain were developed in the material. However, we chose to model the forming operation as a three dimensional process in order to (1) lay the groundwork for the modeling of processes with more complicated tooling geometries, and (2) obtain an initial evaluation of the effectiveness and computational efficiency of the three-dimensional modeling techniques. In this perspective, the finite element model of the blank when using the tooling of Fig. 4.1 was effected by using 40×4 4-node shell elements of reduced integration, see also Fig. 4.1. Due to the symmetry of the tooling only a quarter of the geometry and the blank was modeled. The Kirchhoff constraints for the shell elements were numerically enforced by a transverse shear stiffness of the shell section, ABAQUS [1993]. Five integration points were provided through the thickness of the shell in order to adequately account for both the stretching and the bending incurred by the blank, where Simpson integration was used to calculate the section forces and the section moments. Contact interaction between the blank and the rigid tools was modeled by using a "softened"

description of the rigid tools in which normal pressure develops on the blank surface nodes as a steep function of the penetration to the rigid surface. The frictional stress is developed as a function of the tangential slip, where some limited elastic slip is allowed, prior to the development of irreversible plastic slip. Details on the formulation of a friction law and the associated flow rule by using such an approach can be found in Karafillis and Boyce [1992]. In the present analysis we used the Coulomb friction law with an estimated friction coefficient of 0.08 in order to model the contact between the aluminum blank and the forming tools which were made out of maple wood with a very fine surface finish, using liquid soap as a lubricant. Also, a binder force of 1000 N was applied to the binder to simulate the force applied on the blankholder by means of manual clamps. This binder force, in combination with the considered value of the friction coefficient, allowed a considerable amount of material draw-in, thereby avoiding excessive stretching of the material which could lead to failure by tearing.

The material model for the effected simulations used the yield surface developed by Karafillis and Boyce [1992], in combination with a piecewise linear description of the strain hardening behavior. The yield locus of the considered material (2008-T4 aluminum) is shown in Fig. 4.2, whereas the stress-strain curve is shown in Fig. 4.3. This stress-strain curve was obtained from the stress-strain curve at 0^o degrees with respect to the rolling direction, where the yield stress was scaled by the ratio of the average in-plane stress to the stress at 0^0 deg. and the strain was scaled by the inverse of this ratio. In Fig. 4.2 we have also plotted Hill's [1950] yield locus for the 2008-T4 aluminum alloy. Both yield loci for this material were obtained by fitting the distribution of the R-ratios in different directions to experimental values. The K-B yield surface was found to better describe the yield stress distribution and the yield surface shape for some aluminum alloys, see also Karafillis and Boyce [1993]. As also seen

in Fig. 4.2, the K-B yield function predicts different uniaxial and plane strain yield stresses in 0^o and 90^o when compared with Hill's yield function. The aluminum alloy considered here (a 2008-T4 aluminum blank) had a thickness of 1.055mm. Elastic-plastic response of the material was assumed, with an isotropic stiffness tensor. The Young's modulus was taken to be $70.0GPa$. The Poisson's ratio used in our model was 0.33.

When the implicit finite element analysis is used in combination with an elastic-plastic material model, it allows simulation of springback upon the removal of all the contact loads in the fully loaded state. Springback then occurs as the additional deformation of the processed material from its fully loaded state upon the removal of all contact loads. Springback calculations by using implicit finite element analysis and/or an elastic-plastic material model have been performed by, for example, Karafillis and Boyce [1992], Wang et al. [1993], Taylor et al. [1993].

4.2 Implicit Results vs. Experimental Results

We now consider the finite element analysis of the metal forming operation with the tooling of Fig. 4.1. A side view of the successive stages of the simulation of the stamping with the tooling of Fig. 4.1 and the use of implicit finite element analysis, is shown in Fig. 4.4. Due to the geometry of the process and the frictional interaction between the tools and the workpiece, nearly plane-strain conditions were developed in the whole of the workpiece. The material model used in the implicit simulations was elastic-plastic with Mises plasticity, where the tensile strength in uniaxial tension was modified in order to predict the same strength in plane-strain with the one predicted by using the yield function developed in Chapter 2 and shown in Fig. 4.2. In Fig. 4.4 we have depicted the successive stages of the simulation as obtained in our experiment. We can see that our finite element model can successfully capture the bending of the material, which causes the center section of the blank to lift from

the punch during the loading of the blank. At the fully loaded stage, Fig. 4.4(c), the blank is formed to the punch-matching die shape. In the final stage, all the contact forces are removed and the blank springs back to its final position, Fig. 4.4(d). Note the large deviation of the produced part shape from the die shape due to excessive springback.

In Fig. 4.5 we plot the stress distribution on the top and the bottom of the blank, as obtained from our finite element analysis. The crossovers observed in the stress distribution are due to the bending of the blank over the different curvatures of the tools. The average membrane stress across the thickness is nearly zero, indicating the presence of very little stretching. The amount of the material draw-in obtained from our simulation, was 8.6 mm, in perfect agreement with the measured experimental value, indicating that the simulation correctly captured the material restraint imposed by the binder.

The profile shapes of the produced part upon unloading as obtained from the analysis and the experiment are shown in Fig. 4.6. We can see the good agreement between the analysis and the experiment, indicating the effectiveness of our combination of finite element analysis and material modeling in accurately simulating the forming process.

4.3 Explicit Model of Loading with Implicit Model of Unloading

For this simple geometry, no particular convergence problems were experienced using the implicit finite element analysis of the stamping process depicted in Fig. 4.4. However, we have to note that when large 3-dimensional problems are considered, the implicit integration schemes may require substantially higher amounts of CPU time, see also the discussion by Rebelo et al. [1992]. One of the major disadvantages of the implicit simulation is that it is very difficult to predict the time necessary

for the completion of the analysis, as it strongly depends on the convergence rate of the equilibrium equations during the analysis. In some cases implicit finite element analyses may develop convergence problems associated with sudden changes on the contact conditions between workpiece and tools. Although considerable progress has been achieved in modeling the contact in implicit finite element methods, see also Keum et al. [1990], Rebelo et al. [1990], Karafillis and Boyce [1992], implicit finite element analyses still require careful selection of the contact parameters, whereas there is always a possibility that convergence will not be achieved. For these reasons we have also examined the use of dynamic-explicit analysis as an alternative method which may be more robust and/or more rapid for general three-dimensional forming problems.

The use of explicit dynamic finite element analysis to model sheet metal forming processes was suggested by Honecker and Matiasson [1989]. We used the commercial explicit finite element code ABAQUS Explicit [1993] in our explicit finite element calculations. In this code, the displacements are updated at every time increment by the equation:

$$\boldsymbol{u}^{i+1} = \boldsymbol{u}^{i} + \Delta t^{i+1} \dot{\boldsymbol{u}}^{i+\frac{1}{2}} \tag{4.4}$$

where $\boldsymbol{u}$ is the nodal displacement vector. The velocity vector $\dot{\boldsymbol{u}}$ is calculated by explicitly integrating the acceleration vector $\ddot{\boldsymbol{u}}$ in the time domain by using a central integration operator:

$$\dot{\boldsymbol{u}}^{i+\frac{1}{2}} = \dot{\boldsymbol{u}}^{i-\frac{1}{2}} + \frac{\Delta t^{i+1} + \Delta t^{i}}{2} \ddot{\boldsymbol{u}}^{i} \tag{4.5}$$

The acceleration at the begining of the increment is:

$$\ddot{\boldsymbol{u}}^{i} = \boldsymbol{M}^{-1}(\boldsymbol{F}^{i} - \boldsymbol{I}^{i}) \tag{4.6}$$

where $\boldsymbol{I}^{i}$ are the internal forces calculated in the same way as in the implicit static analysis, see Eqn (4.3), and $\boldsymbol{F}^{i}$ are the external applied forces.

In the case of contact between a deforming workpiece and a rigid body, additional accelerations are applied to the material node in order to constrain the node to remain on the surface of the rigid body and to not allow co-penetration.

Explicit finite element methods exhibit conditional convergence where the stability limit in the time increment is proportional to the ratio of the characteristic element length divided by the speed of sound in the material. Therefore, a dynamic explicit analysis requires a very large number of time increments when modeling phenomena of long time duration. However, the formulation of a diagonal mass matrix for updating displacements and the explicit enforcement of the contact conditions render the analysis very fast in comparison to the implicit algorithm which requires updating and "inverting" the stiffness matrix to update the displacements, Eqn. (4.1). In the case of large three-dimensional forming operations the explicit dynamic analysis may render the simulation substantially faster as compared to the implicit, see also Mattiasson et al. [1992] and Rebelo et al. [1992].

Dynamic explicit finite element analysis of the process of Fig. 4.1 were conducted. The same material model as used in the implicit simulation was used in the explicit simulation. Also, the same number and type of shell elements were used. The tool surfaces were described by using 4-node patches. The punch displacement was described by using the following acceleration history:

$$\gamma(t) = \frac{2\pi u_p}{T_l^2}\sin(\frac{2\pi t}{T_l}) \tag{4.7}$$

where $\gamma(t)$ is the acceleration of the punch at time t, u_p is the total displacement of the punch, and T_l represents the duration of the punch loading motion. Then the obtained displacement $u(t)$ is obtained as:

$$u(t) = \frac{u_p}{T_l}(t - \frac{T_l}{2\pi}\sin(\frac{2\pi t}{T_l})) \tag{4.8}$$

This displacement history is plotted in Fig. 4.7. By using this displacement history we smoothed transitional phenomena due to the acceleration and deceleration of the

punch and thereby reduced the dynamic noise of the solution. The simulation of the forming process was attempted with different durations of the punch motion. Since the duration of the analysis is proportional to the duration of the simulated phenomenon T_l, we would like to decrease T_l as much as possible. In our case the real duration of the stamping operation was of the order of 1 min. However, it would not be feasible to use this value for T_l, as this would require an enormous amount of computational time. Instead we performed the simulation with T_l ranging from $1.0 \times 10^{-4} secs$ to $1.0 \times 10^{-1} secs$. This was done in order to study the effect of T_l on the results of the analysis and in order to establish a criterion for the selection of T_l.

In Fig. 4.8 we plot the maximum value of the ratio r of the kinetic energy of the blank to the recoverable strain energy of the blank in the time history of the analysis, as a function of T_l. Large values of this ratio indicate that the dynamic effects present in the analysis are large and have strong influence on the results, whereas very small values of this energy ratio indicate that dynamic effects are negligible. When attempting to model processes which are virtually static by using dynamic explicit finite element analysis, we must ensure that the value of r is low enough in order to recover the static analysis as a limit of the dynamic analysis.

In Fig. 4.9 the stress disributions along the top fiber of the blank in the fully loaded stage obtained from the different explicit simulations and from the implicit static simulation are plotted. Also the stress distribution for a very fast analysis, (r=6.8) is shown in Fig. 4.10. We can see that when the value of r is lower than 0.07, the response of the material is very similar to the response obtained by the implicit finite element simulation. On the other hand, when the value of r is very large the stress response of the material obtained from the explicit analysis is very different from the results obtained from the static implicit analysis, compare Figs. 4.9 and 4.10. This is clearly due to the large value of the kinetic energy of the blank which

corresponds to the development of strong dynamic phenomena during the analysis thus altering the displacement and strain histories incurred by the blank.

The above observations motivate for the establishment of r as an indicator of the proximity between the implicit static and the explicit dynamic simulation. In the present work we will consider explicit simulations with a value of r lower than 0.05 to be acceptable for the modeling of a static process.

We now consider the simulation of springback in the case of the simulation of loading of the punch by dynamic explicit analysis. We consider springback here as primarily an elastic process (although secondary yielding is a possibility).

Let us consider the displacement vector of the blank $\boldsymbol{u}_l$ at the fully loaded position, and the external force vector $\boldsymbol{F}_l$ applied to the blank by the tools. If we assume static equilibrium at the fully loaded state, (i.e. inertial forces are negligibly small) then the external force vector $\boldsymbol{F}_l$ is balanced by the internal force vector $\boldsymbol{I}_l$. During unloading (removal of the punch and the binder) the external force vector is set to zero following a certain history, causing the development of acceleration in the material nodes in the explicit dynamic analysis, see also Eqn. (4.6). Due to the lack of plasticity and/or frictional dissipation during unloading, the structure will oscillate about the new equilibrium position defined by the displacement vector $\boldsymbol{u}_{ul}$. Therefore, the shape of the blank upon unloading cannot be obtained unless the new equilibrium position is established. In order to establish the equilibrium position upon unloading we developed a method which is based on the combination of the explicit dynamic and the implicit static analysis. In this method we take the internal forces in the fully loaded state from the explicit dynamic analysis and we use them in an implicit static simulation of the unloading procedure. Specifically, the following steps are taken:

- *Step 1:* The blank is deformed to the fully loaded state using dynamic explicit analysis. The displacements and rotations of every node of the finite element mesh

of the blank at the fully loaded state are recorded. The section forces and moments of the shell element at the fully loaded state are also recorded. Section forces and moments are recorded at the local coordinates of the shell elements.

• *Step 2:* A new finite element analysis is performed. This analysis is an implicit static simulation. The finite element mesh of this blank is the same as the initial finite element mesh of the explicit finite element analysis. In the first step of this analysis, all the nodes of the mesh are assigned the displacements and rotations obtained from the explicit finite element analysis at the fully loaded state. Thus, at the end of this step, all the material integration points will have the same strains and rotations as in the explicit finite element analysis. However, all stresses are assumed to remain zero during this step. In other words, this step has simply served to create a mesh for the next step of the new analysis.

• *Step 3:* All nodes of the new mesh from Step 2 are now constrained to have zero displacement and rotation increments. The internal section forces and section moments are assigned the values obtained from the explicit analysis at the fully loaded state. This is effected by using a "User Material Subroutine" which provides a stress distribution by enabling the assignment of a stress state to the material integration points which will yield the desired forces and moments at every shell section in order to obtain the same internal force vector $\boldsymbol{I}_l$ as in the explicit finite element analysis. Since all the nodal degrees of freedom are constrained during this step, the consistent external forces $\boldsymbol{F}_l$ will also be obtained.

• *Step 4:* The constraints from the nodes are now removed (except for the constraints necessary to describe symmetries and the constraints necessary to suppress rigid body motions). All the consistent external nodal forces are ramped to zero at the end of the step. Only elastic response of the material is assumed and a hypoelastic material model is used. The shape of the blank at the end of this step is the shape after the

springback of the material.

The entire procedure described above, consisting of the four steps used for the explicit simulation of loading and implicit simulation of unloading in sheet forming processes has been automated for handling general cases. The software used in this process can be founf in Appendix IV.

The shape of the blank obtained after simulation of springback by using the combination of the explicit and implicit finite element analysis for the forming case of Fig. 4.1 is shown in Fig. 4.11. From Fig. 4.11 we see that there is perfect agreement with the shape obtained by using exclusively implicit static finite element analysis, when the value of r is less than 0.07. This was also expected as this value of r resulted in an almost identical stress distribution between the implicit and the explicit finite element analysis at the fully loaded state. However a small deviation is observed when the value of r is larger ($r = 1.0$). When the explicit dynamic analysis is performed with larger values of r, ($r = 6.8$) the springback response of the material cannot be captured correctly by the subsequent implicit unloading analysis, see Fig. 4.11. This is apparently due to the errors in stress distribution at the fully loaded stage, see Fig. 4.10 which is strongly affected by the dynamic phenomena developed during the explicit dynamic simulation of the higher r analyses.

In sheet metal forming processes like the one modeled here, where the material restraint is not large and therefore considerable amounts of draw-in are allowed, we can assume that most of the shape change of the material during unloading is due to the bending that occurs in the material during loading. We verified this assumption by performing the springback calculations with the explicit + implicit analysis technique, where only the section moments were incorporated in the implicit static analysis subsequent to the explicit dynamic analysis, i.e. section shear and membrane forces were not considered. This simulation resulted in a sidewall springback angle of $\theta_s =$

5.3^o as opposed to $\theta_s = 5.32^o$ when both section moments and section forces were considered.

4.4 A case study: Forming of a three-dimensional pan

In this case we will present the simulation of the forming of a three-dimensional pan with two different materials: Deep drawing quality steel and aluminum (2008-T4). The analysis was performed by using ABAQUS Explicit [1993]. Four node shell elements of reduced integration with 5 integration points through the thickness in order to capture bending phenomena were implemented. Also, the material model developed in Chapters 2 and 3 was used to model the anisotropy of the used alloys. The purposes of the discussed simulations were:

- To evaluate the feasibility of the production by stamping of the pan design.
- To assess the precision of the simulation to predict strain distributions, material draw-in forming loads and springback in a three-dimensional forming operation.

The examined pan design was developed by Walczyck [1994] and is meant to be used as a benchmark part for the evaluation of different adaptive tooling development techniques (e.g. CNC machining, laminated tooling, flexible tooling). With the aid of finite element analysis the design was effected in two stages. The part design of the first stage is depicted in Fig. 4.12. This initial design required a 25.4 mm forming height whereas a binder force of 150KN was considered in order prevent material draw-in during the forming process.The tooling set-up is depicted in Fig. 4.13. A finite element analysis of the process of forming the initial designed part of Fig. 4.12 by using the deep drawing steel as the workpiece material was performed by using ABAQUS explicit [1993]. The value of the energy ratio r was equal to 0.05 for this simulation in order to ensure that the stress and strain distributions obtained in the formed part nearly represent static equilibrium conditions, as also discussed in Section 4.3. A three-dimensional view of the deformed part at the fully loaded

stage is shown in Fig. 4.14 whereas the thickness distribution of the deformed part is shown in Fig. 4.15. By an inspection of Fig. 4.14 we see that excessive stretching of the workpiece is developed in the areas of the punch tip radii. This notion is also clearly demonstrated in Fig. 4.15 where the value of the blank thickness is reduced to values as low as 0.3mm , compared with the $0.635mm$ initial thickness of the blank. It is therefore clear that this initial part design could not be obtained in a single step forming operation with the given part geometry, forming height and forming parameters. Thus the design was altered as indicated in Fig. 4.16. In this design the the part is obtained with a 18.0mm forming height. Also, in order to further reduce the amount of stretching incurred by the workpiece, the corners of the blank were cliped, whereas a binder force of 67KN was used in the binder, whereas the friction coefficient between the workpiece and punch was reduced to 0.03, which corresponds to the use of a Teflon layer as lubricant. The thickness distribution obtained by using this design is shown in Fig. 4.17. As we can see from Fig. 4.17 the thickness reduction incurred by the blank during the forming operation is much smaller as compared to the thickness reduction obtained with the original design. The thickening developed in the blank in the corner regions of the binder can be a reason for concern, as it may lead to the development of local wrinkling during the process. However, this design was adopted as the benchmark design for the assesment of different adaptive tool techniques as it satisfied the requirements set by Walczyk [1994] in representing a combination of different deformation modes (stretching, deep-drawing, bending) in a single workpiece.

Springback calculations by using the combination of explicit dynamic analysis for loading and implicit static analysis for unloading were also performed for this design. It was found that very little springback occurs upon unloading, (maximum displacement = 0.15 mm). This is apparently due to the large amount of constraint imposed

by the binder area of the blank to the center section of the formed workpiece during unloading. In other terms, springback deformation of the blank during unloading leads to the development of tensile "circumferential like" stresses in the binder area of the workpiece, thereby preventing material deflection during unloading.

The final design was also effected experimentally by Walczyk [1994]. The punch force obtained experimentally and the punch force obtained from the simulation are depicted in Fig. 4.18, where excellent agreement between prediction and experiment is observed. In the experimental operation a grid of small circles was also printed on the initialy flat blank in order to measure the surface strains of the blank upon loading. Upon completion of the experiment, diameters of the grid circles were measued with a caliper in the two principal directions. By these measurements we were able to calculate the surface principal strains. The normal surface strain (thinning) was obtained as a Poisson effect by assuming material incompressibility. The thinning strain distribution along line OA, (see also Fig. 4.17) is shown in Fig. 4.19 as obtained by simulation and by experiment.

The same part was also formed by using 2008-T4 aluminum as the workpiece material. The forming load, obtained by experiment and simulation, is shown in Fig. 4.20. Also, the thickness strain distribution along line OA is shown in Fig. 4.21. In Fig. 4.21 the normal strain at both the top and the center section points of the deformed blank along line OA are depicted. It is clear from Fig. 4.21 that the bending effects are quite important for the process modeled here, as they cause a large amount of strain variation through the thickness of the blank. In Fig. 4.22 we plot the top view of the perimeter of the formed part as obtained by analysis and by experiment, where we see the good agreement between the experimentally obtained and the theoretically predicted material draw-in in different directions.

For the case of the aluminum part, we also modeled the process by using Hill's

quadratic yield criterion in order to also demonstrate the advantages of our proposed yield function, developed in Chapter 2. The strain distribution along line OA and the top view of the perimeter of the formed part are plotted in Figs. 4.23 and 4.24 respectively. We see that although there is reasonable agreement in the thickness strain distribution between prediction and experiment, the analysis cannot satisfactorily predict in this case the shape of the perimeter of the formed part as obtained from the plan view, see Fig. 4.24. This is probably due to the fact that Hill's yield theory overestimates the anisotropy of the material, when predicting the in plane yield stress of the material, see Fig. 4.2 and therefore does not predict correctly the material draw-in in different directions, see Fig. 4.23.

4.5 Conclusions

In this chapter, a general computational tool for three-dimensional sheet metal forming and springback calculations was developed. Two separate methods were developed for these calculations. In the first method, static implicit finite element analysis was performed to model both the loading and the unloading of the blank in the sheet metal forming operation. In the second method the simulation of the loading was performed by using an explicit dynamic finite element analysis whereas the unloading of the blank was simulated by using an implicit static analysis. Both methods were found to predict the springback in very good agreement with experimental results. However when the second method was used (explicit loading + implicit unloading), care should be taken in order to ensure that dynamic phenomena developed in the explicit dynamic simulation of loading do not affect the springback unloading calculations.

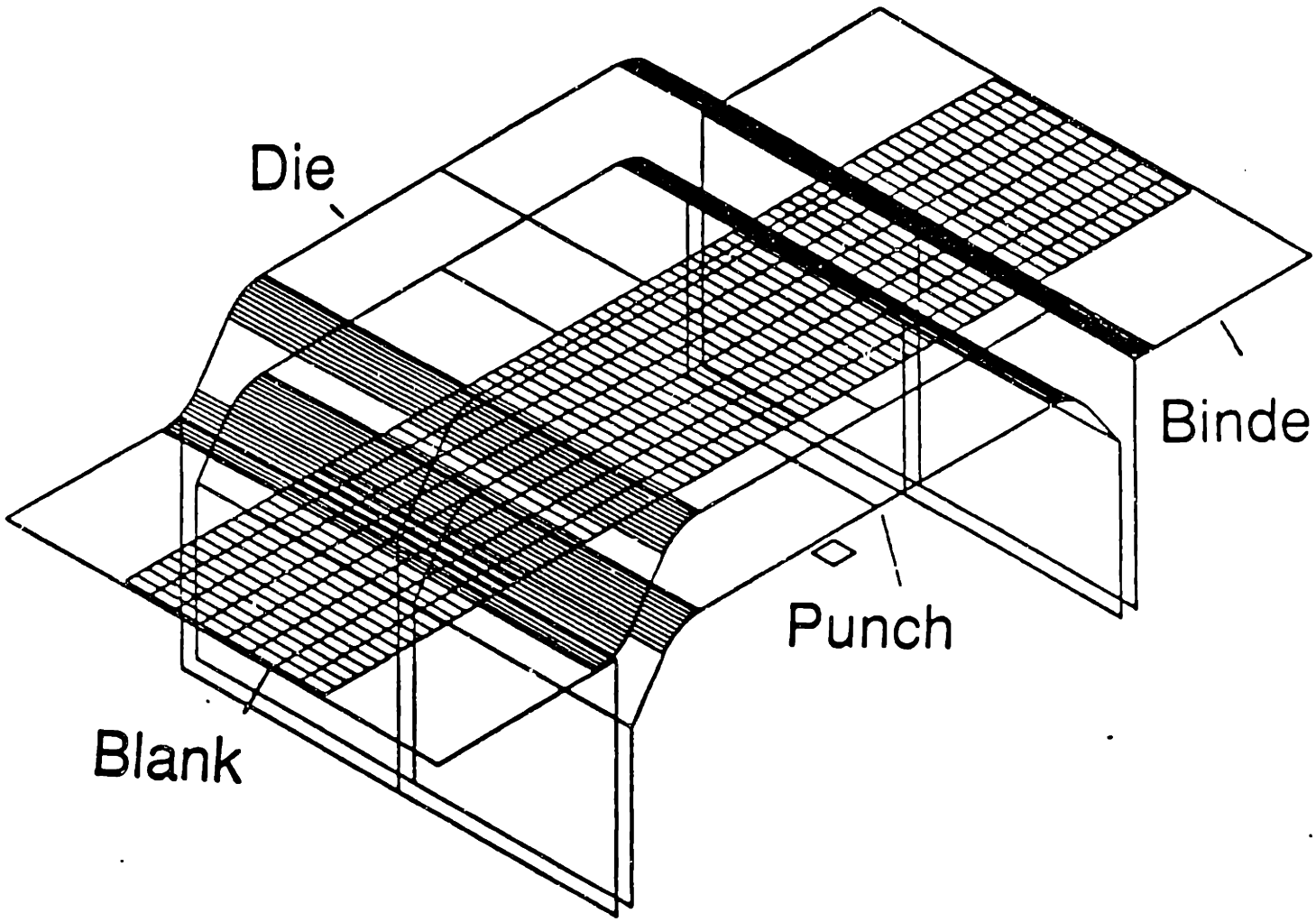


Fig. 4.1: Tooling and undeformed blank for the modeled channel geometry

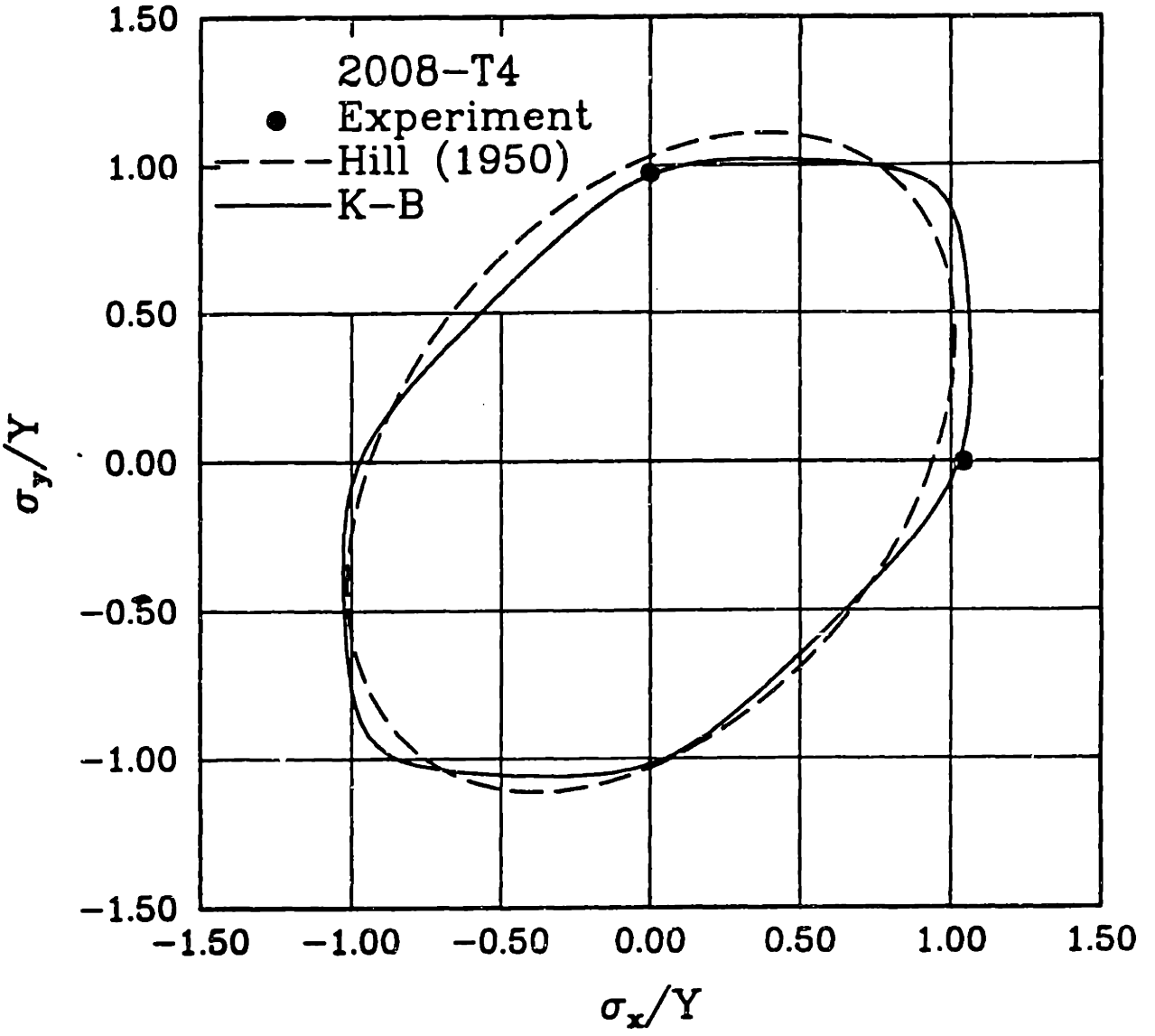


Fig. 4.2: The yield locus of Karafillis and Boyce [21] and Hill [22] for the 2008-T4 aluminum.

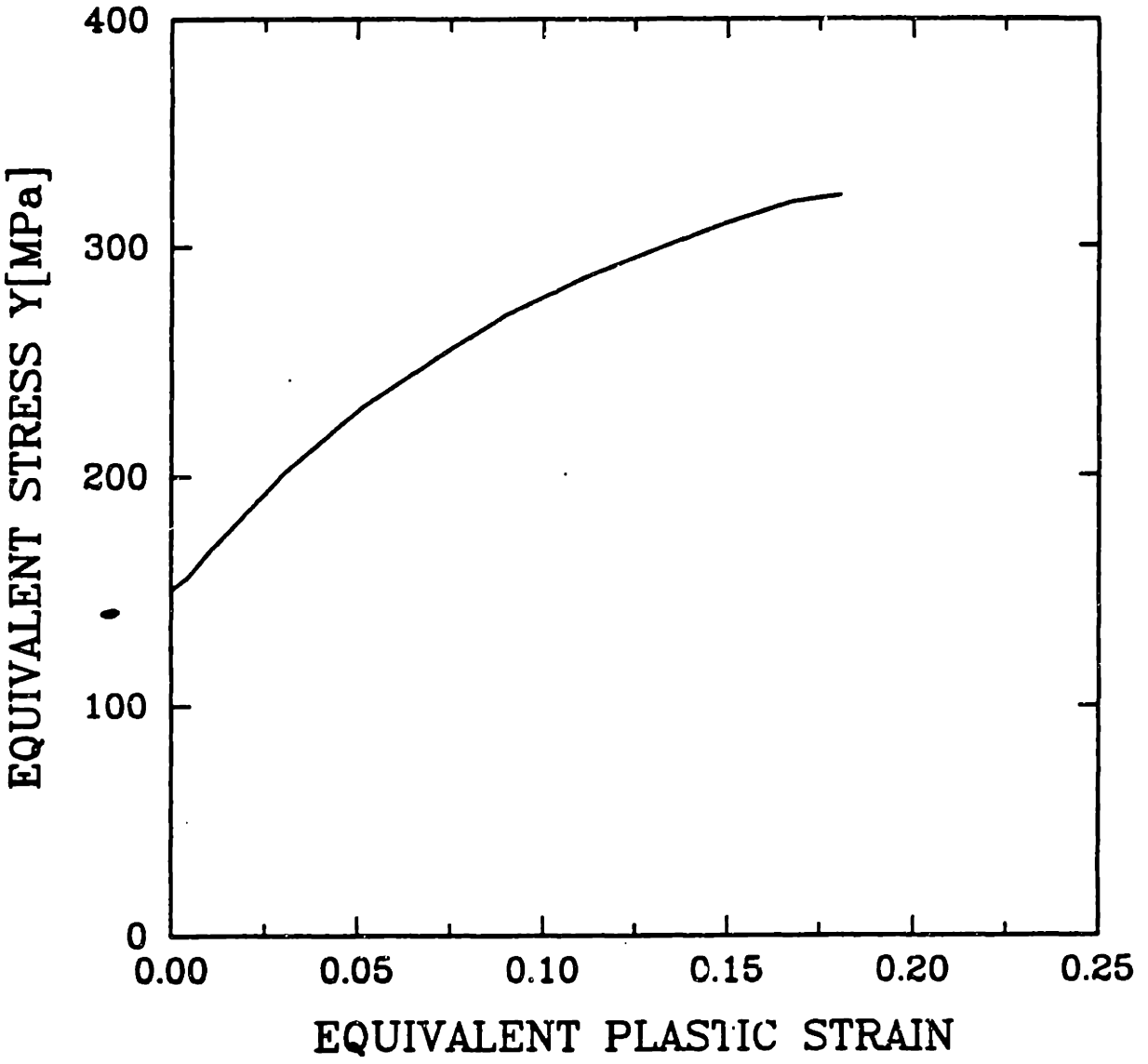
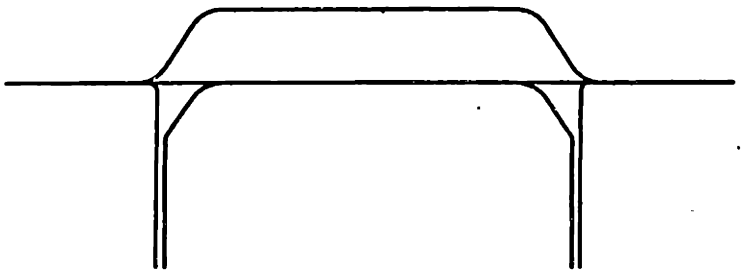
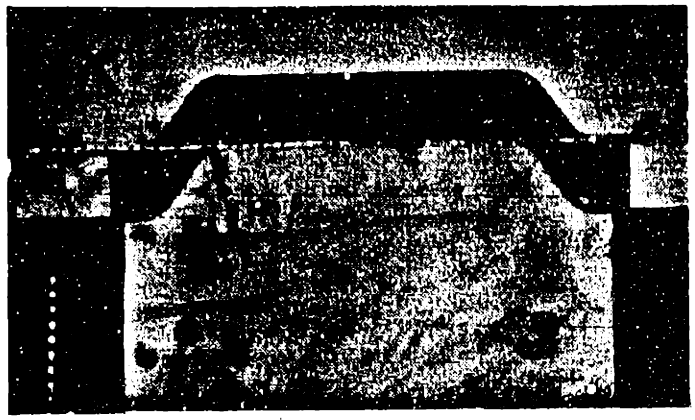
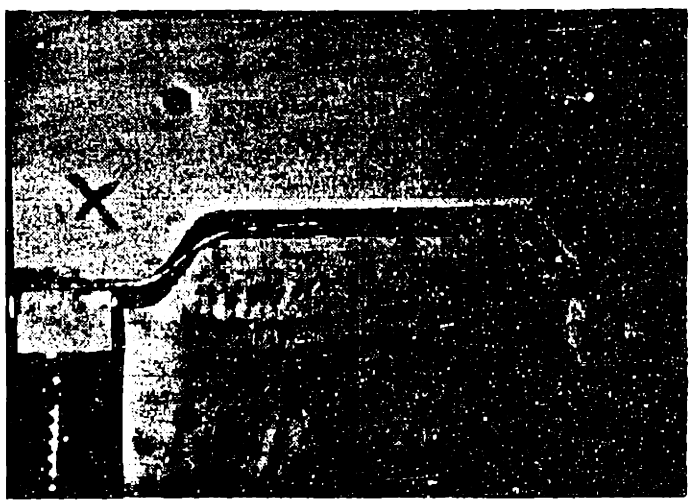


Fig. 4.3: The equivalent stress - equivalent strain curve of the examined 2008-T4 alloy.

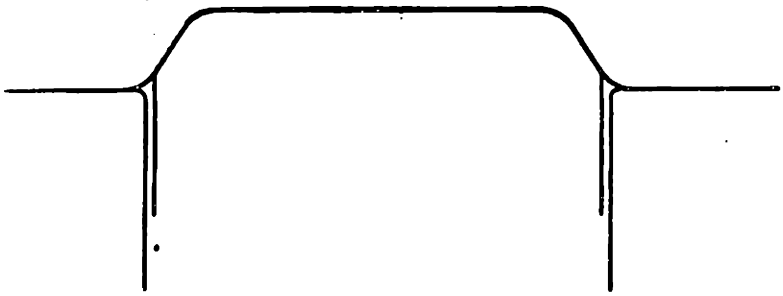




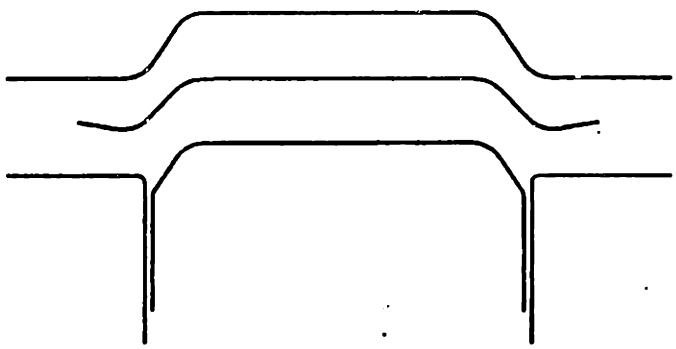
(a)



(b)



(c)



(d)

Simulation

Experiment

Fig. 4.4: Side view of successive steps of the channel forming process obtained by simulation (left) and by experiment (right); (a)-(c) loading; (d) unloading.

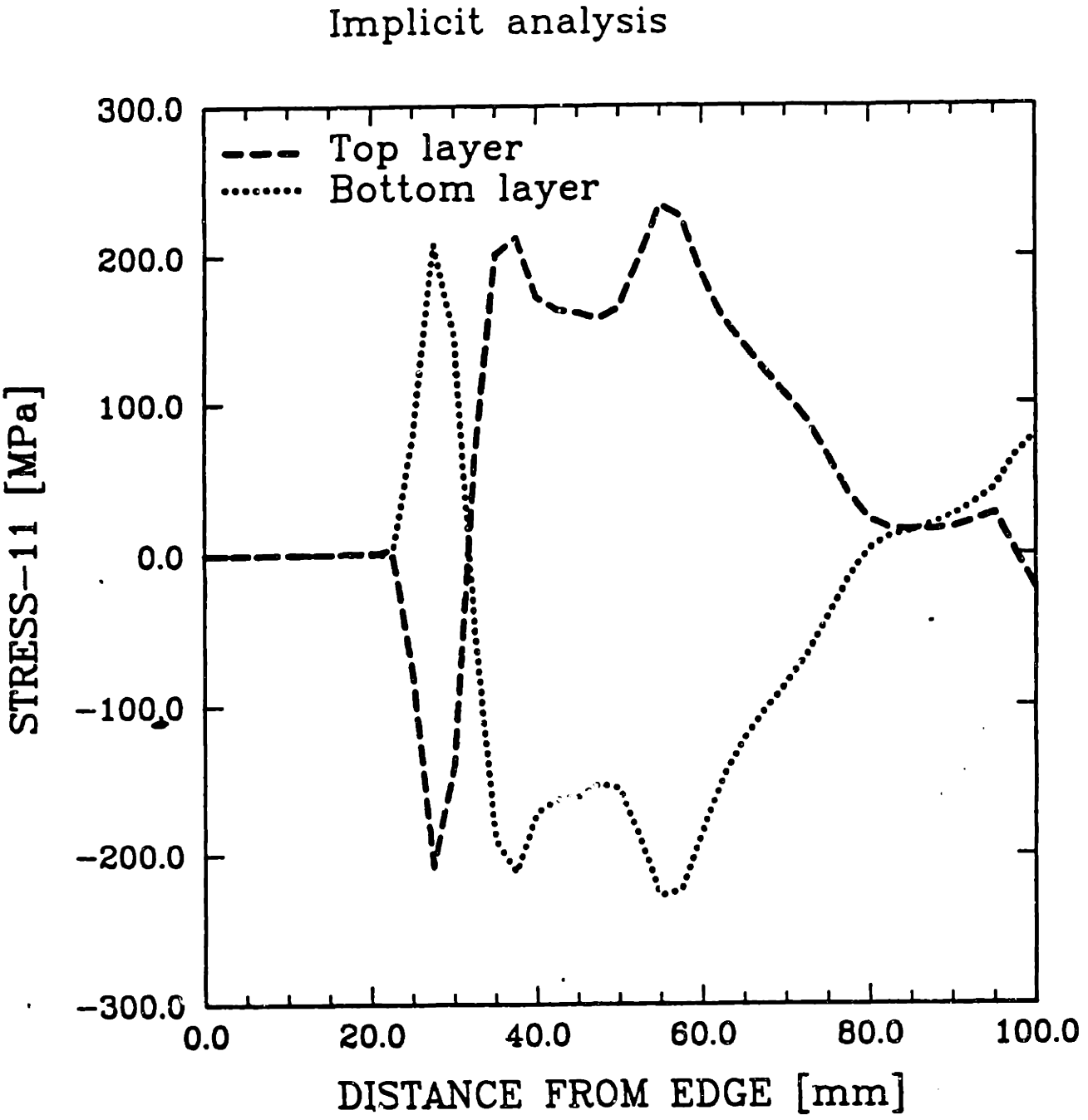


Fig. 4.5: Stress distribution at the top and the bottom of the blank in the fully loaded stage.

