METAL FLOW SIMULATION AND DESIGN OF DIES FOR CLOSED DIE FORGING

BY
FAEK DIKO B Eng, M Eng

This thesis is submitted as the fulfilment of the requirement for the award of Doctor of Philosophy by research to

DUBLIN CITY UNIVERSITY

Sponsoring Establishment

Scientific studies and research centre

DAMASCUS - SYRIA

Sept 1992

DECLARATION

I hereby declare that all the work reported in this thesis was carried out by me at Dublin City University during the period from January 1990 to August 1992

To the best of my knowledge, the results presented in this thesis originated from the presented study, except where references have been made No part of this thesis has been submitted for a degree at any other institution

Signature of Candidate

FAEK DIKO

ABSTRACT

METAL FLOW SIMULATION AND DESIGN OF DIES FOR CLOSE DIE FORGING

INVESTIGATOR FAEK DIKO

The application of computer aided design and computer aided manufacturing (CAD/CAM) technique to forming is gaining popularity as the resulting productivity improvements are becoming more and more apparent Most users are using CAD/CAM and finite element packages as stand alone packages, where the integration among these packages in most cases is difficult due to the differences in the layout format of each one
Finite element packages usually have their own pre- and post processors, however it is unlikely to include the facilities available in a CAD system such as zooming, pan, layer

This thesis describes a PC-based interactive CAD system for closed die forging design This system includes the facilities for drawing the die geometry, simulation of the deformation process and die analysis under forming conditions

First of all, a commercial CAD system has been customized to accommodate the empirical guidelines for closed die forging design Then a Finite Element program FE has been developed based on the rigid plastic/viscoplastic formulation to simulate the metal flow A mesh generation program has been developed as part of this system The CAD system has been used as pre- and post processor for the mesh generation and the FE programs

To overcome the problems encountered in forming processes, such as large deformation and displacements which cause certain computational problems, a rezoning algorithm has been developed

An elastic/plastic FE program has been used for die analysis, the FE simulation results of the forming process are used to find out whether the analyzed die would sustain the forging load or not
This metal flow simulation and die design process has been applied to two closed die forging examples, one in plane-strain condition and the other in axisymmetric condition The results were encouraging and in close agreement with the experiments

ACKNOWLEDGEMENT

The author wishes to express his appreciation to prof M S J Hashmi, Head of the school of mechanical and manufacturing engineering for his supervision, guidance and his constructive suggestions and comments during the course of this project

Sincere thanks are extended to the technicians in the workshop for their help in machining the dies

The author would like also to express his gratitude to the Scientific Studies and Research Centre, Damascus, Syria for their financial support

CONTENT

CHAPTER ONE

INTRODUCTION

1 1 METAL FORMING

Metal forming includes two types of forming processes,

- Bulk forming processes such as forging, extrusion, rolling and drawing
- Sheet metal forming processes such as deep drawing and stretch forming

A common way of classifying metal forming processes is to consider cold (room temperature) and hot (above a recrystallization temperature) forming Usually, the yield stress of a metal increases with increasing strain or deformation during cold forming and with increasing strain-rate during hot forming However, the general principles governing the forming of metals at various temperatures are basically the same, therefore, classification of forming processes based on initial material temperature does not contribute a great deal to the understanding and improvement of these processes In fact, tool design, machinery, automation, part handling and lubrication concepts can be best considered by means of a classification based not on the working temperature but rather on specific input and output geometries, material and production rate conditions

The term forging may be used to describe all mechanical hot and cold working of metals by the application of an intermittent force on the workpiece The workpiece is deformed between two die halves which carry the impressions of the desired shape Modern forgings occupy a prominent place in primary metalworking, the emphasis being to produce parts by forming rather than machining to save material and energy Thus, forgings are becoming more and more complex and diverse

In the past the forging die design procedure was based on the expenence and intuition of the die designer and some empincal guidelines [1] The need for a wider vanety of forgings and faster design procedures coupled with increasing costs led to Computer-Aided Design (CAD) techniques as a feasible altemative in forging die design The advent of high speed computers and their diminishing costs has made possible the development of CAD of forging dies to a point where the forging process can be simulated and stresses and loads predicted The dies can then be designed and manufactured for moderately complex shapes

The advent of interactive computer graphics has helped to increase the productivity of the die designer, allowing him to observe the results and use his expenence and intuition to modify them with ease, if necessary

There are now two different approaches for forging die design using Computer-Aided Methods

1 Computenzation of empincal procedures that are based on the expenence of a die designer or have been developed through expenmentation using model matenals
2 Development of numencal methods such as finite elements that simulate the forging process and therefore can be used in design process

Forging can be classified broadly into two categones Open die and Close die forgings Open die forging is carned out between flat dies or dies of a simple shape This process is used for large parts or small batch sizes In closed die forgings, the workpiece is deformed between two die halves which carry the impressions of the desired shape Deformation occurs under high pressures in the closed cavity leading to precision forgings with close toleiances This process is widely used for the manufacture of simple as well as complex high stiength precision parts

1 1 1 CLASSIFICATION OF CLOSED-DIE FORGING

Closed-die forgings are generally classified as,

1 Blocker type
2 Conventional type
3 Close-tolerance type

Blocker type forgings are produced in relatively inexpensive dies but their weight and dimensions are greater than those of conventional closed-die forging A blocker type forging approximates the general shape of the final part, with relatively generous finish allowance and radii Such forgings are sometimes specified when only a small number of forgings are required and the cost of machining parts to final shape is not excessive Conventional closed-die forgings are the most common type, and are produced with commercial tolerances and this type usually has a flash and gutter for excess material Close-tolerance forgings usually are held to smaller dimensional tolerances than conventional forgings Little or no machining is required after forging

1 1 2 FORCES AND ENERGY REQUIREMENT

In every metal forming process a definite force is transmitted at a given time by the tool into the workpiece This requires a particular amount of energy, depending upon the deformation work performed The force requirement as a function of the travel is different for the various deformation processes, and hence the force-travel variation is also a characteristic parameter It is therefore obvious that a metal forming process can be carried out in a metal-forming machine tool only when the machine can deliver at a given time the necessary force, which is at least equal to or greater than the deformation force, and when the energy available from the machine for the deformation period is sufficient to cover the deformation work In simple terms, the characteristic values of the metal forming process should be available from the machine during the deformation process

In the selection of the metal-forming machine tool, the force and energy available from the machine tool should be only slightly larger than the process requirements of force and energy from the point view of economy An optimum solution is an exact matching of the machine characteristics with the process requirements Such an optimum selection will be possible only in exceptional cases, since there are errors involved in determining the deformation force and work, and their variations during a production run require a certain reserve in the machine capacity

1 1 3 PREDICTION OF FORGING STRESSES AND LOADS

Prediction of forging load and pressure in closed die forging operation is difficult Most forging operations are of a nonsteady-state type in terms of metal flow, stresses and temperatures These variables vary continuously during the process In addition, forgings comprise an enormously large number of geometrical shapes and materials which require different techniques of engineering analysis Because of these difficulties encountered in practice, forging loads are usually estimated on the basis of empirical procedures using empirically developed formulae For example, Neuberger et al [2] have found that the variable which most influences the forging pressure is the average height of the forging

Because most of these empirical methods are not sufficiently general to predict forging loads for a variety of parts and material, other analytical techniques have been used Among these techniques, the relatively simple slab method has been proven to be very practical for predicting forging loads

1 1 4 FRICTION AND LUBRICATION IN FORGING

In forging, friction greatly influences metal flow, pressure distribution, and load and energy requirements In addition, to lubrication effects, the effects of die chilling or heat transfer from a given lubricant, friction data obtained in hydraulic-press forging cannot be used in mechanical-press or hammer forging even if the die and billet temperatures are comparable

In forging, the lubricant is expected to, 1 reduce sliding friction between the dies and the forging in order to reduce pressure requirements, to fill the die cavity and to control metal flow

2 act as a parting agent and prevent local welding and subsequent damage to the die and workpiece surface

3 possess insulating properties to reduce heat losses from the workpiece and minimize temperature fluctuations on the die surface

4 wet the surface uniformly so that local lubricant breakdown and uneven metal flow are prevented

5 be nonabrasive and noncorrosive so as to prevent erosion of the die surface

6 be free of residues that would accumulate in deep impressions

7 develop a balanced gas pressure to assist quick release of the forging from the die cavity This characteristic is particularly important in hammer forging, where ejectors are not used

8 be free of polluting or poisonous components and not produce smoke

No single lubricant can fulfil all these requirements listed above, and therefore, a compromise must be made for each specific application

1.1 5 SELECTION OF DIE MATERIAL

Closed-die forging dies are usually made from low-alloy, pre-hardened steels containing 0 35-0 50 % carbon, 1 50-5 00 % chromium, and additions of nickel, molybdenum, tungsten, and vanadium It is difficult to heat treat die blocks safely after machining because thermal distortion could destroy or reduce the dimensional accuracy of the cavity Therefore, die blocks are machined after the desired hardness has been achieved through heat treatment Die blocks containing shallow or simple cavities can be hardened to R_c 50 However, die blocks with deep cavities, ribs, or complex design require relatively softer, tougher materials to minimize cracking and die breakage

When the volume of parts is high and the size of the forging is limited, die inserts can be incorporated in the die block to minimize wear Inserts are generally installed in locations that are prone to excessive wear due to complexity of design and material flow Table 2 1 lists recommended die block materials for forging various materials [3]

Material Forged	Application	Die Material	Hardness,R_c
Aluminum	Punches,die Die inserts	H11,H12,H13 H11,H12,H13	44-48 46-50
Brass	Punches,dies and inserts	H21,H11,H13	48-52
Steel	Punches,dies, and inserts	H13,H12,H19	38-48
	Trimmer dies	D2,A2 or hardweld on cutting edge of cold-rolled steel	58-60

Table 1 1 Recommended Die Materials for Closed Die forging Dies

1 1 6 MATERIAL FOR FORGING

The most important consideration when selecting a material for forging to be forged is its forgeability Other considerations would be based on the mechanical properties that are inherent in the material or that can be obtained as a result of forging and heat treatment

These properties include elastic modulus, density and strength, resistance to wear, fatigue, shock, or bending, response to heat treatment, machining characteristics, and durability or economy

Forgeability can be expressed as a combination of resistance to deformation and the ability to deform without fracture and can be defined as the capability of the material to deform without failure regardless of the pressure and load applied

Forgeability for a particular material is based on,

1 Metallurgical factors such as crystal structure, composition, purity, number of phases

present and grain size

2 Mechanical properties, the two most significant factors affecting forgeability are strain-rate and stress distribution Rapid deformation of metal can increase the material's temperature significantly during the forging operation and can actually decrease the material forgeability if heated sufficiently for some melting to develop

1 1 7 CAUSES OF DIE FAILURE

Mainly, there are three basic causes of die failure,

1 Overloading

Overloading may cause rapid wear and breakage It can be avoided by careful selection of die steel and hardness, use of blocks of adequate size, proper application of working pressures, proper die design to ensure correct metal flow, and proper installation of the die in the press machine

2 Abrasive action

Abrasive caused by the flow and spreading of hot metal in the cavity of a forging die Abrasion is particularly severe if the design of the forging is complex or in other respects difficult to forge, if the metal being forged has a high strength Abrasion can be eliminated or minimized by good die design, good lubricant, careful selection of die composition and hardness, and proper heating

3 Overheating

As a die becomes hotter, its resistance to wear decreases Overheating is likely to occur in areas of the die cavity In addition, overheating may result from continuous production

1 2 LITERATURE SURVEY

1 2 1 CAD/CAM APPLICATIONS

The design of the forged component and its dies starts with an enquiry from a customer, who provides a machining drawing The designer examines it with reference to the capacity of the available equipment, primarily the maximum load, the energy and the die space After establishing that these are available in the workshop, the design study is initiated in greater detail

The forging process design essentially comprises five steps

1 The conversion of the machined part geometry to the forged part geometry to accommodate design considerations and process limitations
2 Determination of the number of preform stages
3 Design of preform/block dies
4 Design of finisher dies
5 Evaluation of process parameters,namely, forging loads and stresses, energy requirements and stock size

From the machining drawing, the surfaces which require machining allowance are easily identified and allowances are chosen on the basis of past experience or organized standards [4,5] Some designers have changed standard data into polynomial expressions for easy implementation into CAD system [6,7} Similarly, the sharp vertical surfaces are made inclined by adopting suitable draft angles in order to facilitate component removal from the forging dies and to ease metal flow within the die cavities From the customer's point of view this entails some additional machining but in fact this is more than offset by the consistency of the forging and the increased die life Depending on the geometry of the component, the die-parting line separating the top and bottom impressions is decided upon

Empirical guidelines for preform design of H-sections have been compiled by Akgermann et al [8] from a number of sources The effectiveness of the preform design on the basis of these guidelines was tested by Akgermann through the use of transparent dies and modelling materials like plasticine A more general approach to preform design has been developed by Chamouard [9] based on the natural metal flow theory According to this theory ,metal when allowed to flow freely, tends to flow along a logarithmic curve in the direction of forging Chamouard [9], therefore developed guidelines for the use of such curves in preforms for joining the web and rib portions of the forgings Chamouard's work has been used in slightly modified form by many researchers such as [10]

The finishing die design involves the design of the flash and gutter geometries and determination of the centre of loading Since axisymmetric forgings make up the largest percentage of forgings produced [11], extensive work has been done in the finisher die design for such forgings Teterine et al [12] has developed comprehensive quantitative guidelines for flash design of axisymmetric forgings However, Neuberger and Mockel [2] have suggested formulae relating the weight of the forging to the flash geometry These relations have been analyzed by the Drop Forging Research Association (DFRA) [13,14] and found to be reliable

In the design of forging dies an important consideration is the location of the centre of loading Off-centre loading, which occurs if the centre of the ram and the centre of loading do not coincide, causes imperfections in the forging and also leads to shear failure of the dowel pins on the dies Mollineaux and Knight [15] reviewed the various methods for determination of the centre of loading The various factors of affecting die life have been described in reference [16]

It is necessary to estimate the loads and stresses developed during the forging process, as the peak load and energy requirements determine the feasibility of the process as well as die life The energy requirements determine the necessity of preforming as well [10] Apart from the Finite Element method, which can give the stress distribution as well as peak load and stresses, several other methods exist for the determination of the loads and stresses Altan et al [17,18] discussed the principles and limitations of the various analytical, numerical, and experimental methods used to analyze the forging operation One of these methods is the Slab Method Lui and Das [19] have used the slab method for evaluating the loads and stresses in axisymmetric forgings Biswas and Rooks [20] used a modular approach to evaluate the loads and stresses in which the various deformation stages are uncoupled and analyzed separately They also have developed a computer simulation technique to estimate load and energy in axisymmetric closed die forging [21] In this simulation a step-by-step simulation technique has been used and good accuracy has been demonstrated

Van Hoenacker and Dean [22] described means for utilising Upper Bound type of analyses for process involving materials which are not perfectly plastic Predicting the geometry of forgings is shown to be possible, but the choice of velocity field is shown to have a significant effect on the accuracy

Hashmi and Klemz [23] have compared the experimental results with those predicted theoretically using a numerical technique In this numerical technique the strain hardening and strain rate sensitive material property was incorporated

Chan et al [24], have developed a system of programs for the design and manufacture of hot forging dies Each of these programme could be regarded as a module in such an integrated system, but which can be used effectively in isolation also
Choi and Dean [25] developed an interactive computer program for die layout design which is part of a complete CAD/CAM system for forging hammer dies This program deskills the design of die layouts and enable die block manufacture to be speeded up They have also developed an interactive computer program, implemented on a 64k mini-computer to aid the process of preparing data for cost estimation and preform die design for forging on hammers [26]

There are also some empirical formulae which can predict peak loads and stresses These have been reviewed by Altan and Fiorentino [27] Empirical relations for the estimations of loads and energy have also been developed by the DFRA [13] for hammer forgings of various grades of steel

Toren et al [28] have done some work investigating approximate calculation of thermal and mechanical loads on forging dies Guidelines are given in this study for die design and choice of die material in order to avoid critical failure

The guidelines mentioned above have been converted into computer programs for the design of forging dies, Lui and Das [19], and Altan and Henning [29] for the design of axisymmetric forgings, all based on the work of Teterin et al [12] Biswas and Knight [30,31] and Mullineux and Knight [32] have also developed computer programs for preform design based on the work of Chamouard [9] Similar work has been done by Subramaniam and Altan [33], and Ackergmann and Altan [34]

Choi et al [35] have developed an interactive CAD/CAM package to aid the processes of cost estimation, preform die and layout design and manufacturing of die blocks for forging hammers

1 2 2 Finite Element Analysis

Due to the rapid development of computers and numerical methods, the Finite Element Method (FEM) has become popular for the solution of metalworking problems [36] The appeal of the FEM stems from its ability to systematically represent material behaviour and complex boundary conditions of metal forming processes The method has proved very successful and the literature is expanding rapidly

Kobayashi [37] presented a comprehensive review for the analysis of metal forming processes in 1979 Shabaik [38,39] points out the distinctions between the various constitutive formulations used to simulate the deformation of metals

The FEM, though developed in the early 1950's, really progressed in its application to metal forming only in the 1960's One of the first approaches to the problem was the Elastic-Plastic Finite Element Method, developed by Marcal and King [40] Later, Yamada et al [41] and, Lee and Kobayashi [42,43] and Lee and Mallett [44] used the method to solve a variety of problems in elasto-plasticity such as flat punch indentation, upsetting of solid cylinders and extrusion Relatively successful small strain analysis of the above processes was made possible by this method However, it was not economical for the solution of large deformation problems encountered in actual metal forming processes

Besides the Elastic-Plastic FEM, two other basic approaches to solution of forging problem have been developed

Eulerian-Based Analysis

This method makes use of rigid-plastic or rigid-viscoplastic laws With this method the metal flow is equivalent to that of a viscous, incompressible, non-Newtonian fluid It is assumed that elastic strains can be ignored compared to the large plastic strains This simplifies the problem and offers definite computational advantages over the Elastic-Plastic/Viscoplastic approaches

As developed by Lee and Kobayashi [45] and Kobayashi and Shah [46], the Rigid-Plastic FEM is characterized by the variational principles for a material obeying von Mises's yield criterion, with isotropic kinematic hardening [47] Several investigators[48-51] have since contributed to the development of the Rigid-Plastic FEM for the analysis of metal forming problems

In the mid-1970's, Zienkiewicz et al [52,53] generalized the Rigid-Plastic formulation to a third approach, namely the Rigid-Viscoplastic method of analysis, capable of dealing with hot, rate-dependent processes This analysis can be applied to the Rigid-Plastic case when rate-insensitive situations are encountered

In the early 80's, Oh et al [54] refined the Rigid- Viscoplastic formulation to solve a wide variety of problems and the effort culminated in the development of a two-dimensional finite element program for metal forming called 'ALPID' [55] Mitani and Mendoza [56] analyzed open die forging of 134 ton steel ingots for low-pressure rotor shaft using a rigid-plastic FE code RIPLS-FORGE, to examine a practical design of upset forging Maccaini et al [57] investigated the influence of die geometry on cold extrusion forging operations By using the FEM code developed by the authors they could describe the actual processes taking into account the plastic behaviour of the material, the various lubrication conditions and the complex geometry of the die

Lagrangian-Based Analysis

Hibbitt et al [58] introduced the first complete finite element large strain formulation which included elastic strains This was the Total Lagrangian Formulation or TLF, in which the reference state is the original undeformed configuration

Only a few investigators [59] based their analyses on this formulation The Updated Lagrangian Jaumann formulation, or ULJF, which uses the current deformed configuration of the material as the reference state, was a more appealing to investigators because of its ability to model large deformation metal forming problems in a more natural way Elaborate discussion of ULJF can be found in a paper by McMeeking and Rice [60] Several investigators [61-63] have applied the method to problems of extrusion, drawing, rolling, and sheet metal working

1 2 3 FRICTION AND BOUNDARY CONDITIONS

Friction and lubrication are of great importance in forging operations In most cases reducing friction is beneficial since it ieduces the foice and energy required for a given operation This will reduce the stresses imposed on dies and may allow the use of smaller hammers or presses for a given part Alternatively, large changes of shape can be achieved with a given level of force or energy In some operations a controlled amount of friction is necessary to control material flow in order to promote die filling or reduce workpiece spreading In such cases too little friction is as bad as too much, and the lubrication system must be carefully specified and controlled to achieve optimum friction level In the finite element simulation of the metal flow, the friction conditions have been incorporated within the program in different ways Hartly et al [64], solved this problem by using an additional layer of elements which is incorporated on all contacting surfaces to model the influence of interface friction Chen and Kobayashi [65] implemented the finite element scheme for the analysis of ring compression, by introducing velocity dependent frictional stresses The frictional stress, in general, changes its direction at the neutral point, but the location of this point is not known a priori The neutial point problem has been considered by various investigators [64-66] The die boundary condition along curved die-workpiece interfaces have been considered in the framework of FEM by several investigators [67-69]

The values of the friction used in most of this program have been determined experimentally or using approximate methods Eltouney and Stelson [70] presented an approach to calculate the friction coefficient during nonuniform compression of cylinders However, the ring test proved to be very useful in predicting the friction factor under various temperature, lubrication and strain-rate conditions [71-73]

Contact problems arise in metal forming where the determination of contact points and the frictional forces between a deformable body and the rigid die is important Contact problems have long been of considerable interest, and a large literature base is available for a variety of simple to complex boundary problems The solution method can be broadly classified into three categories The earliest solutions to contact problems have been obtained using integral equation methods Various problems were solved in close form by Muskhelishvili [74] and Gladwell [75], and with numerical techniques by others [76,77] In the second method problems are considered as a special case of constrained minimization of either total or complementary potential energy The minimization is formulated as a mathematical programming problem and the solutions are obtained by using either incremental linear programming [78,79] or quadratic programming [80] techniques Extensive research with these techniques has been done in the analysis of classical and non-classical friction at the contact interface [81-83] In the third category, contact conditions are imposed directly from kinematic considerations by imposing geometric capability of the contacting surfaces during the incremental loading process [84-91] The main advantage of this method is that the various frictional conditions at the interface can be easily imposed and the algorithms are generally independent of material constitution [92-94]

1 2 4 MESH GENERATION AND REZONING

The increased use of finite element numerical methods due to the availability of high speed, large memory computers has led to the solution of many unsolved problems In any FEM program the preparation of the input data and mesh generation should be simple Yates et al [95] investigated the cost and stated the total analysis time and cost in preparing the data in conventional ways The 2D topology decomposition approach was developed by Wordenweber [96,97] The important contribution of this approach to mesh generation is the concept of operators, which was perhaps borrowed from the concept of Euler operators pioneered by Baumgart [98] Another approach is the node connection approach [99-101] In conventional mesh generation procedures, FEM users are requires to decide which mesh density will achieve the best solution with minimal use of central processing unit (CPU) time The quality of the FEM mesh depends on the user's experience, and actual mesh construction is time consuming

Many schemes were proposed for automatic mesh generation (AMG) Cavendish et al [102] developed a two-stage approach to automatic triangulation of an arbitrary solid model, and it was later refined by Field and Frey [103] Wordenweber [104] and Woo and Thomasme [105] proposed a different class of schemes for decomposing a solid model into a collection of tetrahedral elements Wu et al [106] developed an AMG for 4-node quadrilateral elements implemented in the DEFORM system Special attention should be given to a full automatic scheme which was introduced by Yerry et al [107,108] In metal forming simulation the mesh can become so distorted that remeshing is absolutely necessary to prevent the degeneracy of the elements A lot of work has already been devoted to the construction of meshes with optimum geometric properties, or with some degree of adaptivity to the solution [109-113] A continuous remeshing technique has been suggested by Cescutti and Chenot [114] which allows a smooth and adaptive mesh during the whole process This method has been illustrated in 2-D examples with four-node linear elements [115] and in 3-D examples with cubic eight-node linear elements [116]

2 3 SCOPE OF THE PRESENT WORK

The objective of this work is to develop a CAD system which can be used by forging designers to design closed-die forging dies and test their processes In general the desired system should reduce the time spent on designing the dies and the trials at the workshop, increase the accuracy of the drawings and calculations, and finally reduce the errors in selecting the design data Errors should be identified and corrected easily before the incorrect data leads to costs and difficulties in manufacturing

In order to achieve such system, this research has been concentrated on three individual points which eventually contribute to the creation of the system This points are summarized as,

1 Customizing a CAD system for closed die forging design so that it will become the framework of the system This customization will include the development of several routines and functions which contain the design rules of close die forging Also a modification of the menu and the creation of a new submenu is carried out Finally, the developed CAD system is used as a post and preprocessor for the finite element program and the geometrical design of the die

2 The development of a rigid-plastic/visco-plastic finite element program for metal flow simulation This program has been developed to simulate the deformation process and give the field variables during the deformation as results Special attention has been paid to the contact problem and the remeshing during the analysis

3 An elastic-plastic finite element program has been used for die analysis

Eventually the system should have the following characteristics,

1 This system should be PC-based because it is less expensive and within the reach of all forgers

2 It should be able to communicate with other systems for drawing exchange or using other CAD/CAM packages

3 It should be able to do area and volume calculations

4 Forging rules should be built-in and implemented in a modular form and can be easily updated if better rules become available

5 It should be able to generate the die geometry using the built-in rules

6 It should be able to generate the billet that will be placed in the die and be deformed

7 The system should be able to simulate the deformation process and calculate the required forging load, using the FE method as a simulation technique

8 It should be able to generate a mesh system on the billet

9 It should be able to remesh as often as necessary

10 It should be able to postprocess the result of the simulation and display them to the user in an easily interpreted form, such as colour contour plots, colour display etc

11 The system should be able to analyze the die and find out if it sustains the forging loads

During the course of the system development, the objective was to select and develop the best algorithms and methods to achieve a compromise between the accuracy of the solution and the computational time For this reason the Rigid Plastic formulation has been used for the metal flow simulation and an explicit method for the contact problem is incorporated

The thesis has been divided into eight chapters Chapter one presents the literature survey of several topics such as the application of CAD/CAM to metal forming and the use of finite element simulation This chapter also gives a brief idea about closed die forging, its classification, design requirements and cause of failure Close die forging has been chosen as a case study for testing the developed system Chapter two discusses the customizing of AutoCAD for metal forming process design Macros and routines developed by the author have been discussed as well Chapter three explains the rigid plastic formulation used for metal forming simulation The governing equations, discritization of the domain, matrices of strain rate and volumetric strain rate , the stiffness matrix, contact formulation and remeshing are discussed in detail in this chapter Chapter four presents the implementation of the rigid plastic formulation and the coding procedures of the individual subroutines of the FEM program Chapters five and six present the examples for plane strain and axisymmetric die design respectively, then the actual experiments of forging process are presented in chapter seven Chapter eight contains the conclusions and discussion and shows the advantages of this system and the comparison between the results produced by the CAD system and the experiments The thesis is concluded by appendices which contain lists of CAD routines, the finite element simulation code and the publications

CHAPTER TWO

CUSTOMIZING A CAD SYSTEM FOR CLOSED DIE FORGING

2 1 INTRODUCTION

Usually CAD systems are general purpose softwares which can be applied on different engineering areas What makes a particular CAD system different from others is its library and other individual routines which can be used for a particular application Such extra facilities are very expensive and if they do exist there might be some limitation of the facilities required In this work an attempt has been made to make use of an existing CAD system by customizing this system to be used for metal forming applications The target was to change a machined part drawing to a forged component then extracting the die block from the forged part To do so in the conventional way of designing, empirical guidelines are used In this system appropriate guidelines and forging data are selected and built within the CAD system in the form of routines and a database These routines are fully interactive and use all the facilities available in the CAD system During the process of designing a die, two finite element programs are used one for simulating the material flow and the other is for die analysis The post and pre-processors of the first FE program are also built within the CAD system Fig 2 1 shows the CAD\CAM procedure for forging die design

2 2 SYSTEM CONFIGURATION

The function of this system, as mentioned before, is to design metal forming dies starting from the machined part geometry which can be in 2D or 3D Using the facilities which have been collected and developed within this system, the user will be able to design the die set with its cavity The steps of using this system are shown in Fig 2 2 and explained as follows,

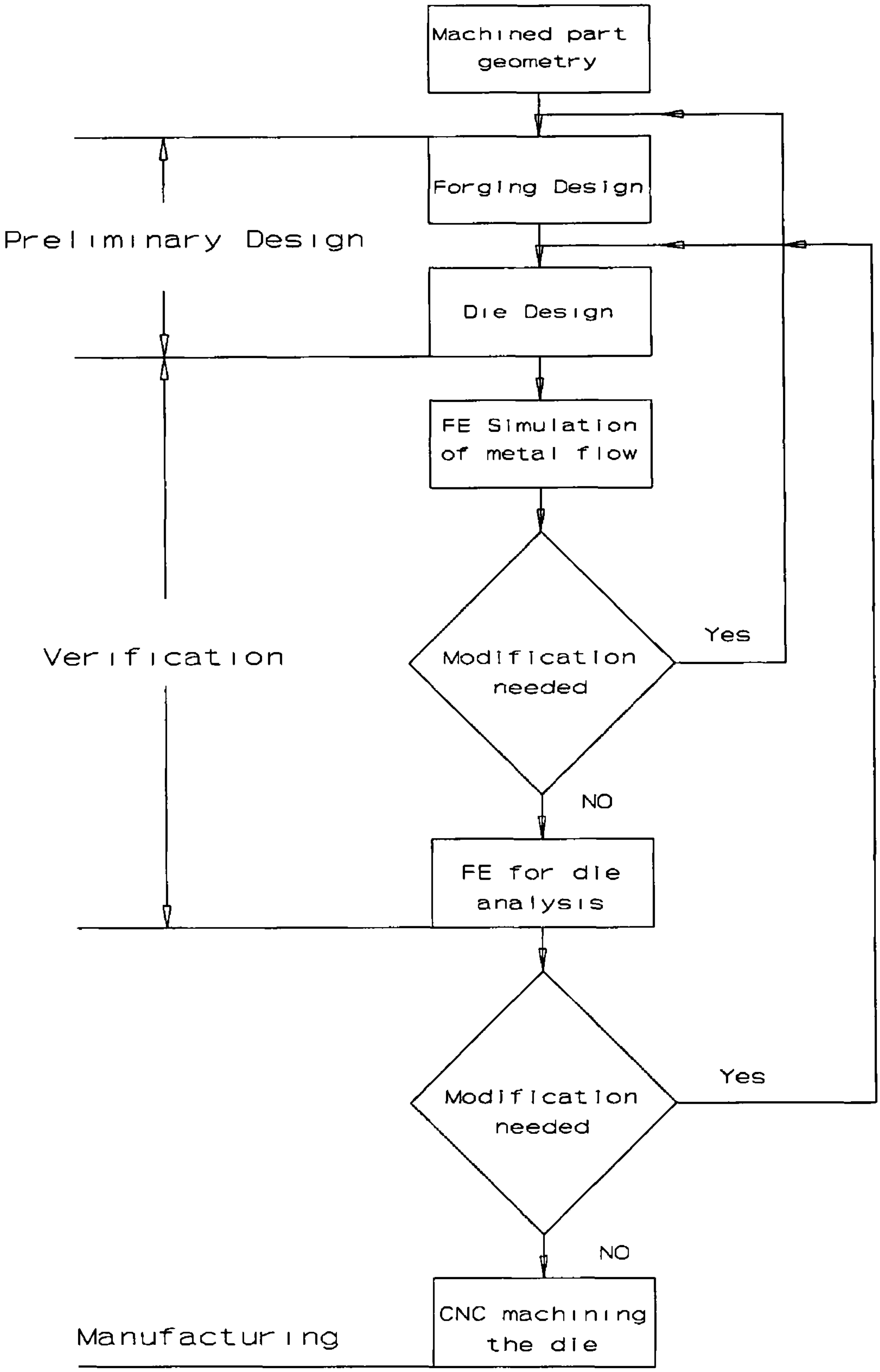
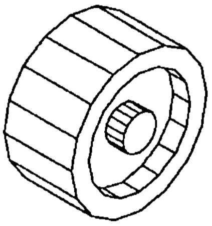
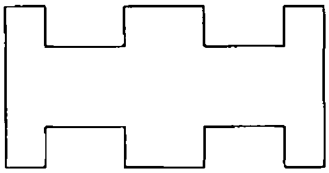


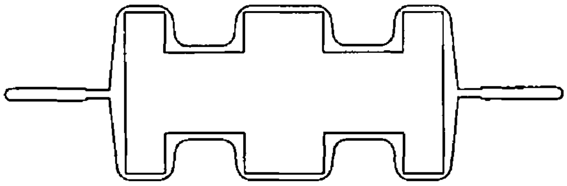
Fig. 2.1 CAD\CAM procedure for forging die design



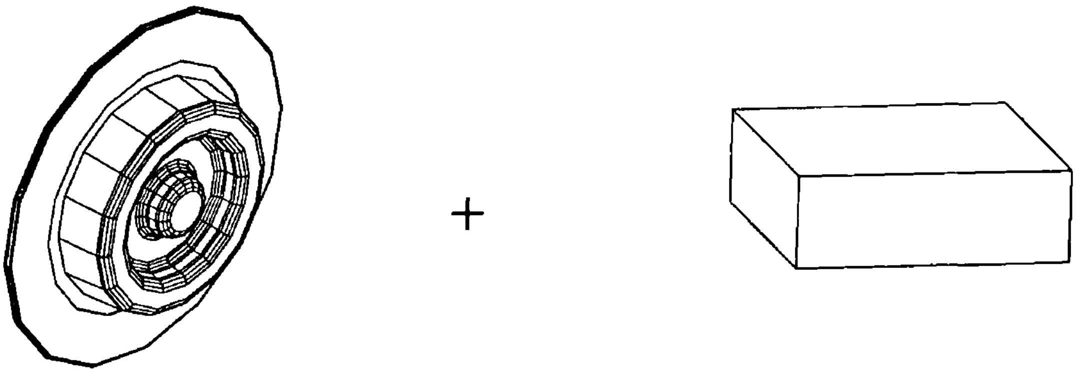
STEP1 3D MACHINED COMPONENT



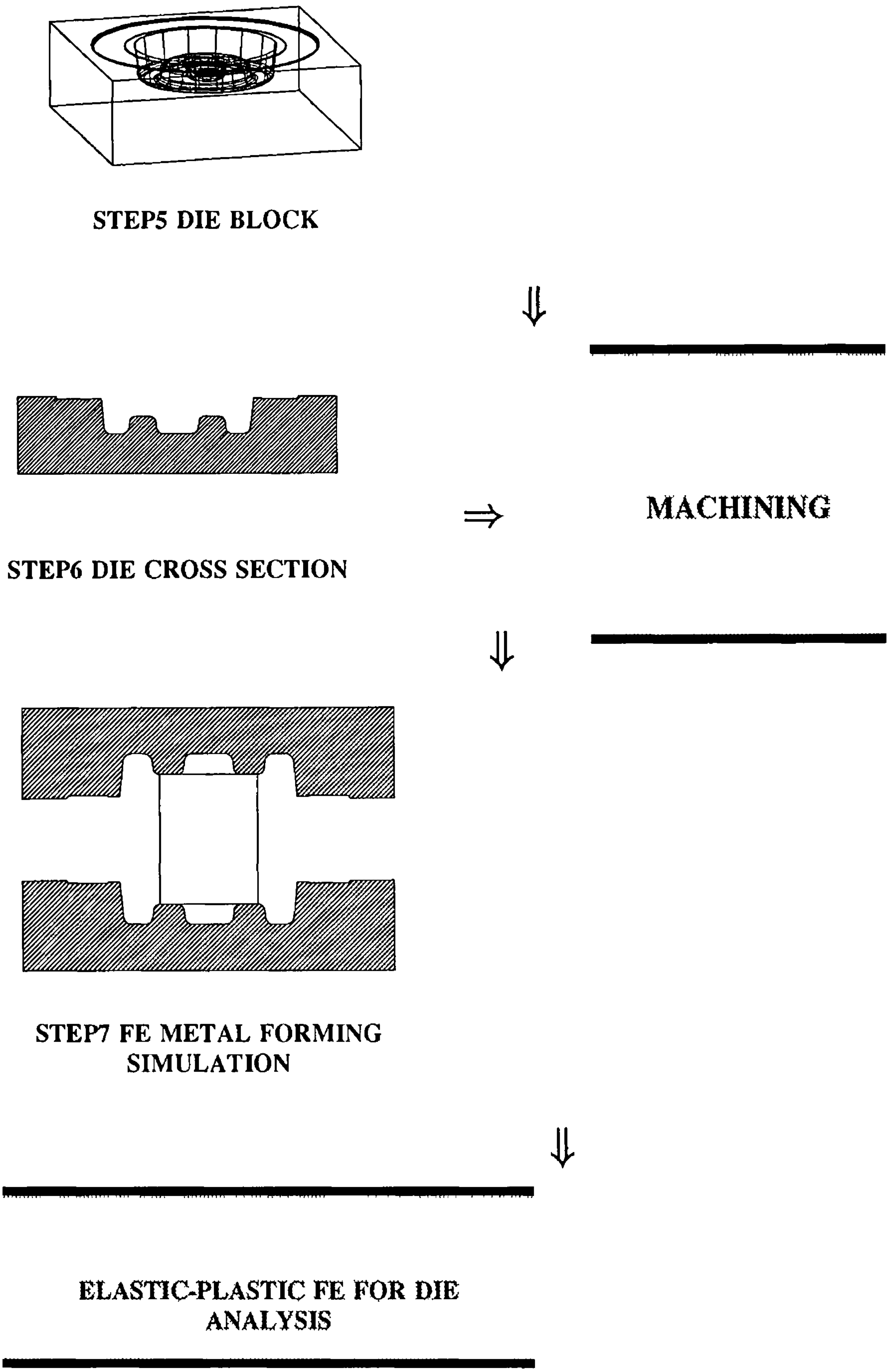
STEP2 CROSS SECTION OF THE
MACHINED COMPONENT



STEP3 FORGING CROSS SECTION



STEP4 INTERACTION BETWEEN THE FORGING AND DIE BLOCK



STEP5 DIE BLOCK

STEP6 DIE CROSS SECTION

MACHINING

STEP7 FE METAL FORMING SIMULATION

ELASTIC-PLASTIC FE FOR DIE ANALYSIS

Fig 2 2 Flow chart of the process

1 If a previous drawing of the machined part is not available, the user can draw a 2D or 3D drawing using the AutoCAD facilities, although it is possible to receive the drawing through a network from other designers

2 If the available drawing is in 3D, a critical cross section is prepared

3 Using the routines built within the CAD system, the cross section is converted to a forging cross section

4 A 3D drawing of the forging part is produced by revolving the 2D drawing around the symmetry line and forging volume with the flash is calculated for determining the dimensions of the billet

5 The die block is produced by using Boolean commands The block and the forging are subtracted along the parting line of the forging to create the die cavity

6 A cross section is produced for the die block and the finite element model is prepared for metal flow simulation

7 If the simulation process is satisfactory and the die cavity is completely filled with the material, the die block is analyzed using the elastic-plastic FE package If not, the geometrical design of the die or the forging conditions are modified

8 If the die block sustains the forging load, it will be sent for machining If not, the die block will be modified

The steps mentioned above consider an axisymmetric component For the plane strain case the same steps are applied, however, instead of revolving the 2D cross section it is extruded For more complex shapes, several cross sections are taken which can be axisymmetric or plane strain and then analyzed and put together

2 2 1 SOFTWARE CONFIGURATION

The softwares used in this work are divided into two categories,

2 2 1 1 The commercial Packages

a AutoCAD, 2D and 3D package release 11

b LUSAS , elastic-plastic finite element package

2.2.1 2 Inhouse built packages

a Finite element simulation package

b Mesh generation package with remeshing

c Routines built in the AutoCAD for die forging design

2 2 2 HARDWARE CONFIGURATION

a A 386 personal computer with Intel 387$^{\text{TM}}$ DX Math CoProceesor, 100 Mb hard disk, 8 Mb RAM and 20 MHZ speed

b VGA graphic display unit

c Digitizer (LDS)

d Printer (Star LC-10)

e Plotter (Roland DXY-1300)

2 3 CUSTOMIZING THE MENU

The menu file in AutoCAD is a simple text file containing AutoCAD command strings Section of the file can be associated with different menu device, such as the screen and tablet menus Only the screen menu has been used in this work to leave room for future work The command ***Die design*** is added to the main menu This command activates several submenus which invoke the developed routines The submenu items temporarily replace all the current menu and it is possible to return to the main menu or the last menu once the user finishes from using a particular function

2 4 ROUTINES FOR CUSTOMIZING THE CAD SYSTEM

AutoLISP is an implementation of the LISP programming language embedded within AutoCAD package By writing programs in AutoLISP, it is possible to add commands to AutoCAD and modify AutoCAD much like the original routine in the package AutoLISP has been used to develop all the routines presented in this work

Metal flow in closed die forging operations is three- dimensional and therefore, difficult to analyze Thus, the desıgn process ıs sımplıfıed by consıdenng cntıcal two-dımensıonal cross-sectıons of the machıned paıt geometry to be produced

Then the cross-sectıon ıs modıfıed by

1 selectıon of the partıng lınes,
2 the addıtıon of the machınıng allowance,
3 the addıtıon of the draft allowance,
4 the addıtıon of the fillet and corner radıı

The above procedures are translated to routınes to carry out thıs procedures ındıvıdually when needed as shown ın Fıg 2 3
The FORTRAN-77 language ıs also used for developıng some functıons

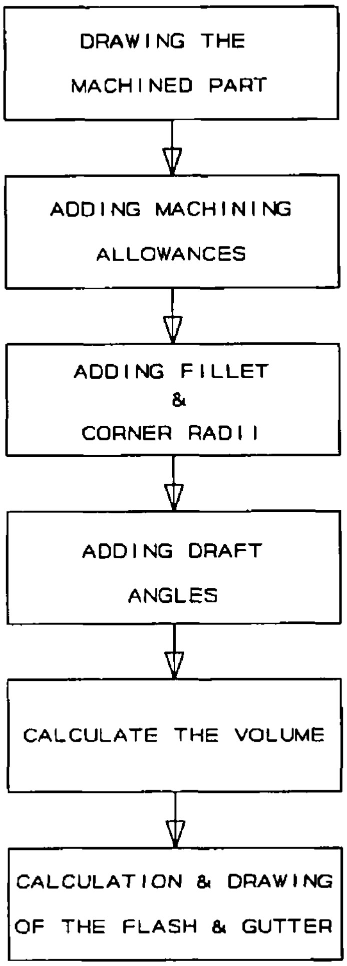

DRAWING THE MACHINED PART
ADDING MACHINING ALLOWANCES
ADDING FILLET & CORNER RADII
ADDING DRAFT ANGLES
CALCULATE THE VOLUME
CALCULATION & DRAWING OF THE FLASH & GUTTER

Fıg 2 3 Machıned part conversıon

2 4 1 MACHINING ALLOWANCE PROGRAM

Thıs program has been constructed usıng two routınes as shown ın Appendıx A The maın target ıs to make good ınteractıon between the user and the graphıc monıtor Eventually, the user can choose the desıred machınıng allowance eıther by usıng automatıc selectıon usıng the database whıch contaıns machınıng allowance values taken from DIN 7523 [4], Table 2 1, or by vısualızıng the same table and assıgnıng a chosen value Thıs table ıs saved as a slıde whıch appears on the screen when needed
The routıne to do thıs selectıon has been wrıtten usıng AutoLISP, whıch calls another functıon wrıtten ın FORTRAN The fırst routıne does the ınteractıon between the AutoCAD and the user, where the second does the selectıon process The memory for

the FORTRAN routine has been saved using the ACAD PGP file facility

Machining allowance command is added to the main menu of the die design By selecting this command a submenu appears which contains two commands for setting the value of the machining allowance and then activating the routine which is the addition process of the machining allowance value to the desired edges of the machined workpiece The user has the choice either to use the direct input from the Keyboard or picking up the commands from the menus

Maximum size (width or thickness)		Maximum length elongated forgings								
Maximum thickness		Maximum diameter of rotationally symmetric forgings								
Over	Up to	up to 40	40 63	63 100	100 160	160 250	250 400	400 630	630 1000	1000 1600
	40	1 5 (1)	1 5 (1)	2 (1 5)	2 (1 5)	2 5 (1 5)	3 (2)	4 (2 5)	5 (3)	6 3 5
40	63	1 5 (1)	2 (1 5)	2 (1 5)	2 5 (1 5)	3 (2)	3 5 (2 5)	4 5 (3)	5 5 (3 5)	6 5 (4)
63	100	2 (1 5)	2 (1 5)	2 5 (1 5)	3 (2)	3 (2)	3 5 (2 5)	4 5 (3)	5 5 (3 5)	6 6 (4)
100	160		2 5 (1 5)	3 (2)	3 (2)	3 5 (2 5	4 (3)	5 (3 5)	6 (4)	7 (4 5)
160	250			3 (2)	3 5 (2 5)	4 (3)	5 (3 5)	6 (4)	7 (4 5)	8 (5)
250	400				4 (3)	5 (3 5)	6 (4)	7 (4 5)	8 (5)	9 (6)
The bracket values shall be avoided where possible owing to the extra cost involved										

Table 2 1 Machining allowances

THE PROGRAMS EXECUTION STEPS

First of all, the value of the machining allowance should be selected by invoking the command ***set value*** either from the menus or using the Keyboard Doing that AutoCAD will prompt

Command Do you prefer automatic selection of the machining allowance (Y or N) ?

A reply by "Yes" or simply "Y" will control the subsequent series of prompts as follows

Command Input the maximum thickness
Command Input the maximum diameter

The user may enter a distance explicitly, "show" AutoCAD a distance by two points, or enter these two values through the Keyboard Then the program will retrieve the suitable value of the machining allowance from the DIN 7523 tables in the database Now the chosen value of machining allowance is set in the memory although it can be changed to any other value if the user wants to

If the reply is "No", the prompt will ask for a value to be entered through the Keyboard At the same time a slide of the DIN 7523 which contains the machining allowance will be displayed on the screen and it will disappear as soon as the input procedure is completed

Command Input the value of the machining allowance

The next and last stage is to modify the geometry according to the value which has just been set up By invoking the command *Offset* from the menu, the AutoCAD will prompt

Command· Select three sides of the geometry wheie the one to be modified is in the middle

The selection process will be done as shown in Fig 2 4, where the target side is (bc) in the example Once the selection process has been done, a special routine will define these lines and replace them by a new set of lines (ab´-b´c´-c´d)
Then AutoCAD will prompt for continuing by

Command. Do you want to modify any other side (Yes or No)?

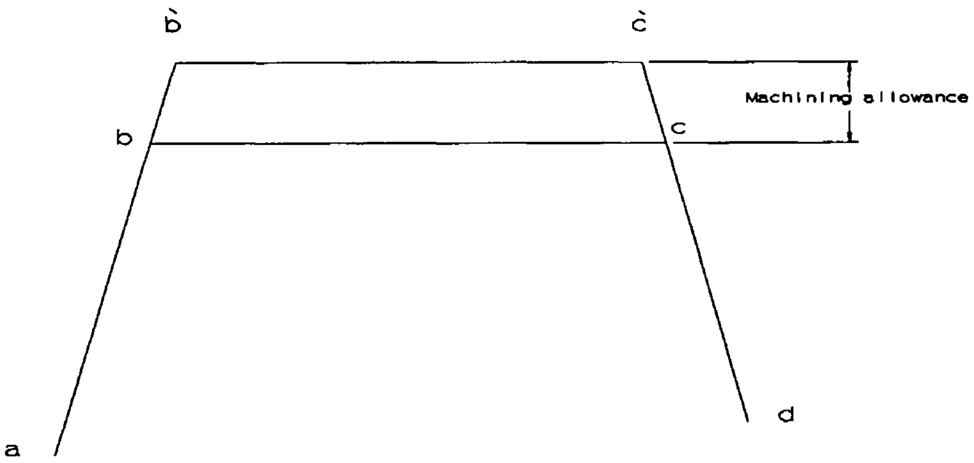


Fig 2 4 Machining allowances

The user can go on modifying the sides he wants considering the possibility of changing the value of the machining allowance whenever he wants

2.4 2 THE DRAFT ANGLE PROGRAM

To enable drop and press forgings to be lifted out of the die cavity it is necessary for their surfaces disposed in the forming direction to be tapered The rate of taper needed differs on the internal and external forged surfaces and depends on the forming process and on the size and shape of the forging If the intended forming machine allows the use of dies incorporating ejectors, the drafts on the forging can be made smaller

Drop forging dies and the upper die halves of forging process are generally made without ejectors The draft applied to the upper die halves can often be reduced if the bottom die halves are equipped with ejectors and feature very small drafts

Small and light weight drop and press forgings, as a rule, necessitate larger amounts of die draft than heavy forgings in order to allow the forgings to be inserted correctly into the trimming die

In order to apply the draft angle on the geometry which has been created using AutoCAD, two programs have been developed to achieve this task as shown in Appendix D These programs are written using AutoLISP and FORTRAN languages and their task is to set up the value of the draft angle and save its value in the memory

Then thıs value ıs applıed on the desıred sıde The setup has also two maın optıons as has been descrıbed ın the prevıous program, automatıc selectıon of the draft angles from DIN 7523 [4] and the DFRA forgıng handbook [117] as shown ın Table 2 2 and Table 2 3 Manually the value ıs ınput through the keyboard to gıve a chance to the user to use hıs own experıence The second program applıes the value of the draft angle on the geometry ın an ınteractıve mode

Internal drafts			External drafts 1)		
Drop or press forgıngs		Upset forgıngs	Drop or press forgıngs		Upset forgıngs
Dıe half			Dıe half		
wıthout ejector	wıth ejector		wıthout ejector	wıth ejector	
6 **1 10** 9° (3°) 1 6 (1 20)	**3** **1 20** 6° (1°30) 1 10(1 40)	**3** **1 20** 6° (0°30) 1 10(1 115)	**4 30** **1 12 5** 6° (2°) 1 10(1 30)	**2** **1 30** 3° (0°30) 1 20(1 115)	**2** **1 30** 3° (0°30) 1 20(1 115)
In practıce the values prınted ın bold type are usually adopted The bracketed values should not be used because of the extra cost ınvolved 1) In the case of flat parts larger angles for the draft on eıther sıde of the flash (or burr) may be requıred to allow for trımmıng operatıons					

Table 2 2 Drafts

	Hammer dıes		Press dıes	
Materıal	External	Internal	External	Internal
Steel Alumınıum alloys Tıtanıum alloys Nı base alloys	5° 7°	7° 10°	3° 5°	5° 7°
Tolerances ın all cases	+1° 1° or +2	0°		

Table 2 3 Drafts (Forgıng Handbook)

THE PROGRAM EXECUTION STEPS

The commands to access the two programs have been added to the AutoCAD menus The procedure of applyıng the draft angles starts by ınvokıng the command ***Set up*** whıch cause a sequence of prompts as

Command Do you want to ınput youı own dıaft angle (Yes or No)?

Thıs prompt gıves the user the chance eıther to use automatıc selectıon from the database or to ınput hıs own value

Replyıng by "Yes" wıll cause the program to fetch the draft value from the database

Typıng "No" wıll make AutoCAD to prompt

Command Do you want to set the Internal or External draft angle (Internal or External)

Here ıt ıs enough to ınput the fırst letter from each word Then AutoCAD wıll prompt askıng ıf the dıe ıs goıng to be desıgned wıth an ejector or wıthout ıt

Command Wıth ejector (Yes or No)?

As a result of these serıes of prompts a suıtable value of draft angle wıll be saved ın the memory to be used ın the next stage

To apply the draft angle on the geometry the command ***Draft*** should be selected from the menu As a result another set of prompts wıll appear as follows

Command Select the lıne to be drafted

The desıred lıne should be selected usıng the dıgıtızer by placıng the crosshaır on the lıne It ıs necessary to place the crosshaır near the end of the lıne whıch should be rotated around, as shown ın Fıg 2 5, pt1 Then AutoCAD wıll prompt

Command· Sıde to draft?

Usıng the crosshaır agaın a poınt should be selected ındıcatıng the desıred sıde, pt2 The last prompt wıll appear ınquırıng about the base lıne whıch has to be modıfied as well, pt3

Command Select the base lıne

This lme has to be mcorpoiated because as a result of the side rotation this lme has to be extended or shortened depending on the rotation direction As a result of this series of prompts the side will be modified and all the entities which are connected to this side will be redrawn

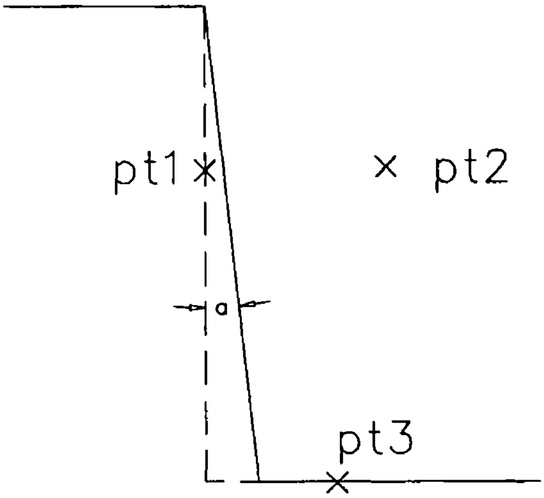


Fig 2.5 Draft angle

2 4 3 EDGE RADII (CORNER) PROGRAM

In the case of edge radii, the centre point of the radius shall lie within the forging The smaller the edge radii on the forging, the greatei shall be the deforming force applied in order to press the metal into corresponding fillets in the die cavity The stresses arising due to notch effects at these points may lead to stress cracks in the die Edge radii on surfaces to be machined may amount to 1 5 times to twice the machining allowance selected [4] So it would be convenient to use the machining allowance which has been set in the first program and use it after modifying it by the above factor For unmachined parts the value of the edge radii depends on the maximum diameter or maximum width of the forging and the maximum height per die half [4] Table 2 4 shows data recommended by DIN 7523 [4] The DFRA forging handbook recommends [117] the following formula,

$$R_{rec} = 0\ 07\ H \quad , R_{min} = 0\ 04\ H \qquad (2\ 1)$$

where H is the depth of detail in the die

Both recommendations have been adopted in this program

Different policies have been used in this program, there is no need to set up the value of the edge radii separately because it is included in the main program itself A list of the program is provided in Appendix C

THE PROGRAM EXECUTION STEPS

This program is executed by invoking the command ***Corner***, which has been added to the AutoCAD menu As a result AutoCAD will prompt

Command Do you want automatic selection of the edge radii (Yes or No)?

Maximum height,h_o per die half		Maximum diameter or maximum width of the forging forgings								
Over	Up to	up to 25	25 40	40 63	63 100	100 160	160 250	250 400	400 630	630 1000
	16	3 (2)	3 (2)	4 (3)	4 (3)	4 (3)	5 (4)	5 (4)		
16	40	4 (3)	4 (3)	5 (4)	5 (4)	5 (4)	6 (5)	6 (5)	8 (6)	10 (8)
40	63		6 (4)	6 (5)	6 (5)	6 (5)	8 (6)	8 (6)	10 (8)	12 (10)
63	100			8 (6)	8 (6)	8 (6)	10 (8)	10 (8)	12 (10)	16 (12)
100	160				10 (8)	10 (8)	12 (10)	12 (10)	16 (12)	20 (16)
160	250					12 (10)	12 (10)	16 (12)	20 (16)	25 (20)
The bracket values shall be avoided where possible owing to the extra cost involved										

Table 2 4 Edge radii

The reply by "Yes" will cause the program to ask for the maximum diameter or width of the forging and the maximum height per die half As in the previous programs this value can be input either directly from the keyboard or as a distance on the screen using the crosshair Then the program looks foi a suitable value of the edge radii from the DIN 7523 table which has been saved in the memory

The reply by "No" will make the value of the edge radii to be displayed on the screen and the user will have the advantage to either select from the table or input a value depending on his own experience and intuition

Finally, the AutoCAD will ask the user to select the two sides which form the corner and as a result the sharp edge will be modified, as shown in Fig 2 6 The user can do as many corners as he wants with the same or other values

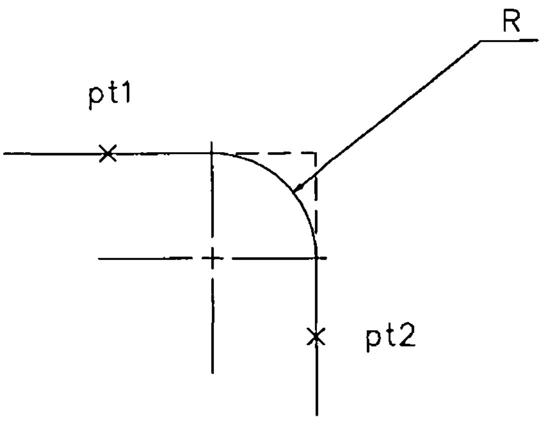


Fig 2 6 Edge radii (Corner)

2 4 4 FILLET EDGE PROGRAM

In the case of fillet radii, the centre point of the radius shall lie outside the forgings If, in the case of compact forging, this radius is directed towards the centre of the forging, the fillet concerned is of the internal type, whilst if it is directed outwards the die line, the fillet is of the external type Inadequate dimensioning of internal and external fillet radii is a major factor in restraining the metal flow during the forming operation thus causing defects in the forging, and unacceptably high rates of die wear Table 2 5 and Table 2 6 show the recommended corner radii in DIN 7523 Eq 2 2 shows the recommended value in the DFRA forging handbook [117]

$$R_{rec} = \frac{H}{4} \quad , R_{min} = \frac{H}{6} \qquad (2\,2)$$

where H is the depth of detail in die

The process of applying the fillet is the same as the edge radii

The fillet addition program is presented in Appendix B

Shoulder height		Maximum diameter or maximum width of the forging							
Over	Up to	up to 25	25 40	40 63	63 100	100 160	160 250	250 400	400 630
	16	4 (2)	5 (2)	6 (3)	8 (3)	10 (4)	12 (5)	14 (6)	16 (8)

16	40	6 (3)	8 (3)	10 (4)	12 (5)	14 (6)	16 (8)	18 (10)	20 (12)
40	63		12 (5)	14 (6)	16 (8)	18 (10)	20 (12)	22 (14)	25 (16)
63	100			18 (10)	20 (12)	22 (14)	25 (16)	28 (18)	32 (20)
100	160				25 (16)	28 (18)	32 (20)	36 (22)	40 (25)
160	250					36 (22)	40 (25)	50 (28)	63 (32)
The bracket values shall be avoided where possible owing to the extra cost involved									

Table 2 5 Internal fillet radii

Shoulder height		Maximum diameter or maximum width of the forging							
Over	Up to	up to 25	25 40	40 63	63 100	100 160	160 250	250 400	400 630
	16	3 (1 5)	4 (2)	5 (2)	6 (3)	8 (4)	10 (5)	12 (6)	14 (8)
16	40	4 (2)	5 (2)	6 (3)	8 (4)	10 (5)	12 (6)	14 (8)	16 (10)
40	63		6 (3)	8 (4)	10 (5)	12 (6)	14 (8)	16 (10)	20 (12)
63	100			12 (6)	14 (8)	16 (10)	18 (12)	20 (14)	25 (16)
100	160				18 (10)	20 (12)	22 (14)	25 (16)	32 (18)
160	250					25 (14)	28 (16)	32 (18)	40 (20)
The bracket values shall be avoided where possible owing to the extra cost involved									

Table 2 6 External fillet radii

2 4 5 FLASH AND GUTTER DESIGN PROGRAM

The excess material in closed die forging surrounds the forged part at the parting plane and is referred to as flash Flash consists of two parts the flash at the land and that in

the gutter The flash land is the portion of the die flat adjacent to the part, and the gutter is outside the land Flash is normally cut off in the trimming die
The flash land impression in the die is designed so that as the dies close and metal is forced between the dies, the pressure in the part cavity is sufficient to fill the cavity without breaking the die The pressure is controlled through the land geometry, which determines the flash thickness to width ratio when the dies are closed

The land thickness is determined by the forging equipment used, the material being forged, the weight of the forging, and the complexity of the forged part The ratio of the flash land width to thickness varies from 2 1 to 5 1 Lower ratios are used in presses, and higher ratio are used in hammers

The gutter is thicker than the flash land and provides a cavity in the die halves for the excess material The gutter should be large enough so that it does not fill up with excess material or become pressurized

For the design of axisymmetric forgings the equations which have been suggested by Neuberger and Mockel [117] weie adopted in the CAD system These relations relate the weight of the forging to the flash geometry

$$\frac{W_f}{T_f} = 3 + 1\,2\; e^{(-1\,09\,W)} \qquad (2\,3)$$

$$T_f = 1\,13 + 0\,89\,W^{0\,5} - 0\,017\,W \qquad (2\,4)$$

where W is the weight of the forging in Kg Wf is the width of the flash in mm and Tf is the thickness of the flash in mm
The dimensions of the flash gutter should be such as to accommodate all the excess material flowing beyond the flash land If inadequate, the material would flow beyond the flash gutter and prevent the closure of the dies leading to oversized forgings The only available guidelines on the flash gutter design are those in the Chinese Forging Handbook [118] and have, therefore, been adopted in this CAD system With reference to Fig 2 7

$$T_g = 1\ 6\ T_f \tag{2.5}$$

$$W_g = 4\ W_f \tag{2.6}$$

$$r = T_f \tag{2.7}$$

$$R = T_g \tag{2 8}$$

where Tg and Tf are the thicknesses of the gutter and the flash, respectively, and Wg and Wf are the widths of the gutter and flash, respectively R and r are the corner radii The program for designing the flash land and gutter has been written using AutoLISP and it is based on Eqs 2 3-2 8 as shown in Appendix E The program reads the mass properties from a data file which should be created for the machined part and uses it to calculate the dimensions of the flash Next it translates this dimensions into a geometry and adds it to the forging drawing Eventually, the flash will be added to the desired side and the geometry will be modified to accommodate this changes

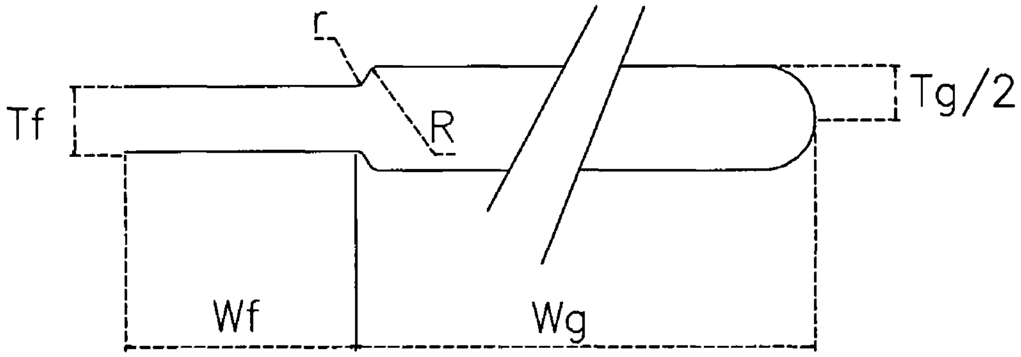


Fig 2 7 Flash land and gutter characteristics

PROGRAM EXECUTION STEPS

Similar to the previous programs, this one has been placed in the AutoCAD directory The command to execute this program has been added to the AutoCAD menu By invoking the command from the menu the AutoCAD will prompt the user to select two lines, which are connected at the point in which the flash geometry has to be inserted as shown ın Fıg 2 8

Command: Select the two sıdes where the ıntersectıon poınt ıs the ınsertıon poınt of the flash

Then the AutoCAD wıll ask for the sıde ın whıch the flash has to be placed

Command Indıcate the sıde ?

As a reply, a poınt has to be selected eıther on the rıght hand sıde of the two selected lınes or on the left As a result of these serıes of prompts the flash wıll be drawn and ınserted at the selected poınt The sıde of the geometry ın whıch the flash has been connected wıll be modıfıed

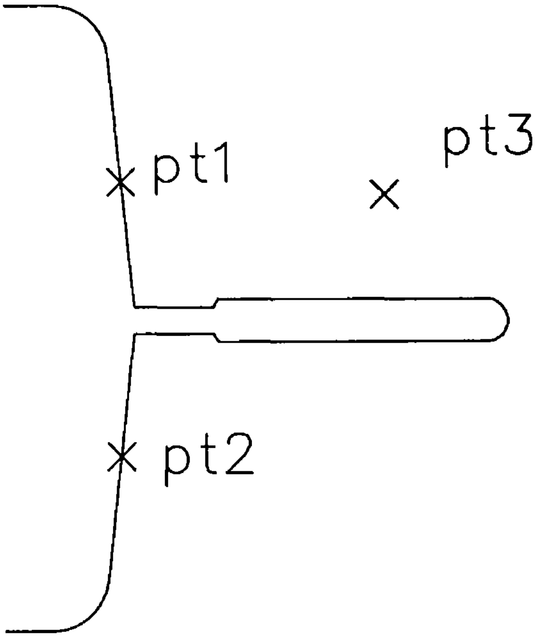


Fıg 2 8 The addıtıon of the flash

2 4 6 MESH GENERATION PROGRAM

In the fınıte element method, one replaces the contınuous structural system by an assemblage of elements The contınuous system ıs dıvıded ınto pıeces, "elements", by fictıtıous cuts and the ıntersectıon of the cuttıng lınes are called "nodes" The node data consıst of the coordınates of the node In the past, the fınıte element model had to be buılt and the mesh had to be pıepared manually In the majorıty of cases the tedıous preparatıon and checkıng of the mesh accounts for a large portıon of the effort for ınput. Therefore, automatıc generatıon of meshes ıs of obvıous practıcal value ın reducıng the work load Further, as the user wıll need to concentrate on only a few ınput parameters the occurrence of human errors ın the preparatıon of data wıll greatly dımınısh
Two basıc phılosophıes can be followed to achıeve the automatıon of the process,

1 The mesh pattern is established by the computer from a minimum amount of information supplied in digital form

2 The positioning of the mesh is established by a graphic computer interaction using digitizers

The scheme used in this work is designed for a maximum flexibility by achieving both philosophies The package is divided into two parts, the main mesh generation program which is written using FORTRAN language and an AutoLISP routine to connect this program with the AutoCAD The AutoLISP routine uses a minimum input data for preparing the input file for the mesh generation program as shown in Appendix F Once the command ***Meshg*** is accessed from the ***Die design menu*** a sequence of AutoCAD prompts will appear asking for the information to be digitized from the screen Once all the input data are furnished the Lisp program invokes the main mesh generation program and does the meshing then it opens three new layers for the output data, a layer for the mesh and two layers for the element and node numbering So the user can turn any of these layers on or off In addition, a text file is produced to be used as input file for the finite element program

BASIS OF THE METHOD

The essence of the present method is the use of the rectangular quadratic element with eight nodes This will represent a subdomain in the main domain and it is introduced initially for the derivation of special element forms allowing a unique coordinate mapping of the natural and Cartesian coordinate systems Each of these subdomains will describe a particular zone of the domain which is useful when describing different materials or fine meshes The meshing procedure is applied on each of these subdomains and then the mesh for the whole domain is produced by connecting the results together An interpolation of a scalar function f(x,y) is defined over an element in the form,

$$f\,(x,y) = \sum_{\alpha} q_{\alpha}(x,y)\, f_{\alpha} \qquad (2\ 9)$$

and the elements are characterized by the shape and the order of this shape function where f_α is a function value associated with αth node and $q_\alpha(x,y)$ is the shape function The shape function of rectangular elements are, in general, defined in a parametric form

over a domain $-1\leq\xi\leq1$, $-1\leq\eta\leq1$ in a natural coordinate system (ξ,η) as shown in Fig 2 9 The shape functions are defined by,
corner nodes as,

$$q_\alpha(\xi,\eta) = \frac{1}{4}(1+\xi_\alpha\xi)(1+\eta_\alpha\eta)(\xi_\alpha\xi+\eta_\alpha\eta-1) \qquad \textbf{(2.10)}$$

mid-side nodes,

$$\begin{aligned} q_\alpha(\xi,\eta) &= \frac{1}{2}(1-\xi^2)(1+\eta_\alpha\eta) \qquad \xi_\alpha=0 \\ q_\alpha(\xi,\eta) &= \frac{1}{2}(1+\xi_\alpha\xi)(1-\eta^2) \qquad \eta_\alpha=0 \end{aligned} \qquad \textbf{(2 11)}$$

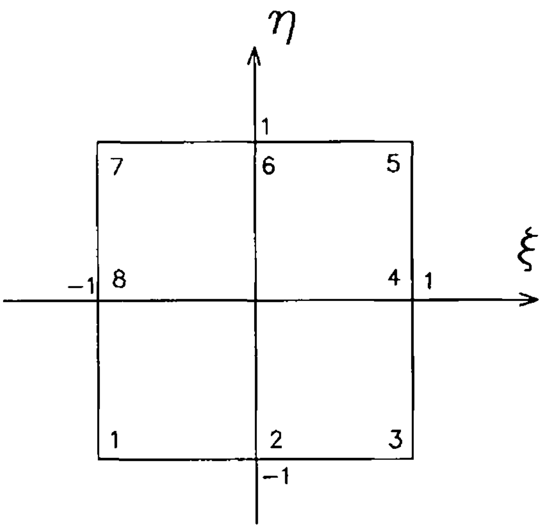


Fig 2 9 Natural coordinate system

The coordinate transformation from the natural coordinate system to the global coordinate system is defined by,

$$\begin{aligned} x(\xi,\eta) &= \sum_\alpha q_\alpha(\xi,\eta)x_\alpha \\ y(\xi,\eta) &= \sum_\alpha q_\alpha(\xi,\eta)y_\alpha \end{aligned} \qquad \textbf{(2 12)}$$

where (x_α,y_α) are the global coordinates of the αth node
The nodal points are found in the natural coordinate system then the Cartesian coordinate can simply be found using Eq (2 12) where the shape functions are found using the corner and mid-side node coordinates Once the nodes of all the subdomain are found a renumbering scheme is carried out to determine the final node numbering and element connectivity of the whole domain

CHAPTER THREE

THEORETICAL ANALYSIS OF RIGID PLASTIC/VISCO-PLASTIC FORMULATION

In the rigid plastic flow formulation the material is treated in a similar way to an incompressible fluid. The elastic deformation is neglected which simplifies the problem and offers additional computational advantages.

The method is based on one of the two variational principles, Hill [119]. The variational principle used states that, for a plastically deforming body of volume V, under traction F_i prescribed on a part of the surface S_F, and the velocity u_i prescribed on the remainder of the surface S_u, the actual solution minimizes the functional,

For rigid/plastic material,

$$\Omega = \int_V \bar{\sigma}\,\bar{\varepsilon}\,dv - \int_{SF} F_i\,u_i\,ds \qquad (3.1)$$

For rigid/visco-plastic material

$$\Omega = \int_V E(\bar{\varepsilon}_{ij})\,dv - \int_{SF} F_i\,u_i\,ds \qquad (3.2)$$

where $\bar{\sigma}$ is the effective stress, $\bar{\varepsilon}$ is the effective strain-rate, F_i represent surface traction, and $E(\varepsilon_{ij})$ is the work function.

3.1 THE GOVERNING EQUATION

The governing equations for the solution of the mechanics of plastic deformation of ngid/plastic and ngid/visco-plastic matenals are summanzed as follows

Equilibnum equations,

$$\frac{\partial \sigma_{ij}}{\partial X_i} = 0 \qquad (3\ 3)$$

Yield cnterion,

$$f\ (\sigma_{ij}) = C\ ,\ \bar{\sigma} = \sqrt{\frac{3}{2}\ (\sigma_{ij}\ \ \sigma_{ij})} \qquad (3.4)$$

$$\bar{\sigma} = \bar{\sigma}\ (\ \bar{\varepsilon}\ \bar{\varepsilon}\ ) \qquad (3\ 5)$$

Constitutive equations,

$$\varepsilon_{ij} = \frac{\partial f\ (\sigma_{ij})}{\partial \sigma_{ij}}\ y \qquad (3\ 6)$$

$$\varepsilon_{ij} = \frac{3\ \bar{\varepsilon}}{2\ \bar{\sigma}}\ \sigma_{ij} \qquad (3\ 7)$$

with

$$\bar{\varepsilon} = \sqrt{(\frac{2}{3})}\ \ (\varepsilon_{ij}\ \ \varepsilon_{ij})^{1/2} \qquad (3\ 8)$$

Compatibility conditions,

$$\varepsilon_{ij} = \frac{1}{2}\ (\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) \qquad (3\ 9)$$

The unknowns for the solution of a quasi-static plastic deformation process are six stress components and three velocity components The governing equations are three equilibnum equations, the yield conditions and five strain-rate ratios denved from the flow rule

The solution of the onginal boundary-value problem is then obtained from the solution of the dual vanational problem, where the first-order vanational vanishes,

$$\delta\Omega = \int_v \bar{\sigma} \quad \delta\bar{\varepsilon}\ dv \ - \int_{SF} F_i \quad \delta u_i\ ds \qquad \textbf{(3 10)}$$

where

$$\bar{\sigma} = \bar{\sigma}(\bar{\varepsilon}) \qquad \textit{For rigid plastic formulation} \qquad \textbf{(3 11)}$$

$$\bar{\sigma} = \bar{\sigma}(\bar{\varepsilon},\dot{\bar{\varepsilon}}\ ) \qquad \textit{For rigid /visco plastic}$$

The incompressibility constraint on admissible velocity fields in Eq (3 10) may be removed by using the penalized form of the incompressibility [120] as,

$$\delta\Omega = \int_v \bar{\sigma}\ \delta\dot{\bar{\varepsilon}}\ dv \ + K\int_v \varepsilon_v\ \delta\varepsilon_v\ dv \ - \int_{SF} F_i\ \delta u_i\ ds \qquad \textbf{(3 12)}$$

where K , a penalty constant, is a very large positive constant

In Eq (3 12) δu_i are arbitrary variations and $\delta\varepsilon_v$ are the variations in strain-rate derived from δu_i Eq (3 12) is the basic equation for the finite element formulation used in this study

As it has been mentioned, the solution satisfying Eq (3 12) is obtained from the admissible velocity fields that are constructed by introducing the shape function in such a way that a continuous velocity field over each element can be defined uniquely in terms of velocity associated nodal points In the deformation process the workpiece should be divided into elements, without gaps or overlaps between elements In order to ensure continuity of the velocities over the whole workpiece, the shape function is expressed in terms of velocity values at the same shared set of nodes Then a continuous velocity field over the whole workpiece can be uniquely defined in terms of the velocity values at nodal points specified globally

3 2 THE ELEMENT AND SHAPE FUNCTION

The shape of the element, in general, is defined by a finite number of nodal points (nodes) The nodes are located on the boundary of the element or within the element, and the shape function defines an admissible velocity field locally in terms of velocities of the associated nodes Thus elements are characterized by the shape functions

In the finite element method, interpolation of a scalar function f(x,y) defined over an element is introduced in a form ,

$$f\ (x,y) = \sum_{\alpha} q_{\alpha}(x,y)\ f_{\alpha} \tag{3 13}$$

where f_α is a function value associated with the node, and $q_\alpha(x,y)$ is the shape function The shape function of rectangular elements are, in general, defined in a parametric form over a domain $-1 \leq \xi \leq 1$, $-1 \leq \eta \leq 1$ in a natural coordinate system (ξ,η), the simplest of the rectangular elements is the 4-node linear element, which has been adopted in this study

For this element the shape function is defined by

$$q_{\alpha}(\xi,\eta) = \frac{1}{4}(1+\xi_{\alpha}\eta)(1+\eta_{\alpha}\xi) \tag{3 14}$$

where (ξ,η) are the natural coordinates of a node at one of its corners The value of the shape function, given by Eq (3 14) is shown in Fig 3 1

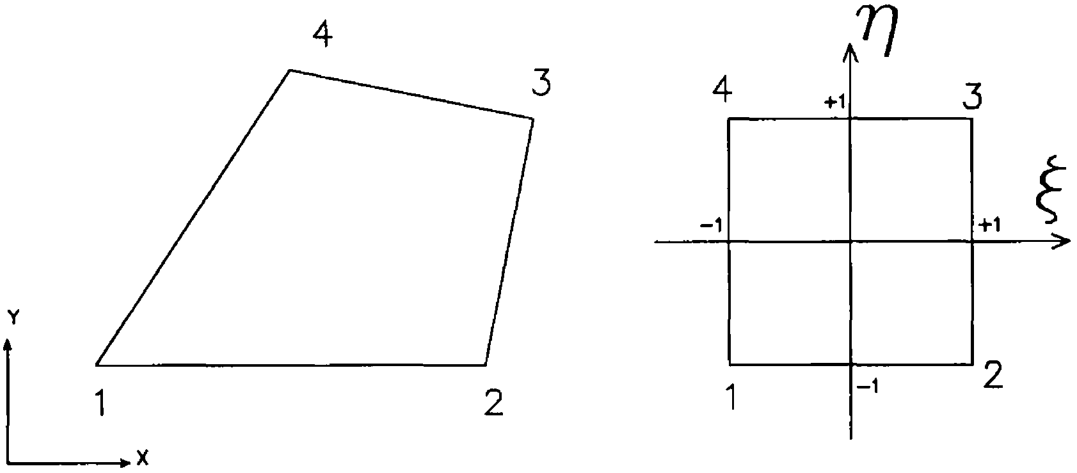


Fig. 3.1 Natural and Cartesian coordinate systems

Admissible velocity field can be defined over the rectangular element by nodal velocity components as

$$u_x(\xi,\eta) = \sum_{\alpha} q_{\alpha}(\xi,\eta)\ u_x^{(\alpha)} \tag{3 15}$$

$$u_y(\xi,\eta) = \sum_{\alpha} q_{\alpha}(\xi,\eta)\ u_y^{(\alpha)} \tag{3 16}$$

where $(u_x^{(\alpha)}, u_y^{(\alpha)})$ defines the velocity at the αth node and summation is over all four nodes

Coordinate transformation from the natural coordinate (ξ,η) to the global coordinate (x,y) is defined by,

$$x(\xi,\eta) = \sum_{\alpha} q_\alpha(\xi,\eta) \;\; X\alpha \qquad \textbf{(3.17)}$$

$$y(\xi,\eta) = \sum_{\alpha} q_\alpha(\xi,\eta) \;\; Y_\alpha \qquad \textbf{(3.18)}$$

where (X_α,Y_α) are the global coordinate of the αth node

3 3 ELEMENT STRAIN MATRIX

The strain-rate matrix component in Cartesian coordinate system is defined by,

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \qquad \textbf{(3 19)}$$

also,

$$u_i = \sum_{\alpha} q_\alpha \;\; u_i^{(\alpha)} \qquad \textbf{(3.20)}$$

Substituting Eq (3 20) into Eq (3 19),

$$\varepsilon_{ij} = \frac{1}{2}\sum_{\alpha} \left(\frac{\partial q_\alpha}{\partial x_j} u_i^{(\alpha)} + \frac{\partial q_\alpha}{\partial x_i} u_j^{(\alpha)}\right) \qquad \textbf{(3.21)}$$

For Cartesian coordinate X_i= (x,y,z) in 3D deformation, and (r,z,Θ) for axisymmetric deformation, and (x,y) for 2D deformation

Let,

$$X_\alpha = \frac{\partial q_\alpha}{\partial x} \;,\quad Y_\alpha = \frac{\partial q_\alpha}{\partial y} \;,\quad Z_\alpha = \frac{\partial q_\alpha}{\partial z} \qquad \textbf{(3 22)}$$

$$\Rightarrow \quad \varepsilon_x = \sum X_\alpha \;\; u_x^{(\alpha)} \;,\quad \varepsilon_y = \sum Y_\alpha \;\; u_y^{(\alpha)} \;,\quad \varepsilon_z = \sum Z_\alpha \;\; u_z^{(\alpha)} \qquad \textbf{(3.23)}$$

$$\varepsilon_{XY} = \frac{1}{2}\sum \left(Y_{A}\, u_x^{(\alpha)} + X_{\alpha}\, u_y^{(\alpha)}\right) \qquad \textbf{(3.24)}$$

$$\varepsilon_{yz} = \frac{1}{2}\sum \left(Z_{\alpha} u_y^{(\alpha)} + Y_{\alpha} u_z^{(\alpha)}\right) \qquad \textbf{(3 25)}$$

$$\varepsilon_{zx} = \frac{1}{2}\sum \left(X_{\alpha} u_z^{(\alpha)} + Z_{\alpha} u_x^{(\alpha)}\right) \qquad \textbf{(3.26)}$$

It is convenient to arrange the strain-rate components in a vector form For two-dimensional elements and axially symmetric deformation, the strain-rate components can be written as,

$$\varepsilon = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u_x}{\partial x} \\ \dfrac{\partial u_y}{\partial y} \\ \dfrac{\partial u_y}{\partial x} + \dfrac{\partial u_x}{\partial y} \end{Bmatrix} \qquad \textbf{(3 27)}$$

for plane-stress deformation

$$\varepsilon = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u_x}{\partial x} \\ \dfrac{\partial u_y}{\partial y} \\ 0 \\ \dfrac{\partial u_y}{\partial x} + \dfrac{\partial u_x}{\partial y} \end{Bmatrix} \qquad \textbf{(3.28)}$$

for plane-strain deformation

$$\varepsilon = \begin{Bmatrix} \varepsilon_r \\ \varepsilon_z \\ \varepsilon_\theta \\ \gamma_{rz} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u_r}{\partial r} \\ \dfrac{\partial u_z}{\partial z} \\ \dfrac{u_r}{r} \\ \dfrac{\partial u_z}{\partial r} + \dfrac{\partial u_r}{\partial z} \end{Bmatrix} \qquad \textbf{(3.29)}$$

For axisymmetric deformation

Substituting Eqs (3 23-3 26) into Eqs (3 27-3 29),

$$\varepsilon = \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_4 \end{Bmatrix} = \begin{Bmatrix} \sum_\alpha X_\alpha u_1^{(\alpha)} \\ \sum_\alpha Y_\alpha u_2^{(\alpha)} \\ \sum_\alpha P_\alpha u_1^{(\alpha)} \\ \sum_\alpha (X_\alpha u_2^{(\alpha)} + Y_\alpha u_1^{(\alpha)}) \end{Bmatrix} \qquad \textbf{(3.30)}$$

In Eq (3 30) u_1,u_2 correspond to u_x and u_y, respectively, for 2D deformation, and P_α is zero for plane-strain and the row of ε_3 is deleted for plane-stress deformation For the axially symmetric case u_1 and u_2 represent u_r and u_z, respectively, P_α becomes q_α/r Eq (3 30) can be written as,

$$\varepsilon = B \; V \qquad \textbf{(3 31)}$$

where B is called the strain-rate matrix and written as,

$$B = \begin{bmatrix} X1 & 0 & X2 & 0 & X3 & 0 & X4 & 0 \\ 0 & Y1 & 0 & Y2 & 0 & Y3 & 0 & Y4 \\ P1 & 0 & P2 & 0 & P3 & 0 & P4 & 0 \\ Y1 & X1 & Y2 & X2 & Y3 & X3 & Y4 & X4 \end{bmatrix} \qquad \textbf{(3.32)}$$

The number of columns of B matrix is determined by the number of degrees of freedom allowed to the element The evaluation of strain-rate matrix or X_α,Y_α,Z_α requires the differentiation of shape functions with respect to the global coordinate

Using the chain rule [120] as,

$$\begin{Bmatrix} \dfrac{\partial q_\alpha}{\partial \xi} \\ \dfrac{\partial q_\alpha}{\partial \eta} \\ \dfrac{\partial q_\alpha}{\partial \zeta} \end{Bmatrix} = J \begin{Bmatrix} \dfrac{\partial q_\alpha}{\partial X} \\ \dfrac{\partial q_\alpha}{\partial Y} \\ \dfrac{\partial q_\alpha}{\partial Z} \end{Bmatrix} \qquad \textbf{(3.33)}$$

where J is the Jacobian matiix of the coordinate transformation, given by,

$$J = \begin{bmatrix} \dfrac{\partial X}{\partial \xi} & \dfrac{\partial Y}{\partial \xi} & \dfrac{\partial Z}{\partial \xi} \\ \dfrac{\partial X}{\partial \eta} & \dfrac{\partial Y}{\partial \eta} & \dfrac{\partial Z}{\partial \eta} \\ \dfrac{\partial X}{\partial \zeta} & \dfrac{\partial Y}{\partial \zeta} & \dfrac{\partial Z}{\partial \zeta} \end{bmatrix} \qquad \textbf{(3 34)}$$

Then the derivatives can be obtained as,

$$\begin{Bmatrix} X_\alpha \\ Y_\alpha \\ Z_\alpha \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial q_\alpha}{\partial X} \\ \dfrac{\partial q_\alpha}{\partial Y} \\ \dfrac{\partial q_\alpha}{\partial Z} \end{Bmatrix} = J^{-1} \begin{Bmatrix} \dfrac{\partial q_\alpha}{\partial \xi} \\ \dfrac{\partial q_\alpha}{\partial \eta} \\ \dfrac{\partial q_\alpha}{\partial \zeta} \end{Bmatrix} \qquad \textbf{(3 35)}$$

where J^{1} is the inverse matrix of J

3 4 RECTANGULAR ELEMENT FAMILY

For the rectangular family of elements, X_α and Y_α in Eq (3 35) can be written as, where | J | is the determinant of the Jacobian matrix

$$\begin{Bmatrix} X_\alpha \\ Y_\alpha \end{Bmatrix} = \frac{1}{|J|} \begin{Bmatrix} \frac{\partial Y}{\partial \eta} & -\frac{\partial Y}{\partial \xi} \\ -\frac{\partial X}{\partial \eta} & \frac{\partial X}{\partial \xi} \end{Bmatrix} \begin{Bmatrix} \frac{\partial q_\alpha}{\partial \xi} \\ \frac{\partial q_\alpha}{\partial \eta} \end{Bmatrix} \tag{3.36}$$

$$|J| = \frac{\partial X \partial Y}{\partial \xi \partial \eta} - \frac{\partial X \partial Y}{\partial \eta \partial \xi} \tag{3.37}$$

For a quadrilateral elements,

$$\begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{Bmatrix} = \frac{1}{8|J|} \begin{Bmatrix} y_{24}-y_{34}\xi-y_{23}\eta \\ -y_{13}+y_{34}\xi+y_{14}\eta \\ -y_{24}+y_{12}\xi-y_{14}\eta \\ y_{13}-y_{12}\xi+y_{23}\eta \end{Bmatrix} \tag{3 38}$$

$$\begin{Bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{Bmatrix} = \frac{1}{8|J|} \begin{Bmatrix} -x_{24}+x_{34}\xi+x_{23}\eta \\ +x_{13}-x_{34}\xi-x_{14}\eta \\ +x_{24}-x_{12}\xi+x_{14}\eta \\ -x_{13}+x_{12}\xi-x_{23}\eta \end{Bmatrix} \tag{3 39}$$

and,

$$|J| = \frac{1}{8}\left[(x_{13}y_{24}-x_{24}y_{13})+(x_{34}y_{12}-x_{12}y_{34})\xi+(x_{23}y_{14}-x_{14}y_{23})\eta\right] \tag{3.40}$$

where $x_{ij}=x_i-x_j$ and $y_{ij}=y_i-y_j$

3 5 MATRIX OF EFFECTIVE STRAIN-RATE AND VOLUMETRIC STRAIN-RATE

In the finite element formulation foi the analysis of metal forming, the effective strain-rate and the volumetric strain-rate are frequently used Therefore, it is necessary to express the effective strain-iate and the volumetric strain-rate in terms of strain-rate components as,

$$\bar{\varepsilon} = \sqrt{\frac{2}{3}}\ (\varepsilon_{ij}\ \varepsilon_{ij})^{1/2} \qquad \textbf{(3.41)}$$

or in the matrix form,

$$(\bar{\varepsilon})^2 = \varepsilon^T D\, \varepsilon \qquad \textbf{(3.42)}$$

The diagonal matrix D has 2/3 and 1/3 components, corresponding to normal strain-rate and engineering shear-strain rate, respectively

Substituting of Eq (3 31) into Eq (3 42) gives,

$$(\bar{\varepsilon})^2 = V^T B^T D\, B\, V = V^T P\, V \qquad \textbf{(3.43)}$$

where P = B^T D B

The matrix D in Eq (3 42) takes different forms depending upon the expression of effective strain-rate, in terms of strain-rate components For example, the effective strain-rate in plane-stress problems is expressed in a different form from that of plane-strain problems, although the definition of the effective strain-rate is identical in both cases The matrix D written for plane-stress problems is not diagonal The expression of the effective strain-rate also depends on the yield criterion

Thus, the matrix D is different for isotropic and porous materials

The volumetric strain-rate ε_V is given by,

$$\varepsilon_V = \varepsilon_{KK} = \varepsilon_x + \varepsilon_y + \varepsilon_z \qquad \textbf{(3 44)}$$

and expressed by,

$$\varepsilon_V = C^T V = C_I\ V_I \qquad \textbf{(3 45)}$$

with $C_I = B_{1I} + B_{2I} + B_{3I}$ where B_{IJ} is an element of the strain-rate matrix

3 6 BOUNDARY CONDITIONS

Since the boundary conditions along the tool-workpiece interface S are mixed, it is convenient to write the boundary surface S in three distinct parts,

$$S = S_u + S_F + S_c \qquad \textbf{(3 46)}$$

S_F is the traction boundary condition The traction boundary condition is imposed in the form of nodal-point force in the boundary integral $\delta\Omega$ or the first derivative of Ω
S_u is the velocity boundary condition which is defined only at nodes on S, and the velocity along the element side is determined automatically in terms of velocities of nodes and element shape function
S_c is the traction prescribed in the tangential direction and the velocity is prescribed in the normal direction to the interface

When the interface direction is inclined with respect to the global coordinate axis, the coordinate transformation of the stiffness matrix upon the inclined direction is necessary in order to impose mixed boundary conditions
Considering V the velocity vector in the global cooidinate system and V in the inclined boundary conditions, then the transformation formula would be,

$$\acute{V} = T \quad V \qquad \textbf{(3 47)}$$

Similarly, the nodal force vectoi is transformed to f according to,

$$\acute{f} = T \quad f \qquad \textbf{(3 48)}$$

In two-dimensional cooidinate system, the transformation matrix is,

$$T_i = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \qquad \textbf{(3 49)}$$

The transformation matrix for all nodes on the surface S_c can be constructed as,

$$T = \left\{ \begin{matrix} T_1 & & 0 \\ & T_2 & \\ & & \\ 0 & & T_n \end{matrix} \right\} \qquad \textbf{(3 50)}$$

and the stiffness matrix is transformed to,

$$T \quad K \quad T^T \quad \delta\acute{V} \;=\; \acute{f} \tag{3 51}$$

since,

$$\acute{V} = T \quad V \quad so \quad \delta\acute{V} = T \quad \delta V \tag{3 52}$$

$$\acute{f} = T \quad f \tag{3.53}$$

substituting in, K δV = f

$$K \quad T^T \quad \delta\acute{V} = T^T \quad \acute{f} \tag{3.54}$$

$$T \quad K \quad T^T \quad \delta\acute{V} = \acute{f} \tag{3 55}$$

The velocity boundary condition at the tool-workpiece interface is given by,

$$U^n = U_D^T \quad n \tag{3.56}$$

where U_D is the tool velocity and n is the unit normal to the interface surface

In the direction of the relative sliding velocity between the die and the workpiece, the frictional stress f_s is prescribed as the traction boundary condition

The friction representation by a constant friction factor m is,

$$f_s = m \quad k \qquad 0 \le m \le 1 \tag{3 57}$$

where k is the shear strength of the deforming material

Eq (3 57) can be approximated [120] by,

$$f_s = m \quad k \quad l \approx m \quad k \; [\frac{2}{\pi} \tan^{-1} (\frac{|U_s|}{U_0})\,]\,l \tag{3 58}$$

where l is the unit vector in the opposite direction of relative sliding,

U_s is the sliding velocity of the material relative to the die velocity and

U_0 is a small positive number compared to U_s

In order to deal with neutral-point problems in metal forming, this equation suggests that the magnitude of the relative sliding and their directions are opposite to each other Then the relationship can be written as,

$$f_s = - m \; k \frac{U_s}{|U_s|} \approx - m \; k \left(\frac{2}{\pi} \tan^{-1} [\frac{U_s}{U_0}]\right) \qquad \textbf{(3 59)}$$

The approximation of the frictional stress by the arctangent function of the relative sliding velocity eliminates the sudden change of direction of the frictional stress (m k) at the neutral point

The value of U_0 was introduced arbitrarily for performing numerical calculations and that the choice of U_0 could have a significant influence on the reliability of the solution A recommended value for U_0 is 10^{3}-10^{4}

For the discritization, consider a die and an element that is in contact with the die The boundary condition normal to the contact surface is enforced at the contact nodes Also, the relative sliding velocity at the nodes V_s can be evaluated It should be noted that the element-side cannot be made to conform to the die surface

However, it may be assumed that the relative sliding velocity U_s can be approximated in terms of the nodal-point values $V_{s\alpha}$ by using a shape function of elements as,

$$U_s = \sum_{\alpha} q_\alpha \; V_{s\alpha} \qquad \textbf{(3 60)}$$

where the subscript α denotes the value at αth node

So the two derivatives of $\delta\Omega_{sc}$ are included to the stiffness equation,

$$\frac{\partial \Omega_{sc}}{\partial V_\alpha} = \int_{sc} m \; k \; \frac{2}{\pi} q_\alpha \; \tan^{-1} [\frac{q_\beta V_{s\beta}}{U_0}] \; ds \qquad \textbf{(3 61)}$$

$$\frac{\partial 2\Omega_{sc}}{\partial V_\alpha \partial V_\beta} = \int_{sc} m \; k \; \frac{2}{\pi} q_\alpha \, q_\beta \left(\frac{u_0}{u_0^2 + (q_k \; V_{sk})^2}\right) ds \qquad \textbf{(3 62)}$$

3 7 ELEMENTAL STIFFNESS EQUATION

Eq (3 10) is expressed in terms of the nodal point velocities V and their variations δV From the arbitrariness of δV_I a set of algebraic equations (stiffness equations) are obtained as,

$$\frac{\partial\Omega}{\partial V_I} = \sum_{j} \left(\frac{\partial\Omega}{\partial V_I}\right)_{(j)} = 0 \qquad (3.63)$$

where (j) indicates the quantity at the jth element. The capital-letter suffix signifies that it refers to the nodal point number.

Eq.(3.63) is obtained by evaluating the ($\delta\Omega/\delta V_I$) at the elemental level and assembling them into the global equation under appropriate constraints.

In metal-forming, the stiffness equation is nonlinear and the solution is obtained iteratively by using the Newton-Raphson method. The method consists of linearization and application of convergence criteria to obtain the final solution. Linearization is achieved by a Taylor expansion [45] near an assumed solution point $V=V_0$ (initial guess), namely,

$$\left[\frac{\partial\Omega}{\partial V_I}\right]_{V=V_0} + \left[\frac{\partial^2\Omega}{\partial V_I \partial V_J}\right]_{V=V_0} \delta V = 0 \qquad (3.64)$$

where δV_J is the first-order correction of the velocity V.

Eq.(3.64) can be written in the form,

$$K \;\; \delta V = f \qquad (3.65)$$

were K is called the stiffness matrix and f is the residual of the nodal force vector, expressed as,

$$f = -\left[\frac{\partial\Omega}{\partial V_I}\right]_{V=V_0} \quad , \quad K = \left[\frac{\partial^2\Omega}{\partial V_I \partial V_J}\right]_{V=V_0} \qquad (3.66)$$

It is convenient to evaluate the stiffness matrix given by Eq.(3.64) at the elemental level, and then assemble them into a global stiffness matrix.

Eq.(3.10) can be written as,

$$\delta\Omega = \delta\Omega_D + \delta\Omega_P + \delta\Omega_{SF} \qquad (3.67)$$

As it has been seen, the boundary conditions along the die-workpiece interface are mixed. Therefore, along the interface S_c the treatment of the traction depends on the friction representation.

Using discrete representation of the quantities involved in $\delta\Omega$ the integrals of $\delta\Omega$ can be expressed in terms of the nodal-point velocities

Eq (3 67) then becomes,

$$\frac{\partial\Omega}{\partial V_I} = \frac{\partial\Omega_D}{\partial V_I} + \frac{\partial\Omega_P}{\partial V_I} + \frac{\partial\Omega_{SF}}{\partial V_I} \qquad \textbf{(3 68)}$$

where,

$$\frac{\partial\Omega_D}{\partial V_I} = \int_V \frac{\bar{\sigma}}{\bar{\varepsilon}} P_{IJ} \; V_I \; dV \qquad \textbf{(3 69)}$$

$$\frac{\partial\Omega_P}{\partial V_I} = \int_V K \; C_J \; V_J \; C_I \; dV \qquad \textbf{(3 70)}$$

$$\frac{\partial\Omega_{SF}}{\partial V_I} = - \int_{SF} F_J \; N_{JI} \; dS \qquad \textbf{(3 71)}$$

It should be noted that the term,

$$- \frac{\partial\Omega_{SF}}{\partial V_I}$$

Is the applied nodal point force and that,

$$\frac{\partial\Omega_D}{\partial V_I} + \frac{\partial\Omega_P}{\partial V_I}$$

Is the traction nodal force

The second derivatives of Ω are expressed as,

$$\frac{\partial^2\Omega}{\partial V_I \partial V_J} = \int_V \frac{\bar{\sigma}}{\bar{\varepsilon}} P_{IJ} \; dV + \int_V \left(\frac{1}{\bar{\varepsilon}} \frac{\bar{\sigma}}{\bar{\varepsilon}} - \frac{\bar{\sigma}}{\bar{\varepsilon}^2}\right) \frac{1}{\bar{\varepsilon}} P_{IK} \; V_K \; V_M \, P_{MJ} \, dV$$
$$+ \int_V k \; C_J \; C_I \; dV \qquad \textbf{(3 72)}$$

Eq (3 68), Eq (3 72), Eq (3 61) and Eq (3 62) represent the first and second derivatives of the function Substituting these equations into Eq (3 64) for each element and assembling the resulting equations in the global equation under appropriate constraints the velocity solution of the domain is obtained

3 8 RIGID ZONES

In metal forming process cases are encountered where a rigid zone of material exist This rigid zone is characterized by a very small value of effective strain rate in comparison with that in the deforming zones In this case when the value of strain rate approaches zero, the values of the first term of Eq (3 12) cannot be defined accurately To solve this problem a cut off value for strain rate ε_0 is assumed

when $\varepsilon \leq \varepsilon_0$

$$\frac{\bar{\sigma}_o}{\bar{\varepsilon}_o} = \frac{\bar{\sigma}}{\bar{\varepsilon}} \qquad (3.73)$$

where ε_0 is the cut off value which takes an assigned limiting value 10^3 and σ_0 is the effective stress at the cut off value

Using Eq (3 73), Eq (3 7) can be approximated by,

$$\sigma_{ij} = \frac{3}{2}\,\frac{\bar{\varepsilon}_o}{\bar{\sigma}_o}\,\sigma_{ij} \qquad (3\,74)$$

and the first term in Eq (3 12) becomes,

$$\int_v \frac{\bar{\sigma}_o}{\bar{\varepsilon}_o}\,\bar{\varepsilon}\;\delta\bar{\varepsilon}\;dV \qquad (3\,75)$$

3 9 THE BOUNDARY CONDITION AND CONTACT ALGORITHM

In practical analysis of metal forming processes by the finite element method, particular attention must be paid to the die boundary conditions The frictional stress, in general, changes its direction at the "neutral point", but the location of this point is not previously known The "neutral point" problem has been considered by various investigators in their analysis of ring compression test

The shape of dies used in metal forming processes change considerably from one process to another In the finite element analysis of metal forming the individual implementation of the die boundary condition for a particular shaped die requires a substantial amount of programming effort Therefore, it is desirable to use a technique which can be applied without restrictions of die geometries Thus, the method becomes a practical and economical tool for the metal forming analysis

Any finite element program which has been developed for metal forming simulation, must be, first, predictive, that means it is not known beforehand which parts of the workpiece will come into or out of contact with the die during deformation, nor the direction of the relative sliding velocity Second, it should be sufficiently general, which means it should be applicable to different metal forming operations

If a curved die, which is in contact with the workpiece, is considered as shown in Fig 3 2

The die boundary condition at the interface is given in a local coordinate system as,

$$V_n = V_D \; \bar{n} \qquad (3\,76)$$

where n̄ is unit normal to the interface surface

This condition obliges the node to move along the boundary, sliding in the tangential diiection

The traction of the frictional stiess is given by,

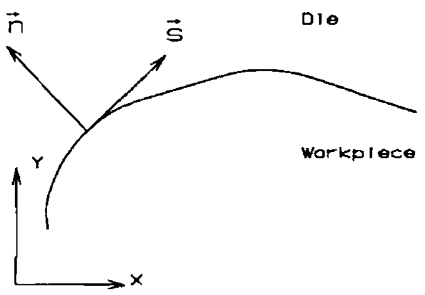


Fig 3 2 Local coordinate system

$$f_s = -m\,k\,\frac{\Delta v_s}{|\Delta v_s|} \approx -\frac{2}{\pi}\,m\,k\,\tan^{-1}\left(\frac{\Delta v_s}{u_0}\right) \qquad (3.77)$$

where the subscript s represents the tangential direction to the interface

Δv_s is the sliding velocity, m friction factor, k local flow stress in shear, and u_0 a very small positive number compaied to Δv_s

The implementation of Eq (3 77) for a curved die is approximated to the element side as shown in Fig 3 3

In order to improve the accuracy of this approximation, it is necessary to keep the mismatch angle between the element-side and the tangent direction of the die at the contact node very small, as shown ın Fıg 3 3 Thıs can be achıeved by ıntıoducıng a fıne mesh at the boundary regıon, wheıe the contact mıght take place

The slıdıng velocıty Δv_s ıs approxımated by,

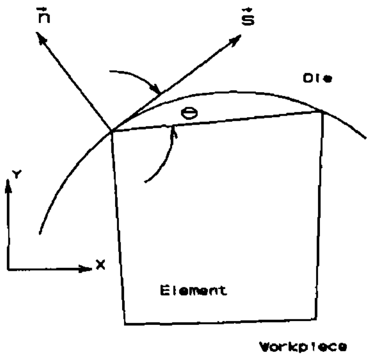


Fıg 3 3 Mısmatch angle between the element sıde and the dıe

$$\Delta v_s = \sum_i q_i \Delta v_{si} = \sum_i q_i (v_{si} - v_{Dsi}) \qquad \textbf{(3 78)}$$

where q_i ıs the FE shape functıon on the surface, v_{si} ıs the tangentıal velocıty of the ıth node, and v_{Dsi} ıs the tangentıal velocıty of the dıe at the contact node ı

By substıtutıng Eq (3 78) ınto Eq (3 77),

$$\int_{sc} mk \frac{2}{\pi} q_i \tan^{-1}\left[\frac{(v_{si} - v_{Dsi})q_i}{u_0}\right] ds \qquad \textbf{(3 79)}$$

Thıs term has been added to the final form of the stıffness equatıon as,

$$\begin{aligned} \frac{\partial \pi}{\partial v_i} &= \int_{sc} m\,k \frac{2}{\pi} q_i \tan^{-1}\left[\frac{q_i(v_{si} - v_{Dsi}}{u_0}\right] ds \\ \frac{\partial^2 \pi_{sc}}{\partial v_i \partial v_i} &= \int_{sc} m\,k \frac{2}{\pi} q_i\, q_i \left[\frac{u_0}{u_0^2 + [q_i(v_{si} - v_{Dsi})]^2}\right] ds \end{aligned} \qquad \textbf{(3 80)}$$

The contact algorıthm technıque presented ın thıs work requıres the followıng procedures,

1 Descrıtızatıon of the dıe boundary ınto segments, and the coordınate and connectıvıty of these segments should be supplıed to the FEM program

2 For the nodes whıch are on the boundary, on the surface sc, a local coordınate ıs set

and both velocity and traction are transferred from global coordinate to this local coordinate system as,

$$\begin{aligned} V &= T\,V \\ f &= T\,f \end{aligned} \qquad \textbf{(3.81)}$$

where T is the transformation matrix,

$$T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \qquad \textbf{(3 82)}$$

The first step in this algorithm is the determination of the boundary nodes of the workpiece Then, for the nodes which are still free (out of contact with the die) the velocity vectors at the nodal points are determined and the relative velocity V_r is calculated for each of these nodes as,

$$\begin{aligned} V_{rx} &= V_{px} - V_{Diex} \\ V_{ry} &= V_{py} - V_{Diey} \end{aligned} \qquad \textbf{(3 83)}$$

where V_{px},V_{py} are the velocity components of the node P, and V_{Diex},V_{Diey} are the velocity components of the die at this point

Next, the algorithm checks each of these relative velocity vectors to find out whether any of these points through any of the segments When a case is encountered where a particular velocity vector points through a die segment as shown in Fig 3 4, the distance D_n of the free node from the die segment is calculated as,

$$D_n = \frac{\overline{I_1P}\;\bar{n}}{|n|} \qquad (3\ 84)$$

where $I_1P=[(x_p-x_1),(y_p-y_1)]$ and $\bar{n}$ is the unit vector in the normal direction to the die segment

This task is performed automatically by the program for all free nodes on the boundary

The time necessary for a node to come into contact with the die is obtained from the minimum time increment DT found for all segments,

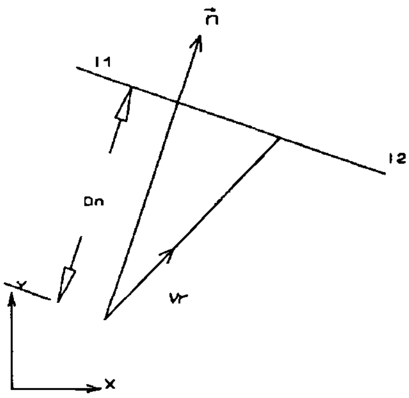


Fig 3 4 Scheme to calculate the minimum time increment

$$DT_{min} = \frac{D_n}{V_n} \qquad (3\ 85)$$

Now, if the minimum time increment for a particular node is less than the maximum step time increment $DT_{min} \leq DT_{max}$ this node will be selected to be attached to the die during the next step which has to be updated using Dt_{min} If more than one node has been selected to come into contact with the die, the geometry will be updated using the maximum value among the minimum time increment values

The boundary condition for the new contact nodes is modified in such a way that the movements along the normal direction of the die surface are zero The contact nodes are forced to move on a tangential direction on the die surface under the friction condition Some nodes may slide along the die moving from one segment to another, such a situation has been taken care of by numbering the die segments as elements and keeping track of each node by changing its parameters when moving from one segments to another

3.10 REZONING IN METAL FORMING

In practical forging processes, deformation is usually very large It is not uncommon to encounter effective strain values of two or more Moreover, the relative motion between the die surface and the deforming material is also large Such large deformation and displacements, encountered in forming processes, cause certain computational problems during the FEM simulation These problems are

1 Difficulties in incorporating the die boundary shape into the FEM mesh, with increasing relative displacement between the die and the workpiece

2 Difficulties in accommodating the considerable change of deformation mode with one mesh system

3 Formation of an acceptable element shape with negative Jacobian due to large local deformation

In order to overcome the above difficulties it is necessary to redefine a new mesh system (Rezoning) Among the various methods [121,122], tested and used for rezoning, it appears that the Area-Weighted Average method is the most convenient and provides sufficient accuracy for remeshing in metal forming simulations

In order to overcome the difficulties resulting from the large deformation encountered in metal forming, it is necessary to redefine the mesh system The rezoning consists of two procedures,

1 The assignment of a new mesh system to the workpiece using the same mesh generation program which has been used to generate the initial mesh

2 The transformation of the field variables from the old to the new mesh through interpolation

In general, temperatures are given at nodal points in Finite Element Programs, thus, its distribution is expressed by using element shape functions over the whole workpiece Interpolation from the old mesh to the new one is done simply by evaluating the temperatures at the new node locations

Interpolation of effective strain are given at the reduced integration point of each element Therefore, before interpolation it is necessary to obtain the effective strain values at the regular interpolation points

In this study the Area-Weighted Average method has been adopted [120] The nodal value is determined on the basis of the average of the adjacent element values weighted by the associated element size Fig 3 5 shows node N surrounded by adjacent elements The nodal value of the effective strain at node N can be written by,

$$\bar{\varepsilon}_N = \frac{\sum_J \bar{\varepsilon}_J \quad A_{JN}}{A_{JN}} \qquad \mathbf{(3\ 86)}$$

where ε_J is the effective strain value at the centre of element j A_{JN} is the area contribution of the jth element to node N and is defined by,

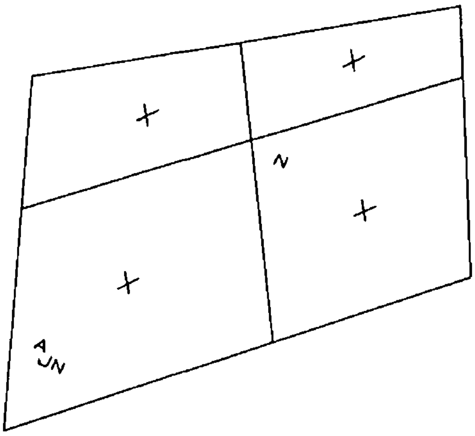


Fig 3 5 Node N surrounded by adjacent elements for Area-weighted average

$$A_{JN} = \int_{AJ} q_N(x,y)\, dA \qquad (3\ 87)$$

where q_{JN} is the element shape function of element j at node N

Once the effective strains are determined at all nodes, the strain distribution over each element can be defined by,

$$\bar{\varepsilon}(x,y) = \sum_{\alpha} q_{\alpha}\, \bar{\varepsilon}_{\alpha} \qquad (3\ 88)$$

where q_{α} is the element shape function

Fig 3 6 shows a schematic diagram of the rezoning algorithm

To find out the nodes from the new mesh which are located within each element of the old mesh, the following procedure has been carried out

- For the isoparametric elements, the transformation matrix of the coordinate obtained by,

$$X = \sum_{i} q_i(\xi,\eta)\, x_i \qquad (3\ 89)$$

$$Y = \sum_{i} q_i(\xi,\eta)\, y_i \qquad (3\ 90)$$

where (x_i,y_i) are the coordinates of the element nodes in the global coordinate system,

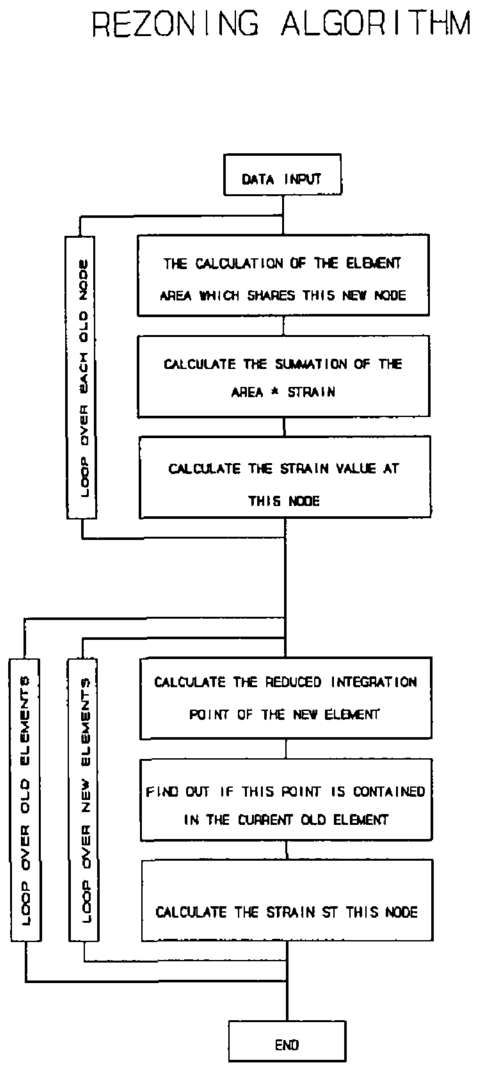

REZONING ALGORITHM
DATA INPUT
LOOP OVER EACH OLD NODE
THE CALCULATION OF THE ELEMENT AREA WHICH SHARES THIS NEW NODE
CALCULATE THE SUMMATION OF THE AREA * STRAIN
CALCULATE THE STRAIN VALUE AT THIS NODE
LOOP OVER OLD ELEMENTS
LOOP OVER NEW ELEMENTS
CALCULATE THE REDUCED INTEGRATION POINT OF THE NEW ELEMENT
FIND OUT IF THIS POINT IS CONTAINED IN THE CURRENT OLD ELEMENT
CALCULATE THE STRAIN ST THIS NODE
END

Fig 3 6 Rezoning algorithm

and i=1,4 for four nodes linear element $q_i(\xi,\eta)$ are the shape function of the element at the nodes as follow,

$$
\begin{aligned}
q_1 &= \frac{1}{4}(1-\eta-\xi+\xi\eta) \\
q_2 &= \frac{1}{4}(1-\eta+\xi-\xi\eta) \\
q_3 &= \frac{1}{4}(1+\eta+\xi+\xi\eta) \\
q_4 &= \frac{1}{4}(1+\eta-\xi-\xi\eta)
\end{aligned}
\qquad (3\ 91)
$$

- Fig 3 7 shows an element from the distorted mesh and P(x,y) is a point from the new mesh

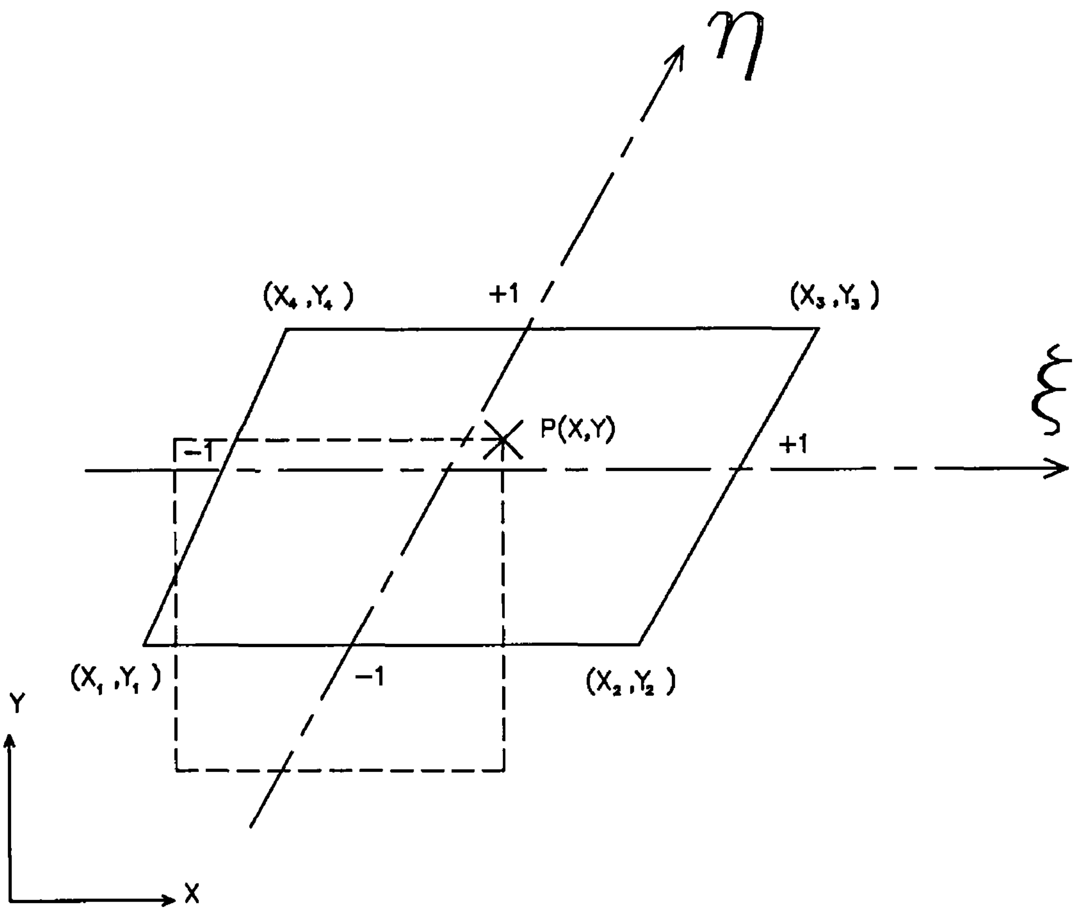


Fig 3 7 The new and the distorted elements

Substituting Eq (3 91) in Eq (3 89) and Eq (3 90),

$$
\begin{aligned}
X &= A1 + A2\,\eta + A3\,\xi + A4\,\xi\eta \\
Y &= B1 + B2\,\eta + B3\,\xi + B4\,\xi\eta
\end{aligned}
\tag{3.92}
$$

where,

$$
\begin{aligned}
A1 &= \frac{1}{4}(\ x1 + x2 + x3 + x4\ ) \\
A2 &= \frac{1}{4}(-x1 - x2 + x3 + x4\ ) \\
A3 &= \frac{1}{4}(-x1 + x2 + x3 - x4\ ) \\
A4 &= \frac{1}{4}(\ x1 - x2 + x3 - x4\ )
\end{aligned}
\tag{3 93}
$$

and

$$\begin{aligned} B1 &= \frac{1}{4}(y1 +y2 +y3 +y4) \\ B2 &= \frac{1}{4}(-y1 -y2 +y3 +y4) \\ B3 &= \frac{1}{4}(-y1 +y2 +y3 -y4) \\ B4 &= \frac{1}{4}(y1 -y2 +y3 -y4) \end{aligned} \qquad \textbf{(3.94)}$$

Groupıng terms of Eq (3 92) ın power of(η) yıelds,

$$\begin{aligned} X &= (A2+A4\ \xi)\ \eta\ +\ (A1+A3\ \xi) \\ Y &= (B2+B4\ \xi)\ \eta\ +\ (B1+B3\ \xi) \end{aligned} \qquad \textbf{(3 95)}$$

Treatıng (ξ) as a constant and consıdenng the monomıal (η) ın Eq (3 95) as the ındependent vanable,

$$\begin{aligned} (A2+A4\ \xi)\ \eta+(A1+A3\ \xi-X) &= 0 \\ (B2+B4\ \xi)\ \eta+(B1+B3\ \xi-Y) &= 0 \end{aligned} \qquad \textbf{(3 96)}$$

$$\begin{aligned} a_2(\xi)\eta+a_1(\xi) &= 0 \\ b_2(\xi)\eta+b_1(\xi) &= 0 \end{aligned} \qquad \textbf{(3 97)}$$

Fırst of all, ξ ıs obtaıned whıch should greater than -1 and less than +1 Usıng thıs condıtıon, η ıs calculated and then by usıng the local coordınate ıt ıs checked whether thıs node ıs contaıned wıthın thıs element or not

Owıng to the specıfıc structure of the mesh generatıon program (4-node elements), the followıng error functıon ıs adopted,

$$Error \quad = \quad \frac{D}{d} \qquad \textbf{(3.98)}$$

where D and d are the large and small dıagonal length respectıvely

Thıs error functıon measures how much the element dıffers from a rectangle

CHAPTER FOUR

IMPLEMENTATION OF THE RIGID PLASTIC FORMULATION

4 1 INTRODUCTION

This chapter is concerned with coding the finite element program for metal forming simulation based on the rigid plastic formulation explained in the previous chapter The development of this code is based on the initial work carried out by Kobayashi et al [120] It is necessary to explain how the equations have been used within the code and what type of approximation has been adopted for the study of practical metal forming operation The process of performing the finite element simulation of metal forming operation is divided into four main parts as shown in Fig 4 1

4 2 DESCRIPTION OF METAL FORMING OPERATION

The starting point in any such analysis is the metal forming operation itself The first stage in the analysis is ,therefore , concerned with obtaining a complete description of the operation in geometrical or numerical form This description will include information about the initial geometry of the workpiece, the shape of the dies and how the relative position and orientation of the dies and workpiece change during deformation, the previous history of the workpiece and the dies, and the particular metal being formed Most of this data and information is obtained from the current CAD system The geometrical designs of the billet and the die are obtained using the facilities built within the current CAD system

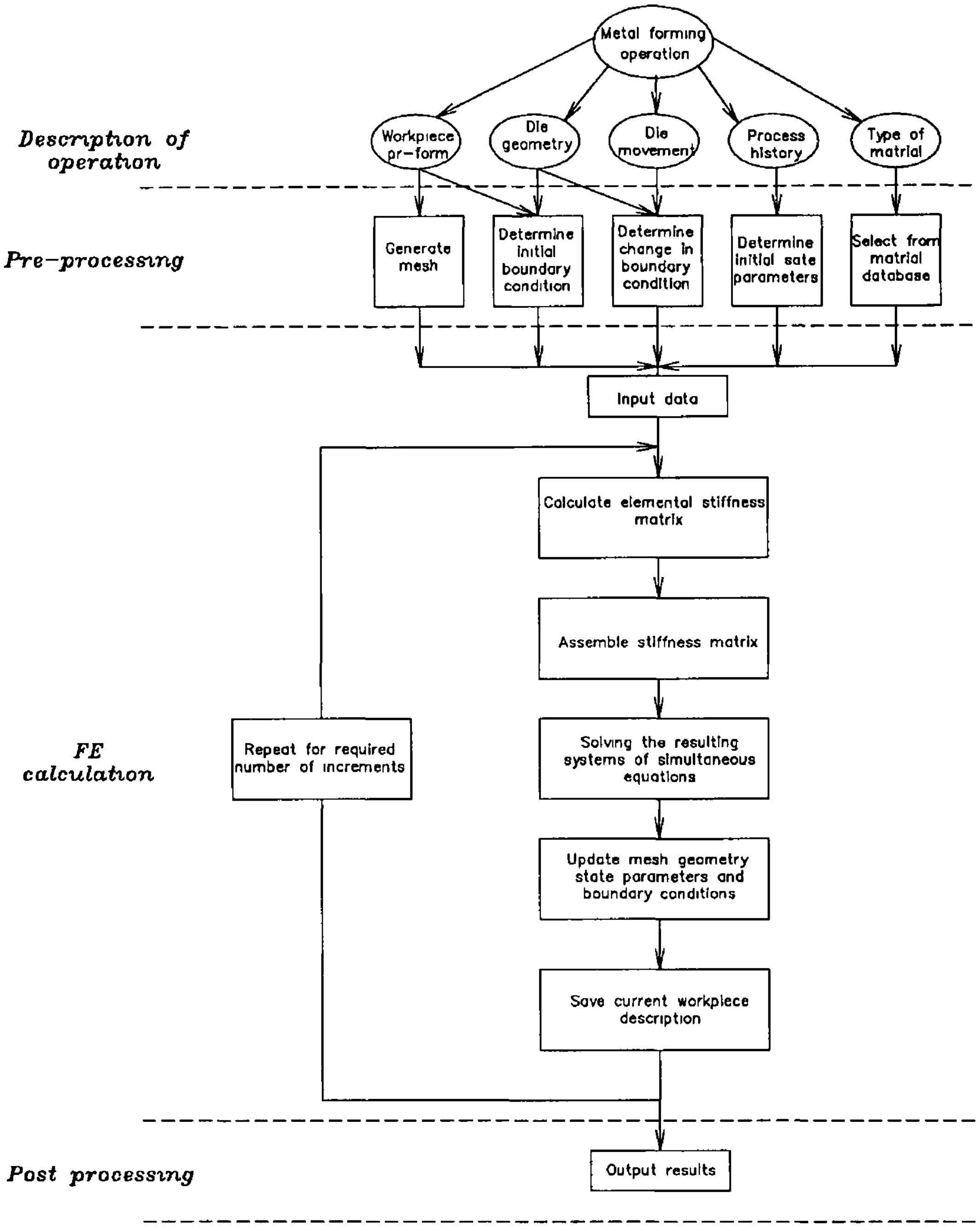

Metal forming operation
Description of operation
Workpiece pr-form
Die geometry
Die movement
Process history
Type of matrial
Pre-processing
Generate mesh
Determine initial boundary condition
Determine change in boundary condition
Determine initial sate parameters
Select from matrial database
Input data
Calculate elemental stiffness matrix
Assemble stiffness matrix
FE calculation
Repeat for required number of increments
Solving the resulting systems of simultaneous equations
Update mesh geometry state parameters and boundary conditions
Save current workpiece description
Post processing
Output results

Fig. 4 1 Schematic representation of the FE analysis process

4 3 PRE-PROCESSING

The pre-processing procedures make use of the description of the metal forming operation and changes them to numerical data to be used as input to the finite element program

4.3.1 THE MESH GENERATION PROGRAM

A mesh system is usually generated on the billet's domain by dividing it into a number of elements, joined together at nodes The mesh is then defined to the finite element program by specifying the nodal coordinates and the connectivity of each element The mesh can be generated manually, and the numerical information obtained can be typed into the computer, but even for a simple mesh this is very time consuming The alternative is either to write a program to generate the mesh or to use a commercial mesh generation package, if a suitable one is available

The disadvantage of the latter option is that it may take a long time to gain hands-on experience in using a commercial mesh generation package properly, besides, such programs are very expensive In addition, it is difficult to integrate most commercial packages with the particular CAD system chosen Because of these disadvantages it was necessary to develop a program that can provide good results without occupying much computer memory The scheme used in this work is designed for maximum flexibility and interactivity The program is capable of generating meshes of linear 4-node elements To achieve the accuracy and interactivity, The AutoCAD, drafting software, has been used as a pre- and post-processor of the mesh data

The AutoLISP language has been used to retrieve the geometric data from the screen and to pass it to the mesh generation program as explained in chapter three The mesh generation program is written in FORTRAN-77 The output of the program is prepared in two forms, a data file with numerical values to be used as input to the finite element program and graphic data in a DXF format to be utilized by the AutoCAD to plot the mesh on the screen

The characteristics of this scheme are,

1 An adequate boundary description, because the original geometry of the component is generated using a CAD system and the data are retrieved from the database of the drawing

2 It has the capabilities for describing zones of different materials

3 A facility for grading the mesh to achieve the required accuracy of idealization

4 A renumbering system to minimize the half bandwidth which results in better computational efficiency

5 An adequate post-processing by visualizing the mesh system with its details (node numbering, element numbering)

6 Node and element numbering are plotted on the drawing proportionally to the corresponding nodes and elements and in different layers, so that the user has the option of using CAD capabilities (Zoom, Pan, Layer on/off)

4 3 2 BOUNDARY CONDITIONS

Constraining conditions have been introduced to determine which of the two Cartesian components of velocities are to be unconstrained, and which are to have some specified value It is often found that a number of nodes are subject to the same constraining condition It was therefore convenient to define each constraining condition once only, and then to specify the constraint-condition number

Nodal constraining conditions apply throughout the deformation, but the boundary conditions resulting from contact between the workpiece and the dies will change as the metal forming operation proceeds To determine the boundary conditions at any part outside the mesh at any stage, it is necessary to determine which nodes are in contact with the dies The shape and position of the dies must therefore be made known to the finite element program

The method adopted in this work is to model each die by discretizing the boundary of the die into segments In this method, the determination of the nodal contact is much simplified and the accuracy is increased by increasing the number of segments It is to be noted here that increasing the number of segments to a certain number will increase the computational time of the solution, so a compiomise has to be made in deciding the number of segments

Nodes which are in contact with the die are forced to be in contact and the only direction these nodes permitted to move is sliding along the tangential direction of the die surface This movement is controlled by the friction condition between the material and die surface In the normal direction of the die surface, the nodes in contact have the value of the die velocity New nodes come into contact with the die as the deformation process continues

In general, it is not known beforehand exactly how a particular node will move during the metal forming operation What is known is how the dies move So in addition to specifying the shape of the die surfaces, it is necessary to provide information to the finite element program about how these positions change during the metal forming operation The method used here is to use the relative velocity between the workpiece and the dies By comparing the time necessary for a particular node to come in contact with the die, with the maximum time increment allowed for each step, it is decided whether this node should come into contact or not

The frictional stress, in geneial, changes its direction at the "neutral point", but the location of this point is not previously known The solution for the "neutral point" problem has been outlined in the previous chapter

Subroutine CONTACT is developed to fulfil the contact process Fig 4 2 shows a schematic diagram of the contact algorithm The first step in this subroutine is a loop over all the boundary nodes which are still fiee

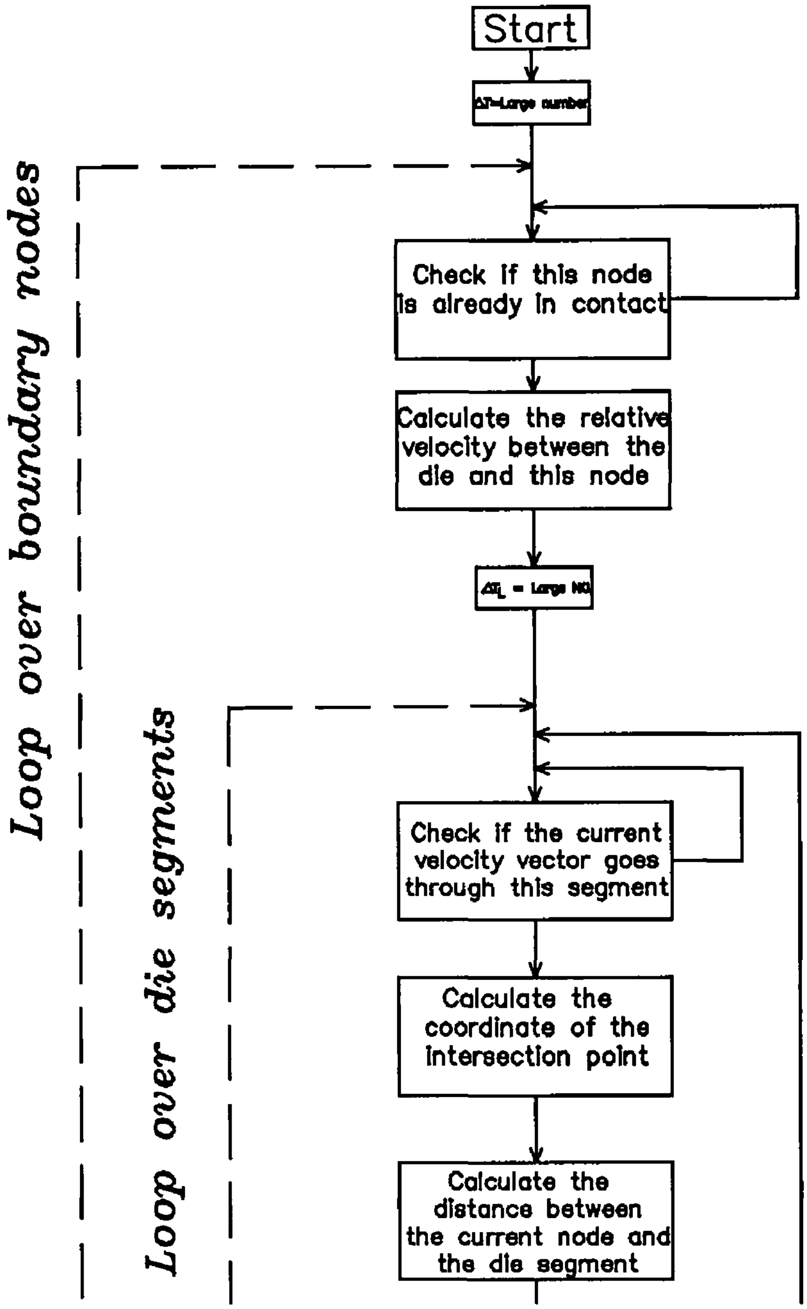

Start
ΔT=Large number
Check if this node is already in contact
Calculate the relative velocity between the die and this node
ΔT_L = Large NO
Check if the current velocity vector goes through this segment
Calculate the coordinate of the intersection point
Calculate the distance between the current node and the die segment
Loop over boundary nodes
Loop over die segments

Fig 4 2 *(continue)*

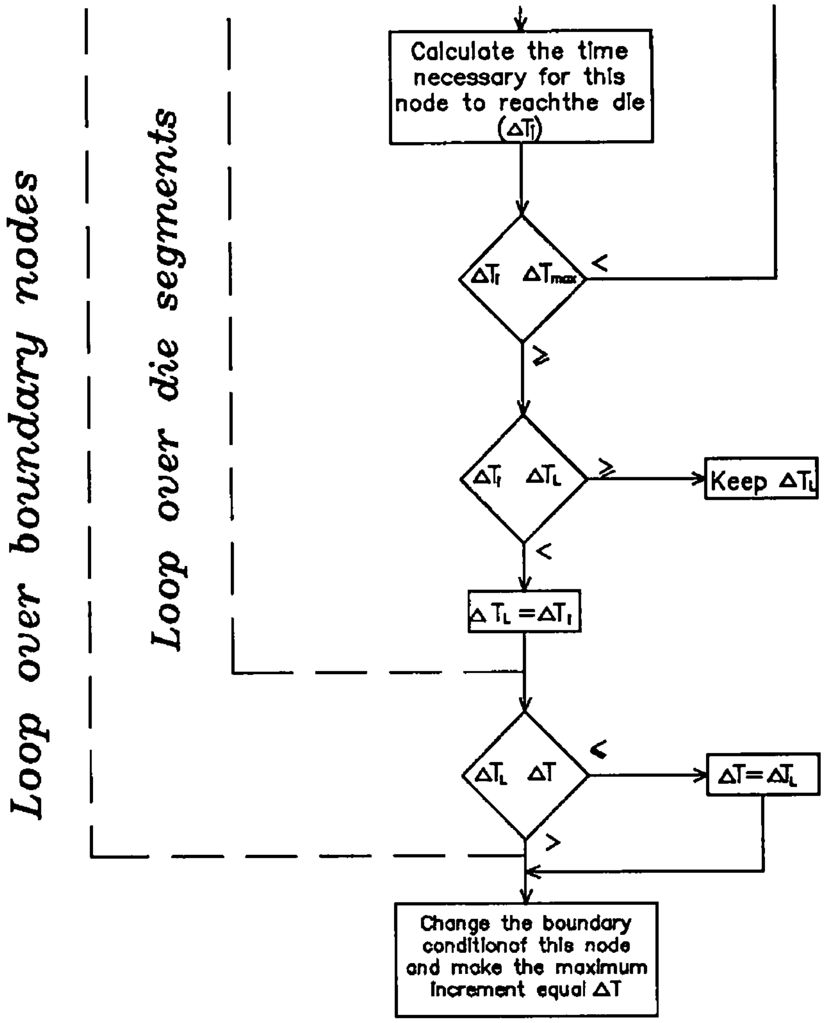

Loop over boundary nodes
Loop over die segments
Calculate the time necessary for this node to reachthe die (ΔTi)
ΔTi ΔTmax
<
≥
ΔTi ΔTL
≥
Keep ΔTL
<
ΔTL =ΔTi
ΔTL ΔT
<
ΔT=ΔTL
>
Change the boundary conditionof this node and make the maximum increment equal ΔT

Fig. 4 2 A schematic diagram of the contact algorithm

The first step m this loop is the determmation of the velocity vectors of nodal points and subsequently calculating the relative velocity V_r for each of these nodes as,

$$V_{rx} = V_{px} - V_{Di\,ex} \qquad (4.1)$$
$$V_{iy} = V_{py} - V_{Di\,ey}$$

where V_{px},V_{py} are the velocity components of the node P, and V_{Diex},V_{Diey} are the velocity components of the die at this point

Next, the algonthm checks each of this relative velocity vectors to find out whether any of these points through any of the die segments When a case is encountered where a particular velocity vectoi points through a die segment the distance D_n of the free node from the die segment is calculated as,

$$D_n = \frac{\overline{I_1P}\,\bar{n}}{|n|} \tag{4.2}$$

where $I_1P=[(x_p-x_1),(y_p-y_1)]$ and n is the unit vector in the normal direction to the die segment This task is performed automatically by the program for all free nodes on the boundary

The time necessary for a node to come into contact with the die is obtained from the minimum time increment DT found for all segments,

$$DT_{\min} = \frac{D_n}{V_n} \tag{4.3}$$

Now, if the minimum time increment for a particular node is less than the maximum step time increment $DT_{\min} \leq DT_{\max}$ this node will be selected to be enforced to the die during the next step which has to be updated using $Dt_{\min}$ If more than one node have been selected to come into contact with the die, the geometry will be updated using the maximum value among the minimum time increment values

The boundary condition for the new contact nodes are modified in such a way that the movements along the normal direction of the die surface are zero the contact nodes are enforced to move along a tangential direction on the die surface under the friction condition

Some nodes may slide along the die moving from one segment to another, such a situation has been taken care of by numbering the die segments as elements and keeping track of each node by changing its parameters when moving from one segment to another

4 4 FINITE ELEMENT CALCULATION

4 4 1 INPUT DATA

The pre-processing stage of the finite element analysis produces a file of numerical information The input file has been divided into several main specifications, firstly, the data required to define the geometry of the billet and dies, then the material properties of the material billet and the die velocity Several control parameters have to be provided for controlling the deformation process

The input data to the finite element program are as follow,

TITLE The title of the case

NINI Initial step number

NSEND Final step number

DTMAX Step size in time unit

ALPH Limiting strain rate (Cut off value)

DIAT Penalty constraint

IPLAS An indicator used to identify the type of material to be employed

0 for rigid plastic materials

1 for rigid visco-plastic materials

STK which represent yield stress K in the material's formula $\sigma=K\varepsilon^n$

EXN represents n in the same formula

IPLNAX Problem type parameter,

1 for axisymmetric analysis

2 for plain strain analysis

FRCFAC Friction factor

NUMNP Number of nodal points

RZ(2,NUMNP) This array stores the coordinates of the nodal points

NUMEL Total number of elements in the workpiece

NOD(4,NUMEL) Element's connectivities

NBNODE Number of the boundary nodes which are in contact with the dies at the initial stage

NBCD(2,NBNODE) The boundary condition codes

0 Nodal force is specified

1 Nodal velocity is specified

3 Node in contact with die

LNBC(2, NBNODE) The local boundary condition codes

0 Nodal force is specified

1 Nodal velocity is specified

3 Node in contact with die

NVNODE Number of nodes which are under external velocity at the initial stage

URZ(2,NVNODE) Nodal velocity components

TEPS(NUMEL) The effective strain

NDIE Number of die nodes which construct the die segments

DCOORD(2,NDIE) This array contains the coordinate of the die nodes

4 4 2 ASSEMBLY OF THE STIFFNESS EQUATIONS

The elemental stiffness matrices are evaluated from Eqs 3 68 and 3 72 Assembling them for the whole workpiece, we obtain a set of simultaneous equations,

$$K \quad \Delta V = f \qquad (4.4)$$

The main subroutine in the finite element program is NONLIN, the function of this subroutine is to control the iteration process The global stiffness matrix is constructed within this subroutine and solved iteratively until the solution is reached Fig 4 3 shows a diagram for subroutine NONLIN

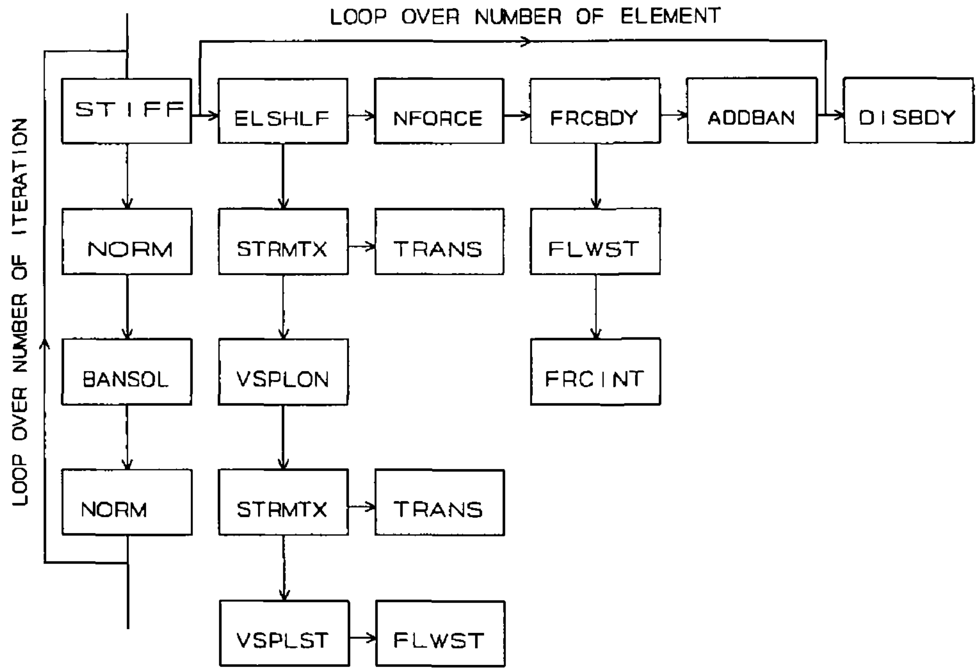


Fig 4 3 Flow chart of NONLIN subroutine

It is clear from this diagram that there are two main loops The first loop works over the number of elements and is contained within the other loop In this loop the elemental stiffness matrix is constructed for each element and is added to the global stiffness matrix At the end of this loop the global stiffness matrix is completed The other loop is an iterative loop in which the direct and Newton Raphson iterations are applied There are limits for the number of iterations, usually the calculations are terminated when these limits are encountered For direct iteration the limit is 200 iterations and 20 iterations for Newton Raphson method

4 4 2 1 STIFF subroutine

The role of this subroutine is to generate the stiffness matrix of the geometry It starts by evaluating the elemental stiffness matrix, then it adds the nodal point forces After that it adds the contribution of the friction to the elemental stiffness matrix Finally at the end of the elemental loop it assembles the global matrix and applies the displacement boundary conditions

Subroutine ELSHLF calls most of the subroutines which calculate the stiffness matrix at the elemental level

4 4 2 2 STRMTX subroutine

The function of this subroutine is to evaluate the strain-rate matrix of quadrilateral elements This subroutine uses the coordinate of the nodes to carry out this calculation using Eq 3 32, where it calculates the terms of this matrix using Eq 3 38, Eq 3 39 and Eq 3 40 After the calculation of the strain-rate matrix B, it has to be multiplied by the transformation matrix using subroutine TRANS

4 4 2 3 TRANS subroutine

The elemental strain matrix B(4,8) is calculated in subroutine STRMTX, where the number of columns in B is determined by the number of degree of freedom of the the element In TRANS all the nodes of the element will be checked and the transformation matrix will be defined by Eq 3 79 for those nodes which are in contact with the dies Then the new matrix B will be constructed by multiplying the old one with the transformation matrix

4 4 2 4 VSPLON subroutine

This subroutine is developed to carry out the reduced integration point of the volumetric strain-rate Two terms of the applied equations are related to the volumetric strain-rate, the first one is in the first derivative Eq 3 70 and the other is the last term of the second derivative Eq 3 72 This subroutine starts by calculating the strain-rate component by multiplying the strain-rate matrix with the velocity vector

$$\varepsilon = B \quad V \qquad (4.5)$$

Then it calculates the volumetric strain-rate using Eq 3 45 Finally, the two terms are calculated and the numerical integration is carried out on one point and the contribution of these two terms are added to the stiffness matrix

4.4.2 5 VSPLST subroutine

This subroutine calculates the rest of the terms of both derivatives in Eq 3 68 and Eq 3 72 The integration is carried out using four integration points First of all the strain rate matrix is calculated then the effective strain rate is calculated using Eq 3 42 where the diagonal matrix D has 2/3 and 1/3 components, corresponding to the normal strain-rate and engineering shear-strain rate respectively Subroutine FLOW is called within this subroutine to calculate the effective stress and the first derivative of the effective stress over the strain-rate using the material formula The value of the calculated effective strain rate is used for calculating the effective stress if the material is strain-rate dependent According to the type of iteration some or all terms in both derivatives are calculated At the end of this subroutine the contribution to the stiffness matrix and force matrix is calculated and added

This stage is the end of ELSHELF subroutine which mark the end of the elemental stiffness matrix calculation

4 4.2.6 NFORCE subroutine

This subroutine is developed to add the nodal point forces to be used in later stages for the evaluation of the friction condition on the interface surface between the workpiece and the material This subroutine is accessed after returning from subroutine ELSHELF back to STIFF

4.4 2.7 FRCBDY subroutine

The function of this subroutine is to check all the element sides and when an element side which is in contact with the die is encountered the friction is calculated along this side When such a case is encountered, first of all the FLOW subroutine is called to calculate the flow stress and then the FRCINT subroutine is called to calculate the friction contribution to the stiffness matrix of the current element This subroutine works within the loop over the number of element as has been mentioned before

4 4 2 8 FRCINT subroutine

As has been explained in the previous chapter, the frictional stress is approximated by the arctangent function of the relative sliding velocity which eliminates the sudden change of direction of the frictional stress Then the two derivatives of this function are found as in Eq 3 61 and Eq 3 62 So the function of this subroutine is to include the calculation of these two derivatives to the stiffness equation First of all the die velocity of the segment, with which the element side is in contact, is transformed to the local coordinate system because the velocity of nodes of the element side which is in contact is already in local coordinate system Then the relative velocity in the tangential direction is then calculated by Eq 3 80 and used in Eq 3 61 and Eq 3 62 Simpson's [120] formulation is used to do the integration of these two derivatives in one dimension The result is then multiplied by the thickness if the case is plane strain or by 2π if the case is axisymmetric Finally the contributions to the stiffness matrix and force matrix are added This subroutine represents the final stage of building the elemental stiffness equation

4 4 2 9 ADDBAN subroutine

This subroutine represents the last step in the loop over the number of elements After the construction of the elemental stiffness matrix for each element , it has to be added to a global stiffness matrix which is completed by the end of the loop over the elements

4 4 2 10 DISBDY subroutine

This subroutine is developed to apply the displacement boundary condition In the finite element discretization, the velocity boundary condition is enforced only at the nodes which are in contact with the dies or the nodes which represent the symmetry lines The velocity along the element side is determined automatically in terms of the velocities of nodes and the element shape functions For the node at which the velocity is defined, the velocity correction ΔV_m is zero Consequently, the corresponding stiffness equation should be removed

The simplest way to implement this procedure [123] is to replace the corresponding rows, and columns by zero and set the diagonal term to 1 This procedure is adopted in this subroutine

4 4 3 SOLUTION OF THE STIFFNESS EQUATIONS

The solution of the system,

$$K \quad \Delta V = F \tag{4.6}$$

is a most impoitant step in the total finite element method of solution The unknown number n is directly proportional to the number of nodes and also to the number of degree of freedom per node The accuracy and range of application of the method is limited only by the number of simultaneous linear equations that can be solved economically using presently available computers

Methods of solutions are generally divided into two broad classes [123],

a Direct methods, also called Gaussian elimination methods

b Iterative methods of which the Gauss-Seidel variation is the most popular

The method used in this work is based on the Gaussian elimination method In this solution the stiffness matrix is stored in a banded matrix form and the Gaussian eliminations applied over the maximum band width

4 5 POST-PROCESSING

Most commercial finite element packages have got their own graphic capabilities for displaying the output result These post processors are sometimes built within the finite element program or as stand alone softwares However, in most cases there is a lack of flexibility and capabilities in these programs and sometimes high consumption of the memory For these reasons it was decided to develop a system where most of the facilities needed by users are available In this work the post processor is partially built inside the finite element program The results of each step of the solution is written in two files The first one is written in a DXF format which is used by the current CAD system in particular and most CAD systems in general DXF is an ASCII drawing interchange file which accepts all types of entities used by CAD systems such as lines, arcs,polylines, blocks and text Also, this format contains the properties of entities such as colour and line type Furthermore, it is possible to create layers and place entities in different layers All these facilities have been incorporated in writing these graphic files When loading these files the simulation results are plotted on the screen in different layers as follow,

1 Die geometry

2 Element numbering

3 Node numbering

4 The deformed mesh

5 Contours of effective strain

6 Contours of strain-rate

7 Contours of effective stress

8 Force vectors

9 Velocity vectors

The information is saved separately in different layers where it is possible to display any combination or individual type of the results

The characteristics of this postprocessor are summarized as follow,

- It is possible to display any combination of layers at the same time, even from different steps of the solutions
- The text size of the node and element numbering is plotted proportionally to the

element sıze, whıch gıves a bettei dısplay when usıng zoom facılıtıes ın regıon wıth fıne mesh

In addıtıon to the DXF file for each step ıncıement another fıle ıs created whıch contaıns the numerıcal results of the step solutıon

The maın subroutıne whıch controls the output ıs PRTSOL as shown ın the Fıg 4 4

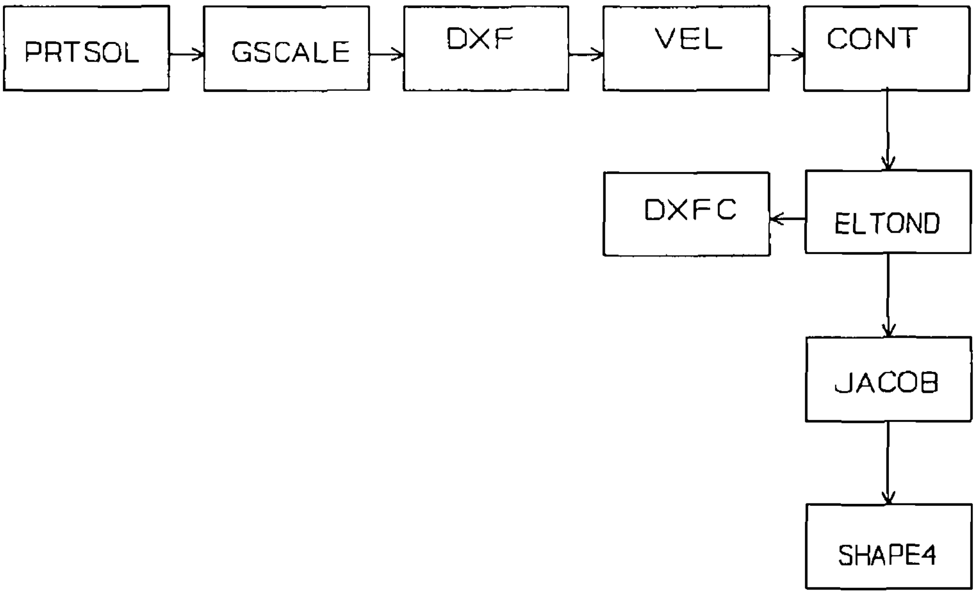


Fıg 4 4 Flow chart of the output subroutınes

CHAPTER FIVE

PLANE STRAIN CLOSED DIE FORGING

5.1 GEOMETRICAL DESIGN OF THE DIE

5 1 1 CONVERSION FROM MACHINED TO FORGED PART CROSS-SECTION

One of the preliminary tasks in forging design procedure is the conversion of the available machined part data into forged part data In the process of conversion, the necessary forging envelope, corner and fillet radii and appropriate draft angles are added to each machined part cross section The conventional conversion of the machined part data into forging data requires a large amount of valuable time

In the present CAD procedure, the process of conversion is largely simplified by making use of the interactivity with the graphic screen This procedure can be applied to a large number of forging sections and the data required to do this conversion have been saved within the computer, so that is available for less experienced users The cross section is obtained from the three dimensional machined part geometry This cross section needs to be modified to conform to process limitations This process involves selection of the parting lines, addition of machining and draft allowances, and fillet and corner radii The selection of these parameters is

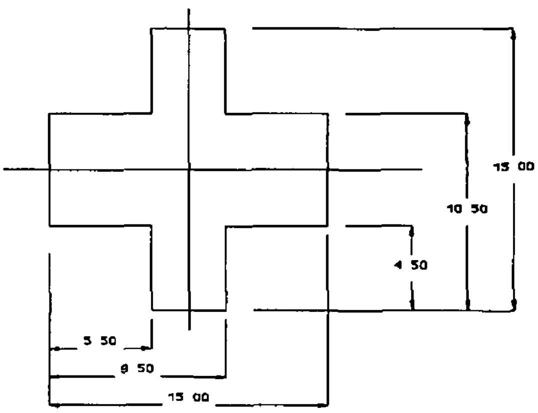


Fig 5 1 Cross-section of the machined part critical for obtaining defect-free forgings A cross section of a machined part is shown in Fig 5 1 All dimensions are in mm and the length of the component is 150 mm The length chosen as 15 times the width to represent plane strain conditions accurately, although it is recomended to be ten times or at least five times [124] The German standard DIN 7523 and the DFRA forging handbook have been adopted as shown in previous chapters and all data have been incorporated within the CAD system as explained in the previous chapters The vertical sides of the cross section inhibit the removal of the finished forging from the die cavity Therefore, all such sides are to be inclined to the vertical, and the angle of inclination is retrieved from the data base This angle is chosen to be 5 degrees The selection of the this angle depends on the forging material, the type of forging equipment and the complexity of the forging Fig 5 2 shows the drawing after adding the draft angle

The next modification to the cross section is the elimination of all sharp corners by adding corner and fillet radii These radii reduce stress concentrations, affect die fill and improve die life The value of the corner radii have been chosen as 1 5 mm and for fillet radii as 2 mm The process of applying these radii is fully interactive and the only thing the user needs to do is to select the two lines that form the corner Fig 5 3 shows the cross section after adding the corner and fillet radii

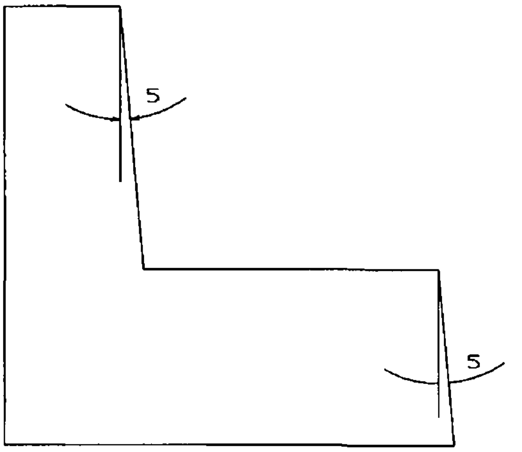


Fig 5 2 Draft angles

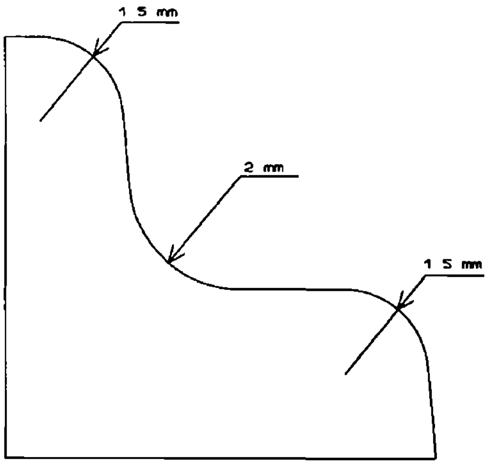


Fig 5 3 Corners and fillets.