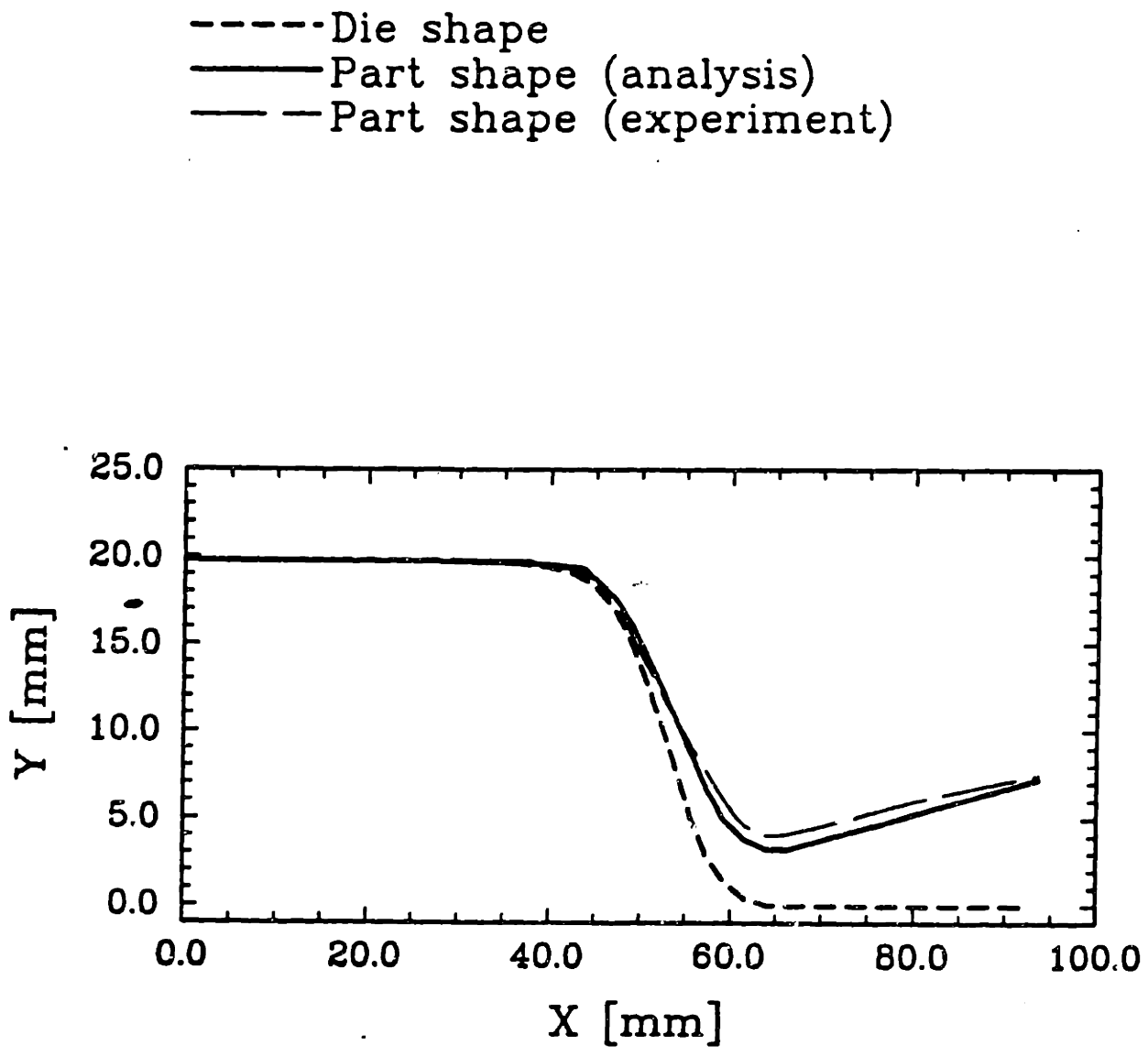


Fig. 4.6: Profile shapes of the produced part upon unloading obtained by theory and experiment.

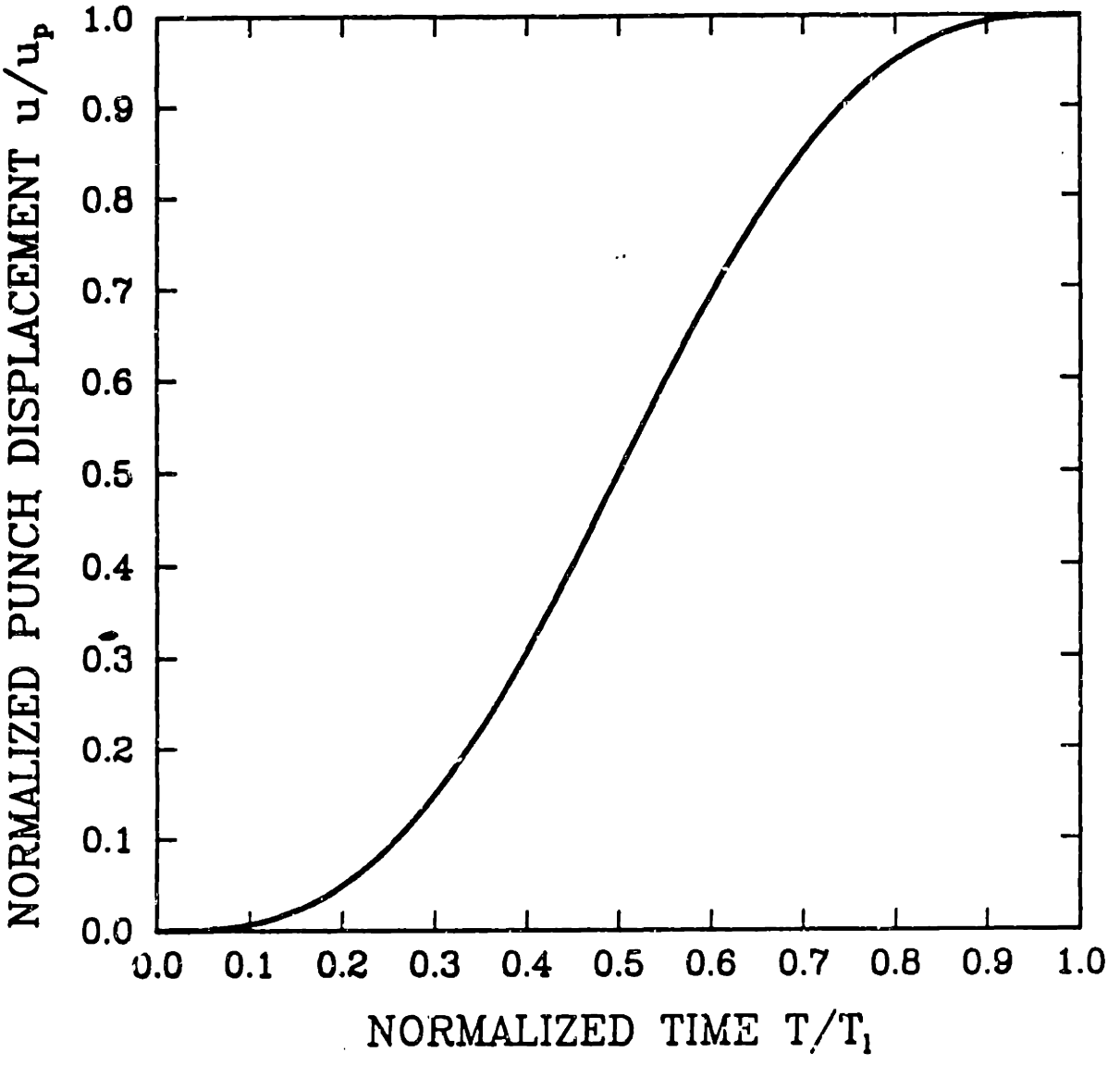


Fig. 4.7: Displacement history used during the explicit loading simulations.

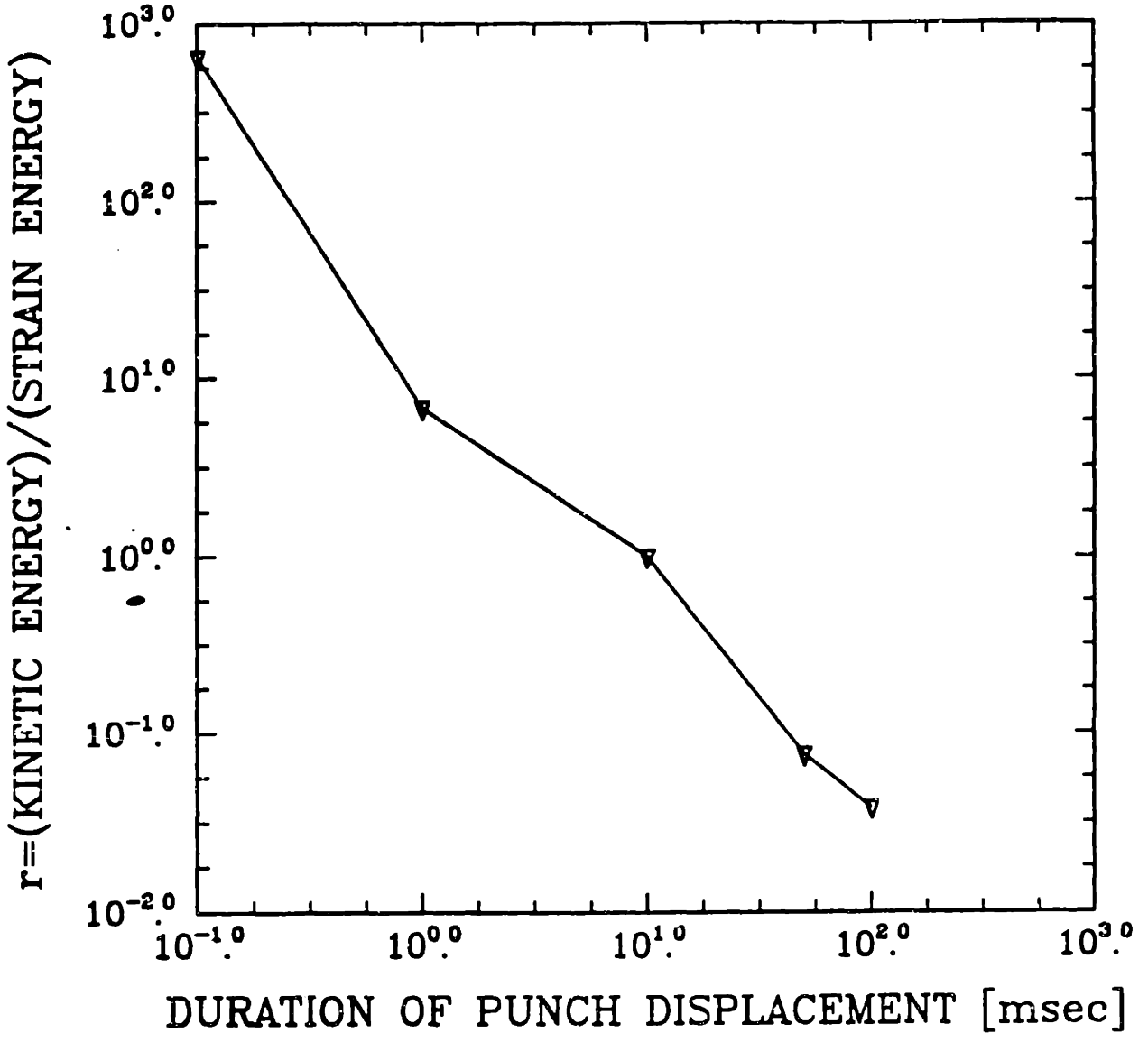


Fig. 4.8: The ratio of (kinetic energy)/(elastic strain energy) of the blank as a function of the duration of the simulated phenomenon.

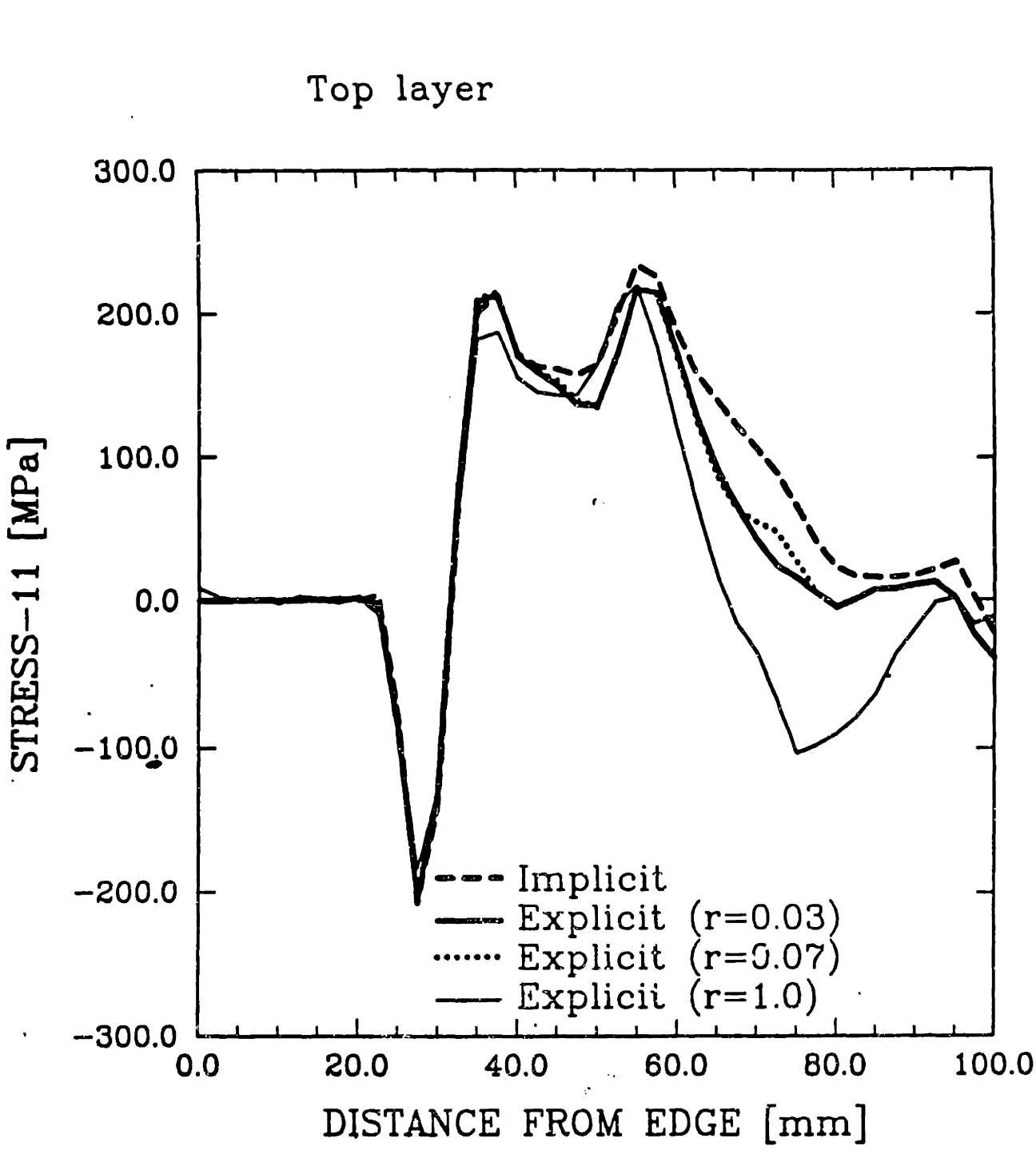


Fig. 4.9: Stress distribution on the top layer of the blank obtained by implicit static and explicit dynamic finite element analysis.

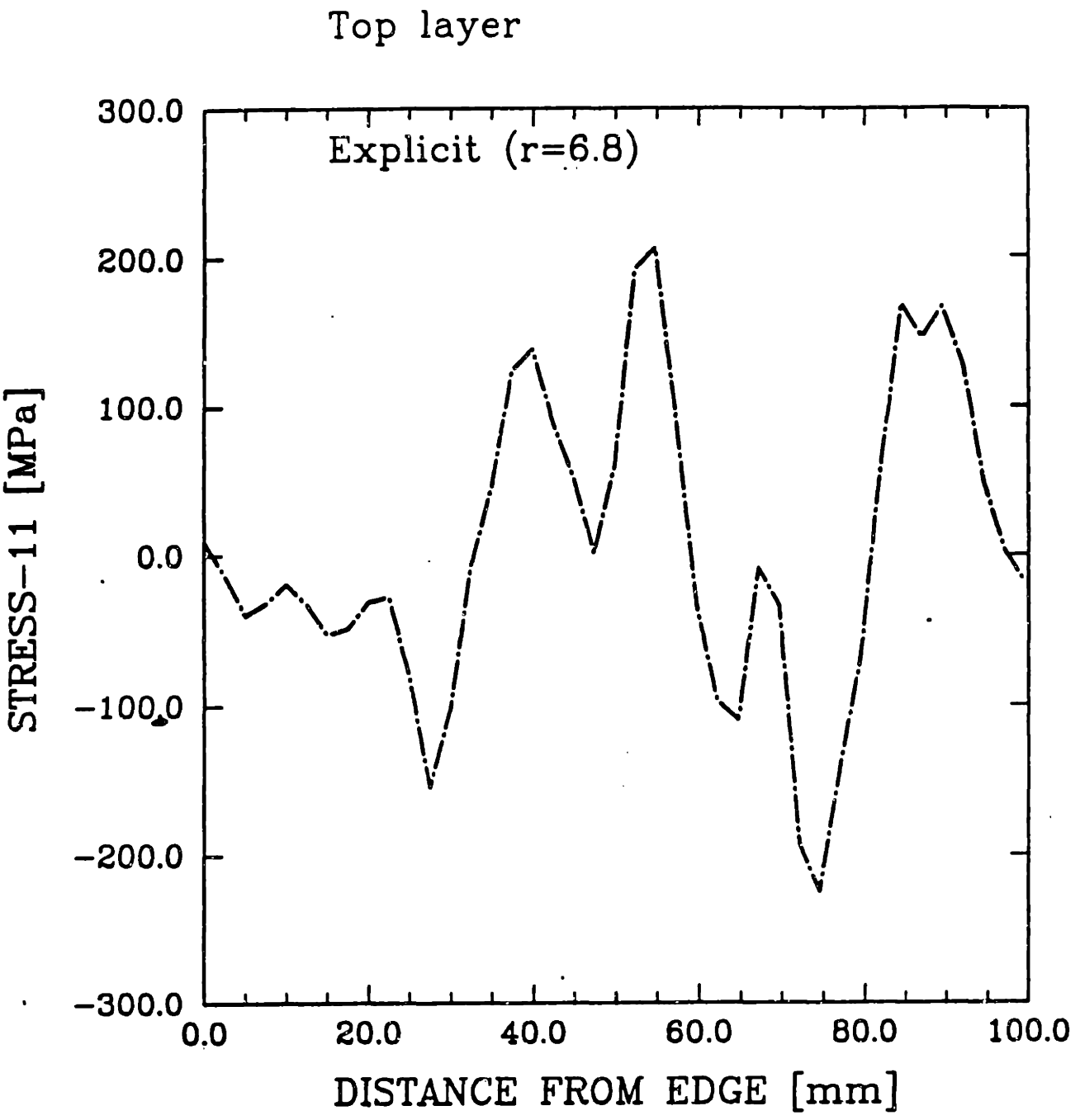


Fig. 4.10: Stress distribution on the top layer of the blank obtained by explicit dynamic analysis with a large value of r.

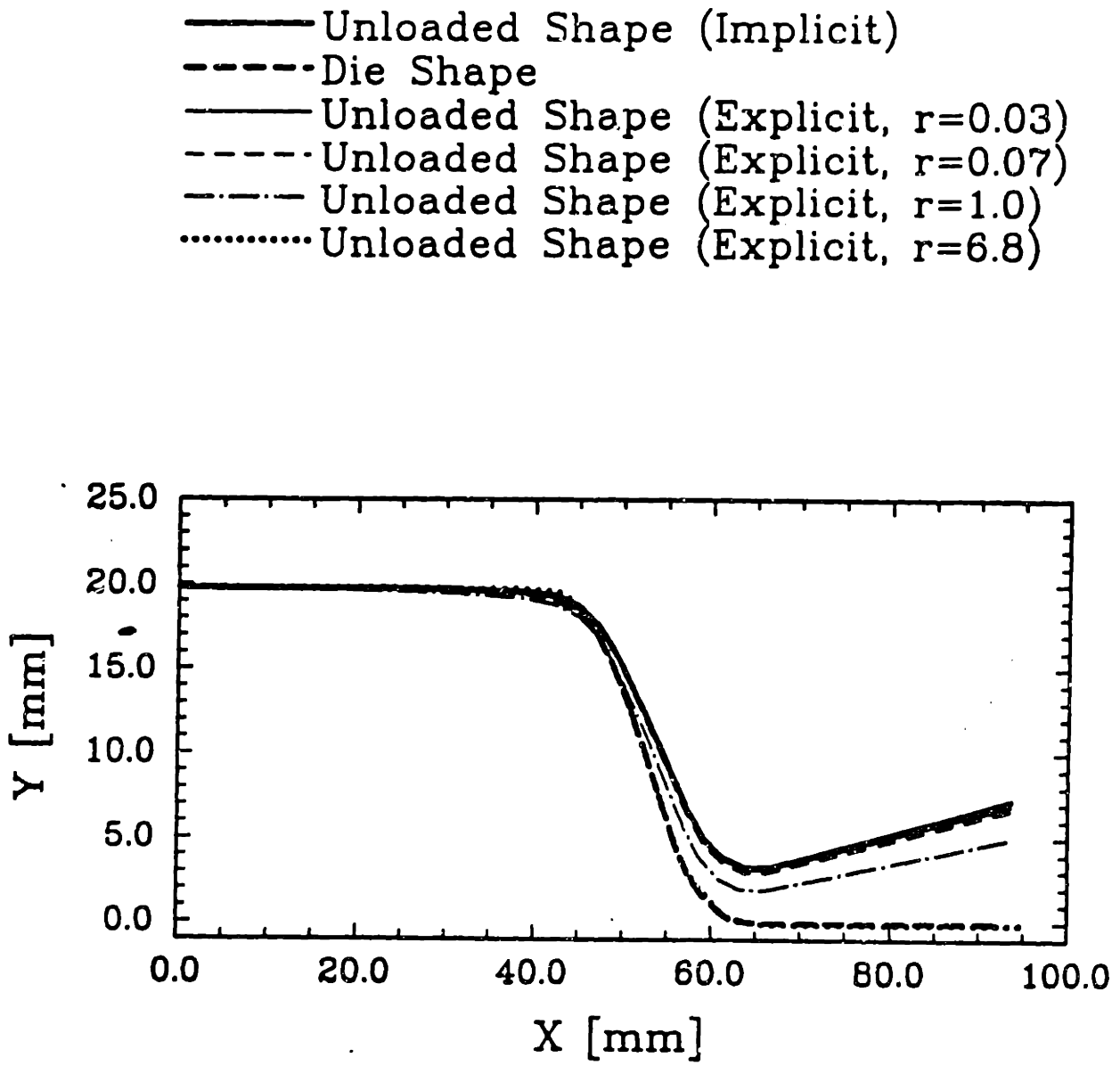


Fig. 4.11: Part shape upon springback calculated with the combination of dynamic explicit analysis for loading and static implicit analysis for unloading.

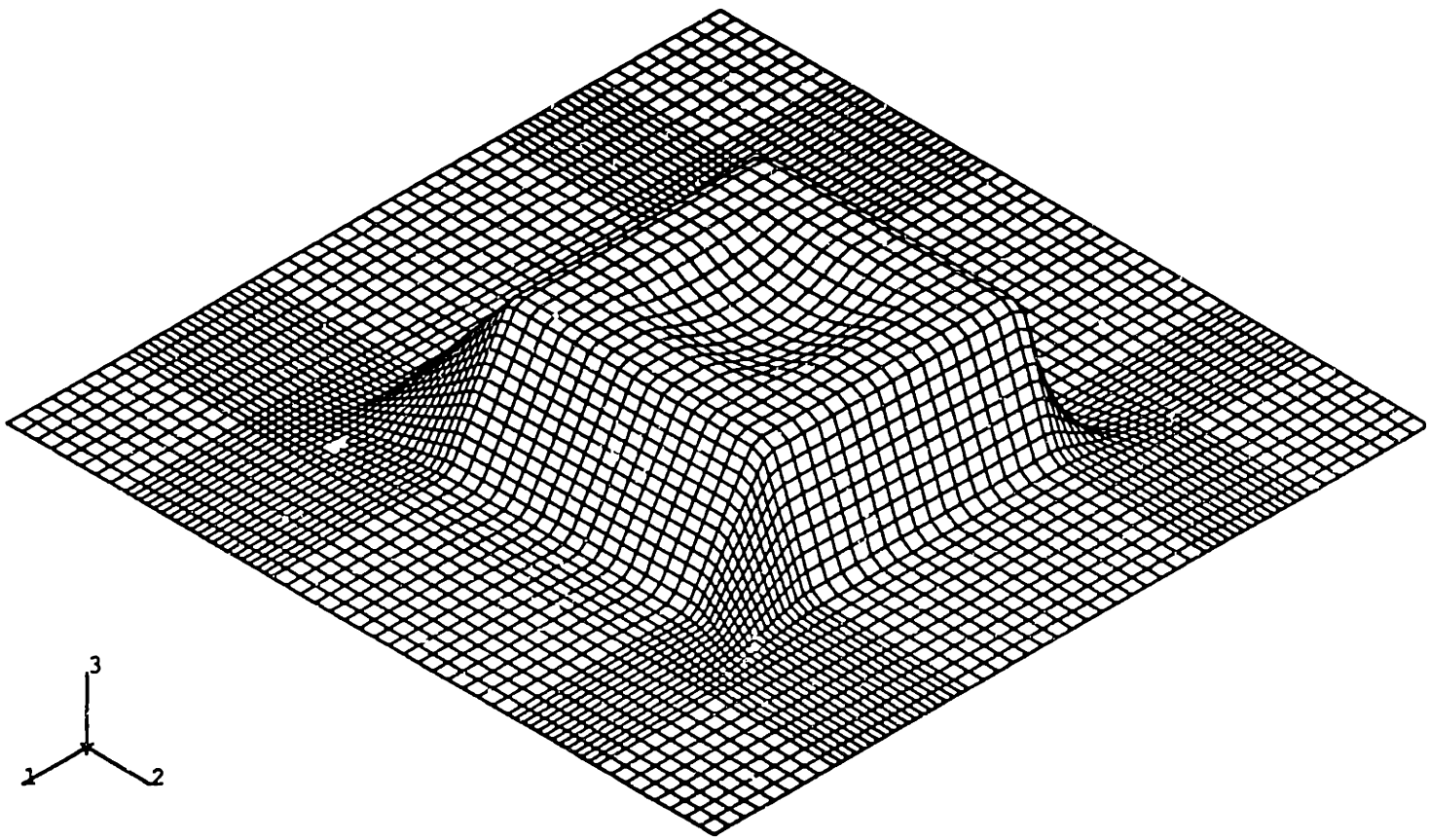


Fig. 4.12: Initial pan design

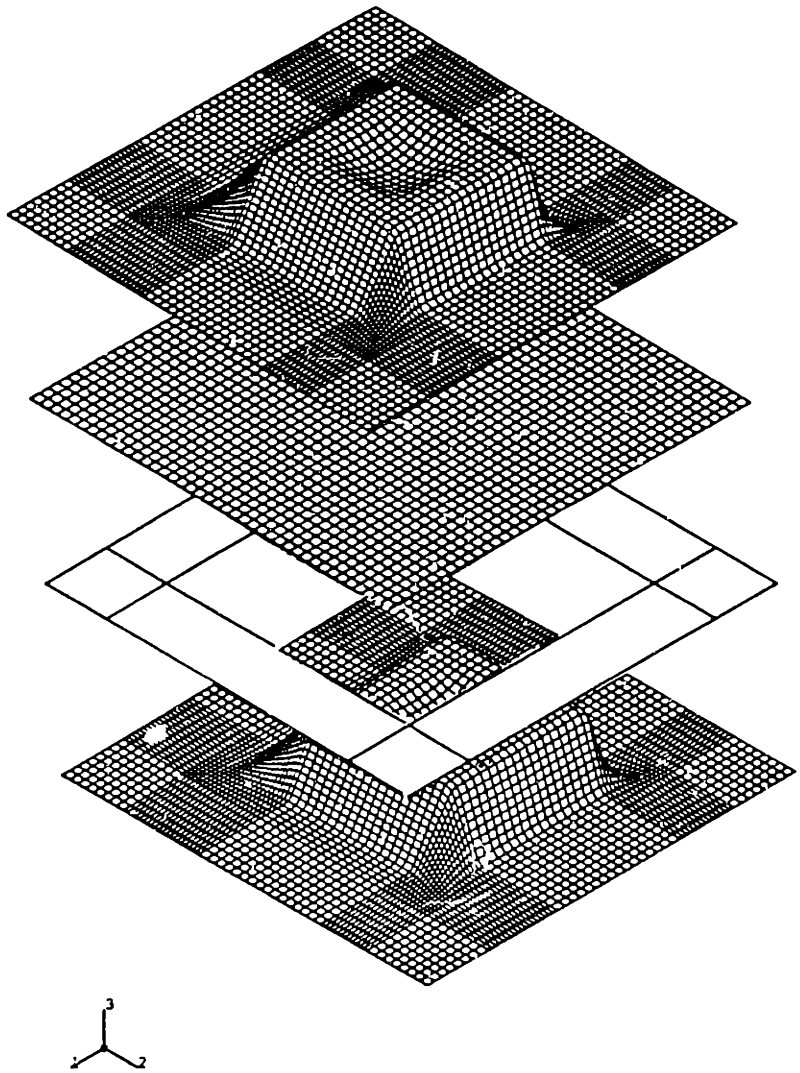


Fig. 4.13: Tooling set-up in finite element model.

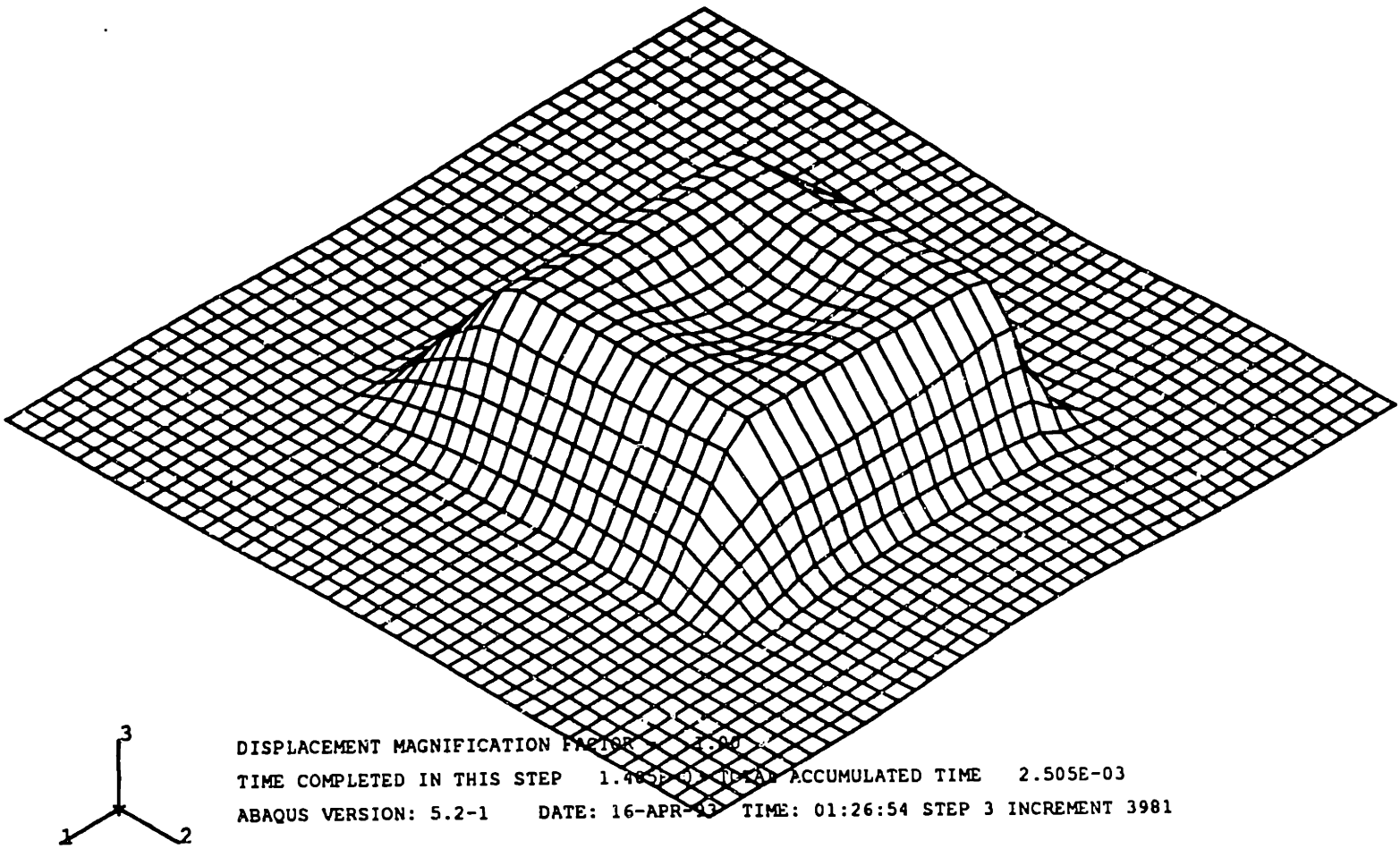


Fig. 4.14: Deformed part at the fully loaded stage.

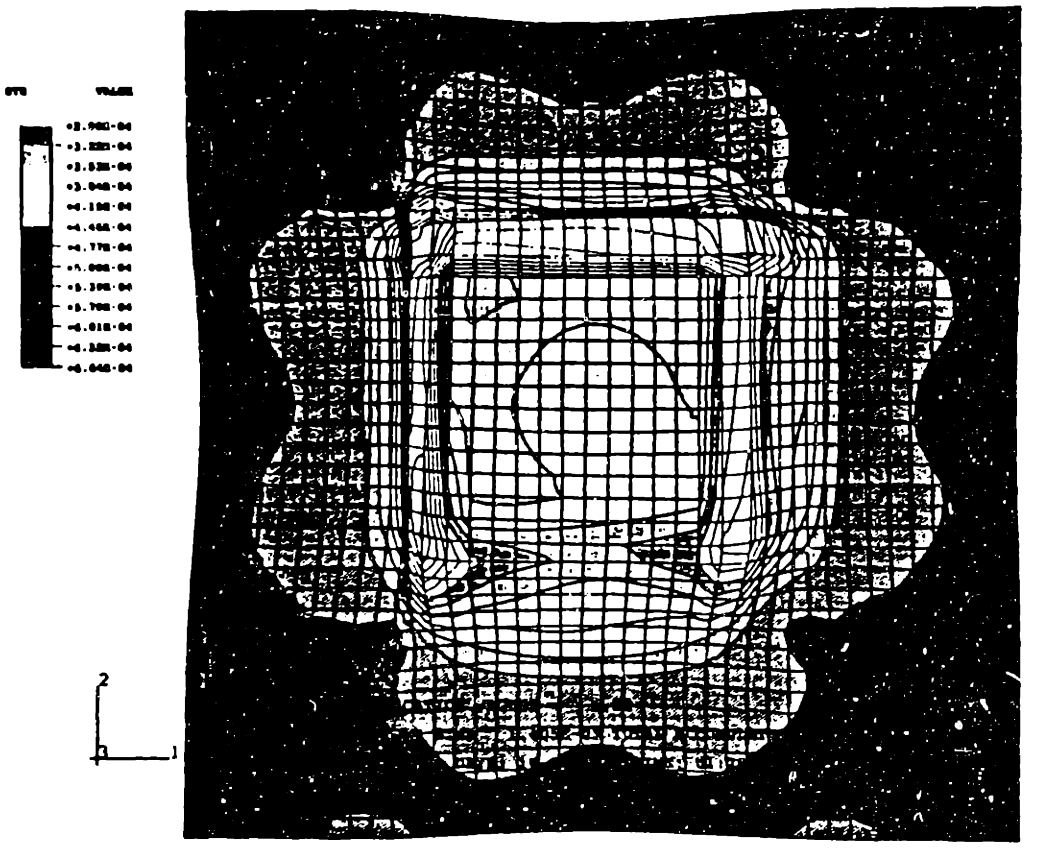


Fig. 4.15: Contour of blank thickness on the formed part (1st design)

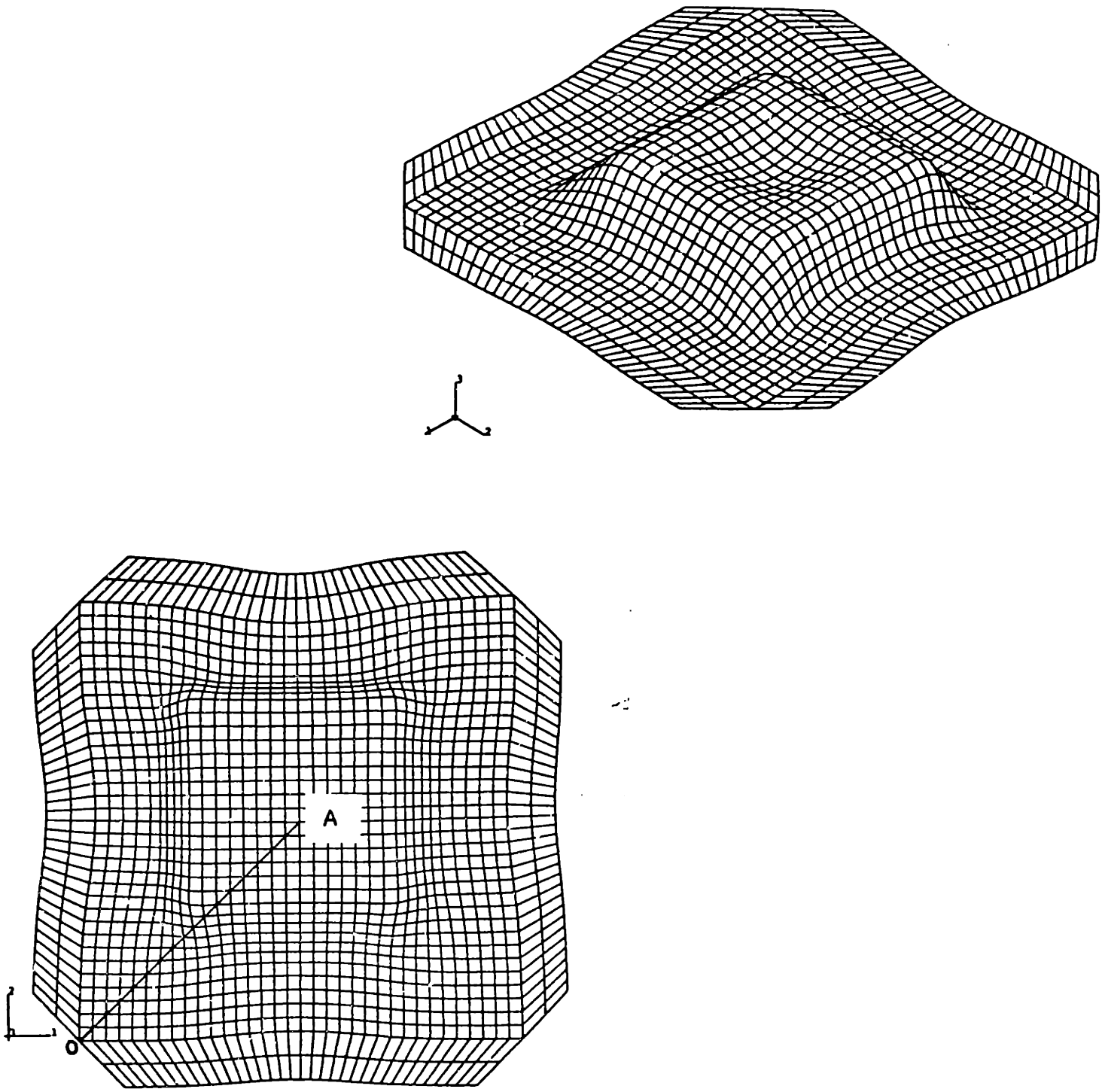


Fig. 4.16: Altered part design

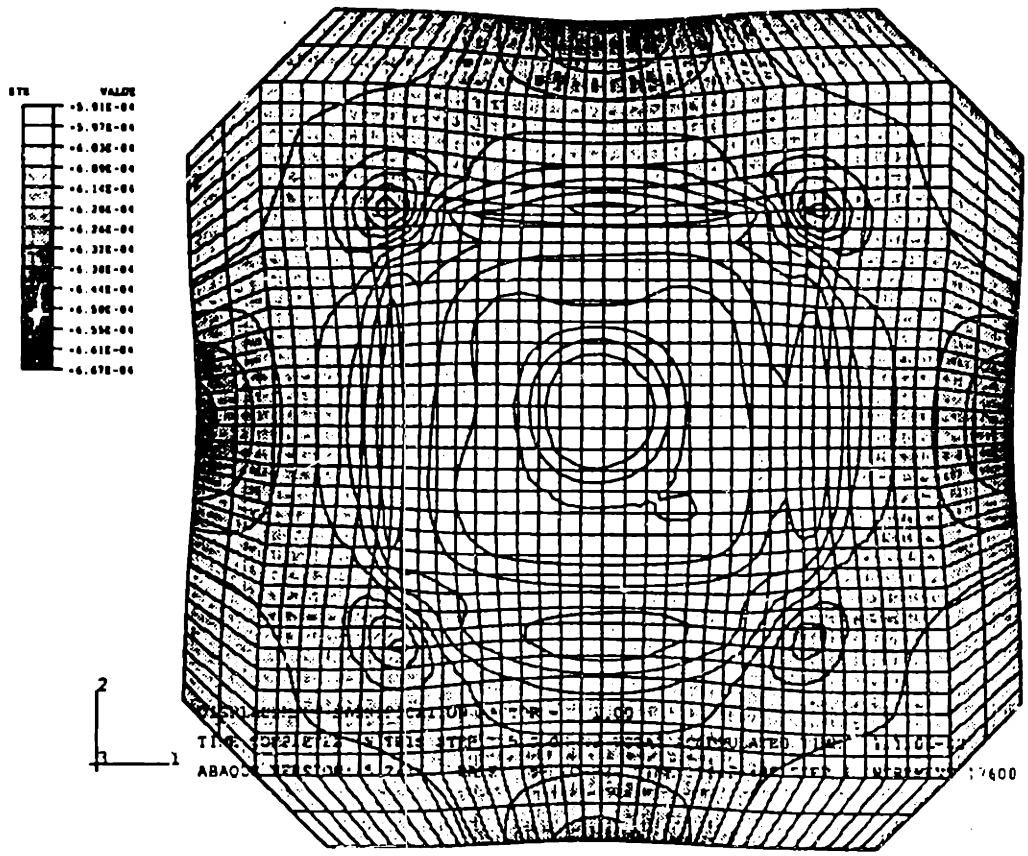


Fig. 4.17: Contour of blank thickness on the formed part (2nd design)

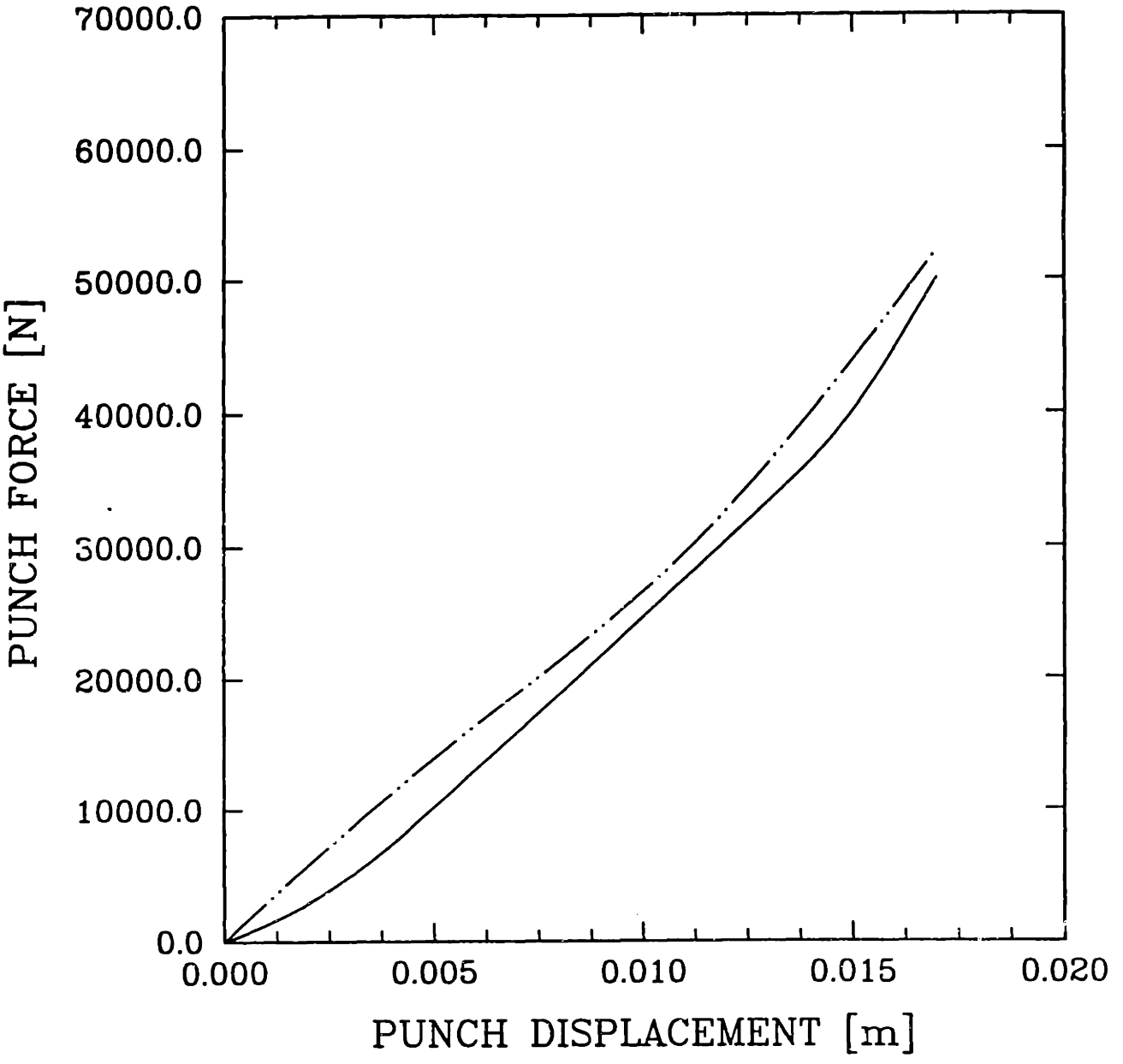


Fig. 4.18: Force-displacement curves for the steel pan.

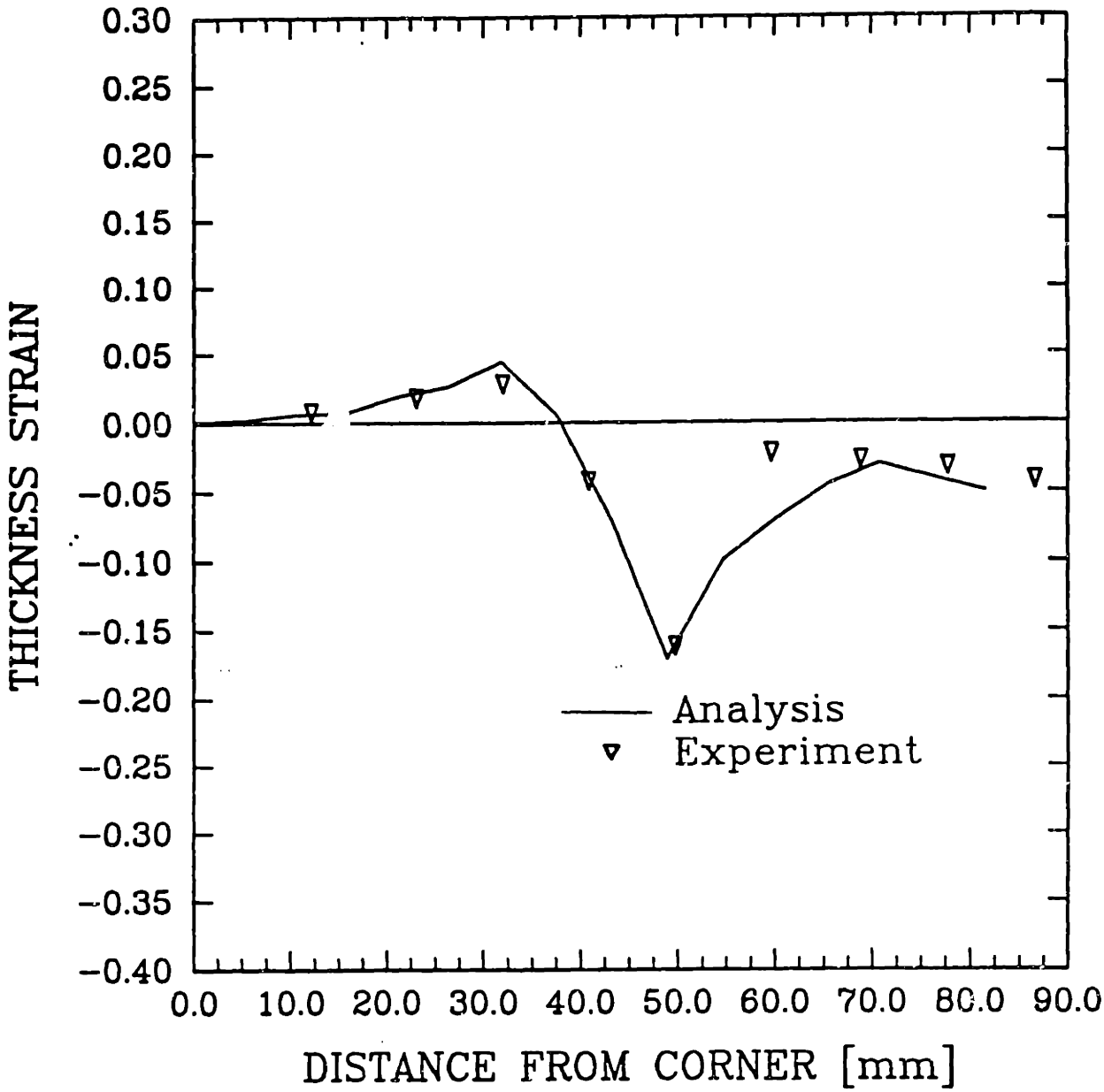


Fig. 4.19: Thinning strain along part diagonal OA on the top of the blank (steel).

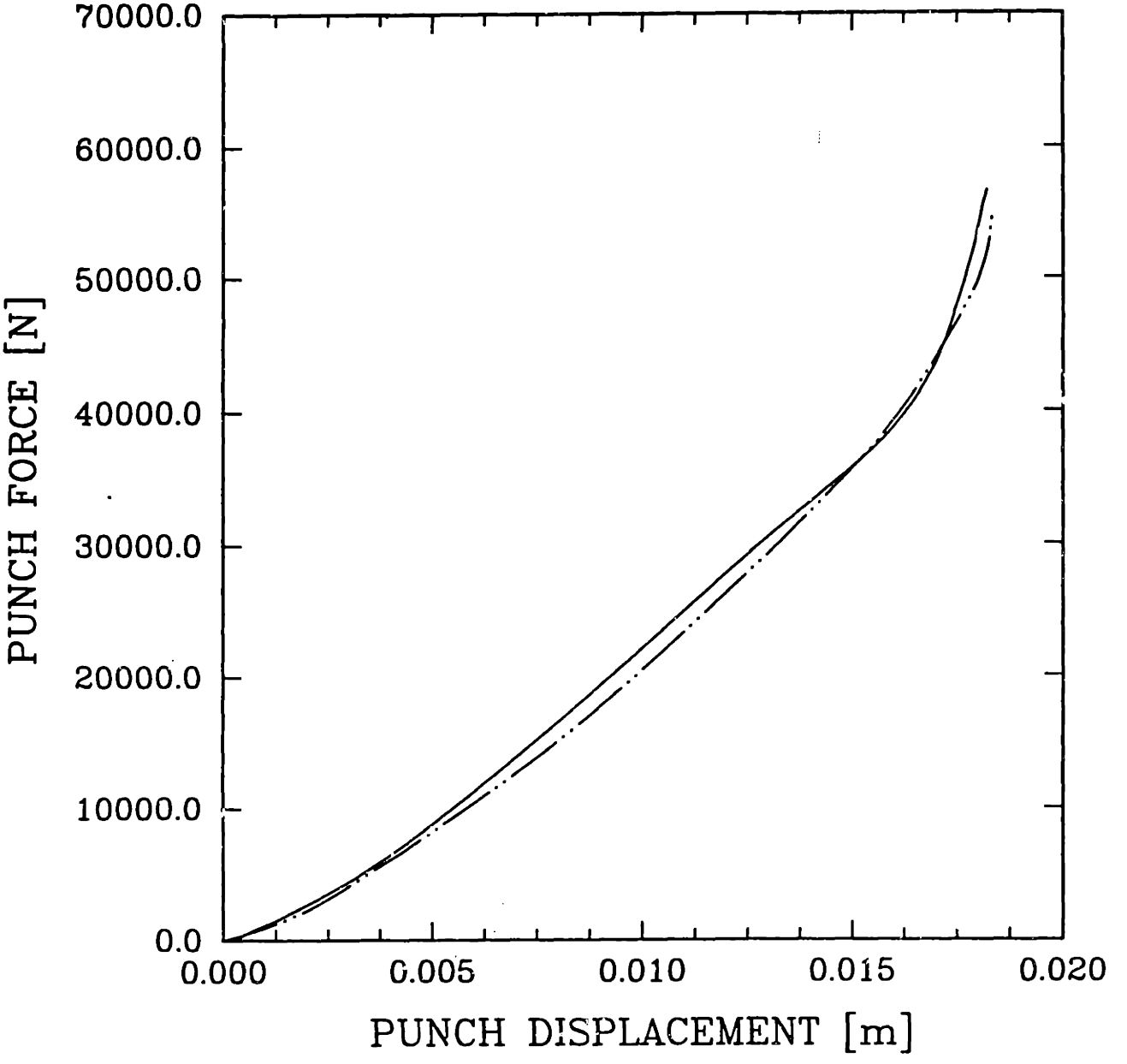


Fig. 4.20: Force-displacement curves for the aluminum pan

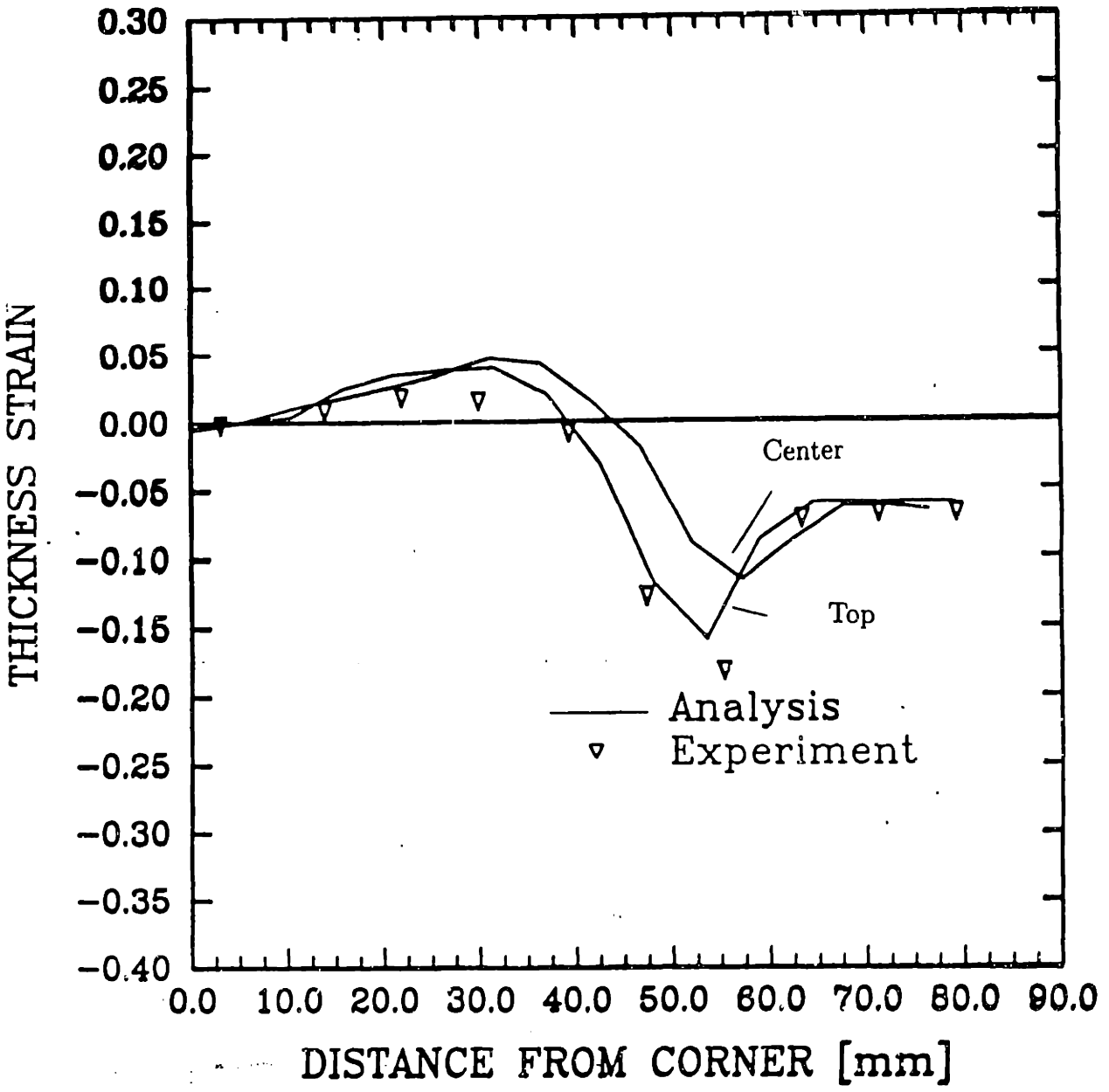


Fig. 4.21: Thinning strain along part diagonal OA on the top of the blank (aluminum).

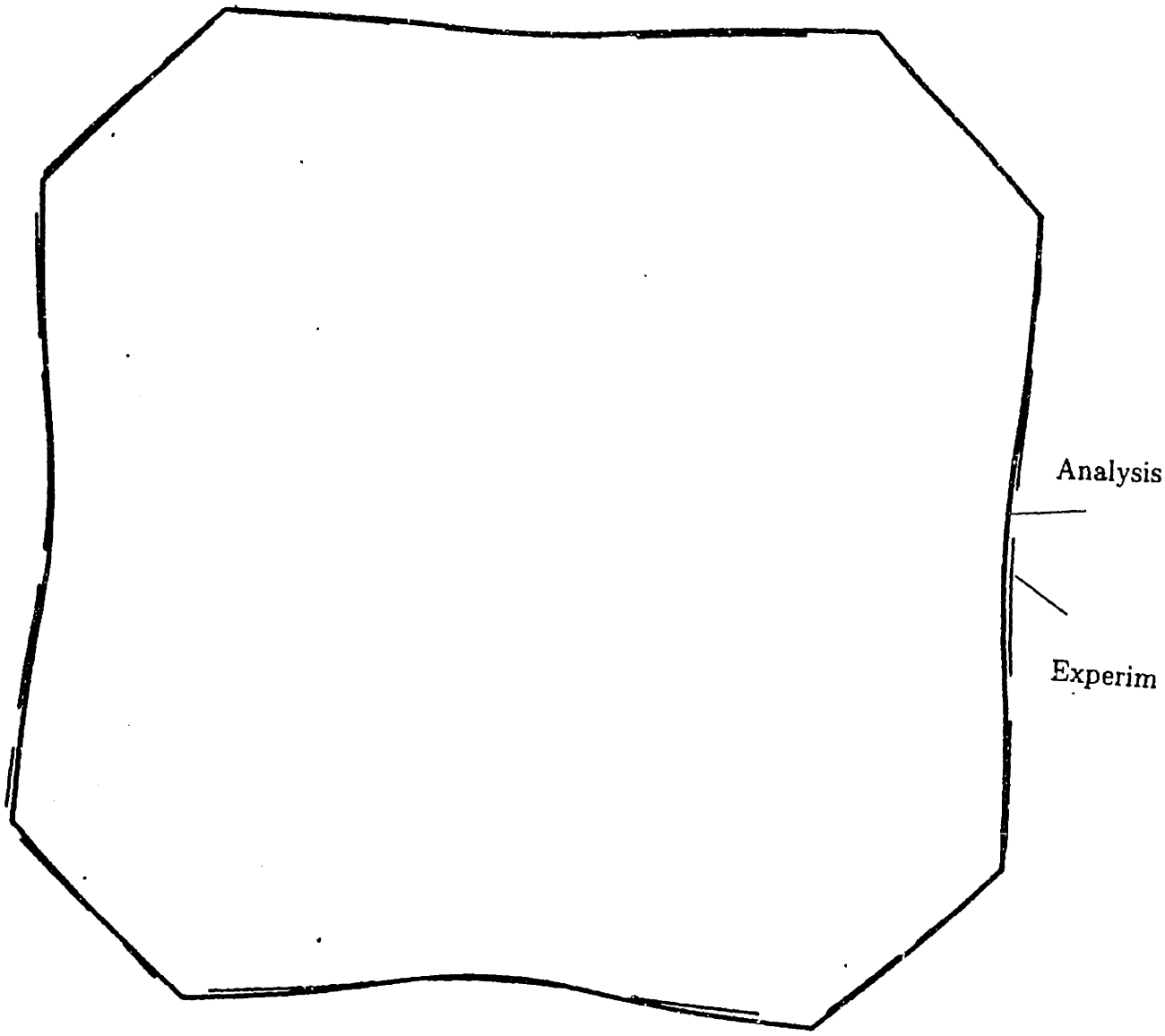


Fig. 4.22: Top view of perimeter (analysis with K-B yield function).

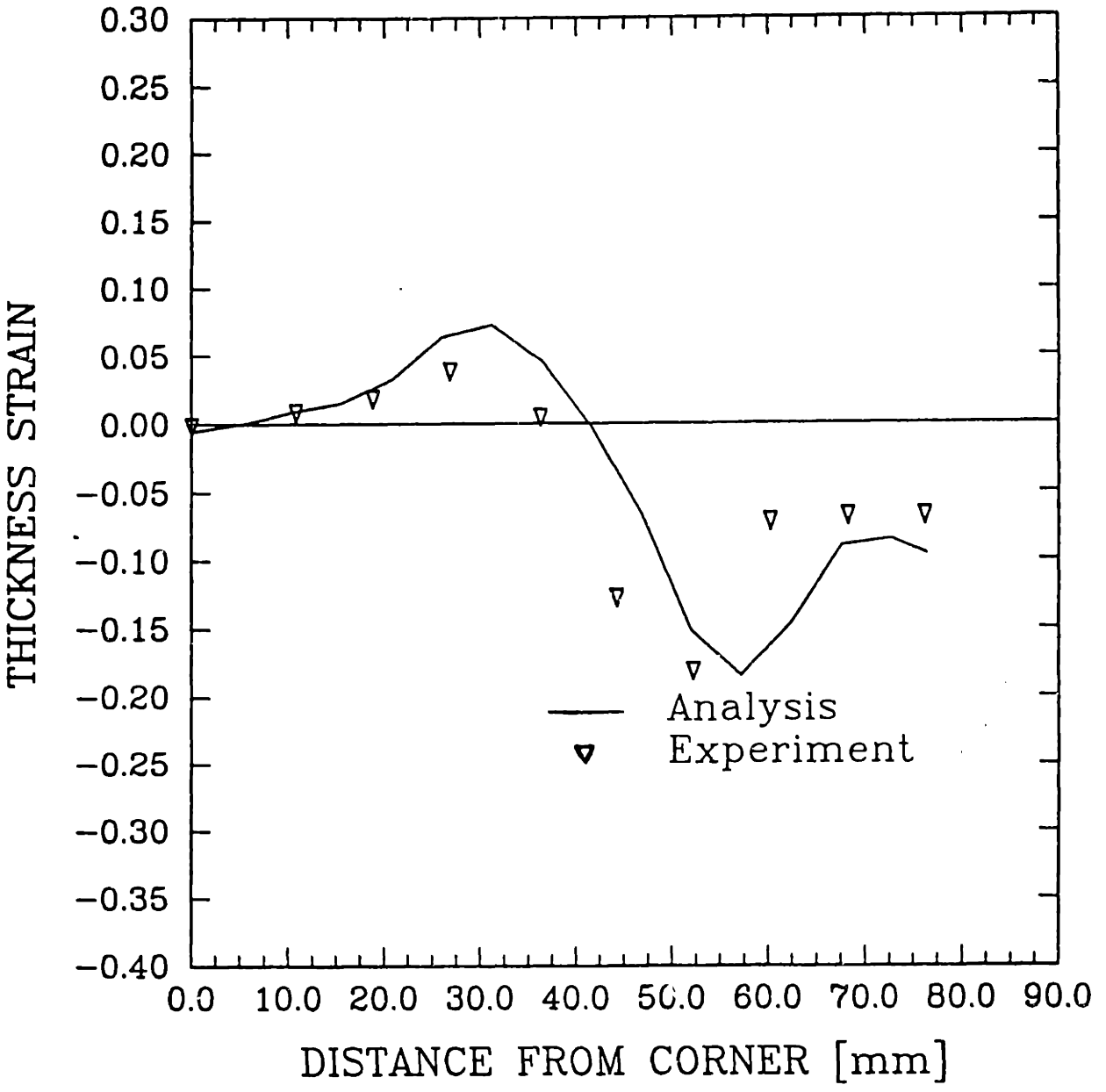


Fig. 4.23: Thinning strain along part diagonal OA on the top of the blank (aluminum), obtained with Hill's quadratic yield function [1950].

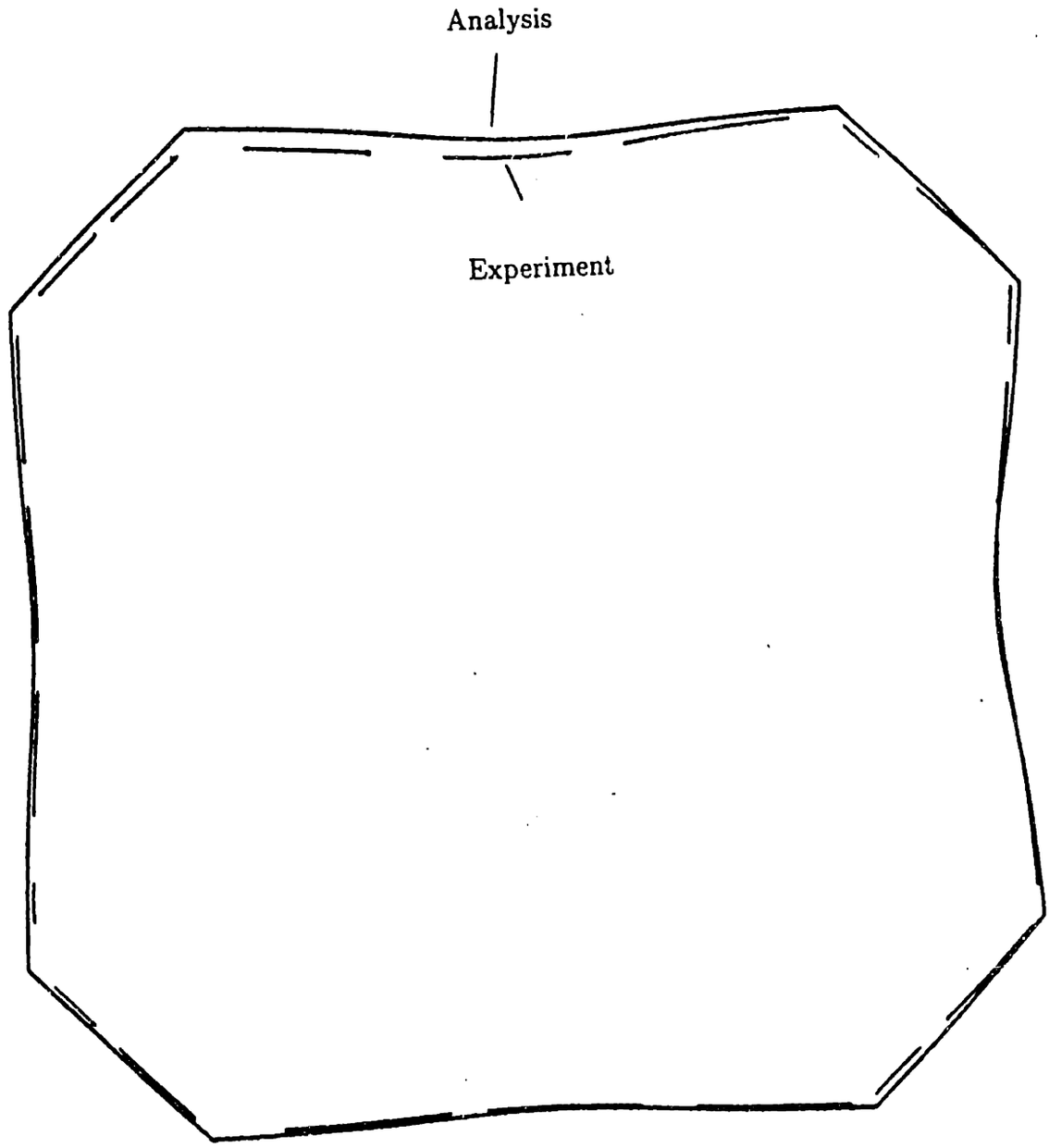


Fig. 4.24: Top view of perimeter (analysis with Hill's quadratic yield function [1950]).

5 CHAPTER 5: TOOLING AND BINDER DESIGN FOR SHEET METAL PROCESSES COMPENSATING FOR PART SHAPE ERROR DUE TO SPRINGBACK

5.1 Introduction

Sheet metal forming processes involve a combination of elastic-plastic bending and stretching deformation of the workpiece. These deformations can lead to large amounts of springback of the formed part upon unloading (e.g. removal of the punch and lifting of the blankholder). Springback has been examined as a manufacturing problem by a number of investigators, (for example Ayres [1984], Liu [1988], Stevenson [1993], Wenner [1983], Karafillis [1992]). Various methods for the reduction of the amount of springback have been proposed by Liu [1988] and Ayres [1984]. The purpose of these methods is to minimize the deviation of the produced part shape from the tooling shape which is identical to the desired part shape. Ayres [1984] has suggested the use of a multi-step process to reduce springback while avoiding part failure. Liu [1988] has proposed variations of the binder force during the forming process, thereby providing tensile pre-loading or post-loading on the formed part in order to reduce springback. In both of these methods, the reduction of springback is effected by the provision of high amounts of stretching of the formed part. Woo and Marshall [1959] and Wenner [1983] showed that tensile stretching stresses superimposed on the bending stresses of an elastic-plastic material cause the reduction of the springback in two-dimensional formed parts. However, as also pointed out by Wenner [1983], it is not always possible to transmit high tensile forces to all parts of a workpiece of a complicated geometry without causing failure by tearing of the workpiece. The concept of the use of the material restraining force as a control variable in order to eliminate tearing and/or wrinkling defects has been examined by Hardt and Fenn

[1990], Sim and Boyce [1991], Jalkh et al. [9], and Cao et al. [10]. However, in the present work we treat the shape control aspect of forming separately from the binder constraining control, in order to assess springback perturbation to shape fidelity independently from material restraining effects which can compromise the forming of the part with respect to tearing.

Hardt and coworkers [1984, 1991] considered part shape error due to springback as a shape control problem. In their approach, the punch and the matching die shape were modified in a closed loop algorithm which acted to correct the part shape error. The material restraining force was large and remained unaltered in all iterations of the algorithm allowing minimal draw-in of the material in the forming area. The punch and the die shape were modified until the part shape coincided with the desired shape. The closed loop control algorithm developed and implemented by Webb and Hardt [1992] used the concept of the "Deformation Transfer Function". This concept relates changes in the die shape to changes in the part shape. Both the die shape and the part shape were described in the frequency domain by the aid of Fourier transform.

As an alternative to this approach, but following the same guidelines, Karafillis and Boyce [1992] developed a tooling design algorithm based on finite element analysis of the stamping process. In their method, the traction distribution on the workpiece was calculated at the fully loaded stage [2]. This traction distribution was used to elastically load a part that had the desired part shape in a finite element model, thus springing forward the part shape to give the required die shape. We call this calculation an inverse simulation of the springback phenomenon. This method showed fast convergence where less than 3% part shape error was obtained even after the first iteration. By comparison, the Deformation Transfer Function Method requires two

[2]In both simulation and analysis, fully loaded stage is the situation where the punch has nested to the die, causing the punch load to start rising at a very rapid rate, due to simultaneous contact of the blank with the punch and the die

cycles for the establishment of the Deformation Transfer Function before any die shape correction can be computed. The inverse springback method of Karafillis and Boyce [1992], called the "Force Descriptor Method", was applied to two different shapes with successful results, see also Karafillis and Boyce [1992]. We note here that in the processes considered by Karafillis and Boyce [1992] no material draw-in was allowed. Also the part shapes and the materials considered were axisymmetric whereas the part shape error was measured in the unloaded configuration (punch removed) but with the binder fully clamped.

In the present method we extend the "Force Descriptor Method" to three-dimensional geometries, where material draw-in is allowed and where the part shape errors are considered at the unclamped stage. Some of these steps were also suggested by Story [1993] in his recent review paper of current trends and issues in the forming of aluminum sheets.

5.2 Tooling and binder design by using springback calculations

The occurrence of large amounts of springback can cause considerable shape deviation from the tooling shape, see also Fig. 4.6. If shape fidelity is required from the produced part of a sheet metal forming process, the correct part shape can be obtained by appropriate modifications of the die and the binder shape.

Systematic methods (as opposed to traditional trial and error approaches) to address tooling and binder design have been proposed by Hardt and coworkers [1984, 1991], and by Karafillis and Boyce [1992]. However, these methods concentrated on forming processes with limited draw-in of the blank, and, also, the formed part shape was considered upon unloading of the blank but with a closed binder, see also Karafillis and Boyce [1992]. In the present work, we extend the procedures developed by Karafillis and Boyce [1992] to the more general case of forming processes with

large amounts of draw-in in order to contain the developed stretching strains and eliminate tearing failure. Also, the formed parts are considered in the fully unloaded configuration.

We now consider an initial finite element analysis of the stamping process, where the tooling shape is identical to the desired part shape. The shape of the blank in the finite element model is described by the displacement vector $\boldsymbol{u}_{l1}$, whereas the material stress-strain response causes the development of the internal force vector $\boldsymbol{I}_{l1}$. Also, we set the tooling to be identical to the desired shape in order to obtain $\boldsymbol{u}_{l1}$, and we therefore obtain $\boldsymbol{u}_{l1} = \boldsymbol{u}_{des}$. Springback occurs upon removal of all contact forces $\boldsymbol{F}_{l1} = \boldsymbol{I}_{l1}$, and the new displacement vector $\boldsymbol{u}_{ul1}$ of the blank is:

$$\boldsymbol{u}_{ul1}(\boldsymbol{u}_{l1}, \boldsymbol{I}_{l1}) = \boldsymbol{u}_{l1} + \boldsymbol{u}_{sb1}(\boldsymbol{u}_{l1}, \boldsymbol{I}_{l1}) \tag{5.1}$$

where $\boldsymbol{u}_{sb1}$ is the displacement increment vector during springback. Also springback can be considered as the additional deformation described by $\boldsymbol{u}_{sb1}$ which occurs due to the application of $-\boldsymbol{F}_{l1}$ on the blank in the fully loaded state and configuration resulting in a zero net load applied to the blank.

We now consider a process where the blank is deformed to a different shape $\boldsymbol{u}_{l2}$ which will springback upon unloading to a new shape $\boldsymbol{u}_{ul2}$:

$$\boldsymbol{u}_{ul2}(\boldsymbol{u}_{l2}, \boldsymbol{I}_{l2}) = \boldsymbol{u}_{l2} + \boldsymbol{u}_{sb2}(\boldsymbol{u}_{l2}, \boldsymbol{I}_{l2}) \tag{5.2}$$

We would like to have $\boldsymbol{u}_{ul2}$ to be identical to the desired shape:

$$\boldsymbol{u}_{ul2} = \boldsymbol{u}_{des} \tag{5.3}$$

where recall that

$$\boldsymbol{u}_{l1} = \boldsymbol{u}_{des} \tag{5.4}$$

giving the desired outcome of

$$\boldsymbol{u}_{ul2} = \boldsymbol{u}_{l1}. \tag{5.5}$$

In this case, $\boldsymbol{u}_{l2}$ would define the tooling shape that will produce the correct part shape, $\boldsymbol{u}_{ul2} = \boldsymbol{u}_{des}$. By rearranging Eqn. (4.10), we have:

$$\boldsymbol{u}_{l2} = \boldsymbol{u}_{ul2} - \boldsymbol{u}_{sb2}(\boldsymbol{u}_{l2}, \boldsymbol{I}_{l2}) \tag{5.6}$$

By combining Eqns. (4.13) and (4.14) we obtain:

$$\boldsymbol{u}_{l2} = \boldsymbol{u}_{l1} - \boldsymbol{u}_{sb2}(\boldsymbol{u}_{l2}, \boldsymbol{I}_{l2}) \tag{5.7}$$

In order to obtain $\boldsymbol{u}_{l2}$, we can perform an inverse simulation of springback where a blank defined by the displacement vector $\boldsymbol{u}_{l1}$ will be subject to the additional deformation $-\boldsymbol{u}_{sb2}$. However, since we do not know $-\boldsymbol{u}_{sb2}$, we can attempt to calculate $\boldsymbol{u}_{l2}$ in an iterative manner, until the part shape error is negligibly small. In the work of Karafiliis and Boyce [5] this was simply effected by considering the blank with the shape $\boldsymbol{u}_{l1}$, i.e. the desired shape in this configuration with internal forces set to zero, and then ramping on the nodes of this blank the contact forces $\boldsymbol{F}_{l1}$ as nodal follower loads. When the external force vector was ramped to its value $\boldsymbol{F}_{l1}$ the blank elastically deflected in a manner opposite to the one that occurs during springback giving a new equilibrium configuration. We called this an inverse springback calculation where we essentially are "springing forward" the desired shape to the required tooling shape. The obtained deformed shape is the next tooling shape used. If the desired shape is not obtained in the first iteration, then this procedure is repeated with the new force vector obtained from the simulation with the new die shape at the fully loaded stage. The algorithm stops when the obtained die shape produces a part shape which deviated from the desired part shape by the preset tolerances.

This method was effective for parts considered with the binder closed (the edge of the blank couldn't move). The binder restraint restricted the blank to small values of displacement during the inverse springback simulation ensuring that the geometry changes due to the displacements of the material nodes do not cause the development

of large unbalanced moments on the material. However, when considering parts at the fully unloaded stage, the deflection under the nodal point loads $\boldsymbol{F}_{l1}$ can be large enough to lead to the development of unbalanced moments which, in turn, generate large amounts of bending deformation of the blank.

In order to overcome this difficulty we modified the inverse springback simulation method. The flowchart of the procedures involved in this operation is shown in Fig. 5.1. In order to effect the inverse springback simulation, we consider a blank at a shape defined by $\boldsymbol{u}_{l1}$. We also consider a finite element analysis step where we constrain all the degrees of freedom of the nodes of the blank to remain constant, while simultaneously embedding into the material the internal forces $\boldsymbol{I}_1^* = -\boldsymbol{I}_{l1}$. The presence of the internal forces $\boldsymbol{I}_1^*$ results in the transmission of the external force vector $\boldsymbol{F}_1^* = -\boldsymbol{I}_{l1}$ to the nodes of the blank due to the constrained nodal degrees of freedom. The subsequent release of the nodes forces the blank to deflect to a new equilibrium position $\boldsymbol{u}_{l2}^*$ in a manner opposite from the one taking place during springback unloading, due to the opposite direction of applied loads, see also Fig. 5.2. However, all nodal loads are reduced to zero in the new equilibrium position $\boldsymbol{u}_{l2}$, and therefore the development of high unbalanced bending moments is avoided.

The new deflected shape defined by $\boldsymbol{u}_{l2}^*$ is the first iteration on the designed tooling shape. If the part shape $\boldsymbol{u}_{ul2}^*$, obtained after unloading of $\boldsymbol{u}_{l2}^*$, is within an acceptable part shape error, then $\boldsymbol{u}_{l2}^*$ is the die shape which will produce the correct part shape. Otherwise we can repeat the inverse springback simulation, by loading the blank defined by $\boldsymbol{u}_{l1}$ with the internal force vector $\boldsymbol{I}_{l2}$ obtained by forming the blank with tools defined by $\boldsymbol{u}_{l2}^*$. This procedure can be repeated until the part shape obtained shows negligible part shape error, see also the flow chart of Figure 5.1.

One problem encountered when using this method was the possible development of structural instabilities in the material, due to the application of $-\boldsymbol{I}_{l1}$ to the blank.

The internal force vector $\boldsymbol{I}_{l1}$ contains bending and membrane forces of the blank in a sheet metal forming process. Therefore if any tensile membrane forces are present, the application of $-\boldsymbol{I}_{l1}$ causes the development of compressive membrane forces which may lead to fictitious structural level buckling instabilities. In order to ensure that buckling instabilities due to compressive membrane forces do not develop we decompose $\boldsymbol{I}_{l1}$ to the force vector obtained by the integration of the shell membrane section forces $\boldsymbol{I}_{l1m}$, and the shell section bending moments $\boldsymbol{I}_{l1b}$. As also observed and shown in Chapter 4, the contribution of $\boldsymbol{I}_{l1m}$ is not important to springback phenomena. We therefore considered only the force vector $-\boldsymbol{I}_{l1b}$ in our inverse springback calculations thereby excluding possible development of fictitious instabilities in the blank due to compressive membrane forces.

5.3 An application of the tooling design algorithm. Forming of a two-dimensional channel

The tooling design algorithm developed in the previous section was used for the design of a part with the channel geometry shown in Fig. 4.1. We will now demonstrate how our tooling design algorithm was used in order to design the forming tools that would produce the desired part.

We would like to form a part with a channel geometry. The profile of the desired part shape is depicted in Fig. 5.3. The workpiece material in this case was the 2008-T4 aluminum alloy, also studied in the previous sections. In order to form the desired part we first used a punch and a matching die with dimensions identical to those of the desired part. This process was simulated by using both implicit and explicit finite element analysis. Successive stages of the process are depicted in Fig. 4.6, obtained by the finite element model of Fig. 4.1. We see that the obtained part upon the occurrence of springback exhibits a severe deviation from the desired part shape. For this reason we performed the "spring forward" simulation, as described

in section 5.2, in the framework of the design algorithm depicted in Fig. 5.1. The shape of the sprung forward part was used as the second tooling shape in our tooling design algorithm. This shape is shown in Fig. 5.3 (second cycle). However, this shape involves deflection of the material in both the punch and the binder area.

The computed part shape obtained by using the modified tooling upon the occurrence of springback is shown in Fig. 5.4 (second cycle). We see that the obtained part shape exhibits only a small error , see also Fig. 5.5. However, further improvement is possible by performing a second iteration in our tooling design algorithm. By doing this we obtain the tooling shape shown in Fig. 5.3 (third cycle).

Note that in order to achieve a fully loaded shape of the blank identical to the shape decribed in Fig. 5.3 (third cycle) we must modify the tooling shape in both the punch area and the binder area as also shown in Fig. 5.6(a). The algorithm indicated that changes in depth, punch wall angle and binder angle were all needed to produce the correct part shape.

Successive stages of the simulation using this design are shown in Fig. 5.6. This design was also experimentally verified. New dies were machined from wood based on the numerically designed tooling and were used to perform the already simulated forming operation, as also shown in Fig. 5.6. Note the existence of a binder-wrap step in this case, Fig. 5.6(a), where there is a curvature produced on the sheet during the clamping stage. The obtained part shape is shown in Fig. 5.4 (third cycle). The part shape error, measured in the y-direction is depicted in Fig. 5.5. We see that the part shape error obtained by simulation and that obtained by experiment is confined to a very small amount. We also see the excellent agreement between experiment and simulation results. Also, in Fig. 5.7 we show the value of the RMS of the part shape error in the different cycles of the algorithm. We see that a substantial decrease is already achieved by the second cycle of the algorithm and that our tooling design

algorithm resulted in a 92% reduction of the RMS of the part shape error from the first to the third cycle.

5.4 Forming of a three-dimensional part

We consider now the forming process of the part depicted in Fig. 5.8. This part possesses a three-dimensional geometry. Its section by the plane $Z = 0$ has the characteristics of a two-dimensional channel, see Fig. 5.9 (a), whereas the section of the shape by the plane $X = 0$ is shown in Fig. 5.9 (b).

The FE model of this part and the tooling set-up is shown in Fig. 5.10, where only a quarter of the model is depicted. Due to the large size of the problem and to the complicated geometry of the tools, we modeled the loading portion of the forming operation by using dynamic explicit analysis only.

We note here that after preliminary dynamic explicit finite element simulations of the process we concluded that we had to develop a divided binder, see Fig. 5.10, in order to prevent excessive draw-in at the edge of the blank , (section E-F of Fig. 5.8). A normal binder force of 1 KN was applied at the center segment of the binder whereas a normal binder force of 10 KN was applied at the edge segment of the binder. A friction coefficient of 0.4 was assumed for the interaction between the binder and the blank.

In the first cycle of our tooling design procedure, the tooling shape matched the desired part shape. The obtained part shape, upon the occurrence of springback possessed a certain shape error, see also Figs. 5.11 - 5.14. In order to reduce the part shape error, we modified the die shape following the algorithm of Fig. 5.1. Sections of the new die shape are depicted in Figs. 5.15 - 5.18. Note the changes in the binder area depicted in Figs. 5.15 - 5.18. As also in the case of the channel geometry, the tooling design compensates for part shape error due to springback by changing the tooling shape in both the punch area and the binder area. Sections of the part

produced with the corrected tooling shape upon the occurrence of springback, as obtained by finite element analysis, are shown in Figs. 5.11 - 5.14. We see that the produced part is nearly identical to the desired part shape. Therefore, our tooling design algorithm has converged to the desired part shape in only one iteration.