5 1 2 FLASH LAND AND GUTTER DESIGN

The flash land and gutter used in dies perform two functions during forging Firstly, the flash land restricts side ways metal flow and thus forces the material to fill die cavities by extrusion Secondly, duiing the final stages of forging when the cavity is filled the flash land allows metal to escape into the flash gutter The calculation method adopted in this work is explained in chapter two Using the flash command from the CAD menu, the user is asked to select the position where the flash has to be located The calculation of the flash land and gutter depends on the mass of the forging Fig 5 4 shows the forging with flash land and gutter

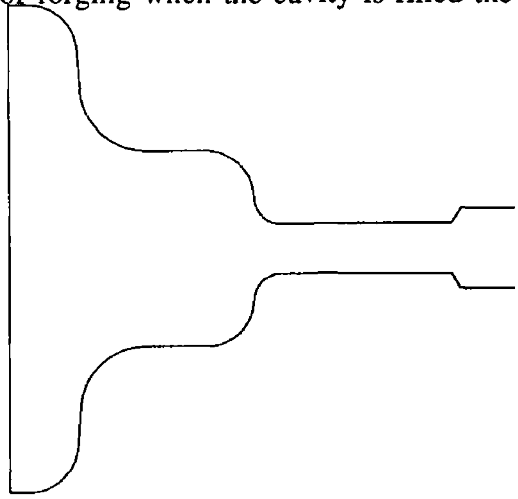


Fig 5 4 Flash and gutter

5 1 3 BILLET CALCULATIONS

Because of the volume constancy, the billet volume should be equal to the forging volume plus the flash land and gutter Using this fact the billet dimensions have been defined as 12 mm in width, 12 5 mm in height and 150 in length

5 2 FINITE ELEMENT SIMULATION

After defining the final shape of the forging it is possible to consider the boundary line of the forging as the boundary for the die cavity A 3D drawing of the forging part is produced using the EXTRUDE command in the CAD system Then the foiging is subtracted from the die block and as a result, the die block with its cavity is produced Finally, a cioss section of the die block is generated to be used in the 2D FE simulation To simulate this foiging piocess it is enough to consider a quaitei of the component because the forging is symmetric along its two centre lines as shown in Fig

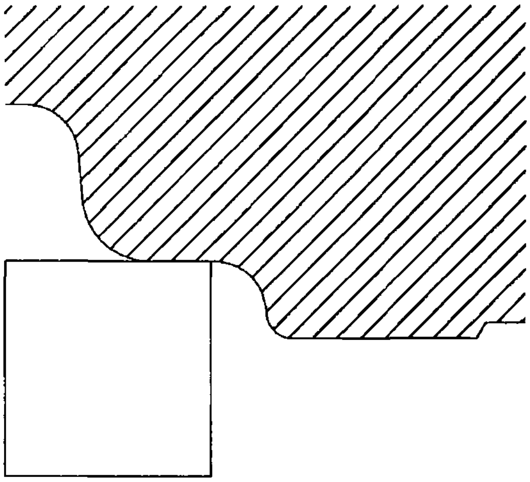


Fig 5 5 The die and the billet

5 5

5.2 1 MESH GENERATION

For this example the billet is drawn and it's domain is divided into seven zones to ensure a fine mesh near the probable contact regions The mesh is created with 97 nodes and 79 elements The bandwidth minimization scheme has been applied and the results are as follow,

Old Bandwidth = 98

New Bandwidth = 34

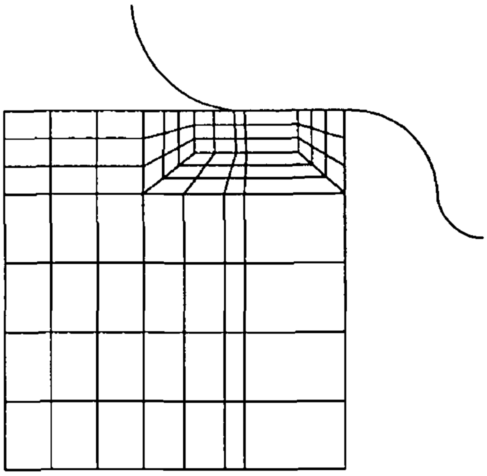


Fig 5 6 The Initial mesh system

Fig 5 6 shows the created mesh system

5 2 2 INPUT DATA FOR FINITE ELEMENT SIMULATION

In addition to the coordinate of the nodes and the element connectivity, which have been produced by the mesh generation program, moie data is still needed The specimen used in this analysis is assumed to be pure lead, which is characterized by a rigid-perfectly plastic (nonwork-hardening) material behaviour with constant flow stress, Y_o=17 236 N/mm^2 The nonsteady state forging process was analyzed in a step-by-step manner with a die displacement at each step equal to 1% of the initial height of the specimen The friction factor has been taken as m=0 035 for the case with lubricant and m=0 3 for dry forging [125] The speed of the machine ram has been taken as V=1 mm/s

5 2 3 FORGING WITH LUBRICANT (m=0 035)

The simulation process has been pioceeded till 17 12% reduction of the initial height when severely distorted elements are encounteied At this point the program automatically stopped at step 20 and the iemeshing procedure started

REMESH 1

A new mesh system is created with 206 nodes and 176 elements A finer mesh is considered on the region of possible contact as shown in Fig 5 7 The bandwidth minimization scheme has been carried out to rearrange the node numbering in order to reduce the computational time needed for the solution The result of bandwidth minimization were as follow,

Old Bandwidth = 132

New Bandwidth = 42

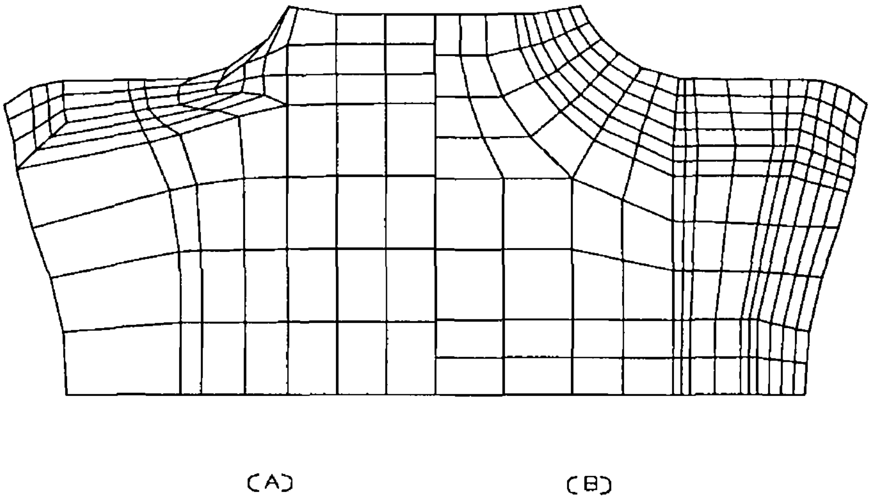


Fig 5 7 Remeshing at 17 12% of the initial height (A) old mesh, (B) new mesh.

REMESH 2

The deformation process has been continued to step 39 ,33 12% reduction of the initial height of the billet, where remeshing is needed to redefine the mesh because of the unacceptable contact between the die and the woikpiece at the die fillet This situation usually takes place when the mesh at the cornei becomes coarse because of the sliding of the nodes on the die surface with different magnitudes of the sliding velocity and sometimes even with different directions When this situation is encountered the material goes into the die partially because the original representation of the curved die is done by approximating it to a number of segments

The new mesh ,Fig 5 8, has 210 nodes and 176 elements A finer mesh is created at the die corner and at the entrance to the flash land to prevent the lack of degrees of freedom The bandwidth minimization scheme resulted in,

Old Bandwidth = 182

New Bandwidth = 46

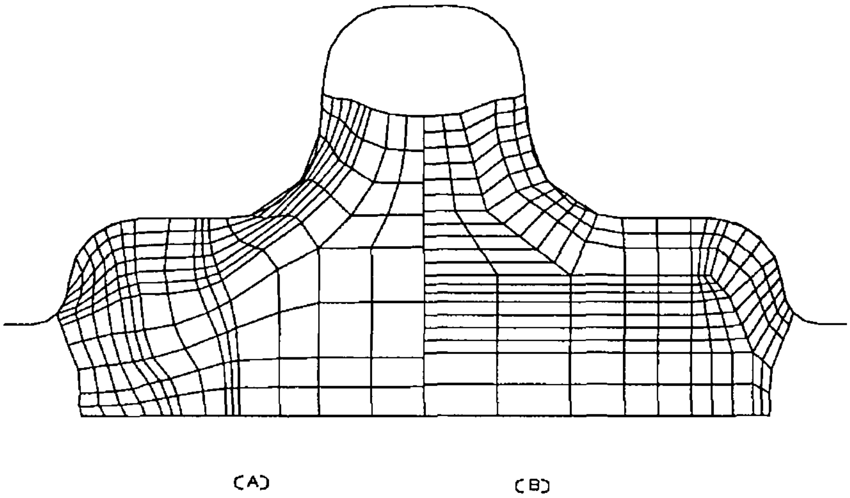


Fig 5 8 Remeshing at 33 12% of the initial height (A) Old mesh, (B) New mesh

As the deformation process continues, unacceptable shaped elements have been encountered again when the material started to flow through the flash land because of the high pressure in this region For this reason and because it is excepted that the material will start to flow to fill the upper cavity, it was necessary to perform the remeshing again

REMESH 3

The third remeshing resulted in 216 nodes and 177 elements as shown in Fig 5 9 and the bandwidth minimization results were as follow,

Old Bandwidth = 144

New Bandwidth = 32

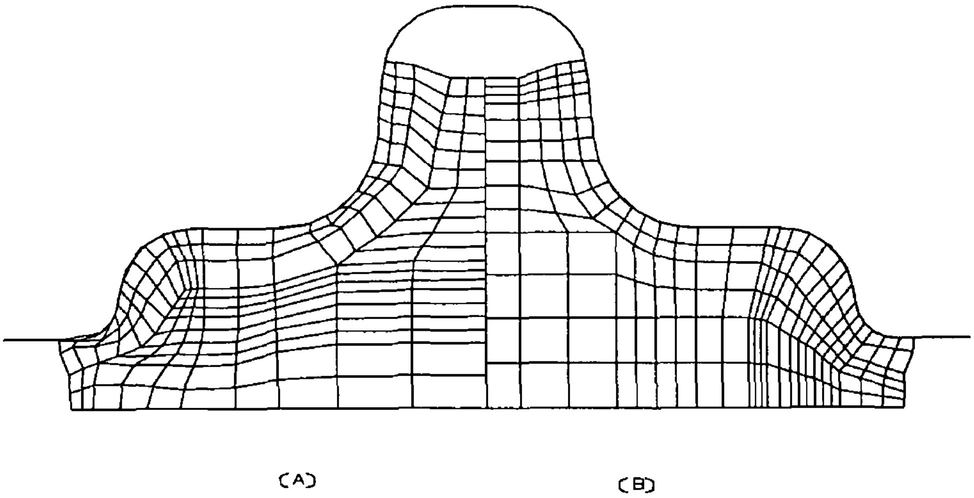


Fig 5 9 Remeshing at 41.6% of the initial height. (A) Old mesh, (B) New mesh

The deformation process has been continued till 41 6% reduction of the initial height of the billet The material flow through the flash which makes it easier for the die to be filled because of the high pressure generated at the flash region At the end of this stage a remeshing for the fourth time was inevitable because of the severely distorted mesh at the flash gate

REMESH 4

The last remeshing resulted in 206 nodes and 169 elements as shown in Fig 5 10 and the bandwidth minimization results were as follow,

Old Bandwidth = 320

New Bandwidth = 38

The deformation process has been concluded at 51 52% reduction of the initial height of the billet and the die is totally filled with the material The simulation process needed 58 steps to fill the die, Fig 5 11

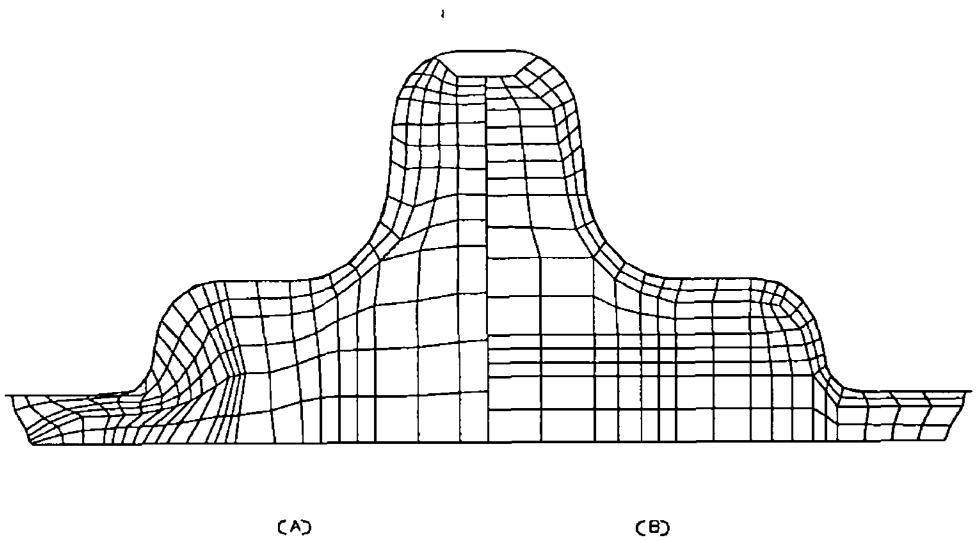


Fig. 5 10 Remeshing at 48 8% reduction of the initial height (A) Old mesh, (B) New mesh

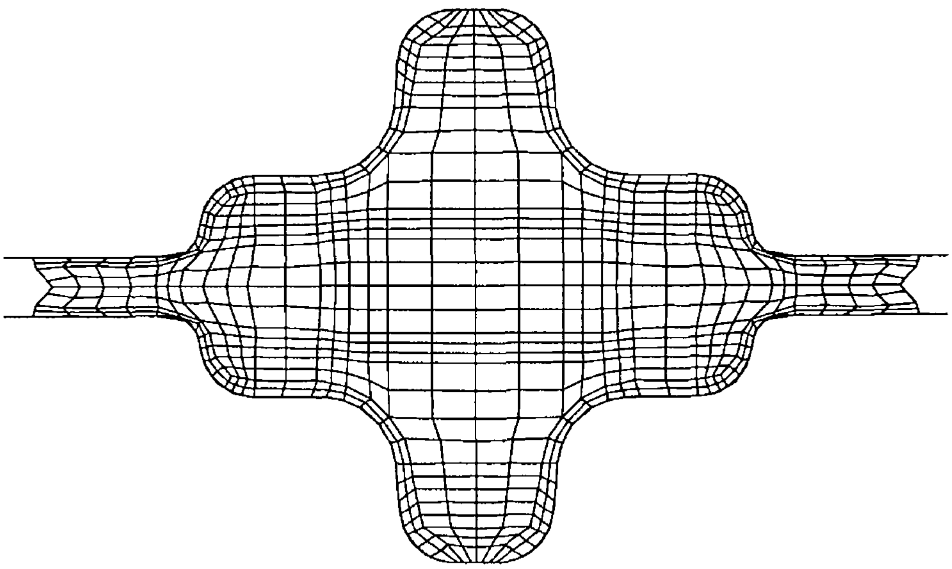


Fig 5 11 The final stage of deformation at 51 52% of the initial height

Effective strain contours have been plotted for the three remeshing stages and for the final stage In Fig 5 12, the maximum strain is concentrated on the upper side of the billet which is in contact with the die fillet The value of the effective strain decreases from the die surface to the core of the forging The minimum value is found to be in the middle, where the material is still rigid Continuing the deformation till the second remeshing as shown in Fig 5 13, it is clear that the effective strain increased extensively throughout the workpiece and specially at the flash gate At the third stage where the material started to flow vertically through the flash land, high strain is exhibited at the flash region which makes the material to flow through the orifice as shown in Fig 5 14 Fig 5 15 shows the contours of effective strain on the last stage where the die is filled with the material

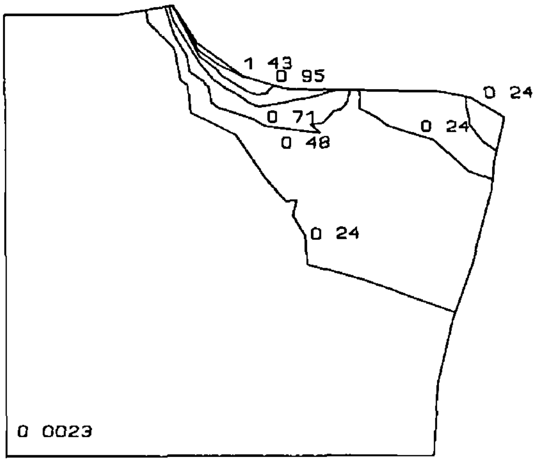


Fig 5 12 Effective strain contours at 17.12% reduction

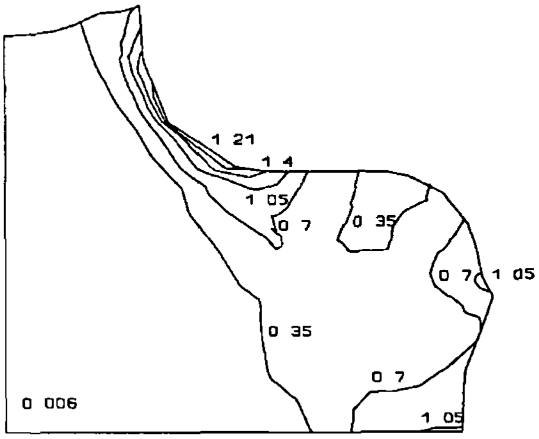


Fig 5 13 Effective strain contours at 33 12% reduction

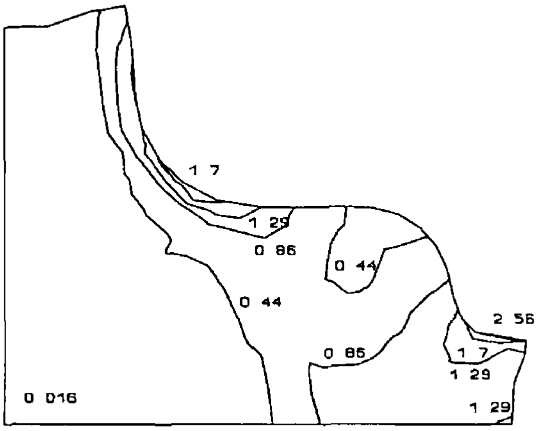


Fig 5 14 Effective strain contours at 41 6% reduction

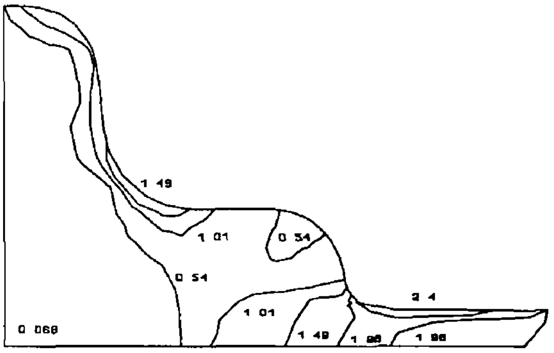


Fig 5 15 Effective strain at the final stage

In addition, the nonuniform flow fields that developed in this case were ascertained by comparison of predicted effective strain-rate fields with observed deformation patterns The effective strain-rate was chosen as a field quantity here, as opposed to strain, because it is the best measure of the instantaneous flow field Fig 5 16-5 19 show the instantaneous values of the effective strain-rate on the four stages

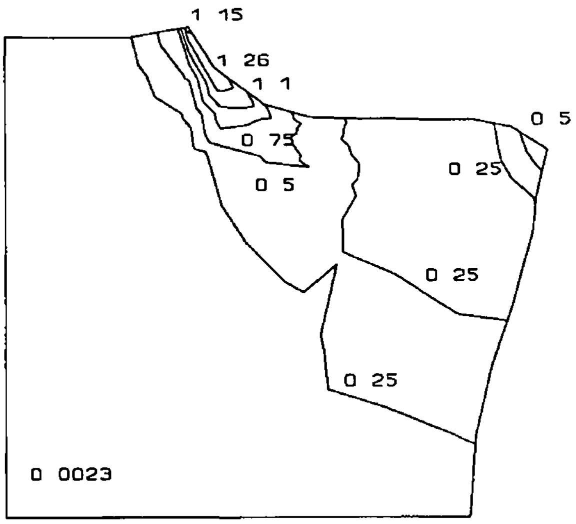


Fig 5 16 Strain rate contours at 17 12% reduction

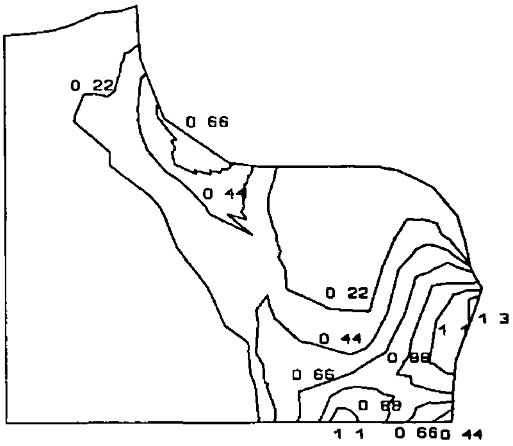


Fig 5 17 Strain rate contours at 33 12% reduction

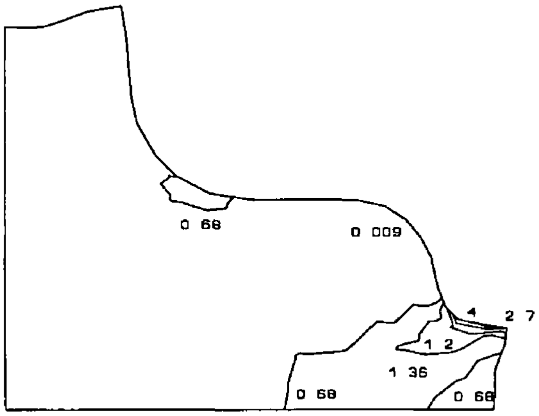


Fig 5 18 Strain rate contours at 41 6% reduction

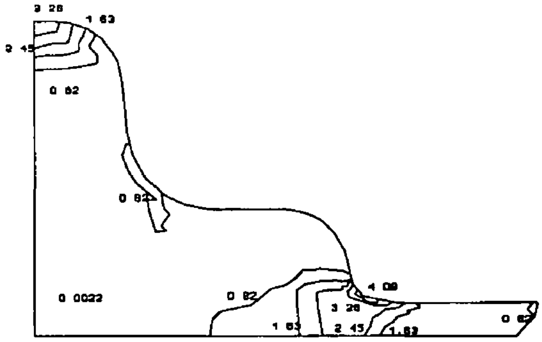


Fig 5 19 Strain rate contours of the final stage

Figs 5 20 to 5 23 show the distribution of the force vectors Two types of forces can be seen, first, the force vectors along the symmetry line which are the equilibrium forces The second type of force is along the contact surface between the material and the die cavity This force is generated as a result of the material movement along the dıe cavıty due to the forgıng load and the tendency to fill the dıe cavıty The load vector of the last stage wıll be used to analyze the dıe block

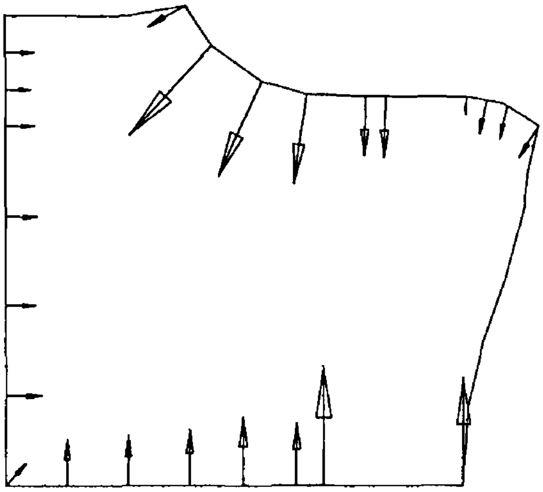


Fıg 5 20 Force vectors at 17 12% reductıon

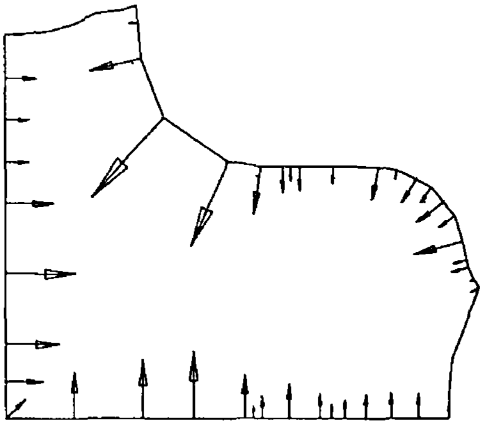


Fıg 5 21 Force vectors at 33 12% reductıon

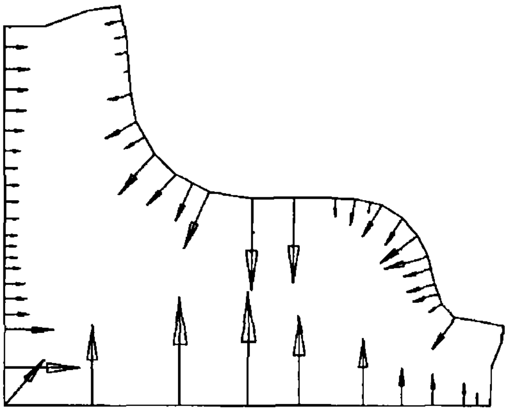


Fıg 5 22 Force vectors at 41 6% reductıon.

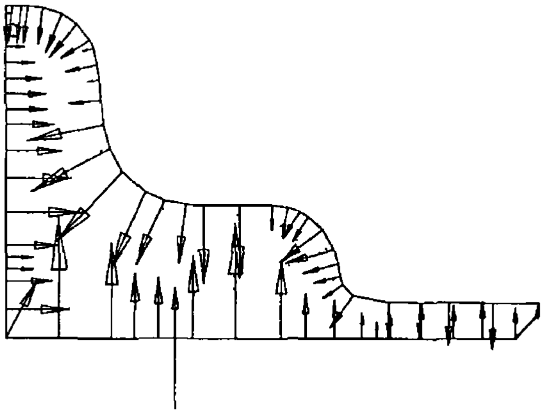


Fıg 5 23 Force vectors for the final stage

The results of the analysıs show three dıstınct modes of flow whıch can be ıdentıfied durıng the course of deformatıon Mode I, Fıg 5 24, ıs very sımılar to sımple upsettıng sınce the upward flow of materıal beneath the orıfice ıs slower ın comparıson wıth the horızontal flow Actually, the relatıve velocıty between the dıe and the materıal at the orıfice gate makes the materıal look as ıf ıt ıs flowıng upward especıally ın the early stages of deformatıon In thıs mode a rıgıd-core regıon around the plane of symmetry ıs found Mode II, Fıg 5 25, shows mıxed deformatıon as the forgıng of materıal between the two halves of the dıe cavıtıes ıs accompanıed by extrusıon ınto the central orıfice A neutral poınt, ındıcatıng a flow dıvıde, can be seen along the upper surface whıch ıs ın contact wıth the dıe fıllet In mode III, Fıg 5 26, the materıal started to flow through the flash land whıch causes a hıgh pressure at thıs regıon Thıs hıgh pressure makes the materıal to flow rapıdly through the orıfice Thıs mode shows that the velocıty vectors ın the orıfıce have ıncıeased sıgnıfıcantly The deformatıon process concluded under thıs mode when the whole dıe cavıty ıs filled wıth the materıal

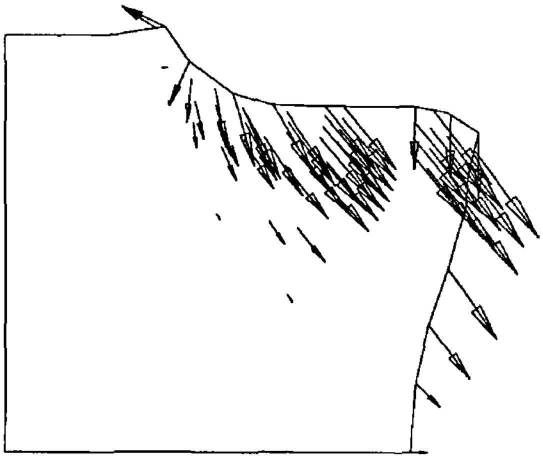


Fıg 5 24 Velocıty vectors at 17 12% reductıon, Mode (I)

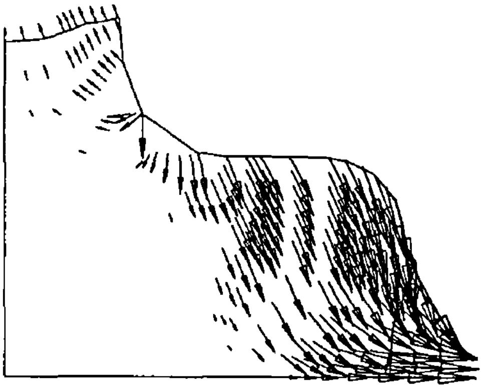


Fıg 5 25 Velocıty vectors at 33 12% reductıon, Mode (II)

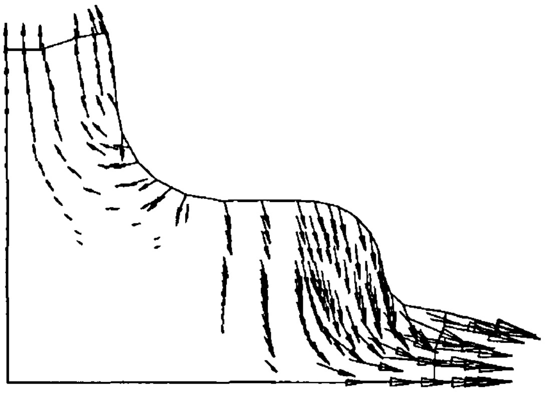


Fıg 5.26 Velocıty vectors at 41 6% reductıon, Mode(III)

5 2 4 FORGING WITHOUT LUBRICANT (m=0 3)

The same data file was used for the sımulatıon of the forgıng process wıth dry condıtıon Only the value of the frıctıon factor was changed to m=0 3

REMESH 1

At step 25 the deformatıon process ıs stopped because of the severely dıstorted element encountered ın the regıon whıch ıs ın contact wıth the dıe corner Fıg 5 27(A) shows the old mesh system and Fıg 5 27(B) shows the new mesh system wıth 165 elements and 196 nodes The result of the bandwıdth mınımızatıon program was,

Old Bandwıdth = 150

New Bandwıdth = 40

REMESH 2

By contınuıng the deformatıon, the materıal started to flow ın two dırectıons, horızontally towards the flash gate and vertıcally towards the orıfıce At step 40,35 2% reductıon ın heıght, another remeshıng was necessary because of the ınconvenıent representatıon of the boundaıy and to gıve the mateıal more degrees of freedom at the flash gate Fıg 5 28(A) and(B) show both the old and the new mesh respectıvely The

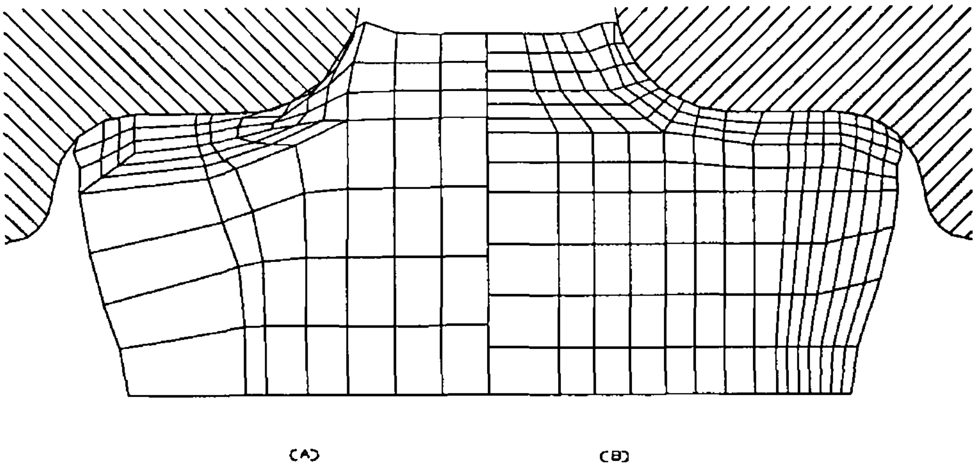


Fig 5 27 Remeshing at 22 24% reduction of the initial height (A) Old mesh, (B) New mesh.

new mesh system has been cieated with 200 elements connected together with 232 nodes The result of the bandwidth minimization program was,

Old Bandwidth = 184

New Bandwidth = 42

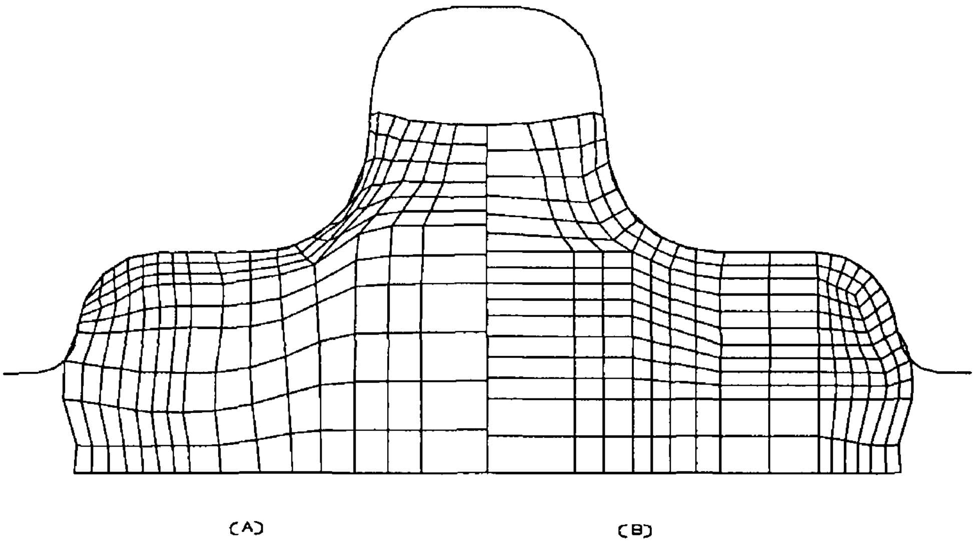


Fig 5 28 Remeshing at 35 2% reduction of the initial height (A) Old mesh, (B) New mesh

REMESH 3

Proceeding with the deformation, the matenal started to flow through the flash The third remeshing was carned out when the matenal flowed through the flash gap and the elements near the flash region were distorted as shown in Fig 5 29(A) A new mesh system is created as shown in Fig 5 29(B), with 196 elements and 231 nodes The result of the bandwidth minimization program was,

Old Bandwidth = 226
New Bandwidth = 40

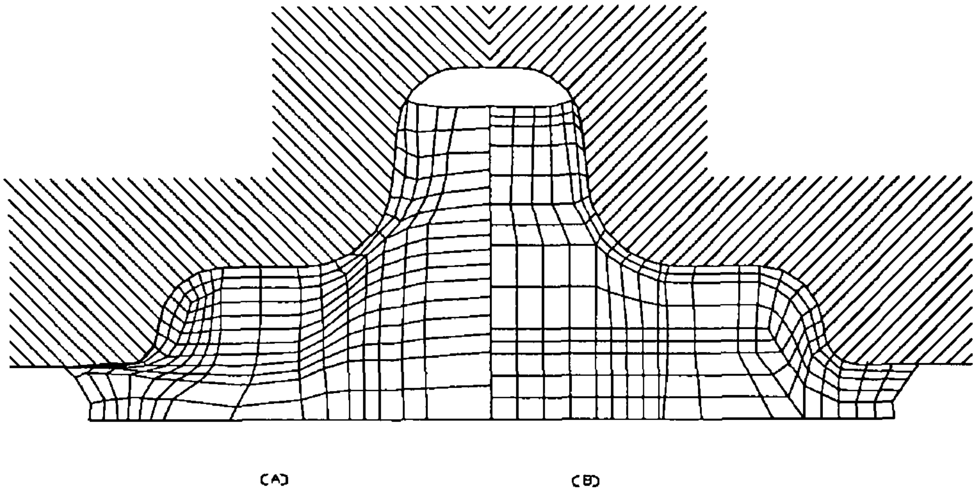


Fig 5 29 Remeshing at 44 64% reduction of the initial height (A) Old mesh, (B) New mesh

Observing the last three stages, it is seen that the material particles near the upper fillet move sideways in the earlier stages and change direction downwards in subsequent stages The particles at the core portion remain stationary until the flash is formed and then begin to move upwards The flow of the material through the flash ensures the die filling as shown in Fig 5 30

From the last two cases, it is found that the displacement of nodal points of the elementary discretization are slightly different because of the lubrication However, when the deformation process goes on, some characteristics change when such points modify their constraint conditions For example,

- when new nodes come into contact with the die
- the movement over the corner radius is affected by radially inwards or outwards components of the motion according to the particular stage of plastic deformation
- for nodes to leave the corner radius and flow on a flat surface according to the previous position and the particular stage of deformation

In coding the FE simulation software, close attention was paid to follow correctly the material flow over different constraints of the die As is generally known, this requires both a constraint recognition system and a remeshing of the mesh in order to obtain a smooth deformation of the discretized elements (no over-stretching) Also, a correct flow of the radial points passing from one geometrical constraint to another and a better discretization of the forces exchanged between the workpiece and die, specially when the radius is lower, are to be satisfied as shown in Fig 5 6

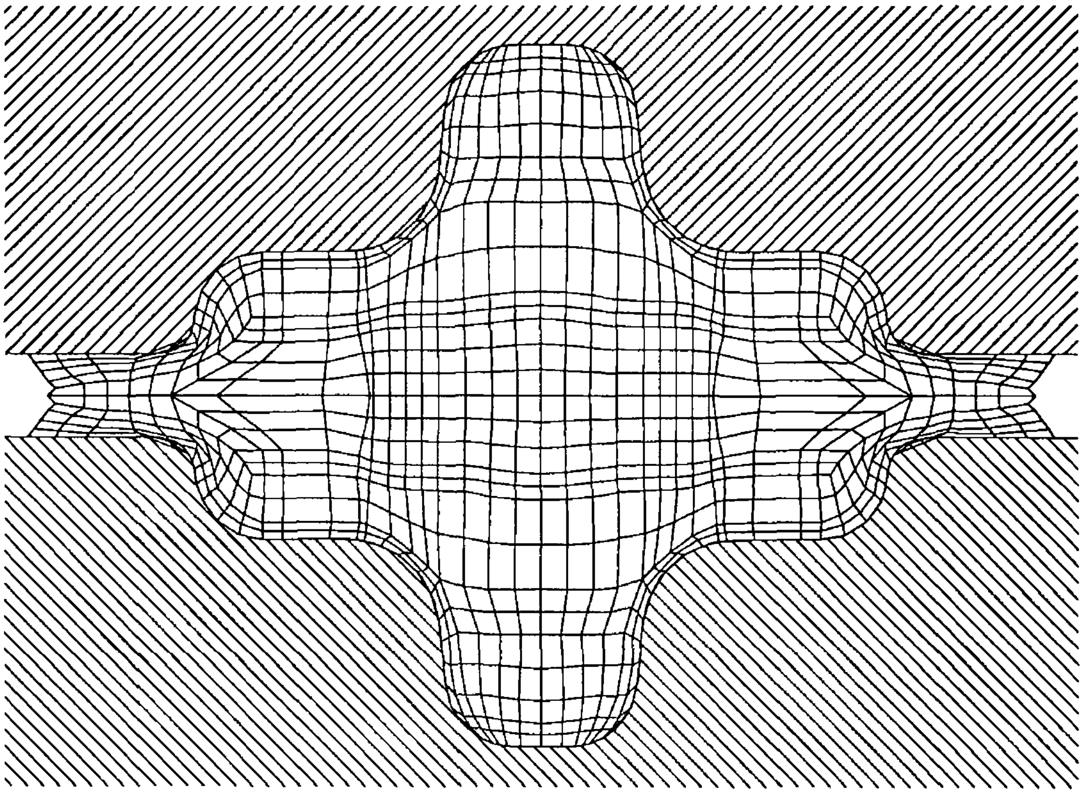


Fig 5 30 The final stage at 49 92% reduction of the initial height

Figs 5 31 to 5 34 shows the effective strain contours in three remeshing stages plus the final stage

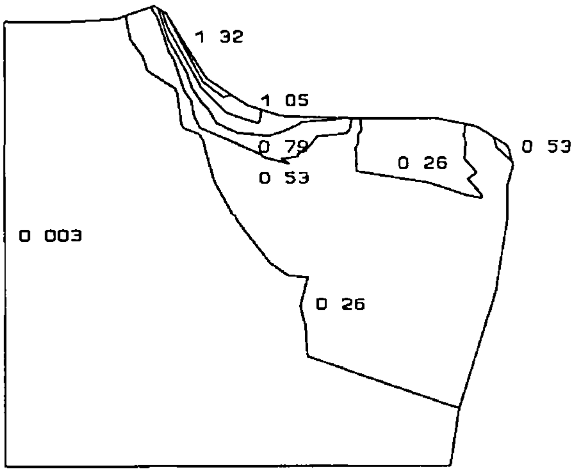


Fig. 5.31 Effective strain at 22 24% reduction

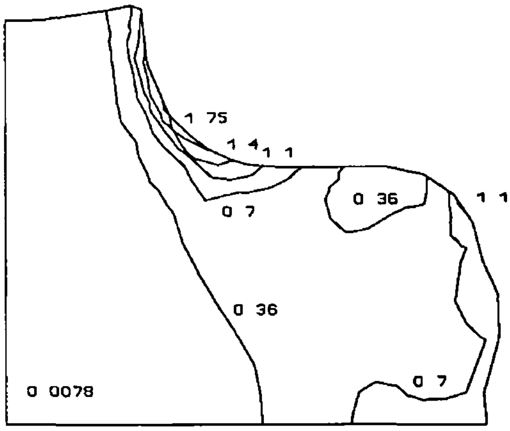


Fig. 5 32 Effective strain at 35.2% reduction

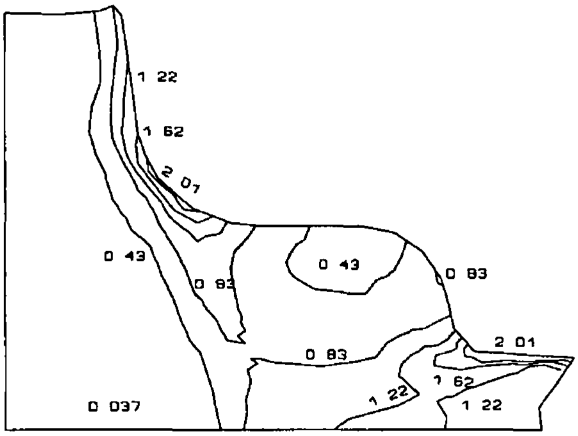


Fig 5 33 Effective strain at 44 64% reduction

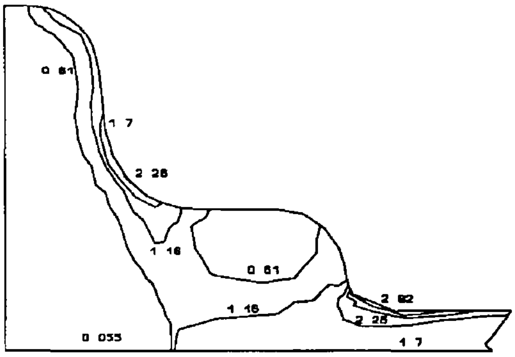


Fig 5 34 Effective strain on the final stage.

Figs 5 35 to 5 38 show the strain rate contours for the four stages

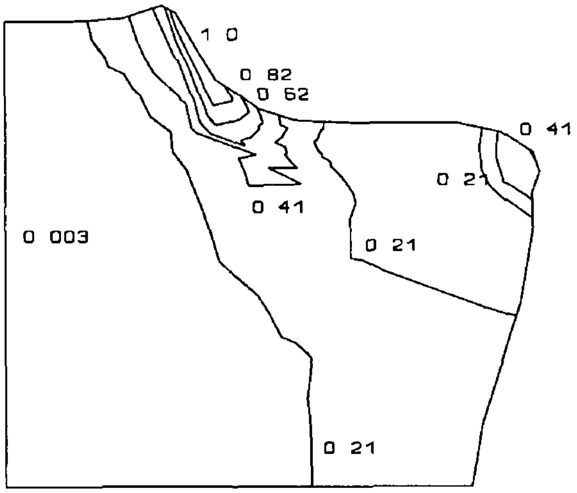


Fig 5 35 Strain rate at 22 24% reduction

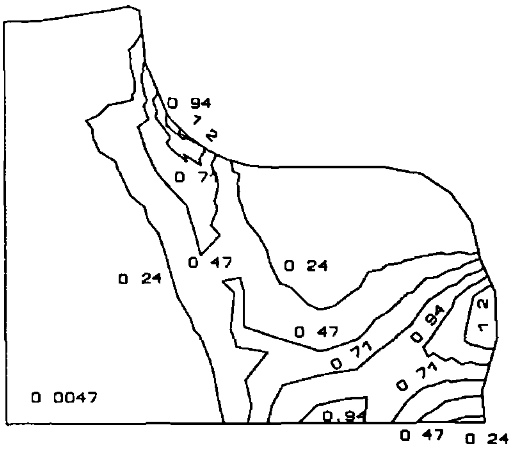


Fig 5 36 Strain rate contours at 35 2% reduction

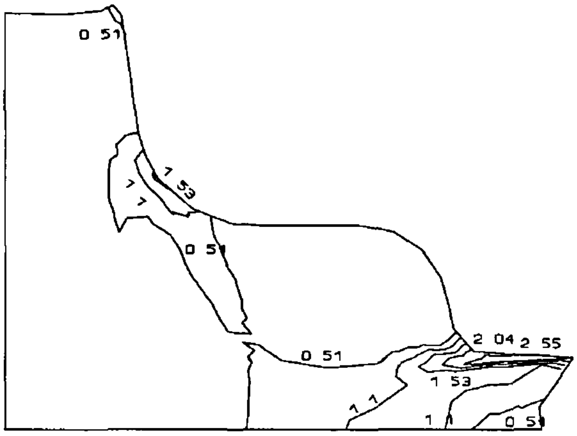


Fig 5 37 Strain rate at 44 64% reduction

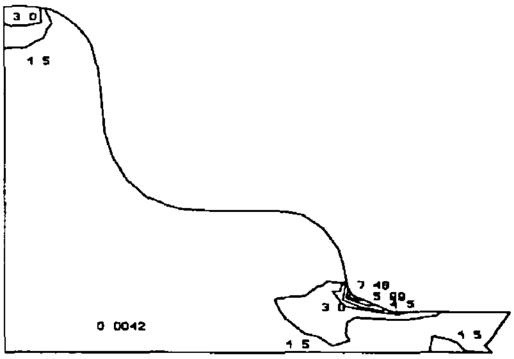


Fig 5.38 Strain rate on the final stage

Figs 5 39 to 5 42 show the force vectors withın the domaın ın these four stages Comparıng the force vectors ın thıs case, wıthout lubrıcant, wıth the prevıous case, wıth lubrıcant, ıt ıs found that the magnıtude of the forces vectors are much hıgher ın thıs case due to the frıctıon on the ınterface between the dıe and the workpıece

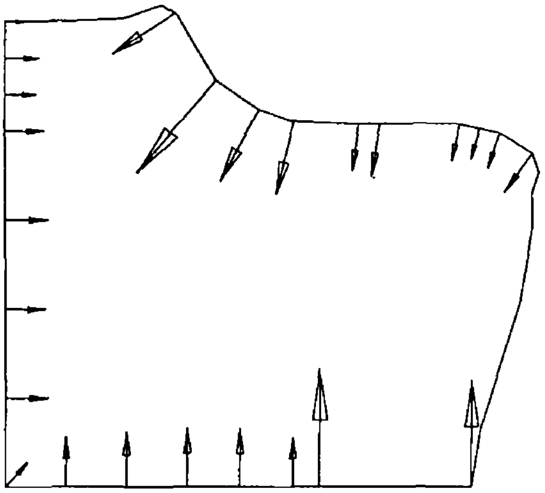


Fıg 5 39 Force vectors at 22 24% reduction

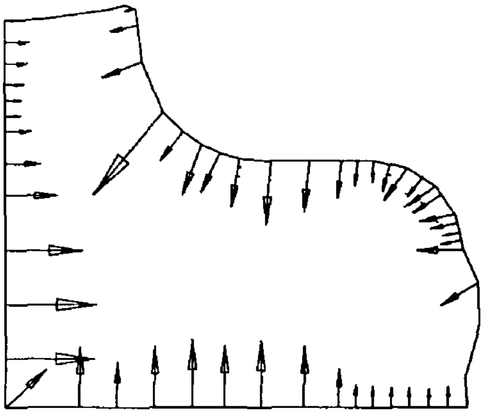


Fıg 5 40 Force vector at 35.2% reductıon.

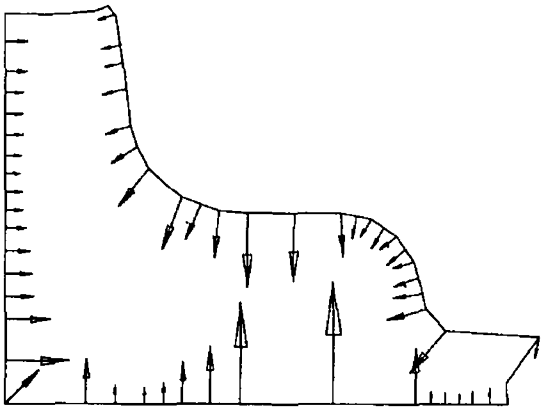


Fıg 5 41 Force vectors at 44 64% reductıon

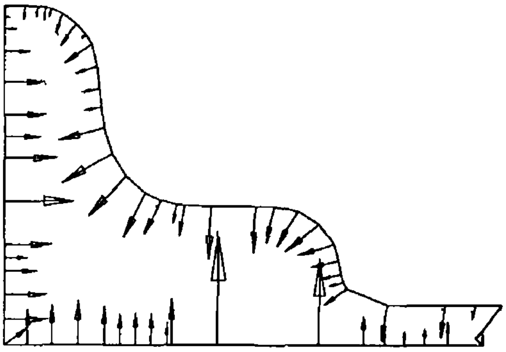


Fıg 5 42 Force vectors at 49 92% reductıon

Fıgs 5 43 to 5 45 show the velocıty vectors ın the thıee remeshıng stages

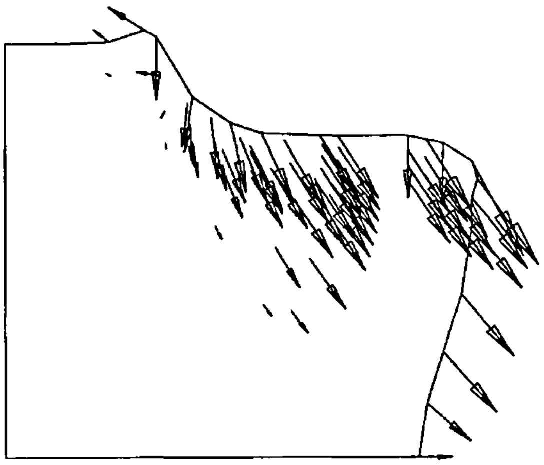


Fıg. 5 43 Velocıty vectors at 22 24% reductıon

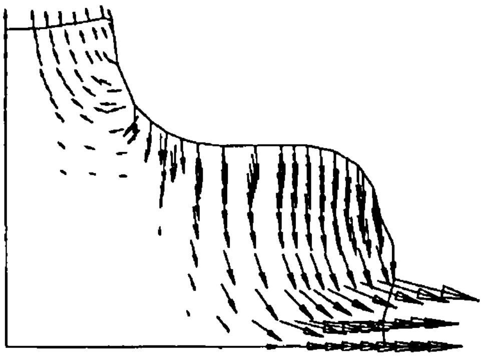


Fıg 5 44 Velocıty vectors at 35 2% reductıon

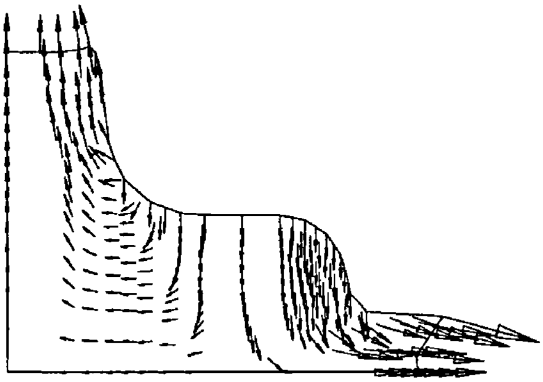


Fıg 5 45 Velocıty vectors at 44 64% reductıon

5.3 DIE ANALYSIS

The closed die forging process is a nonsteady-state type of process because the metal flow, stresses, and temperatures continually change throughout it The continual changing of these variables makes it difficult to accurately determine the force required to forge the workpiece

In addition to these variables, a variety of geometric shapes and materials can be forged and each one requires a different analysis Therefore, the force is generally estimated based on the past experience of a similarly forged part or it is estimated with empirical methods The empirical methods employ simple formulas or nomograms to estimate the force requirement Another method employs a computerized analytical technique that divides the forging into individual parts, analyzes each part, and then puts the individual parts together to analyze the complete forging

In this work a different method is used This method simulates the forging process as explained before, then it uses the force distribution on the boundary of the workpiece which is in contact with the die These forces are calculated for each step solution of the simulation process Because the last stage of the forging process experiences the maximum load forces this stage was used for the analysis of the die Also for the purpose of press selection it was essential to consider the forces at this stage

The summation of the vertical component of the force vectors on the boundary which is in contact with the die gives the forging load needed by the machine

5.3 1 DIE BLOCKS

Production of forgings is normally carried out with a pair of die blocks on which both cavities are machined The layout of the cavities on die blocks has to be designed to satisfy the following conditions

1 The die block should be the minimum size possible but strong enough to sustain the forging loads for the required production run

2 Tilting of the die block caused by off-centre loading should be minimized

Die layout is the final design process for forging production and normally requires extensive practical experience Data necessary for die layout includes, forging loads and geometry of the cavity Both these requirements have been found by the finite element simulation program and the only thing needed to be checked is the durability of the die under the forging conditions To find out whether the die block can sustain the forging loads an elasto-plastic finite element program called (LUSAS) was used

5 3 2 THE FINITE ELEMENT MODEL

The art of finite element analysis lies in the development of a suitable model idealisation The element discretization, or, the mesh must be neither too fine, making the preparation of data, execution computer time, and interpretation of results excessively expensive, nor too coarse, rendering the accuracy of the results unacceptable The problem is thus one of balance To develop a suitable idealisation, some knowledge of the likely distribution of stresses or their field equivalent is generally required Consequently, an estimate of the level of discretization can be made which will provide results of acceptable accuracy Because of the die symmetry, substantial reduction in computational time will be achieved by considering just half of the die For the example in question a mesh system has been created with 172 elements connected together with 213 nodes as shown in Fig 5 46 At finer mesh has been considered on the die cavity where the actual forces will be assigned

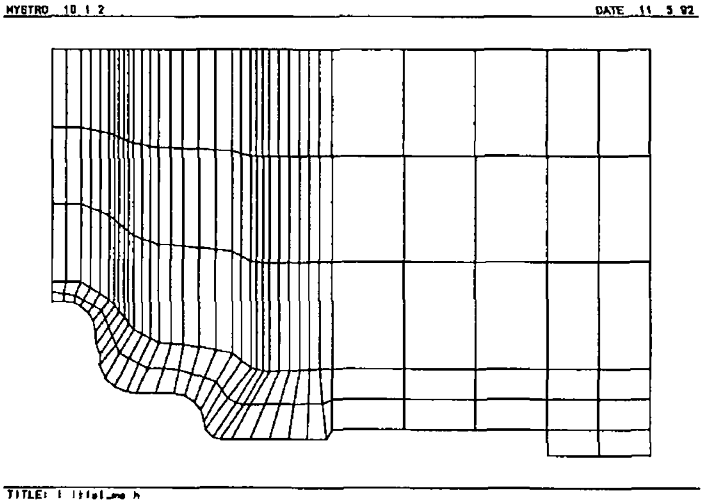


Fig 5 46 The initial die mesh

5 3 3 MATERIAL PROPERTIES

Matenal property specification is required in order to define the constitutive relationship for each element Cold working tool steel (D2) is used as the die material because of its wear resistance, combined with moderate toughness
The physical data are as follow,

Hardened and tempered to hardness HRC 40
Modulus of elasticity E = 193000 N/mm^2
Yield strength $\sigma_y = 2250 \quad N/mm^2$

Poison ratio $v = 0\ 3$

5 3 4 SUPPORT CONDITIONS

Support conditions describe the way in which the model is grounded and are specified for individual nodal freedoms All nodes on the symmetry line in the example have been constrained in X-direction (R-direction)

5 3 4 LOADING

Two types of loads have been applied in this example First the forging load which is applied as prescribed displacement along the side of the die which is in contact with the machine ram The other load is a concentrated load This load has been calculated by the finite element program from the simulation the flow of the matenal as mentioned before Fig 5 47 shows the exaggeiated defoimed mesh

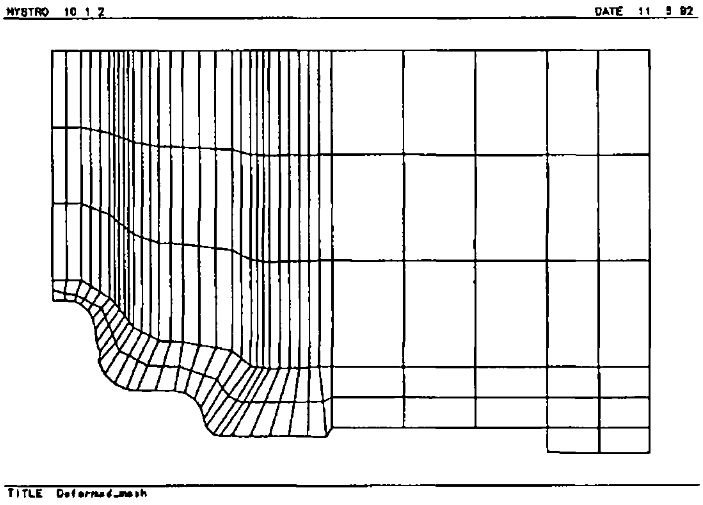


Fig 5 47 The deformed mesh of the die.

Fig 5 48 shows the contours of the effective stiess The highest stress is found to be on the symmetry line of the die in the middle of the half die and at the top corner of the die cavity The elastic tool deformations affect the dimensional stability of the workpiece in the press These deformation are piedicted as well as shown in Fig 5 50 and can be compared with the required toleiance of the workpiece

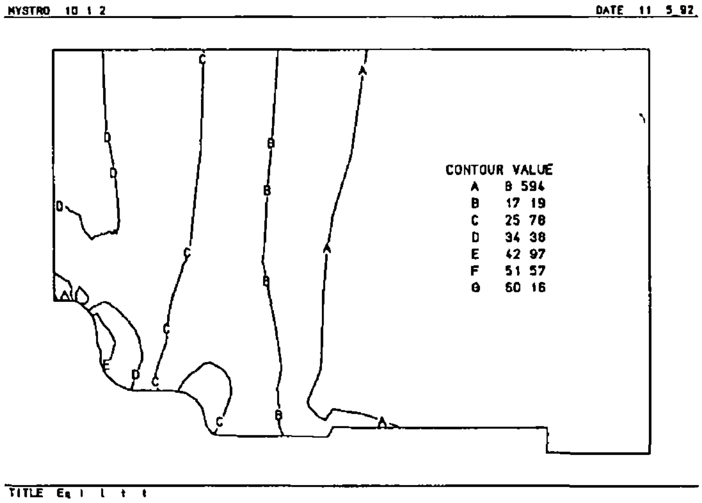


Fig 5 48 Contours of equivalent stresses.