An experimental verification of this tooling design was also performed by effecting the forming operation of the 2008-T4 aluminum blank with the designed forming tools. The tooling shape obtained from the finite element analysis, was first offset and interpolated and then transformed to G-code for CNC machining. The shape offset prior to the CNC machining was equal to the machining tool radius + $\frac{1}{2}$ thickness of the blank. The source code of the program used for interpolation and offset of the surface obtained from the finite element analysis can be found in Appendix V. The punch, die and binder were built by using this G-code. The machined tools are shown in Fig. 5.19. We used the 6061-T4 aluminum alloy as tooling material. Although the 6061 aluminum is not hard enough for standard tooling production, its machinability and moderate hardness render it very appropriate for the rapid CNC machining of tooling subsequently used for the stamping of a few prototypes of a soft material (2008-T4 aluminum), Walczyk (1993). The binder was segmented as established from the finite element analysis. The binder force in the different segments was applied by using vertical threaded rods, see also Fig. 5.19. The vertical force applied to each segment was obtained by torquing the supportive bolts. The punch forming motion was obtained by using an INSTRON 8501 tension-compression machine. Teflon layers were placed between the blank and the die and between the blank and the punch in the forming area in order to obtain the friction coefficient of 0.03 used in the simulation. No lubricant was used in the binder area in order to achieve the limited amount of draw-in obtained in the simulation with a total binder force of 11KN and a friction coefficient of 0.4 (estimated).

The produced part shape was measured in a Brown-Sharpe Coordinate measuring machine. The obtained section shapes are depicted in Figs. 5.11-5.14. We see that the experimentally obtained parts are in very good agreement with the theoretical analysis, see Figs. 5.11-5.14. The slightly higher amount of springback observed in our experimemt is possibly due to the higher amount of draw-in obtained in the experiment (13.5 mm at point A) compared with the draw-in obtained in the simulation (13.2 mm at point A). However the experimentally obtained part shape is in good agreement with the desired part shape.

5.5 Conclusions

The background developed from the springback calculations in Chapter 4 was used in this Chapter in the development of a tooling design algorithm which compensates for part shape error due to springback by modifying the tooling shape. In this algorithm the internal force distribution on the part, obtained by finite element analysis, is used in order to perform an inverse simulation of springback of the fully loaded part. In other terms the fully loaded part, whose shape matches the desired part shape, is sprung forward thereby reversing the springback phenomenon. The sprung-forward shape is obtained to be the next tooling shape in an iterative approach where the procedure is repeated until the desired part is obtained. We note that this tooling design algorithm required shape changes of the tools in both the punch area and the binder area.

We used the tooling obtained by our design algorithm in order to produce two aluminum parts. The designed tooling was found to produce the desired part shape with very small error. The designs were also verified experimentally, where excellent agreement between simulation prediction and experimental results was obtained.

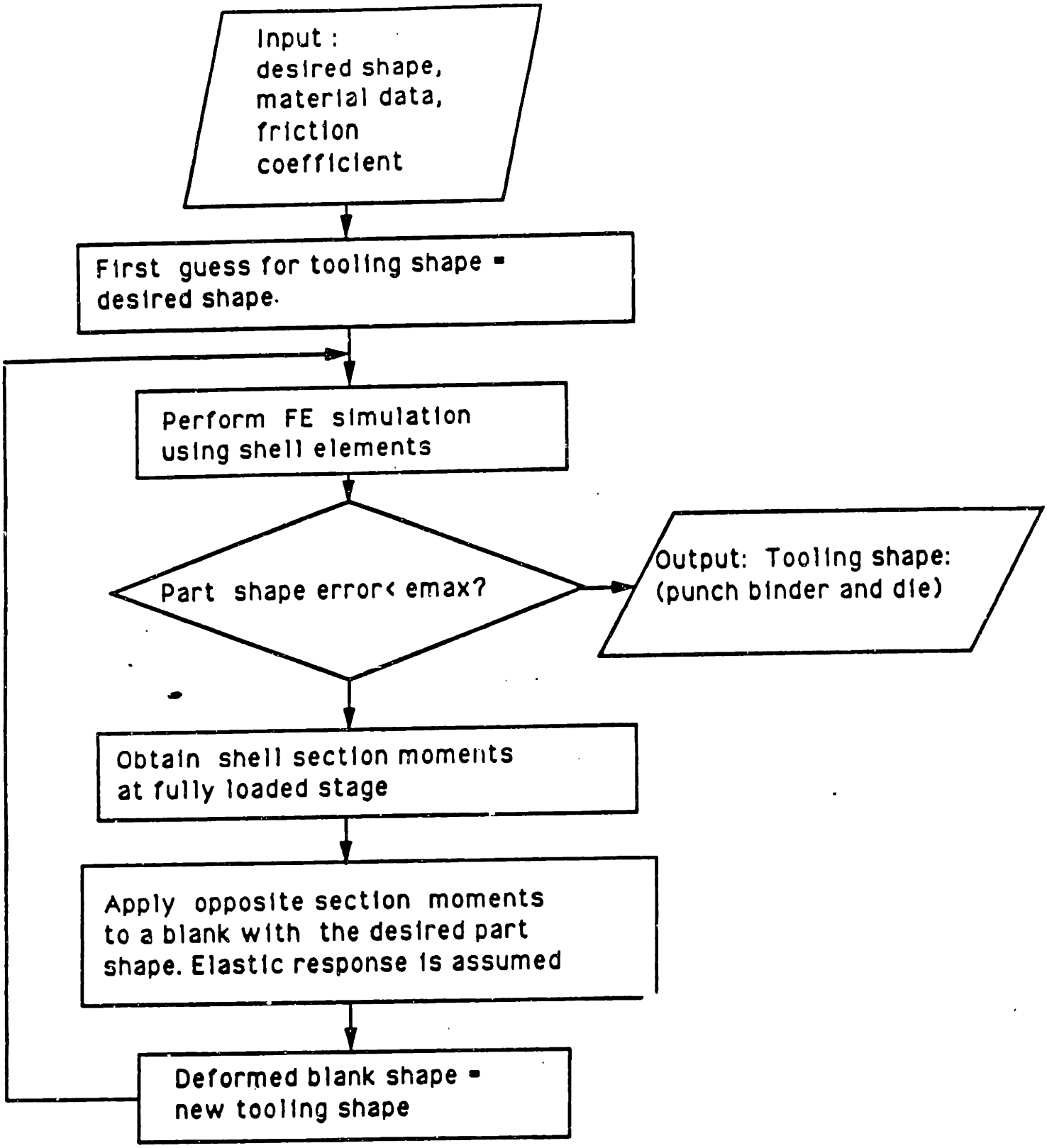
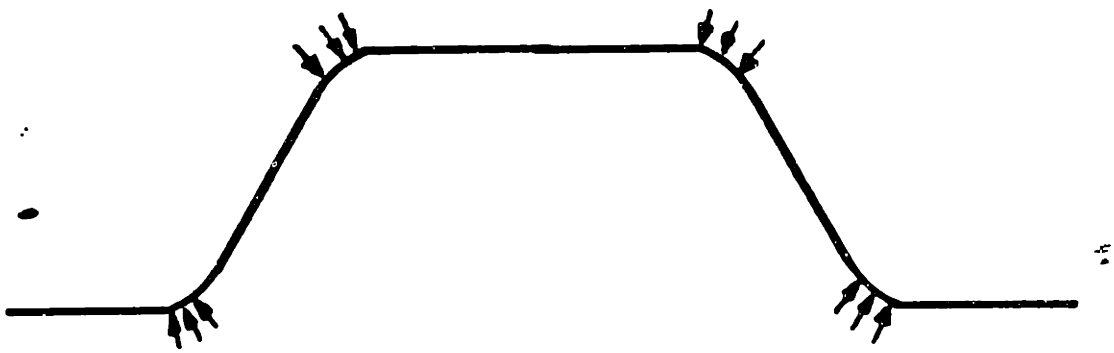


Fig. 5.1: Flowchart of the tooling design algorithm.

Embed opposite section moment distribution at the desired shape



Consistent traction distribution at the fully loaded stage

Part springs forward upon unloading

Fig. 5.2: Schematic graph of the springforward simulations.

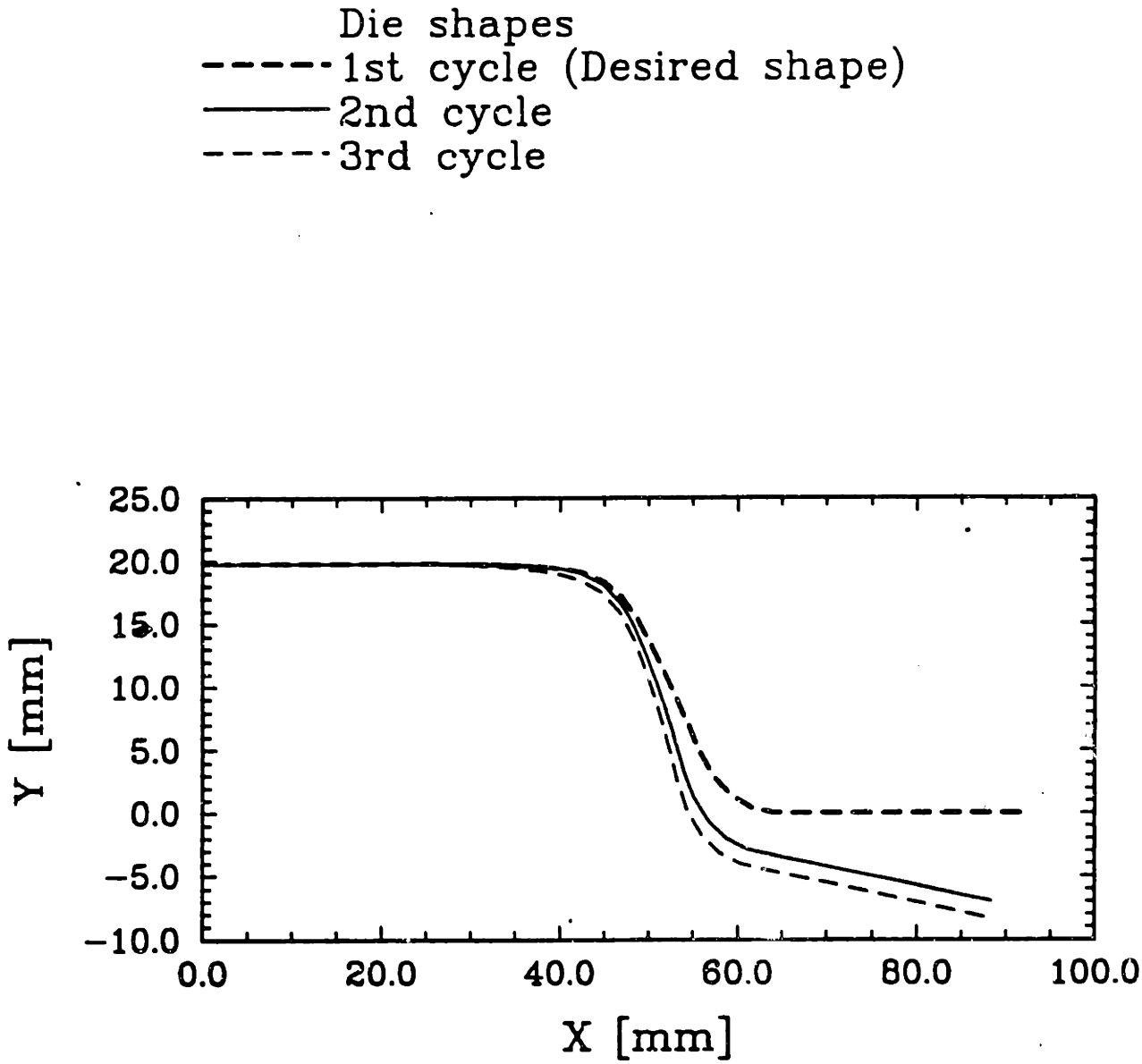


Fig. 5.3: Tool shapes used in the different cycles of the tooling design algorithm.

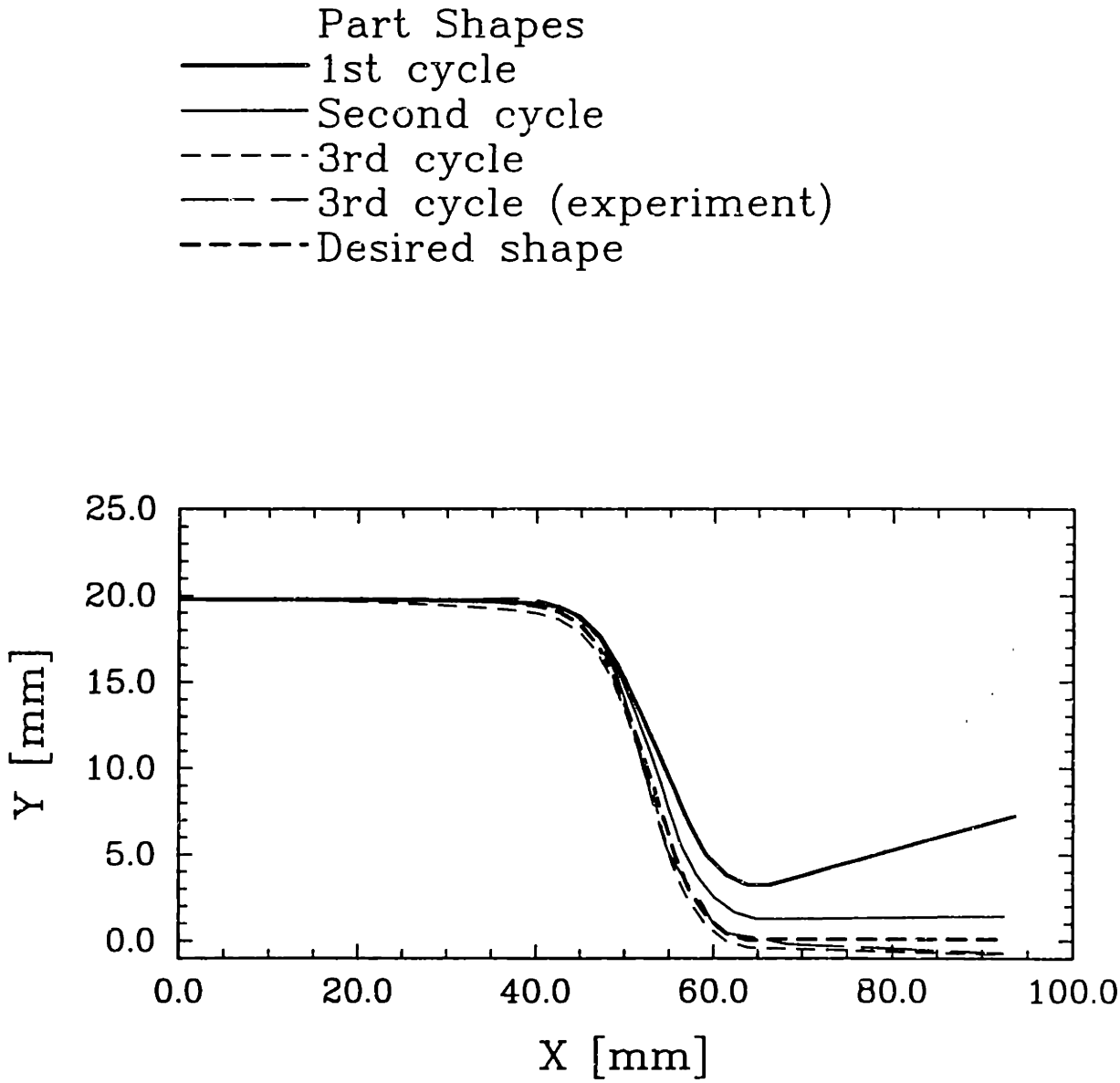


Fig. 5.4: Produced part shapes form the different cycles of the tooling design algorithm.

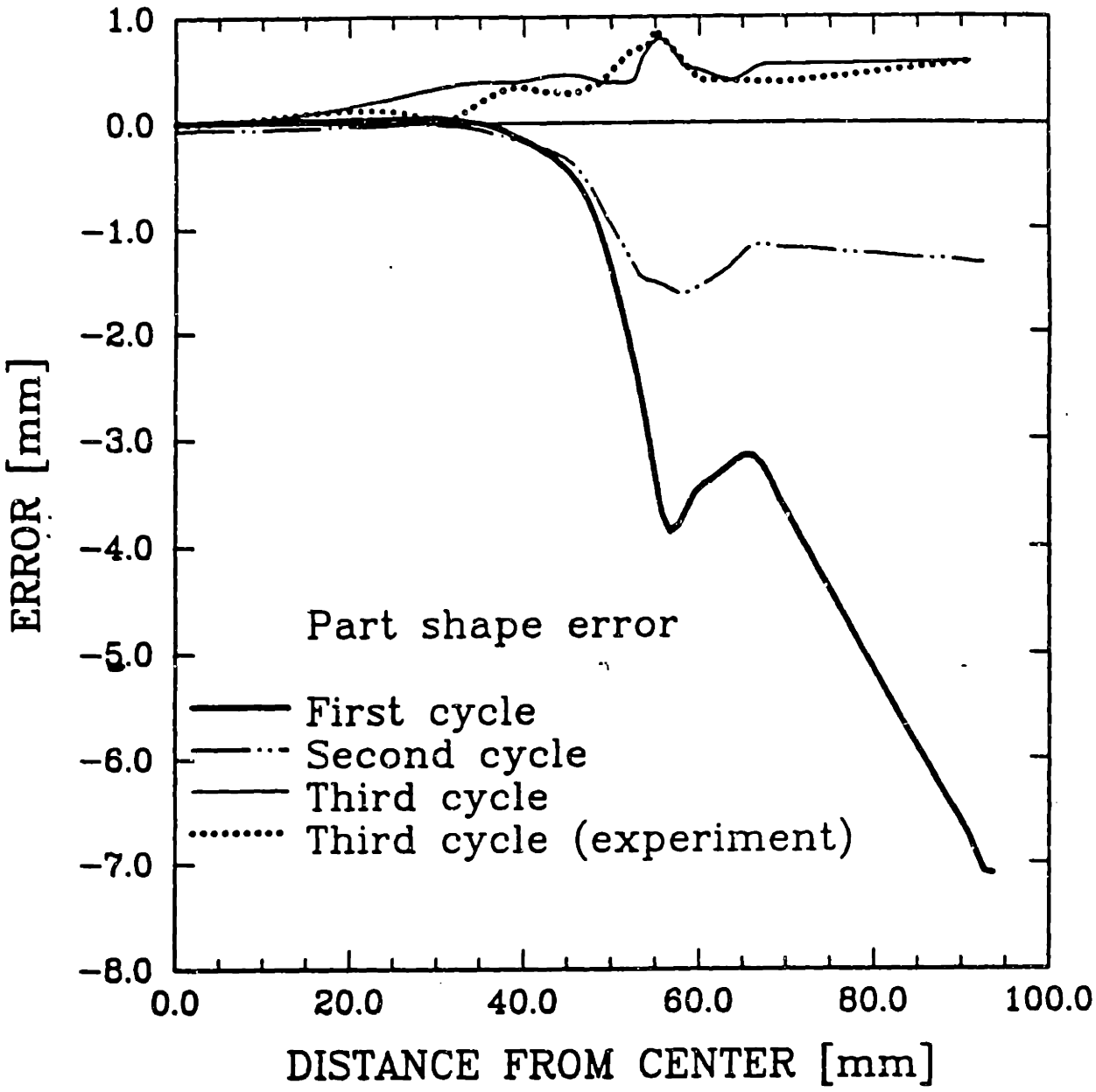
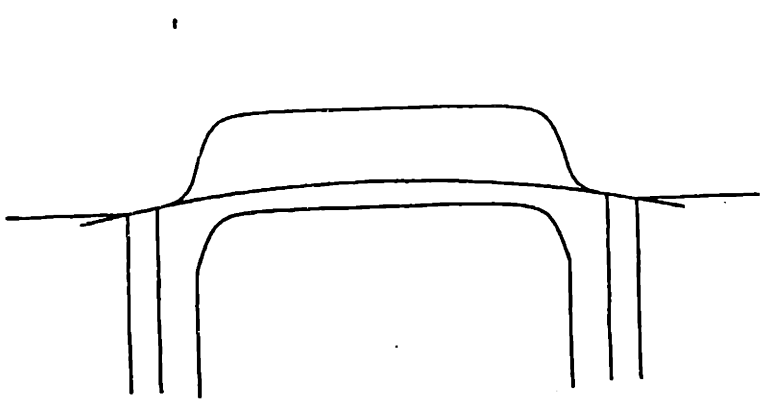
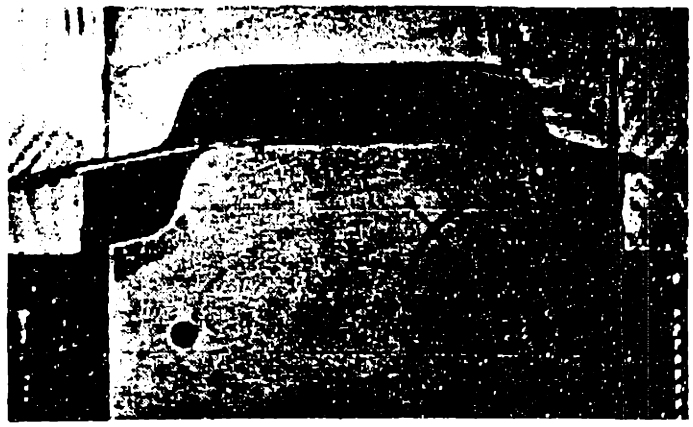
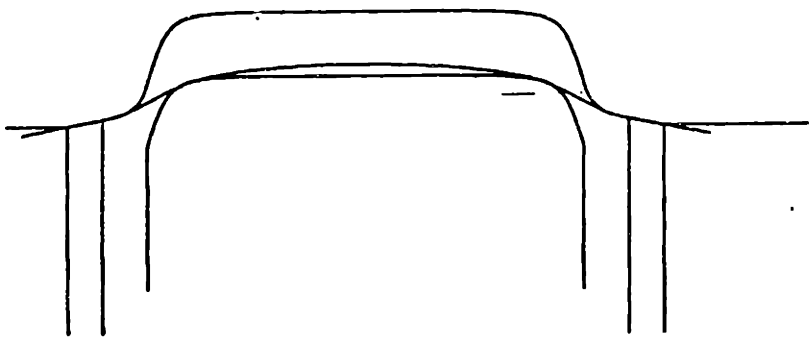


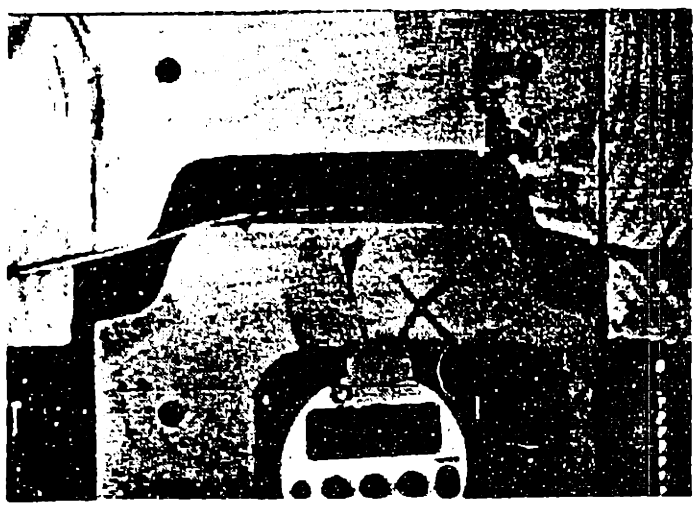
Fig. 5.5: Part shape error in the vertical direction for the different cycles of the tooling design algorithm.



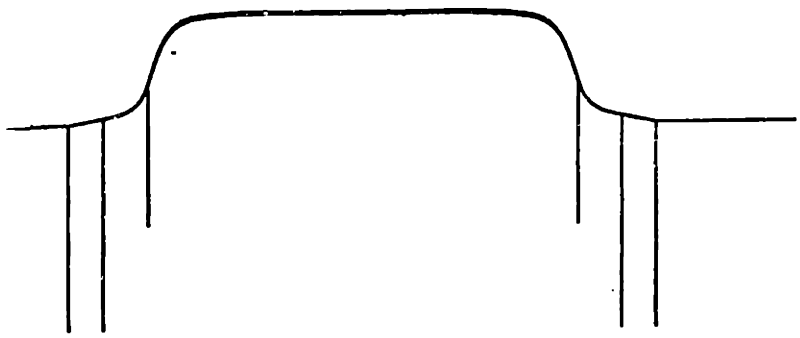


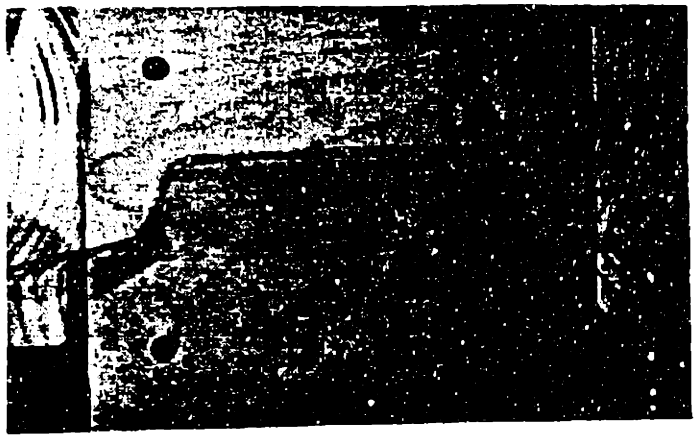
(a)



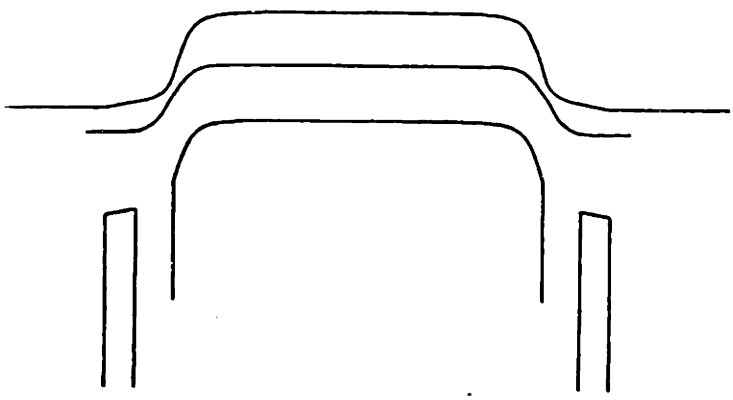


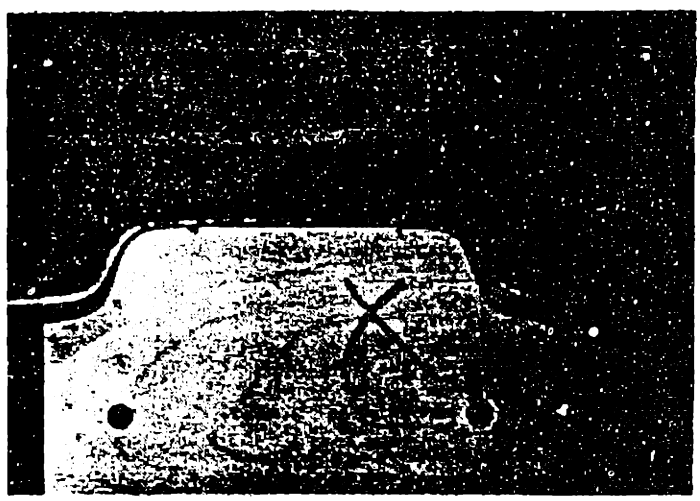
(b)





(c)





(d)

Simulation

Experiment

Fig. 5.6 Side view of successive steps of the channel forming process with the tools obtained from the tooling design algorithm.

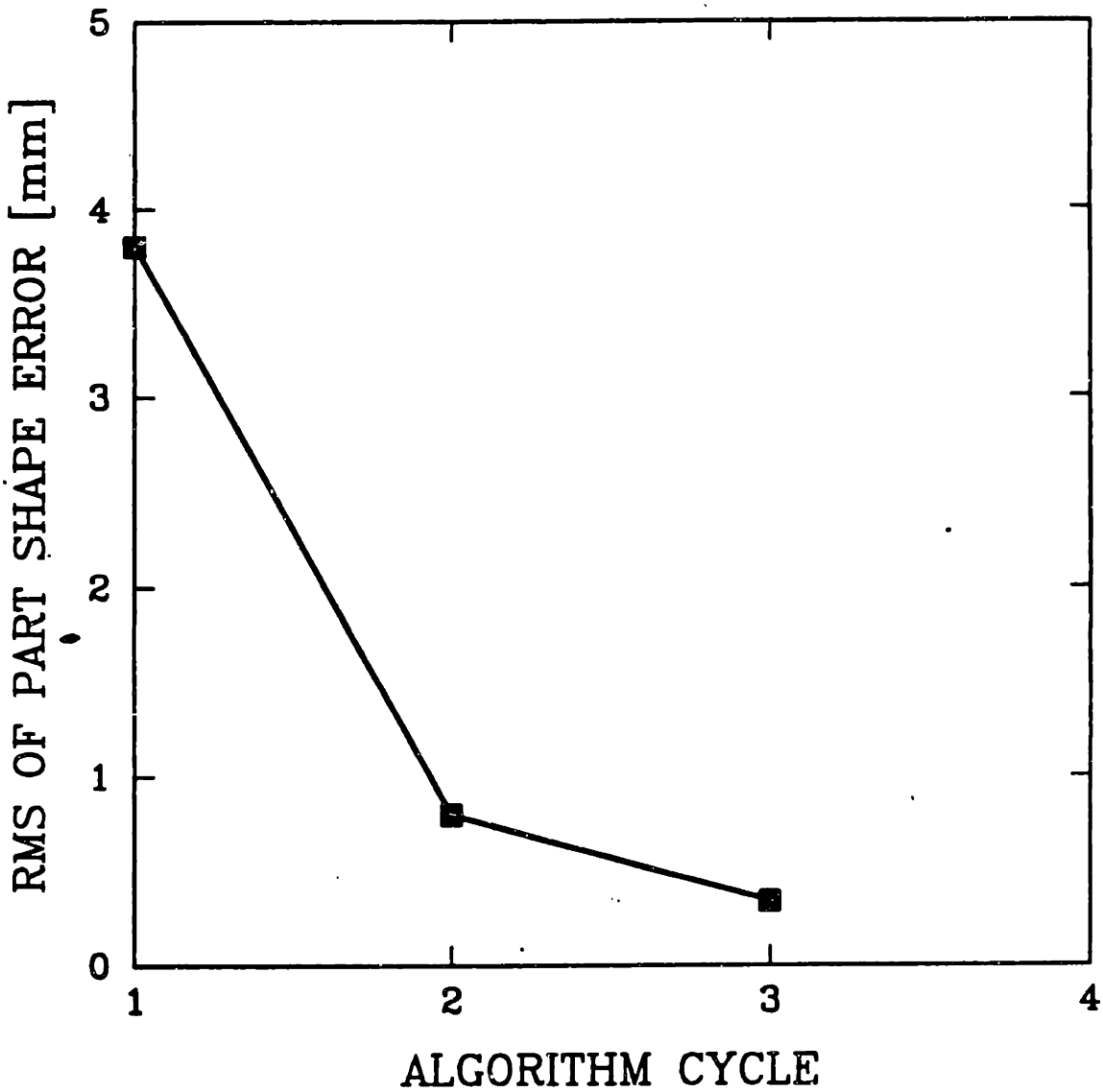


Fig. 5.7: RMS of the part shape error for the different cylcles of the tooling design algorithm.

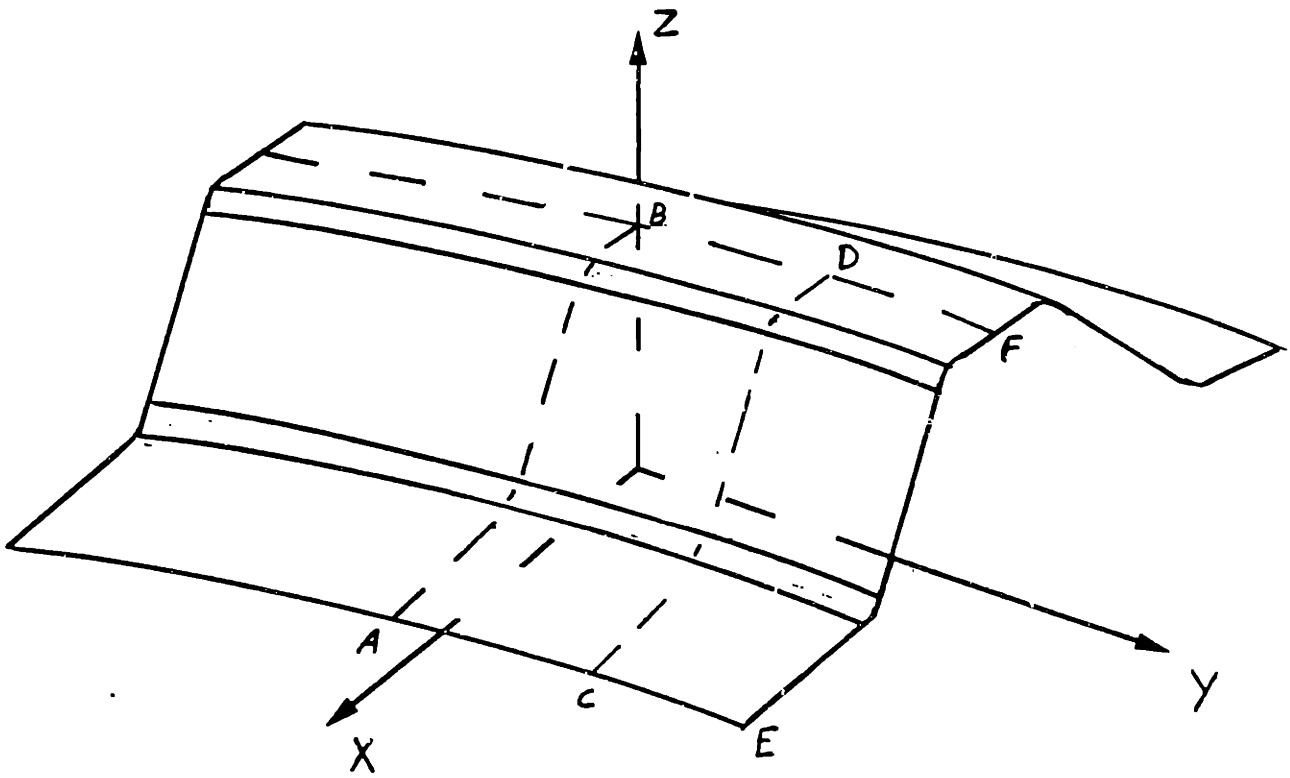
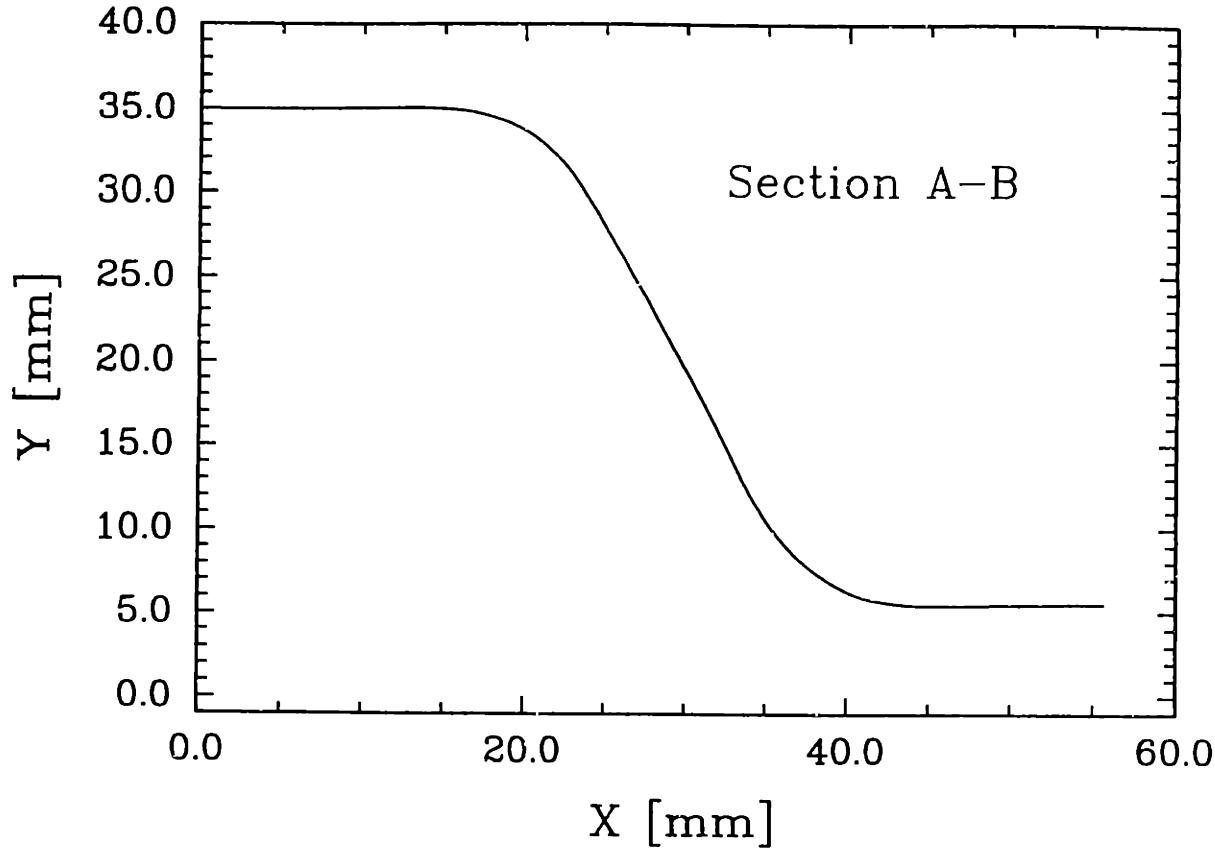


Fig. 5.8: The three-dimensional channel geometry.



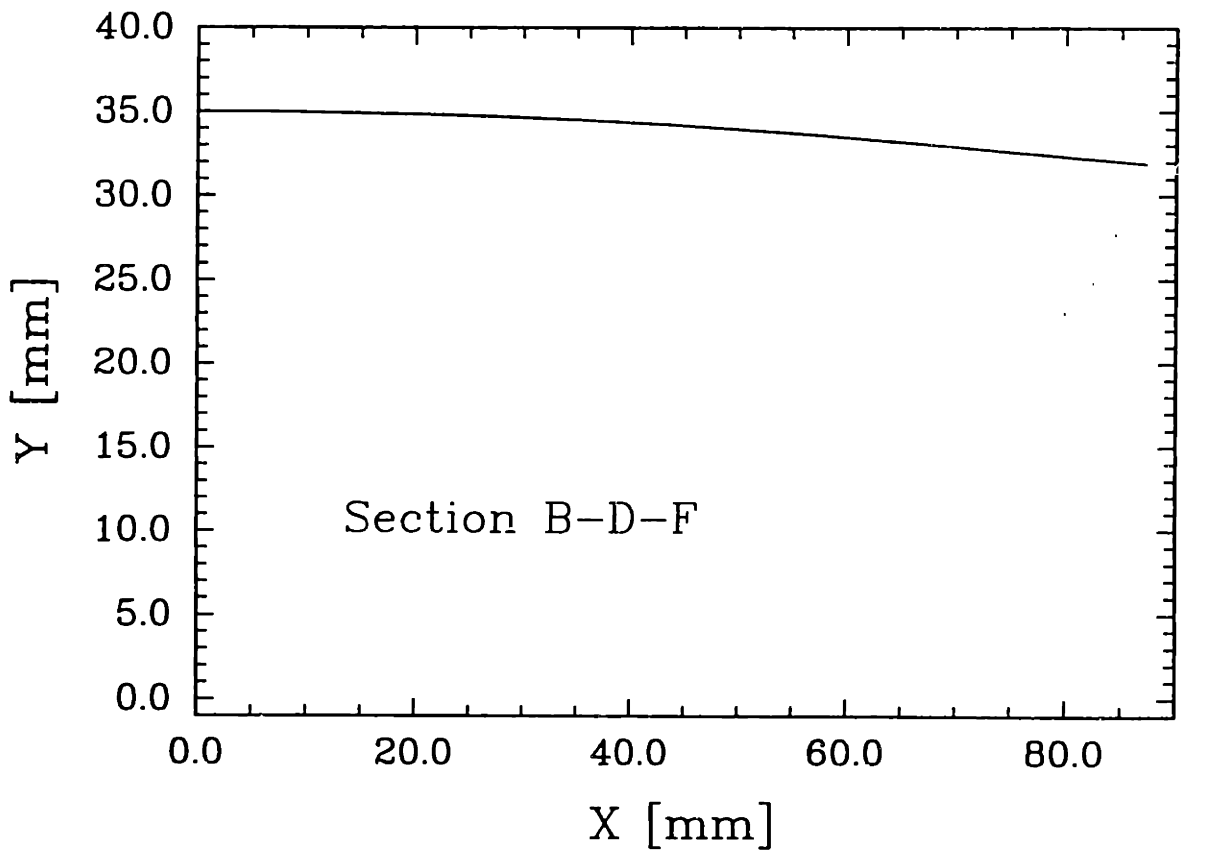


Fig. 5.9: Different sections of the three-dimensional geometry.

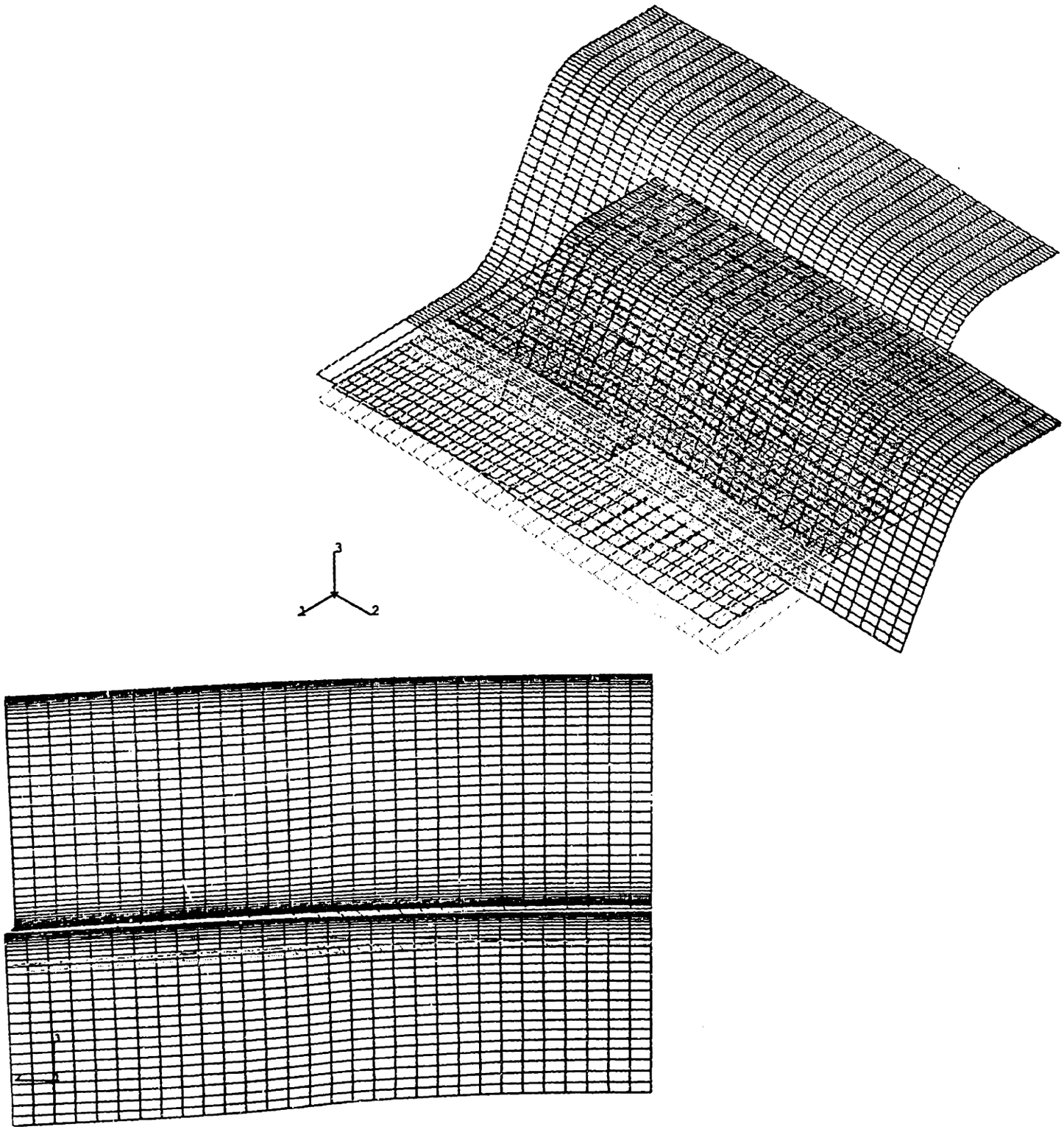


Fig. 5.10: Finite element model of the part.

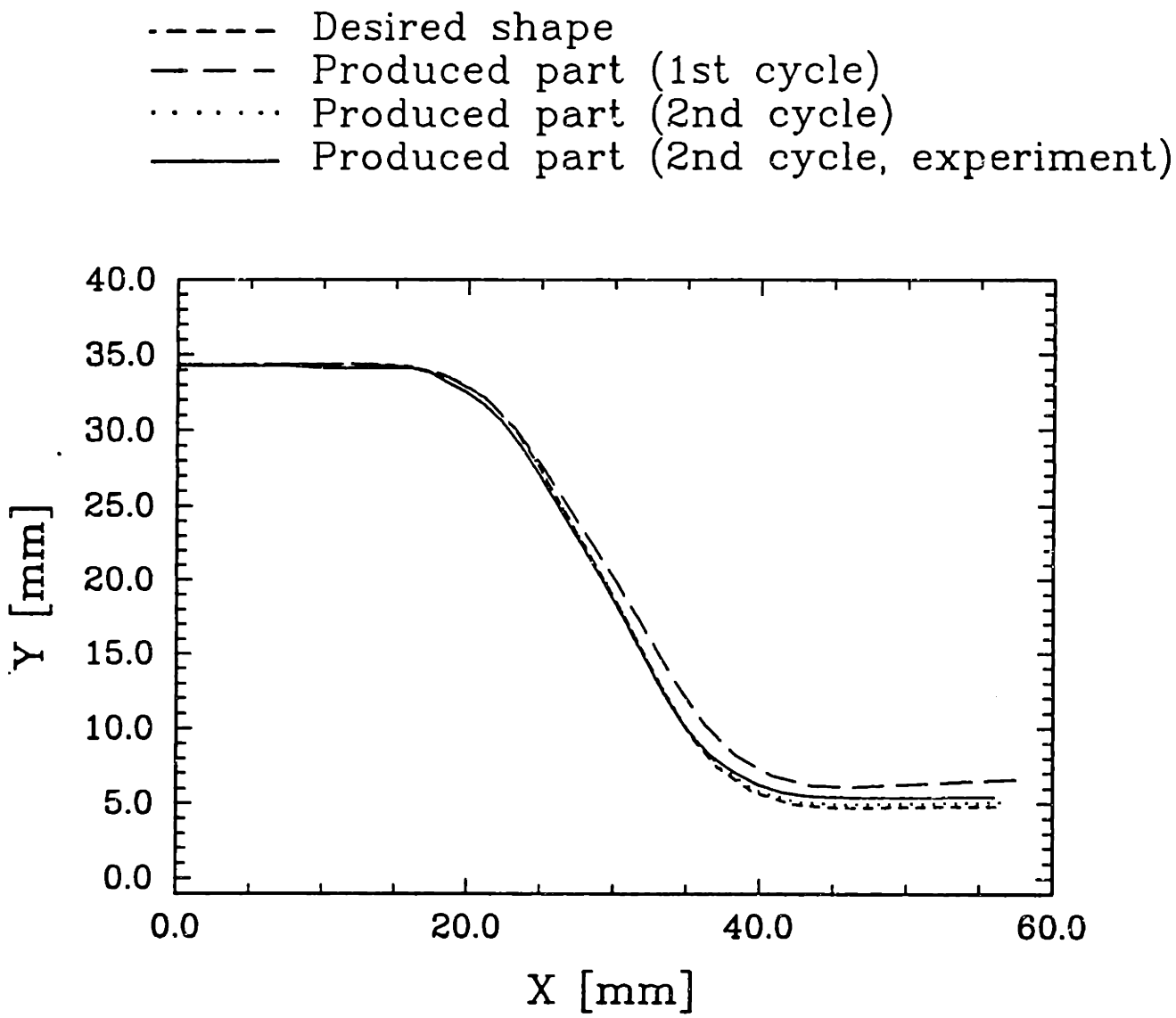


Fig. 5.11: Part shapes at section A-B for different cycles of the tooling design algorithm.

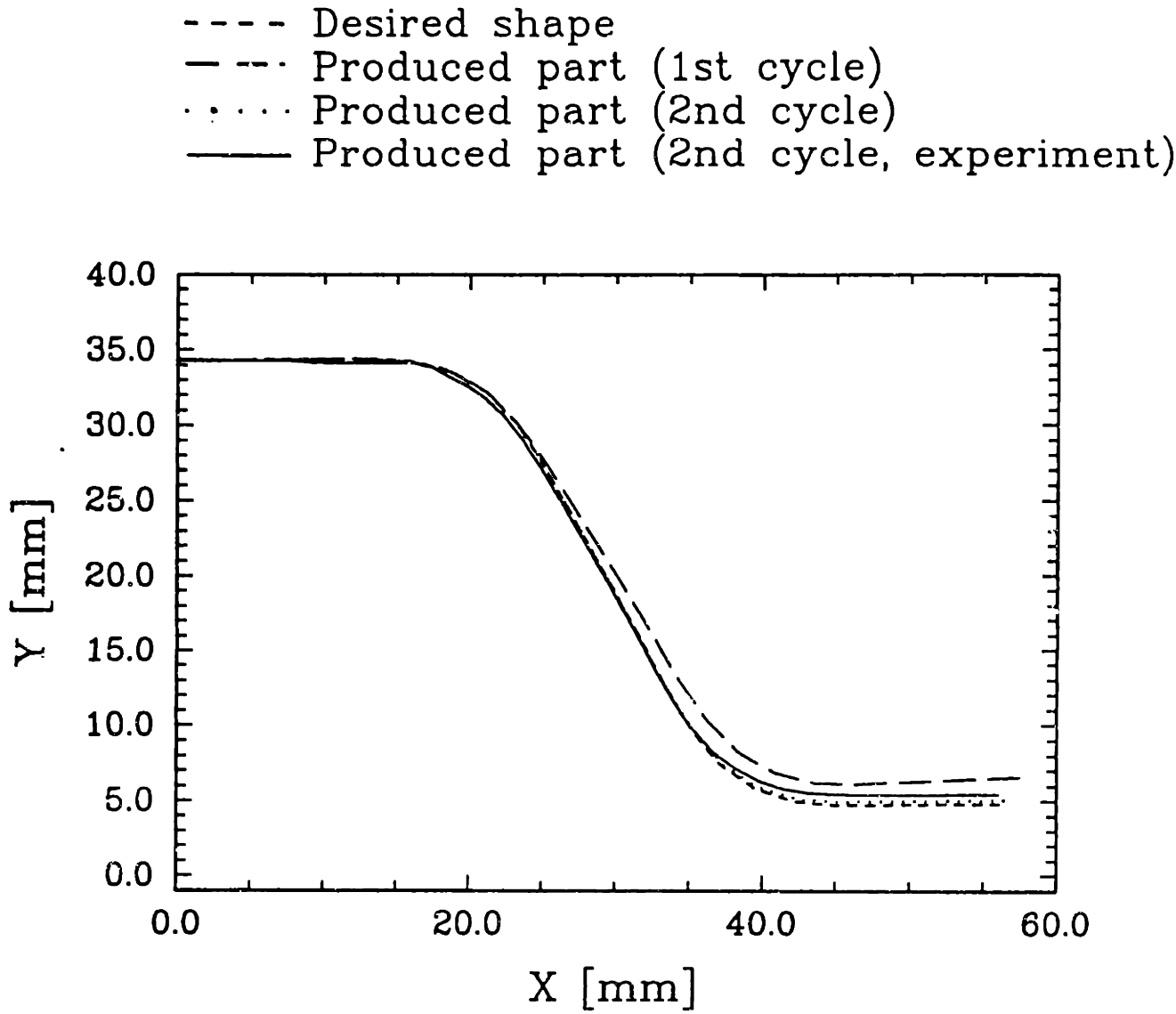


Fig. 5.12: Part shapes at section C-D for different cycles of the tooling design algorithm.

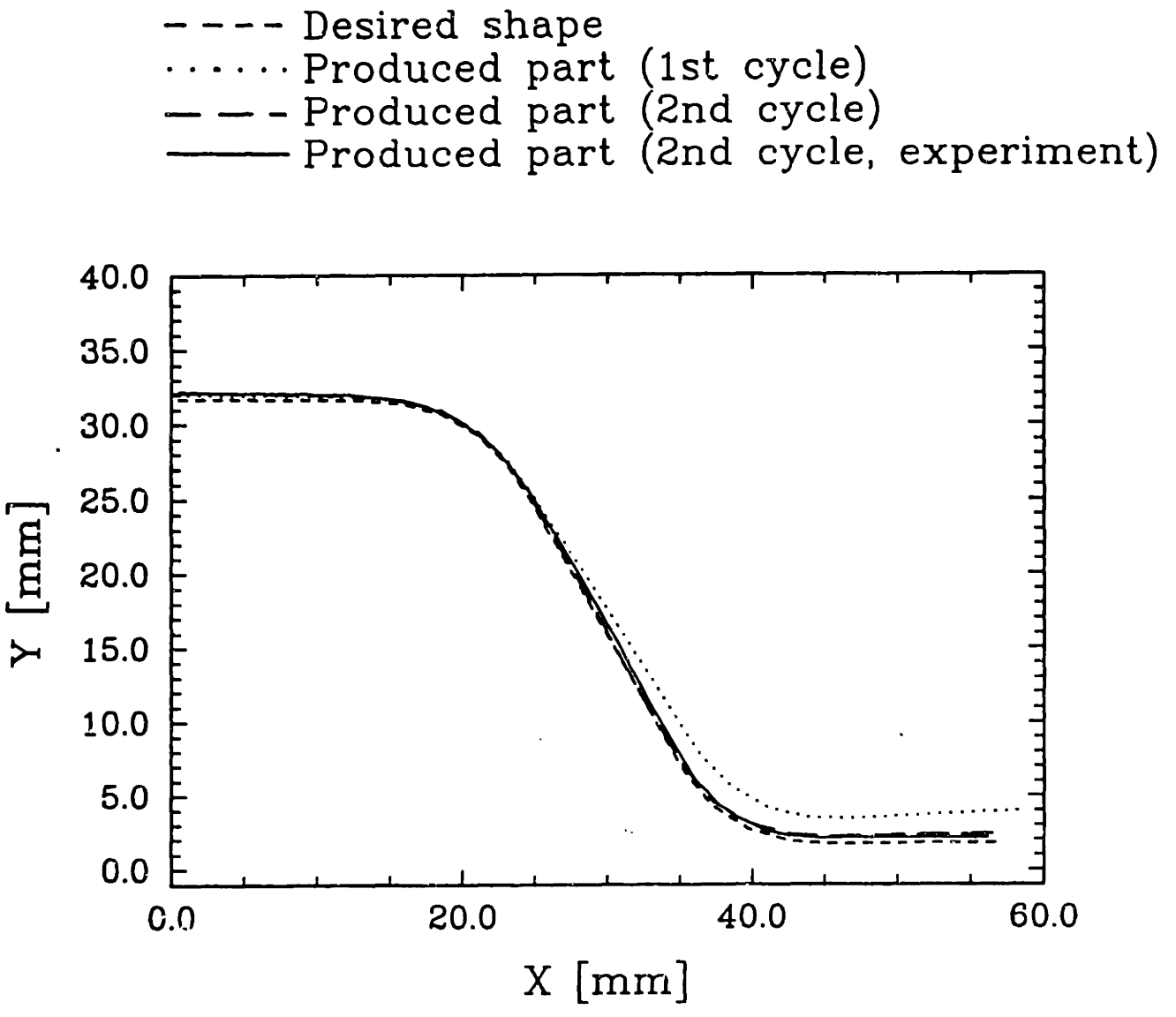


Fig. 5.13: Part shapes at section E-F for different cycles of the tooling design algorithm.

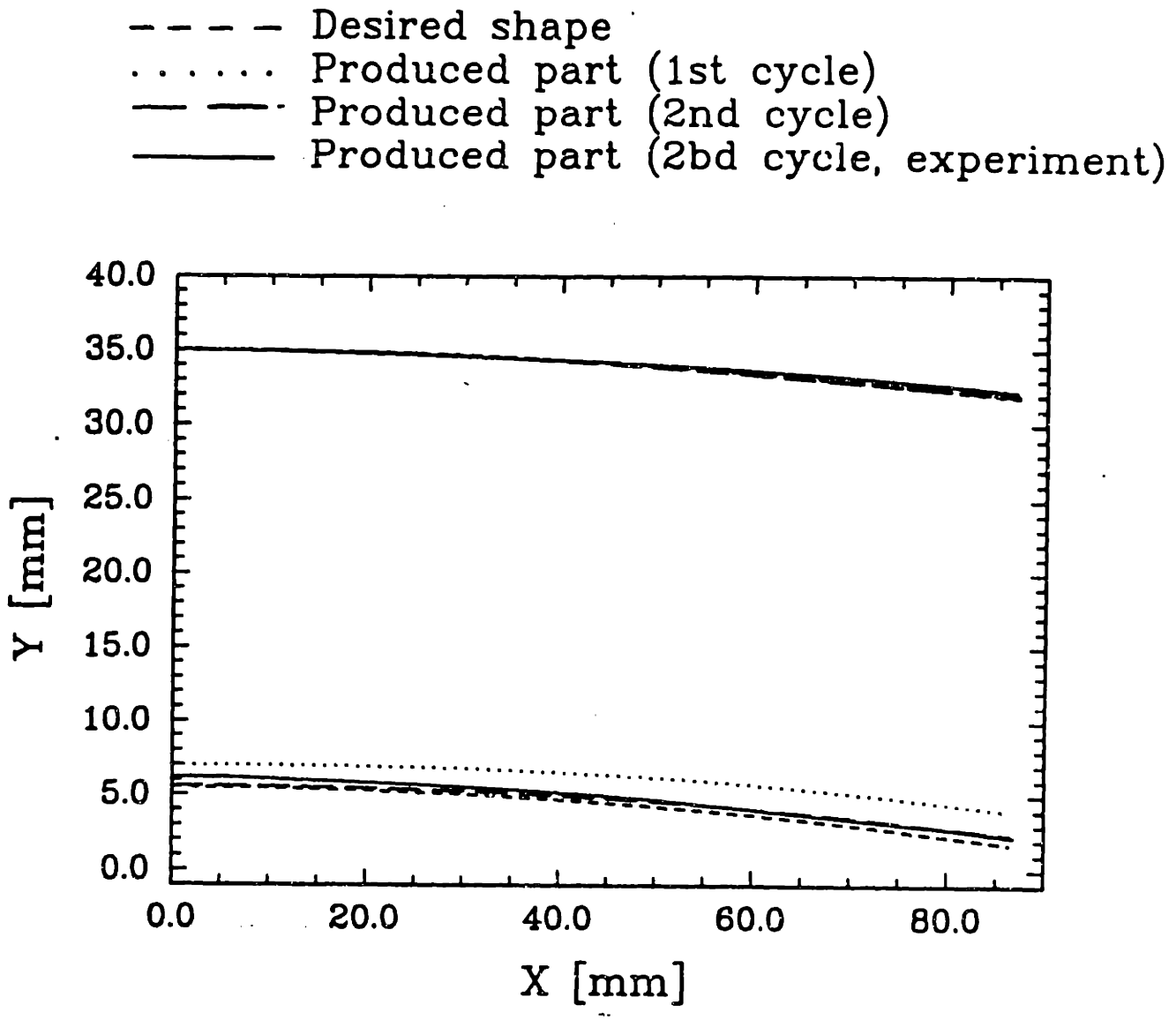


Fig. 5.14: Part shapes at section A-C-E and B-D-F for different cycles of the tooling design algorithm.

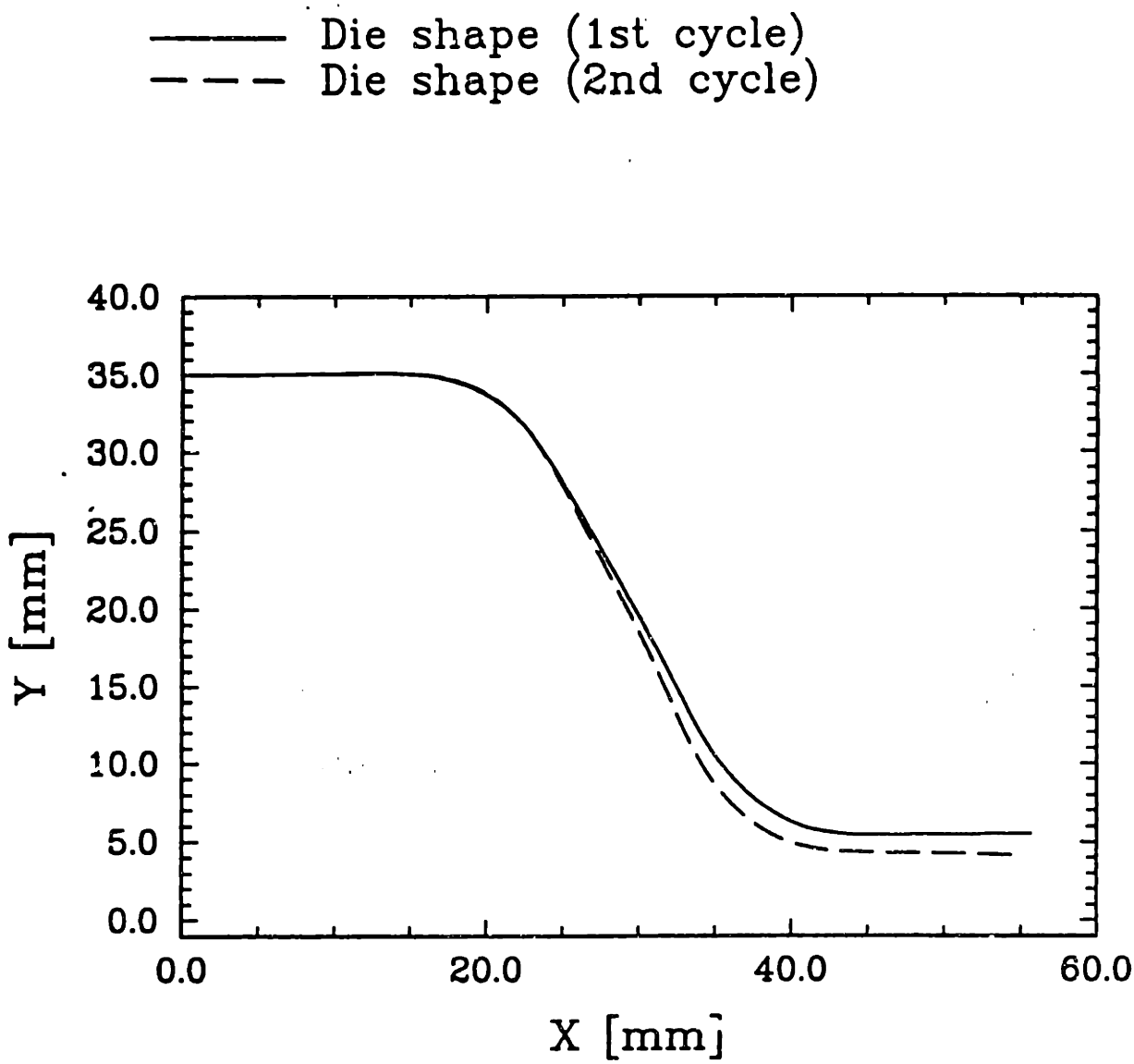


Fig. 5.15: Die shapes at section A-B for different cycles of the tooling design algo-

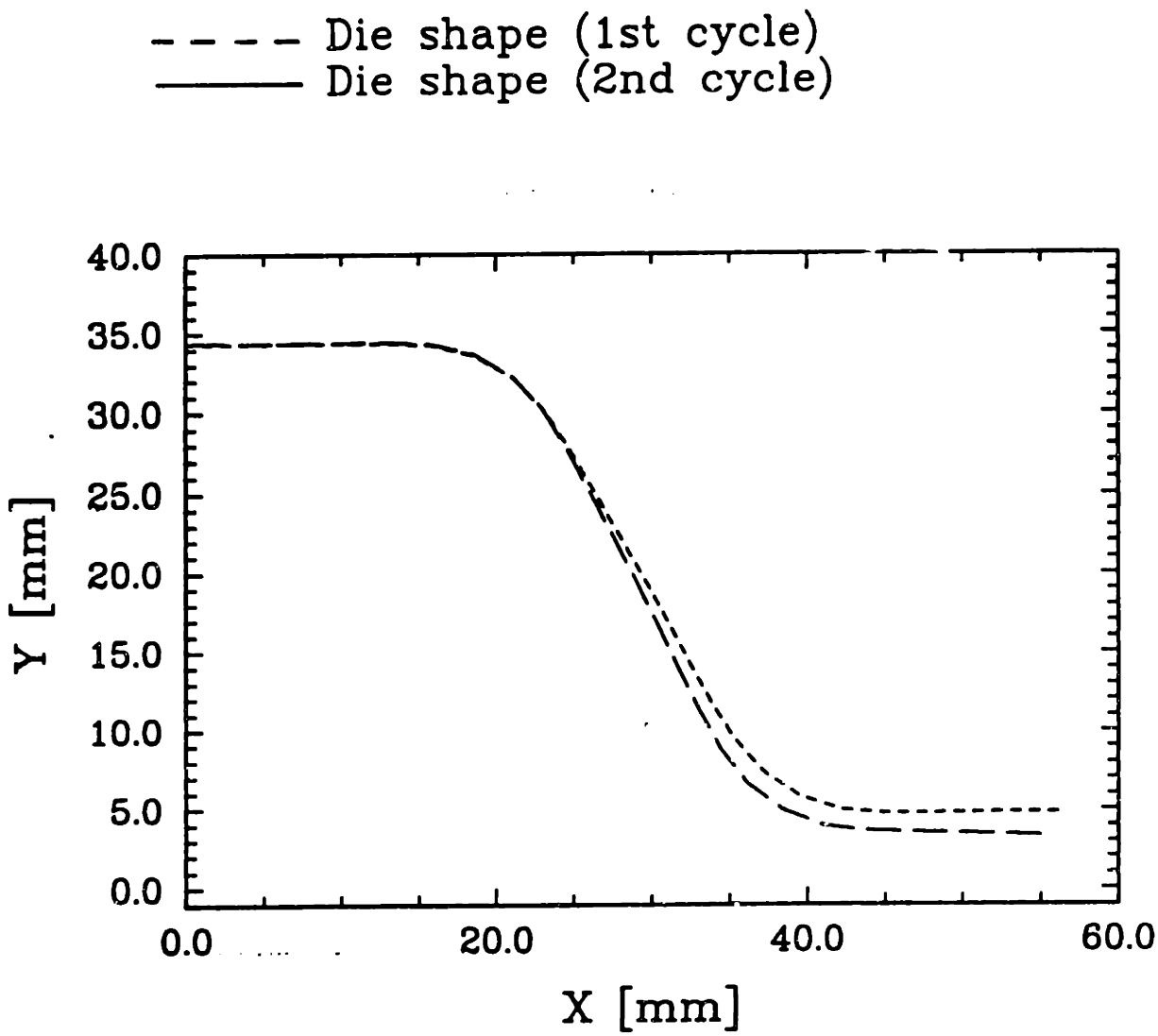


Fig. 5.16: Die shapes at section C-D for different cycles of the tooling design algorithm.

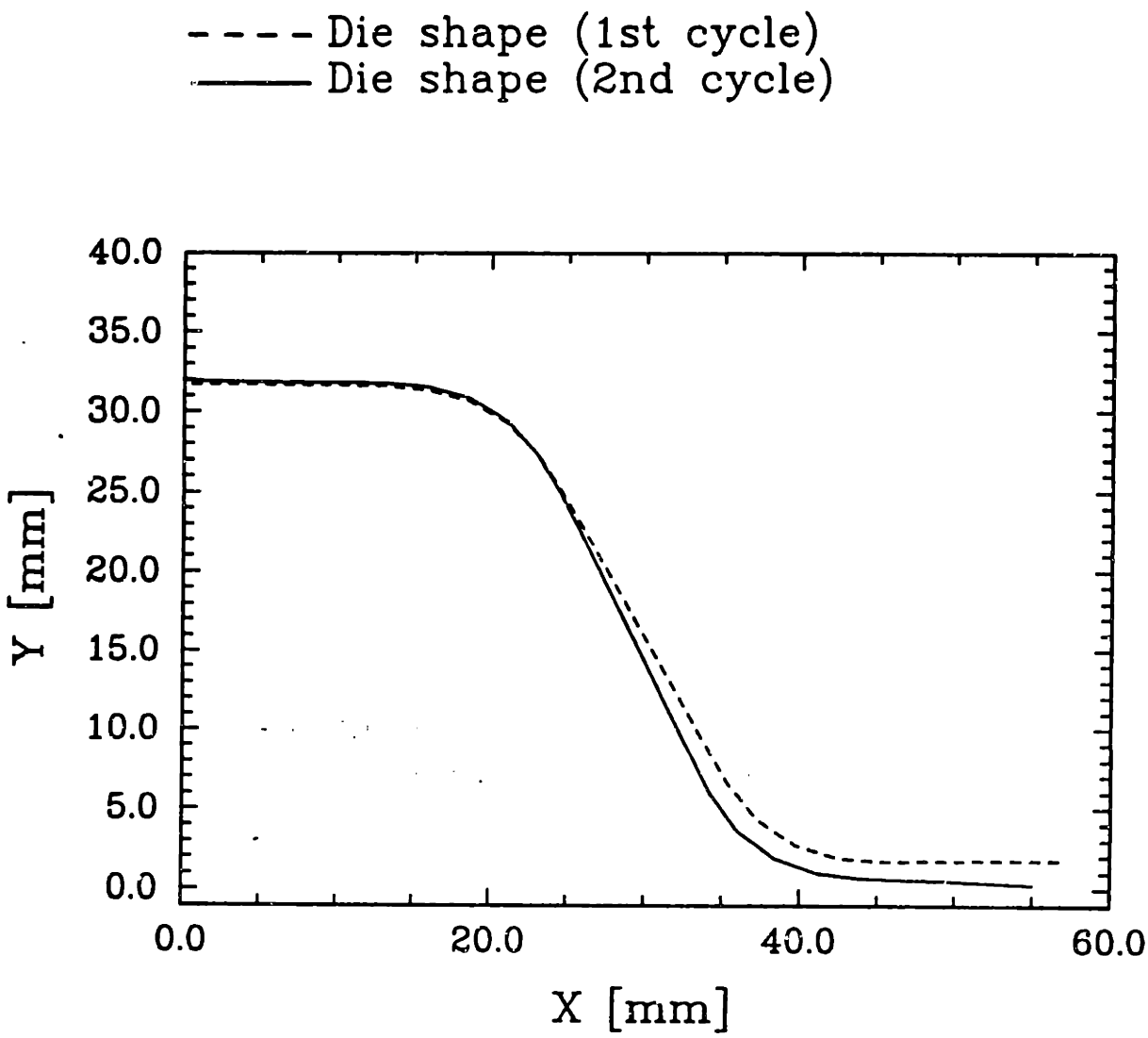


Fig. 5.17: Die shapes at section E-F for different cycles of the tooling design algorithm.

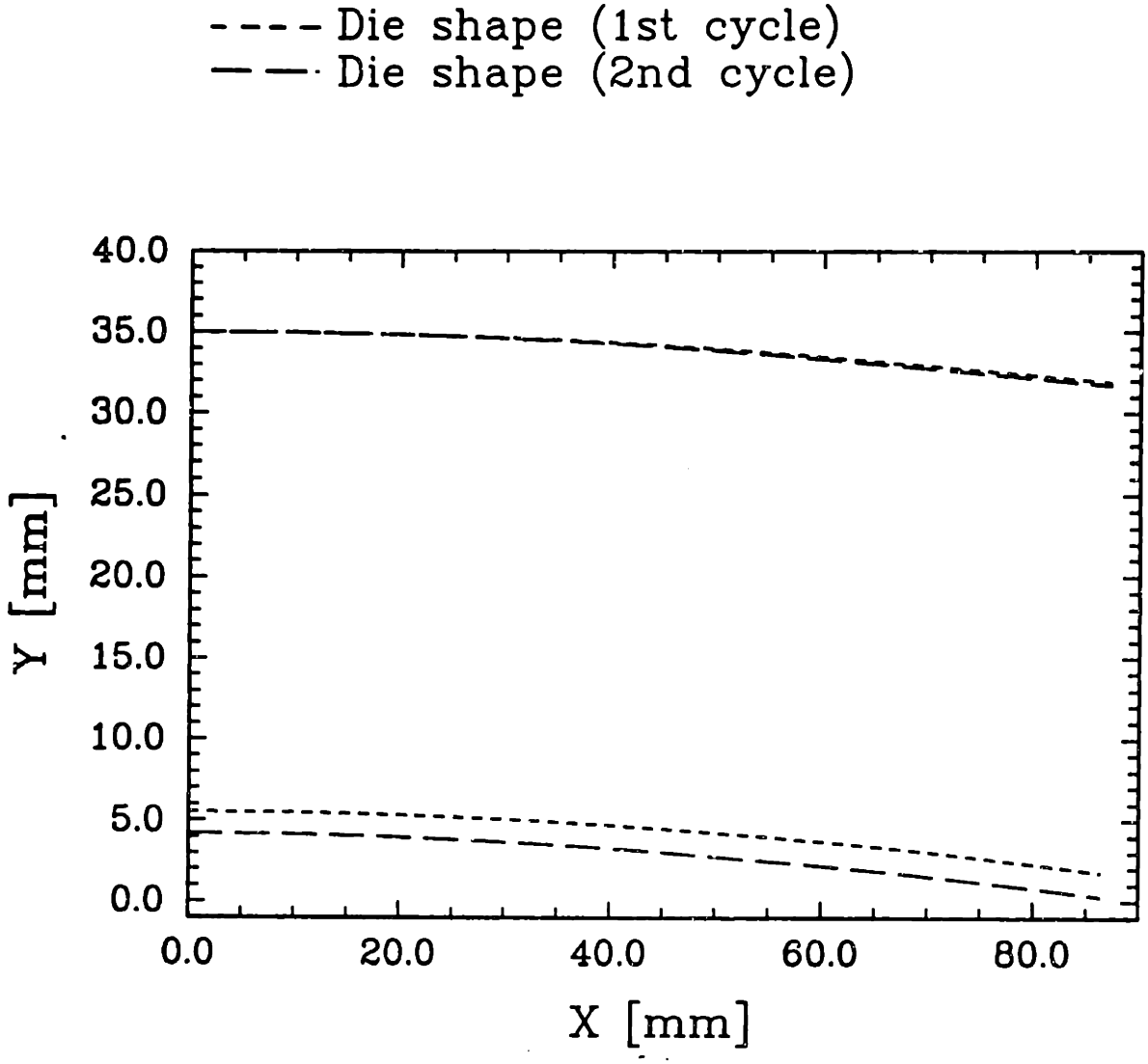
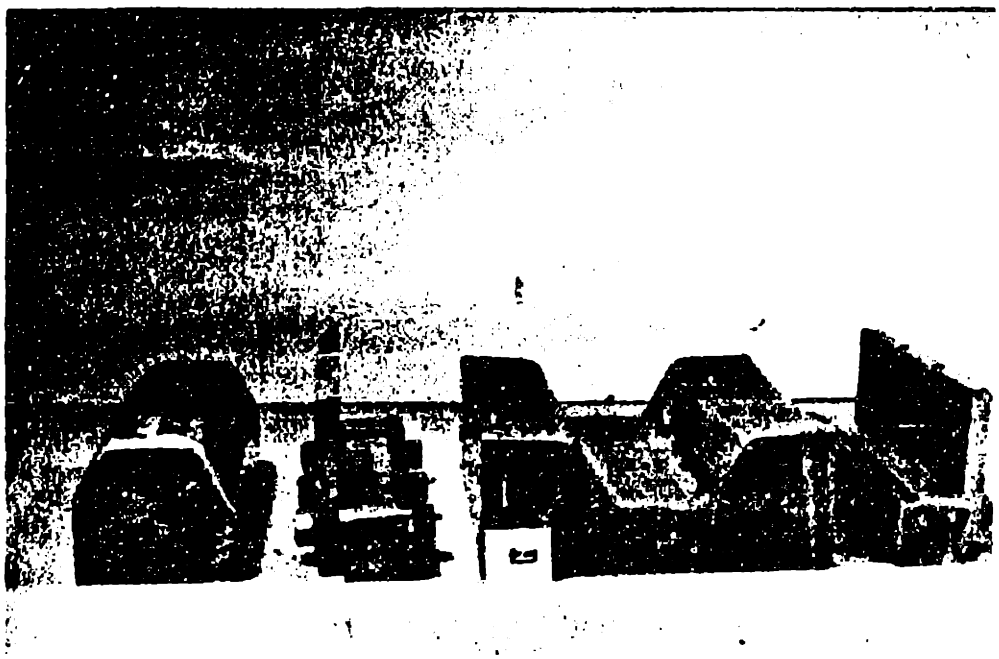
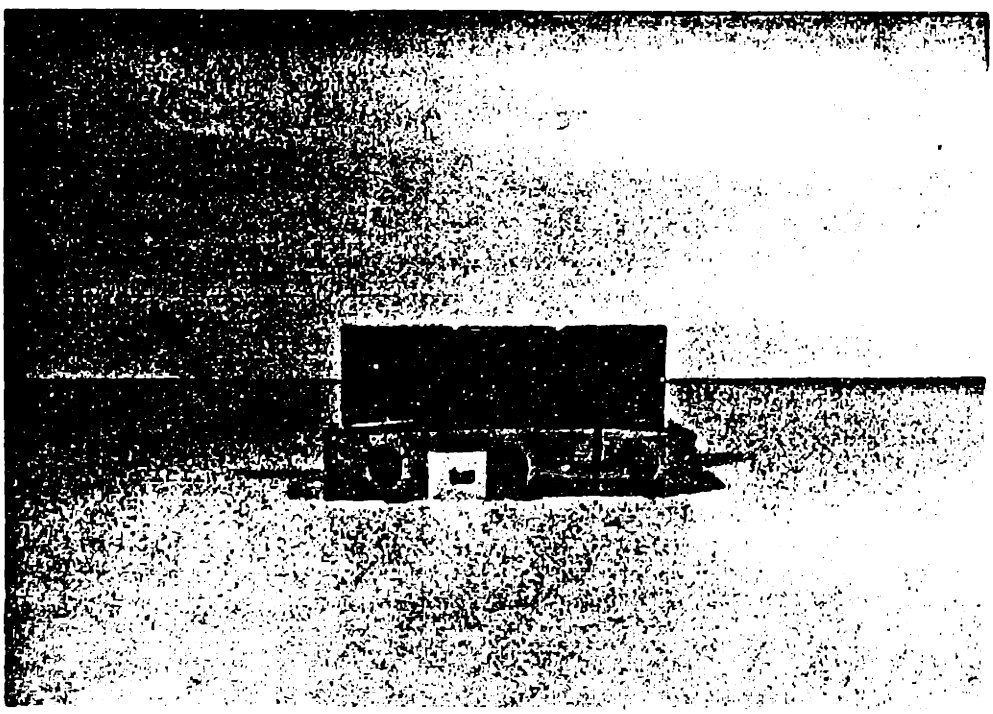


Fig. 5.18: Die shapes at section A-C-E and B-D-F for different cycles of the tooling design algorithm.





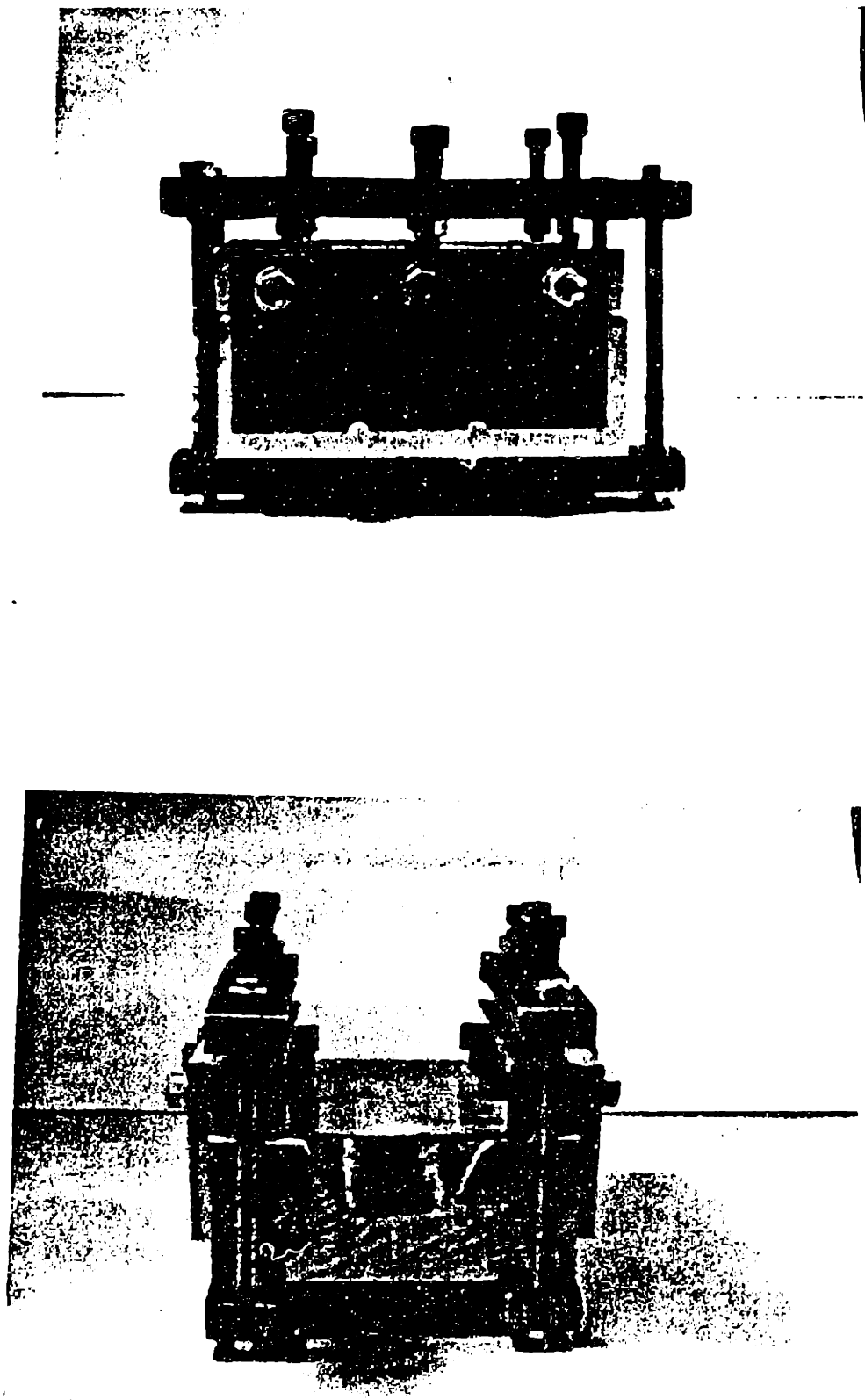


Fig. 5.19: Tooling set-up fot the three-dimensional channel forming.

6 CHAPTER 6: CONCLUSIONS - FUTURE WORK

This thesis work was motivated by the need to solve the manufacturing problem of part shape error due to springback in sheet metal forming processes. Efforts to deal with this manufacturing problem had already been developed at MIT by Webb [1987], Ousterhout, [1991], and Karafillis [1992]. In the work of Webb [1987] and Ousterhout [1991] tooling design algorithms were developed and implemented by using tooling design algorithms in combination with hardware trials. Karafillis [1992] developed and implemented tooling design algorithm in combination with Finite Element Analysis for two-dimensional processes. In the present thesis, a computational tool was developed for the modeling of three-dimensional sheet metal forming processes. This computational tool was then implemented in the tooling design of the forming process compensating for part shape errors due to springback. This design was effected in a systematic manner, by using a design algorithm developed in this thesis work.