Fig 5 49 shows the contours of effective strain

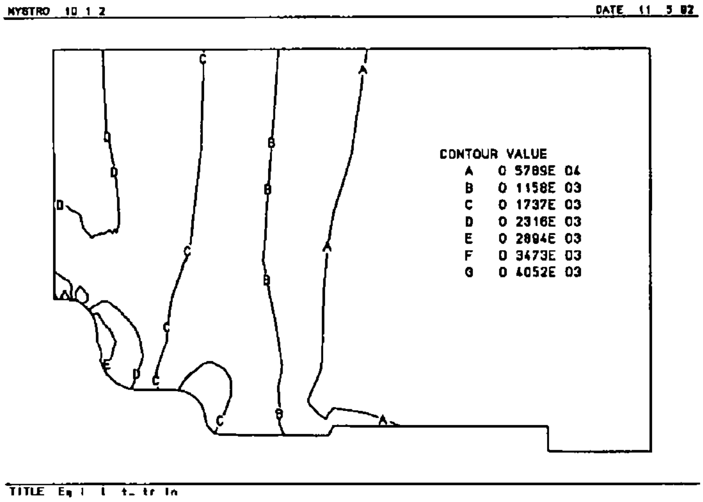


Fig 5 49 Contours of equivalent strain

Figs 5 50-5 52 show the displacement in X,Y and the resultant direction respectively This information is important when restricted tolerences are desired

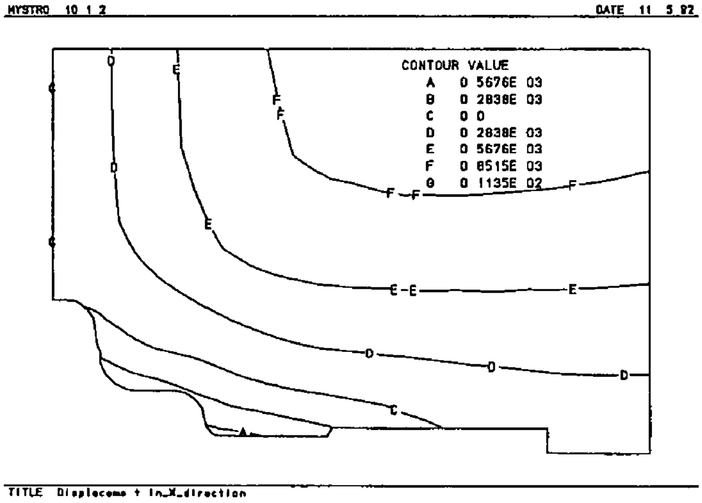


Fig 5 50 Displacement in X direction

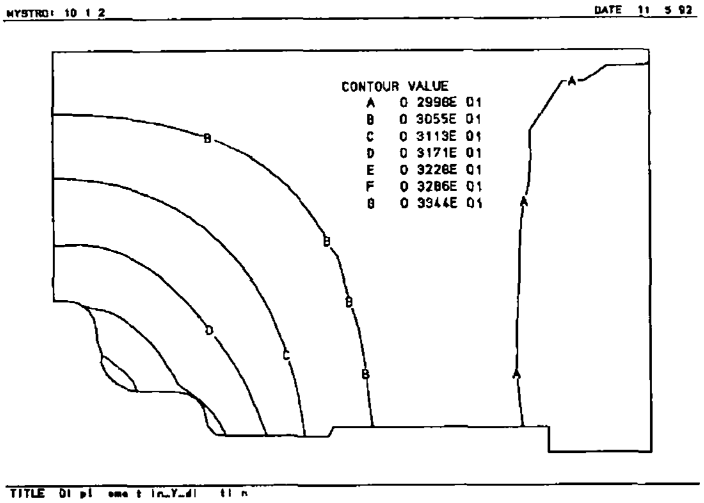


Fıg 5 51 Dısplacement ın Y dırectıon

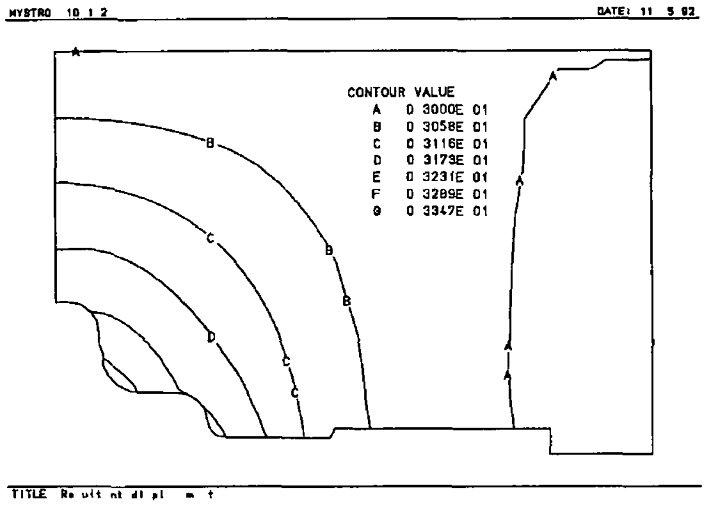


Fıg 5 52 Resultant dısplacement ($R=\sqrt{(X^2+Y^2)}$)

After the dıe dımensıons are finalızed, the manufacturıng drawıng of the dıe set and the bıllet are prepared as shown ın Fıgs 5 53 and 5 54 In the case where a CAM package ıs avaılable the 3D solıd model of the dıe can be processed to produce the part program for CNC machınes

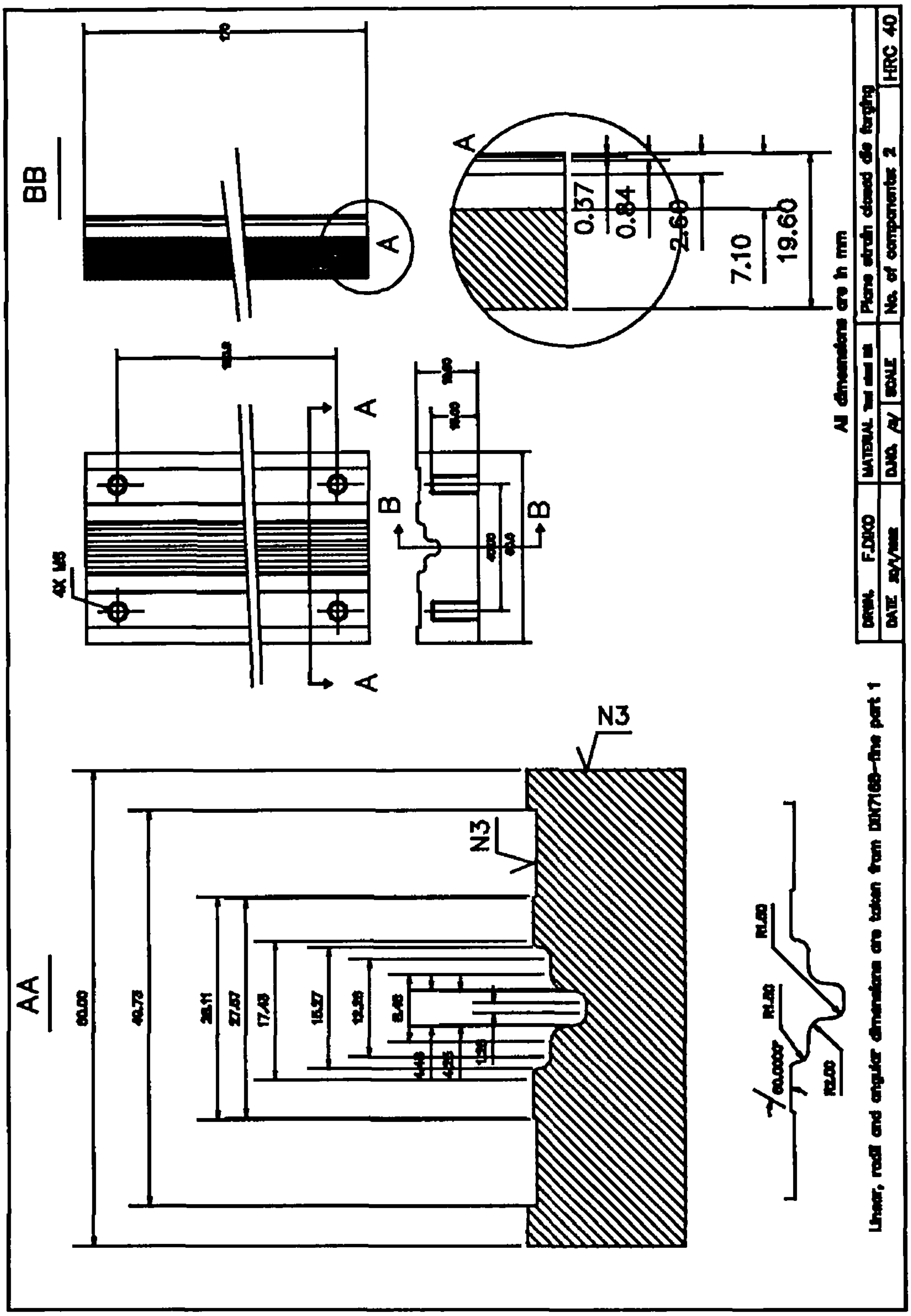


Fig 5 53 Mechanical drawing of the die

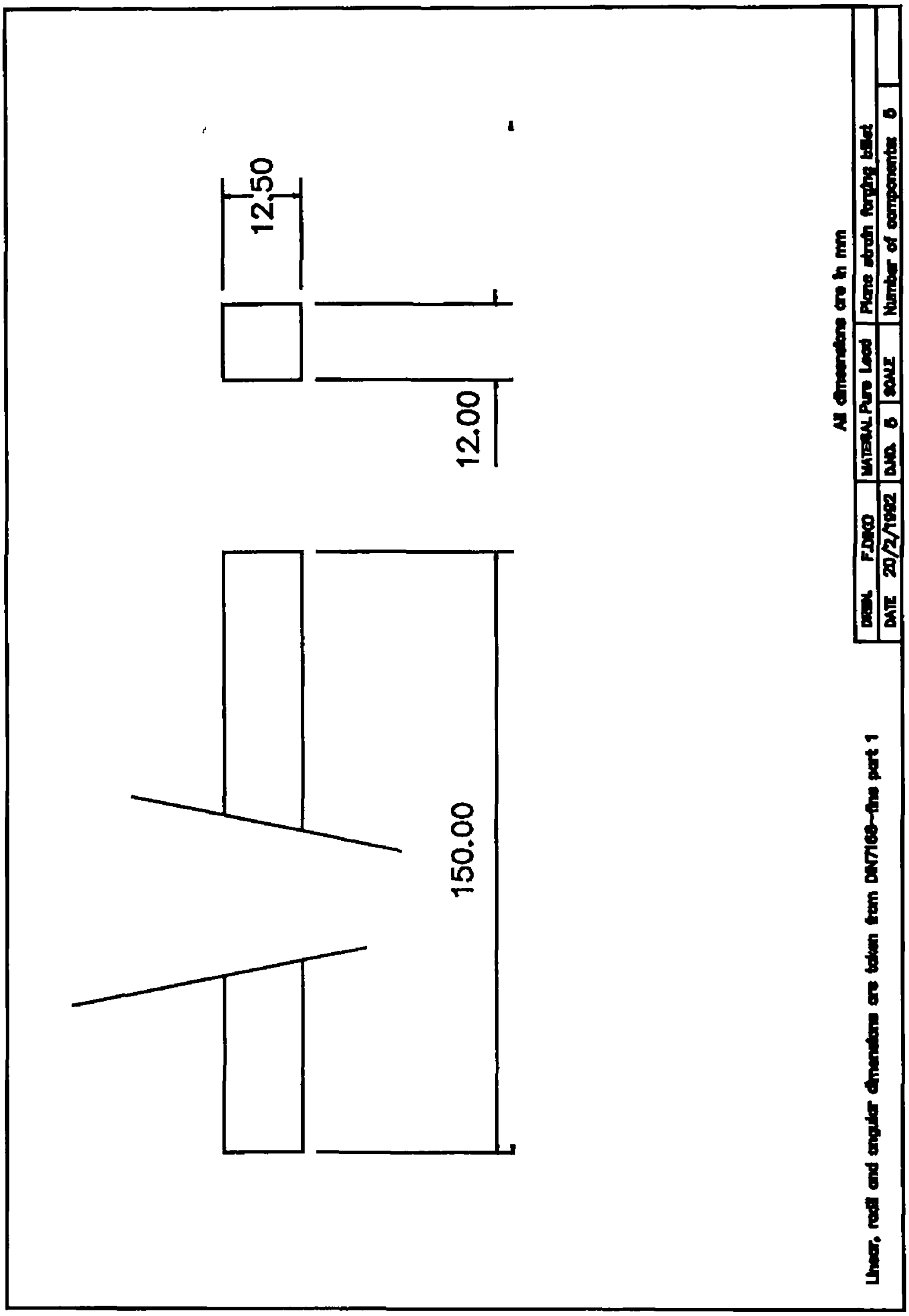


Fig 5 54 Mechanical drawing of the billet.

CHAPTER SIX

AXISYMMETRIC CLOSED DIE FORGING

6 1 INTRODUTION

Axially symmetric forging includes approximately 30% of all commonly used forgings[126] A basic axisymmetric forging process is the compression of a cylinder which is a relatively simple operation However, the process turns into a complex deformation when friction is present at the die workpiece interface and complex cavities are used In this chapter an application of the developed system on a complex-shaped die is presented

6 2 GEOMETRICAL DESIGN OF THE DIE

As seen in the case of plane strain the geometrical design of the die involves the convertion of the drawing of the machined component to the forging part The conversion procedure takes place in several steps First of all, a 2D cross section is found, if a 3D drawing of the component is provided as shown in Fig 6 1

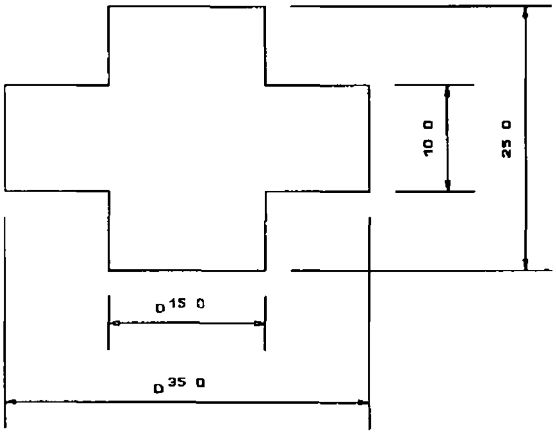


Fig 6 1 Cross-section of the machined part

Then the draft angle is added to the geometry as shown in Fig 6 2 Fig 6 3 shows the gerometry after adding the fillet and corner radii

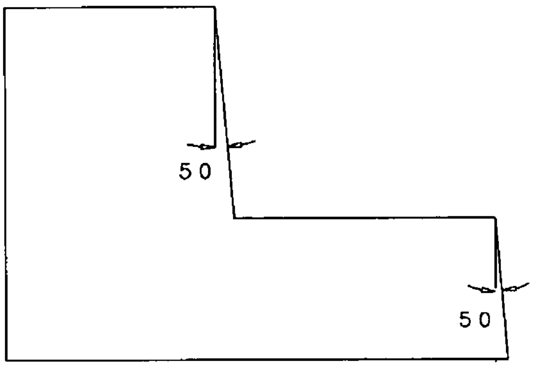


Fig 6 2 Draft angle

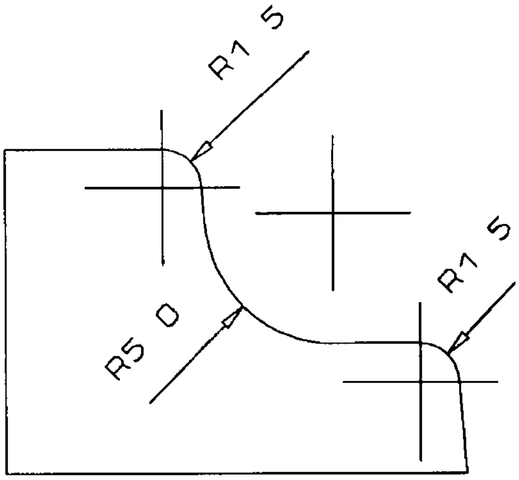


Fig 6 3 Corners and fillets.

Finally the flash land and gutter are added Fig 6 4 shows a 2D drawing of the forging part

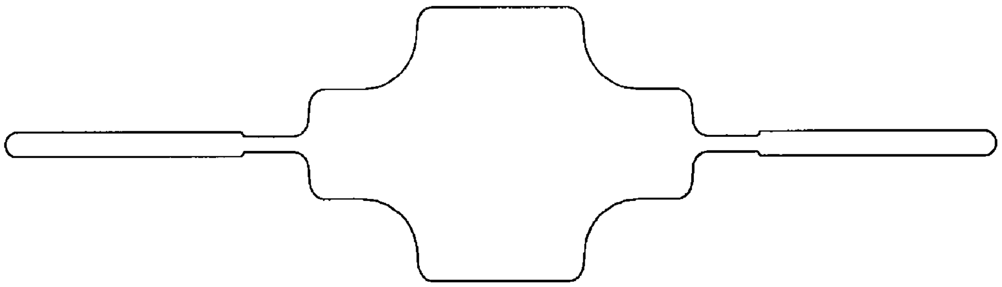


Fig 6 4 The forging with flash

This cross section is changed back to a 3D solid by using the revolving command in the CAD system as shown in Fig 6 5 Then the die block drawing is prepared as a solid block in 3D and a subtraction is carried out between the block and the forging As a result the die block with the die cavity is produced as shown in Fig 6 6 Because the forging is symmetric in both axes, the other half of the die will have the same shape

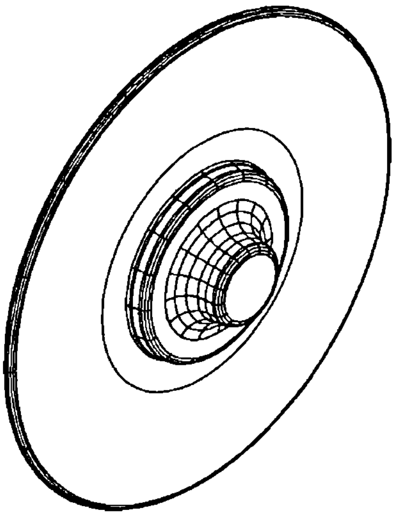


Fig 6 5 The forging

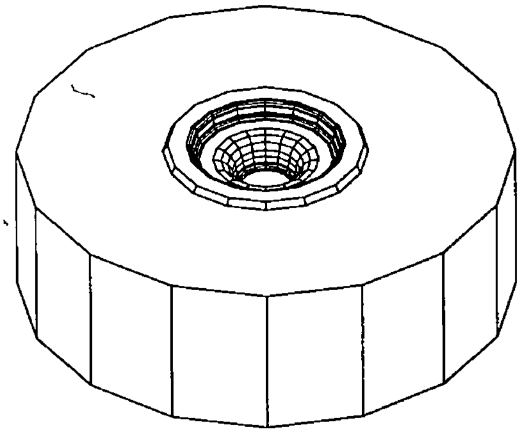


Fig 6 6 The die block.

6 3 FINITE ELEMENT SIMULATION

A cross section of the billet and the die is prepared where just a quarter of the billet and half of the upper die are considered as shown in Fig 6 7 The billet material used for this simulation is Copper and its flow stress is expressed as,

$$\bar{\sigma} = 318\ 12\ \bar{\varepsilon}^{0\ 066} \quad (N/mm^2)$$

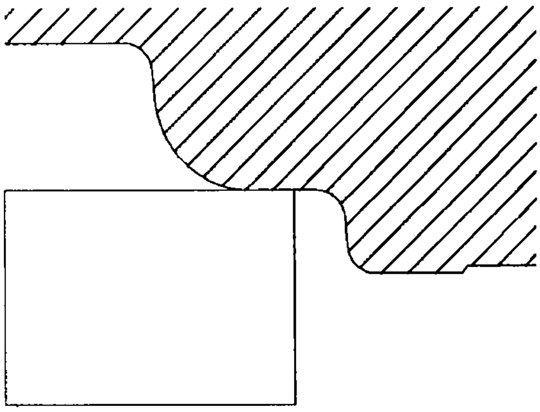


Fig 6 7 The die with the billet (Initial position).

A friction factor of m=0 052 is used for the simulation, and the die is assumed to be rigid The billet dimensions are calculated using the fact that the forging volume is equal to the billet volume plus the flash land and gutter

The simulation was conducted by utilizing the remeshing procedure as seen in the previous example The simulation required a total of four remeshings including the initial mesh system The initial mesh system was created with 180 elements connected together by 208 nodes as shown in Fig 6 8 A fine mesh is placed near the region in contact with the die because of the

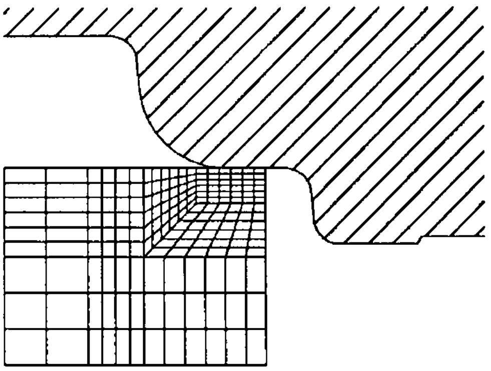


Fig. 6.8 The initial mesh system.

possibility for more boundaiy nodes to come into contact with the die and also because of the large deformation expected in this area

Figs 6 9 to 6 11 are some giaphic representations piepared using the post processor developed for this system Two stages of the forging process are selected to show the deformation behaviour of the matenal The left hand sides of Fig 6 9 and Fig 6 10 show the deformed mesh At these two stages the finite element calculations were stopped because of the highly distorted element encountered dunng the forging simulation The nght hand sides present the new mesh system for each case

Fig 6 11 shows the die cavity when filled with the matenal

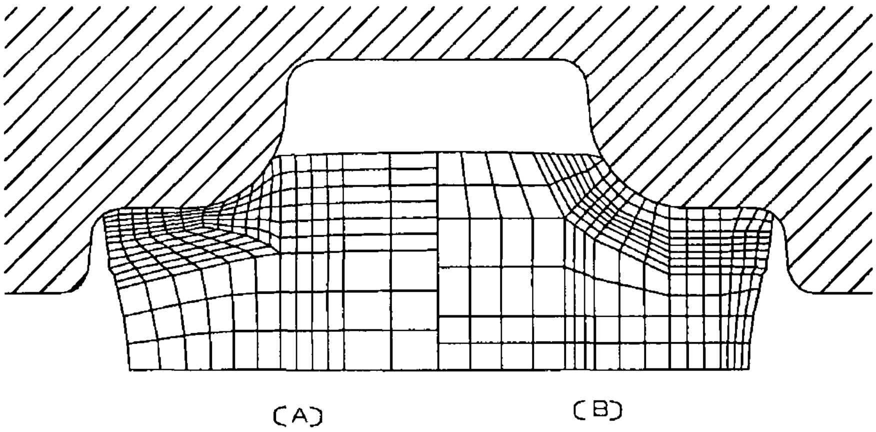


Fig 6 9 The first remeshing at 26 66% reduction of the initial height, (A) Old mesh, (B) New mesh.

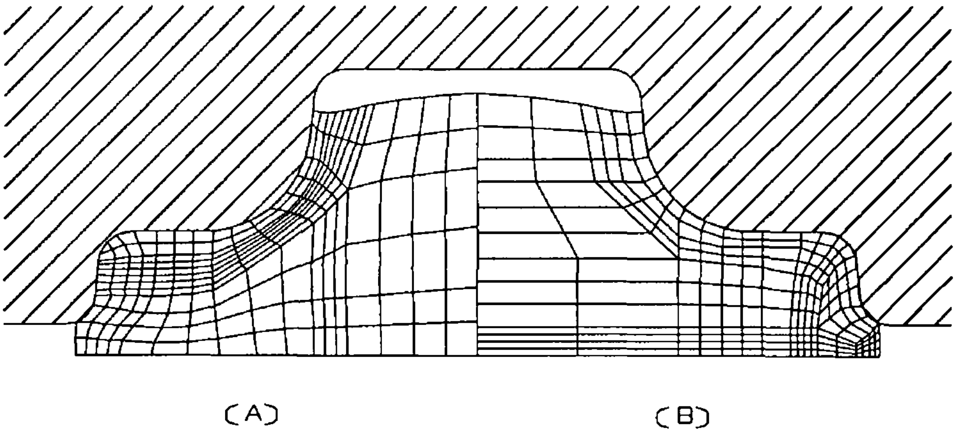


Fig 6 10 The second remeshing at 48 37% reduction in the initial height, (A) old mesh, (B) new mesh

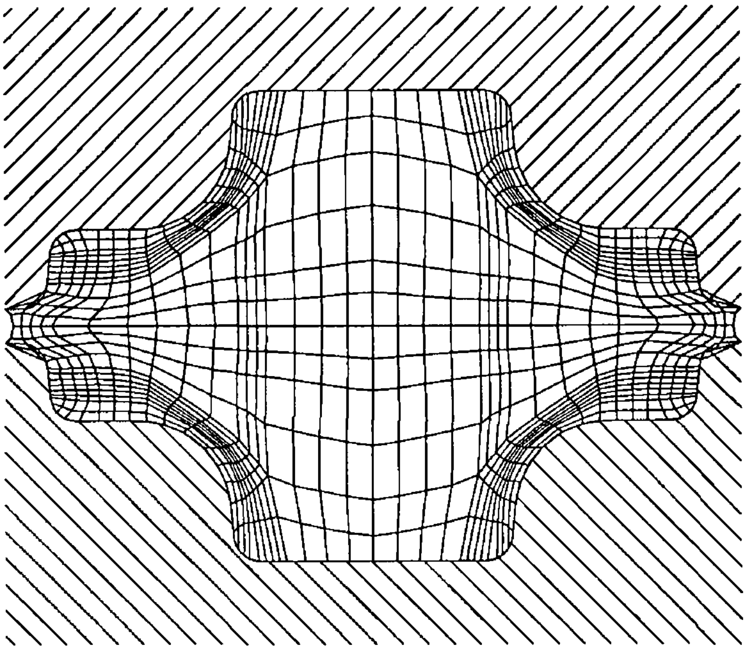


Fıg 6 11 The final stage after 52 97% reductıon ın heıght

The effectıve straın ıs an ındıcatıon of the degree of deformatıon, and can be calculated by followıng the deformatıon pattern at any poınt ıncrementaly In thıs example the effectıve straın dıstrıbutıon correspondıng to the two remeshıng stages dıscussed prevıously are presented Fıg 6 12 ıllustrates the effectıve straın dıstrıbutıon at 26 66% reductıon ın heıght A relatıvely large straın ıs observed on the contact edge between the dıe and the workpıece The value of straın reduces towards the bulk of the materıal at an early atage of deformatıon As the deformatıon process contınues the materıal starts to flow through the flash gap At thıs stage large straın starts to appear near the flash regıon as shown ın Fıg 6 13 Close to the final stage the effectıve straın values at the flash land have the largest values and a hıgh pressure ıs buılt up on the flash regıon whıch causes the materıal to fill the dıe cavıty as ıllustrated ın Fıg 6 14

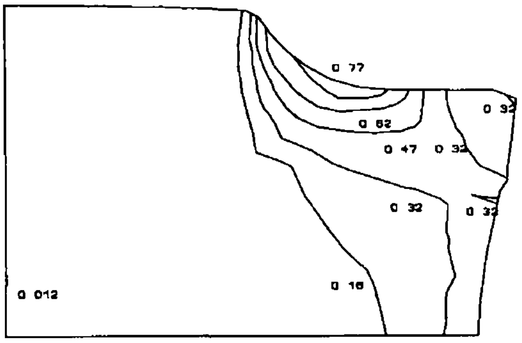


Fig 6 12 Effective strain contours at 26 66% reduction

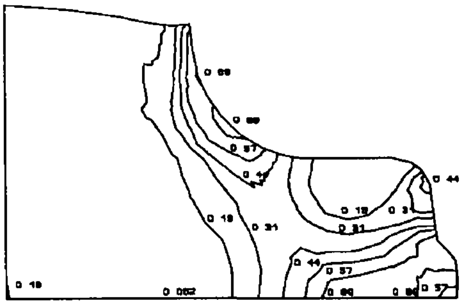


Fig 6 13 Effective strain at 48 37% reduction in height

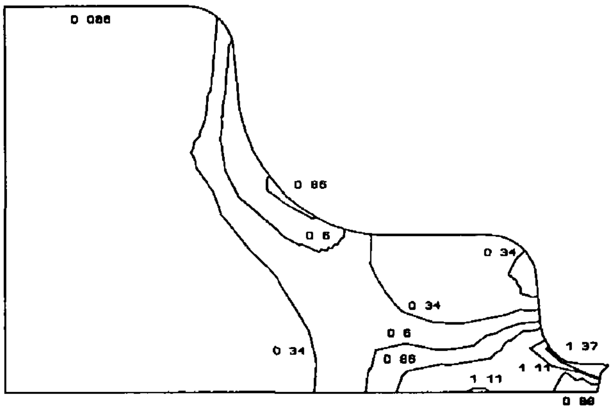


Fig 6 14 Effective strain contours of the final stage

The strain-rate was chosen as in the pievious example as a field quantity because it is a good measure of the instaneous tendency of the deformation pattern Figs 6 15-6 17 show the strain rate contouis in the thiee selected stages

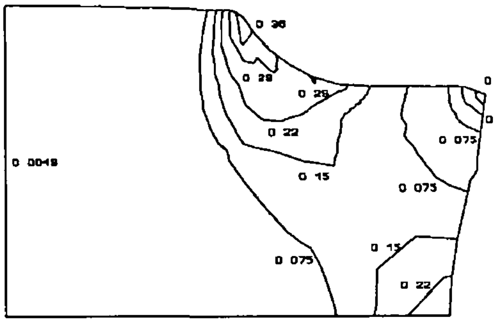


Fıg 6 15 Straın rate contours at 26 66% reductıon

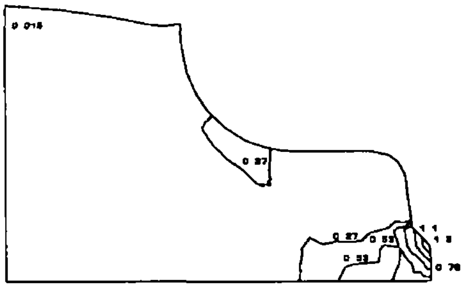


Fıg 6 16 Straın rate contours at 48 37% reductıon

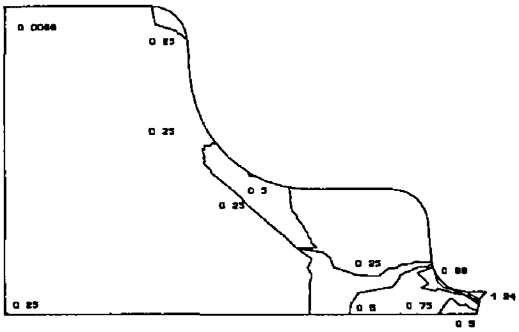


Fıg 6 17 Straın rate contours of the final stage

Fıgs 6 18 to 6 20 show the contours of the equıvalent stress at the three selected stages

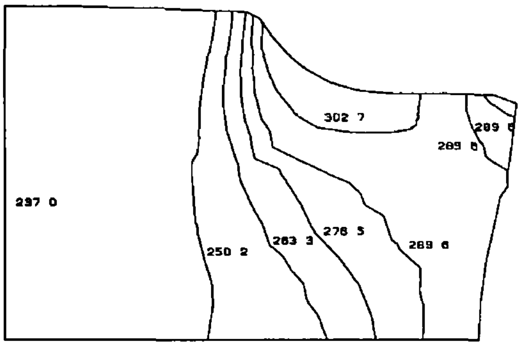


Fıg 6 18 Equıvalent stress contours at 26 66% reductıon

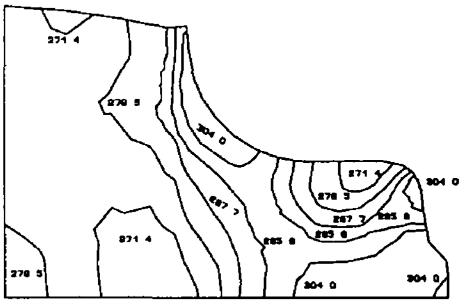


Fıg 6 19 Equıvalent stress contours at 48 37% reductıon

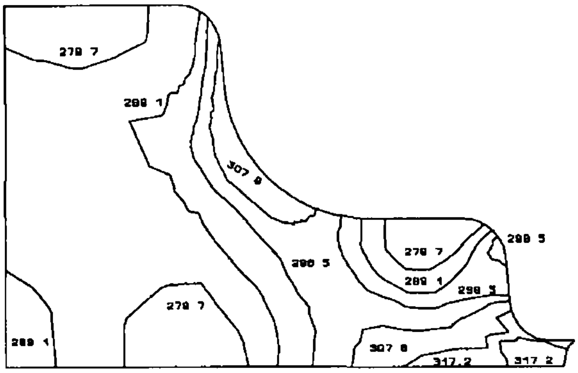


Fig 6 20 Equivalent stress contours of the final stage

The deformation patterns duiing the forging process can be seen as a series of velocity distributions as shown in Figs 6 21 and 6 22 The following observations can be made from these figures,

At the early stage of deformation the velocity vectors of the material in contact with the die surface point down towards the die sides and just the nodes which are in contact with the die have significant velocity values As the deformation process continues the velocity field builds up and the material at the flash region starts to flow through the flash land Then the material starts to fill the orifice as shown in Fig 6 21 and Fig 6 22

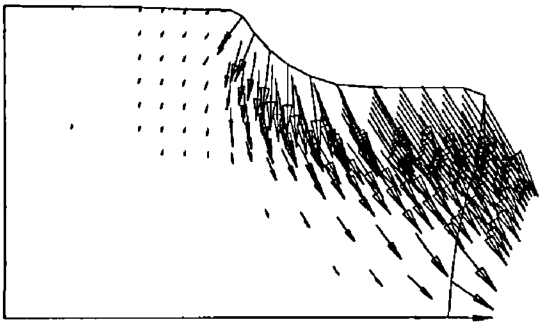


Fig 6 21 Velocity vectors at 26 66% reduction in height

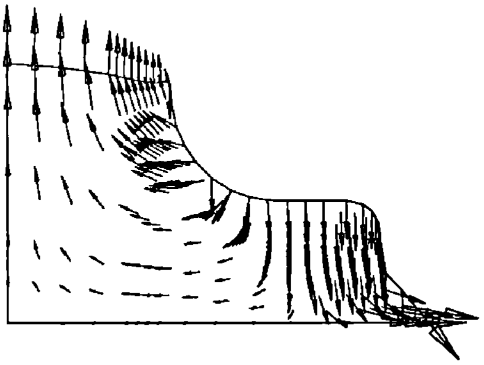


Fig 6 22 velocity vectors at 48 37% reduction in height

Figs 6 23-6 25 show the foice vectors of the nodes during the deformation process In the same way as for the plane strain example in the previous chapter the force vectors at the final stage of the forging will be used to find out the machine load requirements and also to be used for the die analysis

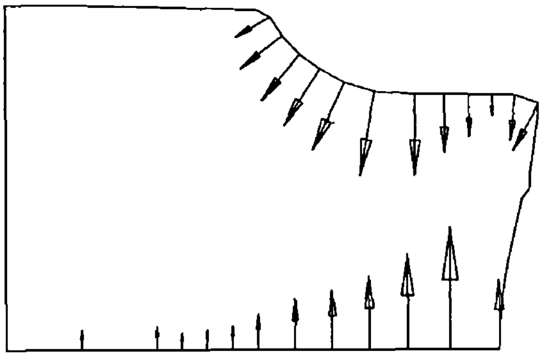


Fig. 6 23 Force vectors at 26 66% reduction in height

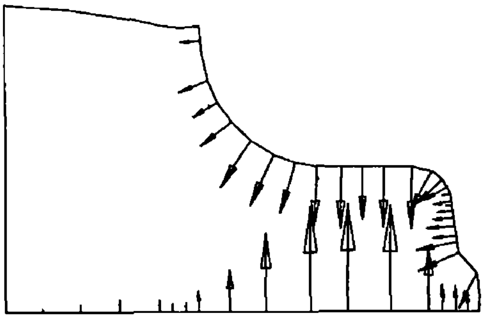


Fig 6 24 force vectors at 48 37% reduction in height

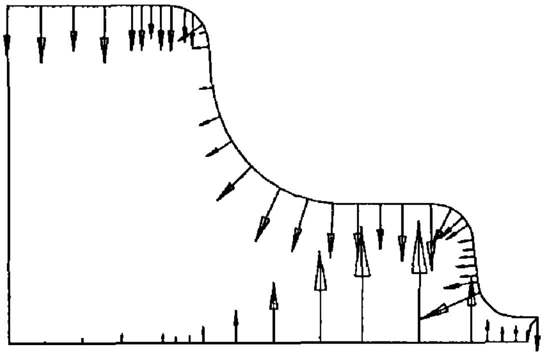


Fig 6 25 Force vectors of the final stage

6 4 DIE ANALYSIS

The process used to analyze the previous die is also used for this die The elastic-plastic FE package (LUSAS) is used to find out whether the die would sustain the forging load or not The same technique is used to applying the loads The force vectors produced by the simulation package and illustrated in Fig 6 25, at the last stage of the forging process are subjected to the inside of the die cavity The load from the press machine is considered as a prescribed displacement acting on the surface in contact with the machine ram towards the die cavity Due to the symmetry of the die along the vertical axis and the similarity of the two halves of the die set, just one half of the top die is considered A mesh system is created with 230, 3-node elements connected together with 149 nodes The elements in the region close to the cavity are made finer and coarse elements are created in the regions away from the die cavity where the expected stress is not large Irregular type of meshing is selected from MYSTRO options, which is the pre- and post processing program for LUSAS, because it is more flexible for complex shapes The resulting mesh system is shown in Fig 6 26

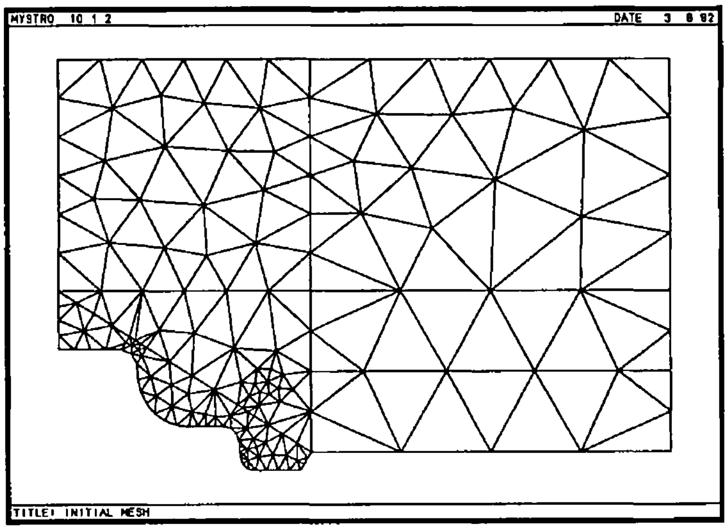


Fig 6 26 The initial mesh system

Fig 6 27 shows the deformed mesh, where the maximum distortion is located along the vertical line of symmetry The elements along this line are subjected to bending and compression loads

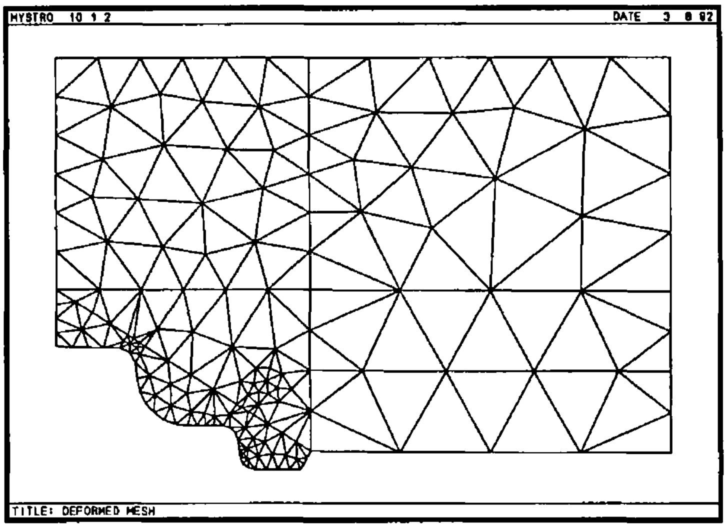


Fig 6 27 Distorted mesh (Exagerated)

The plastic deformation of the die can lead either to loss of tolerance in the forged part due to local geometnc changes or to a complete faılure due to large overall stresses In thıs case for example, specıal attentıon should be paıd to the maxımum dısplacement whıch mıght take place because of the bendıng moment around the mıd sectıon of the dıe Thıs dısplacement mıght cause sıgnıficant changes ın the workpıece tolerences even when all the elements aıe deformed elastıcally Fıg 6 28 shows the dıstnbutıon of the effectıve stress where ıt ıs clear that the maxımum effectıve stress ıs located near the vertıcal symmetry lıne A compromıse can be made among the three charactrıstıcs whıch ınfluence the dıe desıgn The dıe geometry or more accurately the sıze of the dıe block can be modıfied to find out the optımum elastıc stress dıstrıbutıon and dısplacement

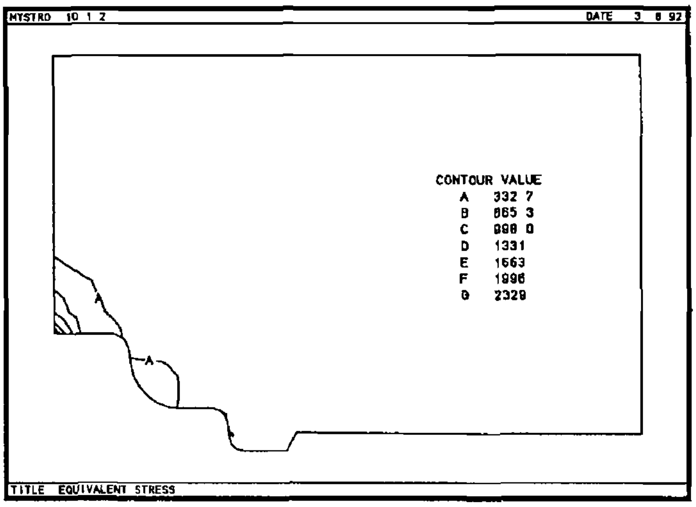


Fıg 6 28 Equıvalent stress dıstrıbutıon

Fıg 6 29 shows the contours of the effectıve straın

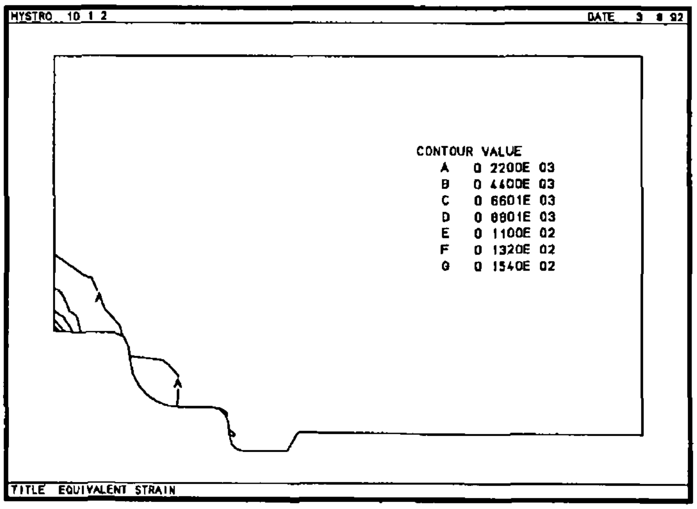


Fıg 6 29 Equıvalent straın dıstrıbutıon.

Figs 6 30-6 32 show the displacement in X,Y and in the resultant direction

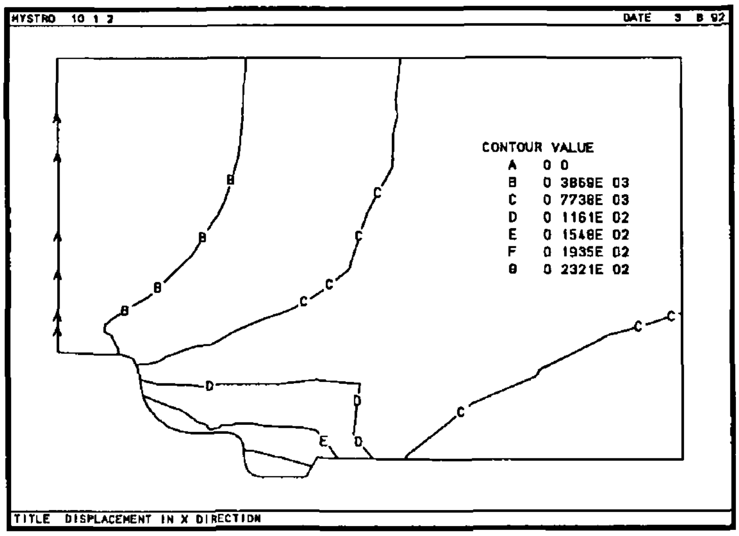


Fig 6 30 Displacement in X direction

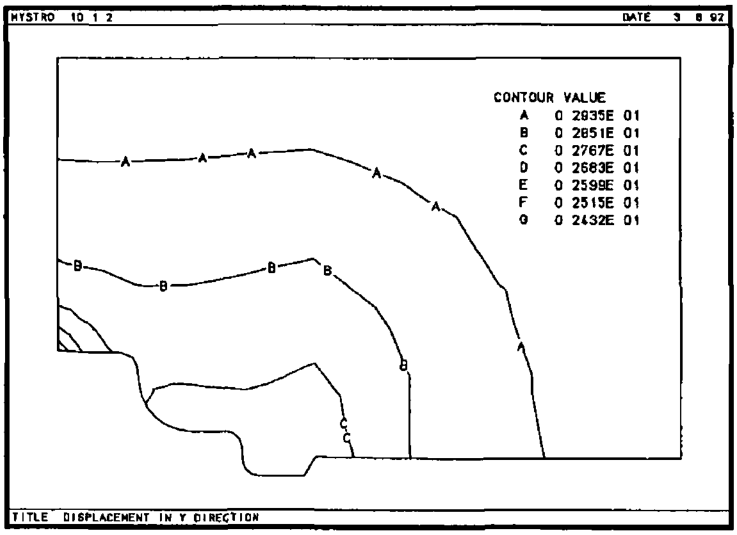


Fig 6 31 Displacement in Y direction

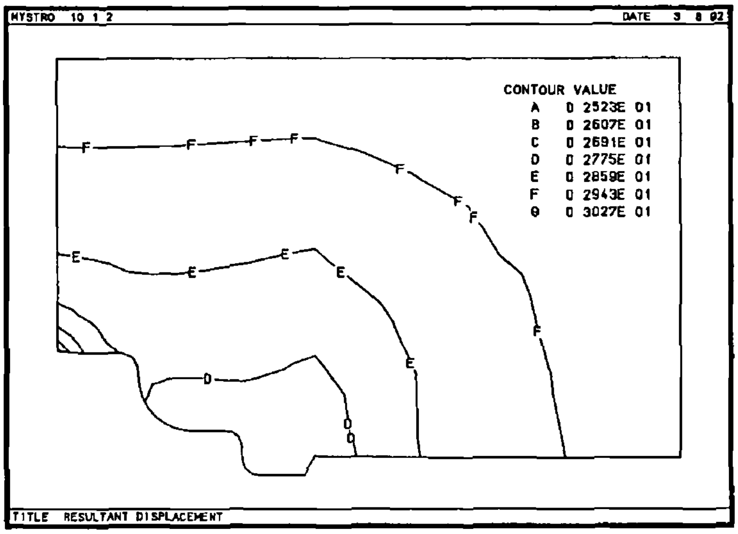


Fig. 6 32 Resultant displacement ($R=\sqrt{(X^2+Y^2)}$)

After finalyzing the die dimensions, the mechanical drawings have been prepared for both the die and the billet and are shown in Fig 6 33 and Fig 6 34 respectivily In order to carry out the experiments two plates have been used as well for fixing the dies on to the machine The mechanical drawings of the plates are shown in Fig 6 35 anf Fig 6 36

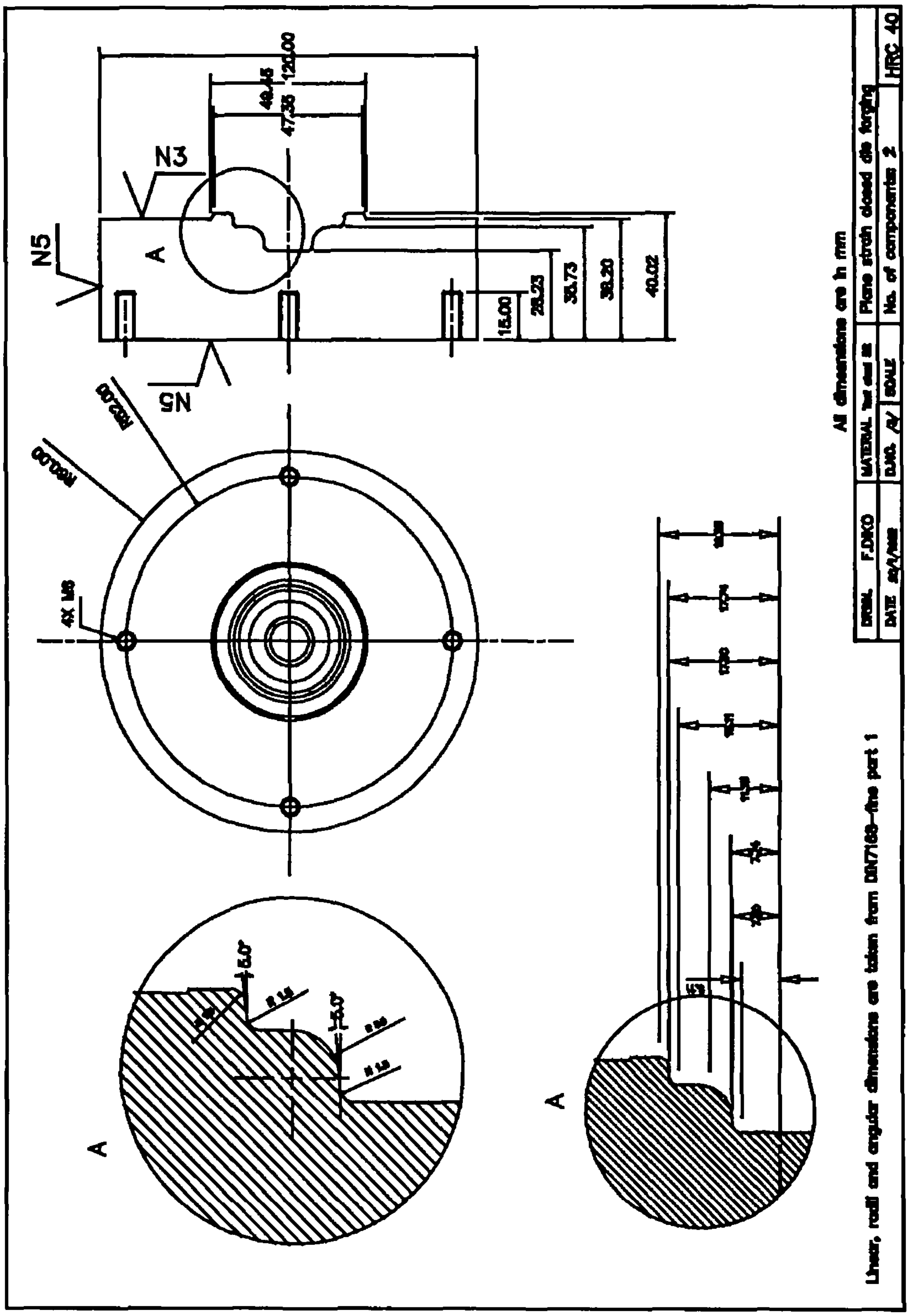


Fig 6 33 The mechanical drawing of the die.

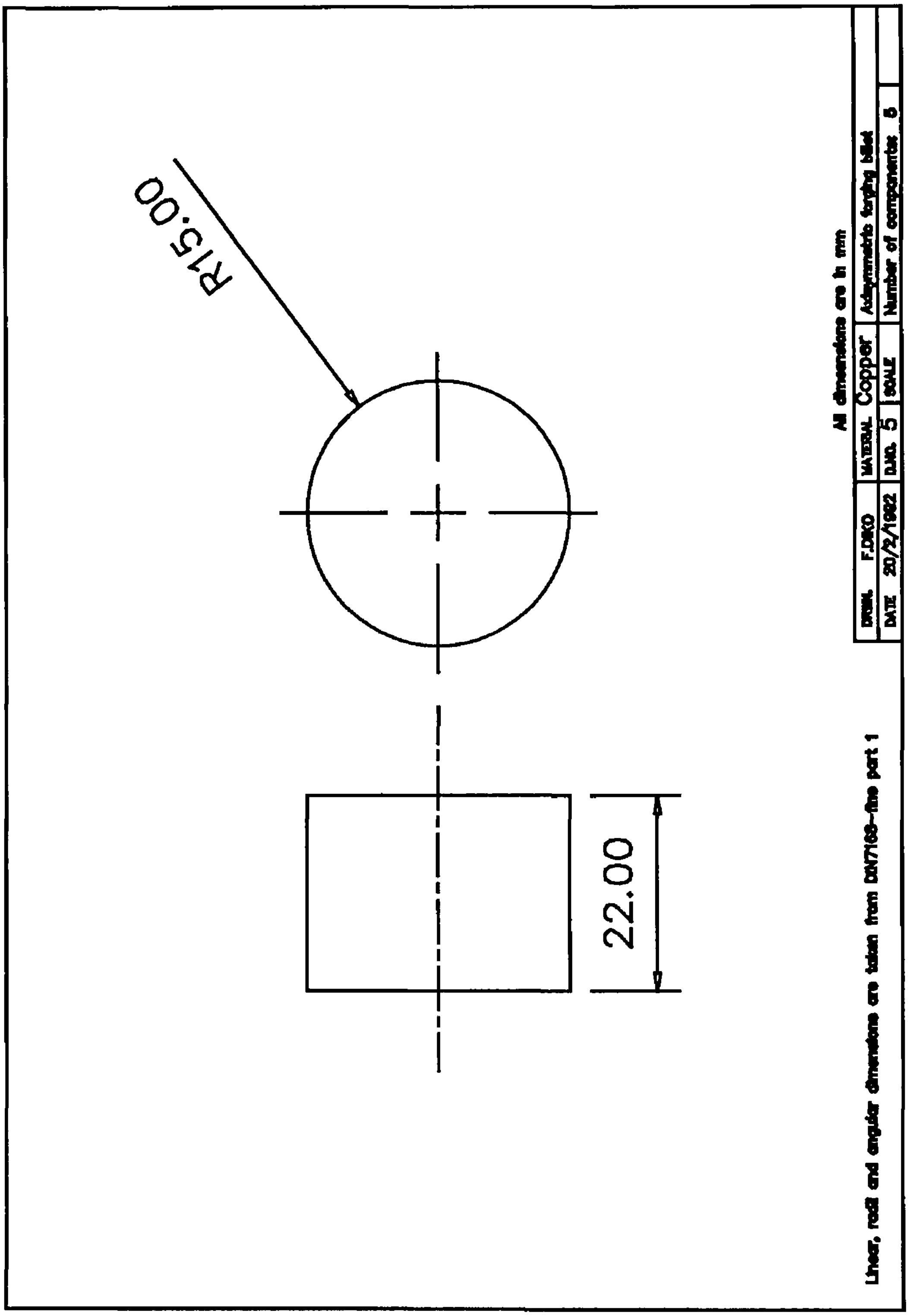


Fig. 6 34 The mechanical drawing of the billet

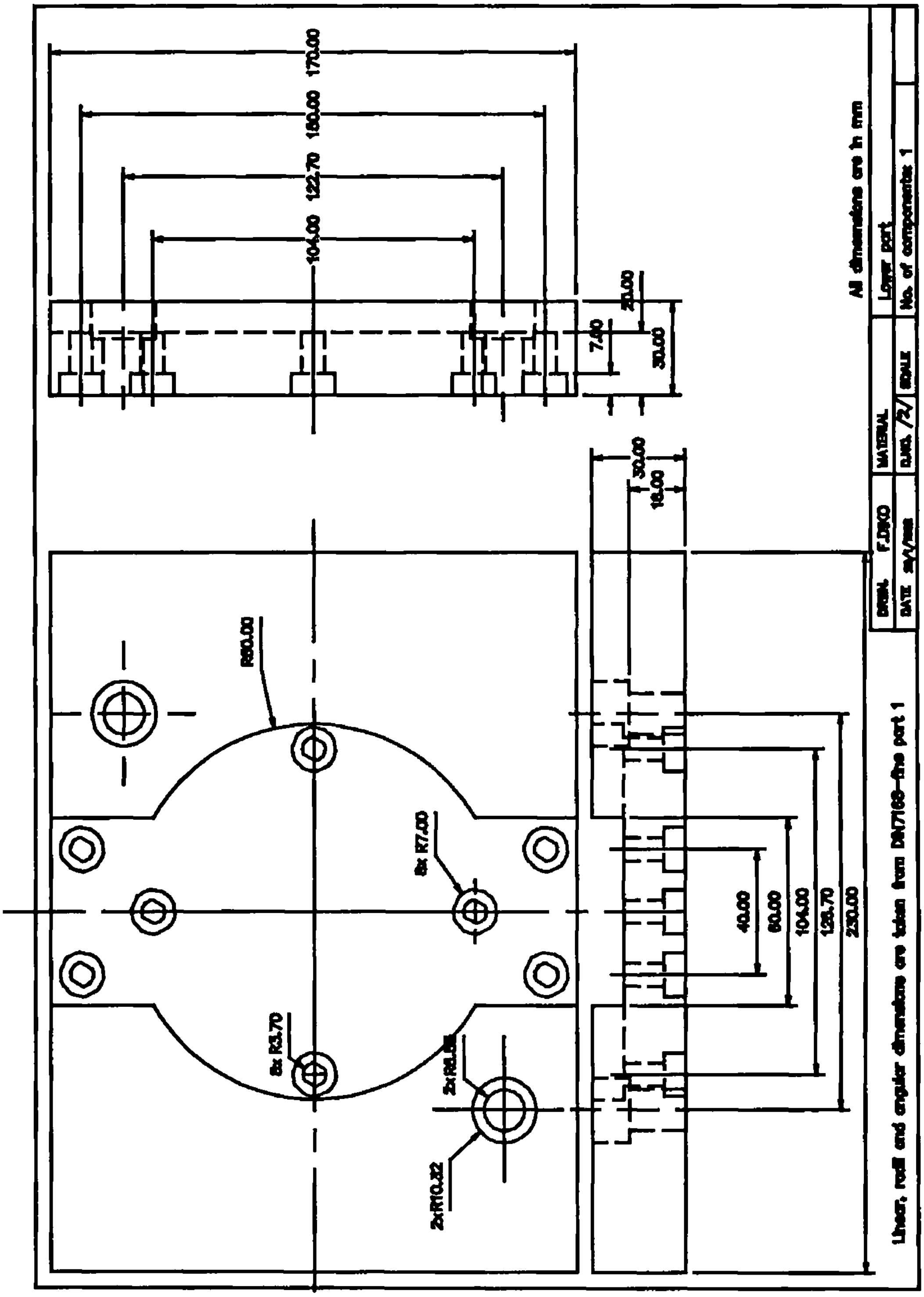


Fig 6 35 The lower plate

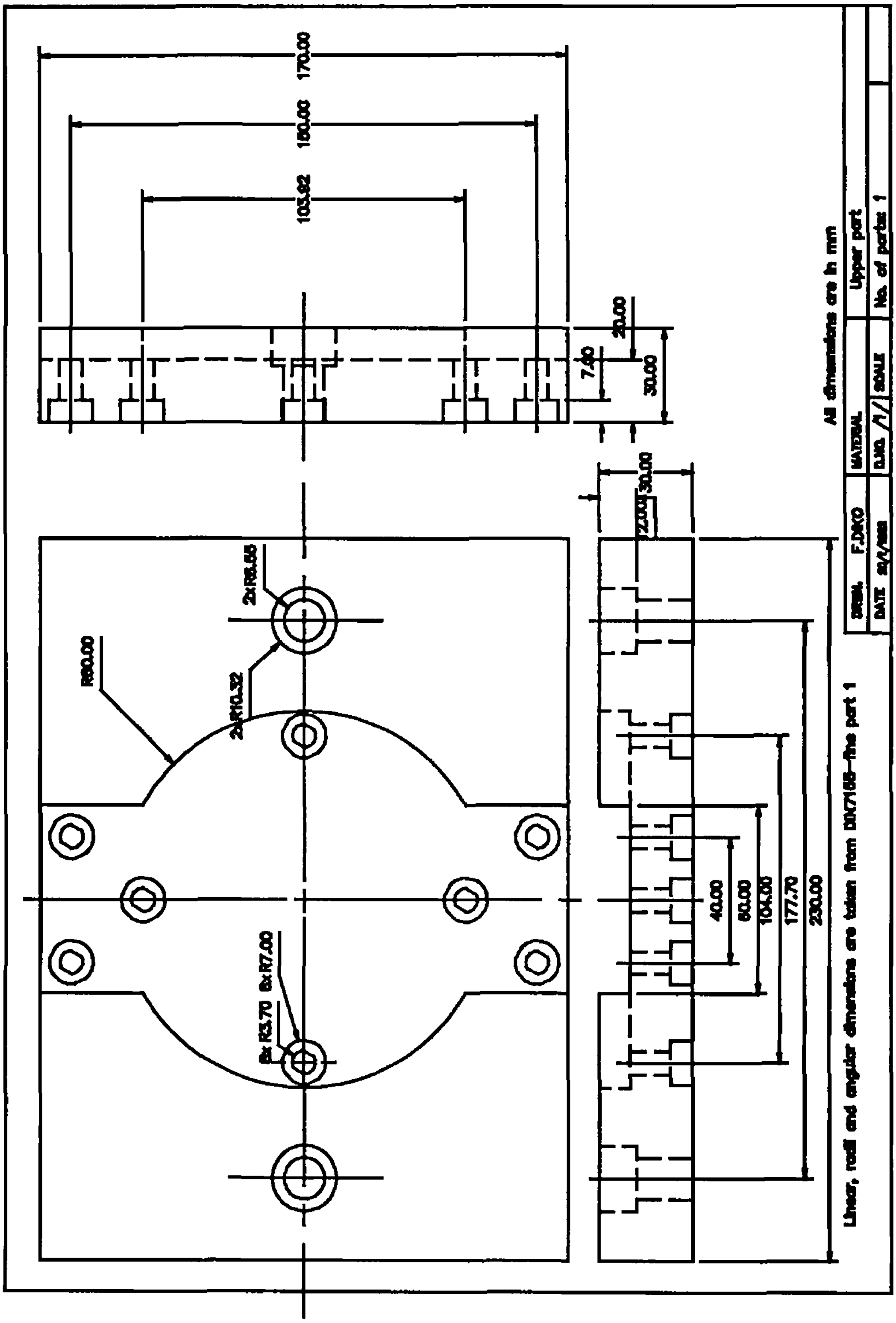


Fig 6 36 Upper plate

CHAPTER SEVEN

EXPERIMENTAL PROCEDURE AND RESULTS

7 1 INTRODUCTION

The local metal flow during a forming process is essentially influenced by

1 Factors related to the material of the workpiece, such as the prior history of deformation, grain size and distribution, dependency of flow stress upon strain, strain-rate, temperature and anisotropy

2 Factors related to tooling such as geometrical shape, lubrication conditions at the tool-working interface and tool temperature

3 Factors related to forming equipment used, such as deformation speed and contact times under load

In cold forming i e , room-temperature forming the equipment behaviour does not significantly influence the metal flow, provided the material is not strain-rate dependent at room temperature and the *friction* conditions do not vary greatly with deformation speed
However, the velocity characteristics of equipment in hot forming greatly influence the metal flow and the deformation process, because most materials are strain-rate dependent in the hot forming range and the friction conditions vary drastically with temperature

Two types of materials have been used in the current forging experiments, lead for the

Two types of materials have been used in the current forging experiments; lead for the plane strain forging and copper for the axisymmetric forging. Experiments to find out the flow stress data and the friction factor are carried out just for copper where for lead these characteristics are taken from the literature [125] because experiments for the same material under the same condition have been carried out before.

7.2 EQUIPMENT AND INSTRUMENTATION

In conducting this study three machines have been used,

1. Instron Testing Machine with load range of up to 50 kN. This machine has been used in conducting the experiments to find out the material characteristics, Plate 7.1.
2. Hydraulic Instron Machine with load range of up to 500 kN. This machine has been used for the forging of the plane strain lead specimens, Plate 7.2.
3. Hydraulic press machine with load range of 1500 kN. This machine has been used for carrying out the axisymmetric closed die forging of the copper billets, Plate 7.3.

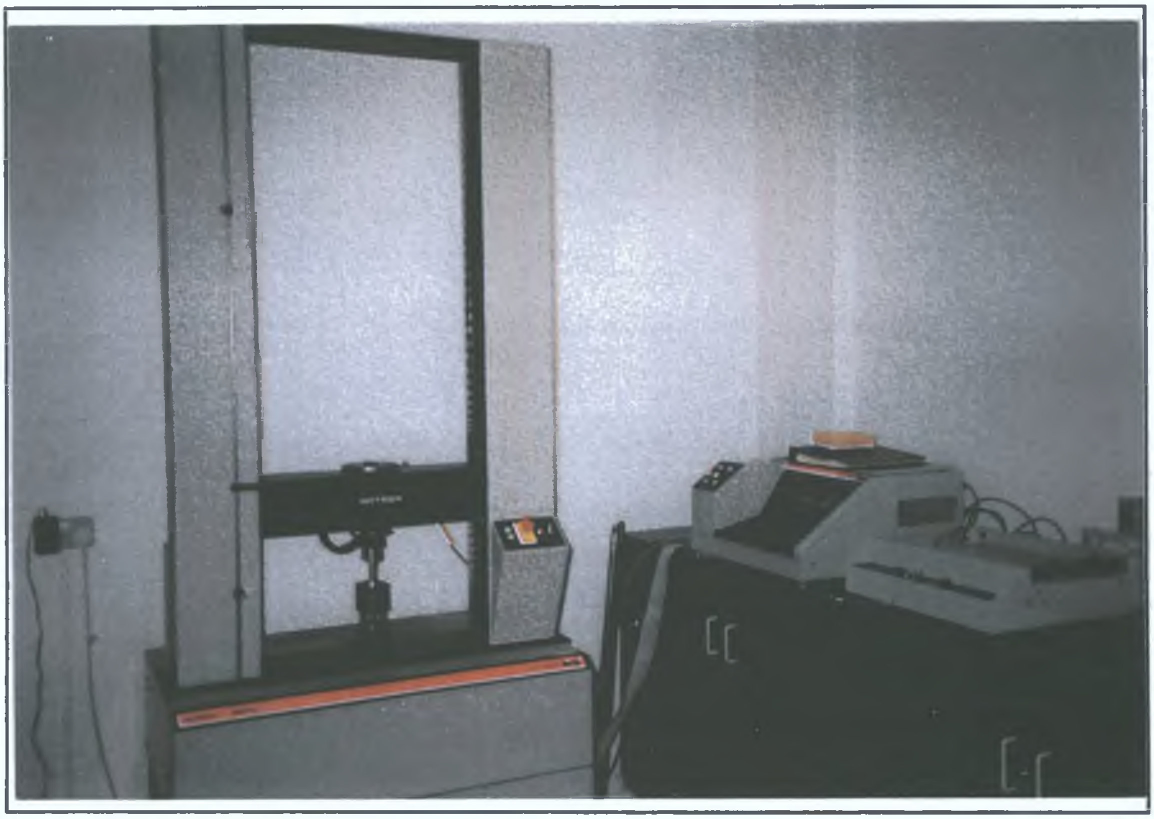


PLATE 7.1 Instron testing machine (50 kN)

PLATE 7.2 Instron machine (500 kN)

PLATE 7.3 Hydraulic press machine (1500 kN)

7.3 DETERMINATION OF THE MATERIAL CHARACTERISTICS

The classic method for determining the flow stress is by a uniform-compression test (without barrelling) or by a torsion test at temperatures and strain-rates of interest. The compression test is usually conducted in a plastometer so that constant strain- rate is maintained throughout the test [127-130].

The friction factor, or the friction coefficient, is most commonly obtained by a ring test [72,131]. In this test, a flat ring-shaped specimen is upset forged to a known reduction. The change in internal diameter, produced by a given amount of reduction in height, is directly related to the friction conditions at the material-tool interface.

In hot forming, the die temperature usually is lower than the billet temperature. The resulting die chilling influences the frictional conditions, and it is included in the measurement of the friction factor by using the ring test at hot-forging temperature Die chilling, however, also influences the temperature of the deforming billet and, consequently, its flow stress It is, therefore, difficult to estimate the actual flow stress, σ, the friction factor, f, or the shear factor, m, under practical forging conditions

Barrelling is prevented by using adequate lubrication, for instance graphite in oil for aluminum alloys, glass for steel, titanium and high temperature alloys
The load and displacement or sample height are measured during the test and thus, the flow stress is obtained at each stage of deformation or for increasing strain
In analyzing metal forming problems, it is useful to define the magnitude of deformation in terms of "logarithmic" strain In the uniform compression test,

$$\epsilon = \int \frac{dh}{h} = \ln\left(\frac{h_0}{h_1}\right) \tag{7.1}$$

The strain rate, ε,is the derivative of strain, ε, with respect to time or

$$\epsilon = \frac{d\epsilon}{dt} = \frac{dh}{h\,dt} = \frac{V}{h} \tag{7.2}$$

where h_0, initial sample height in the compression test

h_1, final height in the compression test

V , instantaneous ram speed

h , the current height

7.3 1 REPRESENTATION OF FLOW STRESS DATA

At room temperature, the flow stress of most metals is strain dependent It was empirically found that the strain dependency of the flow stress can be represented as,

$$\bar{\sigma} = K\,\bar{\epsilon}^{n} \tag{7.4}$$

where K and n are constants expressing strain hardening
$\bar{\sigma}$ and $\bar{\varepsilon}$ are effective stress and effective strain
At higher temperature, above the recrystallization temperature, the flow stress is influenced mainly by the strain rate, and it can be approximated as,

$$\bar{\sigma} = K \bar{\epsilon} \tag{7.5}$$

Specımens have been prepaied ın a cylındrıcal shape wıth 10 mm heıght and 10 mm dıameter To prevent bulgıng, thın Polythene sheet has been used as lubrıcant and the lubrıcatıon has been renewed durıng the process of upsettıng The load dısplacement curves have been plotted as shown ın Fıg 7 1 fıom whıch the stress-straın curves have been produced The dısplacements have been changed to straın by Eq 7 1 where h1 ıs the dıfference between the ınıtıal heıght of the workpıece and the dısplacement

The straın rate whıch ıs the derıvatıon of straın has been calculated usıng Eq 7 2

The ram velocıty used ın the test ıs V= 5 mm/s whıch leads to an average straın rate of 0 5 1/s

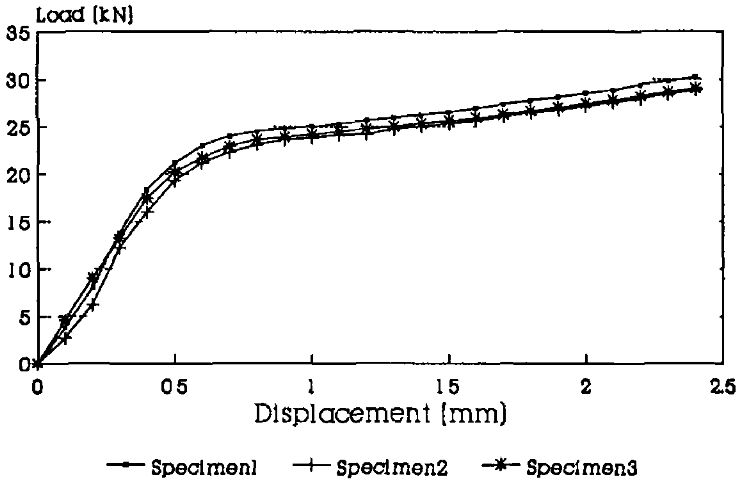


Fıg 7 1 Load dısplacement curves

After plottıng the stress-straın curves fıom three experıments, the average curve has been determıned and a theoretıcal curve has been produced as shown ın Fıg 7 2 The expressıon of the straın dependency of the flow stress ıs expressed as,

$$\bar{\sigma} = 318\ 12\ \bar{\epsilon}^{0\ 066} \tag{7.6}$$

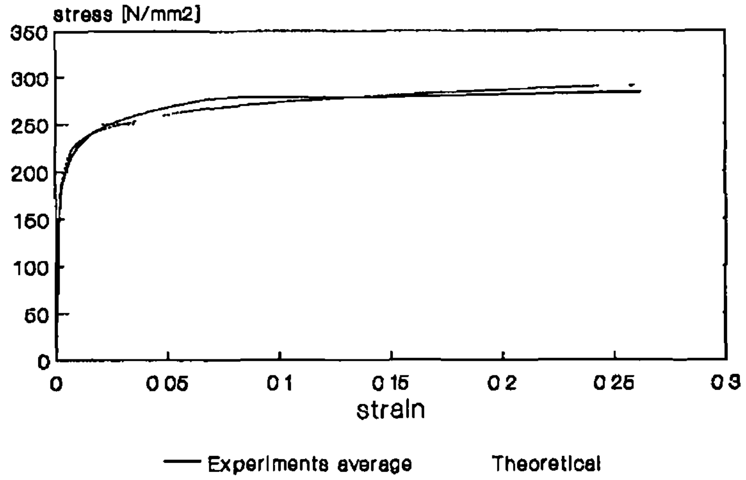


Fig. 7 2 Stress stram curve

7 4 DETERMINATION OF THE COEFFICIENT OF FRICTION

The most common method used for studymg the fnctional behaviour of metals under condtions of bulk plastic defoimation mvolves a simple forgmg operation carned out m a flat nng-shaped specimen, the coefficient of fiiction is related to the change in diameter produced by a given amount of compiession m the thickness direction The internal diameter increases if m is small and decreases if m is large A disadvantage of the method is that a satisfactory theoietical analysis of the compression of a nng is not yet available, so that numerical values of m can be obtained only by an independent calibration method Theoretical studies [132] suggested that maximum accuracy in the determination could be obtained by using a nng of small height and large internal diameter as compared with external diameter

Too large an internal diameter, however, unless coupled with an excessively small height, would make the deformation unstable and the nng would tend to buckle at low values of fnction

7 4 1 EXPERIMENTAL RESULT

Copper nngs of 6 3 2 propoition (O D 18 mm I D 9 mm Height 6 mm) have been machined and prepared foi the fnction test After upsetting the nngs, their dimensions were measured and the fnction sheai factor, m, was determined for each sample using the calıbratıon curves gıven ın Fıg 7 3 These curves were derıved through computer program, based on upper-bound method of analysıs, whıch sımulates the compressıon of a rıng wıth bulgıng at constant frıctıon [131,133]

The lubrıcant used ın thıs experıments was Rocal Tufdraw 3040, whıch ıs an ındustrıal product for cold forgıng The frıctıon factor was found to be 0 052

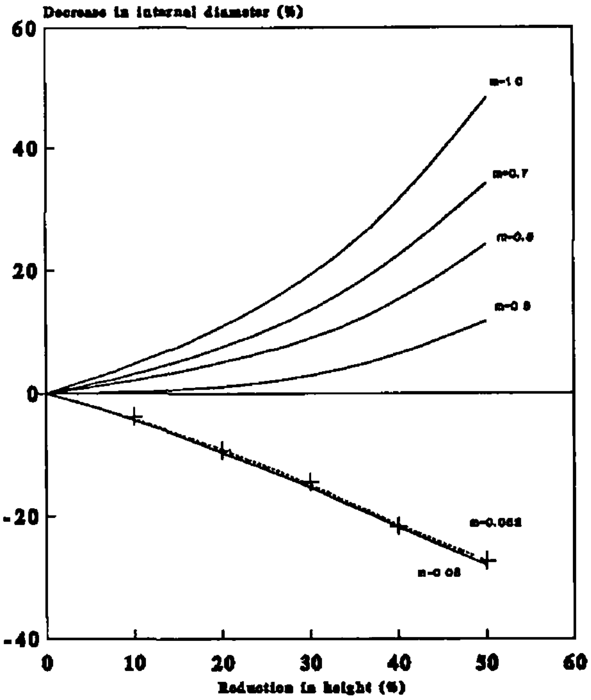


Fıg 7 3 Calıbratıon curves (6 3 2)

7 5 PLANE STRAIN CLOSED DIE FORGING EXPERIMENTS

The dıe set for these experıments consısts of four gıoup of components,

1 The two halves of the dıe whıch have the same shape because of the symmetry of the component to be forged wıth, as shown ın Fıg 5 53
2 Two plates for the placement of the two halves of the dıe on the press machıne as shown ın Fıg 6 35 and Fıg 6 36
3 A component wıth H cross-sectıon to alıgn both dıe halves when ınstallıng the dıe on the machıne as shown ın Fıg 7 4
4 Two L-shaped components to place the bıllets ınsıde the dıe cavıty at the exact posıtıon and along the centre lıne of the dıe as shown ın Fıg 7 5

The experıments have been carrıed out under two frıctıonal condıtıons [125],

- wıth lubrıcant, m = 0 035, usıng Rocal Tufdraw 3040
- hıgh frıctıon, m = 0 3

The bıllets have been machıned wıth the same dımensıons whıch have been used ın the fınıte element sımulatıon as pıesented ın chapter five

Plate 7 4 shows a vıew of the dıe, bıllet and the forgıng

Forgıngs wıth dıfferent reductıon ın heıght have been produced and sectıons for these forgıngs have been prepared to be compaıed wıth those produced by the fınıte element program

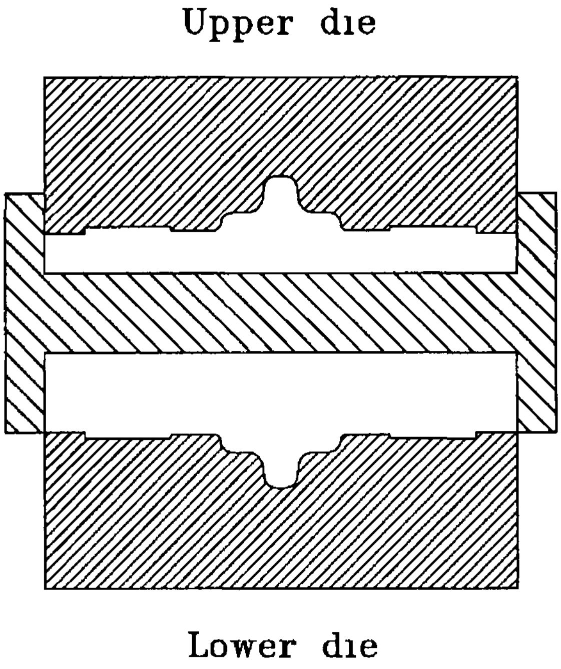


Fıg 7 4 H-shaped component for dıe alıgnment

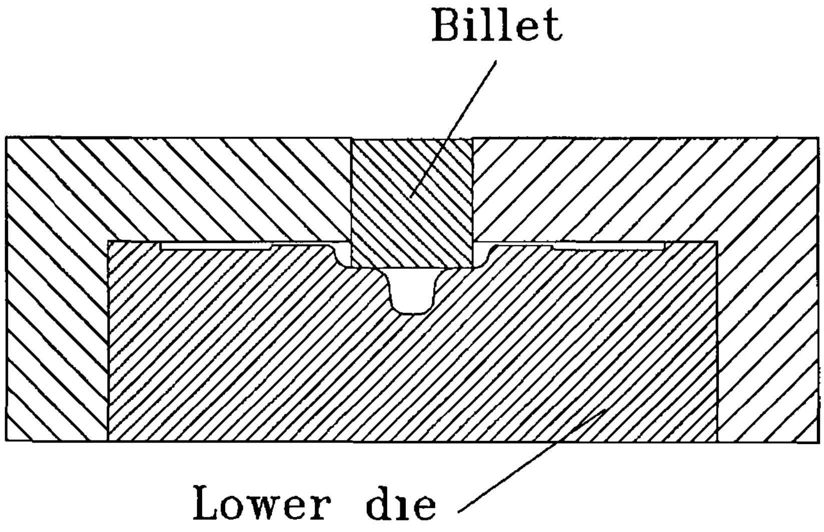


Fıg 7 5 Two L-shaped components for bıllet placement

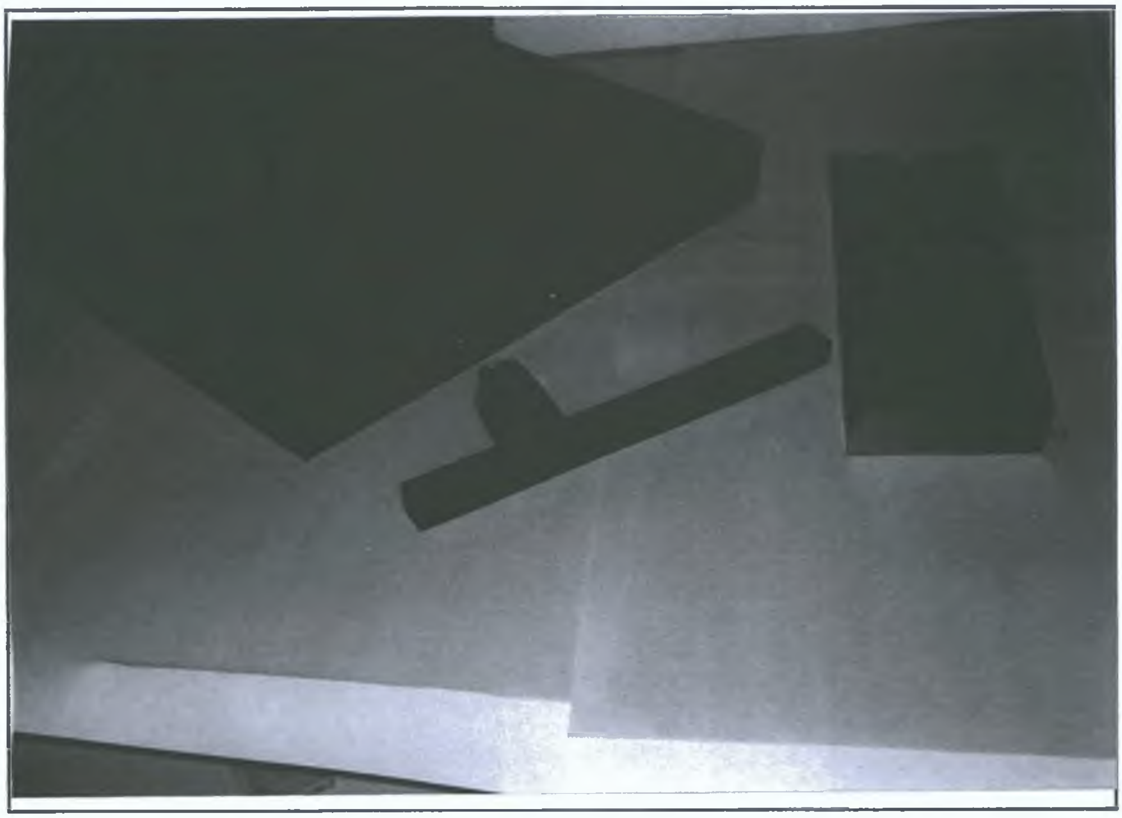


PLATE 7.4 A view of the die, billet and the forging.