A large part of the effort for the development of the computational tool for the modeling of three-dimensional sheet forming processes was devoted to the development of a constitutive models to describe the elastic plastic behavior and the anisotropic plastic behavior of rolled metals. A general anisotropic yield function was developed in order to describe the plastic anisotropy of metals. The essential idea of this theory is the mapping from the stress space of the anisotropic material to the stress space of a new entity called the Isotropic Plasticity Equivalent (IPE) material. Plasticity calculations (flow rule) are first done in the IPE material and then mapped back to the original anisotropic material. The developed material model was found to correctly predict the distribution of the yield stress in different directions given the distribution of the R-ratio in different directions. Another interesting feature of the developed yield function is the use a fourth order tensorial operator to effect the

stress mapping to the IPE material. This tensorial operator can be used to describe different levels of symmetry in the modeled material. In an inverse manner, once the symmetries of the material are known, we know which and how many components of the tensorial operator should be determined in order to describe the anisotropic yield function of the material. The developed material as implemented into the finite element code ABAQUS Explicit with the aid of a user material subroutine VUMAT.

Once an adequate material model was established, different finite element techniques were evaluated and used for the modeling of three-dimensional sheet metal forming processes. Implicit analysis of loading and unloading and dynamic explicit analysis of loading combined with implicit static analysis of unloading were examined in the modeling of a channel forming process where correct prediction of springback was required. In the case of the implicit static analysis we found very good agreement with experimental results in predicting springback. In the case of the dynamic explicit analysis of loading combined with implicit static analysis of unloading, care should be taken to ensure that dynamic phenomena developed in the explicit analysis do not alter the stress and strain response of the modeled deformed material as compared with the implicit static analysis. The combination of dynamic explicit analysis for loading and implicit static analysis for unloading is a novel approach, developed in order to combine the computational advantage obtained by using explicit dynamic analysis in the loading, with the fast elastic static analysis of unloading in order to calculate springback.

In order to solve the manufacturing problem of part shape error due to springback by using finite element analysis, a tooling design algorithm was developed with the finite element analysis as an integral part of it. In this tooling design algorithm we start from the desired shape and we perform an inverse springback analysis (or "spring forward") analysis in order to calculate the tooling shape that will produce

the desired part shape upon springback. The "spring forward" simulation uses the formed material stresses obtained from the finite element analysis at the fully loaded state. The opposite of the material stresses are then embedded into the desired part while the displacement nodal degrees of freedom of the finite element model of the part are fixed. Release of the nodal constraints forces the material to "spring forward" to a new shape which is taken to be the next tooling shape. The procedure is repeated until the produced part shape converges to the desired part shape. This tooling design algorithm requires changes in the punch shape, the die shape and the binder shape in order to produce the desired part shape.

We investigated the effectiveness of the tooling design by using the "spring forward" technique for two shapes, a two-dimensional and a three-dimensional channel geometry. We were able to produce the desired part shape in both cases. The designs were also experimentally verified demonstrating the precision of our analysis and the effectiveness of the tooling design algorithm.

Future work in this area could be addressed towards the following subjects:

• Development of software links for immediate realization of the shapes obtained by the finite element analysis by using different prototype techniques such as those investigated by Walczyk [1994]. Such software is already available for the machining of the tools by using CNC. These techniques combined with the finite element analysis can provide a very powerful tool for the rapid design and production of tooling for sheet metal forming.

• Combination real time binder force control for the reduction of springback, Sunseri et al. [1994], with the tooling design algorithm developed in this thesis. Preliminary work in this subject was performed by Karafillis [1994]. The initial reduction of springback by using real time binder force control provides the potential for rapid convergence of the tooling design algorithm with very small final part shape error of

the part.

• Use of the combination of dynamic explicit analysis of loading with static implicit analysis of unloading in order to predict the part shape distortion due to trimming. By trimming off the binder from stamped sheet metal parts, the section forces and moments along the trimming line are set to zero causing the part to distort. Simulation of that phenomenon could be incorporated in tooling design in order to produce the desired part shape.

7 References

ABAQUS 5.2, User's Manual, (1993)

Asaro, R.J. and Needleman, A., Texture Development and Strain Hardening in Rate Dependent Polycrystalls. *Acta Metall.*, **33**, 923 (1985).

Ayres, R. A., SHAPESET: A Process to Reduce Sidewall Curl Springback in High-Strength Steel Rails, *J. App. Metalworking*, **3**, no. 2, 127, (1984).

Backofen W.A. *Deformation Processing*, Reading, Massachusetts (1972)

Backus, G. A Geometric Picture of Anisotropic Elastic Tensors. *Rev. Geophy. Spac. Phys.*, **8**, 633 (1970).

Barlat, F. and Lian, J. Plastic Behavior and Stretchability of Sheet Metals. Part I: A Yield Function for Orthotropic Sheet Under Plane Stress Conditions. *Int. J. Plast.*, **5**, 51 (1989).

Barlat, F., Lege, D.J and Brem, J.C. A Six Component Yield Function for Anisotropic Materials. *Int. J. Plast.*, **7**, 693 (1991).

Bassani, J.L. Yield Characterization of Metals with Transversely Isotropic Plastic Properties. *Int. J. Mech. Sci.*, **19**, 651 (1977).

Bishop, J.F.W., and Hill R. A theory of the Plastic Distortion of a Polycrystalline Aggregate under Combined Stresses.*Phil. Mayaz.*, **42**, 414 (1951).

Budianski, B. Anisotropic Plasticity of Plane-Isotropic Sheet. *Mechanics of Material Behavior,* eds. Dvorak, G.,J. and Shield, R.T., Amsterdam, p. 15 (1984).

Cao, J., Bakkestuen, R., Jalkh, P., Boyce, M.C., Hardt, D.E., Improvement of Forming Height and Stability of Aluminum Parts Using Active Binder Control, *Proceedings of 18th Biennial IDDRG Congress*, Lisbon, Portugal (1994).

Dawson, P.R. Beaudoin, A.J., Mathur, K.K. Simulation of Deformation-Induced Texture in Metal Forming. *NUMIFORM '92*, eds. Chenot, J.-L. et al., Sophia-Antipolis, France, p. 25 (1992).

Dvorak, G.J., and Bahei-el-Din, Y.A. Plasticity Analysis of Fibrous Composites. *J. App. Mech.*, **49**, 327 (1982).

Eggleston, H.G. *Convexity*, Cambridge University Press, p. 5 (1968).

Eisenberg, M.A., and Yen, C.F. The Anisotropic Deformation of Yield Surfaces. *J. Engng. Mat. Tech.*, **106**, 355 (1984).

Green, A.E. and Naghdi, P.M., A General Theory of an Elastic-Plastic Continuum, *Archive Rational Mech. & Anal.*, **18**, 251 (1965).

Gurtin, M.E. *An Introduction to Continuum Mechanics,* Academic Press, New York, p. 229 (1981).

Hardt, D.E., Hale, M., Closed-Loop Control of a Roll-Straightening Process, *Annals of CIRP*, 33, (1984).

Hardt, D.E., Fenn, R.C., Real-Time Sheet Forming Stability Control, *Proceedings of 16th Biennial IDDRG Congress*, Borlange, Sweden, (1990).

Hershey, A.V. Plasticity of Isotropic Aggregates of Anisotropic Face Centered Cubic Crystalls. *J. App. Mech.*, **76**, 241 (1954).

Hill, R. *The Mathematical Theory of Plasticity.* Clarendon Press, Oxford (1950)

Hill, R. Theoretical Plasticity of Textured Aggregates *Math. Proc. Camb. Phil. Soc.* **85**, 179 (1979).

Hill, R. Constitutive modelling of orthotropic plasticity in sheet metals. *J. Mech. Phys. Solids*, **38**, 405 (1990).

Honecker, A. and Matiasson, K., Finite Element Procedures for 3D sheet forming simulation, *Proc. NUMIFORM '89*, 457 (1989).

Hosford, W.F. A Generalized Isotropic Yield Criterion *J. App. Mech.*, **39**, 607 (1972).

Hughes, J.R.T., Numerical Implementation of Constitutive Models: Rate-Independent Deviatoric Plasticity, *Proceedings of the Theoretical Foundations of Large-Scale Computations of Nonlinear Material Behavior*, eds. Nemat-Nasser, S., Asaro, R.J., Boston (1983).

Hutchinson, J. W. Plastic Deformation of B.C.C. Polycrystalls. *J. Mech. Phys. Sol.* **12**, 25 (1964).

Jalkh P., Cao, J., Hardt, D., and Boyce, M.C., Optimal Forming of Aluminum 2008-T4 Conical Cups Using Force Trajectory Control, SAE pub. 930286 (1993).

Kalidindi, S.R., Bronkhorst, C.A. and Anand, L. Crystallographic Texture Evolution in Bulk Deformation Processing of FCC metals. *J. Mech. Phys. Sol.*, **40**, 537 (1992).

Kallend, J.S., Kocks, U.F., Rollet, A.D. and Wenk, H.-R. *popLA, The prefered orientation package from Los Alamos* (1989).

Karafillis, A.P., Tooling Design for Sheet Metal Forming Using Finite Element Analysis, SMME Thesis, MIT, February 1992.

Karafillis, A.P., Role of Finite Element Analysis in Sheet Metal Forming Processes, Technical Seminar at Gillette Corporate Research and Development, Boston, Massacusetts, May 1994.

Karafillis, A.P. and Boyce, M.C., Tooling Design in Sheet Metal Forming Using Springback Calculations, *Int. J. Mech. Sci.*, **34**, 113 (1992).

Karafillis, A.P. and Boyce, M.C., Tooling Design Accommodating Springback Errors, *Journal of Materials Processing Technology*, **32**, 499 (1992).

Karafillis, A.P. and Boyce, M.C., On the Modelling of Contact in Finite Element Analysis of Forming Processes, *Proceedings of NUMIFORM '92*, Valbonne, France, p. 267 (1992).

Karafillis, A.P. and Boyce, M.C., A General Anisotropic Yield Criterion Using Bounds and a Transformation Weighting Tensor, *J. Mech. Phys. Solids*, **41**, 1859 (1993)

Keum, Y.T., Nakamachi, E., Wagoner, R.H., Lee, J.K., Compatible Description of Tool Surfaces and FEM Meshes for Analysing Sheet Forming Opreations, *Int. J. Num. Meth. Engng.*, **30**, 1471, (1990).

Lege, D.J., Barlat. F., and Brem, J.C. Characterization and Mechanical Modelling of the Mechanical Behavior and Formability of a 2008-T4 Sheet Sample. *Int. J. Mech. Sci.*, **31**, 549 (1989).

Lian, J., Barlat, F., and Baudelet, B. *Int. J. Plast.* **5**, 131 (1989).

Liu, Y.C., The Effect of Restraining Force on Shape Deviations in Flanged Channels, *Trans. ASME, J. Eng. Mat. Tech.*, **110**, 389, (1988).

Lode, W. Versuche ueber den Einfluss der mittleren Hauptspannung auf das Fliessen der Metalle Eisen Kupfer und Nickel. *Z. Physik.*, **36**, 913 (1926).

MacEwen, S.R., Perrin, R.M Green, D., Makinde, A. and Neale, K. An Evaluation of Planar Biaxial Deformation in H19 Can-Stock Sheet. *Proc. 13th RISO,* eds.

Andersen, S.J. et al., Roskilde, Denmark, 539 (1992).

Matiasson, K., Bernspang, L., Samuelsson, A., Hamman, T., Schedin, E., Melander, A., Evaluation of a Dynamic Explicit Approach Using Explicit Integration in 3-D Sheet Forming Simulation *Proceedings of NUMIFORM '92*, Valbonne, France, p. 55, (1992)

Mendelson, A. *Plasticity: Theory and Application*, McMillan Co., New York, p. 87 (1963).

Mises, R. von *Gottinger Nachrichten, math. phys. klasse,* 582 (1913).

Onat, T. Representation of Inelastic Behavior in the Presence of anisotropy and Finite Deformations. *Plasticity of Metals at Finite Strain.*, eds. Lee, E.H., Mallet, R.L. (1981)

Onat, T. Effective Properties of Elastic Materials that Contain Penny Shaped Voids. *Int. J. Engng. Sci.* , **22**, 1013 (1984).

Ousterhout, K. Ph.D. Thesis, MIT (1991).

Rebello, N., Nagtegaal, J.C., and Hibbitt, H.D., Finite Element Analysis of Sheet Forming Processes, *Int. J. Numer. Meth. Eng.*, **30**, 1739, (1990).

Rebello, N., Nagtegaal, J.C., Taylor, L.M., Passmann, R., Comparison of Implicit and Explicit Finite Element Methods in the Simulation of Metal Forming Processes

Proceedings of NUMIFORM '92, Valbonne, France, p. 99, (1992)

Rodin, G.J., Parks, D.M. On Constitutive Relations in Nonlinear Fracture Mechanics. *J. Appl. Mech.*, **53**, 834 (1986).

Rollet, A.D., Jensen, D.J., and Stout, M.G. Modelling the Effect of Microstructure on Yield Anisotropy. *Proc. 13th RISO,* eds. Andersen, S.J. et al., Roskilde, Denmark, 93 (1992).

Shih, C.F. and Lee, D. Further Developments in Anisotropic Plasticity. *J. Engng. Mat. Tech.*, **100**, 294 (1978).

Stevenson, R., Springback in Simple Axisymmetric Stampings, *Metallurgical Transactions A*, **24A**, 925 (1993)

Sim, H.B., Boyce, M.C., Finite Element Analyses of Real-Time Stability Control in Sheet Forming Processes, *Trans. ASME, J. Eng. Mat. Tech.*, **114**, 180 (1991).

Story, J.M., Jarvis, G.W., Zonker, H.R. and Murtha, S.J., Issues and Trends in Automotive Aluminum Sheet Forming, SAE pub. 930277 (1993).

Stout, M.G., Hecker, S.S. and Bourcier, R. An Evaluation of Anisotropic Effective Stress-Strain Criteria for the Biaxial Yield and Flow of 2024 Aluminum Tubes. *J. Engng. Mat. Tech.*, **105**, p. 242 (1983).

Sunseri, M., Karafillis, A.P., Cao, J. and Boyce, M.C., Methods to Obtain the Net

Shape in Aluminum Sheet Forming Using Active Binder Force Control, to appear in Winter Meeting of ASME, Symposium on Computational Material Modeling (1994).

Taylor, G.I. Plastic Strains in Metals. *J. Inst. Met.* **62**, 307 (1938).

Taylor, L., Cao,J., Karafillis, A.P., and Boyce, M.C., Numerical Simulations of Sheet Metal Forming, *NUMISHEET 93* (1993).

Tresca, H. *Comptes Rendus Acad. Sci.*, **59**, 754 (1864).

Tucker, G.E.C. Texture and Earing in Deep Drawing of Aluminum. *Acta Metall.*, **9**, 275 (1961).

Voyadjis, G.Z., and Foroozesh, M. Anisotropic Distortional Yield Model. *J. App. Mech.*, **57**, 537 (1990).

Walczyk, D.F., A new Rapid Tooling Process for Sheet Metal Dies, *Proceedings of the 5th Annual International Conference on Rapid Prototyping*, Dayton, Ohio (1994).

Walczyk, D.F., A comparative study of Rapid Fabrication Methods for Sheet Metal Forming Dies, *Proceedings of the International Body Engineering Conference '94*, Detroit, Michigan, September 26-29 (1994).

Walczyk, D., PhD. thesis in progress, MIT, Dept. of Mechanical Engineering. (1994).

Wang, C.-T., Kinzel, G.L., and Altan, T., Process Simulation and Springback Control

in Plane Strain Sheet Bending, SAE pub. 930280, (1994)

Wang, N.M. and Budiansky, B., Analysis of Sheet Metal Stamping by a Finite Element Method, *Trans. ASME, J. App. Mech.*, **45**, (1978).

Webb, R.D. (1987) Spatial Frequency Based Closed-Loop Control in Sheet Metal Forming, Ph.D. Thesis, MIT, Dept. of Mechanical Engineering (1987).

Webb, R.D., Hardt, D.E., A Transfer Function Description of Sheet Metal Forming for Process Control, *Trans. ASME, J. Eng. Ind.*, **113**, 44, (1991).

Wenner, A.L., On Work Hardening and Springback in Plane Strain Draw Forming, *J. Applied Metalworking*, **2**, 277 (1983).

Woo, D.M. and Marshall, J. *The Engineer*, **208**, 135 (1959).

Woodthrope, J. and Pearce, R. The anomalous behavior of aluminum sheet under balanced biaxial tension. *Int. J. Mech. Sci.* **12**, 341 (1970).

8 Appendix I: Symmetries of the transformation tensor

As also shown by Onat [1984], a fourth order tensor with the symmetries of Eqns. (2.6,7) can be decomposed into a set of irreducible symmetric tensors:

$$\begin{aligned} L_{ijkl} &= a\delta_{ij}\delta_{kl} + b(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \\ &\quad +\delta_{ij}A_{kl} + A_{ij}\delta_{kl} \\ &\quad +\delta_{ik}F_{jl} + \delta_{il}F_{jk} + \delta_{jk}F_{il} + \delta_{jl}F_{ik} \\ &\quad +c_{ijkl} \end{aligned} \tag{AI-1}$$

where a and b are scalars, $\mathbf{A}$ and $\mathbf{F}$ are second order symmetric traceless tensors and $\mathbf{C}$ is a fourth order symmetric traceless tensor. However, by using the constraint of Eqn. (2.8) we obtain

$$\begin{aligned} b &= -\frac{3}{2}a \\ \mathbf{F} &= -\frac{3}{4}\mathbf{A} \end{aligned} \tag{AI-2}$$

Therefore, the final independent irreducible variables which compose $\mathbf{L}$ are $a, \mathbf{A}$ and $\mathbf{C}$. The decomposition of $\mathbf{L}$ according to Eqns. (AI-1,2)) is unique. The form of the irreducible tensorial variables which compose $\mathbf{L}$ depends on the symmetry group of the considered material. When the material is symmetric then all components of $\mathbf{A}$ and $\mathbf{C}$ are zero. When the material is fully asymmetric then both $\mathbf{A}$ and $\mathbf{C}$ are fully developed.

Following Onat's [1981] framework, we consider the irreducible variables $a, \mathbf{A}$ and $\mathbf{C}$ as state variables. These state variables can describe any desired material symmetry as tabulated in Table 1.1. The actual state of the material is represented by the set

$$\{Y, a, \mathbf{A}, \mathbf{C}, \boldsymbol{\sigma}, \mathbf{B}\} \tag{AI-3}$$

Introduction of additional irreducible tensors as state variables will not be considered here for a material that can exhibit only the groups of symmetries tabulated in Table 1.1.

Therefore the adopted form of **L** with the internal symmetries and constraints expressed by Eqns. (2.6-8) is compatible with the use of only three irreducible variables for the state representation of the material anisotropy. However, Eqns. (2.6) and (2.7) are the result of the symmetry of the stress tensor and the pressure independence of the yield condition whereas Eqn. (2.8) is artificially created in order to obtain the state represention of the anisotropy of the material with the minimal population of irreducible state variables.

9 Appendix II: Evaluation of the anisotropy in the case of orthotropic symmetry

We consider a thin rolled metallic sheet and an orthonormal reference frame $Oxyz$, where the $x - y$ plane is in the plane of the sheet and the x-axis is along the rolling direction. Also, we concentrate in using the IPE yield function of Eqn. (2.3) with the exponent k fixed to a high enough value ($k = 15$) to approximately describe the two bounds of the yield surface, whereas the mixing factor c will be determined to provide a best fit to the experimental data. As a first approximation we can use the value of c for isotropic materials, Method I, see Table 1.2.

If we assume orthotropic symmetry, then we can write Eqn. (2.4) in a matrix form:

$$\begin{Bmatrix} S_x \\ S_y \\ S_z \\ S_{yz} \\ S_{zx} \\ S_{xy} \end{Bmatrix} = C \begin{bmatrix} 1 & \beta_1 & \beta_2 & \cdot & \cdot & \cdot \\ \beta_1 & \alpha_1 & \beta_3 & \cdot & \cdot & \cdot \\ \beta_2 & \beta_3 & \alpha_2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \gamma_1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \gamma_2 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \gamma_3 \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{yz} \\ \sigma_{zx} \\ \sigma_{xy} \end{Bmatrix} \qquad \text{(AII-1)}$$

where all the blank components of the matrix representation of $\mathbf{L}$ in Eqn. (AII-1) are zero. However, by using the pressure independence condition (Eqn. (2.8)) we obtain:

$$\begin{aligned} \beta_1 &= \frac{\alpha_2 - \alpha_1 - 1}{2} \\ \beta_2 &= \frac{\alpha_1 - \alpha_2 - 1}{2} \\ \beta_3 &= \frac{1 - \alpha_1 - \alpha_2}{2} \end{aligned} \qquad \text{(AII-2)}$$

Therefore the tensorial operator consists of the following independent parameters:

$$\{C, \alpha_1, \alpha_2, \gamma_1, \gamma_2, \gamma_3\}$$

For an isotropic material, we have:

$$C = \frac{2}{3}$$

$$\alpha_1 = \alpha_2 = 1$$
$$\gamma_1 = \gamma_2 = \gamma_3 = \frac{3}{2} \qquad \text{(AII-3)}$$

We consider stress states of the anisotropic material where the value of the out-of-plane shear stress is negligible (e.g. sheet forming processes). Also we assume that $\mathbf{B} = \mathbf{0}$. Thus, the values of γ_1 and γ_2 are not critical when describing the anisotropy of the material and are set equal to the corresponding isotropic values.

If $R(0)$, $R(90)$ are the measured R-ratios at 0^o and 90^o with respect to the rolling direction, we can show that $R(0)$, $R(90)$ are functions of only α_1, α_2. $R(0)$ and $R(90)$ can be calculated by straightforward use of Eqn. (2.16). Therefore we can calculate α_1, α_2 as the two unknowns in a system of two nonlinear equations. For this purpose we used an error minimization numerical procedure (steepest gradient method).

In order to calculate the value of γ_3 we use the value of the R-ratio at 45^o with respect to the rolling direction $R(45)$. By using Eqn. (2.16) we can show that $R(45)$ is a function of α_1, α_2 and γ_3. Therefore γ_3 can be obtained as a root of a nonlinear equation, given that α_1, α_2 are already known.

We set now the value of the equivalent yield stress Y equal to the average yield stress as obtained by the three tension tests. We also require both the experimental and theoretical stress distributions in different directions in the plane of the sheet to have the same average value. However, by the homogeneity of the proposed yield function the averaged yield stress in the plane is inversely proportional to C. Therefore we can calculate C, as the ratio of the average experimental yield stress to the average yield stress obtained if we take $C = 1$.

Having caclulated all the components of the IPE transformation tensor, we can predict the yield stress distribution and the yield surface. If experiment data for the yield stress distribution are available we can repeat the above described procedure with different values of c, until the error in the prediction of the yield stress distri-

bution in different directions is minimized. Thus, we see that the necessary material parameters can easily be obtained through uniaxial tensile tests only. A flow chart of the numerical procedure used to calculate the material parameters is shown in Fig. 2.AII-1 (a).