7.5.1 WITH LUBRICANT (m = 0.035)

Fig. 7.6 shows the cross-section of the billet at four stages of deformation. This cross-sections are taken from both the finite element simulation program and the experiments. It is clear that the predicted and the experimental profiles , for the case with m=0.035, are in a good agreement. The material starts to flow sideways towards the die corners creating a small concave surface at both vertical sides of the billet. At 38.8% reduction this material has reached the sides and the material in the middle starts to flow horizontally towards the flash land. At 48.8% reduction, the material starts to flow through the die cavity making sure that the die is filled.

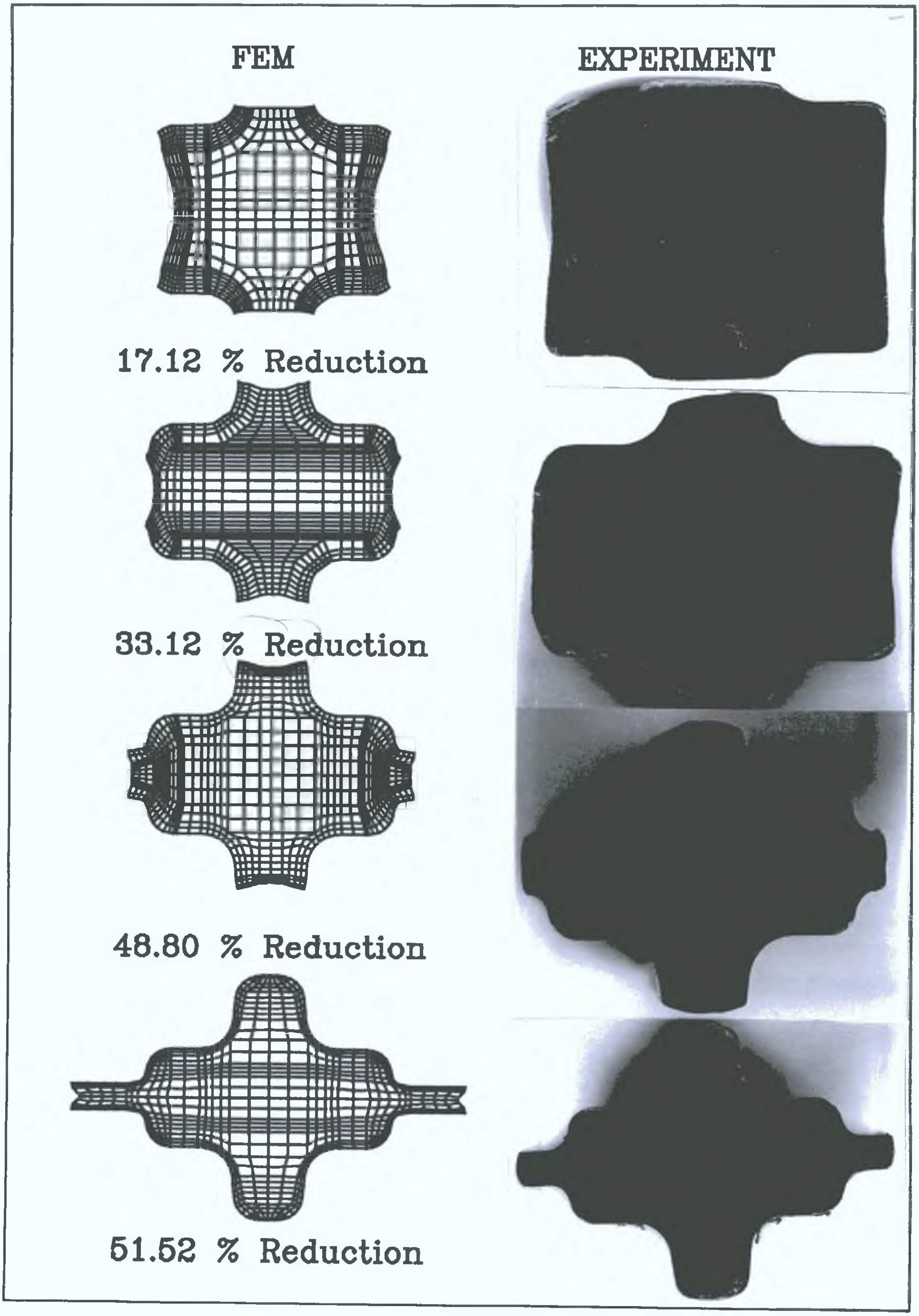


Fig. 7.6 Experimental and FE results for m=0.035

The load-displacement curves for both the experimental and theoretical results are shown in Fig 7 7 The load increases steadily for both cases until the beginning of the flash formation, after which it starts increasing rather sharply due to the increase in the pressure at the flash region This pressure at the flash region causes the die to be filled with the material which finds it easier way to fill the die than flow through the flash land

It is clear from this figure that the curves are close enough to be considered acceptable After the specified amount of reduction in height, further increase in the load will not affect the die filling

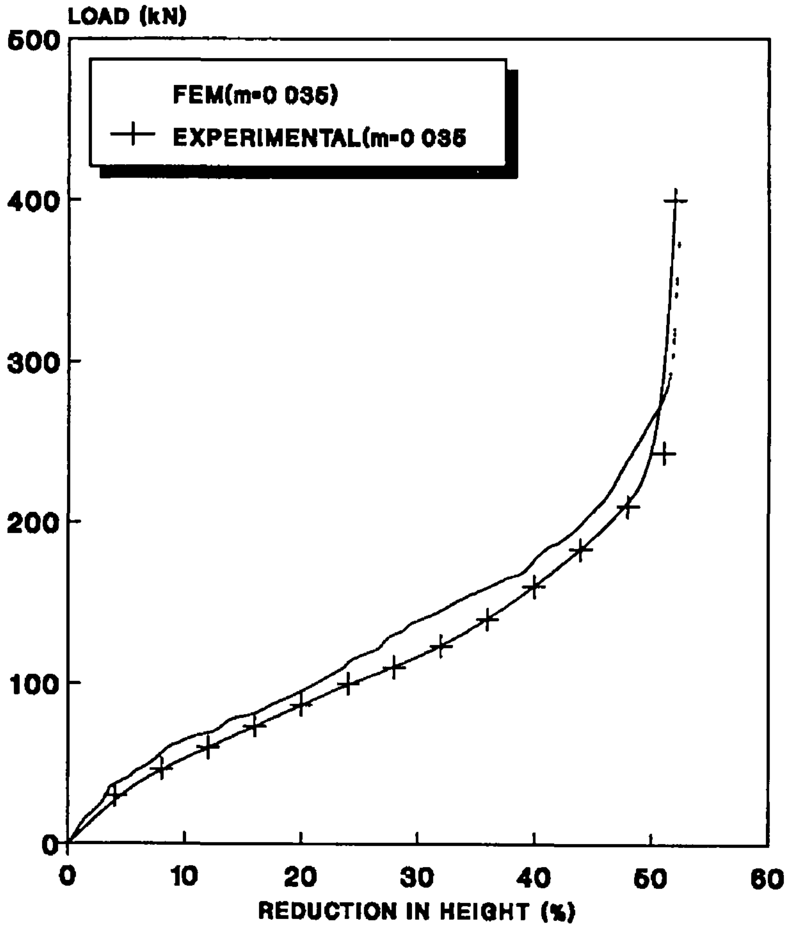


Fig 7 7 Load-Displacement curves (m=0 035)

7 5 2 HIGH FRICTION (m = 0 3)

Fig 7 8 shows four stages of deformation in which remeshing was needed in the FE simulation Experiments have been carried out under the same forging condition but under high friction conditions where no lubricant has been used, m=0 3 There are agreement between the theoretical and experimental results of three stages In the second stage, there is some differences along the slide of the billet which can be related to the coarse mesh at this region This is a good example of the effect of the mesh system on the simulation process A compromise should be made in using a fine mesh in which the computational time is higher and the accuracy of the solution is better The accuracy of the solution increases rapidly till a certain stage after which any further refinement of the mesh will cause only a small increase in the accuracy which can not justify the high cost of the computing

Fig 7 9 shows the load-displacement curves of the forging process under high lubrication condition The agreement of the experimental curve with the one produced by the finite element program are reasonable Comparing this figure with Fig 7 7 which has been produced with the presence of lubricant, it is clear that the forging load needed, without using the lubricant, is higher than that needed when forging under lubrication conditions This behaviour is natural because when using the lubricant the metal resistance to flow and the friction is less and subsequently it will need less load to reach the same amount of reduction

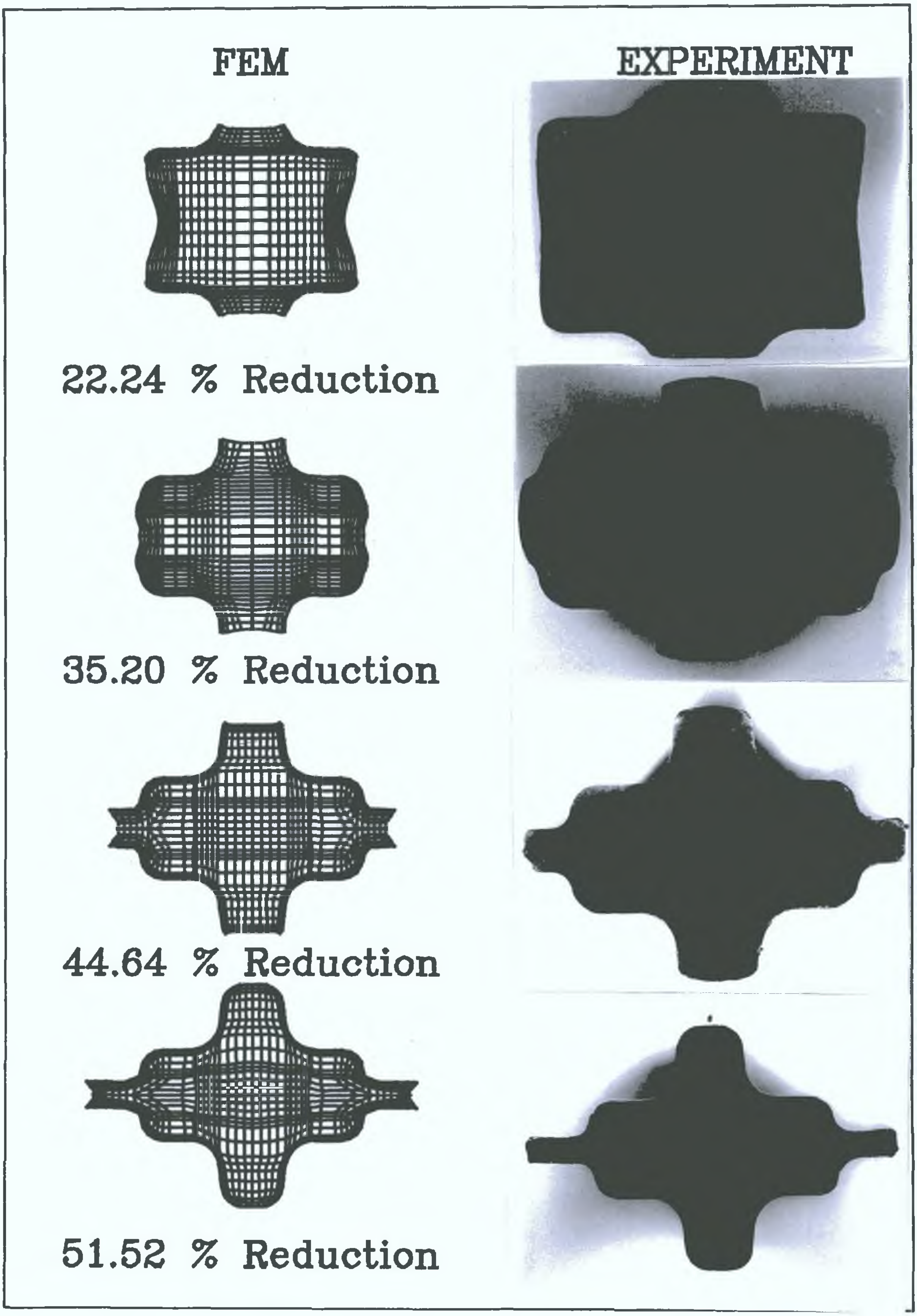


Fig. 7.8 Experimental and FE results for high friction m=0.3

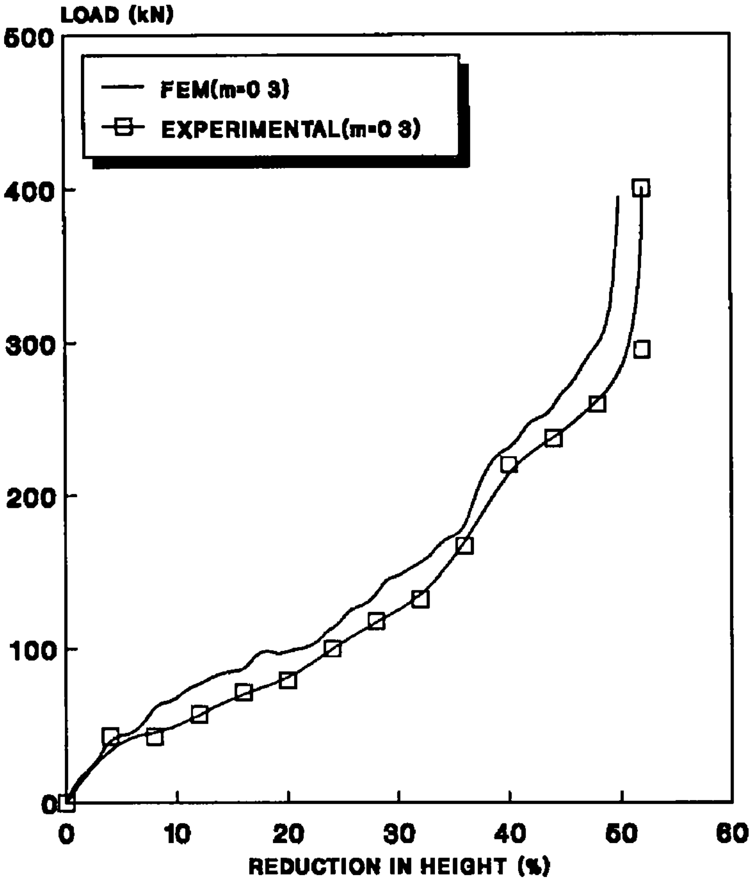


Fig 7 9 Load-Displacement curves for high friction

Examining the forgings produced under both conditions of lubrication, it is found that at both ends of the forgings the material did not completely fill the top and bottom orifices of the die as shown in Plate 7 5 The reason for this behaviour is suggested to be that during the deformation process the material at the end of the billets has three optional routes to flow through These routes are either to flow through the orifice or through the open die ends or, finally, through the flash at the final stage of deformation It is known that during any forming process the material flows through the easiest route in which less resistance exists In these experiments the easiest route for the material at both ends was to flow along the die centre line At the early stages of deformation the force needed for the material to fill the central cavity is less than that needed for the material to flow along the centre line of the die However, when the deformation process proceeded and the material started to flow through the orifice at the ends of the billet, the material flows along the central line due to the high pressure at the orifice At the last stage of deformation and when the flash started to be formed, this phenomenon was still continuing due to the high pressure in the flash region as well as in the orifice region. Although the pressure at the flash region was higher than that at the orifice region, this does not change the deformation mode and the die cavity is partially filled with the material.

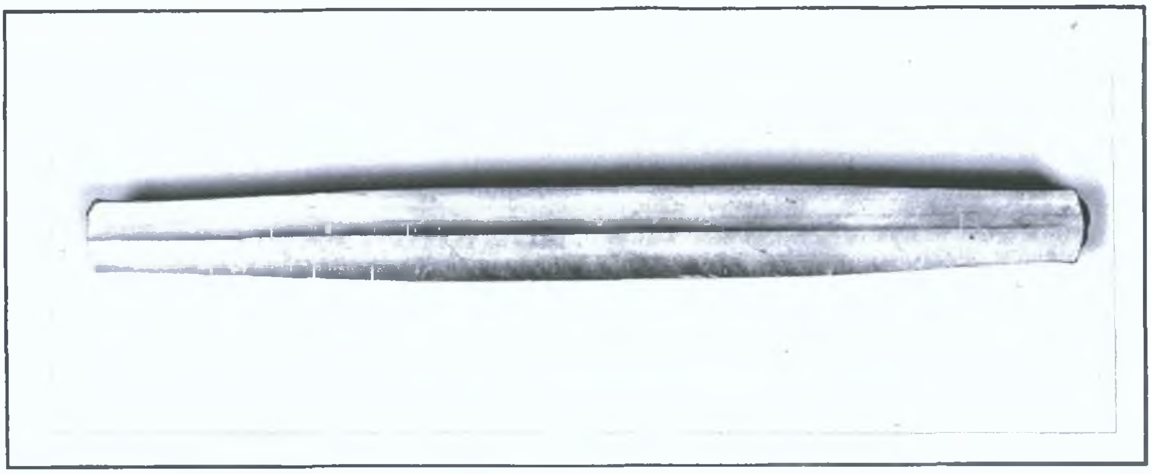


PLATE 7.5 Across-section along the forging in the direction of the forging load

To investigate this phenomenon, both ends of the die have been closed and the experiments have been carried out for the case with lubricant. This modification of the die does not affect the case of plane strain because in the actual forging condition with complex shaped components, critical cross sections are taken from the component. In most cases the plane strain piece of the component is located between two other parts and does not have free ends. In general there should not be too much difference between the two cases but here in this example the special geometry of the cavity caused this phenomenon.

Plate 7.6 shows a view of the die after closing both ends. Two pieces of lead with the same cross section of the billet are placed at both ends of the billet to fill the gap between the billet and the two end plates.

Plate 7.7 shows a cross section along the forging length. Comparing this section with the one shown in Plate 7.5, it can be noticed that the filling of the orifice at both ends is significantly improved when using the closed ended die.

Plate 7.8 shows a cross section of two components produced by the open and closed ended dies. The formation of the flash in the closed ended die is homogeneous in contrast with the open ended one, where the flash land reduces gradually from the middle towards the ends.

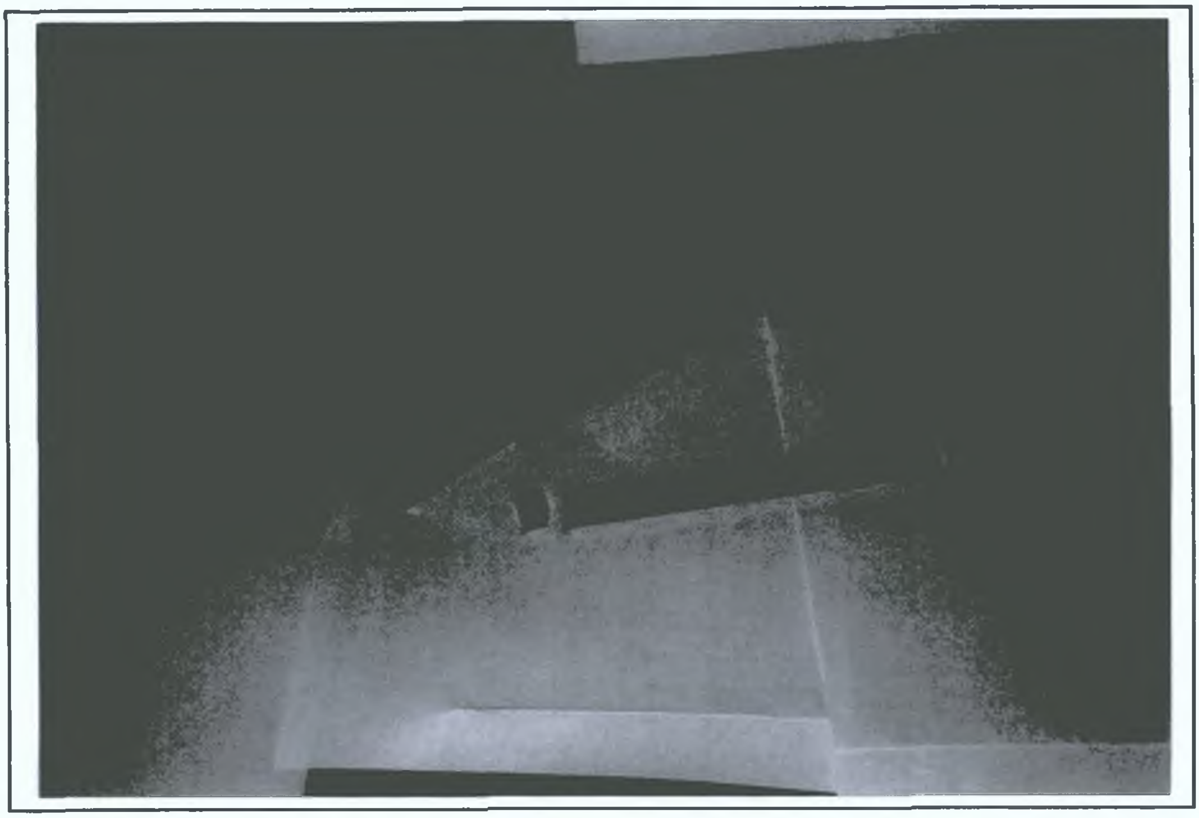


PLATE 7.6 A view of the closed die

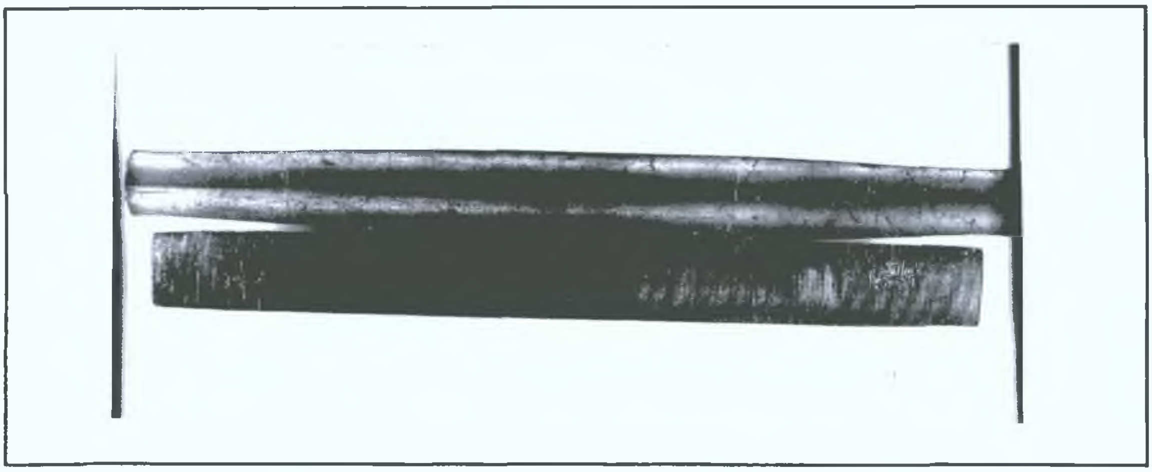


PLATE 7.7 Across-section along the forging length for closed end die

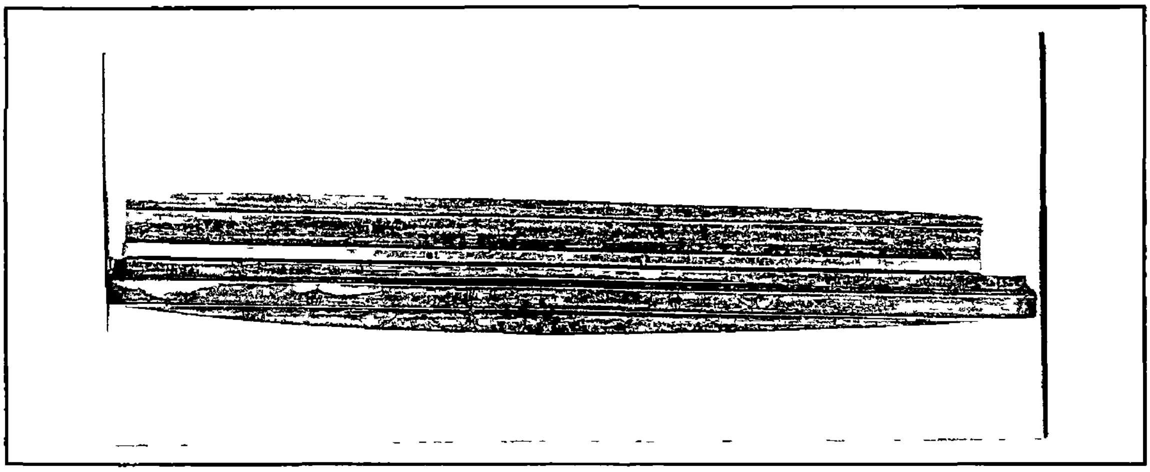


PLATE 7 8 A vıew of the flash formatıon for both cases

Results of the closed ended dıe forgıng are pıesented ın Fıg 7 10 and Fıg 7 11 In Fıg 7 10, the profile of the cross sectıons of both the FE sımulatıon and the expenments, usıng lubncant, are presented It ıs clear from thıs fıgure that the expenmental results are ın good agreement wıth those produced by the FE sımulatıon Fıg 7 11 shows the load dısplacement curves accordıng to the FE sımulatıon and fıom both, the open and closed ended dıe expenments The general trend of the curve produced by the closed ended dıe ıs almost the same as the one produced by the open ended dıe Only the magnıtude of the load ıs hıgher, whıch ıs due to the extra load needed for the matenal to flow through the onfice and the flash at both ends of the bıllet It ıs also clear that the curve for the closed ended dıe ıs much closer to the FE sımulatıon curve whıch ındıcates that ın closıng both ends of the dıe the matenal flow ıs much closer to the plane straın condıtıon

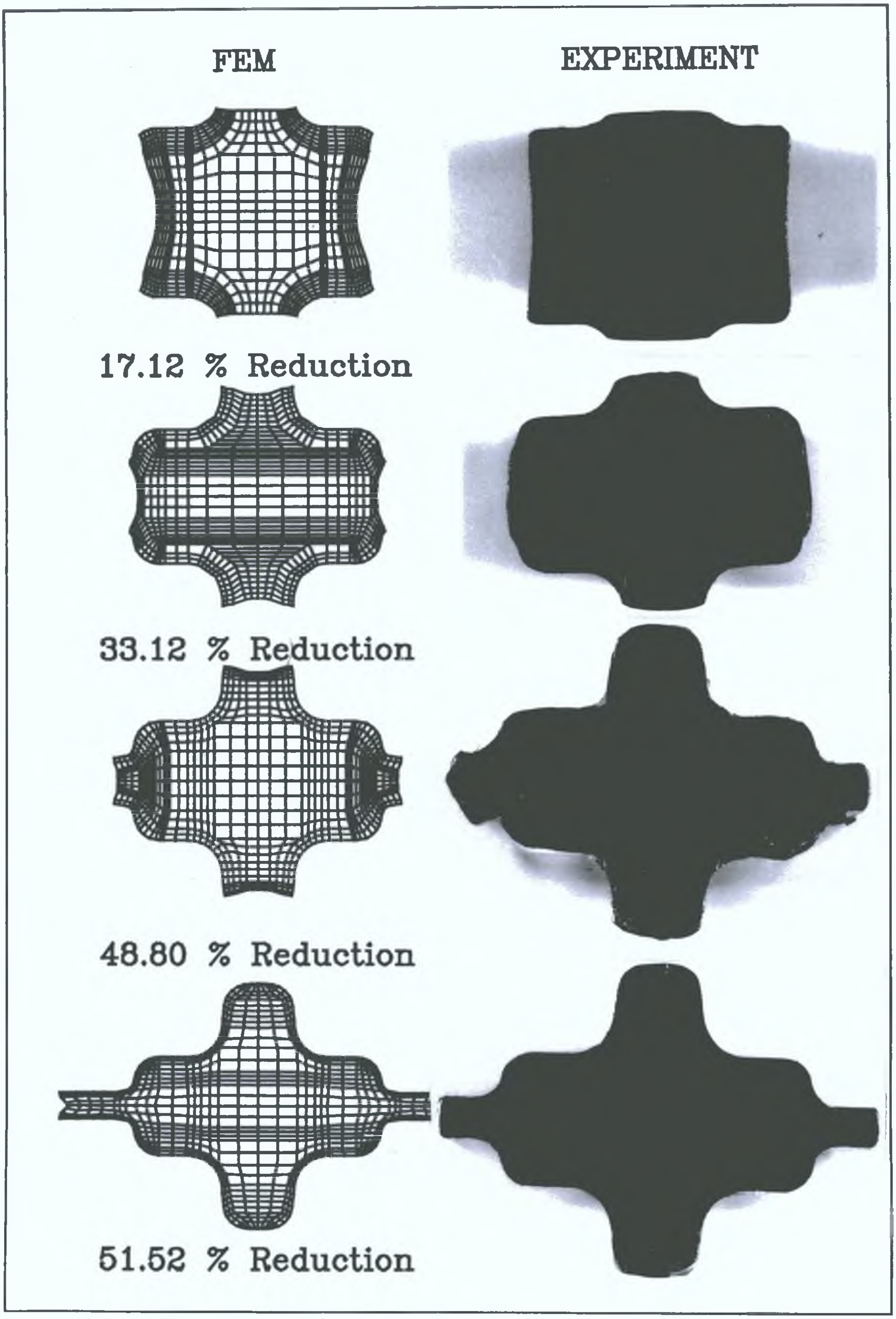


Fig. 7.10 Experimental and FE results for closed ended die, m=0.03

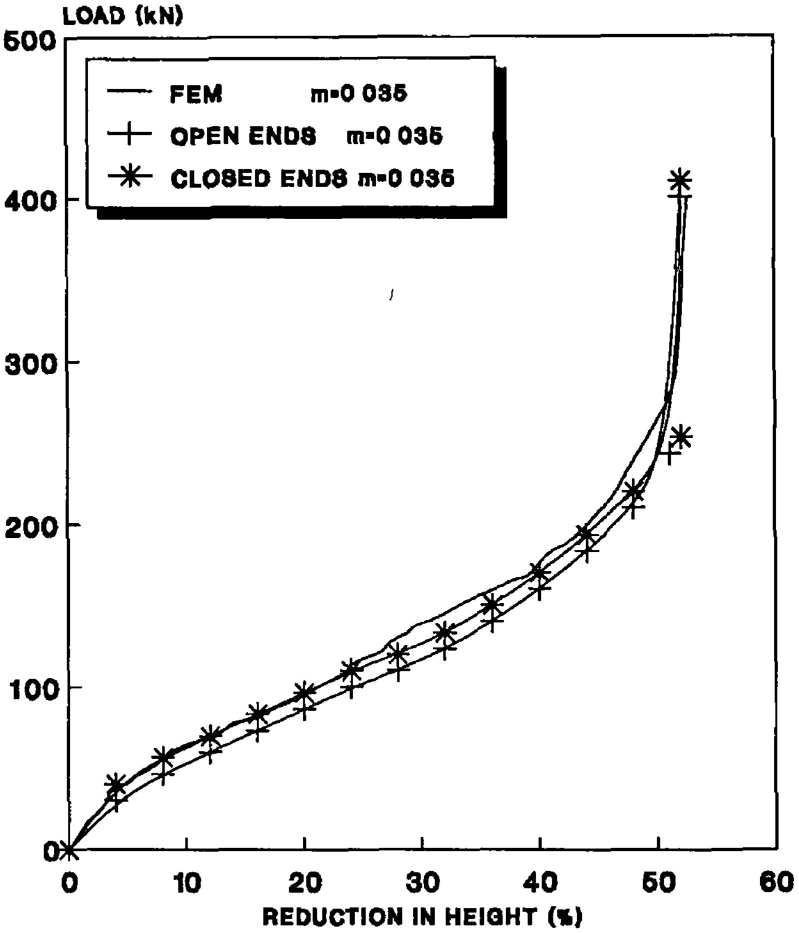


Fig 7 11 Load-Displacement curves

7 6 AXISYMMETRIC CLOSED DIE FORGING EXPERIMENTS

The die set for these experiments consists of four group of components,

1 The two halves of the die which have the same shape because of the symmetry of the component to be forged with, as shown in Fig 6 33

2 Two plates for the placement of the two halves of the die on the press machine as shown in Fig 6 35 and Fig 6 36

3 A cylindrical component with cavities on both side to align both die halves when installing the die on the machine as shown in Fig 7 12

4 Two semi-circular components to place the billets inside the die cavity at the exact position and in the middle of the die cavity as shown in Fig 7 13

The expenments have been carned out undei fnctional conditions with the fnction factor taken as m = 0 052 The billets have been machined to the same dimensions which have been used in the finite element simulation as presented in chapter six

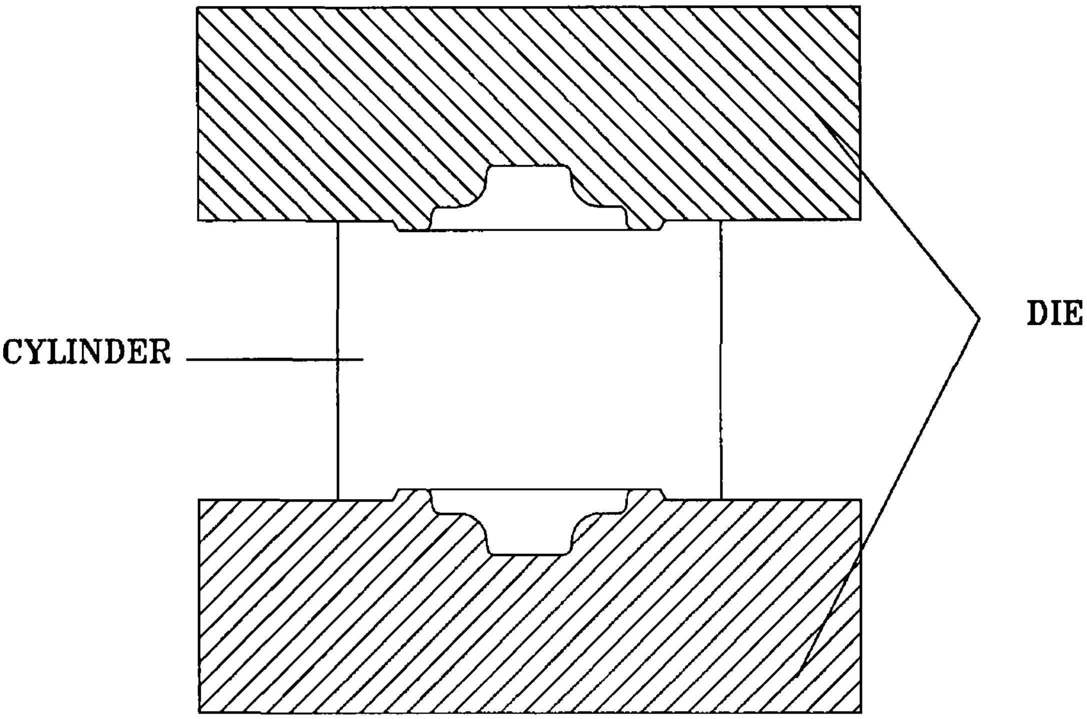


Fig 7 12 A cylindrical component for die alignment

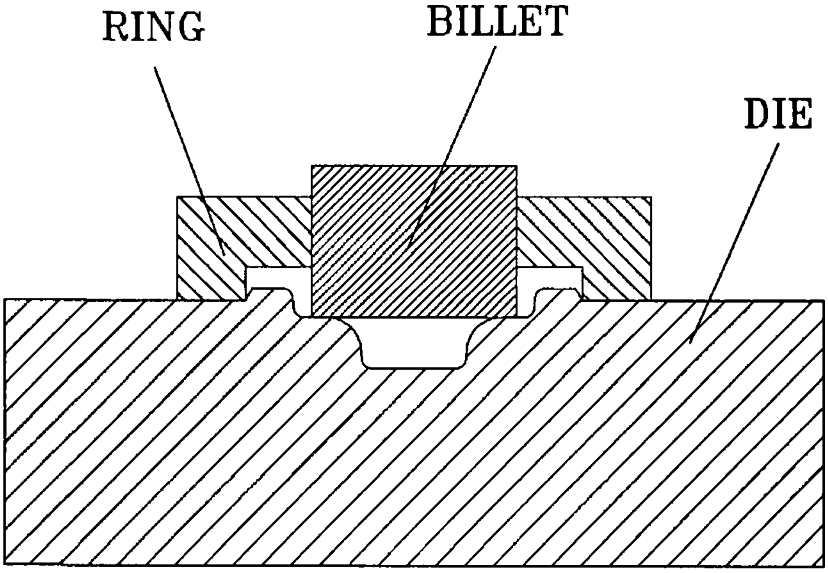


Fig 7 13 A ring shaped component for billet placement

Plate 7 6 shows the die set with the billet and the forging The forging expenments are

PLATE 7.6 The die set ,billet and the forging

carried out under the specified conditions. However, during the final forging test when maximum compression was attempted the upper half of the original die broke along a line close to the centre line of the die as shown in Plate 7.7. This failure has been analyzed and all the possible factors which might have caused this failure have been discussed as follows,

1. Mechanical design

On the basis of the investigation of several thousand tool failure [134] it has been found that two simple factors are most frequently responsible for design failure, either singly or together. These are,

- the improper control of sharp corners.
- the use of extreme section change.

The first factor cannot be the cause of this failure in the present case because all sharp edges have been eliminated and replaced by proper corners and fillets. The second factor is also excluded because there is not much drastic changes in the die section and usually this failure takes place during the hardening process or under light service loads. This failure is likely to have happened due to the internal stresses which appears when tools containing such sections are liquid quenched

2 Machining procedure

This factor can also be excluded because the die dimensions were according to the drawing provided to the manufacturer and no sharp corners exist Also good finishing for the die surfaces is obtained which eliminates the possibility of hidden machining defects

3 Heat-treatment

In a majority of die failures, some faulty heat-treatment practice is found to be responsible Because the heat-treatment for this die has been carried out by manufacturers external to the research place, the possibility of improper heat-treatment does exist and tests should be carries out to make sure that the heat-treatment was properly done

4 Handling and use of the die in service

This title includes the overloading by accident and improper alignment of the dies The two halves of the die have never touched each other and the thickness of the flash land does not reach the target which will exclude the possibility of over loading The improper alignment of the dies is believed to be right and there was some evidence of the misalignment of the billet within the die cavity This misplacement of the billet might have contributed to the die failure

5 Lubrication

The viscosity of the lubricant used in this process is low which caused the lubricant to accumulate in the lower die The evidence of that is shown in Plate 7 8, where it is clear that the distribution of the lubricant is inhomogeneous between the upper and the lower die The material flow through the lower orifice is much greater than the material which flowed into the upper orifice This situation increased the friction forces between the upper die and the material which might have caused the die failure On the other hand, in the FE simulation the lubricant distribution was considered to be the same foi the upper and lower die To solve this problem a thicker lubricant should be used which can stick to the die surface and does not accumulate to the lower die

Going through all these factors it is found that the non-uniform lubrication has the maximum contribution to the die failure followed by the misplacement of the billet within the cavity

In order to continue the experiments a new die was manufactured as an upper die This die has been made of two pieces, an insert and a die case The insert which was press fitted in the die case is made of tool steel D2 and the die case is made of H13

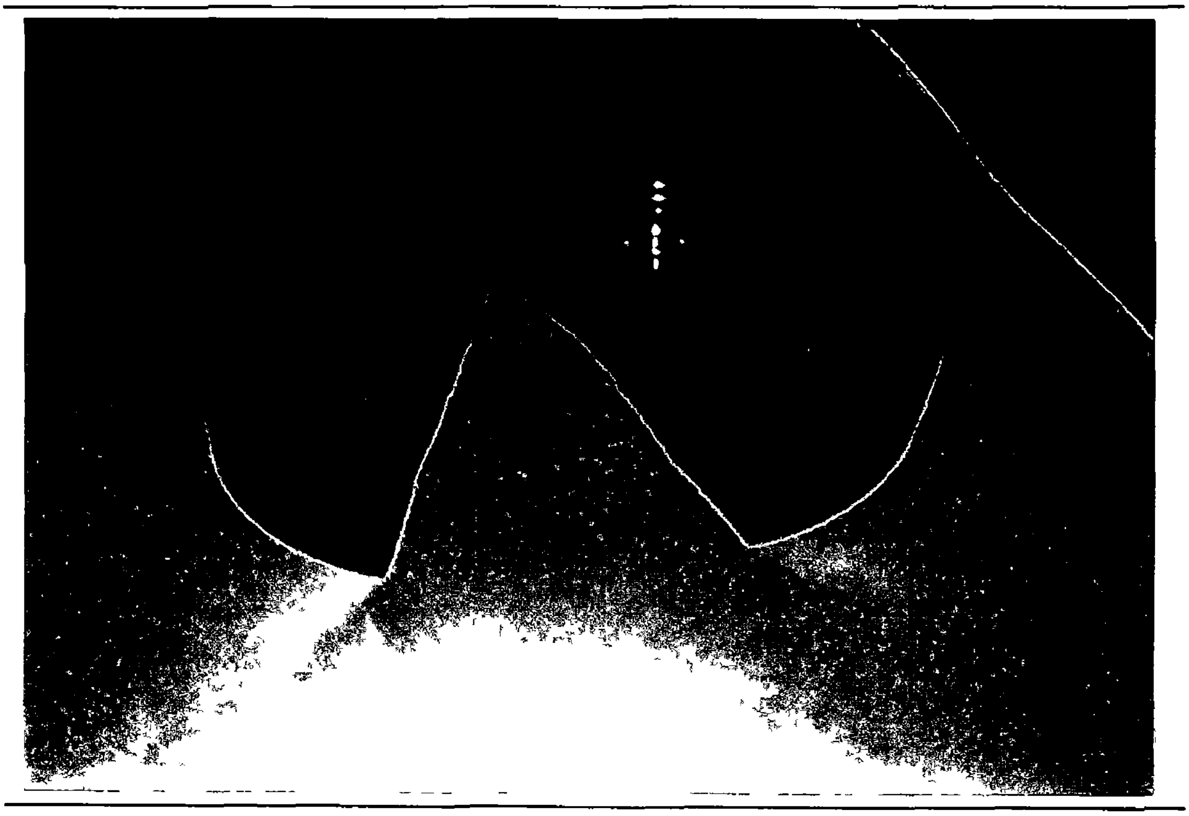


PLATE 7 7 The die breakage

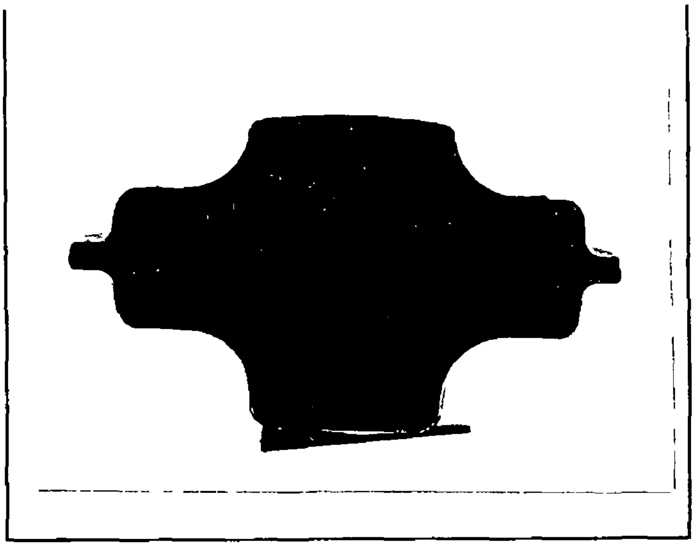


PLATE 7 8 A cross sectıon of the forgıng ȷust before the dıe faılure

Forgıng expenments have been carned out usıng the new dıe and pure petroleum ȷelly wıth thın teflon layers were used as lubncant However, the new dıe also cracked ȷust before the fınal stage of the foıgıng process The trend of the crack was the same as the fırst breakage whıch ındıcates that the reason behınd the dıe faılure ıs not maınly because of the dıfference of the lubıicant dıstnbutıon between the upper and lower dıe halves In fact thıs ınhomogeneıty could not have contnbuted to the dıe faılure because ın the second case the top and the bottom boss heıghts of the forgıng are equal whıch ındıcates that the lubncatıon ınsıde the dıe cavıty was homogeneous Because the maın reason behınd thıs faılure was stıll unknown ıt was necessary to check whether there was any tensıle stress ın the dıe The dıe ınsert, subȷected to dıfferent levels of radıal stress due to press fıttıng has been analyzed and the overall stress dıstnbutıon ın the ınsert under current forgıng condıtıon has been plotted The magnıtude of the external radıal stress has been selected as 10,20,30 and 40% of the dıe-matenal yıeld stress The dıstnbutıon of the radıal stress, stıesses ın Z, hoop stress and the equıvalent stresses are shown ın Fıgs 7 14-7 33 From these figures ıt ıs clear that tensıon stress does not exıst and ıncreasıng the radıal load causes substantıal ıncrease ın the compressıon stresses wıthın the dıe whıch ıncrease the possıbılıty of the dıe faılure due to excessıve compressıon stress For example ın Fıgs 7 18-7 21 The maxımum compressıon stress ın the Z dırectıon ıs more than than the yıeld stıess The same thıng can be seen ın Fıg 7 25 where the hoop stress ıs more than the yıeld stress at the center of the dıe

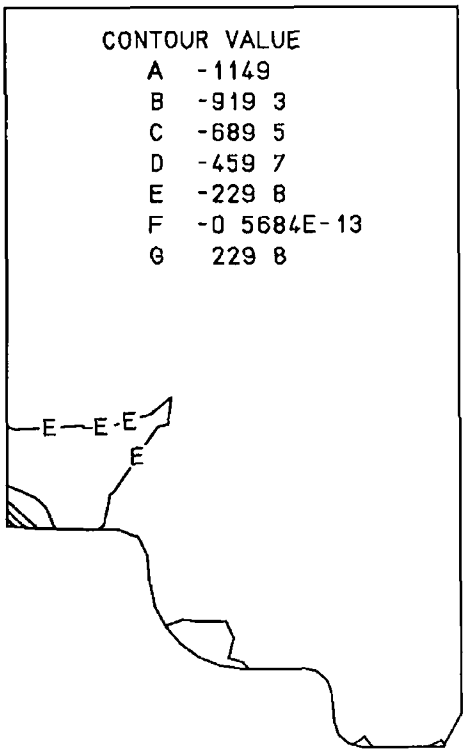


Fig 7 14 Internal radial stress distribution (10% yield)

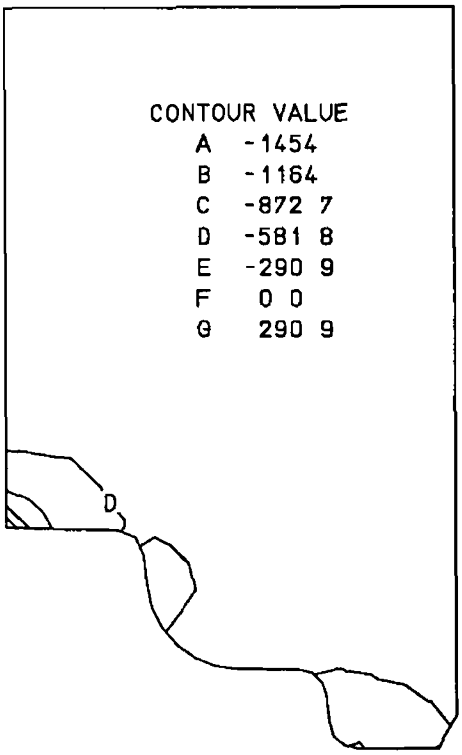


Fig 7 15 Internal radial stress distribution (20% yield)

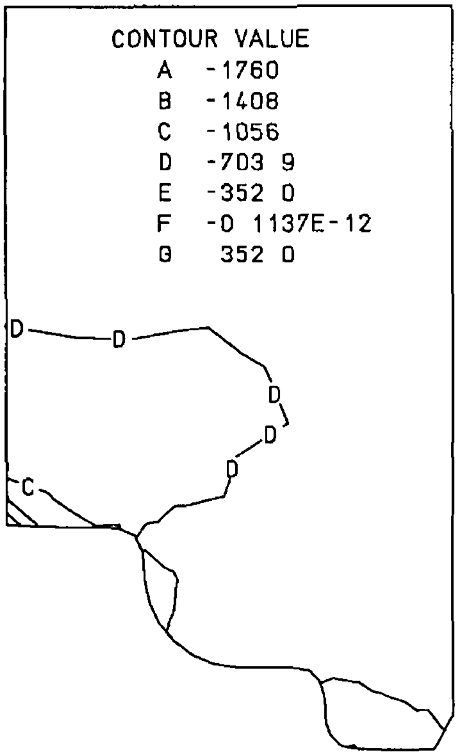


Fig. 7 16 Internal radial stress distribution (30% yield)

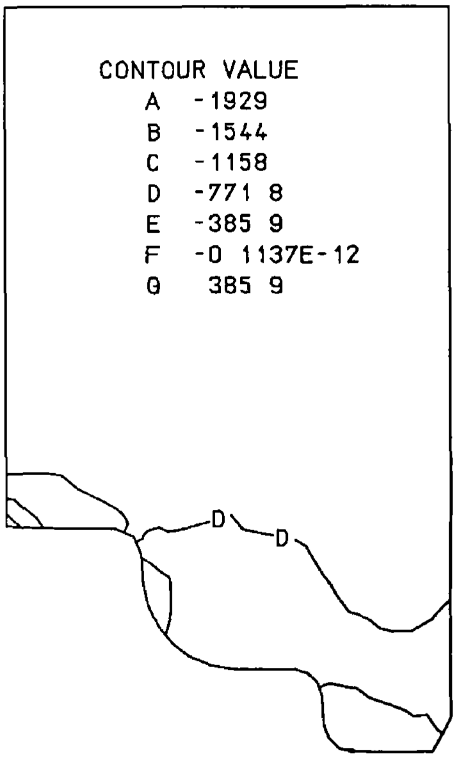


Fig 7 17 Internal radial stress distribution (40% yield)

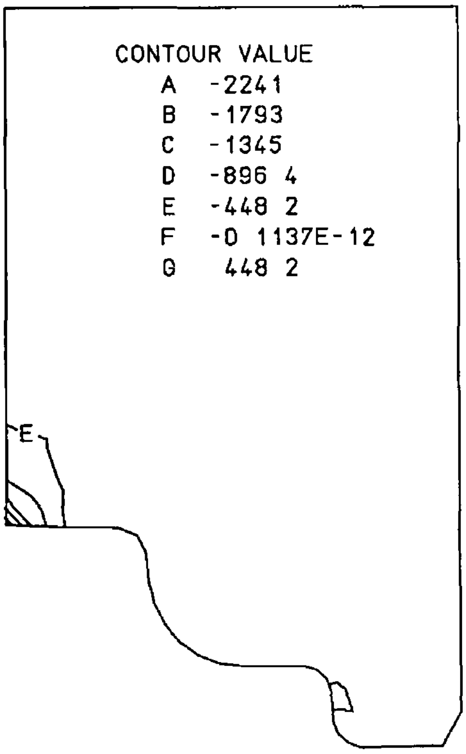


Fig 7.18 Z-Stress distribution (10% yield)

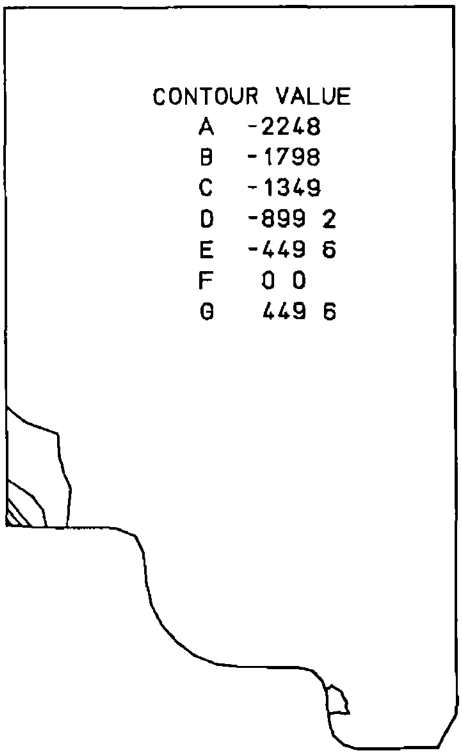


Fig 7 19 Z-Stress distribution (20% yield)

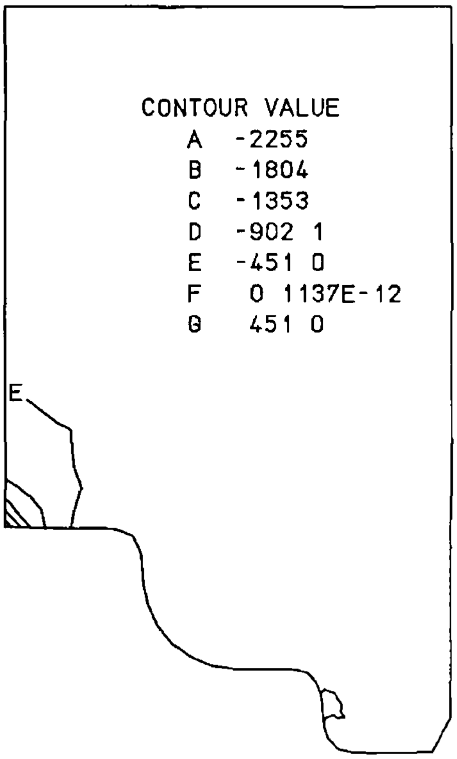


Fig 7 20 Z-Stress distibution (30% yield)

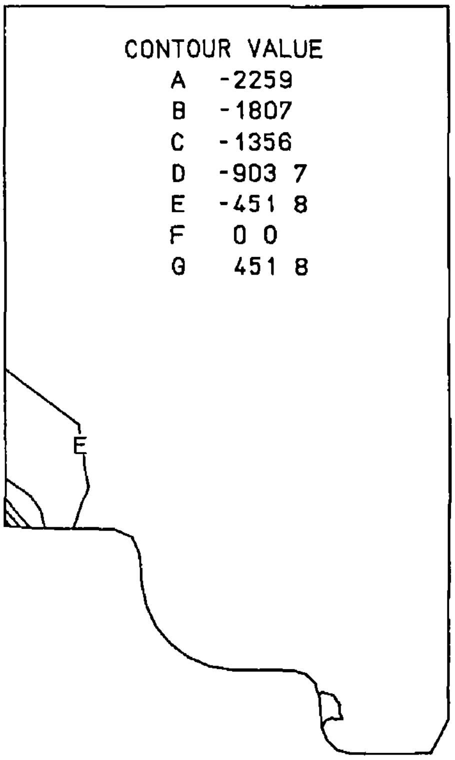


Fig 7 21 Z-Stress distribution (40% yield)

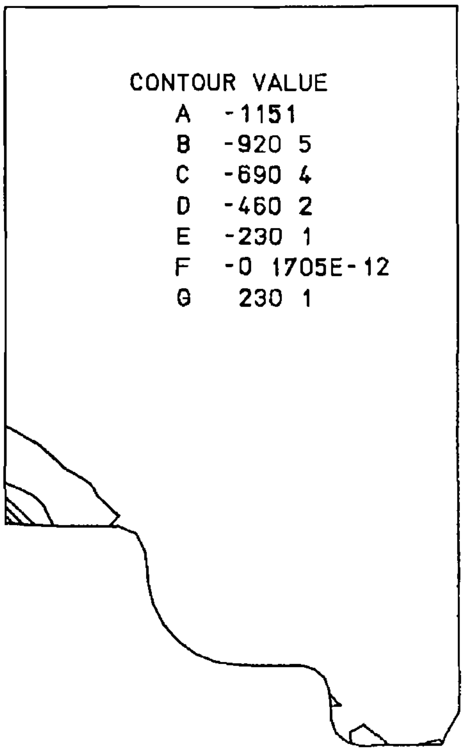


Fig 7 22 Hoop stress distribution (10% yield)

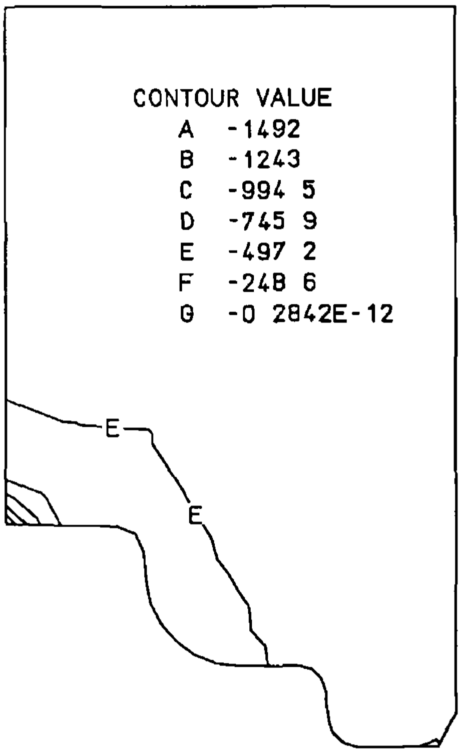


Fig 7 23 Hoop stress distribution (20% yield)

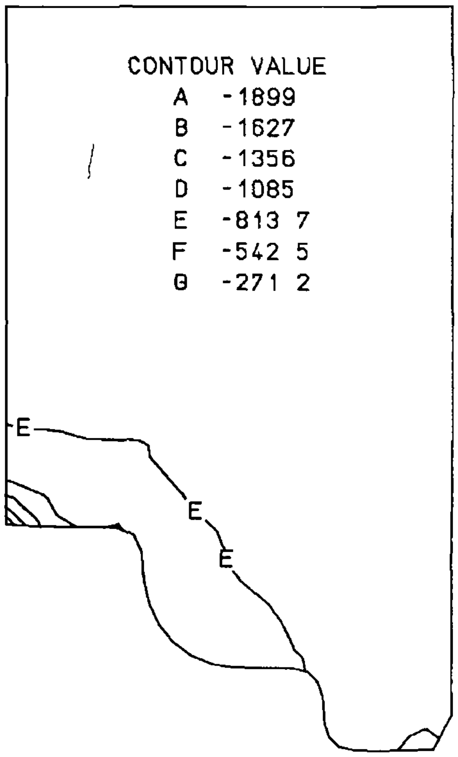


Fig 7 24 Hoop stress distribution (30% yield)

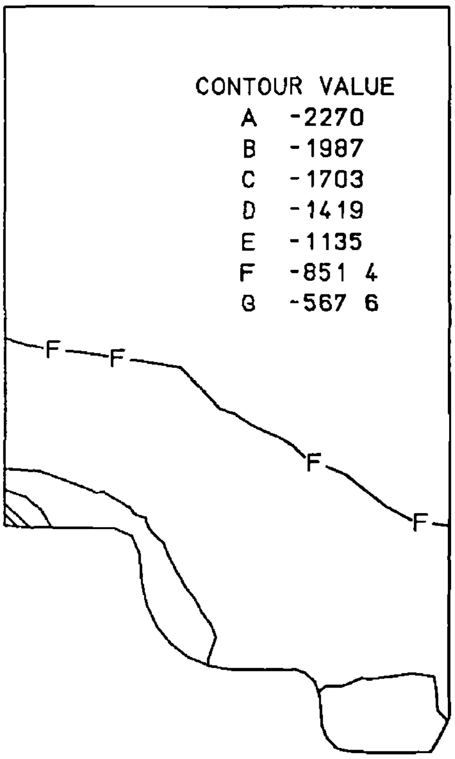


Fig 7 25 Hoop stress distribution (40% yield)

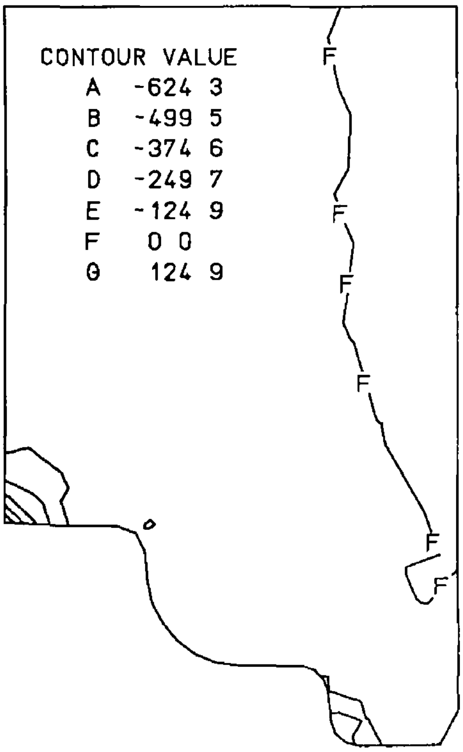


Fig 7 26 Shear stress distribution (10% yield)

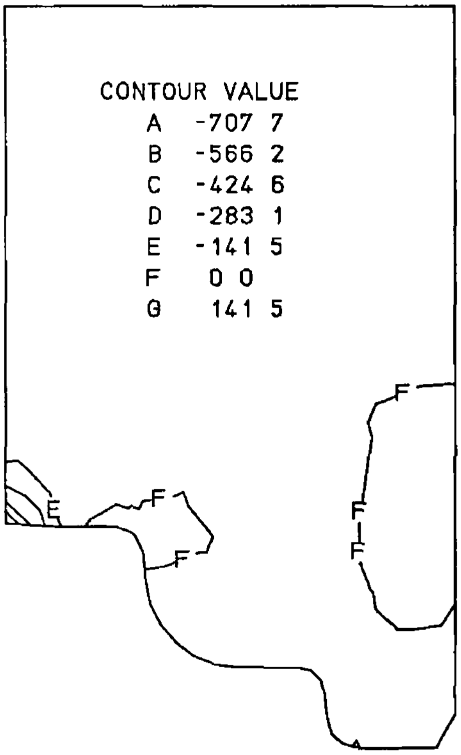


Fig 7 27 Shear stress distribution (20% yield)

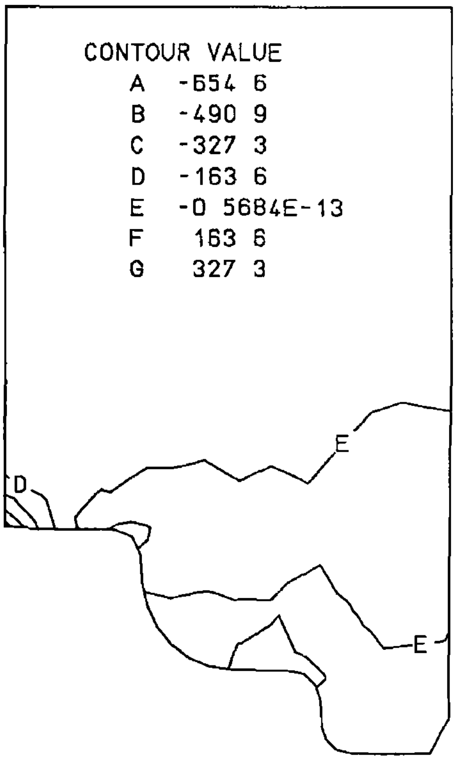


Fig. 7.28 Shear stress distribution (30% yield)

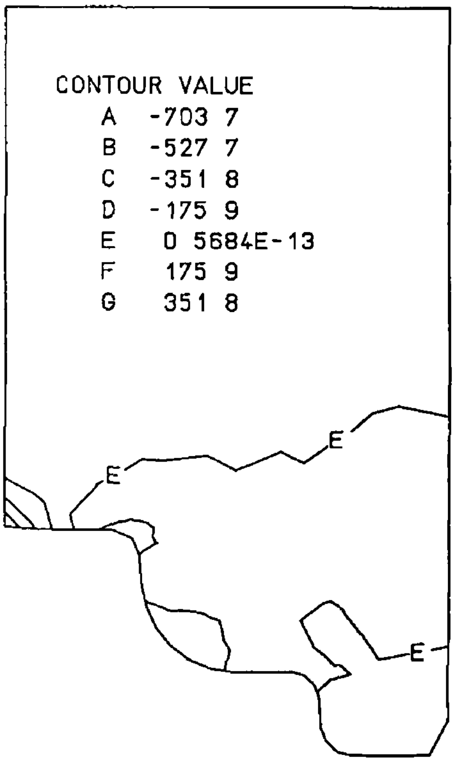


Fig 7 29 Shear stress distribution (40% yield)

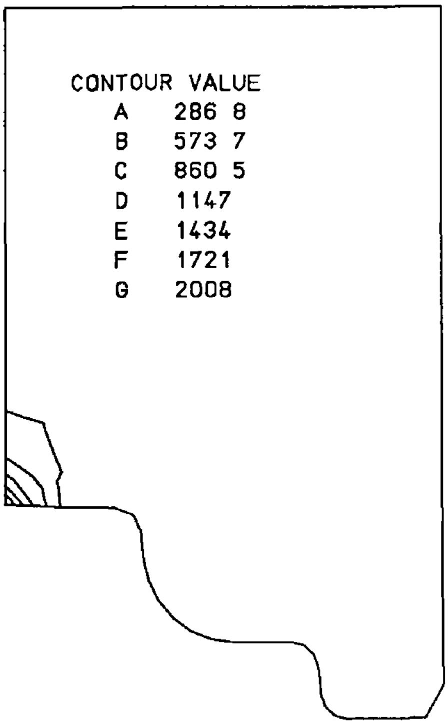


Fig 7 30 Equivalent stress distribution (10% yield)

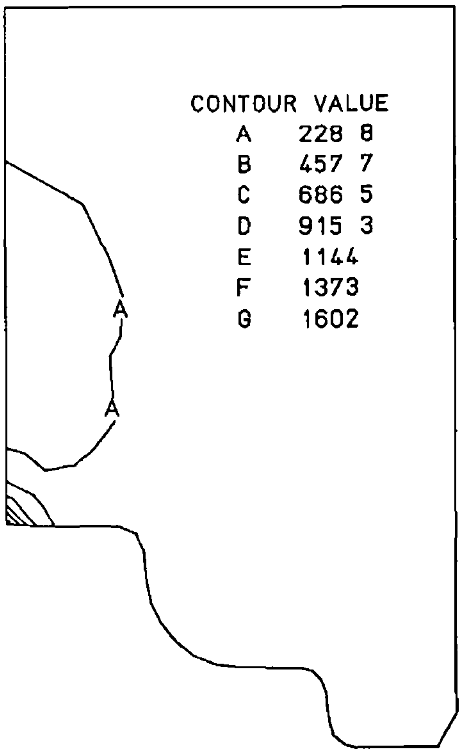


Fig 7 31 Equivalent stress distribution (20% yield)

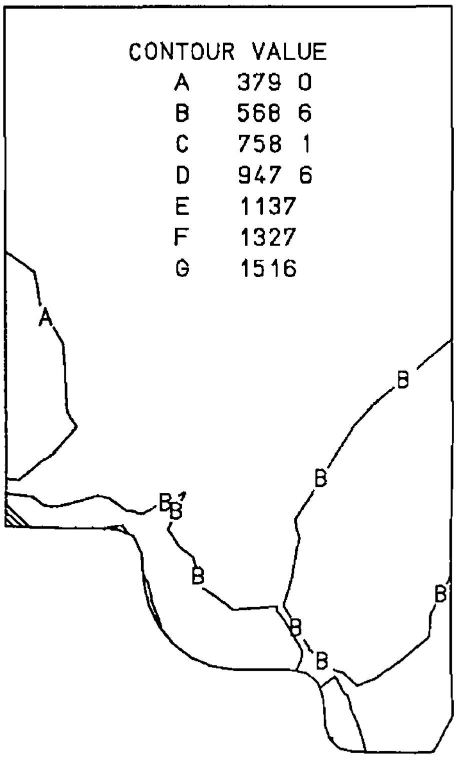


Fig 7 32 Equivalent stress distribution (30% yield)

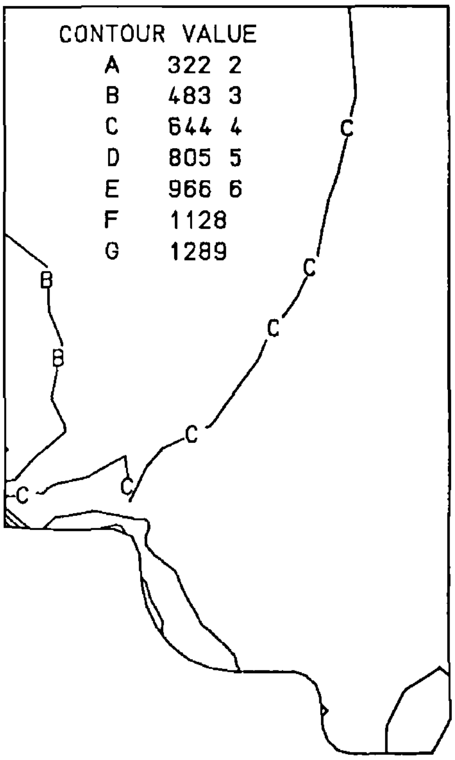


Fig 7 33 Equivalent stress distribution (40% yield)

Because the die failure occurred at the final stage, the forgmg load at this stage from the fnction pomt of view had to be mvestigated In the FE simulation the fnction was considered to be low and constant throughout the forgmg process, which may not be true in reality because of the fact that the lubncant is bemg pushed out by the flowmg matenal dunng the last stage and the fnction factor is believed to be much higher than the value used m the FE simulation Also due to the small thickness of the flash the fnctional stress in this region appioaches the sheai stress which make it difficult for the matenal to flow For this reason the last five steps of the FE simulation have been repeated under high fnction condition to determme the increase the forgmg load at the last stage of deformation The result of this analysis is plotted m Fig 7 34 for the mitial analysis and the case under dry conditions

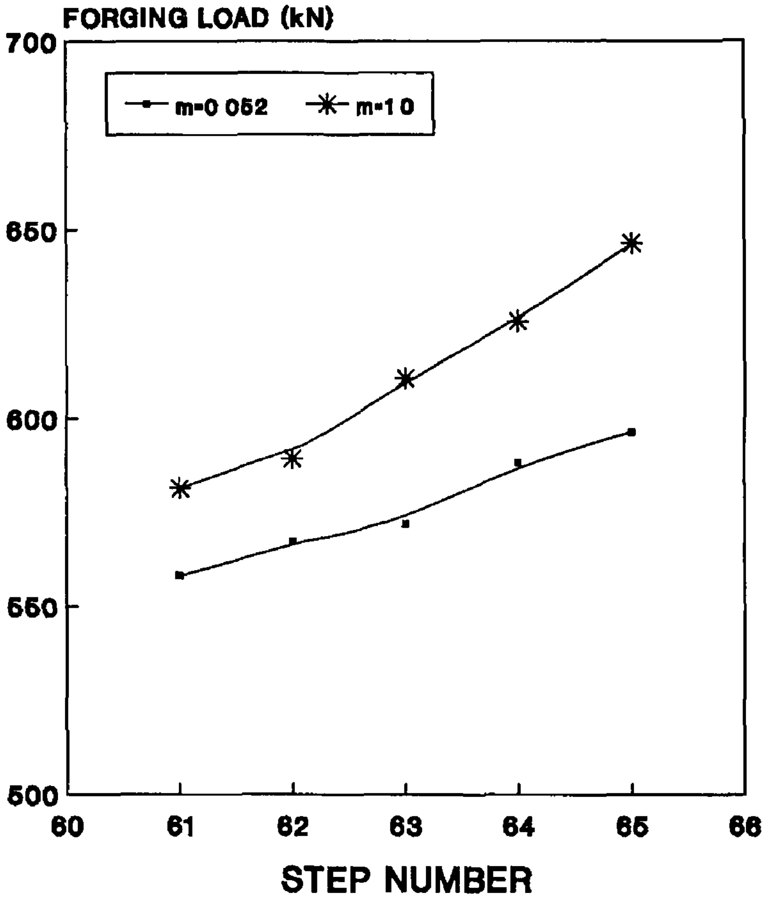


Fig 7 34 Forgmg load durmg the last 5 steps of FE simulation

From this figuie it is cleai that an inciease of neaily 10% m foigmg load is obtamed when the fnction factor is m=1 0 Although the high fiiction is applied along the whole mterface surface between the mateiial and the die, this inciease of load can be attnbuted to the flash region and the upper corner of the die because the rest of the cavity is filled with the material and no flow is taking place there

Because the factor of safety for the initial die design was equal to 1 35, it is clear that in the experiment the die is subjected to much greater load than what has been predicted in the theory

The new calculated load is used to analyze the die and finalize the die design as shown in Fig 7 35 where the same FE model is used with the new load applied in the cavity From this analysis it is found that a few elements on the mid line of the die deformed plastically To solve the problem, the height of the die is increased by 20 mm and the analysis is carried out again as shown in Fig 7 36 In this figure the maximum effective stress is just above half of the yield stress

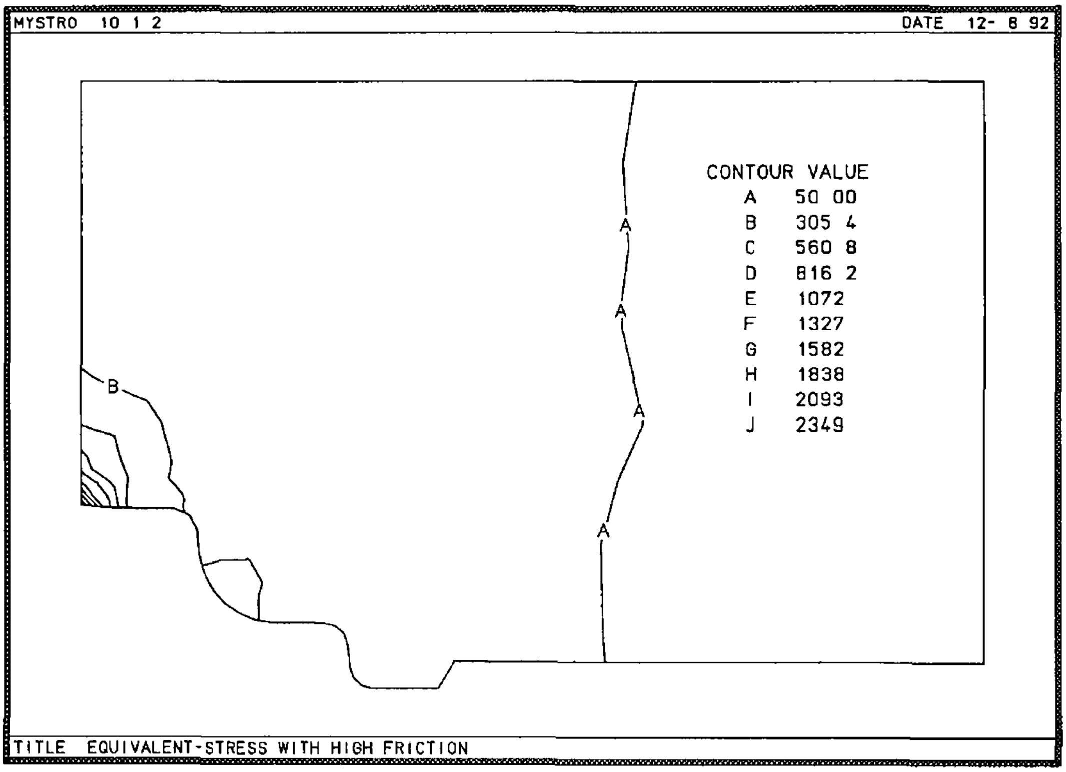


Fig 7 35 The effective stress distribution with the new load and the initial die dimensions

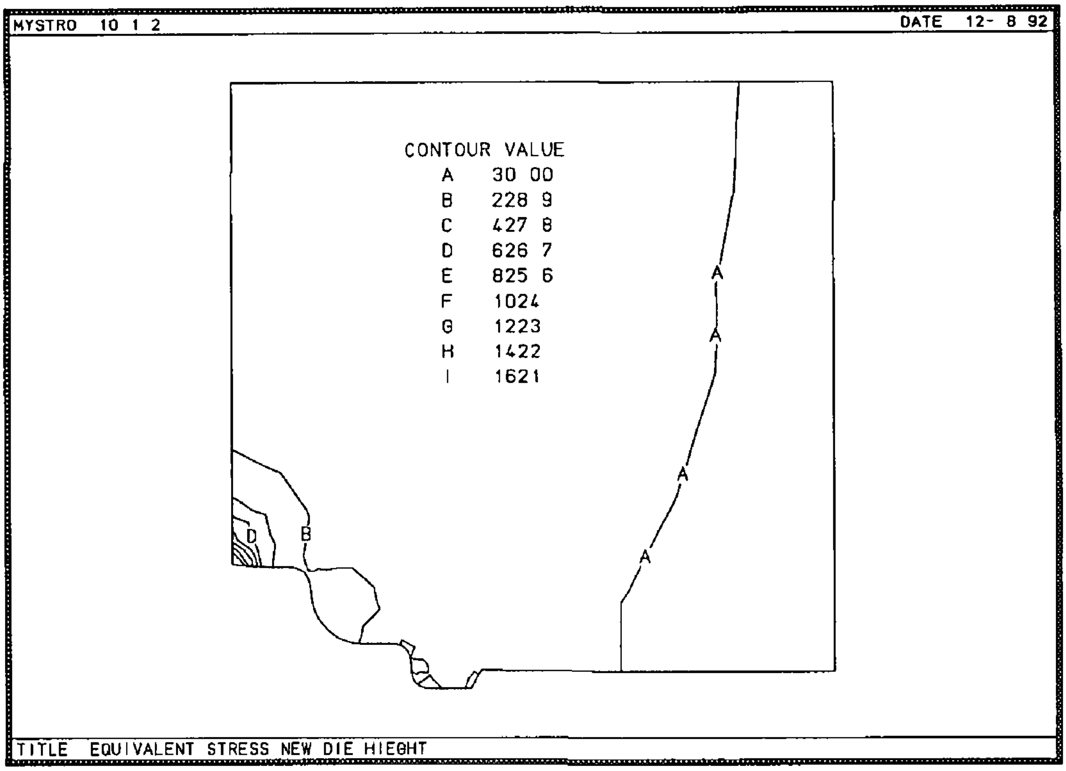


Fig 7 36 The effective stress distribution after changing the die height

Fig 7 37 shows three stages of deformation for the axisymmetric billet in the second trial of the die in which thicker lubricant is used and the die is made of to parts, insert and a case The forging process commenced with the upsetting of the billet and ended with a complete filling of the central flange Then the material started to flow through the flash land to ensure the die filling The last stage of the forging where the forging process is interrupted by a crack is shown as well

Fig 7 38 shows the load-displacement curves of the FE simulation and the experiment The curve of the case with high friction on the final stages of forming simulation has been plotted as well It can be seen that at the last stages the experimental curve goes higher than the theoretical one when friction considered is constant and low throughout the forming process Thus, in reality the maximum stress in the die will be somewhat greater than that predicted according to the simulation at the final stages of forging However, with the increased dimensions this stress is still lower than the yield stress Plate 7 9 shows the initial billet and three stages of deformation

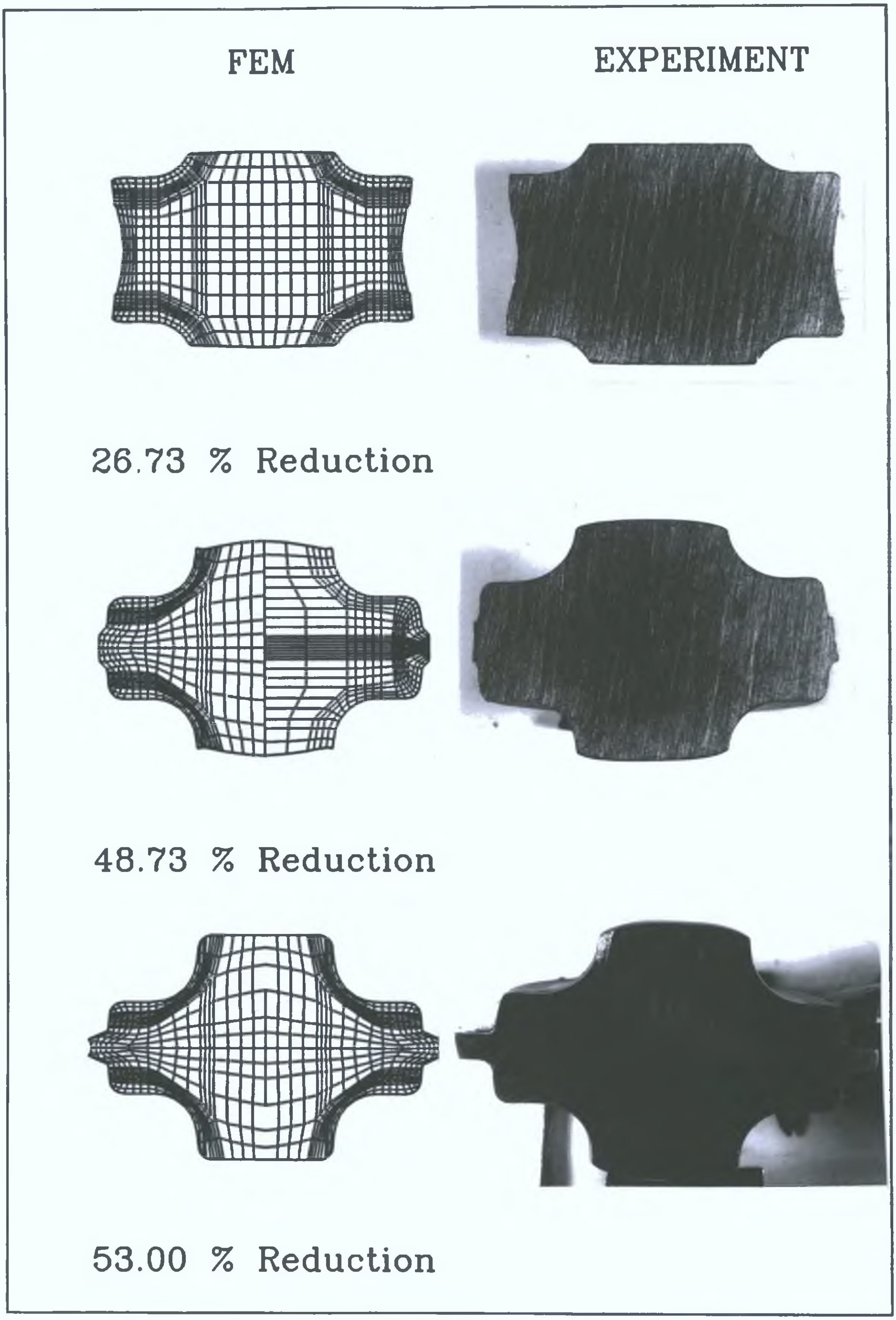


Fig. 7.37 Experimental and FE result of the axisymmetric component

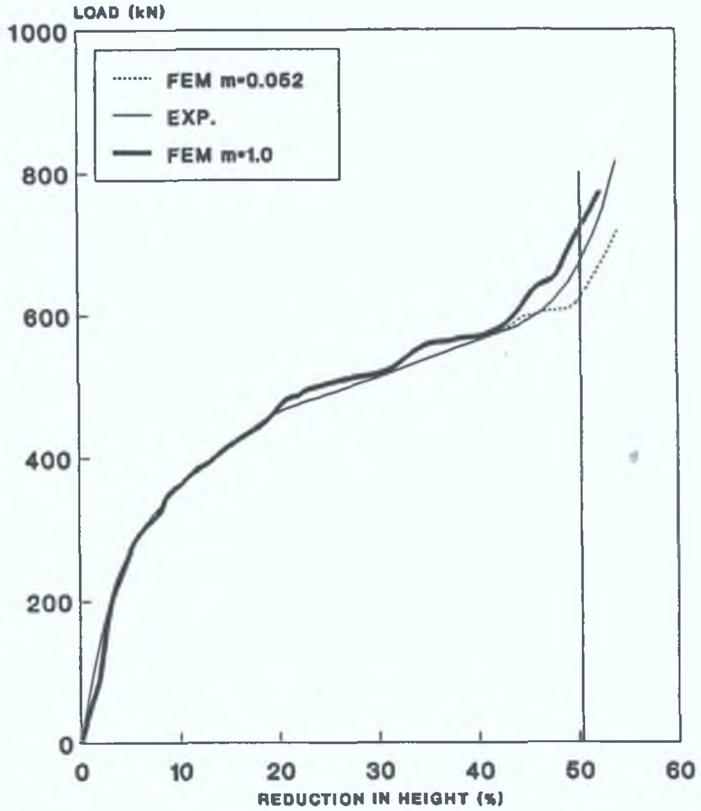


Fig. 7.38 Load-Displacement curves for the axisymmetric forging

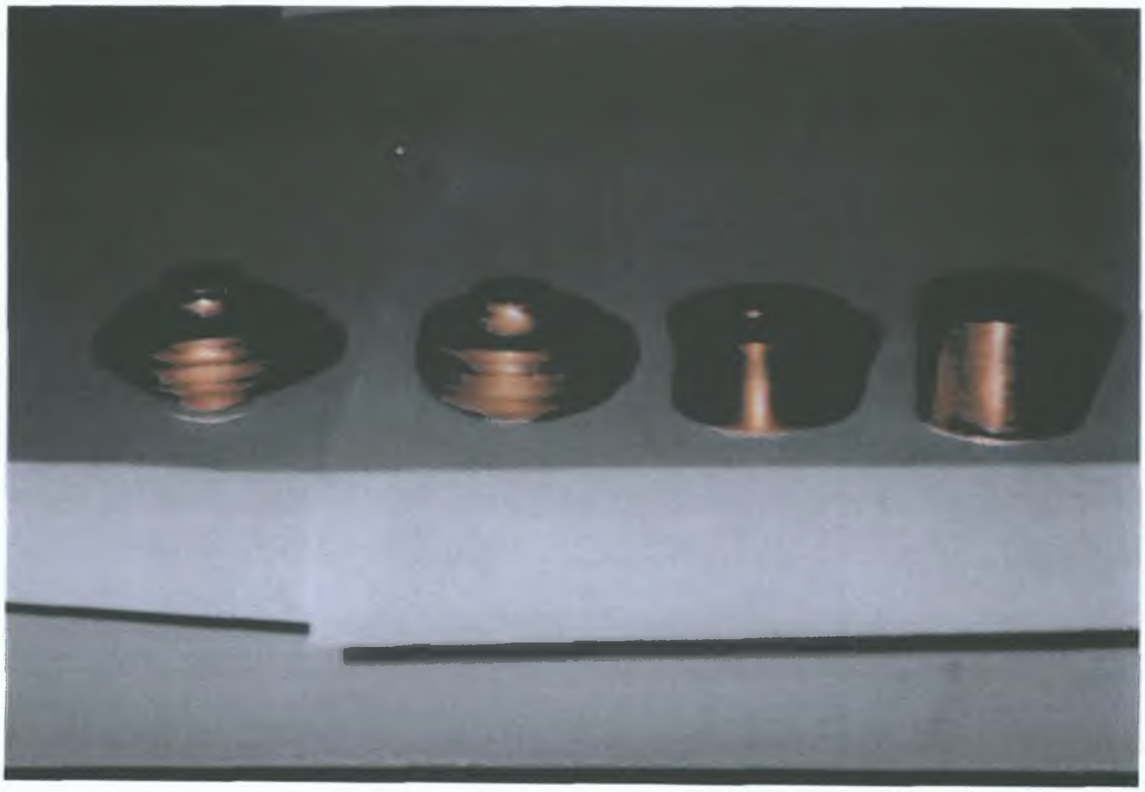


PLATE 7.9 View of the billet and three deformation stages

7 7 MECHANICAL FATIGUE

The stress changes occurring during the forging cycle can cause mechanical fatigue Mechanical fatigue usually occurs in fillets in the die, such as the bottom of the cavities, because they act as stress risers In considering the mechanical fatigue in the design the fatigue strength is used which is proportional to the hardness and tensile strength of the material Fatigue strength is the stress to which the material can be subjected for a specified number of cycle The method of improving fatigue life strength are,

1 eliminate stress raisers by streamlining the part
2 avoid sharp surface
3 prevent the development of surface discontinuities or decarburization
4 improve the details of fabrication

The fatigue ratio (fatigue limit divided by ultimate tensile strength) is approximately 0 5 when determined by polished unnotched specimens subjected to the stress cycle The fatigue ratio varies from 0 4 to 0 6 for engineering material

To calculate the cycles to failure one of two approaches can be used for the case under consideration These two approaches make use of the fatigue data which are available for die material

7 7 1 STRESS-BASED APPROACH TO FATIGUE

The design of a component that will be subjected to cyclic loading can be approached by adjusting the configuration of the part so that the calculated stresses fall safely within the required life line on a constant-life diagram In this method the material is assumed to deform in a nominally elastic manner, local plastic strains are neglected To the extent that these approximations are valid, the stress-based approach is useful These assumptions imply that the stress will be essentially elastic The constant-life fatigue diagrams are available for all type of steel

7 7 2 STRAIN-BASED APPROACH TO FATIGUE

This approach which is developed for the analysis of low cycle fatigue data has proven useful for analyzing long-life fatigue data as well The approach can account for both elastic and plastic responses to applied loading

$$\frac{\Delta\varepsilon_p}{2} = \varepsilon_f \ (2N_f)^c \qquad (7.7)$$

ε_f is the fatigue ductility coefficient

c is the fatigue ductility exponent

N_f is the number of cycle to failure

in stress-based analysis

$$\frac{\Delta\sigma}{2} = \sigma_f \ (2N_f)^b \qquad (7\ 8)$$

σ_f fatigue strength coefficient

b fatigue strength exponent

The elastic strain range is obtained by dividing Eq 2 by E as follows,

$$\frac{\Delta\varepsilon_e}{2} = \frac{\sigma_f}{E} \ (2N_f)^b \qquad (7.9)$$

The total strain range is given by the sum of plastic and elastic component, obtained by adding Eq (7 7) and Eq (7 9),

$$\frac{\Delta\varepsilon}{2} = \varepsilon_t \ (2N_f)^c + \frac{\sigma_f}{E}(2N_f)^b \qquad (7.10)$$

For low cycle fatigue conditions (less than 1000 cycles to failure) the first term of Eq (7 10) is much larger than the second, thus, analyzing and design under such conditions must use the strain-based approach For long life fatigue conditions (more than 10 000 cycles to failure), the second term dominates, and the fatigue behaviour is adequately described by Eq (7 8) in stress-based analysis and design

In the case of the axisymmetric forging die the constants of the die material are taken as,

$\varepsilon_f = 0\ 07$

$c = -0\ 76$

By using the first term of Eq 4 and assuming 50 cycles the result strain in 0 084 which is less than the maximum strain calculated by the FE program for the die under the forging conditions

For industrial design of dies usually the number of forging produced by each die is expected to be in thousands In this case incorporating the calculation of the mechanical fatigue within the design procedure becomes necessary and essential

7 8 COST EFFECTIVENESS

The capital cost of this system is divided into three parts,

1 Hardware cost, which is nearly £5000 including the prephirals

2 Software costs, which is divided into two parts,

  - The commercial packages

AutoCAD	£500
LUSAS	£500 per year

  - The finite element simulation package $1000

3 Training costs £1000

So the total cost of such system is between £8000 and £10 000

The time spent during designing and analyzing the two die is found to be as follow,

1 for the plane strain die it was 26 hrs which includes the first prediction of the die geometry and the simulation of the metal flow and finally analyzing the die and preparing the manufacturing drawing The simulation process has been carried out for two lubricant conditions

2 For the axisymmetric die the time needed to finalize the die design was 13 hrs

So the total time which was spent in designing these two die was 39 hrs which is equal to four working days Enquiries are made to find out the time needed by small forgers to design these two dies Nearly one week was necessary to complete the die design of these two die It looks as if the difference in the time is not significant, but when complex dies are involved the difference between both the conventional design method and the computer aided design becomes very large Especially when the material flow is very complicated and the forging conditions are difficult to consider in the conventional way of designing

CHAPTER EIGHT

CONCLUSION

The application of computers in forging industry continues to increase This is mainly due to,

1 The demand by the customers of forgings for using electronic geometry transfer

2 Increased emphasis on quality, reproducibility and shorter delivery schedules

3 Savings obtained by automatic design, drafting and NC machining of forging dies

4 Advantages of computer simulation in reducing the costs and time in process development

In this study the geometric capabilities of an available CAD/CAM system is augmented by analysis software to calculate forging stresses and loads and to design blocker shapes using metal flow simulation As a result, the need for expensive die layout trials on the forge shop floor will be reduced In addition, material utilization will be improved by optimizing the geometries of blockers through computer aided simulation and by optimizing flash design

The design of closed die forging using the developed CAD system does not differ from that of the conventional design in term of the geometrical design of the die Nevertheless, the CAD procedure reduces the time spent on designing, increases the accuracy of the drawings and reduces the errors in selecting design data
Errors can be identified and corrected easily before the incorrect data leads to costs and difficulties in manufacturing
The design of a forging die, the choice of forging machine and the mounting of the dies on a certain machine essentially require the determination of several parameters There

are difficulties in making these estimations when the conventional planning of forging operation is used The calculation of these parameters has become possible and easier through using the developed system

The system developed in this work has the following features,

1 The system developed uses PC computer system which is less expensive and within the reach of small scale forgers

2 A comprehensive CAD system has been developed that can design finisher dies for a wide variety of forging cross sections, providing a low cost method as a result of customizing a CAD system

3 Most of the geometrical design of dies, pre- and post processing of the finite element simulation have been incorporated within the CAD system either through the data base of the system or by using the standard format file (DXF)

4 The built-in design rules are believed to be the best available and give realistic results

5 The built-in design rules were implemented in module form and can be easily updated if better ones become available

6 The interactiveness and flexibility enable the package to provide results to suit the requirements of individual designer

7 The package has been constructed in an interactive manner Upon execution, suggestive design information is displayed A dialogue, which guides a user to the design processes and the use of package, is maintained Design results are displayed on the screen Hard copies can be obtained through a plotter and printer

8 The system relieves the designer of tedious area and volume calculations, a requirement for die design

9 Another major advantage of the system is its ability to access three-dimensional solid models of the part and retrieve critical two-dimensional cross sections for die design

10 It is also a starting point of an integrated CAD/CAM system for forging die design and manufacture Given the specifications of a forged shape in three-dimension, it will be possible for the user who has a fundamental knowledge of machining operation to manufacture the die block and produce the part program for CNC machines

11 In the mesh generation program (MGP) an adequate boundary description is achieved because the original geometry of the component is generated using CAD system and the data is retrieved from the database of the drawing

12 The MGP has the capability for describing zones of different materials which is also useful in refining the mesh in some regions

13 The MGP has a facility for grading the mesh to achieve the required accuracy of idealization

14 A renumbering system to minimize the half bandwidth is incorporated This feature results in improved computational efficiency

15 Node and element numbering is plotted on the drawing proportionally to the correspondent element and in different layers, so the user has the privilege of using the CAD capabilities (Zoom,Pan,Layer on/off)

16 A rezoning scheme is developed to overcome the difficulties encountered in analyzing metal forming processes caused by large deformation One can, by the rezoning procedure, calculate the process step-by-step to obtain a detailed description of the material flow throughout the process

17 A finite element software package has been developed based on the rigid-Plastic formulation for analyzing non-steady forming processes by means of an incremental

procedure

18 A method has been implemented to treat the boundary conditions and contact problem of arbitrarily shaped dies in a unified way This method is based on the discritization of the die cavity to one dimensional segments

After designing the forging the dies for both processes, experiments were conducted to compare the experimental results with those produced by the FE simulation program

Plane strain experiments were carried out for two friction conditions and the results were in good agreement with the FE simulation The profile of the cross section of the forging and the load-displacement curves have been used for comparison

During the forging experiments of the axisymmetric die, a die failure has been encountered and as aresult of the analysis the following recommendations have been suggested,

- a thick lubricant should be used in closed die forging because it will provide better homogeneity and will not be accumulated in the lower die
- the ring test to find out the friction factor should be conducted under similar condition which is experienced by the flash land This can be achieved by making the ring dimensions as close as possible to the flash ring
- in analyzing the die the factor of safety should be increased by 40% to cover any differences between the experimental and the theoretical forging conditions

8 1 SYSTEM LIMITATION

Some of the limitations of the developed system are,

1 Basically the system is two dimensional and analyzes forgings by cross-sections

2 Because of the modular approach used in the system, it can handle only forgings whose cross-sections are axisymmetric and plane strain

3 Due to the inherent complexity of the forging process, most of the stored values are chosen from the empirical data Further analytical studies are necessary to improve these values and thus make the system more independent and reliable

4 Temperature is not included in the finite element simulation code, it can only handle just the process under isothermal conditions Although the forging process is carried out at room temperature, part of the applied forging energy is consumed as heat which increases the temperature of the billet This increase in heat is produced as a result of plastic deformation

8 2 Future developments

The system described in this thesis has been developed specifically for the application to metal forming, with large associated plastic deformation produced by this process The emphasis is on the application of CAD on metal forming and the simulation of metal flow

Both the CAD part which is used to define the initial gauss of the die shape and the FE simulation are only applicable to 2D components

This system can be extended to 3D by using the AME (Advanced Modelling Extension) which gives AutoCAD-11 the capability to create 3D solid objects All the programs created in this work can be rewritten in C language, which is supported by AME, to do the same functions in 3D

Also the FE simulation program could be extended to include temperature and criteria to detect material defects such as folding and cracks The FE program could be extended to include sheet metal forming A 3D FE simulation program could be developed on the basis of the present work and using appropriate hardware an acceptable level of computer time can be achieved

REFERENCES

1 **P W Wallace and J A Schey,**"*Metal Flow in Forging A practical Study* ", The 8th Mach Tool Des and Res Conf , Birmingham, Sep 1967, p 1361

2 **F Neuberger and L Mockel** Werkstattstechnik,1961, Vol 51

3 **Charles Wick,** *Tool and manufacturing Engineering Handbook* 4th edition, V 11, Socity of manufacturing engineers (SME)

4 **German Standard,** DIN 7523 (1972)

5 **British Standards,** BS 4114,(1967)

6 **A Raikar, I Haque and J Jackson** "*An Empirical Based Computer-Aided Design Procedure for Die Design Forging* " J Applied Metalworking,Vol 4,No 4,January 1987

7 **A Raikar, I Haque and J Jackson,**"*A Computer Aided Design procedure for forging die design*", Computer in Engineering Conference, held on 1986, Chicago Illnois, U S A ,pp 57-65

8 **N Akgermann,J R Becker, and T Altman,**"*Preform Design in Closed Die Forging,*",Metallurgia and Metal Forming,1973, Vol 40, pp 135-138

9 **A.Chamouard,***Estampage el Forge*,Dunod,paris,1964

10 **S K Biswas and W A Knight**,"*Preform Design for Closed Die Forging Experimental Bases for Computer-Aided Design*," Int J Mach tool Des and Res , 1975, Vol 15, pp 179-193

11 **W A Knight and C Poli**, "*Computer-Aided Product Design for Economic Forging*," Computers in Engineering, 1982, Vol 1, pp 7-12

12 **G P.Teterin, et al**,"*Shape Difficulty Criteria for Forgings*," Kuznechuo-Shtamp,Proizv, 1966, Vol 7

13 *Final Report of Load and Energy Estimation in Forging*, DFRA, U K , 1982

14 **S H Choi and T A Dean**,"*Computer-Aided to Data Preparation for Cost Estimation*",Int J Mach Tool Des and Res , 1979, Vol 19, pp 181-193

15 **G Mullineaux and W A Knight**,"*Estimation of the Centre of Loading for Forgings*," Int J Mach Tool Des and Res , 1979, Vol 19, pp 181-193

16 *Metals Handbook, Forging and Casting*, 8th ed , Vol 5, American Society for Metals, 1970

17 **T Altan and G D Lahoti**, "*Limitations, Applicability and Usefulness of Different Methods in Analysing Forming Problems*",Annals of the CIRP, 1979, Vol 28, pp 473-483

18 **T Altan, H J Henning, and A M Sabroff**,"*The Use of Model Materials in Predicting Forming Loads in Metalworking*," Transactions of the ASME, 1970, pp 444-452

19 **S W Lui and M K Das**,"*Interactive Design of Axisymmetric Forging Dies Using a Disk Top Computer*"J Mechanical Working Technology 1981, Vol 5, pp 85-103

20 **S K Biswas and B W Rooks** "*Application of Modular Approach to Estimate Load and Energy in Closed Die Forging*," Proc 13th Int Mach Tool Des and Res Conf

1973, pp 445-453

21 **S K Biswas and B W Rooks,**"*Application of a Computer Simulation Technique to Estimate Load and Energy in Axisymmetric closed die forging*", The 13th M T D R , Conference, Birmingham 1972

22 **Y Van Hoenacker and T A Dean,**" *Approximate Method for the Calculation of Loads and General Flow Patterns when Forging Real Materials* " The 20th Int Mach Tool Des and Res Conf , Birmingham, Sept 1979, p 39

23 **M S Hashmi and W A Klemz,**"*Closed Die Forging of Axisymmetric Shapes from Cylindrical Billets of Strain Hardening and Strain Rate Sensitive materials, Experimental Results and Theoretical Predictions* " The 20th Inter Mach Tool Des and Res Conf , Birmingham, Sept 1979, p 61

24 **Y K Chan, G Mullineux, and W A Knight,**"*Progress in the Computer Aided Design and Manufacture of Hot Forging Dies*", The 20th International M T D R Conference, Birmingham, Sept 1979,pp 29-38

25 **S H Choi and T A Dean**, "*Computer Aided Design of Die Block Layouts*", Int J Mach Tools Manufact Vol 27, No 1, pp 91-163, 1987

26 **S H Choi and T A Dean,**" *Computer Aided to data Preparation for Cost Estimation*", Int J Mach Tool Des Res , Vol 24, No 2 ,pp 105-119, 1984

27 **T Altan and F J Fiorentino,**"*Prediction of Loads and Stresses in Closed Die Forging*," J Engr for Industry, 1971, pp 477-486

28 **S Toren, J Ebbesen, J Slutas, and I Saetre,**"*Die Life Estimation in Forging* " The 15th Inter Mach Tool Des and Res Conf , Birmingham, Sept 1974 p 473

29 **T Altan and H Henning,**"*Closed Die Forging of Round Shapes,*" Metallurgia and Metal Forming, 1972, Vol 39, pp 83-88

30 **S K Biswas and W A Knight**,*"Computer-Aided Design of Axisymmetric Hot Forging Dies,"* Proc 15th Mach tool Des and Res Conf ,Birmingham 1974

31 **S K Biswas and W A Knight**,*"Toward an Integrated Design and Production System for Hot Forging Dies,"* Int J Prod Res , 1976, Vol 14, pp 23-49

32 **G Mullineux and W A Knight**, *"Integrated CAD/CAM of Dies for Hot Forgings,"* The 21 Inter Mach Tool Des and Res Conf , Birmingham, Sept 1980

33 **T Subramanian and T Altan**,*"Application of Computer-Aided Techniques to Precision Closed Die Forging,"* Annals of CIRP, 1978, Vol 27, pp 123-127

34 **N Akgermann and T Altan**,*"CAD/CAM of Forging Dies for Structure Parts,"* presented at the first North American Metal-Working Conference, 1973

35 **S H Choi, P Sims and T A Dean**, *"A CAD/CAM Package for Forging Hammer Dies,"* 25th Int Machine Tool Design and Research Conference Apr 1985,*Birmingham*, England

36 **G W Rowe**, *"Recent developments in the theory and practice of metalforming"*, Proc 3rd North American Metalworking Research Conf , Carnegie-Mellon University, Pittsburgh, PA, 1975, pp 2-25

37 **S Kobayashi**,*"The Mechanics of Plastic Deformation in Metal Forming Processes,"* Proc of the U S Natl Congr Of Appl Mech West Period Co , North Hollywood, California,1979, pp 115-146

38 **A H Shabaik**,*"Mechanics of Plastic Deformation in Metal Forming Processes Experimental and Numerical Methods,"*Current Advances in Mechanical Design and Prod ," Ed G S A Shawki, and S M Metwalli, (Proc conf) Cairo, Egypt, Press, 1981, pp 269-288

39 **A H Shabaik**, *"Plane Strain Extrusion and Forging, Numerical Methods in*

Industrıal Foı mıng Pı ocesses", U K Swansea, (Pıoc)Ed J F T Pıttman, R D Wood, J M Alexandeı, O C Zıenkıewıcz, Pıneııdge Pıess Ltd , U K Swansea, 1982, pp 321-331

40 **P V Marcal, I P Kıng,**"*Elastıc-Plastıc Analysıs of Two Dımensıonal Stress Systems by the Fınıte Element Method,*"Int J Mech Scı 1967, Vol 9,p 143

41 **Y Yamada, N Yoshımura, and T Sakuıaı,**"*Plastıc Stress-Straın Matrıx and ıts Applıcatıon for the Solutıon of Elastıc-Plastıc Pı oblems by the Fınıte Element Method,*"Int J Mech Scı 1968, Vol 10, pp 343-354

42 **C H Lee and S Kobayashı,**" *Elastoplastıc Analysıs of Plane-Straın and Axısymmetrıc Flat Punch Indentatıon by the Fınıte Element Method,*" Int J Mech Scı , 1970, Vol 12,pp 349-370

43 **C H Lee and S Kobayashı,**"*Analysıs of Axısymmetrıc Upsettıng and Plane-Straın Sıde-Pressıng of Solıd Cylındeı s by the Fınıte Element Method,*"Journal of Eng for Ind , Trans ASME,Seııes B, Vol 93,No 2, 1971, p 445

44 **E H Lee and R L Mallett,** "*Stıess and defoı matıon analysıs of metal extrusıon process,* SUDAM, No 76-2, Stanfoıd Unıveısıty Stanfoıd, CA, 1976

45 **C H Lee and S Kobayashı,** "*New Solutıons to Rıgıd-Plastıc Deformatıon problems Usıng a Matıı x Method,*"J of Eng Foı Ind , Trans ASME, 1973, Vol 95, pp 865-873

46 **S Kobayashı and S N Shah,** "*The matıı x method foı the analysıs of metal formıng processes*",Advances ın defoımatıon pıocessıng,Plenum, New York,1978, pp 51-98

47 **O C Zıenkıewıcz and Y K Cheung,**"*The fınıte element method ın structural and contınuum mechanıcs,* Mc Gıaw-Hıll, London 1967

48 **N Rebelo, H Rydstad, and G Schı oedeı ,**"*Sımulatıon of Metal Flow ın Closed Dıe*

Forging by Model Techniques and Rigid-Plastic Finite Element Method,"Numerical Methods in Industrial Forming Processes, U K Swansea, (Proc)ED J F T Pittman, R D Wood, J M Alexander, O C Zienkiewicz, Pineridge Press Ltd ,U K Swansea, 1982, pp 237-246

49 **N L Dung and O Maherenholtz**, "*Progress in the Analysis of Unsteady Metal-Forming Processes Using the Finite Element Method,*"Numerical methods in Industrial Forming Processes, U K Swansea, (Proc) Ed J F T Pittman, R D Wood, J M Alexander, O C Zienkiewicz, Prineridge press Ltd ,U K Swansea, 1982, pp 187-196

50 **M C Tang and S Kobayashi**,"*An Investigation of the Shell Nosing Process By the Finite Element Method,Part I Nosing at Room Temperature,*" J of Engr for Ind ,Trans ASME, Vol 104(3), 1982, pp 305-311

51 **S N Shah, C H Lee and S Kobayashi**, "*Compression of tall, circular, solid cylinders between parallel flat dies*", Proc Int Conf on Production Engineering, Tokyo, Japan, 1974, pp 295-300

52 **O C Zienkiewicz and P N Godbole**, "*Flow of Plastic and Viscoplastic Solids, with Special Reference to Extrusion and forming Processes,*" Int J Num Mech Engeg ,1974, Vol 8, pp 3-16

53 **O C Zienkiewicz, P C Jain and E Onate**, "*Flow of Solids During Forming and Extrusion Some Aspects of Numerical Solutions,*" Int J Solids and Structure, 1978, Vol 14, pp 15-38

54 **S I Oh**, "*Finite Element Analysis of Metal Forming Processes with Arbitrarily Shaped Dies,*" Int J Mech Sci , 1982, Vol 24, No 8, pp 479-493

55 **S I Oh, G D Lahoti,and T altan**, "*ALPID-A General Purpose FEM Program for Metal Forming,*" Proc of NAMRC-IX, May 1981, State College, Pennsylvania, p 83

56 **Y Mitani, V Mendoza**,"*Analysis of rotar shaft forging by rigid-plastic finite element method*", J of material processing technology, 27 (1991) pp 137-149

57 **A Bugini**, "*The influence of die geometry cold extrusion forging operations FEM and experimental results*", J of materials processing technology, 27, 1991, pp 227-238

58 **H D Hibbitt, P V Marcal, and J R Rice**, "*A Finite Element Formulation for Problems of Large Strain, Large Displacement,*" Int J Solids and Structures, 1970, Vol 6, p 1069

59 **N M Wang and B Budiansky**, "*Analysis of Sheet Metal Stamping By a Finite Element Method,*" J Appl Mech , Trans ASME, Vol 45,No 1, 1978, pp 73-82

60 **R M McMeeking and J R Rice**, "*Finite Element Formulation of Problems of Large Elastic-Plastic Deformation,*" Int J Sol and Str , 1975, Vol 11, pp 601

61 **S W Key, R D Kries, and K J Bathe**, "*On the Application of the Finite Element Method to Metal Forming Processes,*" Comput Mech Appl Mech Eng Vol 17/18, Pt 3, 1979, pp 597-608

62 **A S Wifi, Y Yamada**, "*Finite Element Analysis of Metal Working Processes,*" Current Advances in Mechanical Design and Prod , Ed ,G S A Shawki, and S M Metwalli, (Proc Conf), Cairo, Egypt,Pergamon Press, 1981, pp 229-237

63 **A Chandra and S Mukherjee**, "*A Finite Element Analysis of Metal Forming Problems with Viscoplstic Material Model,*" DOE Report No DOE/ER/12038/2, Dept of Theoretical and Applied Mech ,Crnell University, Ithaca, New York, 1982

64 **P Hartly, C E N Sturgess and G W Rowe**, "*Friction in finite element analysis of metal forming processes*", Int J Mech Sci , 21 (1979) pp 301

65 **C C Chen and S Kobayashi**,"*Rigid plastic finite element analysis of ring compression*", ASME Publication, AMD, Vol 28, Application of Numerical Methods to Forming Processes, 1978, pp 163-174

66 **H Matsumoto, S I Oh and S Kobayashi**, "*A note on the matrix method for rigid plastic analysis of ring compression*", Proc 18th MTDR Conf ,London, p 3(Sept 1977

67 **S I Oh and S Kobayashi**,"*Finite element analysis of plane strain sheet bending*", Int J Mech Sci , 22, 1980, pp 538

68 **Y Yamada and H Hirakawa**, "*Large deformation and instability analysis in metal forming process*", Application of Numerical Method to forming process, ASME, AMD, 28,27,1978

69 **E H Lee, R.L Mallett and W H Yang**, "*Stress and deformation analysis of metal extrusion process*", Comput Meth Appl Mech Engng ,10,1977, pp 339

70 **O M Eltouney, K A Stelson**, "*An approximate model to calculate foldover and strain during cold upsetting of cylinders, Part I Formulation and evaluation of the foldover model*",J, of Eng for Industry, Vol 112, Aug 1990, pp 260-266

71 **A D Seikh, J A Dean, M K Das and S A Tobias**, "*The effect of impact speed and die cavity pressure,Part I Interface friction and die cavity*"

72 **A T Male and M G Cockcroft** "*A method for determination of the coefficient of friction of metals under conditions of bulk plastic deformation*", J of the institute of metals, Vol 93, 1963, p 38

73 **A T Male and V De Pierre**," *The validity of mathematical solution for determining friction from the ring compression test*", ASME paper No 69-WA/Lab-8

74 **N I Muskhehshvili**, "*Some basic problems of the mathematical theory of elasticity*",

Nordhoff, The Netherlands, 1963

75 **G M L Gladwell**,*"Contact problems in the classical theory of elasticity"*, Sijthoff and Nordhoff, The Netherland, 1973

76 **J J Kalker**,*"The computation of three dimensional rolling contact with dry friction"*, Int J Num Methods Eng , Vol 14, 1, 1979,pp 293-307

77 **L Nayak and K L Jhonson**,*"Pressure between elastic bodies having slender area of contact and arbitrary profile"*, Int J Mech Sci , Vol 21, 1979, pp 237-247

78 **H Petersson**, *"Application of the finite element method in the analysis of contact problem"*, Proc Non-Linear Solid and Structural Mechanics, pp 845-862, Geilo, Norway, 1977

79 **Bo Tornstenfelt**,*"Contact problems with friction in general purpose finite element computer program"*, Computer and Struct , Vol 16, 1983, pp 487-493

80 **B Fredriksson, G Rydholm and P Sjoblom**,*"Variatinal inequalities in the structural mechanics with emphasis on contact problems"*,Proc Non-Linear Solid and Structural Mechanics, pp 863-884, Geilo, Norway, 1977

81 **J T Oden and N Kikuchi**,*"Finite element methods for constrained problems in elasticity"*, Int J Num Meth Eng Vol 8,1982, pp 701-725

82 **J T Oden and E B Pires**,*"Numerical analysis of certain contact problems in elasticity with non-classical friction laws"*, Computers and Struct , Vol 16, 1983, pp 481-485

83 **E B Pires and J T Oden**,*"Analysis of contact problems with friction under oscillating loads"*, Comp Meth Appl Mech Eng Vol 39, 1983, pp 337-362

84 **A Francavilla and O C Zienkiewicz**,*" A note on numerical computation of elastic*

contact problem",Int J Num Eng , Vol 9, 1975, pp 913-924

85 **T D Sachdeva and C V Ramakrishnan,**"*A finite element solution for the two dimensional elastic contact problems with friction*",Int J Num Meth Eng , Vol 17, 1981, pp 1257-1271

86 **S.K Chan and I S Tuba,**"*A finite element method for contact problems of solid bodies-Part I Theory and validation Part II Application to turbine blade fastening* ", Int J Mech Sci , Vol 13, 1971, pp 627- 639

87 **E A Wilson and B Parsons,**"*Finite element analysis of elastic contact problems using differential displacements* ", Computers and Struct , Vol 2, 1970,pp 387-395

88 **N Okamato and M Nakazawa,**"*Finite element incremental contact analysis with various frictional conditions*", Int J Num Meth Eng ,Vol 14, 1979,pp 337-357

89 **M U Rahman R E Rowlands and** R D Cook,"*An iterative procedure for finite element stress analysis of frictional contact problem*", Computers and Struct , Vol 14 1984, pp 947-954

90 **K J Bathe and A Chaudary,**"*A solution method for planar and axisymmetric contact problems*",Int J Mum Meth Eng Vol 21, 1985, pp 65-88

91 **R B Haber and B H Hariandja,**"*An Eulerian-Lagrangian finite element approach to large deformation frictional contact*", Computers and Struct , Vol 20, 1985, pp 193-201

92 **M J M Barata Marques and P A F Martins,**"*Three dimensional finite element contact algorithm for metal forming*",Int J Num Meth Eng , Vol 30,1990

93 **P Wriggers and J C Simo,**" *A note on tangent stiffness for fully nonlinear contact problems*", Commun Appl Num Meth , Vol 1,1985,pp 199-203

94 **P Wriggers ,J C Simo and R L Tylor**, "*Penalty and augmented Lagrangian formulations for contact problems*", Proc NUMETA 85 Conf , 1985

95 **D N Yates**, et al,"*The development of large scale digital computer codes for production structural analysis*", Lockheed Conference, 1970

96 **B Wordenweber**,"*Volume triangulation, technical report*", University of Cambridge, UK, CAD group document No 110, 1980

97 **B Wordenweber**,"*Finite element mesh generation*",Computer Aided Design, Vol 16, No 5 (sept 1984), pp 285-291

98 **B G Baumgart**,"*Geometric modelling for computer vision*",Report No CS-463 Stanford Artificial Intelligence Laboratory, Comput Sci Dept , Stanford, USA, Oct 1974

99 **J Suhara and J Fukuda**,"*Automatic mesh generation for finite element analysis*",in J T Oden, R W Clough and Y Yamamoto, "Advanced in computational methods in structural mechanics and design", UAH press, Huntsville, Alabama, USA, 1972

100 **R D Shaw and R G Pitchen**,"*Modification to the Suhara-Fukuda method of network generation*",Int J Num Mech Eng Vol 12, 1978, pp 93-99

101 **A O Moscardini, B A Lewis and M Cross**,"*A GTHOM-automatic generation of triangular and higher order meshes*",Int J Num Meth Eng , Vol 19, 1983,pp 1331-1353

102 **J C Cavendish, D A Field and W H Frey**,"*An approach to automatic three-dimensional finite element mesh generation*",Int J Num Mech Eng , Vol 21, part 2, 1985,p 329

103 **D.A.Field and W.H.Frey**,"*Automation of tetrahedral mesh generation*",GMR-4967, G M R L , Michigan, 1985

104 **B Wordenweber**,"*Finite element analysis for the naive user, Solid modelling by computers*", Plenum Press, New York, 1984, p 81

105 **T C Woo and T Thomasme**,"*An algorithm for generating solid elements in objects with holes*", Cop Struct Vol 18, 1984 p 333

106 **W T Wu, S I Oh, T Altan and R A Miller**,"*Automated mesh generation for forming simulation-I*", Proc 1990 ASME, Int Comput Eng , Boston, MA, Vol 1, p 507

107 **M A Yerry and M S Shephard**, "*Automatic three-dimensional mesh generation by the modified-Octress techique*", Int J Num Meth Eng , Vol 20(11) 1984, p 1965

108 **P L Baehmann, S L Wittchen, M S Shephard, K R Grice and M A Yerry**, "*Robust, geometrically based, automatic Two-dimensional mesh generation*", Int J Num Meth Eng , Vol 24(6), 1987, p 1043

109 **J U Brackbill and J S Saltzmann**," *Adaptive rezoning for singular problems in two dimensions*", J Compt Physics, Vol 46, 1982, pp 342-368

110 **A R Diaz, N Kikuchi and J E Taylor**, "*A method of grid optimization for finite element methods*", Comp Meth in Appl Mech Eng , Vol 41, 1983, pp 29-45

111 **N Kikuchi**," *Adaptive grid design method for finite element analysis*", Comp Meth Appl Mech Eng , Vol 55, 1986, pp 129-160

112 **L Demkowicz and J T Oden**," *On a moving mesh strategy based on an interpolation error estimate technique*", Int J Eng Sci , Vol 24(1), 1086, pp 55-68

113 **O C Zienkiewicz, Y C Liu and G C Huang**," *Error estimation and adaptivity in flow formulation for forming problems*", Int J Num Meth Eng , Vol 25,

1988, p 23

114 **J P Cescutti and J L Chenot**, "*A geometrical continuous remeshing procedure for application to finite element calculation of non-steady state forming processes*", NUMETA 87, 2nd Int Conf Adv Num Meth Eng , Theory and Application , Swansea, 1987, p 517

115 **J P Cescutti, E Wey, J L Chenot and P E Mosser**,"*Finite element calculation of hot forging with continuous remeshing*", J L Chenot and E Onate (Eds), "Modelling of metal forming processes", Proc of the Euromech 233 Collquium, Sophia Antipolis, 1988 ,pp 207-216

116 **J P Cescutti, N Soyris, G Surdon and J L Chenot**,"*Thermo-mechanical finite element calculation of three-dimensional hot forging with remeshing*, K Lange (Ed), Proc Adv Technol Plasticity, Stuttgart, 1987, pp 1051-1058

117 **A Thomas**,"*Forging Handbook, Die design*", Drop Forging Research Association, DFRA, (1980)

118 "*Chinese Forging Handbook*", Publ Industrial Machinery Press, Beijing, (1979)

119 **R Hill**, "*The mathematical theory of plasticity*",Oxford University Press, London, 1950

120 **S Kobayashi, S Oh and T Altan**, "*Metal forming and the finite element method*", Oxford University Press, 1989

121 **S I Oh, J P Tang and A Badawy**, "*Finite element mesh rezoning and its application to metal forming analysis*", 1st ICTP confeience, Tokyo, Japan, 1984

122 **R H Grawford, D C Anderson and W N Wagg**, "*Mesh rezoning of 2D isoparametric elements by inversion*", Int J fot Num Method in Eng , Vol 28, pp 523-531, 1989

123 **G Dhatt and G Tou/ot**,*"The fimteelement method displayed"*, Thomson press, India, 1985

124 **G W Rowe**,*"Principles of industrial metalworking processes"* Pitman press, Bath, 1977

125 **W Hu,P Liu and M S J Hashmi**,*"Effect of lubrication and forming speed in plane strain extrusion forging of rectangular billets"*, Proc XIth Conf on Prod Res Hefei, China, Aug 1991

126 **K Lange**,*"Closed die forging of steel"*, (in German), Springer-Verlag, Berlin, 1958

127 **P M Cook**,*"True stress-strain curves for steel in compression at high temperature and strain rates, for application to the calculation of load and torque in hot rolling"*, Institution of Mech Engr Conf on properties of materials at high rates of strain, May 1957

128 **J E Hockett**,*"Compressive stress versus strain at constant strain rate"*, Applied polymer symposia, No 5, p 205, 1967

129 **J F Adler and V A Phillips**, *"The effect of strain rate and temperature on the resistance of aluminum, copper, and steel to compression"*, Journal of Inst of metals, Vol 83, p 80, 1954-1955

130 **H Suzaki, et al**," *Studies on flow stress of metals and alloys*", Vol 18 (3), Serial No 117, reports by the Institute of industrial science, University of Tokyo, Mar 1968

131 **C H Lee and T Altan**, *"Influence of flow stress and friction upon metal flow in upset forging of rings and cylinders*", Journal of Engineering for Industry, Trans ASME, Series B, Vol 94, No 3, Aug 1972, p 775

132 **H Kudo**, Proc 5th Japan Nat Congr Appl Mech , Vol V, p 65 , 1955, Tokyo (science council of Japan)

133 **J R Douglas and T Altan**, "*Flow stress determination for metals at forging rates and temperature*", Journal of engineering end industry, Trans ASME, Vol 97, p 66, 1975

134 **J Y Riedel**,"*Analysis of tool and die failures*", ASM technical report No c7-11 4, 1967, American Society for Metals

Appendix A

Machining allownce program

```
(defun off ()
        (setq w "yes")
        (lin)
        (while (eq w "yes")
        (initget 1 "yes no")
        (setq w (getkword "do you want to machine any other side ?"))
        (if (eq w "yes") (lin))
        )
)
(defun lin ()
        (prompt "Select three sides of the geometry where the one to be")
        (prompt "machined is in the middel ")
        (setq s (ssget))
        (setq e1 (ssname s 0))
        (setq e2 (ssname s 1))
        (setq e3 (ssname s 2))
        (setq x11 (car (cdr (assoc '10 (entget e1)))) y11 (car (cdr (cdr (assoc '10 (entget e1))))))
        (setq x12 (car (cdr (assoc '11 (entget e1)))) y12 (car (cdr (cdr (assoc '11 (entget e1))))))
        (setq x21 (car (cdr (assoc '10 (entget e2)))) y21 (car (cdr (cdr (assoc '10 (entget e2))))))
        (setq x22 (car (cdr (assoc '11 (entget e2)))) y22 (car (cdr (cdr (assoc '11 (entget e2))))))
        (setq x31 (car (cdr (assoc '10 (entget e3)))) y31 (car (cdr (cdr (assoc '10 (entget e3))))))
        (setq x32 (car (cdr (assoc '11 (entget e3)))) y32 (car (cdr (cdr (assoc '11 (entget e3))))))
;
;       Defining thr htree lines
;
        (if (and (or (/= x12 x22) (/= y12 y22)) (or (/= x12 x21) (/= y12 y21)))
        (progn
                (setq x x11 y y11)
                (setq x11 x12 y11 y12)
                (setq x12 x y12 y)
        )
        )
        (if (or (/= x12 x21) (/= y12 y21))
                (setq x x22 y y22 x22 x21 y22 y21 x21 x y21 y)
        )
        (if (or (/= x22 x31) (/= y22 y31))
                (setq x x32 y y32 x32 x31 y32 y31 x31 x y31 y)
        )
        (setq p1 (list x11 y11) p4 (list x32 y32))
;
        (setq p (getpoint "Indicate the side to be machined "))
        (setq xp (car p) yp (cadr P))
        (setq x0 (car (inters (list x21 y21) (list x22 y22) (list 0 0) (list 50 0) nil)))
        (setq y0 (cadr (inters (list x21 y21) (list x22 y22) (list 0 0) (list 0 50) nil)))
        (if (and (/= x0 nil) (/= y0 nil))
                  (progn
                (setq m (/ (- y22 y21) (- x22 x21)))
                (setq h (abs (/ t (sin (atan m)))))
                (setq x (/ (- (* m xp) yp) m))
                (if (> x x0) (progn
                         (setq xt (+ x0 h) yt 0)
```

```
                    (setq ym (* m (- xp xt)) xm xp)
                    )
                    (progn
                    (setq xt (- x0 h) yt 0)
                    (setq ym (* m (- xp xt)) xm xp)
                    )
            )
            (setq p2 (inters (list x11 y11) (list x12 y12) (list xt yt) (list xm ym) nil))
            (setq p3 (inters (list x31 y31) (list x32 y32) (list xt yt) (list xm ym) nil))
                  )
      )
      (if (= x0 nil) (progn
            (if (> yp y21) (setq yp2 (+ y0 t)) (setq yp2 (- y0 t)))
            (setq p2 (inters (list x11 y11) (list x12 y12) (list 0 yp2) (list x12 yp2) nil))
            (setq p3 (inters (list x31 y31) (list x32 y32) (list 0 yp2) (list x31 yp2) nil))
                  )
      )
      (if (= y0 nil) (progn
            (if (> xp x21) (setq xp2 (+ x0 t)) (setq xp2 (- x0 t)))
            (setq p2 (inters (list x11 y11) (list x12 y12) (list xp2 0) (list xp2 y12) nil))
            (setq p3 (inters (list x31 y31) (list x32 y32) (list xp2 0) (list xp2 y12) nil))
                  )
      )
      (entdel e1)
      (entdel e2)
      (entdel e3)
      (command "line" p1 p2 p3 p4)
      (command "")
)
(defun c:off ()
      (off)
)

(defun lsp1 ()
      (initget 1 "yes no")
      (setq s (getkword "Do you prefer automatic selection of the machining allowance?"))
      (if (eq s "no") (progn
                  (setvar "filedia" 0)
                  (command "vslide" "d:/acad/project/machtol/test")
                  (setvar "filedia" 1)
                  (initget (+ 1 2 4))
                  (setq t (getreal "Input the machining allowance you chose "))
                  (redraw)
                  )
                  (progn
      (setq m (getdist "Input the maximum thickness "))
      (setq d (getdist "\nInput the maximum diameter "))
      (setq f (open "d:/acad/project/machtol/tol.dat" "w"))
      (print m f)
      (print d f)
      (print 99 f)
      (close f)
      (command "d:/acad/project/machtol/machtol")
      (graphscr)
      (setq f (open "d:/acad/project/machtol/tol.dat" "r"))
      (setq t (read-line f))
      (setq t (atof t))
      (close f)
```

```
                    )
          )
)
(defun c lsp1 ()
          (lsp1)
)

      DIMENSION J(7),K(10),TOL(6,9)
      OPEN(2,FILE='TOL DAT',STATUS='OLD')
      READ(2,*)TM,DM
      CLOSE(2,STATUS='DELETE')
C
C            Loading the data to the memory
C
      OPEN (1,FILE='DIE1 TOL',STATUS='OLD')
      READ(1,'(I4)')(J(I),I=1,7)
      READ(1,'(I4)')(K(I),I=1,10)
      READ(1,'(F5 2)') ((TOL(I,N),N=1,9),I=1,6)
      CLOSE(1)
C
C        Selecting the proper machining allowance
C
      DO 50 I=1,6
      IF ((TM GE J(I)) AND (TM LE J(I+1))) GOTO 60
50     CONTINUE
60     IL=I
      DO 70 I=1,9
      IF ((DM GE K(I)) AND (DM LE K(I+1))) GOTO 80
70     CONTINUE
80     IC=I
      T=TOL(IL,IC)
      OPEN(2,FILE='TOL DAT',STATUS='NEW')
      WRITE(2,'(F5 2)')T
      CLOSE(2)
      END
```

Appendix B

Fillet addition program

```
(defun fil ()
        (setvar "filedia" 0)
        (fill)
        (setq op2 "Yes")
        (while (equal op2 "Yes")
        (command "fillet" pause pause)
        (initget 1 "Yes No")
        (setq op (getkword "Do you want to do it again with the same fillet radii (Yes or No) ?"))
        (if (equal op "No")
                (progn
                (initget 1 "Yes No")
                (setq op1 (getkword "Do you want to change the fillet radii and continue (Yes or No) ?"))
                (if (equal op1 "No") (setq op2 "No") (fill))
                )
        )
        )
)
(defun fill ()
        (initget 1 "Yes No")
        (setq op3 (getkword "Do you want automatic selection of the fillet radii (Yes or No) ?"))
        (if (equal op3 "Yes") (progn
        (setq m (getdist "Input the maximum shoulder height <250 mm "))
        (setq d (getdist "\nInput the maximum diameter or the maximum width of the forging <630 mm "))
        (setq f (open "d:\acad\project\fillet\fillet.dat" "w"))
        (initget 1 "Internal External")
        (setq op4 (getkword "Is it internal or external fillet radii (Internal or External) ?"))
        (if (equal op4 "Internal") (print 1 f) (print 2 f))
        (print m f)
        (print d f)
        (print 99 f)
        (close f)
        (command "d:\acad\project\fillet\fille")
        (graphscr)
        (setq f (open "d:\acad\project\fillet\fillet.dat" "r"))
        (setq i (read-line f))
        (setq i (atof i))
        (close f)
                           )
                           (progn
                (initget 1 "Internal External")
                (setq op4 (getkword "Is it internal or external fillet radii (Internal or External) ?"))
                (if (equal op4 "Internal") (command "vslide" "d:/acad/project/fillet/filletin")
                (command "vslide" "d:/acad/project/fillet/filletex"))
                (setq i (getreal "Input the radii value either from the table or from your own experience "))
                (redraw)
                           )
        )
        (command "fillet" "r" i)
        (setvar "filedia" 1)
)
(defun c:fil ()
        (fil)
)
```

```
      DIMENSION J(7),K(10),TOL(6,9)
      OPEN(2,FILE='d \acad\project\fillet\FILLET DAT
    1  STATUS='OLD')
      READ(2,*)R,TM,DM
      CLOSE(2,STATUS='DELETE')
C
C            Loading the data to the memory
C
      IF (R EQ 2) GOTO 90
      OPEN (1,FILE='d \acad\project\fillet\DIE3 FIN',STATUS='OLD')
      GOTO 100
90     OPEN (1,FILE='d \acad\project\fillet\DIE4 FEX',STATUS='OLD')
100    READ(1,'(I4)')(J(I),I=1,7)
      READ(1,'(I4)')(K(I),I=1,9)
      READ(1,'(F5 2)') ((TOL(I,N),N=1,8),I=1,6)
      CLOSE(1)
C
C        Selecting the proper fillet radii
C
      DO 50 I=1,6
      IF ((TM GE J(I)) AND (TM LE J(I+1))) GOTO 60
50     CONTINUE
60     IL=I
      DO 70 I=1,8
      IF ((DM GE K(I)) AND (DM LE K(I+1))) GOTO 80
70     CONTINUE
80     IC=I
      T=TOL(IL,IC)
      OPEN(2,FILE='d \acad\project\fillet\FILLET DAT',STATUS='NEW')
      WRITE(2,'(F5 2)')T
      CLOSE(2)
      END
```

Appendix C

Corner radii program

```
(defun cor ()
    (edge)
    (setq op2 "Yes")
    (while (equal op2 "Yes")
    (command "fillet" pause pause)
    (initget 1 "Yes No")
    (setq op (getkword 'Do you want to do it again with the same edge radii (Yes or No) ?"))
    (if (equal op "No")
        (progn
        (initget 1 "Yes No")
        (setq op1 (getkword  Do you want to change the edge radii and continue (Yes or No) ?"))
        (if (equal op1 "No") (setq op2 "No") (edge))
        )
    )
    )
    (print "done")
)
(defun edge ()
    (initget 1 "Yes No")
    (setq op3 (getkword  Do you want automatic selection of the corner radii (Yes or No) ?"))
    (if (equal op3 "Yes") (progn
    (setq m (getdist 'Input the maximum height per die half <250 mm "))
    (setq d (getdist "\nInput the maximum diameter or the maximum width of the forging <1000 mm "))
    (setq f (open "d /acad/project/corner/edge dat  "w"))
    (print m f)
    (print d f)
    (print 99 f)
    (close f)
    (command "d /acad/project/corner/edgerad")
    (graphscr)
    (setq f (open "d /acad/project/corner/edge dat" "r"))
    (setq r (read-line f))
    (setq r (atof r))
    (close f)
                    )
                    (progn
    (setvar "filedia" 0)
        (command "vslide" "d /acad/project/corner/corner")
        (setq r (getreal "Input the radii value either from the table or from your own experience "))
    (setvar "filedia" 1)
        (redraw)
                    )
    )
    (command "fillet" "r" r)
)
(defun c cor ()
    (cor)
)
```

```
      DIMENSION J(7),K(10),TOL(6,9)
      OPEN(2,FILE='EDGE DAT',STATUS='OLD')
      READ(2,*)TM,DM
      CLOSE(2,STATUS='DELETE')
C
C           Loading the data to the memory
C
      OPEN (1,FILE='DIE2 RAD',STATUS='OLD')
      READ(1,'(I4)')(J(I),I=1,7)
      READ(1,'(I4)')(K(I),I=1,10)
      READ(1,'(F5 2)') ((TOL(I,N),N=1,9),I=1,6)
      CLOSE(1)
C
C       Selecting the proper edge radii for unmachined surfaces
C
      DO 50 I=1,6
      IF ((TM GE J(I)) AND (TM LE J(I+1))) GOTO 60
50     CONTINUE
60     IL=I
      DO 70 I=1,9
      IF ((DM GE K(I)) AND (DM LE K(I+1))) GOTO 80
70     CONTINUE
80     IC=I
      T=TOL(IL,IC)
      OPEN(2,FILE='EDGE DAT',STATUS='NEW')
      WRITE(2,'(F5 2)')T
      CLOSE(2)
      END
```

Appendix D

Draft angle program

```
(defun setup ()
        (initget 1 "Yes No")
        (setq op1 (getkword "\nDo you want to input your own draft angle(Yes or No) ?"))
        (if (equal op1 "Yes")
                (progn
                (command "vslide" "d /acad/project/draft/draft")
                (setq ad (getreal "Input the draft angle in degree  "))
                (redraw)
                (setq ad (/ (* ad pi) 180))
                )
        (progn
        (setq in (list 0 1047197 0 0523598) ex (list 0 0785398 0 0349065))
        (initget 1 "Internal External")
        (setq op (getkword "Do you want to set the internal or the external draft angle(in or ex) "))
        (if (equal op "Internal") (progn
                (initget 1 "Yes No")
                (setq op (getkword "With ejector (Yes or No)?"))
                (if (equal op "Yes") (setq ad (cadr in)) (setq ad (car in)))
                                )
                                (progn
                (initget 1 "Yes No")
                (setq op (getkword "With ejector (Yes or No)?"))
                (if (equal op "Yes") (setq ad (cadr ex)) (setq ad (car ex)))
                                )
        )
        (print "The draft angle in degrees is ")
        (setq bd (/ (* ad 180) pi))
        (print bd)
        )
        )
)
(defun c setup ()
        (setup)
)


(defun prog2 ()
        (setq op1 "Yes")
        (while (equal op1 "Yes")
                (draft)
                (initget 1 "Yes No")
                (setq op1 (getkword "Do you want to draft any other line (Yes or No) ?"))
        )
)
(defun c prog2 ()
        (prog2)
)
(defun draft ()
        (setq s (entsel "\nSelect the line to be drafted "))
        (setq l (entget (car s)) p3 (cadr s))
        (setq p4 (getpoint "\nSide to draft ?"))
        (setq x1 (car (cdr (assoc ’10 l))) y1 (car (cdr (cdr (assoc ’10 l)))))
        (setq x2 (car (cdr (assoc ’11 l))) y2 (car (cdr (cdr (assoc ’11 l)))))
        (setq p1 (list x1 y1 0 0) p2 (list x2 y2 0 0))
```

```
     (prompt "\nSelect the base line ")
     (setq s1 (ssget) e (ssname s1 0) n (entget e))
     (setq p5 (cdr (assoc '10 n)) p6 (cdr (assoc '11 n)))
     (setq x4 (car p4) y4 (cadr p4))
     (setq dp (distance p1 p2))
     (if (> (distance p1 p3) (distance p2 p3)) (progn
          (setq p p1 p1 p2 p2 p)
                                  )
     )
     (setq a (angle p1 p2))
     (setq b (angle p1 p4))
     (if (> a b) (setq a (- a ad)) (setq a (+ a ad)))
     (setq xn (+ (* dp (cos a)) (car p1)) yn (+ (* dp (sin a)) (cadr p1)))
     (setq p (inters p1 (list xn yn) p5 p6 nil))
     (command "line" p1 p)
     (command "")
     (if (or (equal p2 p5) (equal p2 p6)) (progn
          (if (equal p2 p5) (progn
                    (command "line" p p6)
                    (command "")
                  )
          )
          (if (equal p2 p6) (progn
                    (print "p2=p6")
                    (command "line" p p5)
                    (command "")
                  )
          )
          (entdel e)
                       )
     )
     (entdel (car s))
     (redraw)
)
```

Appendix E

Flash and gutter program

```
(defun flash ()
        (setq w (getreal 'Input the component's weight in Kg "))
        (setq tf (- (+ 1 13 (* 0 89 (expt w 0 5))) (* 0 017 w)))
        (setq wf (* tf (+ 3 (* 1 2 (exp (* w -1 09))))))
        (setq tg (* 1 6 tf) wg (* 4 wf) r1 tf r2 tg t1 (/ tf 2) t2 (/ tg 2))
        (setq t3 (* t2 1 4142) t4 (+ wf (- wg t2)))
,
,       Drawing the flash and gutter lands
,
        (prompt "Select the two line sides in which the flash has to be connected ")
        (setq o (ssget))
        (setq ent1 (ssname o 0) ent2 (ssname o 1))
        (setq pp (getpoint "Indicate the side?"))

        (setq n1 (cdr (assoc '0 (entget ent1))))
        (if (equal n1 'LINE") (progn
        (setq x1 (car (cdr (assoc '10 (entget ent1)))))
        (setq y1 (car (cdr (cdr (assoc '10 (entget ent1))))))
        (setq x2 (car (cdr (assoc '11 (entget ent1)))))
        (setq y2 (car (cdr (cdr (assoc '11 (entget ent1))))))
                              )
                              (progn
        (setq xc1 (car (cdr (assoc '10 (entget ent1)))))
        (setq yc1 (car (cdr (cdr (assoc '10 (entget ent1))))))
        (setq pc1 (list xc1 yc1 0 0))
        (setq r1 (cdr (assoc '40 (entget ent1))))
        (setq a1 (cdr (assoc '50 (entget ent1))))
        (setq a2 (cdr (assoc '51 (entget ent1))))
        (setq x1 (+ xc1 (* r1 (cos a1))) y1 (+ yc1 (* r1 (sin a1))))
        (setq x2 (+ xc1 (* r1 (cos a2))) y2 (+ yc1 (* r1 (sin a2))))
                              )
        )
        (setq n2 (cdr (assoc '0 (entget ent2))))
        (if (equal n2 "LINE") (progn
        (setq x3 (car (cdr (assoc '10 (entget ent2)))))
        (setq y3 (car (cdr (cdr (assoc '10 (entget ent2))))))
        (setq x4 (car (cdr (assoc '11 (entget ent2)))))
        (setq y4 (car (cdr (cdr (assoc '11 (entget ent2))))))
                              )
                              (progn
        (setq xc2 (car (cdr (assoc '10 (entget ent2)))))
        (setq yc2 (car (cdr (cdr (assoc '10 (entget ent2))))))
        (setq pc2 (list xc2 yc2 0 0))
        (setq r2 (cdr (assoc '40 (entget ent2))))
        (setq a1 (cdr (assoc '50 (entget ent2))))
        (setq a2 (cdr (assoc '51 (entget ent2))))
        (setq x3 (+ xc2 (* r2 (cos a1))) y3 (+ yc2 (* r2 (sin a1))))
        (setq x4 (+ xc2 (* r2 (cos a2))) y4 (+ yc2 (* r2 (sin a2))))
                              )
        )
        (setq p1 (list x1 y1 0 0) p2 (list x2 y2 0 0))
        (setq p3 (list x3 y3 0 0) p4 (list x4 y4 0 0))

        (cond ((equal p1 p3) (setq p1 p2 p2 p3 p3 p4))
    ((equal p1 p4) (setq p1 p2 p2 p4))
    ((equal p2 p3) (setq p3 p4))
)

(setq p4 (list (car p2) (+ (cadr p2) t1)))
(if (equal n1 "LINE")
(setq p4 (inters p1 p2 p4 (list 500 (cadr p4)) nil))
(progn
(if (or (> (car pp) (car p1)) (> (car pp) (car p3)))
(setq x4 (+ xc1 (sqrt (- (expt r1 2) (expt (- (cadr p4) yc1) 2)))))
(setq x4 (- xc1 (sqrt (- (expt r1 2) (expt (- (cadr p4) yc1) 2)))))
)
(setq p4 (list x4 (cadr p4)))
)
)
(setq p5 (list (car p2) (- (cadr p2) t1)))
(if (equal n2 "LINE")
(setq p5 (inters p3 p2 p5 (list 500 (cadr p5)) nil))
(progn
(if (or (> (car pp) (car p1)) (> (car pp) (car p3)))
(setq x5 (+ xc2 (sqrt (- (expt r2 2) (expt (- (cadr p5) yc2) 2)))))
(setq x5 (- xc2 (sqrt (- (expt r2 2) (expt (- (cadr p5) yc2) 2)))))
)
(setq p5 (list x5 (cadr p5)))
)
)
(entdel ent1)
(entdel ent2)
(if (or (> (car pp) (car p1)) (> (car pp) (car p3)))
      (progn
      (setq p6 (list (+ (car p4) wf) (cadr p4)))
      (setq p7 (list (+ (car p5) wf) (cadr p5)))
      (setq p8 (polar p6 1 0471976 (- t2 t1)))
      (setq p9 (list (+ (car p4) (- t4 t2)) (+ (cadr p4) (- t2 t1))))
      )
      (progn
      (setq p6 (list (- (car p4) wf) (cadr p4)))
      (setq p7 (list (- (car p5) wf) (cadr p5)))
      (setq p8 (polar p6 2 0943951 (- t2 t1)))
      (setq p9 (list (- (car p4) (- t4 t2)) (+ (cadr p4) (- t2 t1))))
      )
)
      (setq p (list 0 0 (cadr p9)))
      (setq p8 (inters p6 p8 p p9 nil))
      (if (equal n1  LINE ) (progn
                        (command  line  p1 p4)
                        (command "")
                         )
                        (progn
(if (or (> (car pp) (car p1)) (> (car pp) (car p3)))
   (progn
                        (command "arc" "c' pc1 p4 p1)
                        (command "")
   )
   (progn
                        (command "arc" "c" pc1 p1 p4)
                        (command "")
      )
    )
                    )
        )
        (command "LINE" p4 p6 p8 p9)
        (command "")
        (setq p (list (car p9) (- (cadr p9) t2)))
        (setq p6 (list (car p9) (- (cadr p9) tg)))
    (if (or (> (car pp) (car p1)) (> (car pp) (car p3)))
        (progn
        (command 'arc  "c" p p6 p9)
        (command "")
        )
        (progn
        (command "arc" "c" p p9 p6)
        (command "")
        )
    )
        (setq p4 (list (car p8) (- (cadr p8) tg)))
        (command "line" p6 p4 p7 p5)
        (command ' ")
    (if (equal n2 "ARC") (progn
    (if (or (> (car pp) (car p1)) (> (car pp) (car p3)))
                (command "arc" "C" pc2 p3 p5)
                (command "arc" "c" pc2 p5 p3)
        )
                (command '  )
              )
              (progn
                (command  line  p3 p5)
                (command "")
              )
    )
)
(defun c flash ()
    (flash)
)
```