In the case where $\mathbf{B}$ is nonzero, the procedure remains virtually the same. In this case, both $\mathbf{B}$ and c must be modified in a closed loop towards the direction of steepest descent of the error of the predicted distribution of the tensile *and* compressive yield stress until, the error is minimized. A flow chart of the numerical procedure used to calculate the material parameters in this case is shown in Fig. 2.AII-1 (b).

```
      program tens
C     calculate the non-isotropic transformation tensor.
C     Input: Plastic Poisson ratios obtained form tension tests at 0 and 90
C     deg., mixing coefficient in
C     yield function, Exponent of yield function,
C     backstress at 0 deg., backstress at 90 deg.,
C     lower bound in window of search for A(1,1) component
C     lower bound in window for search for A(2,2) component
C     value of b.
C     An unformatted input deck with the name 'inrat' must be provided.
C     This is an error minimization program. It looks for values of
C     the components of the anisotropic tensor  within a window in order
C     to fit the values of the plastic Poisson ratio at 0 deg. and 90 deg.
C     Output: The output written in file 'res1' will be the Poisson ratio at
C     0 deg., the Poisson ratio at 90 deg. and the ratio of the yield stress
C     at 0 deg. to the yield stress at 90 deg.
C     The output at file 'res2' includes all the components of the
C     fourth order tensorial operator.
C     The output file 'resorien' inludes the yield stress distribution
C     The output file 'resrat' includes the plastic Poisson ratio
C     distribution.
      IMPLICIT REAL*8(A-H,O-Z)
      character*1 alpha
      dimension svar(3,3), anis(3,3,3,3),
     &rotanis(3,3,3,3), unit(3,3),stress(3,3),
     &rota(3,3),tstress(3,3),v(3,3),d(3),phi(3,3),
     &srat(3,3),rsrat(3,3),arr(21,21),specrot(3,3)
      open(49,file='res1',access='sequential')
      open(47,file='res2',access='sequential')
      open(48,file='inrat',access='sequential')
      open(46,file='resrat',access='sequential')
      open(45,file='resorien',access='sequential')
      c=0.75
C      e=1.5
      error=10.
      unit(1,1)=1.
      unit(1,2)=0.
      unit(1,3)=0.
      unit(2,1)=0.
      unit(2,2)=1.
      unit(2,3)=0.
      unit(3,1)=0.
      unit(3,2)=0.
      unit(3,3)=1.
       uniax=1.
       a=-0.33333333333
       n=3
       np=3
      read(48,*)rx,r45des,ry,ccc,expm, del1, del2, rsmat,xllim,xulim,e
      del01=del1
      del02=del2
C     rx: Platic Poisson ratio from tension test at 0 deg.
C     r45des: Plastic Poisson ratio from tension test at 45 deg.
C     ry:  Platic Poisson ratio from tension test at 90 deg.
C     ccc: Mixing coefficient of yield function. Suggested values at Table 2.
C     expm: Exponent of yield function. Suggested values at Table 2.
```

```
C     del1: Backstress B(1,1) expressed as a fraction of the yield
C     stress at 0 deg.. If not known set it equal to 0.
C     del2: Backstress B(2,2) expressed as a fraction of the yield
C     stress at 0 deg.. If not not set equal to 0.
C     rsmat: Experimental value  of the ratio of the ratio of the yield
C     stresses. If not known set equal to 1.
C     xllim: lower bound in window of search for A(1,1) component. The width
C     of this window is equal to 1.0. Suggested value = -0.5
C     xulim: lower bound in window of search for A(2,2) component. The width
C     of this window is equal to 1.0. Suggested value = -0.5
C     Value of the isotropic component b. For pressure independent materials
C     b is equal to 1.5
      rx=-rx
      ry=-ry
      ccc=(1./ccc-1.)*(2.0**(expm-1.)+1)/3.0**expm
      do  i=1,3
        do j=1,3
          stress(i,j)=0.
        end do
      end do
      stress(1,1)=1.
      do mm=1,150
       svar(1,1)=xllim+0.0066666666*(mm-1)
        do nn=1,150
         svar(2,2)=xulim+0.00666666*(nn-1)
         svar(3,3)=-svar(1,1)-svar(2,2)
C
       do i=1,3
        do j=1,3
         do k=1,3
          do l=1,3
            rotanis(i,j,k,l)=unit(i,j)*svar(k,l)+
     &      unit(k,l)*svar(i,j)-
     &      c*unit(i,k)*svar(j,l)-
     &      c*unit(i,l)*svar(j,k)-
     &      c*unit(j,k)*svar(i,l)-
     &      c*unit(j,l)*svar(i,k)+
     &      a*unit(i,j)*unit(k,l)-
     &      e*a*(unit(i,k)*unit(j,l)+
     &      unit(i,l)*unit(j,k))
          end do
         end do
        end do
       end do
        end do
          do i=1,3
             do j=1,3
               sum=0.
                  do ii=1,3
                     do jj=1,3
      sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                  end do
                end do
              tstress(i,j)=sum
             end do
          end do
        call jacobi(tstress,n,np,d,v,nrot)
```

```
      yfuncu=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
        stress(1,1)=1.-del1
        stress(2,2)=0.-del2
        stress(3,3)=del1+del2
            do i=1,3
               do j=1,3
                 sum=0.
                    do ii=1,3
                       do jj=1,3
        sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                       end do
                    end do
                 tstress(i,j)=sum
               end do
             end do
           call jacobi(tstress,n,np,d,v,nrot)
           yfunc=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
           con=(yfuncu/yfunc)**(1./expm)
           cona=con**(expm-1.)
           s=uniax*con
C           sx=stress(1,1)*con
C           sy=stress(2,2)*con
           xas=theta*180./3.14159
           phi(1,1)=expm*(abs(d(1)))**(expm-1.)
     &*sign(1.,d(1))+ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))-ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
           phi(2,2)=expm*(abs(d(2)))**(expm-1.)
     &*sign(1.,d(2))-ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))+ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))
           phi(3,3)=expm*(abs(d(3)))**(expm-1.)
     &*sign(1.,d(3))-ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))+ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
        do is=1,3
          do js=1,3
            sum=0.
              do ks=1,3
                do ls=1,3
                  do ms=1,3
                   do ns=1,3
```

```
      sum=sum+rotanis(is,js,ks,ls)*phi(ms,ns)
     &*v(ks,ms)*v(ls,ns)
                  end do
                end do
              end do
            end do
          srat(is,js)=sum
        end do
      end do
      do irot=1,3
        do jrot=1,3
          sum=0.
          do krot=1,3
            do lrot=1,3
C              sum=sum+specrot(irot,krot)*
C     &specrot(jrot,lrot)*srat(krot,lrot)
              sum=sum+unit(irot,krot)*
     &unit(jrot,lrot)*srat(krot,lrot)
            end do
          end do
          rsrat(irot,jrot)=sum
          end do
        end do
       ill=1*i
        rxp=rsrat(2,2)/rsrat(1,1)
C
        do i=1,3
         do j=1,3
          do k=1,3
           do l=1,3
            rotanis(i,j,k,l)=unit(i,j)*svar(k,l)+
     &      unit(k,l)*svar(i,j)-
     &      c*unit(i,k)*svar(j,l)-
     &      c*unit(i,l)*svar(j,k)-
     &      c*unit(j,k)*svar(i,l)-
     &      c*unit(j,l)*svar(i,k)+
     &      a*unit(i,j)*unit(k,l)-
     &      e*a*(unit(i,k)*unit(j,l)+
     &      unit(i,l)*unit(j,k))
           end do
          end do
         end do
        end do
          do i=1,3
            do j=1,3
              sum=0.
                do ii=1,3
                  do jj=1,3
        sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                  end do
                end do
              tstress(i,j)=sum
            end do
          end do
        call jacobi(tstress,n,np,d,v,nrot)
       yfuncu=(abs(d(1)))**expm+(abs(d(2)))**expm
```

```
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
       stress(1,1)=0.-del1
       stress(2,2)=1.*rsmat-del2
       stress(3,3)=del1+del2
           do i=1,3
              do j=1,3
                sum=0.
                   do ii=1,3
                      do jj=1,3
       sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                      end do
                   end do
                tstress(i,j)=sum
              end do
           end do
         call jacobi(tstress,n,np,d,v,nrot)
         yfunc=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
         con=(yfuncu/yfunc)**(1./expm)
         cona=con**(expm-1.)
         s=uniax*con
C         sx=stress(1,1)*con
C         sy=stress(2,2)*con
         xas=theta*180./3.14159
         phi(1,1)=expm*(abs(d(1)))**(expm-1.)
     &*sign(1.,d(1))+ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))-ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
         phi(2,2)=expm*(abs(d(2)))**(expm-1.)
     &*sign(1.,d(2))-ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))+ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))
         phi(3,3)=expm*(abs(d(3)))**(expm-1.)
     &*sign(1.,d(3))-ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))+ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
       do is=1,3
         do js=1,3
           sum=0.
             do ks=1,3
               do ls=1,3
                 do ms=1,3
                  do ns=1,3
      sum=sum+rotanis(is,js,ks,ls)*phi(ms,ns)
```

```
     &*v(ks,ms)*v(ls,ns)
                  end do
                 end do
               end do
             end do
           srat(is,js)=sum
         end do
       end do
       do irot=1,3
         do jrot=1,3
            sum=0.
           do krot=1,3
             do lrot=1,3
C               sum=sum+specrot(irot,krot)*
C     &specrot(jrot,lrot)*srat(krot,lrot)
               sum=sum+unit(irot,krot)*
     &unit(jrot,lrot)*srat(krot,lrot)
            end do
           end do
           rsrat(irot,jrot)=sum
           end do
         end do
       ill=1*i
C       backspace(unit=49)
       ryp=rsrat(1,1)/rsrat(2,2)
       try=abs(rx-rxp)+abs(ry-ryp)
       if(try.lt.error) then
          error=try
          ila=mm
          jla=nn
            end if
           end do
         end do
       error=10.
       ipox=ila-1
       ipoy=jla-1
       do mm=1,20
        svar(1,1)=-0.00666666+ipox*0.0066666666
     &+xllim+0.00066666666*(mm-1)
         do nn=1,20
          svar(2,2)=-0.00666666+ipoy*0.0066666666
     &+xulim+0.00066666666*(nn-1)
          svar(3,3)=-svar(1,1)-svar(2,2)
C
         do i=1,3
          do j=1,3
           do k=1,3
            do l=1,3
              rotanis(i,j,k,l)=unit(i,j)*svar(k,l)+
     &        unit(k,l)*svar(i,j)-
     &        c*unit(i,k)*svar(j,l)-
     &        c*unit(i,l)*svar(j,k)-
     &        c*unit(j,k)*svar(i,l)-
     &        c*unit(j,l)*svar(i,k)+
     &        a*unit(i,j)*unit(k,l)-
     &        e*a*(unit(i,k)*unit(j,l)+
```

```
     &      unit(i,l)*unit(j,k))
           end do
          end do
         end do
        end do
            do i=1,3
               do j=1,3
                 sum=0.
                    do ii=1,3
                       do jj=1,3
        sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                       end do
                    end do
                  tstress(i,j)=sum
               end do
            end do
          call jacobi(tstress,n,np,d,v,nrot)
       yfuncu=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
        stress(1,1)=1.-del1
        stress(2,2)=0.-del2
        stress(3,3)=del1+del2
            do i=1,3
               do j=1,3
                 sum=0.
                    do ii=1,3
                       do jj=1,3
        sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                       end do
                    end do
                  tstress(i,j)=sum
               end do
            end do
          call jacobi(tstress,n,np,d,v,nrot)
          yfunc=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
          con=(yfuncu/yfunc)**(1./expm)
          cona=con**(expm-1.)
          s=uniax*con
C          sx=stress(1,1)*con
C          sy=stress(2,2)*con
          xas=theta*180./3.14159
          phi(1,1)=expm*(abs(d(1)))**(expm-1.)
     &*sign(1.,d(1))+ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))-ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
          phi(2,2)=expm*(abs(d(2)))**(expm-1.)
     &*sign(1.,d(2))-ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
```

```
     &sign(1.,(d(1)-d(2)))+ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))
         phi(3,3)=expm*(abs(d(3)))**(expm-1.)
     &*sign(1.,d(3))-ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))+ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
       do is=1,3
         do js=1,3
           sum=0.
             do ks=1,3
               do ls=1,3
                 do ms=1,3
                  do ns=1,3
      sum=sum+rotanis(is,js,ks,ls)*phi(ms,ns)
     &*v(ks,ms)*v(ls,ns)
                  end do
                 end do
               end do
             end do
           srat(is,js)=sum
         end do
       end do
       do irot=1,3
         do jrot=1,3
            sum=0.
           do krot=1,3
             do lrot=1,3
C               sum=sum+specrot(irot,krot)*
C     &specrot(jrot,lrot)*srat(krot,lrot)
               sum=sum+unit(irot,krot)*
     &unit(jrot,lrot)*srat(krot,lrot)
             end do
           end do
           rsrat(irot,jrot)=sum
           end do
         end do
       ill=1*i
C      backspace(unit=49)
       rxp=rsrat(2,2)/rsrat(1,1)
C
       do i=1,3
        do j=1,3
         do k=1,3
          do l=1,3
            rotanis(i,j,k,l)=unit(i,j)*svar(k,l)+
     &      unit(k,l)*svar(i,j)-
     &      c*unit(i,k)*svar(j,l)-
     &      c*unit(i,l)*svar(j,k)-
     &      c*unit(j,k)*svar(i,l)-
     &      c*unit(j,l)*svar(i,k)+
     &      a*unit(i,j)*unit(k,l)-
     &      e*a*(unit(i,k)*unit(j,l)+
```

```
     &      unit(i,l)*unit(j,k))
          end do
         end do
        end do
       end do
      end do
          do i=1,3
             do j=1,3
               sum=0.
                  do ii=1,3
                     do jj=1,3
      sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                     end do
                  end do
                tstress(i,j)=sum
             end do
          end do
        call jacobi(tstress,n,np,d,v,nrot)
     yfuncu=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
       stress(1,1)=0.-del1
       stress(2,2)=1.*rsmat-del2
       stress(3,3)=del1+del2
          do i=1,3
             do j=1,3
               sum=0.
                  do ii=1,3
                     do jj=1,3
       sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                     end do
                  end do
                tstress(i,j)=sum
             end do
          end do
        call jacobi(tstress,n,np,d,v,nrot)
        yfunc=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
          con=(yfuncu/yfunc)**(1./expm)
          cona=con**(expm-1.)
          s=uniax*con
C          sx=stress(1,1)*con
C          sy=stress(2,2)*con
          xas=theta*180./3.14159
          phi(1,1)=expm*(abs(d(1)))**(expm-1.)
     &*sign(1.,d(1))+ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))-ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
          phi(2,2)=expm*(abs(d(2)))**(expm-1.)
     &*sign(1.,d(2))-ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
```

```
     &sign(1.,(d(1)-d(2)))+ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))
         phi(3,3)=expm*(abs(d(3)))**(expm-1.)
     &*sign(1.,d(3))-ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))+ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
       do is=1,3
         do js=1,3
           sum=0.
             do ks=1,3
               do ls=1,3
                 do ms=1,3
                  do ns=1,3
      sum=sum+rotanis(is,js,ks,ls)*phi(ms,ns)
     &*v(ks,ms)*v(ls,ns)
                  end do
                 end do
               end do
             end do
           srat(is,js)=sum
         end do
       end do
       do irot=1,3
         do jrot=1,3
            sum=0.
           do krot=1,3
             do lrot=1,3
C               sum=sum+specrot(irot,krot)*
C     &specrot(jrot,lrot)*srat(krot,lrot)
               sum=sum+unit(irot,krot)*
     &unit(jrot,lrot)*srat(krot,lrot)
             end do
           end do
           rsrat(irot,jrot)=sum
           end do
         end do
        ill=1*i
C        backspace(unit=49)
         ryp=rsrat(1,1)/rsrat(2,2)
         try=abs(rx-rxp)+abs(ry-ryp)
         if(try.lt.error) then
            error=try
            ila=mm
            jla=nn
              end if
             end do
          end do

         svar(1,1)=-0.00666666+ipox*0.0066666666
     &+xllim+0.00066666666*(ila-1)
          svar(2,2)=-0.00666666+ipoy*0.0066666666
     &+xulim+0.00066666666*(jla-1)
         svar(3,3)=-svar(1,1)-svar(2,2)
```

```
      ze=0.0

      error=10.
      ipox=ila
      ipoy=jla
      ccx=svar(1,1)
      ccy=svar(2,2)
      do mm=1,20
       svar(1,1)=-0.00066666666+ccx
     &+0.000066666666*(mm-1)
        do nn=1,20
         svar(2,2)=-0.0006666666+ccy
     &+0.000066666666*(nn-1)
         svar(3,3)=-svar(1,1)-svar(2,2)
C
       do i=1,3
        do j=1,3
         do k=1,3
          do l=1,3
            rotanis(i,j,k,l)=unit(i,j)*svar(k,l)+
     &      unit(k,l)*svar(i,j)-
     &      c*unit(i,k)*svar(j,l)-
     &      c*unit(i,l)*svar(j,k)-
     &      c*unit(j,k)*svar(i,l)-
     &      c*unit(j,l)*svar(i,k)+
     &      a*unit(i,j)*unit(k,l)-
     &      e*a*(unit(i,k)*unit(j,l)+
     &      unit(i,l)*unit(j,k))
          end do
         end do
        end do
       end do
           do i=1,3
              do j=1,3
                sum=0.
                   do ii=1,3
                      do jj=1,3
       sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                      end do
                   end do
                tstress(i,j)=sum
              end do
           end do
         call jacobi(tstress,n,np,d,v,nrot)
      yfuncu=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
       stress(1,1)=1.-del1
       stress(2,2)=0.-del2
       stress(3,3)=del1+del2
           do i=1,3
              do j=1,3
                sum=0.
                   do ii=1,3
                      do jj=1,3
```

```
        sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                      end do
                    end do
                  tstress(i,j)=sum
                end do
              end do
              call jacobi(tstress,n,np,d,v,nrot)
              yfunc=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
              con=(1./yfunc)**(1./expm)
              cona=con**(expm-1.)
              s=uniax*con
              s1=s
C              sx=stress(1,1)*con
C              sy=stress(2,2)*con
              xas=theta*180./3.14159
              phi(1,1)=expm*(abs(d(1)))**(expm-1.)
     &*sign(1.,d(1))+ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))-ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
              phi(2,2)=expm*(abs(d(2)))**(expm-1.)
     &*sign(1.,d(2))-ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))+ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))
              phi(3,3)=expm*(abs(d(3)))**(expm-1.)
     &*sign(1.,d(3))-ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))+ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
            do is=1,3
              do js=1,3
                sum=0.
                  do ks=1,3
                    do ls=1,3
                      do ms=1,3
                       do ns=1,3
       sum=sum+rotanis(is,js,ks,ls)*phi(ms,ns)
     &*v(ks,ms)*v(ls,ns)
                       end do
                      end do
                    end do
                  end do
                srat(is,js)=sum
              end do
            end do
            do irot=1,3
              do jrot=1,3
                 sum=0.
```

```
            do krot=1,3
              do lrot=1,3
C                sum=sum+specrot(irot,krot)*
C     &specrot(jrot,lrot)*srat(krot,lrot)
               sum=sum+unit(irot,krot)*
     &unit(jrot,lrot)*srat(krot,lrot)
              end do
            end do
            rsrat(irot,jrot)=sum
            end do
          end do
         ill=1*i
C         backspace(unit=49)
          rxp=rsrat(2,2)/rsrat(1,1)
C
          do i=1,3
           do j=1,3
            do k=1,3
             do l=1,3
              rotanis(i,j,k,l)=unit(i,j)*svar(k,l)+
     &        unit(k,l)*svar(i,j)-
     &        c*unit(i,k)*svar(j,l)-
     &        c*unit(i,l)*svar(j,k)-
     &        c*unit(j,k)*svar(i,l)-
     &        c*unit(j,l)*svar(i,k)+
     &        a*unit(i,j)*unit(k,l)-
     &        e*a*(unit(i,k)*unit(j,l)+
     &        unit(i,l)*unit(j,k))
             end do
            end do
           end do
          end do
            do i=1,3
               do j=1,3
                 sum=0.
                    do ii=1,3
                       do jj=1,3
          sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                       end do
                    end do
                 tstress(i,j)=sum
               end do
            end do
           call jacobi(tstress,n,np,d,v,nrot)
         yfuncu=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
          stress(1,1)=0.-del1
          stress(2,2)=1.*rsmat-del2
          stress(3,3)=del1+del2
            do i=1,3
               do j=1,3
                 sum=0.
                    do ii=1,3
                       do jj=1,3
```

```
        sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                        end do
                      end do
                    tstress(i,j)=sum
                  end do
                end do
              call jacobi(tstress,n,np,d,v,nrot)
              yfunc=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
              con=(1./yfunc)**(1./expm)
              cona=con**(expm-1.)
              s=uniax*con
              s2=s
C              sx=stress(1,1)*con
C              sy=stress(2,2)*con
              xas=theta*180./3.14159
              phi(1,1)=expm*(abs(d(1)))**(expm-1.)
     &*sign(1.,d(1))+ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))-ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
              phi(2,2)=expm*(abs(d(2)))**(expm-1.)
     &*sign(1.,d(2))-ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))+ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))
              phi(3,3)=expm*(abs(d(3)))**(expm-1.)
     &*sign(1.,d(3))-ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))+ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
            do is=1,3
              do js=1,3
                sum=0.
                  do ks=1,3
                    do ls=1,3
                      do ms=1,3
                       do ns=1,3
       sum=sum+rotanis(is,js,ks,ls)*phi(ms,ns)
     &*v(ks,ms)*v(ls,ns)
                       end do
                      end do
                    end do
                  end do
                srat(is,js)=sum
              end do
            end do
            do irot=1,3
              do jrot=1,3
                 sum=0.
```

```
            do krot=1,3
              do lrot=1,3
C                sum=sum+specrot(irot,krot)*
C     &specrot(jrot,lrot)*srat(krot,lrot)
                 sum=sum+unit(irot,krot)*
     &unit(jrot,lrot)*srat(krot,lrot)
              end do
            end do
            rsrat(irot,jrot)=sum
            end do
          end do
        ill=1*i
C        backspace(unit=49)
         ryp=rsrat(1,1)/rsrat(2,2)
         try=abs(rx-rxp)+abs(ry-ryp)
         if(try.lt.error) then
            error=try
            prx=rxp
            pry=ryp
C            ps=(s2+s2*del2)/(s1+s1*del1)
            ps=s2/s1
            ila=mm
            jla=nn
              end if
            end do
          end do

        svar(1,1)=-0.0006666666666666+ccx
     &+0.000066666666*(ila-1)
          svar(2,2)=-0.00066666666666+ccy
     &+0.000063666666*(jla-1)
         svar(3,3)=-svar(1,1)-svar(2,2)
         ze=0.0
         write(49,*) svar(1,1),ze,ze,ze,
     &svar(2,2),ze,ze,ze,svar(3,3)
         write(49,*) 'Obtained Poisson ratio at 0 deg:', -prx
         write(49,*) 'Obtained Poisson ratio at 90 deg:', -pry
         write(49,*) 'Obtained yield straength ratio Y90/Y0:', ps
        c=0.75
C        e=1.5
        unit(1,1)=1.
        unit(1,2)=0.
        unit(1,3)=0.
        unit(2,1)=0.
        unit(2,2)=1.
        unit(2,3)=0.
        unit(3,1)=0.
        unit(3,2)=0.
        unit(3,3)=1.
C        read(48,*) a, svar(1,1), svar(1,2), svar(1,3),
C       &svar(2,1), svar(2,2), svar(2,3),
C       &svar(3,1), svar(3,2), svar(3,3)
        do i=1,3
         do j=1,3
          do k=1,3
           do l=1,3
```

```
            anis(i,j,k,l)=unit(i,j)*svar(k,l)+
     &      unit(k,l)*svar(i,j)-
     &      c*unit(i,k)*svar(j,l)-
     &      c*unit(i,l)*svar(j,k)-
     &      c*unit(j,k)*svar(i,l)-
     &      c*unit(j,l)*svar(i,k)+
     &      a*unit(i,j)*unit(k,l)-
     &      e*a*(unit(i,k)*unit(j,l)+
     &      unit(i,l)*unit(j,k))
            rotanis(i,j,k,l)=anis(i,j,k,l)
C            write(47,*) i,j,k,l,anis(i,j,k,l)
          end do
         end do
        end do
       end do
       specrot(1,1)=1.
       specrot(1,2)=0.
       specrot(1,3)=0.
       specrot(2,1)=0.
       specrot(2,2)=1.
       specrot(2,3)=0.
       specrot(3,1)=0.
       specrot(3,2)=0.
       specrot(3,3)=1.
       stress(1,1)=1.
       stress(1,2)=0.
       stress(1,3)=0.
       stress(2,1)=0.
       stress(2,2)=0.
       stress(2,3)=0.
       stress(3,1)=0.
       stress(3,2)=0.
       stress(3,3)=0.
       NP=3
       N=3
C      write(6,3)
C    3 format('enter mixing coefficient:',$)
          do i=1,3
             do j=1,3
               sum=0.
                  do ii=1,3
                     do jj=1,3
       sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                     end do
                  end do
               tstress(i,j)=sum
             end do
          end do
        call jacobi(tstress,n,np,d,v,nrot)
      yfuncu=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
      yat0=(yfuncu/2.)**(1./expm)
      del1=del01*yat0
```

```
      del2=del02*yat0
      stress(2,1)=stress(1,2)
      do  in=1,16
         theta=3.14159/30.0*(in-1)
         specrot(1,1)=cos(theta)
         specrot(1,2)=sin(theta)
         specrot(1,3)=0.
         specrot(2,1)=-sin(theta)
         specrot(2,2)=cos(theta)
         specrot(2,3)=0.
         specrot(3,1)=0.
         specrot(3,2)=0.
         specrot(3,3)=1.
         uniax=1.
  628     stress(1,1)=0.5*uniax*(1.+cos(2.*theta))-del1
         stress(2,2)=0.5*uniax*(1.-cos(2.*theta))-del2
         stress(1,2)=0.5*uniax*sin(2.*theta)
         stress(2,1)=stress(1,2)
         stress(3,3)=del1+del2
C         stress(2,2)=ratio
          do i=1,3
             do j=1,3
               sum=0.
                  do ii=1,3
                     do jj=1,3
          sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                     end do
                  end do
               tstress(i,j)=sum
             end do
           end do
         call jacobi(tstress,n,np,d,v,nrot)
         yfunc=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
         con=(yfuncu/yfunc)**(1./expm)
         cona=con**(expm-1.)
         buniax=uniax
         uniax=uniax*con
         if(in.eq.1) reg1=yat0*uniax
         if(in.eq.6) reg2=yat0*uniax
         if(in.eq.7) reg3=yat0*uniax
         if(in.eq.16) reg4=yat0*uniax
         if(abs(buniax-uniax).gt.1.0D-4) goto 628
C         sx=stress(1,1)*con
C         sy=stress(2,2)*con
         xas=theta*180./3.14159
C         write(49,*)xas, uniax, con
         phi(1,1)=expm*(abs(d(1)))**(expm-1.)
     &*sign(1.,d(1))+ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))-ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
```

```
          phi(2,2)=expm*(abs(d(2)))**(expm-1.)
     &*sign(1.,d(2))-ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))+ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))
          phi(3,3)=expm*(abs(d(3)))**(expm-1.)
     &*sign(1.,d(3))-ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))+ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
        do is=1,3
          do js=1,3
            sum=0.
              do ks=1,3
                do ls=1,3
                  do ms=1,3
                   do ns=1,3
      sum=sum+rotanis(is,js,ks,ls)*phi(ms,ns)
     &*v(ks,ms)*v(ls,ns)
                   end do
                  end do
                end do
              end do
            srat(is,js)=sum
          end do
        end do
        do irot=1,3
          do jrot=1,3
             sum=0.
            do krot=1,3
              do lrot=1,3
                sum=sum+specrot(irot,krot)*
     &specrot(jrot,lrot)*srat(krot,lrot)
              end do
            end do
            rsrat(irot,jrot)=sum
            if(in.eq.1) rreg1=-rsrat(2,2)/rsrat(1,1)
            if(in.eq.6) rreg2=-rsrat(2,2)/rsrat(1,1)
            if(in.eq.7) rreg3=-rsrat(2,2)/rsrat(1,1)
            if(in.eq.16) rreg4=-rsrat(2,2)/rsrat(1,1)
            end do
          end do
C       write(47,*)xas,rsrat(2,2)/rsrat(3,3)
C       write(47,*) xas,rsrat(1,1)*cona,
C      &rsrat(2,2)*cona,rsrat(3,3)*cona
        end do
        r45c1=(rreg2+rreg3)/2.0
        rotan1=rotanis(1,2,1,2)
        rotanis(1,2,1,2)=rotanis(1,2,1,2)*1.05
        rotanis(2,1,1,2)=rotanis(1,2,1,2)*1.05
        rotanis(1,2,2,1)=rotanis(1,2,1,2)*1.05
        rotanis(2,1,2,1)=rotanis(1,2,1,2)*1.05
        specrot(1,1)=1.
```

```
      specrot(1,2)=0.
      specrot(1,3)=0.
      specrot(2,1)=0.
      specrot(2,2)=1.
      specrot(2,3)=0.
      specrot(3,1)=0.
      specrot(3,2)=0.
      specrot(3,3)=1.
      stress(1,1)=1.
      stress(1,2)=0.
      stress(1,3)=0.
      stress(2,1)=0.
      stress(2,2)=0.
      stress(2,3)=0.
      stress(3,1)=0.
      stress(3,2)=0.
      stress(3,3)=0.
      NP=3
      N=3
C     write(6,3)
C    3 format('enter mixing coefficient:',$)
          do i=1,3
             do j=1,3
               sum=0.
                  do ii=1,3
                     do jj=1,3
      sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                     end do
                  end do
               tstress(i,j)=sum
             end do
          end do
        call jacobi(tstress,n,np,d,v,nrot)
       yfuncu=(abs(d(1)))**expm+(abs(d(2)))**expm
      &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
      &+(abs(d(2)-d(3)))**expm
      &+(abs(d(3)-d(1)))**expm)*ccc
       yat0=(yfuncu/2.)**(1./expm)
       del1=del01*yat0
       del2=del02*yat0
       stress(2,1)=stress(1,2)
       do  in=1,16
          theta=3.14159/30.0*(in-1)
          specrot(1,1)=cos(theta)
          specrot(1,2)=sin(theta)
          specrot(1,3)=0.
          specrot(2,1)=-sin(theta)
          specrot(2,2)=cos(theta)
          specrot(2,3)=0.
          specrot(3,1)=0.
          specrot(3,2)=0.
          specrot(3,3)=1.
          uniax=1.
  625      stress(1,1)=0.5*uniax*(1.+cos(2.*theta))-del1
          stress(2,2)=0.5*uniax*(1.-cos(2.*theta))-del2
```

```
          stress(1,2)=0.5*uniax*sin(2.*theta)
          stress(2,1)=stress(1,2)
          stress(3,3)=del1+del2
C          stress(2,2)=ratio
            do i=1,3
               do j=1,3
                 sum=0.
                    do ii=1,3
                       do jj=1,3
           sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                    end do
                 end do
              tstress(i,j)=sum
               end do
            end do
          call jacobi(tstress,n,np,d,v,nrot)
          yfunc=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
          con=(yfuncu/yfunc)**(1./expm)
          cona=con**(expm-1.)
          buniax=uniax
          uniax=uniax*con
          if(in.eq.1) reg1=yat0*uniax
          if(in.eq.6) reg2=yat0*uniax
          if(in.eq.7) reg3=yat0*uniax
          if(in.eq.16) reg4=yat0*uniax
          if(abs(buniax-uniax).gt.1.0D-4) goto 625
C          sx=stress(1,1)*con
C          sy=stress(2,2)*con
          xas=theta*180./3.14159
C          write(49,*)xas, uniax, con
          phi(1,1)=expm*(abs(d(1)))**(expm-1.)
     &*sign(1.,d(1))+ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))-ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
          phi(2,2)=expm*(abs(d(2)))**(expm-1.)
     &*sign(1.,d(2))-ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))+ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))
          phi(3,3)=expm*(abs(d(3)))**(expm-1.)
     &*sign(1.,d(3))-ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))+ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
        do is=1,3
          do js=1,3
            sum=0.
```

```
              do ks=1,3
                do ls=1,3
                  do ms=1,3
                   do ns=1,3
       sum=sum+rotanis(is,js,ks,ls)*phi(ms,ns)
      &*v(ks,ms)*v(ls,ns)
                   end do
                  end do
                end do
              end do
            srat(is,js)=sum
          end do
        end do
        do irot=1,3
          do jrot=1,3
             sum=0.
            do krot=1,3
              do lrot=1,3
                sum=sum+specrot(irot,krot)*
      &specrot(jrot,lrot)*srat(krot,lrot)
              end do
            end do
            rsrat(irot,jrot)=sum
            if(in.eq.1) rreg1=-rsrat(2,2)/rsrat(1,1)
            if(in.eq.6) rreg2=-rsrat(2,2)/rsrat(1,1)
            if(in.eq.7) rreg3=-rsrat(2,2)/rsrat(1,1)
            if(in.eq.16) rreg4=-rsrat(2,2)/rsrat(1,1)
            end do
          end do
C       write(47,*)xas,rsrat(2,2)/rsrat(3,3)
C       write(47,*) xas,rsrat(1,1)*cona,
C      &rsrat(2,2)*cona,rsrat(3,3)*cona
        end do
        r45c2=(rreg2+rreg3)/2.0
        rotan2=rotanis(1,2,1,2)
        slope=(r45c1-r45c2)/0.05
        solu=rotan2-(r45c2-r45des)/slope
        rotanis(1,2,1,2)=solu
        rotanis(2,1,1,2)=solu
        rotanis(1,2,2,1)=solu
        rotanis(2,1,2,1)=solu
        specrot(1,1)=1.
        specrot(1,2)=0.
        specrot(1,3)=0.
        specrot(2,1)=0.
        specrot(2,2)=1.
        specrot(2,3)=0.
        specrot(3,1)=0.
        specrot(3,2)=0.
        specrot(3,3)=1.
        stress(1,1)=1.
        stress(1,2)=0.
        stress(1,3)=0.
        stress(2,1)=0.
        stress(2,2)=0.
```

```
        stress(2,3)=0.
        stress(3,1)=0.
        stress(3,2)=0.
        stress(3,3)=0.
        NP=3
        N=3
C       write(6,3)
C     3 format('enter mixing coefficient:',$)
            do i=1,3
               do j=1,3
                 sum=0.
                    do ii=1,3
                       do jj=1,3
     sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                    end do
                 end do
                tstress(i,j)=sum
              end do
            end do
          call jacobi(tstress,n,np,d,v,nrot)
       yfuncu=(abs(d(1)))**expm+(abs(d(2)))**expm
      &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
      &+(abs(d(2)-d(3)))**expm
      &+(abs(d(3)-d(1)))**expm)*ccc
       yat0=(yfuncu/2.)**(1./expm)
       del1=del01*yat0
       del2=del02*yat0
       stress(2,1)=stress(1,2)
       do  in=1,16
          theta=3.14159/30.0*(in-1)
          specrot(1,1)=cos(theta)
          specrot(1,2)=sin(theta)
          specrot(1,3)=0.
          specrot(2,1)=-sin(theta)
          specrot(2,2)=cos(theta)
          specrot(2,3)=0.
          specrot(3,1)=0.
          specrot(3,2)=0.
          specrot(3,3)=1.
          uniax=1.
  626      stress(1,1)=0.5*uniax*(1.+cos(2.*theta))-del1
          stress(2,2)=0.5*uniax*(1.-cos(2.*theta))-del2
          stress(1,2)=0.5*uniax*sin(2.*theta)
          stress(2,1)=stress(1,2)
          stress(3,3)=del1+del2
C          stress(2,2)=ratio
           do i=1,3
              do j=1,3
                sum=0.
                   do ii=1,3
                      do jj=1,3
           sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                   end do
                end do
               tstress(i,j)=sum
              end do
```

```
          end do
        call jacobi(tstress,n,np,d,v,nrot)
        yfunc=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
        con=(2./yfunc)**(1./expm)
        cona=con**(expm-1.)
        buniax=uniax
        uniax=uniax*con
        if(in.eq.1) reg1=uniax
        if(in.eq.6) reg2=uniax
        if(in.eq.7) reg3=uniax
        if(in.eq.16) reg4=uniax
        write(*,*) reg1, reg2, reg3, reg4
C        if(abs(buniax-uniax).gt.1.0D-4) goto 626
C        sx=stress(1,1)*con
C        sy=stress(2,2)*con
        xas=theta*180./3.14159
C        write(45,*)xas, uniax
        phi(1,1)=expm*(abs(d(1)))**(expm-1.)
     &*sign(1.,d(1))+ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))-ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
        phi(2,2)=expm*(abs(d(2)))**(expm-1.)
     &*sign(1.,d(2))-ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))+ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))
        phi(3,3)=expm*(abs(d(3)))**(expm-1.)
     &*sign(1.,d(3))-ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))+ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
      do is=1,3
        do js=1,3
          sum=0.
            do ks=1,3
              do ls=1,3
                do ms=1,3
                 do ns=1,3
     sum=sum+rotanis(is,js,ks,ls)*phi(ms,ns)
     &*v(ks,ms)*v(ls,ns)
                 end do
                end do
              end do
            end do
          srat(is,js)=sum
        end do
      end do
```

```
        do irot=1,3
          do jrot=1,3
             sum=0.
            do krot=1,3
              do lrot=1,3
                sum=sum+specrot(irot,krot)*
     &specrot(jrot,lrot)*srat(krot,lrot)
              end do
            end do
            rsrat(irot,jrot)=sum
            if(in.eq.1) rreg1=-rsrat(2,2)/rsrat(1,1)
            if(in.eq.6) rreg2=-rsrat(2,2)/rsrat(1,1)
            if(in.eq.7) rreg3=-rsrat(2,2)/rsrat(1,1)
            if(in.eq.16) rreg4=-rsrat(2,2)/rsrat(1,1)
            end do
          end do
        write(46,*)xas,-rsrat(2,2)/rsrat(1,1)
C       write(47,*) xas,rsrat(1,1)*cona,
C      &rsrat(2,2)*cona,rsrat(3,3)*cona
        end do
        regav=(reg1+reg2+reg3+reg4)/4.
       do i=1,3
        do j=1,3
         do k=1,3
          do l=1,3
             rotanis(i,j,k,l)=rotanis(i,j,k,l)*regav
             write(47,*) i,j,k,l,rotanis(i,j,k,l)
           end do
          end do
         end do
        end do
        close(unit=48)
        close(unit=49)
C   146 format(f4.2,4X)
C             end do
C           write(49,147)
C   147 format(/)
        close(unit=47)
        specrot(1,1)=1.
        specrot(1,2)=0.
        specrot(1,3)=0.
        specrot(2,1)=0.
        specrot(2,2)=1.
        specrot(2,3)=0.
        specrot(3,1)=0.
        specrot(3,2)=0.
        specrot(3,3)=1.
        stress(1,1)=1.
        stress(1,2)=0.
        stress(1,3)=0.
        stress(2,1)=0.
        stress(2,2)=0.
        stress(2,3)=0.
        stress(3,1)=0.
        stress(3,2)=0.
        stress(3,3)=0.
```

```
      NP=3
      N=3
C     write(6,3)
C    3 format('enter mixing coefficient:',$)
        do i=1,3
          do j=1,3
            sum=0.
              do ii=1,3
                do jj=1,3
      sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                end do
              end do
            tstress(i,j)=sum
          end do
        end do
        call jacobi(tstress,n,np,d,v,nrot)
       yfuncu=(abs(d(1)))**expm+(abs(d(2)))**expm
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
       yat0=(yfuncu/2.)**(1./expm)
       del1=del1*yat0
       del2=del2*yat0
       stress(2,1)=stress(1,2)
       do  in=1,16
          theta=3.14159/30.0*(in-1)
          specrot(1,1)=cos(theta)
          specrot(1,2)=sin(theta)
          specrot(1,3)=0.
          specrot(2,1)=-sin(theta)
          specrot(2,2)=cos(theta)
          specrot(2,3)=0.
          specrot(3,1)=0.
          specrot(3,2)=0.
          specrot(3,3)=1.
          uniax=1.
  627      stress(1,1)=0.5*uniax*(1.+cos(2.*theta))-del1
          stress(2,2)=0.5*uniax*(1.-cos(2.*theta))-del2
          stress(1,2)=0.5*uniax*sin(2.*theta)
          stress(2,1)=stress(1,2)
          stress(3,3)=del1+del2
C          stress(2,2)=ratio
           do i=1,3
              do j=1,3
                sum=0.
                  do ii=1,3
                    do jj=1,3
           sum=sum+rotanis(i,j,ii,jj)*stress(ii,jj)
                  end do
                end do
               tstress(i,j)=sum
              end do
            end do
          call jacobi(tstress,n,np,d,v,nrot)
          yfunc=(abs(d(1)))**expm+(abs(d(2)))**expm
```

```
     &+(abs(d(3)))**expm+((abs(d(1)-d(2)))**expm
     &+(abs(d(2)-d(3)))**expm
     &+(abs(d(3)-d(1)))**expm)*ccc
         con=(2./yfunc)**(1./expm)
         cona=con**(expm-1.)
         buniax=uniax
         uniax=uniax*con
         if(in.eq.1) reg1=uniax
         if(in.eq.6) reg2=uniax
         if(in.eq.7) reg3=uniax
         if(in.eq.16) reg4=uniax
         if(abs(buniax-uniax).gt.1.0D-4) goto 627
C         sx=stress(1,1)*con
C         sy=stress(2,2)*con
         xas=theta*180./3.14159
         write(45,*)xas, uniax
         phi(1,1)=expm*(abs(d(1)))**(expm-1.)
     &*sign(1.,d(1))+ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))-ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
         phi(2,2)=expm*(abs(d(2)))**(expm-1.)
     &*sign(1.,d(2))-ccc*expm*
     &(abs(d(1)-d(2)))**(expm-1.)*
     &sign(1.,(d(1)-d(2)))+ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))
         phi(3,3)=expm*(abs(d(3)))**(expm-1.)
     &*sign(1.,d(3))-ccc*expm*
     &(abs(d(2)-d(3)))**(expm-1.)*
     &sign(1.,(d(2)-d(3)))+ccc*expm*
     &(abs(d(3)-d(1)))**(expm-1.)*
     &sign(1.,(d(3)-d(1)))
       do is=1,3
         do js=1,3
           sum=0.
             do ks=1,3
               do ls=1,3
                 do ms=1,3
                   do ns=1,3
      sum=sum+rotanis(is,js,ks,ls)*phi(ms,ns)
     &*v(ks,ms)*v(ls,ns)
                 end do
               end do
             end do
           end do
         srat(is,js)=sum
       end do
     end do
     do irot=1,3
       do jrot=1,3
          sum=0.
         do krot=1,3
```

```
            do lrot=1,3
              sum=sum+specrot(irot,krot)*
     &specrot(jrot,lrot)*srat(krot,lrot)
            end do
          end do
          rsrat(irot,jrot)=sum
          if(in.eq.1) rreg1=-rsrat(2,2)/rsrat(1,1)
          if(in.eq.6) rreg2=-rsrat(2,2)/rsrat(1,1)
          if(in.eq.7) rreg3=-rsrat(2,2)/rsrat(1,1)
          if(in.eq.16) rreg4=-rsrat(2,2)/rsrat(1,1)
          end do
        end do
C       write(47,*)xas,rsrat(2,2)/rsrat(3,3)
C       write(47,*) xas,rsrat(1,1)*cona,
C      &rsrat(2,2)*cona,rsrat(3,3)*cona
        end do
        end




      SUBROUTINE JACOBI(A,N,NP,D,V,NROT)
C taken from 'Numerical Recipes in Fortran, William Press et al.,
C Cambridge University Press (1992).
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER (NMAX=100)
      DIMENSION A(NP,NP),D(NP),V(NP,NP),B(NMAX),Z(NMAX)
      DO 12 IP=1,N
        DO 11 IQ=1,N
          V(IP,IQ)=0.
11      CONTINUE
        V(IP,IP)=1.
12    CONTINUE
      DO 13 IP=1,N
        B(IP)=A(IP,IP)
        D(IP)=B(IP)
        Z(IP)=0.
13    CONTINUE
      NROT=0
      DO 24 I=1,50
        SM=0.
        DO 15 IP=1,N-1
          DO 14 IQ=IP+1,N
            SM=SM+ABS(A(IP,IQ))
14        CONTINUE
15      CONTINUE
        IF(SM.EQ.0.)RETURN
        IF(I.LT.4)THEN
          TRESH=0.2*SM/N**2
        ELSE
          TRESH=0.
        ENDIF
        DO 22 IP=1,N-1
          DO 21 IQ=IP+1,N
            G=100.*ABS(A(IP,IQ))
            IF((I.GT.4).AND.(ABS(D(IP))+G.EQ.ABS(D(IP)))
```

```
     *          .AND.(ABS(D(IQ))+G.EQ.ABS(D(IQ))))THEN
                A(IP,IQ)=0.
              ELSE IF(ABS(A(IP,IQ)).GT.TRESH)THEN
                H=D(IQ)-D(IP)
                IF(ABS(H)+G.EQ.ABS(H))THEN
                  T=A(IP,IQ)/H
                ELSE
                  THETA=0.5*H/A(IP,IQ)
                  T=1./(ABS(THETA)+SQRT(1.+THETA**2))
                  IF(THETA.LT.0.)T=-T
                ENDIF
                C=1./SQRT(1+T**2)
                S=T*C
                TAU=S/(1.+C)
                H=T*A(IP,IQ)
                Z(IP)=Z(IP)-H
                Z(IQ)=Z(IQ)+H
                D(IP)=D(IP)-H
                D(IQ)=D(IQ)+H
                A(IP,IQ)=0.
                DO 16 J=1,IP-1
                  G=A(J,IP)
                  H=A(J,IQ)
                  A(J,IP)=G-S*(H+G*TAU)
                  A(J,IQ)=H+S*(G-H*TAU)
16              CONTINUE
                DO 17 J=IP+1,IQ-1
                  G=A(IP,J)
                  H=A(J,IQ)
                  A(IP,J)=G-S*(H+G*TAU)
                  A(J,IQ)=H+S*(G-H*TAU)
17              CONTINUE
                DO 18 J=IQ+1,N
                  G=A(IP,J)
                  H=A(IQ,J)
                  A(IP,J)=G-S*(H+G*TAU)
                  A(IQ,J)=H+S*(G-H*TAU)
18              CONTINUE
                DO 19 J=1,N
                  G=V(J,IP)
                  H=V(J,IQ)
                  V(J,IP)=G-S*(H+G*TAU)
                  V(J,IQ)=H+S*(G-H*TAU)
19              CONTINUE
                NROT=NROT+1
              ENDIF
21          CONTINUE
22        CONTINUE
          DO 23 IP=1,N
            B(IP)=B(IP)+Z(IP)
            D(IP)=B(IP)
            Z(IP)=0.
23        CONTINUE
24      CONTINUE
        PAUSE '50 iterations should never happen'
        RETURN
        END
```

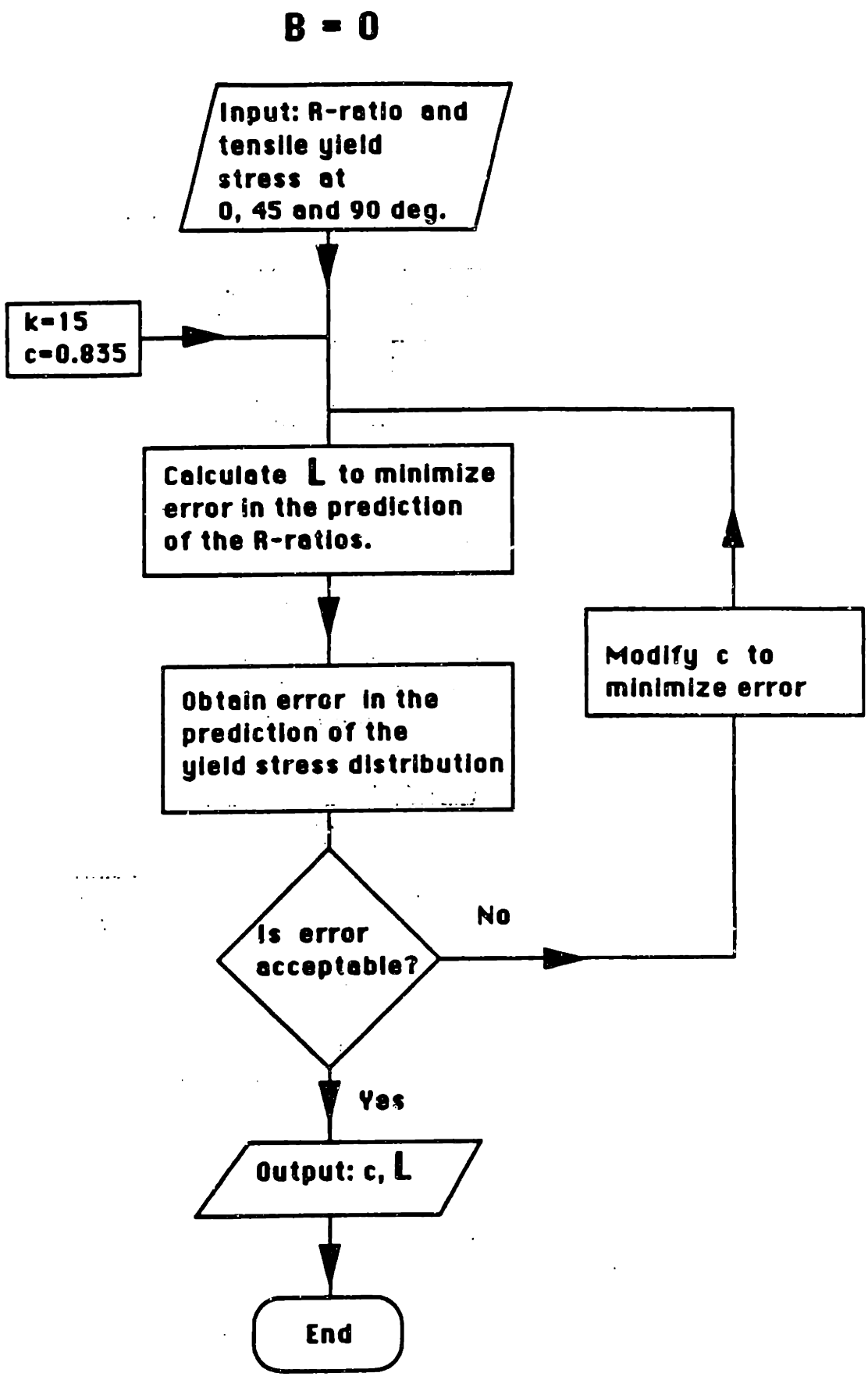


Fig. AII-1(a): A flow chart of the numerical procedure used to calculate the yield surface when $B = 0$.

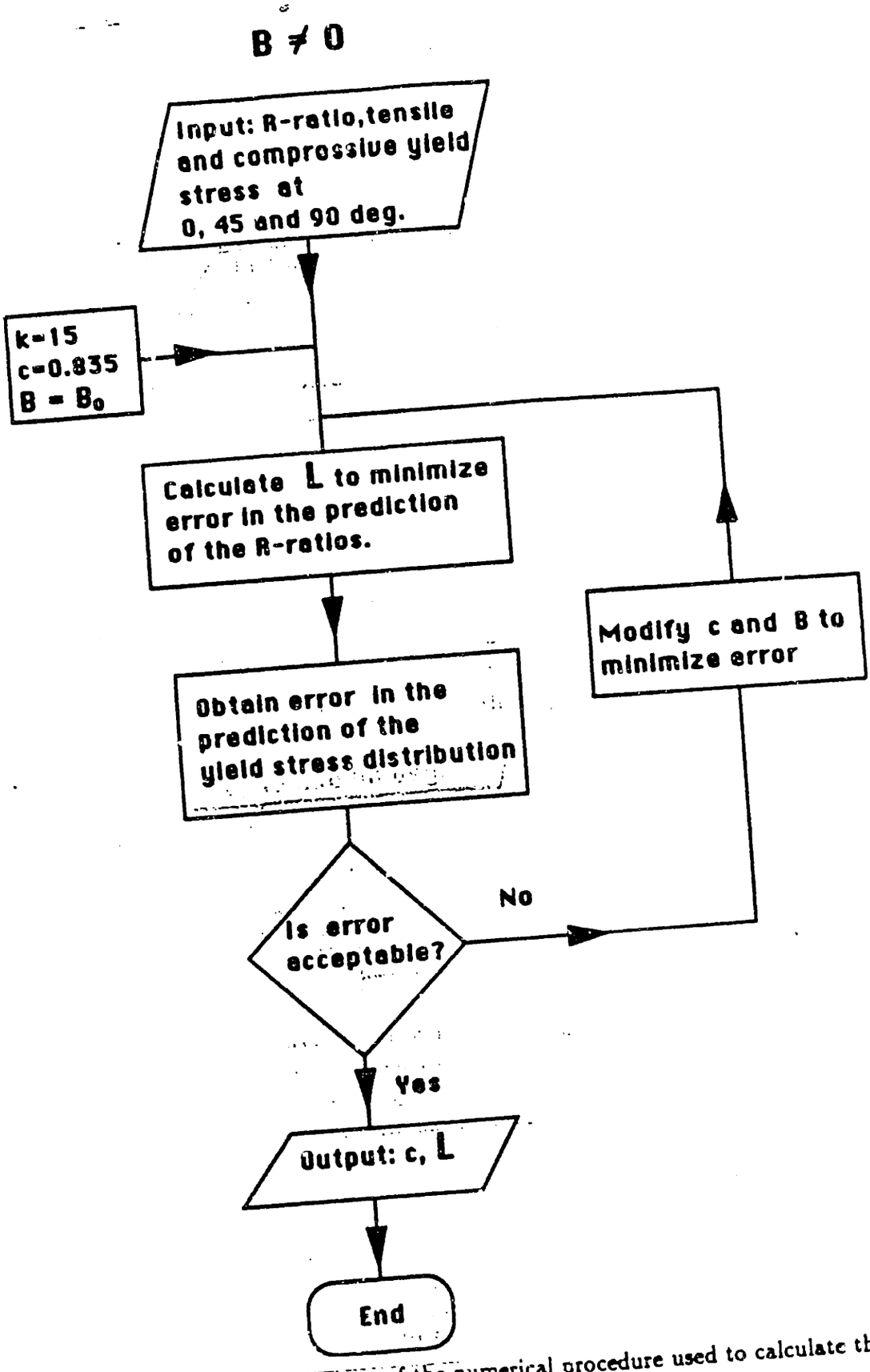


Fig. AII-1(b): A flow chart of the numerical procedure used to calculate the yield surface when $B \neq 0$.

10 Appendix III: Listing of input decks and VUMAT subroutine

The user needs to enter in the input deck the material properties. They are entered under the *USER MATERIAL command (8 properties in every line)

First line: Young' modululus, Poisson ratio, FLD_0, slope of the FLD in the right half plane, slope of the FLD in the left half plane, L_{1111}, L_{2222}, L_{3333}. L is the transformation tensor, see also Chapter 2.

Second line: L_{1212}, L_{2323}, L_{1313}, B_{11}, B_{22}, exponent $2K$, mixing factor c, dummy variable equal to 1.0.

Third line: 4 pairs of stress-strain data from the stress strain curve. First strain given must be zero. Strains should be at least 0.1E-02 apart.

Following lines: Same as the third line until the whole stress-strain curve has been described. Last strain given must be large enough to ensure that its value is not exceeded during the analysis.

The subroutine has 5 state variables:

1st sv: Equivalent yield stress

2nd sv: unimportant, used for bookkeeping

3rd sv: unimportant used for bookkeeping

4th sv: safety margin (distance from the FLD)

5th sv: Equivalent plastic strain.

These state variables can be contoured by using "*contour,var=sdvi" in abaqus post.

In order to obtain L_{ijkl} you need the R-ratio distribution and the yield stress distribution in different directions. The method to obtain L_{ijkl} is described in Chapter 2.

An example input deck for the uniaxial test of a single shell element is provided (uni.inp). The material modeled is a deep-drawing steel.