Appendix F

Mesh generation program

```
(defun mesh ()
    (command "osnap" "end')
    (setq n1 (getint "Type 1 to mesh a new shape or 2 to optimize  one & 3 for remesh "))
    (if (= n1 1)
        (progn
    (setq f (open "mesh dat" "w"))
    (setq s (getstring "Input the title of the case "))
    (write-line s f)
    (print n1 f)
    (initget (+ 1 2 4))
    (setq nbloc (getint "Input the number of blocks ' ))
    (print nbloc f)
    (setq npoi (getint "Input the number of points which form the blocks "))
    (print npoi f)
    (setq nnode (getint "Do you want 4-Element or 3-Element node(4/3) ?"))
    (print nnode f)
    (if (= nnode 3) (progn
            (setq ndiag (getint  Type 1 to divide the element by it's long diagonal or 2 for short
diagonal "))
            (print ndiag f)
            )
    )
    (setq n 1 l (list "con') ll (list "con"))
    (prompt "Start digitizing the points which form the blocks ")
    (while (<= n npoi)
        (print n f)
        (setq pt (getpoint))
        (print (car pt) f)
        (print (cadr pt) f)
        (setq p (list n (car pt) (cadr pt)))
        (setq l (cons p l))
        (setq n (+ n 1))
    )
    (setq l (reverse l))
    (command "osnap" "none")
    (redraw)
;--------------------------------------------------;
;           The connectivities of the blocks      ;
;--------------------------------------------------;
    (setq n 1)
    (while (<= n nbloc)
    (command "osnap" "end")
    (prompt "\n The block  ")
    (print n)
    (setq m (getint "\nInput the number of material block  ))
    (prompt "\ndigitize the connictivity of block No ")
    (print n)
    (print n f)
    (print m f)
;
        (setq i 1)
        (while (<= i 8)
            (setq j 1)
            (setq k1 0)
            (setq pt (getpoint "Point No "))
            (print i)
            (while (<= j npoi)
            (setq x (car (cdr (nth j l))))
            (setq y (car (cdr (cdr (nth j l)))))
            (if (and (= (car pt) x) (= (car (cdr pt)) y))
            (progn
                (setq k (car (nth j l)))
                (print k f)
                (setq k1 1)
            )
            )
            (setq j (+ j 1))
            )
            (if (= k1 0)
            (progn
            (prompt '\n It is the wrong point! Pick the point again ')
            (setq i (- i 1))
            )
        (setq l1 (cons pt l1))
            )
,
,
        (setq i (+ i 1))
        )
    (setq l1 (reverse l1))
    (command 'osnap' 'none')
    (redraw)
    (prompt "\nSelect the 8 sides of the block")
    (setq ss (ssget))
    (setq ne 0 i 1)
    (while (<= ne 7)
        (setq d (entget (ssname ss ne)))
        (setq s2 (cdr (assoc '0 d)))
        (if (= s2 "ARC")
            (progn
            (setq x1 (car (nth (+ ne 1) l1)))
            (setq y1 (car (cdr (nth (+ ne 1) l1))))
            (if (= ne 6) (progn
            (setq x2 (car (nth 1 l1)))
            (setq y2 (car (cdr (nth 1 l1))))
            )
            (progn
            (setq x2 (car (nth (+ ne 3) l1)))
            (setq y2 (car (cdr (nth (+ ne 3) l1))))
            )
            )
            (print i f)
            (setq xc (car (cdr (assoc '10 d))))
            (print xc f)
            (setq yc (car (cddr (assoc '10 d))))
            (print yc f)
            (setq r (cdr (assoc '40 d)))
            (print r f)
            (print x1 f)
            (print y1 f)
            (print x2 f)
```

```
                (print y2 f)
                )
            (print 0 f)
            )
            (setq i (+ i 1) ne (+ ne 2))
    )
    (redraw)
    (setq n (+ n 1))
    )
;-------------------------------------------------------;
;   Input the division in X and Y for each block        ;
;-------------------------------------------------------;
    (setq n 1)
    (while (<= n nbloc)
            (setq i 1)
            (prompt "\nInput the division in X for block No ")
            (print n)
            (setq dx (getint))
            (print n f)
            (print dx f)
            (prompt "\nInput the proportional division in X for each part ")
            (while (<= i dx)
                (prompt "\nFor division number ")
                (print i)
                (setq a (getreal))
                (print a f)
                (setq i (+ i 1))
            )
            (prompt "\nInput the division in Y for block No ")
            (print n)
            (setq dy (getint))
            (print dy f)
            (prompt "\nInput the proportional division in Y for each part ")
            (setq i 1)
            (while (<= i dy)
                (prompt "\nFor division number ")
                (print i)
                (setq a (getreal))
                (print a f)
                (setq i (+ i 1))
            )
            (setq n (+ n 1))
    )
    (print "end" f)
    (close f)
    (command"meshg")
    (graphscr)
    (command "layer" "new" "mesh")
    (command "")
    (command "layer" "set" "mesh")
    (command "")
    (command "dxfin" "mesh")
            )
    (progn
    (setq k (findfile "mesh.inp"))
    (if (= k nil) (prompt "\nThe file of the first meshing does not exist ")
            (progn
                (command "ren    mesh dat mesh old' )
                (command "cop" "mesh ımp mesh dat ')
                (command   meshg")
                (graphscr)
                (command "layer" "new '  'mesh")
                (command "")
                (command ' layer" "set" "mesh")
                (command  ' )
                (command "dxfin" "mesh' )
                )
        )
        )
        )
)
(defun C mesh ()
        (mesh))
```

Appendix G

Remeshing program

```
      PROGRAM REMESH
C*****************************************************
      IMPLICIT REAL*8 (A H O Z)  INTEGER*4 (I N)
      DIMENSION
      COORD(2 100),LNODS(4 100) STRT(100) STRT1(100)
      DIMENSION
      COORD1(2 4) W(2) S2(2) SHAPE(4) COORD2(2 100)
   1          ,LNOD(4 100),ESTRT(100) COORD3(2 4)
      DATA S2/ 0 57735026918963D0 0 57735026918963D0/
      DATA W/2*1 0D0/

C      Read the data of the old mesh
C           1  The coordinate of each node
C           2  The effective strain for each element
C
      CALL INPUT (COORD,LNODS STRT NPOIN NELEM
                COORD2,NPOIN1,LNOD,NELEM1)


C
      NNODE=4
      DO 5 INODE=1,NPOIN
      STRT1(INODE)=0 0D0
5      CONTINUE

      DO 10 IPOIN=1 NPOIN
      AREA=0 0D0
      JPOIN=IPOIN
C       WRITE(6 *)IPOIN
      UP=0 0D0
      DO 20 IELEM=1 NELEM
      DO 30 INODE=1,NNODE
      IF(IPOIN EQ LNODS(INODE IELEM)) THEN
      SS=STRT(IELEM)
      A=0 0D0

      DO 40 I=1 4
      NE=LNODS(I IELEM)
      COORD1(1 I)=COORD(1,NE)
      COORD1(2 I)=COORD(2,NE)
40     CONTINUE

      DO 50 I=1 2
      S=S2(I)
      DO 60 J=1 2
      T=S2(J)

C
C      Calculate the Jacobian matrix for the old element
C
      CALL JACOB (COORD1 WDXJ S T)

      CALL SHAPE4 (T S SHAPE)
      DO 70 II=1 4
C
C      Calculate the area of a shared element to a particular node
C
      A=A+W(1)*W(2)*WDXJ*SHAPE(II)
70     CONTINUE
60     CONTINUE
50     CONTINUE

      UP=UP+SS*A

C       WRITE(6 *)SS A
      ELSE
      SS=0 0D0
      GO TO 30
      END IF
C
```

```
C      Calculate the the sum of the element area which share
C                  the same node
C
      AREA=AREA+A
30     CONTINUE
20     CONTINUE
C       WRITE(6 *) IPOIN,AREA
C       WRITE(6 *) IPOIN UP
C
C      Calculate the effective strain at the new node
C
      STRT1(IPOIN)=UP/AREA
C       WRITE(6 *) IPOIN STRT1(IPOIN)
10     CONTINUE

      DO 80 IELEM=1,NELEM
C       WRITE(6 *)  ELEMENT NO
C       WRITE(6 *) IELEM
      DO 100 I=1 4
      NE=LNODS(I IELEM)
      COORD1(1 I)=COORD(1,NE)
      COORD1(2 I)=COORD(2,NE)
100    CONTINUE

      DO 90 JELEM=1,NELEM1
      X=0 0
      Y=0 0
      E=0 0
      DO 120 I=1 4
      NE1=LNOD(I JELEM)
      COORD3(1 I)=COORD2(1,NE1)
      COORD3(2 I)=COORD2(2,NE1)
120    CONTINUE
      T=0 0D0
      S=0 0D0
      CALL SHAPE4(T S SHAPE)
      DO 110 L=1 4
      X=X+SHAPE(L)*COORD3(1,L)
      Y=Y+SHAPE(L)*COORD3(2,L)
110    CONTINUE
C
C       Check the centers of new element which are located in a
particular
C                   old element
C
      CALL FIND (COORD1,X Y,JELEM,PSI,ETA)

      IF (PSI GE  1 0D0 AND PSI LE 1 0D0.AND
    1   ETA GE  1 0D0 AND ETA LE 1 0D0) THEN
C      WRITE(6 *)PSI ETA

      CALL SHAPE4 (ETA,PSI SHAPE)
      DO 130 I=1 4
      NE=LNODS(I IELEM)
      E=E+STRT1(NE)*SHAPE(I)
130    CONTINUE

      ESTRT(JELEM)=E

      ELSE
      END IF

90     CONTINUE
80     CONTINUE
      OPEN (2,FILE= RES DAT  STATUS= UNKNOWN )
      DO 140 I=1 NELEM1
      WRITE(2 *)I,ESTRT(I)
140    CONTINUE
      CLOSE(2)
      OPEN (3,FILE= RES DXF  STATUS= UNKNOWN )
      CALL CONT (ESTRT COORD2,LNOD NELEM1,NPOIN1)
      CLOSE (3)


      END

      SUBROUTINE FIND (COORD,X Y IPOIN PSI,ETA)
C
C      TO CHECK THE NEW NODES WHICH ARE C
    CONTAINED IN EACH ELEMENT OF THE OLD MESH
C
      IMPLICIT REAL*8 (A H O Z)  INTEGER*4 (I N)
      DIMENSION COORD(2 4)
      ET(PSI)= ((A1+B1)+(A3+B3)*PSI (X+Y))/
            ((A2+B2)+(A4+B4)*PSI)

      A1=(COORD(1 1)+COORD(1 2)+COORD(1 3)
        +COORD(1 4))*0 25D0
      A2=( COORD(1 1)-COORD(1 2)+COORD(1 3)
        +COORD(1 4))*0 25D0
      A3=( COORD(1 1)+COORD(1 2)+COORD(1 3)
         COORD(1 4))*0 25D0
      A4=(COORD(1 1)-COORD(1 2)+COORD(1 3)
         COORD(1 4))*0 25D0
      B1=(COORD(2 1)+COORD(2 2)+COORD(2 3)
        +COORD(2 4))*0 25D0
      B2=( COORD(2 1) COORD(2 2)+COORD(2 3)
        +COORD(2 4))*0 25D0
      B3=( COORD(2 1)+COORD(2 2)+COORD(2 3)
         COORD(2 4))*0 25D0
      B4=(COORD(2 1)-COORD(2 2)+COORD(2 3)
         COORD(2 4))*0 25D0


      A=A3*B4 A4*B3
      B=B2*A3+B4*(A1 X) A2*B3 A4*(B1 Y)
      C=B2*(A1 X) A2*(B1 Y)

      PSI1=5 0D0
      PSI2=5 0D0
      ETA=5 0D0
      JPOIN=0

      IF(A EQ 0 0D0) THEN

      IF(C NE 0 0D0) GO TO 90
      PSI1=0 0D0
      PSI2=PSI1
      GO TO 30
90     IF(B EQ 0 0D0) GO TO 50
      PSI1= C/B
      PSI2=PSI1
      GO TO 30

      ELSE

      IF(B NE 0 0D0) GO TO 10

      IF(C EQ 0 0D0) THEN
      PSI1=0 0D0
      PSI2=PSI1
      GO TO 30
      ELSE
      P1= C/A
      IF(P1 LT 0 0D0) GO TO 50
      PSI1=DSQRT(P1)
      PSI2= PSI1
      GO TO 30
      END IF

10     IF(C NE.0 0D0) GO TO 100
      PSI1=0 0D0
      PSI2= B/A
      GO TO 30
100    CONTINUE
      DELTA=B**2 4*A*C
      IF(DELTA LT 0 0D0) GO TO 50
      IF(DELTA EQ 0 0D0) GO TO 20

      PSI1=( B+DSQRT(DELTA))/(2*A)
      PSI2=( B DSQRT(DELTA))/(2*A)

      GO TO 30
20     PSI1= B/(2*A)
      PSI2=PSI1
      GO TO 30
      END IF

30     IF (PSI1 GE  1 0D0 AND PSI1 LE 1 0D0) THEN
      PSI=PSI1
      ETA=ET(PSI)
      IF (ETA GE  1 0D0 AND ETA LE 1 0D0) GO TO 40
      ETA=ET(PSI)
      IF (ETA GE  1 0D0 AND ETA LE 1 0D0) GO TO 40
      ELSE
      GO TO 60
      END IF

60     CONTINUE
      IF (PSI2 GE  1 0D0 AND PSI2 LE.1 0D0) THEN
      PSI=PSI2
      ETA=ET(PSI)
      IF (ETA GE  1 0D0 AND ETA LE 1 0D0) GO TO 40
      ETA=ET(PSI)
      IF (ETA GE  1 0D0 AND ETA LE 1 0D0) GO TO 40
      ELSE
      GO TO 50
      END IF
      GO TO 50
40     JPOIN=IPOIN
      WRITE(6 *)JPOIN
50     CONTINUE
      RETURN
      END


SUBROUTINE INPUT (COORD LNODS STRT,NPOIN,NELEM
           COORD2,NPOIN1,LNOD,NELEM1)
C
C     THIS SUBROUTINE IS TO READ THE INPUT DATA
C
      IMPLICIT REAL*8 (A H O Z)  INTEGER*4 (I N)
      DIMENSION COORD(2 100) LNODS(4 100) STRT(100)
           COORD2(2 100)
      DIMENSION LNOD(4 100)

      NNODE=4
      OPEN(1,FILE= REM DAT  STATUS= OLD )
C
C     Read the coordinate nodes of the old mesh
C
      READ(1 *)NPOIN
      DO 10 INODE=1,NPOIN
      READ(1 *) J (COORD(N INODE),N=1 2)
10     CONTINUE
C
C     Read the conectivities of the old mesh
C
      READ(1 *)NELEM
      DO 20 IELEM=1 NELEM
      READ(1,*) J,(LNODS(N,IELEM),N=1 NNODE)
20     CONTINUE
C
C     Read the effective strain of the old mesh
C
```

```
      DO 30 IELEM=1 NELEM
      READ(1 *) J STRT(IELEM)
30     CONTINUE
C
C      Read the coordinate nodes of the new mesh
C
      READ(1 *) NPOIN1
      DO 15 INODE=1,NPOIN1
      READ(1 *) J (COORD2(N INODE),N=1 2)
15     CONTINUE
C
C      Read the conectivities of the new mesh
C
      READ(1 *)NELEM1
      DO 40 IELEM=1 NELEM1
      READ(1 *) J (LNOD(N IELEM),N=1,NNODE)
      WRITE(6 *) J (LNOD(N IELEM),N=1 NNODE)
40     CONTINUE

      CLOSE(1)
      RETURN
      END


      SUBROUTINE JACOB (COORD WDXJ S T)
      IMPLICIT DOUBLE PRECISION (A H O Z)

C  EVALUATE THE AREA OF QUADRILATERAL ELEMENT

C          COORD(2 4)          NODE COORDINATES
C          (S T)          NATURAL COORDINATE

      DIMENSION COORD(2 4)


      R12=COORD(1 1)-COORD(1 2)
      R13=COORD(1 1) COORD(1 3)
      R14=COORD(1 1) COORD(1 4)
      R23=COORD(1 2)-COORD(1 3)
      R24=COORD(1 2) COORD(1 4)
      R34=COORD(1 3)-COORD(1 4)

      Z12=COORD(2 1) COORD(2 2)
      Z13=COORD(2 1)-COORD(2 3)
      Z14=COORD(2 1) COORD(2 4)
      Z23=COORD(2 2) COORD(2 3)
      Z24=COORD(2 2) COORD(2 4)
      Z34=COORD(2 3) COORD(2 4)

      DXJ8=((R13*Z24 R24*Z13)+(R34*Z12 R12*Z34)*S+
    1       (R23*Z14 R14*Z23)*T)
      WDXJ=DXJ8/8




      RETURN
      END

      SUBROUTINE SHAPE4 (ETA,PSI SHAPE)
C
C
C  CALCULATE THE SHAPE FUNCTION FOR
C         THE SHARED NODE
C
C
      IMPLICIT INTEGER*4 (I N)  REAL*8 (A H O Z)
      DIMENSION SHAPE(4)

      S=PSI
      T=ETA
      ST=S*T
```

```
      SHAPE(1)=(1  T S+ST)*0 25
      SHAPE(2)=(1  T+S ST)*0 25
      SHAPE(3)=(1 +T+S+ST)*0 25
      SHAPE(4)=(1 +T S ST)*0 25

      RETURN
      END

      SUBROUTINE CONT (F,RZ,NOD,NUMEL,NUMNP)
C
      IMPLICIT DOUBLE PRECISION (A H O-Z)

      DIMENSION F(100) RZ(2 100),NOD(4 100)
    1          XE(4) YE(4),FE(4) FCONT(10),EX(99) EY(99)
    2          IARY1(6),F1(100),ND(3 100) ITXT(10)
      DATA IARY1/1 2 2 3 3 1/

      WRITE(3  (3H  0) )
      WRITE(3  (7HSECTION) )
      WRITE(3  (3H  2) )
      WRITE(3  (8HENTITIES) )
      WRITE(3  (3H  0) )

      DO 85 I=1 10
85     ITXT(I)=0
      XORG=0 0
      YORG=0 0


      CALL ELTOND (RZ,NOD F,F1,NUMEL,NUMNP)


                         CALL      GSCALE
(NUMNP,RZ,XMIN YMIN XMAX YMAX,SCALE)

C
C      CALCULATE THE HIGHT OF THE TEXT
C
      DY=YMAX YMIN
      DX=XMAX XMIN
      HI=(DY+DX)/60

      J=0
      DO 21 I=1 NUMEL
      J=J+1
      ND(1,J)=NOD(1 I)
      ND(2,J)=NOD(2 I)
      ND(3,J)=NOD(3 I)
      J=J+1
      ND(1,J)=NOD(1 I)
      ND(2,J)=NOD(3 I)
      ND(3,J)=NOD(4 I)
21     CONTINUE

C
C      Determine the interval of the contour line
C
      NNODE=3
      FMIN=1 E20
      FMAX= 1 E20
      DO 10 I=1 NUMNP
      FI=F1(I)
      IF (FI GT FMAX) FMAX=FI
10     IF (FI LT FMIN) FMIN=FI
      EPP=0 00001*(FMAX FMIN)
C
C      Calculate the values of the contour lines
C
      NCONT=7
      DF=(FMAX FMIN)/6
      FF=FMIN
      FCONT(1)=FF
      DO 15 I=2 NCONT
      FF=FF+DF
15    FCONT(I)=FF

C
C     Write the contour line in DXF format
C
      DO 20 IELEM=1 NUMEL*2

      DO 30 I=1,NNODE
      INOD=ND(I IELEM)
      XE(I)=RZ(1 INOD)
      YE(I)=RZ(2 INOD)
30    FE(I)=F1(INOD)

      DO 50 N=1 NCONT
      FSI=FCONT(N)
      LIN=1

      DO 60 J=1 NNODE
      J1=2*(J 1)+1
      J2=J1+1
      J1A=IARY1(J1)
      J2A=IARY1(J2)
      XE1=XE(J1A)
      YE1=YE(J1A)
      XE2=XE(J2A)
      YE2=YE(J2A)
      FE1=FE(J1A)
      FE2=FE(J2A)
      IF (FE2 FE1 GT EPP) GO TO 300
      IF (FE1 FE2 GT EPP) GO TO 400
      GO TO 500
300   IF (FSI GT FE2 OR FSI LT FE1) GO TO 60
      GO TO 600
400   IF (FSI GT FE1 OR FSI LT FE2) GO TO 60
600   TA=(FSI FE2)/(FE1 FE2)
      EX(LIN)=(XE2+TA*(XE1 XE2) XMIN)+XORG
      EY(LIN)=(YE2+TA*(YE1 YE2) YMIN)+YORG
      LIN=LIN+1
      GO TO 60
500   IF (ABS(FSI FE1) GT EPP) GO TO 60
      EX(1)=(XE1 XMIN)+XORG
      EY(1)=(YE1 YMIN)+YORG
      EX(2)=(XE2 XMIN)+XORG
      EY(2)=(YE2 YMIN)+YORG
      LIN=3
60    CONTINUE

      LIN1=LIN 1
      IF (LIN GE.3) THEN
      CALL DXFC (LIN1,EX EY,N,FCONT ITXT HI)
      ELSE
      END IF
50    CONTINUE
20    CONTINUE
      WRITE(3 (6HENDSEC) )
      WRITE(3 (3H  0) )
      WRITE(3 (3HEOF) )

      RETURN
      END
      SUBROUTINE DXFC (K,EX EY,N FCONT ITXT,HI)
C
C
C         THIS SUBROUTINE IS TO CREATE THE
C            DXF FILE FOR THE CONTOUR
C
      IMPLICIT REAL*8 (A H O Z) INTEGER*4 (I N)
      DIMENSION EX(99),EY(99) FCONT(10) ITXT(10)
C
C
C     CREATING THE DXF FILE
C
```

```
C
      DO 20 JNODE=1 K 1
      WRITE(3 (4HLINE) )
      WRITE(3 (3H  8) )
      WRITE(3 (1H2) )
      WRITE(3 (3H 62) )
      WRITE(3 *)N
      WRITE(3 (3H 10) )
      WRITE(3 (F10 6) )EX(JNODE)
      WRITE(3 (3H 20) )
      WRITE(3 (F10 6) )EY(JNODE)
      WRITE(3 (3H 30) )
      WRITE(3 (3H0 0) )

C       IF (JNODE EQ K) THEN
C       NODE=1
C       ELSE
C       NODE=JNODE+1
C       END IF

      WRITE(3 (3H 11) )
      WRITE(3 (F10 6) )EX(JNODE+1)
      WRITE(3 (3H 21) )
      WRITE(3 (F10 6) )EY(JNODE+1)
      WRITE(3 (3H 31) )
      WRITE(3 (3H0 0) )
      WRITE(3 (3H  0) )

      IF (ITXT(N) EQ 0) THEN
      WRITE(3 (4HTEXT) )
      WRITE(3 (3H  8) )
      WRITE(3 (1H2) )
      WRITE(3 (3H 62) )
      WRITE(3 *)N
      WRITE(3 (3H 10) )
      WRITE(3 (F10 6) )EX(JNODE)
      WRITE(3 (3H 20) )
      WRITE(3 (F10 6) )EY(JNODE)
      WRITE(3 (3H 30) )
      WRITE(3 (3H0 0) )
      WRITE(3 (3H 40) )
      WRITE(3 (F10 6) )HI
      WRITE(3 (3H  1) )
      WRITE(3 (F10 6) )FCONT(N)
      WRITE(3 (3H  0) )
      ITXT(N)=1
      ELSE
      GO TO 20
      END IF
20    CONTINUE
      RETURN
      END

      SUBROUTINE GSCALE (NUMNP RZ,XMIN YMIN
     &                   XMAX YMAX,ASIZE)
      IMPLICIT DOUBLE PRECISION (A H O-Z)
C
      DIMENSION RZ(2 100)

      XMIN=1 E20
      YMIN=1 E20
      XMAX= 1 E20
      YMAX= 1 E20

      DO 10 I=1 NUMNP
      XI=RZ(1 I)
      IF (XI LT XMIN) XMIN=XI
      IF (XI GT XMAX) XMAX=XI
      YI=RZ(2 I)
      IF (YI LT YMIN) YMIN=YI
10    IF (YI GT YMAX) YMAX=YI
      XSIZE=XMAX XMIN
      YSIZE=YMAX YMIN
      IF (YSIZE GE XSIZE) THEN
      ASIZE=YSIZE
      ELSE
      ASIZE=XSIZE
      END IF
      RETURN
      END

      SUBROUTINE ELTOND (RZ,NOD,F1,F NELEM,NPOIN)
C*****************************************************
      IMPLICIT DOUBLE PRECISION (A H O Z)
      DIMENSION RZ(2 100) NOD(4 100) F1(100) F(100)
      DIMENSION RZ1(2 4) W(2) S2(2) SHAPE(4)
      DATA S2/ 0 57735026918963D0 0 57735026918963D0/
      DATA W/2*1 0D0/


      NNODE=4
      DO 5 INODE=1,NPOIN
      F(INODE)=0 0D0
5      CONTINUE

      DO 10 IPOIN=1 NPOIN
      AREA=0 0D0
      JPOIN=IPOIN
      UP=0 0D0
      DO 20 IELEM=1 NELEM
      DO 30 INODE=1,NNODE
      IF(IPOIN EQ NOD(INODE IELEM)) THEN
      SS=F1(IELEM)
      A=0 0D0

      DO 40 I=1 4
      NE=NOD(I IELEM)
      RZ1(1 I)=RZ(1 NE)
      RZ1(2 I)=RZ(2 NE)
40     CONTINUE

      DO 50 I=1 2
      S=S2(I)
      DO 60 J=1 2
      T=S2(J)

C
C      Calculate the Jacobian matrix for the  element.
C
      CALL JACOB (RZ1 WDXJ S T)

      CALL SHAPE4 (T S SHAPE)
      DO 70 II=1 4
C
C      Calculate the area of a shared element to a particular node
C
      A=A+W(1)*W(2)*WDXJ*SHAPE(II)
70     CONTINUE
60     CONTINUE
50     CONTINUE

      UP=UP+SS*A

      ELSE
      SS=0 0D0
      GO TO 30
      END IF
C
C      Calculate the the sum of the element area which share
C                  the same node
C
      AREA=AREA+A
30     CONTINUE
20     CONTINUE
C
C      Calculate the effective strain at the new node
C
      F(IPOIN)=UP/AREA
10     CONTINUE
      RETURN
      END
```