VUMAT subroutine
\baselineskip0.2in

```
      SUBROUTINE VUMAT(
C Read only (unmodifiable) variables
     &NBLOCK, NDIR, NSHR, NSTATEV, NFIELDV, NPROPS, LANNEAL,
     &STEP_TIME,TOTAL_TIME, DT, CMNAME, COORD_MP, CHAR_LENGTH,
     &PROPS, DENSITY, STRAIN_INC, REL_SPIN_INC,
     &TEMP_OLD, STRETCH_OLD, DEFGRAD_OLD, FIELD_OLD,
     &STRESS_OLD, STATE_OLD, ENER_INTERN_OLD, ENER_INELAS_OLD,
     &TEMP_NEW, STRETCH_NEW, DEFGRAD_NEW, FIELD_NEW,
C WRITE ONLY (modfiable) variables
     &STRESS_NEW, STATE_NEW, ENER_INTERN_NEW, ENER_INELAS_NEW)
C
      INCLUDE 'VABA_PARAM.INC'
C
      DIMENSION  COORD_MP(NBLOCK,*),CHAR_LENGTH(NBLOCK),
     & PROPS(NPROPS),
     &DENSITY(NBLOCK), STRAIN_INC(NBLOCK,NDIR+NSHR),
     &REL_SPIN_INC(NBLOCK,NSHR), TEMP_OLD(NBLOCK),
     &STRETCH_OLD(NBLOCK,NDIR+NSHR),
     &DEFGRAD_OLD(NBLOCK,NDIR+NSHR+NSHR), FIELD_OLD(NBLOCK,NFIELDV),
     &STRESS_OLD(NBLOCK,NDIR+NSHR), STATE_OLD(NBLOCK,NSTATEV),
     &ENER_INTERN_OLD(NBLOCK), ENER_INELAS_OLD(NBLOCK),
     &TEMP_NEW(NBLOCK), STRETCH_NEW(NBLOCK,NDIR+NSHR),
     &DEFGRAD_NEW(NBLOCK,NDIR+NSHR+NSHR), FIELD_NEW(NBLOCK,NFIELDV),
     &STRESS_NEW(NBLOCK,NDIR+NSHR), STATE_NEW(NBLOCK,NSTATEV),
     &ENER_INTERN_NEW(NBLOCK), ENER_INELAS_NEW(NBLOCK)
C
      CHARACTER*8 CMNAME
C
      PARAMETER(ZERO=0., ONE=1., TWO=2., THREE=3.,
     & THIRD= ONE/THREE, HALF=.5, TWO_THIRDS=TWO/THREE,
     & THREE_HALFS=1.5, FOUR=4., VERY_SMALL=1.0E-3,
     & XHARD=0.6E+9,ONEM=-1., VERY_SMALL_3=1.0E-3,
     & VERY_SMALL_4=1.0E-3,VERY_SMALL_7=1.0E-7)
C
      E=PROPS(1)
      XNU=PROPS(2)
      FLD0=PROPS(3)
      FLDS1=PROPS(4)
      FLDS2=PROPS(5)
      XL11=PROPS(6)
      XL22=PROPS(7)
      XL33=PROPS(8)
      XL44=PROPS(9)
      XL55=PROPS(10)
      XL66=PROPS(11)
      B11=PROPS(12)
      B22=PROPS(13)
      EXPM=PROPS(14)
      CC=PROPS(15)
      ALPHA=PROPS(16)
      FIRST_YIELD=PROPS(17)
C obtain a very small stress
      VERY_SMALL_STRESS=VERY_SMALL_7*E
C normalize the mixing coefficient
```

```
      CCC=CC*THREE**EXPM
     &/(TWO**(EXPM-ONE)+ONE)
C initalize state variable (yield stress)
C      WRITE(*,*)TOTAL_TIME,DT
C evaluate the remaining components of the IPE transformation tensor
      XL12=(XL33-XL11-XL22)/TWO
      XL13=(XL22-XL11-XL33)/TWO
      XL23=(XL11-XL22-XL33)/TWO
      B33=-B11-B22
C calculate the elastic constants (isotropic material)
      TWOMU=E/(ONE+XNU)
      THREMU=THREE_HALFS*TWOMU
      SIXMU=THREE*TWOMU
      ALAMDA=TWOMU*(E-TWOMU)/(SIXMU-TWO*E)
C
      DO 100 I=1,NBLOCK
C Obtain actual value of the yield stress
C
      NAD=0
      TESTVAR=TOTAL_TIME-DT
      FFLAG=ZERO
      IF(ABS(TOTAL_TIME-ONE).LT.1.0E-6) FFLAG=ONE
      IF((TESTVAR-VERY_SMALL_7).LT.ZERO) FFLAG=ONE
      IF(TOTAL_TIME.LT.VERY_SMALL_7) FFLAG=ONE
      NIN=INT(ONE+VERY_SMALL_3)*FFLAG+(ONE-FFLAG)
     &*STATE_OLD(I,2)
      FSEV=FFLAG*FLDO+(ONE-FFLAG)*STATE_OLD(I,4)
      YIELD=FFLAG*FIRST_YIELD+
     &(ONE-FFLAG)*STATE_OLD(I,1)
      EPL=(ONE-FFLAG)*STATE_OLD(I,5)
      BYIELD=YIELD
      YRAT=YIELD/FIRST_YIELD
      SB11=B11*YRAT/BYIELD
      SB22=B22*YRAT/BYIELD
      SB33=B33*YRAT/BYIELD
      YIELD=ONE
      YFUNCEQ=TWO
C
C Trial stress
C
C First make a guess for the 33-strain (elastic)
      GSTRAIN_INC_3=-XNU/(ONE-XNU)*(STRAIN_INC(I,1)+
     &STRAIN_INC(I,2))
      TRACE = STRAIN_INC(I,1)+ STRAIN_INC(I,2)+GSTRAIN_INC_3
      SIG1=(STRESS_OLD(I,1)+ALAMDA*TRACE+TWOMU*STRAIN_INC
     &(I,1))/BYIELD
      SIG2=(STRESS_OLD(I,2)+ALAMDA*TRACE+TWOMU*STRAIN_INC
     &(I,2))/BYIELD
      SIG4=(STRESS_OLD(I,4)+TWOMU*STRAIN_INC(I,4))
     &/BYIELD
C Calculate IPE stress
      SIG_IPE1=XL11*(SIG1-SB11)+XL12*(SIG2-SB22)
      SIG_IPE2=XL12*(SIG1-SB11)+XL22*(SIG2-SB22)
      SIG_IPE3=XL13*(SIG1-SB11)+XL23*(SIG2-SB22)
      SIG_IPE4=XL44*SIG4
C Calculate eigenvalues
      FTHREE=ZERO
```

```
      ROOT3=SIG_IPE3
      XINV3=SIG_IPE1*SIG_IPE2-
     &SIG_IPE4*SIG_IPE4
      XTERM=SQRT(ABS
     &(ROOT3*ROOT3/FOUR-XINV3))
      ROOT1=(-ROOT3/TWO+XTERM)
      ROOT2=(-ROOT3/TWO-XTERM)
      BNORM=ABS(ROOT1)+ABS(ROOT2)+ABS(ROOT1+ROOT2)
      FNORM=0.
      ANORM=BNORM+VERY_SMALL_3
      BROOT1=ROOT1/ANORM
      BROOT2=ROOT2/ANORM
      ROOT3=-ROOT1-ROOT2
      BROOT3=(-ROOT1-ROOT2)/ANORM
      BYFUNC=(ONE-CC)*(ABS(BROOT1-BROOT2)**EXPM+
     &ABS(BROOT2-BROOT3)**EXPM+
     &ABS(BROOT3-BROOT1)**EXPM)+CC*THREE**EXPM
     &/(TWO**(EXPM-ONE)+ONE)*(ABS(BROOT1)**EXPM+
     &ABS(BROOT2)**EXPM+ABS(BROOT3)**EXPM)
      TYFUNC=ANORM*(RYFUNC/TWO)**(ONE/EXPM)
C  Check for plasticity
      FACYLD=ONE
      COELA=ZERO
      IF((TWO*TYFUNC-YFUNCEQ).LT.ZERO) THEN
      OFUNC=0.
      TER1=0.
      WFUNC=0.
      slope11=0.
      slope22=0.
      slope12=0.
      SLOPE33=0.
      SLOC=1.0E-5
      COELA=ONE
      SEIGEN1=ZERO
      SEIGEN2=ZERO
      FAILFLAG=ZERO
      GOTO 999
      END IF
      IF(ABS(TOTAL_TIME-ONE).LT.VERY_SMALL_7) FACYLD=ZERO
C
C  Calculate F (WFUNC) function (dot W= F*dot Y)
C
      IF(EPL.GE.PROPS(2*(9+NIN)))  NAD=1
      NIN=NIN+NAD
      SLOC=(PROPS(2*(9+NIN)-1)-PROPS(2*(8+NIN)-1))/
     &(PROPS(2*(9+NIN))-PROPS(2*(8+NIN)))+VERY_SMALL_3
      WFUNC=ONE/TWO/EXPM/SLOC
      YN=TYFUNC+VERY_SMALL_3
      SIG_IPE1_OLD=SIG_IPE1/YN
      SIG_IPE2_OLD=SIG_IPE2/YN
      SIG_IPE3_OLD=SIG_IPE3/YN
      SIG_IPE4_OLD=SIG_IPE4/YN
      PROOT1=ROOT1/YN
      PROOT2=ROOT2/YN
      PROOT3=ROOT3/YN
C Find the eigenvectors of the IPE stress tensor
      V11=ONE
```

```
      ADTER=SIGN(VERY_SMALL_3,SIG_IPE4_OLD)
      V21=(PROOT1-SIG_IPE1_OLD)/(SIG_IPE4_OLD+ADTER)
      V31=ZERO
      V12=ONE*SIG_IPE4_OLD
      V22=(PROOT2-SIG_IPE1_OLD)
      V32=ZERO
      V13=ZERO
      V23=ZERO
      V33=ONE
      XMAG1=SQRT(ABS(V11*V11+V21*V21+V31*V31))
      XMAG3=SQRT(ABS(V13*V13+V23*V23+V33*V33))
      PV11=V11/XMAG1
      PV21=V21/XMAG1
      PV31=V31/XMAG1
      PV13=V13/XMAG3
      PV23=V23/XMAG3
      PV33=V33/XMAG3
      pv12=pv23*pv31-pv21*pv33
      pv22=pv33*pv11-pv31*pv13
      pv32=pv13*pv21-pv11*pv23
C Calclulate IPE slope (principal directions)
      XSIGN1=ONE
      XSIGN2=ONE
      XSIGN3=ONE
      XSIGN4=ONE
      XSIGN5=ONE
      XSIGN6=ONE
      IF(PROOT1.LT.ZERO) XSIGN1=ONEM
      IF(PROOT2.LT.ZERO) XSIGN2=ONEM
      IF(PROOT3.LT.ZERO) XSIGN3=ONEM
      IF((PROOT1-PROOT2).LT.ZERO) XSIGN4=ONEM
      IF((PROOT2-PROOT3).LT.ZERO) XSIGN5=ONEM
      IF((PROOT3-PROOT1).LT.ZERO) XSIGN6=ONEM
      phi1=ccc*expm*(abs(PROOT1))**(expm-ONE)
     &*xsign1+(ONE-CC)*expm*
     &(abs(PROOT1-PROOT2))**(expm-ONE)*
     &xsign4-(ONE-CC)*expm*
     &(abs(PROOT3-PROOT1))**(expm-ONE)*
     &xsign6
      phi2=ccc*expm*(abs(PROOT2))**(expm-ONE)
     &*xsign2-(ONE-CC)*expm*
     &(abs(PROOT1-PROOT2))**(expm-ONE)*
     &xsign4+(ONE-CC)*expm*
     &(abs(PROOT2-PROOT3))**(expm-ONE)*
     &xsign5
      phi3=ccc*expm*(abs(PROOT3))**(expm-ONE)
     &*xsign3-(ONE-CC)*expm*
     &(abs(PROOT2-PROOT3))**(expm-ONE)*
     &xsign5+(ONE-CC)*expm*
     &(abs(PROOT3-PROOT1))**(expm-ONE)*
     &xsign6
C Calculate IPE slope
      cslope11=pv11*pv11*phi1+pv12*pv12*phi2
     &+pv13*pv13*phi3
      cslope12=pv11*pv21*phi1+pv12*pv22*phi2
     &+pv13*pv23*phi3
```

```
      cslope21=cslope12
      cslope22=pv21*pv21*phi1+pv22*pv22*phi2
     &+pv23*pv23*phi3
      cslope33=phi3
C Calculate slope
      slope11=xl11*cslope11+xl12*cslope22
     &+xl13*cslope33
      slope12=xl44*cslope12
      slope21=slope12
      slope22=xl12*cslope11+xl22*cslope22
     &+xl23*cslope33
      slope33=xl13*cslope11+xl23*cslope22
     &+xl33*cslope33
C Operate elasticity on slope of yield surface
C The result is ELPLST
      ELPLST11=TWOMU*SLOPE11
      ELPLST12=TWOMU*SLOPE12
      ELPLST21=ELPLST12
      ELPLST22=TWOMU*SLOPE22
      ELPLST33=TWOMU*SLOPE33
C
C Calculate dot product between slope of yield surface
C and the tensor ELPLST
C
      XDPROTDEN=SLOPE11*ELPLST11+SLOPE12*ELPLST12
     &+SLOPE21*ELPLST21+SLOPE22*ELPLST22+
     &SLOPE33*ELPLST33
C
C Now we calculate the value of the 33 component of the strain increment
C tensor (this component is called GSTRAIN_INC_3)
      XDENOM1=TWO*EXPM
      XDENOM2=WFUNC*XDPROTDEN
      OFUNC=FACYLD/(XDENOM1+XDENOM2)
      TER1=SLOPE11*
     &TWOMU*
     &STRAIN_INC(I,1)
     &+SLOPE12*TWO*TWOMU*
     &STRAIN_INC(I,4)+SLOPE22*
     &TWOMU*
     &STRAIN_INC(I,2)
      XXNOM=OFUNC*TER1*WFUNC*TWOMU*SLOPE33
     &-ALAMDA*(STRAIN_INC(I,1)
     &+STRAIN_INC(I,2))
      XXDENOM1=ALAMDA+TWOMU
      XXDENOM2=FACYLD*TWOMU*TWOMU*SLOPE33
     &*SLOPE33*WFUNC*OFUNC
      GSTRAIN_INC_3=XXNOM/(XXDENOM1-XXDENOM2)
C Calculate principal strains
      CRAZY1=(STRETCH_NEW(I,1)-STRETCH_NEW(I,2))/TWO
      CRAZY2=(STRETCH_NEW(I,1)+STRETCH_NEW(I,2))/TWO
      CRAZY3=STRETCH_NEW(I,4)
      CRAZY4=SQRT(CRAZY1*CRAZY1+CRAZY3*CRAZY3)
      SEIGEN1=LOG(CRAZY2+CRAZY4)
      SEIGEN2=LOG(CRAZY2-CRAZY4)
C Calculate forming severity
      FAILFLAG=-ONE
      IF(SEIGEN2.GT.ZERO) FAILFLAG=ZERO
```

```
  999      STRAIN_INC(I,3)=GSTRAIN_INC_3
C
C Now we calculate the stress increment tensor
C
C calculate trace of the strain increment tensor
       TTRACE=STRAIN_INC(I,1)+STRAIN_INC(I,2)
     &+GSTRAIN_INC_3
C calculate yield stress increment DYIELD
       TER2=TER1+SLOPE33*TWOMU*GSTRAIN_INC_3
       DYIELD=TER2*OFUNC
C Operate elasticity tensor on STRAIN_INC and get stress increment
       DSIG1=ALAMDA*TTRACE+TWOMU*(STRAIN_INC(I,1)
     &-DYIELD*WFUNC*SLOPE11)
       DSIG2=ALAMDA*TTRACE+TWOMU*(STRAIN_INC(I,2)
     &-DYIELD*WFUNC*SLOPE22)
       DSIG4=TWOMU*(STRAIN_INC(I,4)
     &-DYIELD*WFUNC*SLOPE12)
C Update stress
       STRESS_NEW(I,1)=STRESS_OLD(I,1)+DSIG1
       STRESS_NEW(I,2)=STRESS_OLD(I,2)+DSIG2
       STRESS_NEW(I,4)=STRESS_OLD(I,4)+DSIG4
C Update yield stress
       STATE_NEW(I,1)=(ONE-FFLAG)*STATE_OLD(I,1)
     &+DYIELD+FFLAG*FIRST_YIELD
        DEPL=DYIELD/SLOC
       STATE_NEW(I,5)=(ONE-FFLAG)*STATE_OLD(I,5)
     &+DEPL
C  Update internal energy
       STRESS_POWER= HALF*(
     &(STRESS_OLD(I,1)+STRESS_NEW(I,1))*STRAIN_INC(I,1)
     &+(STRESS_OLD(I,2)+STRESS_NEW(I,2))*STRAIN_INC(I,2)
     &+TWO*(STRESS_OLD(I,4)+STRESS_NEW(I,4))*STRAIN_INC(I,4))
C
       STATE_NEW(I,3)=NAD
       STATE_NEW(I,2)=NIN
       ENER_INTERN_NEW(I)=ENER_INTERN_OLD(I)
     &    +STRESS_POWER/DENSITY(I)
C  Update dissipated energy
       PLASTIC_WORK_INC=ONE/TWO*DYIELD*DEPL
     &+BYIELD*DEPL
       ENER_INELAS_NEW(I)=ENER_INELAS_OLD(I)
     &+PLASTIC_WORK_INC/DENSITY(I)
       STATE_NEW(I,4)=COELA*FSEV+(ONE-COELA)*(FLDO-SEIGEN1
     &+(FAILFLAG+ONE)*FLDS1*SEIGEN2
     &-FAILFLAG*FLDS2*SEIGEN2)
  100 CONTINUE
       RETURN
       END
```

INPUT DECK UNI.INP

```
*HEADING
A SINGLE SHELL ELEMENT WITH MY VUMAT
*NODE
10001, 0.0, 0.0
10031,110.0E-03,0.0
13001,0.0,110.0E-03
13031,110.0E-03,110.0E-03
*NSET,NSET=LEFT
10001,13001
*NSET,NSET=BOTTOM
10001,10031
*NSET,NSET=TOP
13001,13031
*NSET,NSET=RIGHT
10031,13031
*NSET,NSET=ALL
LEFT,BOTTOM,TOP,RIGHT
*ELEMENT,TYPE=S4RF
10001,10001,10031,13031,13001
*ELSET, ELSET=BLANK
10001
*SHELL SECTION,ELSET=BLANK,MATERIAL=ALUMINUM
1.05E-03,5
*MATERIAL,NAME=ALUMINUM
*DENSITY
7800.
*USER MATERIAL,CONSTANTS=120
206.E+9,0.33,0.2658,0.1,-0.1,0.6666,0.6666,0.6666
1.00,1.0000,1.0000,0.0,0.0,10.,0.5,1.0
178.0E+06,0.0,0.179E+09,0.189E-02,0.180E+09,0.193E-02,0.181E+09,0.200E-02
0.182E+09,0.265E-02,0.183E+09,0.271E-02,0.184E+09,0.277E-02,0.185E+09,0.281E-02
0.203E+09,0.973E-02,0.221E+09,0.161E-01,0.235E+09,0.224E-01,0.249E+09,0.286E-01
0.260E+09,0.348E-01,0.270E+09,0.409E-01,0.280E+09,0.471E-01,0.287E+09,0.531E-01
0.295E+09,0.592E-01,0.302E+09,0.653E-01,0.309E+09,0.714E-01,0.316E+09,0.775E-01
0.321E+09,0.836E-01,0.327E+09,0.898E-01,0.332E+09,0.960E-01,0.336E+09,0.102E+00
0.341E+09,0.108E+00,0.345E+09,0.115E+00,0.349E+09,0.121E+00,0.352E+09,0.127E+00
0.356E+09,0.134E+00,0.360E+09,0.140E+00,0.363E+09,0.146E+00,0.366E+09,0.153E+00
0.369E+09,0.159E+00,0.372E+09,0.165E+00,0.375E+09,0.172E+00,0.378E+09,0.178E+00
0.381E+09,0.185E+00,0.383E+09,0.191E+00,0.386E+09,0.198E+00,0.389E+09,0.204E+00
0.391E+09,0.211E+00,0.394E+09,0.217E+00,0.397E+09,0.224E+00,0.398E+09,0.231E+00
0.402E+09,0.237E+00,0.404E+09,0.244E+00,0.405E+09,0.250E+00,0.408E+09,0.257E+00
0.410E+09,0.262E+00,0.411E+09,0.268E+00,0.412E+09,0.272E+00,0.513E+09,3.275E+00
*DEPVAR
5
*BOUNDARY
LEFT,1,1
BOTTOM,2,2
ALL,3,6
**TOP,2,2
*RESTART,WRITE,NUMBER INTERVAL=50
*STEP
*DYNAMIC,EXPLICIT
,0.005
***BULK VISCOSITY
```

```
**0.1,1.
*BOUNDARY,TYPE=VELOCITY
RIGHT,1,1,2.
*FILE OUTPUT,NUMBER INTERVAL=50
*EL FILE
S,E
*NODE FILE
COORD,U,V,A,RF
*END STEP
*STEP
*DYNAMIC,EXPLICIT
,0.005
***BULK VISCOSITY
**0.1,1.
*BOUNDARY,TYPE=VELOCITY
RIGHT,1,1,-2.
*FILE OUTPUT,NUMBER INTERVAL=50
*EL FILE
S,E
*NODE FILE
COORD,U,V,A,RF
*END STEP
*STEP
*DYNAMIC,EXPLICIT
,0.005
***BULK VISCOSITY
**0.1,1.
*BOUNDARY,TYPE=VELOCITY
RIGHT,1,1,2.
*FILE OUTPUT,NUMBER INTERVAL=50
*EL FILE
S,E
*NODE FILE
COORD,U,V,A,RF
*END STEP
```

11 Appendix IV: Software for data transfer from explicit to implicit analysis

```
C*****************************************************************
C             FILE OUTPUT PROGRAM                                *
C         Obtain output from the .fil file                       *
C         evaluation.                                            *
C                                                                *
C   This  program was developed at the Mechanics and Materials   *
C   group of MIT.                                                *
C                                                 09/03/93       *
C         Modified by Apostolos Karafillis for the HP735         *
C*****************************************************************
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION JRRAY(2,512),ARRAY(512),NPL(9),XMISES(9),lrunit(2,1)
      EQUIVALENCE (ARRAY(1),JRRAY(1,1))
      DATA NUMSTEP/0/,NUMINC/1/,YTOTAL/0.0/
C apost
      character*80 fname
      character*10 flag
      common /jobpar/ /fname
C apost
C
      WRITE(6,'('' INPUT OUTPUT VALUABLE ''/
     1''   1:STRESS   2:DISPLACEMENT   3:ACCELARATION    4:VELOCITY''/
     2''   5:ENERGY   6:MISES STRESS   7:REACTION FORCE  8:N FORCE ''/
     3''   9:STRAIN  10:STATE VARIABLES  11:TEMPERATURE  12:coord  ''/
     4''   13:node_cord  14:PLASTIC STRAIN  ''/
     5''   15:ELASTIC STRAIN  16:section forces 20:EVERYTHING  '')')
      READ(5,*) IPL
C---- STEP AND INCREMENT NUMBER, SET NUMSTEP=0 IF ALL INCREMENTS
      WRITE(6,'('' INPUT STEP NUMBER  0 IF ALL INCREMENTS  '')')
      READ(5,*) NUMSTEP
      IF(NUMSTEP.NE.0) THEN
       WRITE(6,'('' INPUT INCREMENT NUMBER                  '')')
       READ(5,*) NUMINC
      ENDIF
      IF (IPL .EQ. 7) THEN
       WRITE(6,'('' INPUT LARGEST NODE NUMBER ON THE TOP    '')')
       READ (5,*) NSTOP
      ENDIF
C---- CALL TO OPEN FILES
      WRITE(6,'('' INPUT FILE.OUT W/O .08  '')')
C apost
      nru=1
      lrunit(1,1)=8
      lrunit(2,1)=2
      read(*,374)fname
  374 format(A80)
C apost
      CALL OPFILE(IPL)
      CALL INITPF(fname,nru,lrunit,loutf)
      junit=8
```

```
      call dbrnu(junit)
      JRCD = 0
C---- CALL FIXVAL
      DO 100 K1=1,1000000
      CALL DBFILE(0,ARRAY(1),JRCD)
      IF(JRCD.NE.0) GO TO 101
      KEY=JRRAY(1,2)
      IF(KEY.NE.2000) GOTO 20
      TIME  = ARRAY(3)
      NSTEP = JRRAY(1,8)
      INC   = JRRAY(1,9)
 20   CONTINUE
      IF(NUMSTEP.EQ.0) GOTO 30
      IF((NSTEP.EQ.NUMSTEP.AND.INC.GT.NUMINC).OR.(NSTEP.GT.NUMSTEP))
     1                                                          GOTO 101
      IF(NSTEP.NE.NUMSTEP.OR.INC.NE.NUMINC) GOTO 100
 30   CONTINUE
C
      IF(KEY.NE.1) GOTO 115
      NELEM = JRRAY(1,3)
      NPT   = JRRAY(1,4)
C 110  IF(((IPL.NE.1).AND.(IPL.NE.20)).OR.(KEY.NE.11)) GOTO 115
C      SX  = ARRAY(3)
C      SY  = ARRAY(4)
C      SZ  = ARRAY(5)
C      SXY = ARRAY(6)
C      SXZ = ARRAY(7)
C      SYZ = ARRAY(8)
C      WRITE(41,1000) NELEM,NPT,SX,SY,SZ,SXY,SXZ,SYZ
 115  IF(((IPL.NE.12).AND.(IPL.NE.20)).OR.(KEY.NE.8)) GOTO 116
      coord1  = ARRAY(3)
      coord2  = ARRAY(4)
      coord3  = ARRAY(5)
      WRITE(61,1111) coord1,coord2
 116  IF(((IPL.NE.9).AND.(IPL.NE.20)).OR.(KEY.NE.21)) GOTO 120
      EX  = ARRAY(3)
      EY  = ARRAY(4)
      EZ  = ARRAY(5)
      EXY = ARRAY(6)
C comm by apost     EXZ = ARRAY(7)
C comm by apost     EYZ = ARRAY(8)
      WRITE(49,1311) NELEM,NPT,EX,EY,EZ,EXY,EXZ,EYZ
 120  IF(((IPL.NE.2).AND.(IPL.NE.20)).OR.(KEY.NE.101)) GOTO 121
      NODE  = JRRAY(1,3)
      XDISP = ARRAY(4)
      YDISP = ARRAY(5)
      ZDISP = ARRAY(6)
      R1DISP = ARRAY(7)
      R2DISP = ARRAY(8)
      R3DISP = ARRAY(9)
      WRITE(42,1114) NODE,XDISP,YDISP,ZDISP,R1DISP,R2DISP,R3DISP
 121  IF(((IPL.NE.13).AND.(IPL.NE.20)).OR.(KEY.NE.107)) GOTO 122
      NODE  = JRRAY(1,3)
      XCOORD = ARRAY(4)
      YCOORD = ARRAY(5)
      ZCOORD = ARRAY(6)
C      ROT1 = ARRAY(7)
```

```
C       ROT2 = ARRAY(8)
C       ROT3 = ARRAY(9)
C       WRITE(62,1114) NODE,XCOORD,YCOORD,ZCOORD,ROT1,ROT2,ROT3
        WRITE(62,1112) NODE,XCOORD,YCOORD,ZCOORD
  122   IF(((IPL.NE.11).AND.(IPL.NE.20)).OR.(KEY.NE.201)) GOTO 123
        NODE   = JRRAY(1,3)
        TEMP   = ARRAY(4)
        WRITE(60,1100) NODE,TEMP
  123   IF (KEY.NE.1) GOTO 125
        NELEM = JRRAY(1,3)
        NPT = JRRAY(1,4)
  125   IF(((IPL.NE.10).AND.(IPL.NE.20)).OR.(KEY.NE.5)) GOTO 130
        SDV1 = ARRAY(3)
        SDV2 = ARRAY(4)
        WRITE(50,1000) NELEM,NPT,SDV1,SDV2
  130   IF(((IPL.NE.3).AND.(IPL.NE.20)).OR.(KEY.NE.103)) GOTO 140
        NODE = JRRAY(1,3)
        XACC = ARRAY(4)
        YACC = ARRAY(5)
        WRITE(43,1100) NODE,XACC,YACC
  140   IF(((IPL.NE.4).AND.(IPL.NE.20)).OR.(KEY.NE.102)) GOTO 150
        NODE = JRRAY(1,3)
        XVEL = ARRAY(4)
        YVEL = ARRAY(5)
        WRITE(44,1100) NODE,XVEL,YVEL
  150   IF(((IPL.NE.5).AND.(IPL.NE.20)).OR.(KEY.NE.14)) GOTO 155
        EW   = ARRAY(3)
        SE   = ARRAY(4)
        PD   = ARRAY(5)
        WRITE(45,1100) NELEM,EW,SE,PD
  155   IF(((IPL.NE.1).AND.(IPL.NE.20)).OR.(KEY.NE.11)) GOTO 160
        SX  = ARRAY(3)
        SY  = ARRAY(4)
        SZ  = ARRAY(5)
        SXY = ARRAY(6)
        SXZ = ARRAY(7)
        SYZ = ARRAY(8)
        WRITE(41,1000) NELEM,NPT,SX,SY,SZ,SXY,SXZ,SYZ
  160   IF(((IPL.NE.6).AND.(IPL.NE.20)).OR.(KEY.NE.12)) GOTO 170
        XMISES(NPT) = ARRAY(3)
        IF(NPT.NE.4) GOTO 100
        WRITE(46,1100) NELEM,XMISES(1),XMISES(2),XMISES(4),XMISES(3)
  170   IF(((IPL.NE.7).AND.(IPL.NE.20)).OR.(KEY.NE.104)) GOTO 180
        NODE   = JRRAY(1,3)
        XFORCE = ARRAY(4)
        YFORCE = ARRAY(5)
        ZFORCE = ARRAY(6)
C       WRITE(47,1100) NODE,XFORCE,YFORCE,ZFORCE
          YTOTAL = YTOTAL + YFORCE
         IF(NODE.EQ.NSTOP) THEN
          WRITE(47,1000) NSTEP,INC,YTOTAL
          YTOTAL = 0.0
         ENDIF
  180   IF(((IPL.NE.8).AND.(IPL.NE.20)).OR.(KEY.NE.15)) GOTO 299
        LNODE  = JRRAY(1,3)
        XFORCE = ARRAY(4)
        YFORCE = ARRAY(5)
```

```
      ZFORCE = ARRAY(6)
      WRITE(48,1000) NELEM,LNODE,XFORCE,YFORCE,ZFORCE
 299  IF(((IPL.NE.14).AND.(IPL.NE.20)).OR.(KEY.NE.22)) GOTO 2995
      errp=array(3)
      ezzp=array(4)
      ehhp=array(5)
      erzp=array(6)
C      flag=array(7)
      write(63,1311)nelem,npt,errp,ezzp,ehhp,erzp
 2995 IF(((IPL.NE.16).AND.(IPL.NE.20)).OR.(KEY.NE.13)) GOTO 300
      sx=array(3)
      sy=array(4)
      sxy=array(5)
      smx=array(8)
      smy=array(9)
      smxy=array(10)
      write(64,1113) nelem, sx,sy,sxy,smx,smy,smxy
 300  CONTINUE
 100  CONTINUE
 101  CONTINUE
      STOP
 1000 FORMAT(2I7,6(1PG16.8))
 1100 FORMAT(I8,5(1PG16.8))
 1113 FORMAT(I8,6(1PG16.8))
 1111 format(2(1PG16.8))
 1311 format(2I5,4(1PG16.8))
 1312 format(2I5,3(1PG16.8))
 1112 format(I8,3(',',1PG16.8))
 1114 format(I8,6(',',1PG16.8))
      END
C
      SUBROUTINE OPFILE(IPL)
      IMPLICIT REAL*8(A-H,O-Z)
C---- OPEN OUTPUT FILES
      IF(IPL.EQ.1.OR.IPL.EQ.20) OPEN(UNIT=41,FILE='stress.out')
      IF(IPL.EQ.2.OR.IPL.EQ.20) OPEN(UNIT=42,FILE='disp.out')
      IF(IPL.EQ.3.OR.IPL.EQ.20) OPEN(UNIT=43,FILE='acc.out')
      IF(IPL.EQ.4.OR.IPL.EQ.20) OPEN(UNIT=44,FILE='vel.out')
      IF(IPL.EQ.5.OR.IPL.EQ.20) OPEN(UNIT=45,FILE='energy.out')
      IF(IPL.EQ.6.OR.IPL.EQ.20) OPEN(UNIT=46,FILE='mises.out')
      IF(IPL.EQ.7.OR.IPL.EQ.20) OPEN(UNIT=47,FILE='rforce.out')
      IF(IPL.EQ.8.OR.IPL.EQ.20) OPEN(UNIT=48,FILE='nforce.out')
      IF(IPL.EQ.9.OR.IPL.EQ.20) OPEN(UNIT=49,FILE='strain.out')
      IF(IPL.EQ.10.OR.IPL.EQ.20) OPEN (UNIT=50,FILE='sdv.out')
      IF(IPL.EQ.11.OR.IPL.EQ.20) OPEN (UNIT=60,FILE='temp.out')
      IF(IPL.EQ.12.OR.IPL.EQ.20) OPEN (UNIT=61,FILE='coord.out')
      IF(IPL.EQ.13.OR.IPL.EQ.20) OPEN (UNIT=62,FILE='nd_cod.out')
      IF(IPL.EQ.14.OR.IPL.EQ.20) OPEN (UNIT=63,FILE='pstrain.out')
      IF(IPL.EQ.16.OR.IPL.EQ.20) OPEN (UNIT=64,FILE='sforce.out')
      RETURN
      END
      SUBROUTINE DUMMY
      ENTRY ZZERO
      ENTRY PLENDT
      END
```

```
C  This program reads the shell section forces and moments created from
C  Explicit and writes the source code for the UMAT subroutine subsequently
C  used in the implicit analysis
      program sw
      open(unit=1,file='input',status='old',access='sequential')
      open(unit=2,file='source',access='sequential')
      write(6,*)'Enter shell thickness:'
      read(5,*)xh
      write(6,*)'Enter number of integration pts'
      read(5,*)ntot
      zero=0.
      nzero=0
      nsflag=1
      do i=1,100000
      read(1,*,end=999) ne,fx,fy,fxy,xmy,xmx,xmxy
      do npt=1,ntot
      j=npt
      call calc(fx,xmy,s,j,ntot,xh,nzero)
      write(2,1)  ne,npt,s
      call calc(fy,xmx,s,j,ntot,xh,nzero)
      write(2,2)  ne,npt,s
      call calc(fxy,xmxy,s,j,ntot,xh,nzero)
      write(2,3)  ne,npt,s
      end do
      end do
    1 format(6x,'if (noel.eq.',i7,'.and.kspt.eq.',i1,')',/,5x,
     &'&stress(1)=',1PG16.8)
    2 format(6x,'if (noel.eq.',i7,'.and.kspt.eq.',i1,')',/,5x,
     &'&stress(2)=',1PG16.8)
    3 format(6x,'if (noel.eq.',i7,'.and.kspt.eq.',i1,')',/,5x,
     &'&stress(3)=',1PG16.8)
  999 stop
      end

      subroutine calc(f,xm,s,npt,ntot,xh,nsflag)
       xnpt=float(npt)
       xntot=float(ntot)
       xk=12.*xm/xh/xh/xh
       xonem=-1.
C       write(*,*)xk
       s0=f/xh*float(nsflag)
       s=xonem*(s0+(xnpt-(xntot+1.)/2.)/(xntot-1.)*xk*xh)
      return
      end
```

```
C   UMAT subroutine for the implicit ''springforward'' analysis
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     &RPL,DDSDDT,DRPLDE,DRPLDT,
     &STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     &NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     &CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*8 CMNAME
      DIMENSION STRESS(NTENS), STATEV(NSTATV),
     &DDSDDE(NTENS,NTENS), DDSDDT(NTENS), DRPLDE(NTENS),
     &STRAN(NTENS), DSTRAN(NTENS), TIME(2), PREDEF(1), DPRED(1),
     &PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3)
C
      PARAMETER(ZERO=0., ONE=1., TWO=2., THREE=3.,
     & THIRD= ONE/THREE, HALF=.5, TWO_THIRDS=TWO/THREE,
     & THREE_HALFS=1.5, FOUR=4., VERY_SMALL=1.0E-3,
     & XHARD=0.6E+9,ONEM=-1., VERY_SMALL_3=1.0E-3,
     & VERY_SMALL_4=1.0E-3,VERY_SMALL_7=1.0E-7)
C
      E=PROPS(1)
      XNU=PROPS(2)
C      DO I=1,100
C      PSTRE(I)=PROPS(16+2*i-1)
C      PSTRA(I)=PROPS(16+2*i)
C      END DO
C calculate the elastic constants (isotropic material)
      TWOMU=E/(ONE+XNU)
      THREMU=THREE_HALFS*TWOMU
      SIXMU=THREE*TWOMU
      ALAMDA=TWOMU*(E-TWOMU)/(SIXMU-TWO*E)
C
C      DO 100 I=1,NBLOCK
C      write(*,*) strain_inc(i,1),strain_inc(i,2),
C     &strain_inc(i,3),strain_inc(i,4)
C Obtain actual value of the yield stress
C
C     This line is only to correct a mistake in the stress-strain data of
C     the input deck
C      NMALA=1
C
C      STATE_NEW(I,4)=0.
C      NAD=0
C      TESTVAR=TOTAL_TIME-DT
C      FFLAG=ZERO
C      IF(ABS(TOTAL_TIME-ONE).LT.1.0E-6) FFLAG=ONE
C      IF((TESTVAR-VERY_SMALL_7).LT.ZERO) FFLAG=ONE
C      IF(TOTAL_TIME.LT.VERY_SMALL_7) FFLAG=ONE
C      NIN=INT(ONE+VERY_SMALL_3)*FFLAG+(ONE-FFLAG)
C     &*STATE_OLD(I,2)
C      FSEV=FFLAG*FLD0+(ONE-FFLAG)*STATE_OLD(I,4)
C      YIELD=FFLAG*FIRST_YIELD+
C     &(ONE-FFLAG)*STATE_OLD(I,1)
C      EPL=(ONE-FFLAG)*STATE_OLD(I,5)
C      BYIELD=YIELD
C      YRAT=YIELD/FIRST_YIELD
```

```
C       SB11=B11*YRAT/BYIELD
C       SB22=B22*YRAT/BYIELD
C       SB33=B33*YRAT/BYIELD
C       YIELD=ONE
CC       WRITE(*,*) TOTAL_TIME,DT
C       YFUNCEQ=TWO
C
C Trial stress
C
C First make a guess for the 33-strain (elastic)
       if(kstep.eq.1) then
       STRESS(1)=ZERO
       STRESS(2)=ZERO
       STRESS(3)=ZERO
       DDSDDE(1,1)=ZERO
       DDSDDE(1,2)=ZERO
       DDSDDE(1,3)=ZERO
       DDSDDE(2,1)=ZERO
       DDSDDE(2,2)=ZERO
       DDSDDE(2,3)=ZERO
       DDSDDE(3,1)=ZERO
       DDSDDE(3,2)=ZERO
       DDSDDE(3,3)=ZERO
       return
       END IF
       if(kstep.eq.3) then
       GSTRAIN_INC_3=-XNU/(ONE-XNU)*(DSTRAN(1)+
      &DSTRAN(2))
       TRACE = DSTRAN(1)+ DSTRAN(2)+ GSTRAIN_INC_3
       STOL1=STRESS(1)
       STOL2=STRESS(2)
       STOL3=STRESS(3)
       STOL4=STRESS(4)
       SIG1=STOL1+ALAMDA*TRACE+TWOMU*DSTRAN(1)
       SIG2=STOL2+ALAMDA*TRACE+TWOMU*DSTRAN(2)
       SIG3=STOL3+TWOMU*DSTRAN(3)
       DDSDDE(1,1)=ALAMDA*(ONE-XNU/(ONE-XNU))+TWOMU
       DDSDDE(1,2)=ALAMDA*(ONE-XNU/(ONE-XNU))
       DDSDDE(1,3)=ZERO
       DDSDDE(2,1)=DDSDDE(1,2)
       DDSDDE(2,2)=DDSDDE(1,1)
       DDSDDE(2,3)=ZERO
       DDSDDE(3,1)=ZERO
       DDSDDE(3,2)=ZERO
       DDSDDE(3,3)=TWOMU
       STRESS(1)=SIG1
       STRESS(2)=SIG2
       STRESS(3)=SIG3
       RETURN
       end if
       if (kstep.eq.2) then
C  Insert here code generated from program sw
       return
       end if
       end
```

```
C  Input deck to drive the UMAT used in the ''springforward'' analysis.

*HEADING
New 3D problem with lots of springback
*NODE,NSET=MATED
200001,0.699D-01,0.0D-03,-3.805302D-03
200014,34.D-03,0.0D-03,-3.805302D-03
200016,27.D-03,0.0D-03,-3.805302D-03
200021,13.D-03,0.0D-03,-3.805302D-03
200025,0.0,0.0D-03,-3.805302D-03
*NGEN,NSET=MLINE1
200001,200014,1
*NGEN,NSET=MLINE2
200014,200016,1
*NGEN,NSET=MLINE3
200016,200021,1
*NGEN,NSET=MLINE4
200021,200025,1
*NSET,NSET=MLINE,GENERATE
200001,200025,1
***NODE,NSET=BINDER
**899996,-70.0D-03,-30.D-03,0.
**899997,70.0D-03,-30.D-03,0.
**899998,70.0D-03,110.0D-03,0.
**899999,-70.0D-03,110.0D-03,0.
***NCOPY,SHIFT,NEW SET=MIDMAT,OLD SET=MLINE,CHANGE NUMBER=15000
**0.,0.,0.,0.,0.,0.
**0.,0.,1.
*NCOPY,SHIFT,NEW SET=EDGMAT,OLD SET=MLINE,CHANGE NUMBER=30000
0.,87.1557D-03,0.,0.,0.,0.
0.,0.,1.
*NFILL,NSET=MATERIAL
MLINE,EDGMAT,30,1000
*NSET,NSET=CLINE,GENERATE
200025,230025,1000
*ELEMENT,TYPE=S4RF
200001,200001,200002,201002,201001
*ELGEN,ELSET=MATERIAL
200001,24,1,1,30,1000,1000
*SHELL SECTION, ELSET=MATERIAL,MATERIAL=ALU
0.001055,3
*TRANSVERSE SHEAR STIFFNESS
2.76E+07, 2.76E+07
*MATERIAL,NAME=ALU
*USER MATERIAL, CONSTANTS=2
70.E+09,0.33
*STEP,INC=1000,NLGEOM
*STATIC
1.0,1.0
*BOUNDARY
****
***********************************************
** Include here boundary conditions that describe the fully loaded
** configuration
***********************************************
*MONITOR, NODE=200001, DOF=3
*END STEP
```

```
*STEP,NLGEOM,INC=1000
*STATIC
1.0,1.0
*BOUNDARY,OP=NEW,FIXED
MATERIAL,1,6
*RESTART,WRITE,FREQUENCY=20
*END STEP
*STEP,NLGEOM,INC=1000
*STATIC
0.1,1.
*RESTART,WRITE,FREQUENCY=5
*BOUNDARY,OP=NEW,FIXED
200025,3,3
MLINE,4,4
MLINE,6,6
MLINE,2,2
CLINE,1,1
CLINE,5,5
CLINE,6,6
*END STEP
```

12 Appendix V: Code for the interpolation and offset of the surface obtained from the Finite Element Analysis

```
      PROGRAM OFFSET
C Interpolates and
C offsets a known 3D shape in cartesian coordinates
      IMPLICIT REAL*8(A-H,O-Z)
      IMPLICIT INTEGER(I-N)
      DIMENSION X(1000),C(4,1000),XDIV(80),DER(80),YDIV(80),
     &XOFF(80),YOFF(80),XI(500,500),CI(500,500),YI(500,500),
     &Y(1000),YP(1000)
      PI=3.14159
      NDIV=80
      OPEN(UNIT=1,FILE='indata',access='sequential')
      OPEN(UNIT=2,FILE='outdata',access='sequential')
      WRITE(6,2)
    2 FORMAT('Enter the number of columns:',$)
      READ(5,*)N
      WRITE(6,3)
    3 FORMAT('Enter the number of rows:',$)
      READ(5,*) N1
      WRITE(6,9)
    9 FORMAT('Enter offset value :',$)
      READ(5,*)OFFSET
      DO 10 I=1,N1
        DO 8 J=1,N
             READ(1,*,ERR=9999,END=999)XI(I,J),YI(I,J),CI(I,J)
    8   CONTINUE
   10 CONTINUE
  999 DO LMAM=1,N1
       DO I=1,N
        X(I)=XI(LMAM,I)
        Y(I)=YI(LMAM,I)
        C(1,I)=CI(LMAM,I)
        C(2,I)=0.
        C(3,I)=0.
        C(4,I)=0.
       END DO
      X(N+1)=1.0E+10
      C(2,1)=(C(1,2)-C(1,1))/(X(2)-X(1))
      C(2,N)=(C(1,N)-C(1,N-1))/(X(N)-X(N-1))
      DERI=C(2,1)
      DERF=C(2,N)
      CALL CUBSPL(X,C,N)
      DIFF=X(N)-X(1)
      DINTER=DIFF/DFLOAT(NDIV-1)
      XDIV(1)=X(1)
      DO 20 K=1,NDIV-1
      XDIV(K+1)=XDIV(K)+DINTER
   20 CONTINUE
      J=1
      DO  K=1,NDIV
        IND=K
         IF(X(N).GT.X(1)) THEN
```

```
        IF(J.EQ.(N-1).AND.X(N).GT.X(1)) GOTO 222
  26    DO WHILE(XDIV(IND).GT.(X(J+1)+1.0E-5))
        J=J+1
   4    FORMAT('J=',I3)
         END DO
 222    END IF
       IF(X(N).LT.X(1)) THEN
         IF(J.EQ.(N-1).AND.X(N).LT.X(1)) GOTO 1222
1026    DO WHILE(XDIV(IND).LT.(X(J+1)-1.0E-5))
        J=J+1
          END DO
1222   END IF
        H=XDIV(IND)-X(J)
        YDIV(IND)=C(1,J)+H*(C(2,J)+H*(C(3,J)+H*C(4,J)/3.)/2.)
        DER(IND)=C(2,J)+H*(C(3,J)+H*C(4,J)/2.)
        ARG=DATAN(DER(IND))
        ARGN=ARG+PI/2.
        XOFF(IND)=XDIV(IND)+OFFSET*DCOS(ARGN)
        YP(IND)=Y(J)+H*(Y(J+1)-Y(J))/(X(J+1)-X(J))
        YOFF(IND)=YDIV(IND)+OFFSET*DSIN(ARGN)
        WRITE(2,130) XOFF(IND), YP(IND), YOFF(IND)
        END DO
        END DO
 130 FORMAT(3(D17.9,3X))
     CLOSE(UNIT=1)
     CLOSE(UNIT=2)
9999 WRITE(6,*)'ERROR IN DATA FILE'
1000 STOP
     END

     SUBROUTINE CUBSPL(TAU,C,N)
     IMPLICIT REAL*8(A-H,O-Z)
     INTEGER N,I,J,L,M
     DIMENSION C(4,N), TAU(N)
     L=N-1
     DO 10 M=2,N
       C(3,M)=TAU(M)-TAU(M-1)
  10   C(4,M)=(C(1,M)-C(1,M-1))/C(3,M)
     C(4,1)=1.0
     C(3,1)=0.0
     IF(N.EQ.2) GOTO 30
     DO 20 M=2,L
         G=-C(3,M+1)/C(4,M-1)
         C(2,M)=G*C(2,M-1)+3.*(C(3,M)*C(4,M+1)+C(3,M+1)*C(4,M))
  20     C(4,M)=G*C(3,M-1)+2.*(C(3,M)+C(3,M+1))
  30 DO 40 J=L,1,-1
  40 C(2,J)=(C(2,J)-C(3,J)*C(2,J+1))/C(4,J)
     DO 50 I=2,N
       DTAU=C(3,I)
       DIVDF1=(C(1,I)-C(1,I-1))/DTAU
       DIVDF3=C(2,I-1)+C(2,I)-2.*DIVDF1
       C(3,I-1)=2.*(DIVDF1-C(2,I-1)-DIVDF3)/DTAU
  50   C(4,I-1)=(DIVDF3/DTAU)*(6./DTAU)
     RETURN
     END
```