Appendix H

Rigid plastic finite element program

```
      PROGRAM FEM
      IMPLICIT DOUBLE PRECISION (A H O Z)
      CHARACTER TITLE*70
      COMMON /TITL/ TITLE
      COMMON /RIGD/ RTOL,ALPH,DIAT IPLAS STK,EXN
      COMMON /CNEQ/ NEQ MBAND
      COMMON /RVA1/ RZ(2 250) URZ(2 250) FRZ(2 250)
                DCOORD(2 100)
      COMMON /RVA2/ EPS(5 200) STS(5 200) TEPS(200)
      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
                NSIDE URD(2 100)
      COMMON /INOT/ INPT MSSG IUNIT IUNI2 ISCRN
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /INVR/ NOD(4 200) LNBC(2 250)
                NBCD(2 250),LOC(250)
      COMMON /TSTP/ NINI,NCUR NSEND,NITR DTMAX
      COMMON /ITRC/ ITYP ICONV
      COMMON /T/ W11 W22
      COMMON /BNOD/ NTOT NB1(2 250)
      COMMON /FILE/ MESHD,NODED ELEMENTD
      INPT=5
      MSSG=2
      IUNIT=3
      IUNI2=4
      ISCRN=6

C      Read Input

      CALL INPRED
      OPEN (IUNIT FILE= FEM OUT
         FORM= FORMATTED  STATUS= UNKNOWN )
      OPEN (MSSG  FILE= FEM MSG
         FORM= FORMATTED  STATUS= UNKNOWN )
      WRITE (MSSG 1020) TITLE
      WRITE (ISCRN 1020) TITLE


      CALL PRTINP
      CALL BNODE (NUMEL,NOD NTOT,NB1)
      TT=DTMAX

      CALL BAND (NOD,NUMEL,NUMNP)

C      Step Solutions

      NINI=NINI+1
      U=0 0
      DO 300 N=NINI NSEND

      CALL GLTOL
      NCUR=N

      WRITE (MSSG 1050) N
      WRITE (ISCRN 1050) N
      IF (N NE NINI) GOTO 80
      ICOUNT=0
50     ITYP=2

      CALL NONLIN
      ICOUNT=ICOUNT+1

80     ITYP=1
```

```
      CALL NONLIN

      IF (ICONV EQ 2 AND ICOUNT GT 50) GOTO 900
      IF (ICONV EQ 2) GOTO 50

      CALL CONTACT
      CALL LTOGL
      CALL POTSOL
      CALL PRTSOL (U)
      DTMAX=TT
      CALL RSTFIL
      IREM=0
      CALL REMESH (NCUR IREM)
      IF (IREM EQ 1) STOP
300    CONTINUE

      CLOSE (IUNIT)
      STOP

900    CONTINUE
      WRITE (MSSG 1070)
      WRITE (ISCRN 1070)
      CLOSE (MSSG)
      STOP

1020   FORMAT (1H1,// 5X  OUTPUT OF FEM  //
   1             5X    MESSAGE FILE FOR  / 5X,A,//)
1050   FORMAT (///     ITERATION
                 PROCESS FOR STEP    ,I5 //)
1070   FORMAT (/     STOP BECAUSE SOLUTION
              DOES NOT CONVERGE    )
      END



      SUBROUTINE ADDBAN (B,A,NQ,LM QQ,PP)
      IMPLICIT DOUBLE PRECISION (A H O Z)

C  ASSEMBLE GLOBAL STIFFNESS MATRIX FROM
C      ELEMENTAL STIFFNESS MATRIX

      DIMENSION B(1),A(NQ 1) QQ(1) PP(8 8),LM(1)

      DO 100 I=1 8
      II=LM(I)
      DO 50 J=1 8
      JJ=LM(J) LM(I)+1
      IF(JJ LE 0) GOTO 50
      A(II,JJ)=A(II JJ)+PP(I J)
50     CONTINUE
      B(II)=B(II)+QQ(I)
100    CONTINUE
      RETURN
      END


      SUBROUTINE BAND (NOD NUMEL NUMNP)
      IMPLICIT DOUBLE PRECISION (A H O Z)

C
C
C          DETERMINE
C                 MAXIMUM HALF BANDWIDTH MBAND
```

```
C             TOTAL NUMBER OF EQUATIONS   NUMEQ
C
C
      COMMON /CNEQ/ NEQ MBAND
      DIMENSION NOD (4 1)

      MBAND=0
      DO 100 N=1 NUMEL
      NMIN=NOD(1 N)
      NMAX=NOD(1,N)

      DO 50 I=2 4
      IF (NMIN GT NOD(I,N)) NMIN=NOD(I,N)
      IF (NMAX LT NOD(I N)) NMAX=NOD(I,N)
50     CONTINUE

      MB=(NMAX NMIN+1)*2
      IF (MBAND LT MB) MBAND=MB
100    CONTINUE

      NEQ=NUMNP*2

      RETURN
      END


      SUBROUTINE BANSOL ( B A NQ MM )

      IMPLICIT DOUBLE PRECISION(A H O Z)
C *****************************************************
C     *                                           *
C     *              [A] [X] = [B]                   *
C     *                                           *
C*****************************************************
C     VARIABLES
C  A = COEF  MATRIX  SYMETRIC  BANDED  POSIT  DEF
C             B = LOAD MATRIX    INPUT
C             SOLUTION MATRIX    OUTPUT
C  NQ = NUMBER OF EQUATIONS IN COEF  MATRIX
C             MM = BAND WIDTH
C  NQ = MAX  NUMBER OF LINES AT THE COEF  MATRIX
C
C
      COMMON /INOT/ INPT MSSG IUNIT IUNI2 ISCRN
      DIMENSION A(NQ 1) B(1)

      NRS = NQ 1
      NR = NQ
      DO 120  N= 1 NRS
      M = N 1
      MR = MIN0( MM   NR M )
      PIVOT = A( N   1 )
      DO 120  L= 2 MR
      CP = A( N L ) / PIVOT
      I = M+L
      J = 0
      DO 110  K= L MR
      J=J+1
110    A( I,J ) = A ( I,J )   CP * A ( N,K )
120    A( N,L ) = CP
      DO 220  N= 1 NRS
      M = N 1
      MR = MIN0( MM   NR M )
      CP = B( N )
      B( N ) = CP / A( N 1 )
      DO 220  L= 2 MR
      I = M+L
 220   B( I ) = B( I )   A ( N L ) *CP
      B( NR ) = B ( NR ) / A( NR 1 )
      DO 320  I= 1 NRS
      N = NR   I
      M = N   1
      MR = MIN0( MM NR M )
```

```
      DO 320  K= 2 MR
      L = M + K
 320   B( N ) = B( N )   A( N,K ) * B( L )
      RETURN
      END


      SUBROUTINE BNODE (NELEM,NOD NTOT,NB1)
      IMPLICIT DOUBLE PRECISION (A H O-Z)
C==============================================
C                                              C
C     This subroutine is to define the boundary nodes      C
C                                              C
C     NELEM  Total number of element                  C
C     NOD  The element conectivity                  C
C     NB     An array to save the element side which are on  C
C            the boundary                          C
C     NTOT   Total number of element/node side.           C
C     NAD    The node boundary                        C
C                                              C
C==============================================

      DIMENSION NOD(4 200) NBOUND(250),NB(2,250)
              NAD(200),NB1(2 250)


      NTOT=0
      DO 10 IELEM=1 NELEM
      DO 10 I=1 4
      I1=NOD(I IELEM)
      I3=I+1
      IF (I3 GE 5) I3=1
      I2=NOD(I3 IELEM)

      N=0
      K=1
20     N=N+1
      IF (N GT NELEM) GO TO 40
      DO 30 J=1 4
      J1=NOD(J N)
      J3=J+1
      IF (J3 GE 5) J3=1
      J2=NOD(J3 N)
      IF (I1 EQ J2 AND I2 EQ J1) K=0
30     CONTINUE
      IF (K EQ 0) GO TO 10
      GO TO 20

40     NTOT=NTOT+1
      NBOUND(2*NTOT 1)=I1
      NBOUND(2*NTOT)=I2
10     CONTINUE

C
C     SELECT THE ELEMENT SIDE BOUNDARY
C

      J=1
      DO 50 I=1 NTOT
      NB(1 I)=NBOUND(J)
      NB(2 I)=NBOUND(J+1)
      J=J+2
50     CONTINUE

      NB1(1 1)=NB(1 1)
      NB1(2 1)=NB(2 1)
      L=1
      DO 100 I=1 NTOT
      DO 110 J=1 NTOT
      IF (NB(2,L) EQ NB(1,J)) THEN
      NB1(1 I+1)=NB(1,J)
      NB1(2 I+1)=NB(2,J)
      GO TO 130
      END IF
110   CONTINUE
130   L=J
100   CONTINUE

C
C     SELECT THE NODE BOUNDARY
C
      L=2
      NAD(1)=NBOUND(1)
      DO 60 I=2 NTOT*2
      M=NBOUND(I)
      DO 80 J=1 I 1
      IF(M EQ NBOUND(J)) THEN
      GO TO 60
      END IF
80    CONTINUE
      NAD(L)=M
      L=L+1
60    CONTINUE

      RETURN
      END

      SUBROUTINE CONT (F,NCU2 SSS)
C
      IMPLICIT DOUBLE PRECISION (A H O Z)
      CHARACTER SSS*7
      COMMON /INOT/ INPT MSSG IUNIT IUNI2 ISCRN
      COMMON /TSTP/ NINI,NCUR NSEND,NITR DTMAX
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /RVA1/ RZ(2 250) URZ(2 250) FRZ(2 250)
                ,DCOORD(2 100)
      COMMON /INVR/ NOD(4 200),LNBC(2 250),NBCD(2 250)
                LOC(250)
      COMMON /BNOD/ NTOT NB1(2 250)

      DIMENSION F(250)
   1          XE(4) YE(4),FE(4) FCONT(10),EX(99) EY(99)
   2          IARY1(6),F1(250),ND(3 500) ITXT(10)
      DATA IARY1/1 2 2 3 3 1/


      WRITE(NCU2  (7HSECTION) )
      WRITE(NCU2  (3H  2) )
      WRITE(NCU2  (8HENTITIES) )
      WRITE(NCU2  (3H  0) )

      DO 85 I=1 10
85    ITXT(I)=0
      XORG=0 0
      YORG=0 0


      CALL ELTOND (RZ,NOD F,F1,NUMEL,NUMNP)


      CALL GSCALE (NUMNP,RZ XMIN YMIN XMAX
              YMAX SCALE)

C
C     CALCULATE THE HIGHT OF THE TEXT
C
      DY=YMAX YMIN
      DX=XMAX XMIN
      HI=(DY+DX)/60

      J=0
      DO 21 I=1 NUMEL
      J=J+1
      ND(1,J)=NOD(1 I)
      ND(2,J)=NOD(2 I)
```

```
      ND(3,J)=NOD(3 I)
      J=J+1
      ND(1,J)=NOD(1 I)
      ND(2,J)=NOD(3 I)
      ND(3,J)=NOD(4 I)
21    CONTINUE

C
C     Determine the interval of the contour line
C
      NNODE=3
      FMIN=1 E20
      FMAX= 1 E20
      DO 10 I=1 NUMNP
      FI=F1(I)
      IF (FI GT FMAX) FMAX=FI
10    IF (FI LT FMIN) FMIN=FI
      EPP=0 00001*(FMAX FMIN)
C
C     Calculate the values of the contour lines
C
      NCONT=7
      DF=(FMAX FMIN)/6
      FF=FMIN
      FCONT(1)=FF
      DO 15 I=2 NCONT
      FF=FF+DF
15    FCONT(I)=FF

C
C     Write the contour line in DXF format
C
      DO 20 IELEM=1 NUMEL*2

      DO 30 I=1 NNODE
      INOD=ND(I IELEM)
      XE(I)=RZ(1 INOD)
      YE(I)=RZ(2 INOD)
30    FE(I)=F1(INOD)

      DO 50 N=1 NCONT
      FSI=FCONT(N)
      LIN=1

      DO 60 J=1 NNODE
      J1=2*(J 1)+1
      J2=J1+1
      J1A=IARY1(J1)
      J2A=IARY1(J2)
      XE1=XE(J1A)
      YE1=YE(J1A)
      XE2=XE(J2A)
      YE2=YE(J2A)
      FE1=FE(J1A)
      FE2=FE(J2A)
      IF (FE2 FE1 GT EPP) GO TO 300
      IF (FE1 FE2 GT EPP) GO TO 400
      GO TO 500
300   IF (FSI GT FE2 OR FSI LT FE1) GO TO 60
      GO TO 600
400   IF (FSI GT FE1 OR FSI LT FE2) GO TO 60
600   TA=(FSI FE2)/(FE1 FE2)
      EX(LIN)=(XE2+TA*(XE1 XE2)-XMIN)+XORG
      EY(LIN)=(YE2+TA*(YE1 YE2)-YMIN)+YORG
      LIN=LIN+1
      GO TO 60
500   IF (ABS(FSI FE1) GT EPP) GO TO 60
      EX(1)=(XE1 XMIN)+XORG
      EY(1)=(YE1 YMIN)+YORG
      EX(2)=(XF2 XMIN)+XORG
      EY(2)=(YE2 YMIN)+YORG
      LIN=3
60    CONTINUE
```

```
      LIN1=LIN 1
      IF (LIN GE.3) THEN
      CALL DXFC (LIN1,EX EY,N,ΓCONT ITXT,HI,NCU2 SSS)
      ELSE
      END IF
50     CONTINUE
20     CONTINUE

      RETURN
      END

      SUBROUTINE LTOGL
      IMPLICIT DOUBLE PRECISION (A H O Z)
C
C      This subroutine is to change the velocity and of
C       forces of the contact node to global coordinate
C

      COMMON /RVA1/ RZ(2 250) URZ(2 250) FRZ(2 250)
               DCOORD(2 100)
      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
               NSIDE URD(2 100)
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /INVR/ NOD(4 200) LNBC(2 250)
              ,NBCD(2 250),LOC(250)
      COMMON /BNOD/ NTOT NB1(2 250)
      DATA PI/3 1415926535898D0/

      DO 10 I=1 NTOT
      J=NB1(1 I)

C      Check if this node is already on contact with the die

      IF (LNBC(2,J) NE 3) GO TO 10

      I1=ND(1 LOC(J))
      I2=ND(2 LOC(J))
      DA=DCOORD(1 I2) DCOORD(1 I1)
      DB=DCOORD(2 I2)-DCOORD(2 I1)
      IF (DA EQ 0 0) DA=1 D 10
      SM=DB/DA

      ALPHA=DATAN(SM)
      IF (DB GT 0 0 AND DA GT 0 0)
     1      ALPHA=ABS(ALPHA)
      IF (DB GT 0 0 AND DA LT 0 0)
     2      ALPHA=PI ABS(ALPHA)
      IF (DB LT 0 0 AND DA LT 0 0)
     3      ALPHA=ABS(ALPHA)
      IF (DB LT 0 0 AND DA GT 0 0)
     4      ALPHA=PI ABS(ALPHA)
      IF (DB EQ 0 0 AND DA GT 0 0)
     5      ALPHA=PI
      IF (ALPHA EQ PI) THEN
      CO=DCOS(ALPHA)
      SI=0 0D0
      ELSE
      CO=DCOS(ALPHA)
      SI=DSIN(ALPHA)
      END IF

      VX=CO*URZ(1 J) SI*URZ(2,J)
      VY=SI*URZ(1 J)+CO*URZ(2,J)
      IF (VX LT 1 D 10) URZ(1 J)=0 0
      URZ(1,J)=VX
      IF (VY LT 1 D 10) URZ(2 J)=0 0
      URZ(2 J)=VY

      FX=CO*FRZ(1 J)-SI*FRZ(2 J)
      FY=SI*FRZ(1 J)+CO*FRZ(2,J)
      FRZ(1,J)=FX
      FRZ(2 J)=FY
10     CONTINUE
```

```
      RETURN
      END

      SUBROUTINE GLTOL
      IMPLICIT DOUBLE PRECISION (A H O Z)
C
C      This subroutine is to change the velocity of
C          the contact node to global coordinate
C

      COMMON /RVA1/ RZ(2 250) URZ(2,250),FRZ(2,250)
               DCOORD(2 100)
      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
               NSIDE URD(2 100)
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH,NDIE
      COMMON /INVR/ NOD(4 200),LNBC(2,250)
               NBCD(2 250),LOC(250)
      COMMON /BNOD/ NTOT NB1(2 250)
      DATA PI/3 1415926535898D0/

      DO 10 I=1 NTOT
      J=NB1(1 I)

C      Check if this node is already on contact with the die

      IF (LNBC(2,J) NE 3) GO TO 10

      I1=ND(1 LOC(J))
      I2=ND(2 LOC(J))
      DA=DCOORD(1 I2) DCOORD(1 I1)
      DB=DCOORD(2 I2) DCOORD(2,I1)
      IF (DA EQ 0 0) DA=1 D 10
      SM=DB/DA

      ALPHA=DATAN(SM)
      IF (DB GT 0 0 AND DA GT 0 0)
     1      ALPHA=ABS(ALPHA)
      IF (DB GT 0 0 AND DA LT 0 0)
     2      ALPHA=PI ABS(ALPHA)
      IF (DB LT 0 0 AND DA LT 0 0)
     3      ALPHA=ABS(ALPHA)
      IF (DB LT 0 0 AND DA GT 0 0)
     4      ALPHA=PI ABS(ALPHA)
      IF (DB EQ 0 0 AND DA GT 0 0)
     5      ALPHA=PI

      IF (ALPHA EQ PI) THEN
      CO=DCOS(ALPHA)
      SI=0 0D0
      ΓI SE
      CO=DCOS(ALPHA)
      SI=DSIN(ALPHA)
      END IΓ

      VX=CO*URZ(1 J)+SI*URZ(2,J)
      VY= SI*URZ(1 J)+CO*URZ(2,J)
      URZ(1 J)=VX
      URZ(2 J)=VY

10     CONTINUE

      RETURN
      END

      SUBROUTINE CONTACT
      IMPLICIT DOUBLE PRECISION (A H O-Z)
C
C      This subroutine is to provide contact facilities
C          when apdatting the field variables
C

      COMMON /RVA1/ RZ(2 250) URZ(2 250)
```

```
               FRZ(2 250) DCOORD(2 100)
      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
               NSIDE URD(2 100)
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /INVR/ NOD(4 200) LNBC(2 250)
               NBCD(2 250),LOC(250)
      COMMON /BNOD/ NTOT NB1(2 250)
      COMMON /TSTP/ NINI,NCUR NSEND,NITR DTMAX
      DATA PI/3 1415926535898D0/


      DTMIN=1 D20
      TMIN=1 D20
      T=1 D-20
      II=0
C
C      Loop over all boundary nodes
C
      DO 10 I=1 NTOT
      J=NB1(1 I)

C      Check if this node is already on contact with the die

      IF (LNBC(2,J) EQ 3) GO TO 10

C      Calculate the relative velocity

      VRX=URZ(1 J)-VDIEX
      VRY=URZ(2 J)-VDIEY
      V=DSQRT(VRX*VRX+VRY*VRY)

C      Calculate the slop of the relative velocity vector

      IF (VRX EQ 0 0) VRX=1 D 10
      SM1=VRY/VRX

C      Calculate the angle of the velocity vector

      ALPHA=DATAN(SM1)
      IF (VRY GT 0 0 AND VRX GT 0 0)
     1        ALPHA=ABS(ALPHA)
      IF (VRY GT 0 0 AND VRX LT 0 0)
     2        ALPHA=PI ABS(ALPHA)
      IF (VRY LT 0 0 AND VRX LT 0 0)
     3        ALPHA=PI+ABS(ALPHA)
      IF (VRY LT 0 0 AND VRX GT 0 0)
     4        ALPHA=2*PI ABS(ALPHA)

C
C      Loop over all die segments to check if the
C          velocity vector go through any
C

      DO 20 N=1 NSIDE
      I1=ND(1 N)
      I2=ND(2 N)
      A=DCOORD(1 I2)-DCOORD(1 I1)
      B=DCOORD(2 I2) DCOORD(2 I1)
      IF (A EQ 0 0) A=1 D 10
      SM2=B/A

C      Check if the velocity vector is paralell to the die side

      IF (SM1 EQ SM2) GO TO 20

C      Calculate the angle of the line connecting the current
C            with the first node of the segment

      AL1=DCOORD(1 I1) RZ(1,J)
      AL2=DCOORD(2 I1) RZ(2,J)
      IF (AL1 EQ 0 0) AL1=1 D-10
      ALPHA1=DATAN(AL2/AL1)
      IF (AL2 GT 0 0 AND AL1 GT 0 0)
```

```
     1        ALPHA1=ABS(ALPHA1)
      IF (AL2 GT 0 0 AND AL1 LT 0 0)
     2        ALPHA1=PI ABS(ALPHA1)
      IF (AL2 LT 0 0 AND AL1 LT 0 0)
     3        ALPHA1=PI+ABS(ALPHA1)
      IF (AL2 LT 0 0 AND AL1 GT 0 0)
     4        ALPHA1=2*PI ABS(ALPHA1)

C      Calculate the angle of the line connecting the current
C            with the second node of the segment

      AL1=DCOORD(1 I2) RZ(1,J)
      AL2=DCOORD(2 I2)-RZ(2,J)
      IF (AL1 EQ 0 0) AL1=1 D-10
      ALPHA2=DATAN(AL2/AL1)
      IF (AL2 GT 0 0 AND AL1 GT 0 0)
     1        ALPHA2=ABS(ALPHA2)
      IF (AL2 GT 0 0 AND AL1 LT 0 0)
     2        ALPHA2=PI ABS(ALPHA2)
      IF (AL2 LT 0 0 AND AL1 LT 0 0)
     3        ALPHA2=PI+ABS(ALPHA2)
      IF (AL2 LT 0 0 AND AL1 GT 0 0)
     4        ALPHA2=2*PI ABS(ALPHA2)

C      Check if the current velocity vector goes through
C                this segment

C     IF (ALPHA LT ALPHA1 AND ALPHA GT ALPHA2) THEN

C     Calculate the coordinates of the intersection point

      SM=SM1 SM2
      IF (SM EQ 0 0) SM=1 D-10
      X=(SM1*RZ(1 J)-SM2*DCOORD(1 I1)+
     1  DCOORD(2 I1) RZ(2,J))/SM

      Y=SM2*X (SM2*DCOORD(1 I1)-DCOORD(2,I1))

      P1P=DSQRT((DCOORD(2 I1) Y)*(DCOORD(2 I1)-Y)+
     %        (DCOORD(1 I1)-X)*(DCOORD(1 I1)-X))
      P2P=DSQRT((DCOORD(2 I2) Y)*(DCOORD(2 I2)-Y)+
     %        (DCOORD(1 I2) X)*(DCOORD(1,I2)-X))
      P1P2=DSQRT((DCOORD(2 I2) DCOORD(2 I1))*
     %         (DCOORD(2 I2) DCOORD(2 I1))+
     %         (DCOORD(1 I2) DCOORD(1 I1))*
     %         (DCOORD(1 I2) DCOORD(1 I1)))

C      Check if the velocity vector go through this side

      P=P1P+P2P
      IF (ABS(P P1P2) GT 0 01) GO TO 20

C      Calculate the distance between the node and the side

      DN=ABS((((DCOORD(1 I1) RZ(1 J))*B)+
     %   ((DCOORD(2 I1) RZ(2,J))*( A)))
     %   /DSQRT(B*B+A*A))
C
C      The time necessary for this node to reach the die
C
      A=DCOORD(1 I1) DCOORD(1 I2)
      B=DCOORD(2 I1) DCOORD(2 I2)
      TETA=DACOS(((A*VRX)+(B*VRY))/
     %        (DSQRT(A*A+B*B)*V))
      VN=V*DSIN(TETA)
      DT=DN/ABS(VN)

C      Comparing this time with the maximum time increment

      IF(DT GT DTMAX) GOTO 20

C      Keep the infomation of this node and die segment where
C        at the end of the die segments loop the closest
```

```
C          segment from this node will be considered

      IF (DT LT DTMIN) THEN
      DTMIN=DT
      X1=X
      Y1=Y
      K=J
      II=I1
      S=SM2
      A1=A
      B1=B
      ELSE
      GOTO 20
      END IF

20     CONTINUE

C
C      To find out the minimum contact time of the first node
C                goes into contact for this step
C
      IF(II EQ 0) GOTO 10

      IF (DTMIN GT T) T=DTMIN

C
C      Change the boundary code of the new node in contact
C         Assign the die velocity to this node
C
      WRITE(6 *) J II,DT
      RZ(1,K)=X1
      RZ(2,K)=Y1
      LNBC(1 K)=0
      LNBC(2 K)=3
      IF(B1 EQ 0 0) NBCD(1,K)=3
      IF(A1 EQ 1 0D 10) NBCD(2 K)=3
      IF(B1 NE 0 0 AND A1 NE 1 0D-10) THEN
      NBCD(1,K)=3
      NBCD(2,K)=3
      ELSE
      ENDIF

      ALPHA=DATAN(S)
      IF (B1 GT 0 0 AND A1 GT 0 0)
    1         ALPHA=ABS(ALPHA)
      IF (B1 GT 0 0 AND A1 LT 0 0)
    2         ALPHA=PI ABS(ALPHA)
      IF (B1 LT 0 0 AND A1 LT 0 0)
    3         ALPHA=ABS(ALPHA)
      IF (B1 LT 0 0 AND A1 GT 0 0)
    4         ALPHA=PI ABS(ALPHA)
      IF (B1 EQ 0 0 AND A1 GT 0 0)
    5         ALPHA=PI

      IF (ALPHA EQ PI) THEN
      CO=DCOS(ALPHA)
      SI=0 0D0
      ELSE
      CO=DCOS(ALPHA)
      SI=DSIN(ALPHA)
      END IF

      URZ(1 K)=CO*VDIEX+SI*VDIEY
      URZ(2 K)= SI*VDIEX+CO*VDIEY

      LOC(K)=II
10     CONTINUE
      IF(II EQ 0) RETURN
      DTMAX=T

      RETURN
      END
```

```
      SUBROUTINE DIESEG (NCU2 NCUR)
      IMPLICIT DOUBLE PRECISION (A H O-Z)
C
C      THIS SUBROUTINE PLOT THE DIE SEGMENTS
C
      CHARACTER DIED*6,DIE*8 S*11 F*1 F1*2
      COMMON /RVA1/ RZ(2 250) URZ(2,250) FRZ(2,250)
                 DCOORD(2 100)
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /FILE/ MESHD,NODED ELEMENTD

      DIED= DIESEG
      S= 01234567890
      I=NCUR

      IF (I LT 10) THEN
      F=S((I+1) (I+1))
      DIE=DIED// 0 //F
      ELSE
      J=I/10
      F1=S((J+1) (J+1))//S((I ((J 1)*10)-9) (I ((J 1)*10)-9))
      DIE=DIED//F1
      END IF


      WRITE(NCU2  (7HSECTION) )
      WRITE(NCU2  (3H  2) )
      WRITE(NCU2  (8HENTITIES) )


      DO 10 K=1 NDIE 1
      WRITE(NCU2  (3H  0) )
      WRITE(NCU2  (4HLINE) )
      WRITE(NCU2  (3H  8) )
      WRITE(NCU2  (A) )DIE
      WRITE(NCU2  (3H 62) )
      WRITE(NCU2  (1H9) )
      WRITE(NCU2  (3H 10) )
      WRITE(NCU2  (F10 6) )DCOORD(1,K)
      WRITE(NCU2  (3H 20) )
      WRITE(NCU2  (F10 6) )DCOORD(2,K)
      WRITE(NCU2  (3H 30) )
      WRITE(NCU2  (3H0 0) )

      WRITE(NCU2  (3H 11) )
      WRITE(NCU2  (F10 6) )DCOORD(1,K+1)
      WRITE(NCU2  (3H 21) )
      WRITE(NCU2  (F10 6) )DCOORD(2,K+1)
      WRITE(NCU2  (3H 31) )
      WRITE(NCU2  (3H0 0) )
10     CONTINUE
      WRITE(NCU2  (3H  0) )
      RETURN
      END

      SUBROUTINE DISBDY (URZ,LNBC B,A NEQ
                 MBAND ITYP)
      IMPLICIT DOUBLE PRECISION (A H,O Z)

C      APPLY DISPLACEMENT BOUNDARY CONDITION

      DIMENSION B(1),A(NEQ 1),LNBC(1) URZ(1)

      IF (ITYP EQ 2) GOTO 120
      DO 100 N=1 NEQ
      IF (LNBC(N) EQ 0) GOTO 100
      DO 70 I=2 MBAND
      II=N I+1
      IF (II LE 0) GOTO 50
      A(II I)=0
50     CONTINUE
      II=N+I 1
      IF (II GT NEQ) GOTO 70
      A(N I)=0
70     CONTINUE
      B(N)=0
      A(N 1)=1
100    CONTINUE
      RETURN

120    CONTINUE
      DO 200 N=1 NEQ
      IF (LNBC(N) EQ 0) GOTO 200
      DO 170 I=2 MBAND
      II=N I+1
      IF (II LE.0) GOTO 150
      B(II)=B(II) A(II I)*URZ(N)
      A(II I)=0
150    CONTINUE
      II=N+I 1
      IF (II GT NEQ) GOTO 170
      B(II)=B(II) A(N I)*URZ(N)
      A(N I)=0
170    CONTINUE
      B(N)=URZ(N)
      A(N 1)=1
200    CONTINUE
      END
```

```
      SUBROUTINE DXFC (K,EX EY,N FCONT ITXT,HI
                 NCU2 SSS)
C
C
C        THIS SUBROUTINE IS TO CREATE THE
C          DXF FILE FOR THE CONTOUR
C
      IMPLICIT REAL*8 (A H O Z) INTEGER*4 (I N)
      DIMENSION EX(99),EY(99),FCONT(10) ITXT(10)
      CHARACTER SSS*7
C
C
C      CREATING THE DXF FILE
C
C
      DO 20 JNODE=1 K 1
      WRITE(NCU2 (4HLINE) )
      WRITE(NCU2 (3H 8) )
      WRITE(NCU2 (A) )SSS
      WRITE(NCU2 (3H 62) )
      WRITE(NCU2 *)N
      WRITE(NCU2 (3H 10) )
      WRITE(NCU2 (E11 4) )EX(JNODE)
      WRITE(NCU2 (3H 20) )
      WRITE(NCU2 (E11 4) )EY(JNODE)
      WRITE(NCU2 (3H 30) )
      WRITE(NCU2 (3H0 0) )

C       IF (JNODE EQ K) THEN
C       NODE=1
C       ELSE
C       NODE=JNODE+1
C       END IF

      WRITE(NCU2 (3H 11) )
      WRITE(NCU2 (E11 4) )EX(JNODE+1)
      WRITE(NCU2 (3H 21) )
      WRITE(NCU2 (E11 4) )EY(JNODE+1)
      WRITE(NCU2 (3H 31) )
      WRITE(NCU2, (3H0 0)')
      WRITE(NCU2 (3H  0) )

      IF (ITXT(N) EQ 0) THEN
      WRITE(NCU2 (4HTEXT) )
      WRITE(NCU2 (3H 8) )
      WRITE(NCU2 (A) )SSS
      WRITE(NCU2 (3H 62) )
      WRITE(NCU2 *)N
      WRITE(NCU2 (3H 10) )
      WRITE(NCU2 (E11 4) )EX(JNODE)
      WRITE(NCU2 (3H 20) )
      WRITE(NCU2 (E11 4) )EY(JNODE)
      WRITE(NCU2 (3H 30) )
      WRITE(NCU2 (3H0 0) )
      WRITE(NCU2 (3H 40) )
      WRITE(NCU2 (F10 6) )HI
      WRITE(NCU2 (3H  1) )
      WRITE(NCU2 (E11 4) )FCONT(N)
      WRITE(NCU2 (3H  0) )
      ITXT(N)=1
      ELSE
      GO TO 20
      END IF
20     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ELSHLF (PP QQ RZ,URZ,EPS
                 TEPS IPLNAX TH IDREC NEL L)
      IMPLICIT DOUBLE PRECISION (A H O-Z)

C      EVALUATION OF ELEMENTAL STIFFNESS MATRIX
C
C          IDREC  =  1       NEWTON RAPHSON ITERATION
C                    2       DIRECT ITERATION

      COMMON /RIGD/ RTOL,ALPH,DIAT,IPLAS,STK,EXN
      COMMON /T/ W11 W22
      COMMON /TSTP/ NINI,NCUR NSEND,NITR DTMAX
      DIMENSION RZ(2 1) URZ(2 1) BB(4 8) EPS(1) TEPS(1)
      DIMENSION QQ(1),PP(8 8) S2(2) W2(2),L(4)
      DATA S2/ 0 57735026918963D0 0 57735026918963D0/
   1     W2/2*1 0D0/

      DO 10 I=1 8
      QQ(I)=0
      DO 10 J=1 8
      PP(I,J)=0
10     CONTINUE

C      CARRY OUT ONE POINT INTEGRATION

      S=0
      T=0
      CALL STRMTX (RZ BB WDXJ S T IPLNAX TH
                NEL L IDREC)
      CALL VSPLON (QQ,PP BB URZ,EPS WDXJ IDREC)

C      REGULAR INTEGRATION

      DO 100 I=1 2
      S=S2(I)
      DO 50 J=1 2
      T=S2(J)
      CALL STRMTX (RZ BB WDXJ S T IPLNAX
                TH NEL,L IDREC)
      W11=W2(I)
      W22=W2(J)
      CALL VSPLST (QQ PP BB URZ,WDXJ IDREC TEPS)
50     CONTINUE
100    CONTINUE
      RETURN
      END

      SUBROUTINE ELTOND (RZ,NOD FI F NELEM,NPOIN)
```

```
C*******************************************************
   IMPLICIT DOUBLE PRECISION (A H O Z)
     DIMENSION RZ(2 250) NOD(4 200) F1(200) F(200)
     DIMENSION RZ1(2 4) W(2) S2(2) SHAPE(4)
     DATA S2/ 0 57735026918963D0 0 57735026918963D0/
     DATA W/2*1 0D0/


     NNODE=4
     DO 5 INODE=1,NPOIN
     F(INODE)=0 0D0
5     CONTINUE

     DO 10 IPOIN=1,NPOIN
     AREA=0 0D0
     JPOIN=IPOIN
     UP=0 0D0
     DO 20 IELEM=1 NELEM
     DO 30 INODE=1,NNODE
     IF(IPOIN EQ NOD(INODE,IELEM)) THEN
     SS=F1(IELEM)
     A=0 0D0

     DO 40 I=1 4
     NE=NOD(I IELEM)
     RZ1(1 I)=RZ(1 NE)
     RZ1(2 I)=RZ(2 NE)
40    CONTINUE

     DO 50 I=1 2
     S=S2(I)
     DO 60 J=1 2
     T=S2(J)

C
C     Calculate the Jacobian matrix for the  element.
C
     CALL JACOB (RZ1 WDXJ S T)

     CALL SHAPE4 (T S SHAPE)
     DO 70 II=1 4
C
C     Calculate the area of a shared element to a particular node
C
     A=A+W(1)*W(2)*WDXJ*SHAPE(II)
70    CONTINUE
60    CONTINUE
50    CONTINUE

     UP=UP+SS*A

     ELSE
     SS=0 0D0
     GO TO 30
     END IF
C
C     Calculate the the sum of the element area which share
C                   the same node
C
     AREA=AREA+A
30    CONTINUE
20    CONTINUE
C
C     Calculate the effective strain at the new node
C
     F(IPOIN)=UP/AREA
10    CONTINUE
     RETURN
     END


     SUBROUTINE FLWST1 (YS,FIP STRRT)
     IMPLICIT DOUBLE PRECISION (A H O Z)
```

```
C     USER SUPPLIED SUBROUTINE TO DESCRIBE THE
C                 MATERIAL FLOW STRESS

C      THIS SUBROUTINE SHOWS THE VISCO PLASTIC
C                  MATERIALS
C      YS=STK*(STRAIN RATE)**EXN

      COMMON /RIGD/ RTOL,ALPH,DIAT,IPLAS STK,EXN
C
C   Ys = K * E **n        dYs / dE  = K * n * E **(n 1)
C
C   CUT OFF E o = ALPH
C
C      Yo = K * E o ** n
C      Ys = Yo / E o * E      dYs /dE  = Yo /E.o
C

      IF (EXN EQ 0 0) THEN
      FIP=0 0
      YS=STK
      RETURN
      END IF
      IF (STRRT LT ALPH) GOTO 100
      YS=STK*STRRT**EXN
      FIP=STK*EXN*STRRT**(EXN 1)
      RETURN

100    YO=STK*ALPH**EXN
      FIP=YO/ALPH
      YS=FIP*STRRT
      RETURN
      END

      SUBROUTINE FLWST2 (YS,FIP,EFSTR)
      IMPLICIT DOUBLE PRECISION (A H,O-Z)

C      USER SUPPLIED SUBROUTINE TO DESCRIBE THE
C                MATERIAL FLOW STRESS

C      THIS SUBROUTINE SHOWS THE RIGID PLASTIC
C                  MATERIALS
C      YS=STK*(EFFECTIVE STRAIN)**EXN

      COMMON /RIGD/ RTOL,ALPH,DIAT,IPLAS STK,EXN
C
C   Ys = K   E**n        dYs / dE. = 0 0
C
C   CUT OFF Eo = ALPH
C
C      Yo = K   Eo ** n
C      Ys = Yo / Eo E     dYs /dE  = 0 0
C

      IF (EXN EQ 0 0) THEN
      FIP=0 0
      YS=STK
      RETURN
      END IF
      FIP=0 0
      IF (EFSTR LT ALPH) GOTO 100
      YS=STK*EFSTR**EXN
      RETURN

100    YO=STK*ALPH**EXN
      F=YO/ALPH
      YS=F*EFSTR
      RETURN
      END


      SUBROUTINE FRCBDY (RZ URZ,LNBCE,EFSTR
    1                  EFTR QQ PP IPLNAX TH ITYP)
      IMPLICIT DOUBLE PRECISION (A H O Z)
```

```
C      APPLY FRICTION BOUNDARY CONDITION

      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
                 NSIDE URD(2 100)
      COMMON /INOT/ INPT MSSG IUNIT IUNI2 ISCRN
      COMMON /RIGD/ RTOL,ALPH,DIAT IPLAS STK,EXN
      COMMON /TSTP/ NINI,NCUR NSEND,NITR DTMAX
      DIMENSION RZ(2 1) URZ(2 1) LNBCE(2 1) QQ(1)
    1         PP(8 1) ER(2 2) FR(2),XY(2 2) VXY(2 2)

      DO 100 N=1 4
      I1=N+1
      I2=N
      IF (N.EQ 4) I1=1
      IF(LNBCE(2 I1) NE.3 OR LNBCE(2 I2) NE 3) GOTO 100

          IF (IPLAS EQ 1) THEN
      IF (NITR EQ 1 AND NCUR EQ NINI
         AND ITYP EQ 2) EFSTR=ALPH
C           EFSTR=EPS(5 1)
          CALL FLWST1 (FLOW,DUM EFSTR)
          ELSE
      IF (NITR EQ 1 AND NCUR EQ NINI
         AND ITYP EQ 2) EFSTR=ALPH
C           EFTR=TEPS(1)
          CALL FLWST2 (FLOW,DUM EFTR)
          END IF
      XY(1 1)=RZ(1 I1)
      XY(2 1)=RZ(2 I1)
      XY(1 2)=RZ(1 I2)
      XY(2 2)=RZ(2 I2)
      VXY(1 1)=URZ(1 I1)
      VXY(2 1)=URZ(2 I1)
      VXY(1 2)=URZ(1 I2)
      VXY(2 2)=URZ(2 I2)

      CALL FRCINT (XY VXY FLOW,FR,ER IPLNAX TH)

      J1=I1*2 1
      J2=I2*2 1
      QQ(J1)=QQ(J1)+FR(1)
      QQ(J2)=QQ(J2)+FR(2)
      PP(J1,J1)=PP(J1 J1)+ER(1 1)
      PP(J2 J2)=PP(J2 J2)+ER(2 2)
      PP(J1,J2)=PP(J1 J2)+ER(1 2)
      PP(J2,J1)=PP(J2,J1)+ER(2 1)
100    CONTINUE
      RETURN
      END

      SUBROUTINE FRCINT (RZ URZ,FLOW FR
                   ER IPLNAX TH)
      IMPLICIT DOUBLE PRECISION (A H O Z)

C      INTEGRATION METHOD   SIMPSON S FORMULA
C      THIS ROUTINE CALCULATES THE FRICTION
C               MATRIX
C      USED FOR BOTH TYPES OF ITERATION SCHEME

      COMMON /INOT/ INPT MSSG IUNIT IUNI2 ISCRN
      COMMON /ITRC/ ITYP ICONV
      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
                 NSIDE URD(2 100)
      DIMENSION RZ(2 1) URZ(1),ER(2 2) FR(2)
      DATA PI/3 1415926535898D0/
      DATA UA/0 0005D0/


C      INITIALIZE FR AND ER ARRAY

      DO 10 I=1 2
      FR(I)=0
      DO 10 J=1 2
      ER(I,J)=0
10     CONTINUE

      NINT=5
      FAC=DSQRT((RZ(1 2)-RZ(1 1))**2+(RZ(2,2)-RZ(2 1))**2)
      FK=FLOW*FRCFAC/SQRT(3 0)
      DH=2 /(NINT 1)
      CON=2 /PI*FK
      WD=DH/3 *FAC*0 5*CON

      B2=RZ(2 2) RZ(2 1)
      A2=RZ(1 2) RZ(1 1)
      IF (A2 EQ 0 0) A2=1 D 10

      TETA=DATAN(B2/A2)
      IF (B2 GT 0 0 AND A2 GT 0 0)
    1        TETA=ABS(TETA)
      IF (B2 GT 0 0 AND A2 LT 0 0)
    2        TETA=PI ABS(TETA)
      IF (B2 LT 0 0 AND A2 LT 0 0)
    3        TETA=ABS(TETA)
      IF (B2 LT 0 0 AND A2 GT 0 0)
    4        TETA=PI ABS(TETA)
      IF (B2 EQ 0 0 AND A2 GT 0 0)
    5        TETA=PI

      IF (TETA EQ PI) THEN
      CO=DCOS(TETA)
      SI=0 0D0
      ELSE
      CO=DCOS(TETA)
      SI=DSIN(TETA)
      END IF

      VS=CO*VDIEX+SI*VDIEY
      VN=-SI*VDIEX+CO*VDIEY

      S= 1  DH
      DO 300 N=1 NINT
      S=S+DH
      H1=0 5*(1  S)
      H2=0 5*(1 +S)
      WDXJ=WD
      IF (IPLNAX EQ 1) THEN
      RR=H1*RZ(1 1)+H2*RZ(1 2)
      WDXJ=2 0*PI*RR*WDXJ
      ELSE
      WDXJ=TH*WDXJ
      END IF

      IF (N EQ 1 OR N EQ NINT) GOTO 100
      NMOD=N N/2*2
      IF (NMOD EQ 0) WDXJ=WDXJ*4
      IF (NMOD EQ 1) WDXJ=WDXJ*2
100    CONTINUE

      US=H1*(URZ(1) VS)+H2*(URZ(3)-VS)

      AT=DATAN(US/UA)

      IF (ITYP EQ 2) GOTO 200
      US2=US*US
      USA=US2+UA*UA
      CT1=AT*WDXJ
      CT2=UA/USA*WDXJ
      GOTO 250

C      FOR D-ITERATION CASE

200    CONTINUE
      IF (DABS(US) LE 1 0D 5) SLOP=UA/(UA*UA+US*US)
      IF (DABS(US) GT 1 0D-5) SLOP=AT/US
      CT1=0
      CT2=SLOP*WDXJ

C      CALCULATE CONTRIBUTION TO STIFFNESS

250    CONTINUE
      FR(1)=FR(1)-H1*CT1
      FR(2)=FR(2)-H2*CT1
      ER(1 1)=ER(1 1)+H1*H1*CT2
      ER(1 2)=ER(1 2)+H1*H2*CT2
      ER(2,2)=ER(2 2)+H2*H2*CT2
      ER(2 1)=ER(1 2)
300    CONTINUE
      RETURN
      END

      SUBROUTINE TRANS (L BB)
      IMPLICIT DOUBLE PRECISION (A H O Z)
C                                                  C
C                                                  C
C      This subroutine is to add the transformation matrix    C
C        T^T to the strain rate matrix B                   C
C                                                  C
C                                                  C
      COMMON /INVR/ NOD(4 200) LNBC(2 250)
                 NBCD(2 250),LOC(250)
      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
                 NSIDE URD(2 100)
      COMMON /RVA1/ RZ(2 250) URZ(2 250) FRZ(2 250)
                 ,DCOORD(2 100)
      DIMENSION L(4) BB(4 8),ANG(8 8),RES(4 8)
      DATA PI/3 141592653589 8D0/


C
C      BUILD THE ELEMENTAL TRANSFORMATION
C                MATRIX
C
      DO 10 I=1 4
      IF (L(I) EQ 0) GO TO 10
      K1=2*I 1
      K2=2*I

      J=L(I)
      I1=ND(1 LOC(J))
      I2=ND(2 LOC(J))
      DA=DCOORD(1 I2) DCOORD(1 I1)
      DB=DCOORD(2 I2) DCOORD(2 I1)
      IF (DA EQ 0 0) DA=1 D 10
      SM=DB/DA

      ALPHA=DATAN(SM)
      IF (DB GT 0 0 AND DA GT 0 0)
   1        ALPHA=ABS(ALPHA)
      IF (DB GT 0 0 AND DA LT 0 0)
   2        ALPHA=PI ABS(ALPHA)
      IF (DB LT 0 0 AND DA LT 0 0)
   3        ALPHA=ABS(ALPHA)
      IF (DB LT 0 0 AND DA GT 0 0)
   4        ALPHA=PI ABS(ALPHA)
      IF (DB EQ 0 0 AND DA GT 0 0)
   5        ALPHA=PI

      IF (ALPHA EQ PI) THEN
      CO=DCOS(ALPHA)
      SI=0 0D0
      ELSE
      CO=DCOS(ALPHA)
      SI=DSIN(ALPHA)
      END IF

      DO 20 J=1 4
      X=BB(J,K1)*CO+BB(J K2)*SI
      Y= BB(J K1)*SI+BB(J,K2)*CO
```

```
      BB(J K1)=X
      BB(J,K2)=Y
20     CONTINUE

10     CONTINUE

      RETURN
      END

      SUBROUTINE GSCALE (NUMNP,RZ,XMIN YMIN
                   XMAX YMAX,ASIZE)
      IMPLICIT DOUBLE PRECISION (A H O Z)
C
      DIMENSION RZ(2 250)

      XMIN=1 E20
      YMIN=1 E20
      XMAX= 1 E20
      YMAX= 1 E20

      DO 10 I=1 NUMNP
      XI=RZ(1 I)
      IF (XI LT XMIN) XMIN=XI
      IF (XI GT XMAX) XMAX=XI
      YI=RZ(2 I)
      IF (YI LT YMIN) YMIN=YI
10     IF (YI GT YMAX) YMAX=YI
      XSIZE=XMAX XMIN
      YSIZE=YMAX YMIN
      IF (YSIZE GE XSIZE) THEN
      ASIZE=YSIZE
      ELSE
      ASIZE=XSIZE
      END IF
      RETURN
      END

      SUBROUTINE INPRED
      IMPLICIT DOUBLE PRECISION (A H O-Z)
C
C
C           READ INPUT FROM INPUT FILE
C
C

      CHARACTER TITLE*70
      COMMON /TITL/ TITLE
      COMMON /TSTP/ NINI,NCUR NSEND,NITR DTMAX
      COMMON /RVA1/ RZ(2 250) URZ(2 250) FRZ(2 250)
                 DCOORD(2 100)
      COMMON /RVA2/ EPS(5 200) STS(5,200) TEPS(200)
      COMMON /INVR/ NOD(4 200),LNBC(2 250),NBCD(2,250)
                 LOC(250)
      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
                 NSIDE  URD(2 100)
      COMMON /RIGD/ RTOL,ALPH,DIAT,IPLAS STK,EXN
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /INOT/ INPT MSSG IUNIT IUNI2,ISCRN

C      READ MASTER CONTROL DATA
C
      OPEN (INPT FILE= FEM DAT
         FORM= FORMATTED  STATUS= OLD )
      READ (INPT 1000) TITLE
      READ (INPT *) NINI NSEND,DTMAX
      READ (INPT *) ALPH DIAT
      READ (INPT *) IPLAS STK,EXN
      READ (INPT *) VDIEX VDIEY
      READ (INPT *) IPLNAX
      IF (IPLNAX EQ 2) READ(INPT *) TH

C      READ DIE DATA
C
      READ (INPT *) FRCFAC
```

```
C     READ FEM NODE INFORMATION
C
      READ (INPT *) NUMNP
      IF (NUMNP GT 250) GOTO 500
      DO 20 I=1 NUMNP
      READ (INPT *) N (RZ(J N),J=1 2)
20    CONTINUE

      DO 310 I=1 NUMNP
      DO 310 J=1 2
      FRZ(J I)=0 0
310   CONTINUE

C     READ ELEMENT INFORMATION
C
      READ (INPT *) NUMEL
C       DO 320 I=1 100
C       DO 320 J=1 4
C320    NOD(J I)=0

      IF (NUMEL GT 200) GOTO 500
      DO 40 I=1 NUMEL
      READ (INPT *) N (NOD(J,N),J=1 4)
40    CONTINUE


C     READ BOUNDARY CONDITION DATA
C
      DO 60 N=1 NUMNP
      LOC(N)=0
      DO 60 I=1 2
      NBCD(I,N)=0
      LNBC(I N)=0
60    CONTINUE
C
C      READ NUMBER OF BOUNDARY NODE AND NODE
C              IN CONTACT WITH DIE
C      NBNODE NUMBER OF BOUNDARY NODE IN
C                CONTACT
C  NBCD(1,NBNODE) BOUNDARY CONDITION IN X OR R

C               0 NODAL FORCE IS SPECIFIED
C               1 NODAL VELOCITY IS SPECIFIED
C               3 NODE IS IN CONTACT WITH THE DIE
C    NBCD(2,NBNODE) BOUNDARY CONDITION CODE IN
C                  Y OR Z
C               0 NODAL FORCE IS SPECIFIED
C               1 NODAL VELOCITY IS SPECIFIED
C               3 NODE IS IN CONTACT WITH THE DIE
C
      READ (INPT *) NBNODE

      DO 80 N=1 NBNODE
      READ (INPT *) M,NBCD(1 M),NBCD(2 M),LOC(M)
      IF (NBCD(1 M) EQ 3 OR NBCD(2 M) EQ 3) THEN

      IF (NBCD(1 M) EQ 3) THEN
      LNBC(1 M)=0
      ELSE
      LNBC(1 M)=NBCD(1 M)
      END IF

      LNBC(2 M)=3
      ELSE
      LNBC(1 M)=NBCD(1 M)
      LNBC(2 M)=NBCD(2 M)
      END IF
80    CONTINUE

C     READ NODE VELOCITY DATA
C
      DO 120 N=1 NUMNP
      DO 120 I=1 2
      URZ(I N)=0 0
120   CONTINUE

C       READ THE NUBER OF NODES
C  WHICH ARE AFFECTED BY EXTERNAL VELOCITY
      READ(INPT *) NVNODE

      DO 140 N=1 NVNODE
      READ (INPT *) M (URZ(I M) I=1 2)
140   CONTINUE

C     READ STRAIN DATA
C
      IF (NINI EQ 0) THEN
      DO 200 N=1 NUMEL
      IF (IPLAS EQ 0) TEPS(N)=0 001D0
      IF (IPLAS EQ 1) TEPS(N)=0 0
200   CONTINUE
      ELSE
      DO 240 N=1 NUMEL
      READ (INPT *) M TEPS(M)
240   CONTINUE
      END IF
      READ(INPT *) NDIE
      DO 250 N=1 NDIE
      READ(INPT *)I DCOORD(1 I),DCOORD(2 I)
   %            URD(1 I) URD(2 I)
250   CONTINUE
      READ(INPT *)NSIDE
      DO 260 N=1 NSIDE
      READ(INPT *)I (ND(J I),J=1 2)
260   CONTINUE

      CLOSE (INPT)
      RETURN

500   CONTINUE
      WRITE (MSSG 1010)
      STOP

1000   FORMAT (A)
1010   FORMAT (/ SORRY  THIS PROGRAM
           CANNOT HANDLE MORE THAN 250
   1          NODES OR ELEMENTS )
      END



      SUBROUTINE JACOB (COORD WDXJ S T)
C
      IMPLICIT DOUBLE PRECISION (A H O Z)

C      EVALUATE THE AREA OF QUADRILATERAL
C                ELEMENT

C        COORD(2 4)       NODE COORDINATES
C        (S T)          NATURAL COORDINATE

      DIMENSION COORD(2 4)


      R12=COORD(1 1) COORD(1,2)
      R13=COORD(1 1) COORD(1 3)
      R14=COORD(1 1) COORD(1 4)
      R23=COORD(1 2) COORD(1 3)
      R24=COORD(1 2) COORD(1 4)
      R34=COORD(1 3) COORD(1 4)

      Z12=COORD(2 1) COORD(2 2)
      Z13=COORD(2 1) COORD(2 3)
      Z14=COORD(2 1) COORD(2 4)
      Z23=COORD(2 2) COORD(2 3)
      Z24=COORD(2 2) COORD(2 4)
      Z34=COORD(2 3) COORD(2 4)

      DXJ8=((R13*Z24 R24*Z13)+(R34*Z12 R12*Z34)*S+
    1      (R23*Z14 R14*Z23)*T)
      WDXJ=DXJ8/8

      RETURN
      END

      SUBROUTINE NFORCE (QQ,FRZ,LM)
      IMPLICIT DOUBLE PRECISION (A H O Z)

C      ADD NODAL POINT FORCE

      DIMENSION QQ(1),FRZ(1) LM(1)

      DO 100 I=1 8
      N=LM(I)
      FRZ(N)=FRZ(N) QQ(I)
100    CONTINUE
      RETURN
      END



      SUBROUTINE NONLIN
      IMPLICIT DOUBLE PRECISION (A H O Z)

C
C        THIS ROUTINE CONTROLS THE ITERATIONS
C

      COMMON /INOT/ INPT MSSG IUNIT IUNI2 ISCRN
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /TSTP/ NINI,NCUR NSEND,NITR DTMAX
      COMMON /ITRC/ ITYP ICONV
      COMMON /CNEQ/ NEQ MBAND
      COMMON /RVA1/ RZ(2 250) URZ(2 250) FRZ(2 250)
                    DCOORD(2 100)
      COMMON /RIGD/ RTOL,ALPH,DIAT IPLAS STK,EXN
      DIMENSION UNORM(2) ENORM(2) FNORM(2)
      COMMON A(25000) B(500)

      RTOL=0 005
      IF (ITYP EQ 2) RTOL=0 005
      ACOEF=0 5
      NSTEL=NEQ*MBAND
      IF (NSTEL LE.25000.AND NEQ LE 500) GOTO 10
      WRITE (MSSG 1010)
      STOP

10     CONTINUE
      DO 30 N=1 2
      UNORM(N)=0 0
      ENORM(N)=0 0
      FNORM(N)=0 0
30     CONTINUE

      ITRMAX=40
      IF (ITYP EQ 2) ITRMAX=200

      DO 200 N=1 ITRMAX
      NITR=N

      CALL STIFF (B,A NEQ MBAND ITYP)
      IDREC=1

      CALL NORM (FRZ B FDUM DFN,NEQ IDREC)

      IF (ITYP EQ 2) DFN=0
      CALL BANSOL (B,A,NEQ MBAND)
```

```
      IDREC=ITYP

      CALL NORM (URZ B UC EC NEQ IDREC)

      IF (ITYP EQ 1) WRITE(MSSG 1030) N
      IF (ITYP EQ 1) WRITE(ISCRN 1030) N
      IF (ITYP EQ 2) WRITE(MSSG 1050) N
      IF (ITYP EQ 2) WRITE(ISCRN 1050) N
      WRITE(MSSG 1070) UC EC,DFN
      WRITE(ISCRN 1070) UC,EC,DFN
C       WRITE (MSSG 1100) (NN (URZ(II NN) II=1 2)
C    1                   (FRZ(II NN) II=1 2),NN=1,NUMNP)
      IF (N EQ 1) GOTO 130
      IF (EC LT RTOL AND DFN LT RTOL) GOTO 300
      IF (ITYP EQ 2) GOTO 130


      IF (EC LT ENORM(2)) GOTO 100


C      ADJUST THE ACOEF

      ACOEF=ACOEF*0 7
      GOTO 130
100    CONTINUE

      IF (ENORM(1) GT ENORM(2) AND ENORM(2) GT EC)
    1          ACOEF=ACOEF*1 3
      IF (ACOEF GT 1 0) ACOEF=1 0


C      VELOCITY UPDATE

130    CONTINUE

      NB=0
      DO 150 I=1 NUMNP
      DO 150 J=1 2
      NB=NB+1
      IF (ITYP EQ 1) URZ(J I)=URZ(J I)+ACOEF*B(NB)
      IF (ITYP EQ 2) URZ(J I)=B(NB)
150    CONTINUE

170    CONTINUE
      UNORM(1)=UNORM(2)
      ENORM(1)=ENORM(2)
      FNORM(1)=FNORM(2)
      UNORM(2)=UC
      ENORM(2)=EC
      FNORM(2)=DFN
200    CONTINUE

C      SET FLAG

      ICONV=2
      RETURN

300    CONTINUE

C      CONVERGED CASE
C      SET FLAG


      ICONV=1
      RETURN

1010   FORMAT (/  YOU NEED MORE SPACE
              IN BLANK COMMON  )
1030   FORMAT (/ ' N R ITERATION NO    I5 /)
1050   FORMAT (/ ' DRT ITERATION NO    I5,/)
1070   FORMAT (      VELOCITY NORM          =  F15 7,/
    1              REL. ERROR NORM       =  F15 7 /
    2              REL  FORCE ERROR NORM =  ,F15 7,/)
```

```
1100  FORMAT (3X I5 3X 4Г15 7)
      END



      SUBROUTINE NORM (URZ,V UC,EROR NEQ ITYP)
      IMPLICIT DOUBLE PRECISION (A H O Z)

C
C      CALCULATE THE ERROR NORM FOR LINEAR AND
C               NONLINEAR CASE
C

      DIMENSION URZ(1) V(1)

      UC=0 0
      EROR=0 0

      DO 100 N=1 NEQ
      UC=UC+URZ(N)*URZ(N)
      IF (ITYP EQ 1) EROR=EROR+V(N)*V(N)
      IF (ITYP EQ 2) EROR=EROR+(URZ(N) V(N))**2
100   CONTINUE

      UC=DSQRT(UC)
      EROR=DSQRT(EROR)
      IF (UC NE 0 ) EROR=EROR/UC

      RETURN
      END



      SUBROUTINE POTSOL
      IMPLICIT DOUBLE PRECISION (A H O Z)

C      THIS SUBROUTINE HANDLES THE
C     POST SOLUTION PROCEDURES IE
C            GEOMETRY UPDATES
C           DIE GEOMETRY APDATE
C            STRESS EVALUATION
C            TOTAL STRAIN EVALUATION

      COMMON /TSTP/ NINI,NCUR NSEND,NITR DTMAX
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /RIGD/ RTOL,ALPH,DIAT IPLAS STK,EXN
      COMMON /RVA1/ RZ(2 250) URZ(2 250) FRZ(2 250)
                DCOORD(2 100)
      COMMON /RVA2/ EPS(5 200) STS(5 200) TEPS(200)
      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
                NSIDE URD(2 100)
      COMMON /INVR/ NOD(4 200) LNBC(2 250)
                NBCD(2 250),LOC(250)
      DATA PI/3 14159265358980D0/

C      DIE GEOMETRY UPDATES

      DO 400 N=1 NSIDE
      I=ND(1 N)
      DCOORD(1 I)=DCOORD(1 I)+DTMAX*URD(1 I)
      DCOORD(2 I)=DCOORD(2 I)+DTMAX*URD(2 I)
      IF (N EQ NSIDE) THEN
      I=ND(2 N)
      DCOORD(1 I)=DCOORD(1 I)+DTMAX*URD(1 I)
      DCOORD(2 I)=DCOORD(2 I)+DTMAX*URD(2 I)
      END IF
400   CONTINUE

C      GEOMETRY UPDATES

      DO 100 N=1 NUMNP
```

```
      IF (LOC(N) EQ 0) GOTO 500
C     Calculate the characteristic of the current segment

      I1=ND(1 LOC(N))
      I2=ND(2 LOC(N))
      X=RZ(1 N)+DTMAX*URZ(1 N)
      Y=RZ(2 N)+DTMAX*URZ(2 N)
      A=DCOORD(1 I2) DCOORD(1 I1)
      B=DCOORD(2 I2) DCOORD(2 I1)
      IF (A EQ 0 0) A=1 0D 10
      SM1=B/A

      DX1=ABS(A)
      DY1=ABS(B)

      P1P=DSQRT((DCOORD(2 I1)-Y)*(DCOORD(2 I1)-Y)+
     %          (DCOORD(1 I1)-X)*(DCOORD(1,I1)-X))
      P2P=DSQRT((DCOORD(2 I2)-Y)*(DCOORD(2 I2)-Y)+
     %          (DCOORD(1 I2) X)*(DCOORD(1 I2)-X))
      P1P2=DSQRT(DY1*DY1+DX1*DX1)

C
C      To check if this node changed the contact to another
C                die segment
C
      IF (P1P GT P2P AND P1P GT P1P2) THEN

C      Calculate the characteristic of the new segment

      LOC(N)=LOC(N)+1
      I1=ND(1 LOC(N))
      I2=ND(2 LOC(N))

      A2=DCOORD(1 I2) DCOORD(1 I1)
      B2=DCOORD(2 I2) DCOORD(2 I1)
      II (A2 EQ 0 0) A2=1 D 10
      SM2=B2/A2

      DX=ABS(A2)
      DY=ABS(B2)
      DXX=DX*P1P/P1P2
      DYY=DY*P1P/P1P2

C      Check if the old and new segments have the same slope

      IF (SM1 EQ SM2) GOTO 500

      ALPHA=DATAN(SM2)
      AL=ALPHA
      IF (B2 GT 0 0 AND A2 GT 0 0)
     1      ALPHA=ABS(ALPHA)
      IF (B2 GT 0 0 AND A2 LT 0 0)
     2      ALPHA=PI ABS(ALPHA)
      IF (B2 LT 0 0 AND A2 LT 0 0)
     3      ALPHA=PI+ABS(ALPHA)
      IF (B2 LT 0 0 AND A2 GT 0 0)
     4      ALPHA=2*PI ABS(ALPHA)
      IF (B2 EQ 0 0 AND A2 GT 0 0)
     5      ALPHA=PI

      IF (ALPHA EQ PI) THEN
      CO=DCOS(ALPHA)
      SI=0 0D0
      ELSE
      CO=DCOS(ALPHA)
      SI=DSIN(ALPHA)
      END IF

C      Calculate the new coordinate of the node

      RZ(1,N)=X
      RZ(2,N)=SM2*(X DCOORD(1 I1))+DCOORD(2 I1)
```

```
C       Boundary condition in global coordinate

        URZ(1 N)=0
        URZ(2 N)=VDIEY

C       Change the boundary code

        LNBC(1 N)=0
        LNBC(2,N)=3

        IF (ALPHA EQ 0 0 OR ALPHA EQ PI) THEN
        NBCD(1,N)=0
        NBCD(2,N)=3
        ELSE
        IF (ALPHA EQ (PI/2) OR ALPHA EQ (3*PI/2)) THEN
        NBCD(1,N)=3
        NBCD(2,N)=0
        ELSE
        NBCD(1,N)=3
        NBCD(2,N)=3
        ENDIF
        ENDIF
        GOTO 100
        ELSE

        IF (P2P GT P1P AND P2P GT P1P2) THEN
        LOC(N)=LOC(N)-1
        I1=ND(1 LOC(N))
        I2=ND(2 LOC(N))

        A2=DCOORD(1 I2)-DCOORD(1 I1)
        B2=DCOORD(2 I2) DCOORD(2 I1)
        IF (A2 EQ 0 0) A2=1 D 10
        SM2=B2/A2

        DX=ABS(A2)
        DY=ABS(B2)
        DXX=DX*P1P/P1P2
        DYY=DY*P1P/P1P2

C       Check if the old and new segments have the same slope

        IF (SM1 EQ SM2) GOTO 500

        SM2=B2/A2

        ALPHA=DATAN(SM2)
        IF (B2 GT 0 0 AND A2 GT 0 0)
     1         ALPHA=ABS(ALPHA)
        IF (B2 GT 0 0 AND A2 LT 0 0)
     2         ALPHA=PI ABS(ALPHA)
        IF (B2 LT 0 0 AND A2 LT 0 0)
     3         ALPHA=PI+ABS(ALPHA)
        IF (B2 LT 0 0 AND A2 GT 0 0)
     4         ALPHA=2*PI ABS(ALPHA)
        IF (B2 EQ 0 0 AND A2 GT 0 0)
     5         ALPHA=PI

        IF (ALPHA EQ PI) THEN
        CO=DCOS(ALPHA)
        SI=0 0D0
        ELSE
        CO=DCOS(ALPHA)
        SI=DSIN(ALPHA)
        END IF

        RZ(1,N)=X
        RZ(2,N)=SM2*(X DCOORD(1 I1))+DCOORD(2 I1)

C       Boundary condition in global coordinate

        URZ(1 N)=0
        URZ(2 N)=VDIEY
```

```
C       Change the boundary code

        LNBC(1 N)=0
        LNBC(2 N)=3

        IF (ALPHA EQ 0 0 OR ALPHA EQ PI) THEN
        NBCD(1,N)=0
        NBCD(2,N)=3
        ELSE
        IF (ALPHA EQ (PI/2) OR ALPHA EQ (3*PI/2)) THEN
        NBCD(1 N)=3
        NBCD(2,N)=0
        ELSE
        NBCD(1,N)=3
        NBCD(2,N)=3
        ENDIF
        ENDIF
        GOTO 100

        ELSE

        ENDIF
        ENDIF
500     RZ(1 N)=RZ(1,N)+DTMAX*URZ(1 N)
        RZ(2 N)=RZ(2 N)+DTMAX*URZ(2 N)

100     CONTINUE

C       STRESS EVALUATION

        DO 200 N=1 NUMEL

            AL=EPS(5,N)
            IF (IPLAS EQ 1) THEN
            CALL FLWST1 (EFSTS STRT,AL)
            ELSE
            AL1=TLPS(N)
            CALL FLWST2 (EFSTS STRT,AL1)
            END IF
        EM=(EPS(1,N)+EPS(2,N)+EPS(3 N))/3

        DO 150 I=1 3
        STS(I,N)=2 /3 *EFSTS*(EPS(I,N)-EM)/AL+DIAT*EM*3
150     CONTINUE
        STS(4 N)=EFSTS*EPS(4 N)/AL/3
        STS(5 N)=EFSTS
200     CONTINUE

C       UPDATE TOTAL EFFECTIVE STRAIN

        DO 300 N=1 NUMEL
        TEPS(N)=TEPS(N)+EPS(5 N)*DTMAX
300     CONTINUE
        RETURN
        END


        SUBROUTINE PRTINP
        IMPLICIT DOUBLE PRECISION (A H O-Z)

C
C         THIS SUBROUTINE PRINTS THE INPUT DATA
C
        CHARACTER TITLE*70
        COMMON /TITL/ TITLE
        COMMON /TSTP/ NINI,NCUR NSEND,NITR,DTMAX
        COMMON /RVA1/ RZ(2 250) URZ(2 250) FRZ(2 250)
                DCOORD(2 100)
        COMMON /RVA2/ EPS(5 200) STS(5 200) TEPS(200)
        COMMON /INVR/ NOD(4 200),LNBC(2 250),NBCD(2 250)
                LOC(250)
```

```
      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
              NSIDE URD(2 100)
      COMMON /RIGD/ RTOL,ALPH,DIAT IPLAS STK,EXN
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /INOT/ INPT MSSG IUNIT IUNI2 ISCRN

C     INPUT SUMMARY
C
      WRITE (IUNIT 1010) TITLE
      WRITE (IUNIT 1020)
      WRITE (IUNIT 1030) NINI,NSEND DTMAX
      WRITE (IUNIT 1050) ALPH DIAT
      WRITE (IUNIT 1052) IPLAS STK,EXN
      WRITE (IUNIT 1053)
      WRITE (IUNIT 1054) VDIEX VDIEY

      WRITE (IUNIT 1070) IPLNAX
      IF(IPLNAX EQ 2) WRITE (IUNIT 1071) TH
      WRITE (IUNIT 1110) FRCFAC
      WRITE (IUNIT 1130) NUMNP
      WRITE (IUNIT 1150)
      WRITE (IUNIT 1180) (N (RZ(I N) I=1 2),N=1,NUMNP)

C     PRINT NODE VELOCITY
C
      WRITE (IUNIT 1220)
      WRITE (IUNIT 1180) (N (URZ(I N) I=1 2),N=1,NUMNP)

C     ELEMENT INFORMATION
C
      WRITE (IUNIT 1270) NUMEL
      WRITE (IUNIT 1330)
      WRITE (IUNIT 1350) (N (NOD(I N) I=1 4) N=1,NUMEL)

C     BOUNDARY CONDITION
C
      WRITE (IUNIT 1400)
      WRITE (IUNIT 1430)
    1     (N NBCD(1,N),NBCD(2,N) LOC(N) N=1 NUMNP)

C     WRITE STRAIN DISTRIBUTION AT INPUT STAGE
C
      WRITE (IUNIT 1500)
      WRITE (IUNIT 1550) (N TEPS(N) N=1 NUMEL)

      WRITE(IUNIT 1560)
      WRITE(IUNIT 1570) NDIE
      WRITE(IUNIT 1580)
      DO 250 N=1 NDIE
      WRITE(IUNIT 1590)N (DCOORD(J,N),J=1 2)
250   CONTINUE


      RETURN

C     FORMATS
C
1010  FORMAT (1H1 /// 5X  OUTPUT DO F E M    J/ 5X A ///)
1020  FORMAT (5X  INITIAL INPUT SUMMARY  J//)
1030  FORMAT (    INITIAL STEP No              =   I5J
   1          FINAL STEP No             =   I5 /
   2          STEP SIZE IN TIME UNIT          =   F10 5)
1052  FORMAT (    PROCESSES CODE <PLASTIC  0
              VISCO  1>   =   I5J
   1          Y = K * E(OR E )**N  WHERE  J
   2          K            =   ,F10 5 /
   3          N          =   ,F10 5)
1050  FORMAT (    LIMITING STRAIN RATE    =   ,F15 7J
   1          PENALTY CONSTANT             =   ,F15 7)
1070  FORMAT (    DEFORMATION CODE          =   I5J
   1                  1   AXISYMMETRIC      J
   2                  2   PLAIN STRAIN  )
1071  FORMAT (    THICHNESS =    F4 1)
```

```
1110  FORMAT (   FRICTION FACTOR            =  ,F15 7,/)
1130  FORMAT (   NUMBER OF NODAL POINTS   =  ,I5,/)
1150  FORMAT (              NODE COORDINATES        //
   1          No          X Coord       Y Coord  /)
1053  FORMAT (            DIE VELOCITY        J/
   1                   X Component       Y Cocomponent ,/)
1054  FORMAT (12X 2F15 7)
1180  FORMAT (5X I5 5X 2F15 7)
1220  FORMAT (///                 NODE VELOCITY     J/
   1              No       X VELOCITY     Y VELOCITY /)
1270  FORMAT (//   NUMBER OF
              ELEMENTS          =   I5 /)
1330  FORMAT (//              ELEMENT CONNECTIVITY
   1       //    ELEM No    I     J     K     L   J)
1350  FORMAT (5I7)
1400  FORMAT (//   BOUNDARY CONDITION CODE  J/
   1       No  X1 CODE    X2 CODE   X3 CONTACT ,/)
1430  FORMAT (4I7)
1500  FORMAT (///     STRAIN DISTRIBUTION AT INPUT
   1           STAGE  J/    No        STRAIN       J)
1550  FORMAT (I5 5X F15 7)
1560  FORMAT(///    THE NUMBER OF NODES IN
   1  CONTACT WITH DIE    J  AT THE INITIAL STAGE /)
1570  FORMAT(             NDIE=  I3)
1580  FORMAT (    CONTACT NODE COORDINATES
   1    //     No         X Coord        Y Coord ,/)
1590  FORMAT (5X I5 5X 2F15 7)
      END


      SUBROUTINE PRTSOL (U)
      IMPLICIT DOUBLE PRECISION (A H O-Z)
C     THIS SUBROUTINE PRINT THE SOLUTION RESULTS

      CHARACTER ST*4,F*1 T*10 S*11 FI*2 TT*10
      CHARACTER TITLE*70 SS*5 SSS*7
      CHARACTER MSHD*4,NDED*4 EEMENTD*7
              VVEC*7 FOR*7
      CHARACTER MESHD*6,NODED*6 ELEMENTD*9
              VVECT*9 FORC*9
      CHARACTER SS1*5 SS2*5 S1S*7 S2S*7
      COMMON /FILE/ MESHD,NODED ELEMENTD
      COMMON /TITL/ TITLE
      COMMON /RIGD/ RTOL,ALPH,DIAT,IPLAS STK,EXN
      COMMON /INOT/ INPT MSSG,IUNIT IUNI2 ISCRN
      COMMON /TSTP/ NINI,NCUR NSEND,NITR DTMAX
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /RVA1/ RZ(2 250) URZ(2 250) FRZ(2 250)
              DCOORD(2 100)
      COMMON /RVA2/ EPS(5 200) STS(5 200) TEPS(200)
      COMMON /INVR/ NOD(4 200) LNBC(2,250)
              NBCD(2 250),LOC(250)
      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
              NSIDE URD(2 100)
      COMMON /BNOD/ NTOT NB1(2 250)
      DIMENSION HH1(750) F2(200)


C
C     Calculate the scale of the drawing
C
                        CALL  GSCALE
(NUMNP RZ XMIN YMIN XMAX YMAX SCALE)

C
C     CREATE FILES FOR EACH STEP SOLUTION
C
      SS= ESTRN
      ST= STEP
      SS1= ESTRR
      SS2= ESTRS
      S= 01234567890
      MSHD= MESH
      NDED= NODE
      EEMENTD= ELEMENT
      VVEC= VVECTOR
      FOR= VFORCES
      I=NCUR

      IF (I LT 10) THEN
      F=S((I+1) (I+1))
      T=ST//F// SOL
      TT=ST//F// DXF'
      MESHD=MSHD// 0 //F
      NODED=NDED// 0 //F
      ELEMENTD=EEMENTD// 0 //F
      VVECT=VVEC// 0 //F
      FORC=FOR// 0 //F
      SSS=SS// 0 //F
      S1S=SS1// 0 //F
      S2S=SS2// 0 //F

      ELSE

      J=I/10
      F1=S((J+1) (J+1))//S((I ((J 1)*10) 9) (I ((J 1)*10) 9))
      T=ST//F1// SOL
      TT=ST//F1// DXF'
      MESHD=MSHD//F1
      NODED=NDED//F1
      ELEMENTD=EEMENTD//F1
      VVECT=VVEC//F1
      FORC=FOR//F1
      SSS=SS//F1
      S1S=SS1//F1
      S2S=SS2//F1
      END IF

      NCU1=NCUR+6

C
C      CALCULATE THE EXTERNAL FORCES
C
      TF=0 0
      DO 30 I=1 NTOT
      J=NB1(1 I)
      IF (LNBC(2,J) NE 3) GO TO 30
      TF=TF+FRZ(2 J)
30     CONTINUE

C
C      CALCULATE THE DEFORMATION ENERGY
C
      DO 40 I=1 NUMEL
      U=U+STK*TEPS(5)**(EXN+1)/(EXN+1)
40     CONTINUE

C
C      CALCULATE THE REDUCTION IN HEIGHT
C
      H=VDIEY*DTMAX

      OPEN(NCU1,FILE=T STATUS= UNKNOWN )

C      PRINT NODE COORDINATES

      WRITE (NCU1 1010) TITLE NCUR DTMAX
      WRITE (NCU1 *) NUMNP NUMEL
      WRITE (NCU1 1020)
      WRITE (NCU1 1040) (N (RZ(I N) I=1 2) N=1 NUMNP)

C      PRINT NODE VELOCITY  NODAL FORCE

      WRITE (NCU1 1080)
      WRITE (NCU1 1100) (N (URZ(I N) I=1 2)
     1                  (FRZ(I N) I=1 2) N=1 NUMNP)
      WRITE (NCU1 1110) TF U H

C      STRAIN RATE  STRESS  TOTAL EFFECTIVE STRAIN

      WRITE (NCU1 1130)
      WRITE (NCU1 1180) (N (EPS(I,N) I=1 5),N=1,NUMEL)
      WRITE (NCU1 1230)
      WRITE (NCU1 1180) (N (STS(I,N) I=1 5),N=1,NUMEL)
      WRITE (NCU1 1330)
      WRITE (NCU1 1360) (N TEPS(N),N=1,NUMEL)
      WRITE (NCU1 1370)
      WRITE (NCU1 1380) (N (DCOORD(I,N),I=1,2) N=1 NDIE)
      WRITE (NCU1 1390)
      WRITE (NCU1 1400) (N,NBCD(1,N),NBCD(2 N)
     1                  LNBC(1 N) LNBC(2,N) N=1 NUMNP)

      CLOSE (NCU1)

      NCU2=NSEND+1

C
C      CREATE THE DXF FILE FOR EACH STEP SOLUTION
C
      OPEN(NCU2,FILE=TT STATUS= UNKNOWN )
      CALL DXF (NUMNP,NUMEL RZ,NOD HH1 NCU2)

C
C      Plot the the velocity vector
C
      CALL VEL (NCU2 VVECT URZ,RZ,VVEC SCALE)
C
C      Plot the forces vector
C
      CALL VEL (NCU2 FORC FRZ,RZ,FOR SCALE)
      WRITE(NCU2  (3H  0) )

C
C      PLot Isolines of the effective strain rate
C

      DO 10 I=1 NUMEL
10     F2(I)=EPS(5 I)
      CALL CONT (F2 NCU2 S1S)

C
C      PLot Isolines of the effective stress
C
      DO 20 I=1 NUMEL
20     F2(I)=STS(5 I)
      CALL CONT (F2,NCU2 S2S)

C
C      Plot the isoline contour of the effective strain
C
      CALL CONT (TEPS NCU2 SSS)

      CALL DIESEG (NCU2,NCUR)

      WRITE(NCU2  (6HENDSEC) )
      WRITE(NCU2  (3H  0) )
      WRITE(NCU2  (3HEOF) )
      CLOSE(NCU2)

      RETURN

1010   FORMAT (1H1,/// 5X  OUTPUT OF F E M  // 5X,A,//
     1        10X  SOLUTION AT STEP NUMBER  =   I5 //
     2         5X           TIME INCREMENT  =    ,F15 7 ///)
1020   FORMAT (/                NODE COORDINATES          ,//
     1                No          X Coord        Y Coord ,/)
1040   FORMAT (5X I5 5X 2F15 7)
```

```
1080  FORMAT (///  NODAL VELOCITY  AND FORCE  //
  1        NODE NO    X VELOCITY    Y VELOCITY
  1                   X FORCE       Y FORCE //)
1100  FORMAT (3X I5 3X 4F15 7)
1110  FORMAT (//  MACHINE FORCE           =  ,E14 7/
  1           ENERGY PER UNIT VOLUME  =  ,E14 7 /
  1           THE REDUCTION IN HEIGHT =   E14 7)
1130  FORMAT (///   STRAIN RATE COMPONENTS   //
  1           ELE. NO     E11         E22        E33
  1                  E12        EBAR //)
1180  FORMAT (I5 5F15 7)
1230  FORMAT (//                 STRESS COMPONENTS  //
  1           ELEM NO     S11         S22        S33
  1                  S12        SBAR  //)
1330  FORMAT (///        TOTAL EFFECTIVE STRAIN  //
  1              ELE  NO        EFFECTIVE STRAIN  //)
1360  FORMAT (5X I5 5X,F15 7)
1370  FORMAT (/      DIE NODE COORDINATES        //
  1           No        X Coord       Y Coord  /)
1380  FORMAT (5X I5 5X 2F15 7)
1390  FORMAT (/    BOUNDARY CONDITION
  1      /  No   X Code   Y Code   xi Code   eta Code )
1400  FORMAT (5I5)
      END

      SUBROUTINE REMESH (NCUR IREM)
      IMPLICIT DOUBLE PRECISION (A H O Z)
C
C     THE FUNCTION OF THIS SUBROUTINE IS TO CHECK
C          THE ELEMENTS FOR REMESHING
C  -
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /RVA1/ RZ(2 250) URZ(2 250) FRZ(2 250)
               DCOORD(2 100)
      COMMON /RVA2/ EPS(5 200) STS(5 200) TEPS(200)
      COMMON /INVR/ NOD(4 200) LNBC(2 250)
               NBCD(2 250),LOC(250)
      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
               NSIDE URD(2 100)


      DO 10 I=1 NUMEL
      N1=NOD(1 I)
      N2=NOD(2 I)
      N3=NOD(3 I)
      N4=NOD(4 I)
      D1=DSQRT((RZ(2 N1) RZ(2,N3))**2+
  1        (RZ(1,N1) RZ(1 N3))**2)
      D2=DSQRT((RZ(2 N2) RZ(2,N4))**2+
  1        (RZ(1,N2) RZ(1 N4))**2)
      IF (D1 GT D2) ERR=D1/D2
      IF (D2 GT D1) ERR=D2/D1

      IF (ERR GT 20 0) THEN
      WRITE (6 20) I,NCUR
20    FORMAT (  ELEMENT NO   I3   IS TOO DISTORTED
  1          /,REMESHING IS NEEDED AT STEP NO   I3)
      IREM=1
      RETURN
      ELSE
      GO TO 10
      ENDIF
10    CONTINUE
      RETURN
      END


      SUBROUTINE RSTFIL
      IMPLICIT DOUBLE PRECISION (A H,O Z)

C     GENERATE RESTART FILE

      CHARACTER TITLE*70
      COMMON /TITL/ TITLE
      COMMON /TSTP/ NINI,NCUR NSEND,NITR,DTMAX
      COMMON /RVA1/ RZ(2 250) URZ(2 250) FRZ(2,250)
               DCOORD(2 100)
      COMMON /RVA2/ EPS(5 200) STS(5 200) TEPS(200)
      COMMON /INVR/ NOD(4 200) LNBC(2 250)
               NBCD(2 250),LOC(250)
      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
              ,NSIDE URD(2 100)
      COMMON /RIGD/ RTOL,ALPH,DIAT IPLAS STK,EXN
      COMMON /MSTR/ NUMNP,NUMEL,IPLNAX TH,NDIE
      COMMON /INOT/ INPT MSSG IUNIT,IUNI2,ISCRN

      NN=NCUR+1
      OPEN (IUNI2,FILE= FEM RST
         FORM= FORMATTED  STATUS= UNKNOWN )
      WRITE (IUNI2 1010) TITLE
      WRITE (IUNI2 1040) NCUR NN DTMAX
      WRITE (IUNI2 1060) ALPH DIAT
      WRITE (IUNI2 1070) IPLAS STK,EXN
      WRITE (IUNI2 1060) VDIEX VDIEY
      WRITE (IUNI2 1080) IPLNAX
      WRITE (IUNI2 1060) TH
      WRITE (IUNI2 1060) FRCFAC
      WRITE (IUNI2 1080) NUMNP
      WRITE (IUNI2 1120) (N (RZ(I N) I=1 2),N=1,NUMNP)
      WRITE (IUNI2 1080) NUMEL
      WRITE (IUNI2 1090) (N (NOD(I N) I=1 4) N=1 NUMEL)
      WRITE (IUNI2 1080) NUMNP
      WRITE (IUNI2 1160) (N (NBCD(I N) I=1 2)
                   LOC(N),N=1,NUMNP)
      WRITE (IUNI2 1080) NUMNP
      WRITE (IUNI2 1120) (N (URZ(I N) I=1 2) N=1 NUMNP)
      WRITE (IUNI2 1200) (N TEPS(N),N=1,NUMEL)
      WRITE (IUNI2 1080) NDIE
      WRITE (IUNI2 1300) (N,DCOORD(1,N),DCOORD(2 N)
  1   URD(1 N) URD(2 N) N=1 NDIE)

      WRITE (IUNI2 1080) NSIDE
      WRITE (IUNI2 1085) (N (ND(I N) I=1 2) N=1,NSIDE)

      CLOSE (IUNI2)
      RETURN

1010   FORMAT (1X A)
1040   FORMAT (2I10 F20 7)
1060   FORMAT (3F20 10)
1070   FORMAT (I7 2F20 10)
1080   FORMAT (5I7)
1085   FORMAT (3I7)
1120   FORMAT (I5 2F20 10)
1160   FORMAT (4I7)
1200   FORMAT (I7 F20 10)
1300   FORMAT (I5 4F10 5)
      END


      SUBROUTINE STIFF(B,A,NEQ MBAND ITYP)
      IMPLICIT DOUBLE PRECISION (A H O Z)

C      STIFFNESS MATRIX GENERATION
C         ITYP=1   NEWTON RAPHSON ITERATION
C         ITYP=2   DIRECT ITERATION

      COMMON /INOT/ INPT MSSG IUNIT IUNI2,ISCRN
      COMMON /RVA1/ RZ(2 250) URZ(2 250),FRZ(2,250)
               DCOORD(2 100)
      COMMON /RVA2/ EPS(5 200) STS(5 200) TEPS(200)
      COMMON /INVR/ NOD(4 200) LNBC(2 250)
               NBCD(2 250) LOC(250)
      COMMON /DIES/ FRCFAC VDIEX VDIEY,ND(2 100)
               NSIDE URD(2 100)
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /RIGD/ RTOL,ALPH,DIAT IPLAS STK,EXN
      COMMON /TSTP/ NINI,NCUR NSEND,NITR DTMAX
      DIMENSION A(NEQ 1) B(1)
      DIMENSION RZE(2 4) URZE(2 4) NBCDE(2 4)
               PP(8 8) QQ(8) LM(8),LNBCE(2 4) L(4)

C   INITIALIZE LOAD VECTOR  STIFFNESS MATRIX  AND
C        NODAL POINT FORCE ARRAY

      DO 20 N=1 NEQ
      B(N)=0
      DO 20 I=1 MBAND
      A(N I)=0
20     CONTINUE

      DO 50 N=1 NUMNP
      DO 50 I=1 2
50     FRZ(I N)=0

      DO 200 N=1 NUMEL

C      CHANGE RZ  URZ  AND NBCD FROM GLOBAL
C      ARRANGEMENT TO ELEMENTAL ARRANGEMENT

      DO 100 I=1 4
      L(I)=0
      I2=I*2
      I1=I2 1
      NE=NOD(I N)
      RZE(1 I)=RZ(1,NE)
      RZE(2 I)=RZ(2,NE)
      URZE(1 I)=URZ(1,NE)
      URZE(2 I)=URZ(2,NE)
      NBCDE(1 I)=NBCD(1,NE)
      NBCDE(2 I)=NBCD(2,NE)
      LNBCE(1 I)=LNBC(1,NE)
      LNBCE(2 I)=LNBC(2,NE)
      IF (LNBCE(2 I) EQ 3) L(I)=NE
      LM(I2)=NOD(I N)*2
      LM(I1)=LM(I2) 1
100    CONTINUE
      CALL ELSHLF (PP QQ,RZE URZE EPS(1,N)
    1            TEPS(N) IPLNAX TH ITYP N L)
      IF (ITYP EQ 1) CALL NFORCE (QQ,FRZ LM)
      EFSTR=EPS(5,N)
      EFTR=TEPS(N)
      IF (FRCFAC NE 0 )
    1     CALL FRCBDY (RZE URZE LNBCE,EFSTR
    2                 EFTR QQ PP IPLNAX TH ITYP)


      CALL ADDBAN (B,A NEQ LM QQ PP)
200    CONTINUE


C      APPLY DISPLACEMENT BOUNDARY CONDITION

      CALL DISBDY (URZ LNBC B A NEQ MBAND ITYP)
      RETURN
      END



      SUBROUTINE STRMTX (RZ BB WDXJ S T IPLNAX TH
                   NEL L IDREC)
      IMPLICIT DOUBLE PRECISION (A H O Z)

C      EVALUATE STRAIN RATE MATRIX OF
C          QUADRILATERAL ELEMENT

C        BB(4 8)         STRAIN RATE MATRIX
C        RZ(2 4)         NODE COORDINATES
```

```
C         (S T)          NATURAL COORDINATE

      COMMON /INOT/ INPT MSSG IUNIT IUNI2 ISCRN
      DIMENSION RZ(2 1) BB(4 1) L(4)

      R12=RZ(1 1) RZ(1 2)
      R13=RZ(1 1) RZ(1 3)
      R14=RZ(1 1) RZ(1 4)
      R23=RZ(1 2) RZ(1 3)
      R24=RZ(1 2) RZ(1 4)
      R34=RZ(1 3) RZ(1 4)

      Z12=RZ(2 1) RZ(2 2)
      Z13=RZ(2 1) RZ(2 3)
      Z14=RZ(2 1) RZ(2 4)
      Z23=RZ(2 2) RZ(2 3)
      Z24=RZ(2 2) RZ(2 4)
      Z34=RZ(2 3) RZ(2 4)

      DXJ8=((R13*Z24 R24*Z13)+(R34*Z12 R12*Z34)*S+
    1       (R23*Z14 R14*Z23)*T)
      DXJ=DXJ8/8
      IF (DXJ GT 0 ) GOTO 10
      WRITE (MSSG 1010) NEL
      WRITE (MSSG 1030) DXJ S T
      STOP

10     CONTINUE

      X1=( Z24 Z34*S Z23*T)/DXJ8
      X2=( Z13+Z34*S+Z14*T)/DXJ8
      X3=( Z24+Z12*S Z14*T)/DXJ8
      X4=( Z13 Z12*S+Z23*T)/DXJ8

      Y1=( R24+R34*S+R23*T)/DXJ8
      Y2=( R13 R34*S R14*T)/DXJ8
      Y3=( R24 R12*S+R14*T)/DXJ8
      Y4=( R13+R12*S R23*T)/DXJ8

      DO 20 I=1 4
      DO 20 J=1 8
      BB(I J)=0
20     CONTINUE

      BB(1 1)=X1
      BB(1 3)=X2
      BB(1 5)=X3
      BB(1 7)=X4
      BB(2 2)=Y1
      BB(2 4)=Y2
      BB(2 6)=Y3
      BB(2 8)=Y4

      WDXJ=DXJ
      IF (IPLNAX NE 1) GOTO 40
      Q1=(1  S)*(1  T)*0 25
      Q2=(1 +S)*(1  T)*0 25
      Q3=(1 +S)*(1 +T)*0 25
      Q4=(1  S)*(1 +T)*0 25

      R=Q1*RZ(1 1)+Q2*RZ(1 2)+Q3*RZ(1 3)+Q4*RZ(1 4)
      BB(3 1)=Q1/R
      BB(3 3)=Q2/R
      BB(3 5)=Q3/R
      BB(3 7)=Q4/R
      WDXJ=WDXJ*R

40     CONTINUE
      BB(4 1)=Y1
      BB(4 3)=Y2
      BB(4 5)=Y3
      BB(4 7)=Y4
      BB(4 2)=X1
      BB(4 4)=X2
      BB(4 6)=X3
      BB(4 8)=X4
C       IF (IDREC EQ 2) RETURN
      CALL TRANS (L BB)
      RETURN


1010   FORMAT (/   SORRY  NEGATIVE JACOBIAN
               DETECTED AT ELEMENT NO
   1            I5)
1030   FORMAT (  DXJ S T  =   3F15 7)
      END




      SUBROUTINE VSPLON (QQ PP BB URZ,EPS
                   WDXJ IDREC)
      IMPLICIT DOUBLE PRECISION (A H O Z)

C     REDUCED INTEGRATION OF VOLUME STRAIN RATE

C      PP   =  ELEMENTAL STIFFNESS MATRIX
C      QQ   =  ELEMENTAL LOAD VECTOR
C      BB    =  STRAIN RATE MATRIX

      COMMON /RIGD/ RTOL,ALPH,DIAT IPLAS STK,EXN
      COMMON /T/ W11 W22
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /TSTP/ NINI,NCUR NSEND,NITR DTMAX
      DIMENSION PP(8 8) QQ(8) BB(4 8) URZ(1),EPS(1)
      DIMENSION D(6) XX(8) W(2)
                                           D     A     T     A
D/3*0 6666666666666667D0 3*0 3333333333333333D0/
      TWOPI=2*3 1415926535898D0
      W(1)=2 0D0
      W(2)=2 0D0

C      GENERATE DILATATIONAL STRAIN RATE MATRIX

      DO 20 I=1 8
      XX(I)=BB(1 I)+BB(2 I)+BB(3 I)
20     CONTINUE

C      CALCULATE STRAIN RATE COMPONENTS

      DO 40 I=1 5
      EPS(I)=0
40     CONTINUE

      XVOL=0
      DO 60 J=1 8
      XVOL=XVOL+XX(J)*URZ(J)
      DO 60 I=1 4
      EPS(I)=EPS(I)+BB(I J)*URZ(J)
60     CONTINUE

      EB2=(EPS(1)**2+EPS(2)**2+EPS(3)**2)*D(1)+
    1    EPS(4)**2*D(4)
      EPS(5)=DSQRT(EB2)


      DVOLU=WDXJ*W(1)*W(2)
      IF(IPLNAX EQ 2) DVOLU=DVOLU*TH
      IF(IPLNAX EQ 1) DVOLU=DVOLU*TWOPI

C      EVALUATE VOLUMETRIC CONTRIBUTION OF
C            STIFFNESS MATRIX

      DO 80 I=1 8
      IF (IDREC EQ 1) QQ(I)=QQ(I) DIAT*XVOL*
                XX(I)*DVOLU
```

```
      TEM=DIAT*XX(I)*DVOLU
      DO 80 J=1 8
      PP(I,J)=PP(I,J)+TEM*XX(J)
      PP(J I)=PP(I J)
80     CONTINUE
      RETURN
      END




      SUBROUTINE VSPLST (QQ,PP BB URZ WDXJ
                   IDREC TEPS)
      IMPLICIT DOUBLE PRECISION (A H O Z)

C      FOUR POINTS INTEGRATION OF VOLUME STRAIN
C                    RATE

C        PP  =  ELEMENTAL STIFFNESS MATRIX
C        QQ  =  ELEMENTAL LOAD VECTOR
C        BB   =  STRAIN RATE MATRIX

      COMMON /TSTP/ NINI,NCUR NSEND,NITR,DTMAX
      COMMON /RIGD/ RTOL,ALPH,DIAT IPLAS STK,EXN
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      COMMON /T/ W11 W22
      DIMENSION PP(8 8) QQ(8) BB(4 8) URZ(1)
      DIMENSION D(6) FDV(8) E(4) XX(8) TEPS(1)
      DATA D/3*0 6666666666666667D0
          3*0 3333333333333333D0/
      TWOPI=2*3 1415926535898D0

C      ELIMINATE DIALATATIONAL COMPONENT FROM
C                STRAIN RATE MATRIX

      DO 20 I=1 8
      XX(I)=(BB(1 I)+BB(2 I)+BB(3 I))/3
20     CONTINUE

      DO 40 I=1 8
      DO 40 J=1 3
      BB(J I)=BB(J I) XX(I)
40     CONTINUE

C      CALCULATE STRAIN RATE

      DO 60 J=1 4
      E(J)=0
      DO 60 I=1 8
      E(J)=E(J)+BB(J I)*URZ(I)
60     CONTINUE

      EFSR2=D(1)*E(1)*E(1)+D(2)*E(2)*E(2)+D(3)*E(3)*E(3)+
    1      D(4)*E(4)*E(4)

      IF (NITR EQ 1 AND NCUR EQ NINI AND IDREC EQ 2)
    1                EFSR2=(ALPH*100 )**2
      ALPH2=ALPH**2
      IF (EFSR2 LT ALPH2) EFSR2=ALPH2
          EFSR=DSQRT(EFSR2)
          IF (IPLAS EQ 1) THEN
          CALL FLWST1 (EFSTS STRAT EFSR)
          ELSE

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    IF (NITR EQ 1 AND NCUR EQ NINI AND IDREC EQ 2)
    1                TEPS(1)=0 3D-9
                    EFR=TEPS(1)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          CALL FLWST2 (EFSTS STRAT EFR)
          END IF

C      CALCULATE FIRST DERIVATE OF EFSR**2
```

```
      DO 80 I=1 8
      FDV(I)=0
      DO 80 J=1 4
      FDV(I)=FDV(I)+D(J)*E(J)*BB(J I)
80     CONTINUE

C     ADD POINT CONTRIBUTION TO STIFFNESS MATRIX
      DVOLU=WDXJ*W11*W22
      IF(IPLNAX EQ 2) DVOLU=DVOLU*TH
      IF(IPLNAX EQ 1) DVOLU=DVOLU*TWOPI

      F1=EFSTS/EFSR*DVOLU
      IF (IDREC EQ 2) GOTO 200
      F2=STRAT/EFSR2*DVOLU F1/EFSR2
      DO 120 I=1 8
      QQ(I)=QQ(I) FDV(I)*F1
      DO 110 J=I 8
      TEM=0
      DO 100 K=1 4
      TEM=TEM+D(K)*BB(K I)*BB(K,J)
100    CONTINUE
      PP(I,J)=PP(I J)+TEM*F1
      IF (EFSR2 LT ALPH2) GOTO 105
      PP(I,J)=PP(I J)+FDV(I)*FDV(J)*F2
105    PP(J I)=PP(I J)
110    CONTINUE
120    CONTINUE
      RETURN

200    CONTINUE
      DO 300 I=1 8
      DO 280 J=I 8
      TEM=0
      DO 250 K=1 4
      TEM=TEM+D(K)*BB(K I)*BB(K,J)
250    CONTINUE
      PP(I,J)=PP(I J)+TEM*F1
      PP(J I)=PP(I J)
280    CONTINUE
300    CONTINUE
      RETURN
      END

      SUBROUTINE DXF (NPOIN NELEM COORD NOD
                H1 NCU2)
C
C
C         THIS SUBROUTINE IS TO CREATE THE
C           DXF FILE FOR THE MESH
C
      IMPLICIT REAL*8 (A H O Z) INTEGER*4 (I N)
      CHARACTER MESHD*6,NODED*6 ELEMENTD*9
      COMMON /FILE/ MESHD NODED ELEMENTD
C
      DIMENSION COORD(2 250) NOD(4 200) H(2)
            XX(2) YY(2),H1(750) SHAPE(4)
            XGASO(750) YGASO(750)
C
C      CREATING THE DXF FILE
C
      DO 50 I=1 NPOIN
      H1(I)=0 0
50     CONTINUE
C
      NNODE=4
      WRITE(NCU2  (3H  0) )
      WRITE(NCU2  (7HSECTION) )
      WRITE(NCU2  (3H  2) )
      WRITE(NCU2  (8HENTITIES) )
      WRITE(NCU2  (3H  0) )
C
      CALL SHAPE4 (0 0D0 0 0D0 SHAPE)
```

```
      DO 20 IEI EM=1 NELEM
      XGASO(IELEM)=0
      YGASO(IELEM)=0
      DO 20 INODE=1,NNODE
      NODE=NOD(INODE IELEM)
      X=COORD(1,NODE)
      Y=COORD(2,NODE)
      XGASO(IELEM)=XGASO(IELEM)+SHAPE(INODE)*X
      YGASO(IELEM)=YGASO(IELEM)+SHAPE(INODE)*Y
20     CONTINUE

      DO 30 JELEM=1,NELEM
      I1=NOD(1 JELEM)
      I2=NOD(2 JELEM)
      I3=NOD(3 JELEM)
      I4=NOD(4 JELEM)
      XA=COORD(1 I1)
      YA=COORD(2 I1)
      XB=COORD(1 I2)
      YB=COORD(2 I2)
      XC=COORD(1 I3)
      YC=COORD(2 I3)
      XD=COORD(1 I4)
      YD=COORD(2 I4)
C
      DO 40 JNODE=1 NNODE
      INODE=JNODE
      IELEM=JELEM
      WRITE(NCU2  (4HLINE) )
      WRITE(NCU2  (3H  8) )
      WRITE(NCU2  (A) )MESHD
      WRITE(NCU2  (3H 62) )
      WRITE(NCU2  (3H 13) )
      IPOIN=NOD(INODE IELEM)
      X1=COORD(1 IPOIN)
      Y1=COORD(2 IPOIN)
      WRITE(NCU2  (3H 10) )
      WRITE(NCU2  (F10 6) )X1
      WRITE(NCU2  (3H 20) )
      WRITE(NCU2  (F10 6) )Y1
      WRITE(NCU2  (3H 30) )
      WRITE(NCU2  (3H0 0) )
      IF(INODE EQ NNODE) THEN
      INODE=1
      ELSE
      INODE=INODE+1
      END IF
      IPOIN=NOD(INODE IELEM)
      X=COORD(1 IPOIN)
      Y=COORD(2 IPOIN)
      WRITE(NCU2  (3H 11) )
      WRITE(NCU2  (F10 6) )X
      WRITE(NCU2  (3H 21) )
      WRITE(NCU2  (F10 6) )Y
      WRITE(NCU2  (3H 31) )
      WRITE(NCU2  (3H0 0) )
      WRITE(NCU2  (3H  0) )
C
      IF(INODE EQ 2 AND (INODE-1) EQ 1) THEN
      U1=ABS(YD-YA)
      U2=ABS(XA XD)
      IF(U1 GT U2) THEN
      H(1)=U1/8 0
      ELSE
      H(1)=U2/8 0
      END IF
      XX(1)=XA
      YY(1)=YA
      U3=ABS(YA YC)
      U4=ABS(XB XC)
      IF(U3 GT U4) THEN
      H(2)=U3/8 0
      ELSE
      H(2)=U4/8 0
      END IF
      XX(2)=XB
      YY(2)=YB
      ELSE
      GOTO 40
      END IF
      DO 70 II=1 2
      WRITE(NCU2  (4HTEXT) )
      WRITE(NCU2  (3H  8) )
      WRITE(NCU2  (A) )NODED
      WRITE(NCU2  (3H 62) )
      WRITE(NCU2  (3H  5) )
      WRITE(NCU2  (3H 10) )
      WRITE(NCU2  (F10 6) )XX(II)
      WRITE(NCU2  (3H 20) )
      WRITE(NCU2  (F10 6) )YY(II)
      WRITE(NCU2  (3H 30) )
      WRITE(NCU2  (3H0 0) )
      WRITE(NCU2  (3H 40) )
      WRITE(NCU2  (F10 6) )H(II)
      WRITE(NCU2  (3H  1) )
      IF(NOD(II IELEM) LT 10) THEN
      WRITE(NCU2  (I1) )NOD(II IELEM)
      ELSE IF(NOD(II IELEM) LT 100) THEN
      WRITE(NCU2  (I2) )NOD(II IELEM)
      ELSE IF(NOD(II IELEM) LT 1000) THEN
      WRITE(NCU2  (I3) )NOD(II IELEM)
      END IF
      H1(NOD(II IELEM))=H(II)
      WRITE(NCU2  (3H  0) )
      F=H(II)
70     CONTINUE
40     CONTINUE
      WRITE(NCU2  (4HTEXT) )
      WRITE(NCU2  (3H  8) )
      WRITE(NCU2  (A) )ELEMENTD
      WRITE(NCU2  (3H 62) )
      WRITE(NCU2  (3H  2) )
      WRITE(NCU2  (3H 10) )
      WRITE(NCU2  (F10 6) )XGASO(IELEM)
      WRITE(NCU2  (3H 20) )
      WRITE(NCU2  (F10 6) )YGASO(IELEM)
      WRITE(NCU2  (3H 30) )
      WRITE(NCU2  (3H0 0) )
      WRITE(NCU2  (3H 40) )
      F1=H1(NOD(1 IELEM))
      WRITE(NCU2  (F10 6) )F1
      WRITE(NCU2  (3H  1) )
      IF(IELEM LT 10) THEN
      WRITE(NCU2  (I1) )IELEM
      ELSE IF(IELEM LT 100) THEN
      WRITE(NCU2  (I2) )IELEM
      ELSE IF(IELEM LT 1000) THEN
      WRITE(NCU2  (I3) )IELEM
      END IF
      WRITE(NCU2  (3H  0) )
30     CONTINUE
      DO 80 IPOIN=1 NPOIN
      IF(H1(IPOIN) NE 0 0) GOTO 80
      WRITE(NCU2  (4HTEXT) )
      WRITE(NCU2  (3H  8) )
      WRITE(NCU2  (A) )NODED
      WRITE(NCU2  (3H 62) )
      WRITE(NCU2  (3H  5) )
      WRITE(NCU2  (3H 10) )
      WRITE(NCU2  (F10 6) )COORD(1 IPOIN)
      WRITE(NCU2  (3H 20) )
      WRITE(NCU2  (F10 6) )COORD(2 IPOIN)
      WRITE(NCU2 '(3H 30)')
      WRITE(NCU2  (3H0 0) )
      WRITE(NCU2  (3H 40) )
      WRITE(NCU2  (F10 6) )F
      WRITE(NCU2  (3H  1) )
      IF(IPOIN LT 10) THEN
      WRITE(NCU2  (I1) )IPOIN
      ELSE IF(II OIN LT 100) THEN
      WRITE(NCU2  (I2) )IPOIN
      ELSE IF(IPOIN LT 1000) THEN
      WRITE(NCU2  (I3) )IPOIN
      END IF
      WRITE(NCU2  (3H  0) )
80     CONTINUE
      WRITE(NCU2  (6HENDSEC) )
      WRITE(NCU2  (3H  0) )
      RETURN
      END

      SUBROUTINE SHAPE4 (ETA,PSI SHAPE)
C
C
C           CALCULATE THE SHAPE FUNCTION FOR THE
SHARED NODE
C
C
      IMPLICIT INTEGER*4 (I N)  REAL*8 (A H O-Z)
      DIMENSION SHAPE(4)

      S=PSI
      T=ETA
      ST=S*T

      SHAPE(1)=(1  T S+ST)*0 25
      SHAPE(2)=(1  T+S ST)*0 25
      SHAPE(3)=(1 +T+S+ST)*0 25
      SHAPE(4)=(1 +T S ST)*0 25

      RETURN
      END

      SUBROUTINE VEL (NCU2 VVECT,ZZ,RZ,FOR SCALE)
C
      IMPLICIT DOUBLE PRECISION (A H O Z)
      CHARACTER VVECT*9,FOR*7
      COMMON /MSTR/ NUMNP NUMEL IPLNAX TH NDIE
      DIMENSION TETA(250) ZZ(2 250),RZ(2 250) UR(250)
      DATA PI/3 14159265358 98D0/

      URMIN=1 E20
      URMAX= 1 E20


      IF (FOR EQ  VFORCES ) GO TO 20

      WRITE(NCU2  (7HSECTION) )
      WRITE(NCU2  (3H  2) )
      WRITE(NCU2  (6HBLOCKS) )
      WRITE(NCU2  (3H  0) )
      WRITE(NCU2  (5HBLOCK) )
      WRITE(NCU2  (3H  8) )
      WRITE(NCU2  (1H0) )
      WRITE(NCU2  (3H  2) )
      WRITE(NCU2  (3HARR) )
      WRITE(NCU2  (3H 70) )
      WRITE(NCU2  (5H   64) )
      WRITE(NCU2  (3H 10) )
      WRITE(NCU2  (3H0 0) )
      WRITE(NCU2  (3H 20) )
      WRITE(NCU2  (3H0 0) )
      WRITE(NCU2  (3H 30) )
      WRITE(NCU2  (3H0 0) )
      WRITE(NCU2  (3H  0) )


      WRITE(NCU2  (4HLINE) )
      WRITE(NCU2  (3H  8) )
WRITE(NCU2  (1H0) )
WRITE(NCU2  (3H 10) )
WRITE(NCU2  (5H0 594) )
WRITE(NCU2  (3H 20) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 30) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 11) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 21) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 31) )
WRITE(NCU2  (3H0 0) )

WRITE(NCU2  (3H  0) )
WRITE(NCU2  (4HLINE) )
WRITE(NCU2  (3H  8) )
WRITE(NCU2  (1H0) )
WRITE(NCU2  (3H 10) )
WRITE(NCU2  (5H2 259)')
WRITE(NCU2  (3H 20) )
WRITE(NCU2  (8H0 007066) )
WRITE(NCU2  (3H 30) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 11) )
WRITE(NCU2  (8H1 283358) )
WRITE(NCU2  (3H 21) )
WRITE(NCU2  (9H 0 147232) )
WRITE(NCU2  (3H 31) )
WRITE(NCU2  (3H0 0) )

WRITE(NCU2  (3H  0) )
WRITE(NCU2  (4HLINE) )
WRITE(NCU2  (3H  8) )
WRITE(NCU2  (1H0) )
WRITE(NCU2  (3H 10) )
WRITE(NCU2  (8H1 283358) )
WRITE(NCU2  (3H 20) )
WRITE(NCU2  (9H 0 147232) )
WRITE(NCU2  (3H 30) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 11) )
WRITE(NCU2  (8H1 283358) )
WRITE(NCU2  (3H 21) )
WRITE(NCU2  (9H 0 002238) )
WRITE(NCU2  (3H 31) )
WRITE(NCU2  (3H0 0) )


WRITE(NCU2  (3H  0) )
WRITE(NCU2  (5HSOLID) )
WRITE(NCU2  (3H  8) )
WRITE(NCU2  (1H0) )
WRITE(NCU2  (3H 10) )
WRITE(NCU2  (8H1 281977) )
WRITE(NCU2  (3H 20) )
WRITE(NCU2  (9H 0 002238) )
WRITE(NCU2  (3H 30) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 11) )
WRITE(NCU2  (8H2 213537) )
WRITE(NCU2  (3H 21) )
WRITE(NCU2  (9H 0 000114) )
WRITE(NCU2  (3H 31) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 12) )
WRITE(NCU2  (8H1 283358) )
WRITE(NCU2  (3H 22) )
WRITE(NCU2  (9H 0 147232) )
WRITE(NCU2  (3H 32) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 13) )
WRITE(NCU2  (8H1 283358) )
```

```
WRITE(NCU2  (3H 23) )
WRITE(NCU2  (9H 0 147232) )
WRITE(NCU2  (3H 33) )
WRITE(NCU2  (3H0 0) )

WRITE(NCU2  (3H  0) )
WRITE(NCU2  (5HSOLID) )
WRITE(NCU2  (3H  8) )
WRITE(NCU2  (1H0) )
WRITE(NCU2  (3H 10) )
WRITE(NCU2  (5H1 282) )
WRITE(NCU2  (3H 20) )
WRITE(NCU2  (6H 0 007) )
WRITE(NCU2  (3H 30) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 11) )
WRITE(NCU2  (8H2 213537) )
WRITE(NCU2  (3H 21) )
WRITE(NCU2  (9H 0 000253) )
WRITE(NCU2  (3H 31) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 12) )
WRITE(NCU2  (5H1 282) )
WRITE(NCU2  (3H 22) )
WRITE(NCU2  (5H0 138) )
WRITE(NCU2  (3H 32) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 13) )
WRITE(NCU2  (5H1 282) )
WRITE(NCU2  (3H 23) )
WRITE(NCU2  (5H0 138) )
WRITE(NCU2  (3H 33) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H  0) )
WRITE(NCU2  (4HLINE) )
WRITE(NCU2  (3H  8) )
WRITE(NCU2  (1H0) )
WRITE(NCU2  (3H 10) )
WRITE(NCU2  (5H1 282) )
WRITE(NCU2  (3H 20) )
WRITE(NCU2  (5H0 138) )
WRITE(NCU2  (3H 30) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 11) )
WRITE(NCU2  (5H1 282) )
WRITE(NCU2  (3H 21) )
WRITE(NCU2  (6H 0 007) )
WRITE(NCU2  (3H 31) )
WRITE(NCU2  (3H0 0) )

WRITE(NCU2  (3H  0) )
WRITE(NCU2  (4HLINE) )
WRITE(NCU2  (3H  8) )
WRITE(NCU2  (1H0) )
WRITE(NCU2  (3H 10) )
WRITE(NCU2  (5H2 259) )
WRITE(NCU2  (3H 20) )
WRITE(NCU2  (6H 0 007) )
WRITE(NCU2  (3H 30) )
WRITE(NCU2  (3H0 0) )
WRITE(NCU2  (3H 11) )
WRITE(NCU2  (5H1 282) )
WRITE(NCU2  (3H 21) )
WRITE(NCU2  (5H0 138) )
WRITE(NCU2  (3H 31) )
WRITE(NCU2  (3H0 0) )


WRITE(NCU2  (3H  0) )
WRITE(NCU2  (4HLINE) )
WRITE(NCU2  (3H  8) )
WRITE(NCU2  (1H0) )
WRITE(NCU2  (3H 10) )
```

```
      WRITE(NCU2  (5H0 594) )
      WRITE(NCU2  (3H 20) )
      WRITE(NCU2  (3H0 0) )
      WRITE(NCU2  (3H 30) )
      WRITE(NCU2  (3H0 0) )
      WRITE(NCU2  (3H 11) )
      WRITE(NCU2  (5H2 259) )
      WRITE(NCU2  (3H 21) )
      WRITE(NCU2  (6H 0 007) )
      WRITE(NCU2  (3H 31) )
      WRITE(NCU2  (3H0 0) )
      WRITE(NCU2  (3H  0) )
      WRITE(NCU2  (6HENDBLK) )
      WRITE(NCU2  (3H  8) )
      WRITE(NCU2  (1H0) )
      WRITE(NCU2  (3H  0) )
      WRITE(NCU2  (6HENDSEC) )

20     DO 10 I=1 NUMNP
      IF(ZZ(1 I) EQ 0 0 AND ZZ(2 I) EQ 0 0) GO TO 10
      IF(ZZ(1 I) EQ 0 0) ZZ(1 I)=1 D 10
      IF (ZZ(1 I) GE 0 0 AND ZZ(2 I) GE 0 0)
    %    TETA(I)=DATAN(ZZ(2 I)/ZZ(1 I))
      IF (ZZ(1 I) LT 0 0 AND ZZ(2 I) LE 0 0)
    %    TETA(I)=PI+DATAN(ZZ(2 I)/ZZ(1 I))
      IF (ZZ(1 I) LE 0 0 AND ZZ(2 I) GT 0 0)
    %    TETA(I)=PI+DATAN(ZZ(2 I)/ZZ(1 I))
      IF (ZZ(1 I) GT 0 0 AND ZZ(2 I) LT 0 0)
    %    TETA(I)=2*PI+DATAN(ZZ(2 I)/ZZ(1 I))

      TETA(I)=TETA(I)*180 0/PI
      UR(I)=DSQRT(ZZ(1 I)*ZZ(1 I)+ZZ(2 I)*ZZ(2 I))
      X=UR(I)
      IF (X LT URMIN) URMIN=X
      IF (X GT URMAX) URMAX=X
10     CONTINUE

      DO 30 I=1 NUMNP
      UR(I)=SCALE*UR(I)/(10 *URMAX)
      IF (UR(I) LT 1 D-3) UR(I)=1 D 3
      WRITE(NCU2  (3H  0) )
      WRITE(NCU2  (7HSECTION) )
      WRITE(NCU2  (3H  2) )
      WRITE(NCU2  (8HENTITIES) )
      WRITE(NCU2  (3H  0) )
      WRITE(NCU2  (6HINSERT) )
      WRITE(NCU2  (3H  8) )
      WRITE(NCU2  (A) )VVECT
      WRITE(NCU2  (3H  2) )
      WRITE(NCU2  (3HARR) )
      WRITE(NCU2  (3H 10) )
      WRITE(NCU2  (F10 5) )RZ(1 I)
      WRITE(NCU2  (3H 20) )
      WRITE(NCU2  (F10 5) )RZ(2 I)
      WRITE(NCU2  (3H 30) )
      WRITE(NCU2  (3H0 0) )
      WRITE(NCU2  (3H 41) )
      WRITE(NCU2  (E14 6) )UR(I)
      WRITE(NCU2  (3H 42) )
      WRITE(NCU2  (E14 6) )UR(I)
      WRITE(NCU2  (3H 43) )
      WRITE(NCU2  (3H1 0) )
      WRITE(NCU2  (3H 50) )
      WRITE(NCU2  (F10 5) )TETA(I)
      WRITE(NCU2  (3H  0) )
      WRITE(NCU2  (6HENDSEC) )
30     CONTINUE

      RETURN
      END
```

Appendix I

Publication

1 **F Diko and M S J Hashmi,** *"Integrated computer aided engineering for 2-D components"*, Proc Sixth IMC conference on Advance Manufacturing Technology, DCU, Dublin, Aug 1989

2 **F Diko and M S J Hashmi,** *"Two-dimensional finite element contact algorithm for metal forming processes"*, Proc Int Conf on Manufacturing technology, Hong Kong, 1991, pp 229- 231

3 **F Diko and M S J Hashmi,** *"A mesh and rezoning algorithm for finite element simulations of metal forming processes"*, Proc of the 7th National Conf on Production Res , Hatfield, U K ,1991

4 **F Diko and M S J Hashmi,***"Customizing a CAD system for closed die forging design"*,Proc of the Int Conf on Computer integrated manufacturing, singapore, 1991, PP 253-256

5 **F.Diko and M S J Hashmi,** *"Finite element simulation of metal forming processes and die design"*, published at the NUMIFORM92 conference on Sept 1992, Sophia Antipolis, France

6 **F Diko and M S J Hashmi,** *"Finite element simulation of non-steady state of metal forming processes"*, accepted to be published in the Journal of Materials Processing Technology and The Asia Pacific Conference on Materials processing to be held on 23 Feb 1993, Singapore

7 **F.Diko and M S J Hashmi,** *"Finite element simulation and experimental investigation of plane strain closed die forging"*, published on IMF8, Irish Materials Forum No 8, on 14 Sept 1992, UCD, Dublin

8 **F.Diko and M S J Hashmi,** *"Computer aided metal flow simulation and die design optimization for axisymmetric forging process"* to be published on the 30th International MATADOR Conference, 31th March 1993, Umist, U K