ARBITRARY LAGRANGIAN-EULERIAN FORMULATION FOR QUASI-STATIC AND DYNAMIC METAL FORMING SIMULATION

By

Hassan N. Bayoumi

B.Sc., M.Sc. Cairo University, Egypt, 1992, 1995

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES
DEPARTMENT OF MECHANICAL ENGINEERING

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

May 2001

Department of Mechanical Engineering
The University of British Columbia
2324 Main Mall
Vancouver, Canada
V6T 1Z4

Date:
June 1, 2001

Abstract

Many engineering problems involve large material deformation, large boundary motion and continuous changes in boundary conditions. The Arbitrary Lagrangian Eulerian (ALE) formulation has emerged in recent years as a technique that can alleviate many of the shortcomings of the traditional Lagrangian and Eulerian formulations in handling these types of problems. Using the ALE formulation the computational grid need not adhere to the material (Lagrangian) nor be fixed in space (Eulerian) but can be moved arbitrarily. Two distinct techniques are being used to implement the ALE formulation, namely the operator split approach and the fully coupled approach. A survey of the ALE literature shows that the majority of ALE implementations for quasi-static and dynamic analyses are based on the computationally convenient operator split technique. In addition, all previous dynamic ALE formulations are based on explicit time integration where no linearization is needed.

This thesis presents a fully coupled implicit ALE formulation for the simulation of quasi-static and dynamic large deformation and metal forming problems. ALE virtual work equations are derived from the basic principles of continuum mechanics. A new method for the treatment of convective terms that sidesteps the computation of the spatial gradients of stresses is used in the derivation. The ALE virtual work equations are discretized using isoparametric finite elements. Full expression for the resulting ALE finite element matrices and vectors are given. A new relation that relates grid displacements with material displacements is introduced.

The ALE finite element equations are implemented into a 2-D computer code for plane stress, plane strain and axisymmetric problems. The transfinite mapping method is used as the mesh motion scheme for internal nodes. A new treatment for mesh motion on material boundaries is introduced and implemented. Implicit, explicit and mixed implicit-explicit time integration schemes are implemented in the code. A line search technique is employed to accelerate the convergence of implicit calculations.

Several quasi-static and dynamic large deformation applications are solved using the developed code. Experimental analysis of a simple V-bending process is conducted for comparison. Comparison of ALE predictions for deformed shapes, equivalent stress and plastic strain distributions and loading curves with analytical, numerical and experimental results are presented. ALE results are in good agreement with other methods of analysis. ALE is shown to prevent mesh distortion and eliminate the need for special contact treatments for problems with corner contact.

Table of Contents

List of Tables

List of Figures

Nomenclature

${}^{t}(\,)$	quantity occurs at time t
${}_{t}(\,)$	quantity referred to time t
${}^{t}\bar{\mathbf{a}}$	element nodal referential material acceleration vector
$\bar{\mathbf{a}}$	element nodal incremental referential material acceleration vector
${}^{t}a_{i}$	components of referential material acceleration vector
a_{i}	components of incremental referential material acceleration vector
${}_{t}\mathbf{B}^{A1}$	element shape function derivative matrix related to first ALE stiffness
${}_{t}\mathbf{B}^{A2}$	element shape function derivative matrix related to second ALE stiffness
${}_{t}\mathbf{B}^{L1}$	element shape function derivative matrix related to Lagrangian material stiffness
${}_{t}\mathbf{B}^{L2}$	element shape function derivative matrix related to Lagrangian geometric stiffness
${}^{t}\mathbf{C}$	element elastic-plastic material constitutive matrix
${}^{t}\mathbf{C}^{A1}$	element first convective velocity-stiffness matrix due to ALE
${}^{t}\mathbf{C}^{A2}$	element second convective velocity-stiffness matrix due to ALE
${}^{t}\mathbf{C}^{A3}$	element third convective velocity-stiffness matrix due to ALE
${}^{t}\mathbf{C}^{A4}$	element fourth convective velocity-stiffness matrix due to ALE
${}^{t}C_{ijkl}$	components of fourth order material constitutive tensor
${}^{t}D_{kl}$	components of rate of deformation tensor
${}^{t}dV$	elemental volume

${}^{t}dS$	elemental surface area
E	Young’s modulus
E^{p}	plastic modulus
${}_{t}\mathbf{e}$	element strain vector
${}_{t}e_{ij}$	components of strain tensor
$\mathbf{f}^{*}$	element equivalent nodal force vector
${}^{t}\mathbf{f}$	element internal force vector
${}^{t}f$	arbitrary function
${}^{t}\dot{f}$	material derivative of arbitrary function
${}^{t}f'$	grid derivative of arbitrary function
${}^{t}f_{i}^{B}$	components of body force vector per unit mass
${}^{t}f_{i}^{S}$	components of surface traction vector
$\mathbf{H}$	element interpolation matrix
$\mathbf{I}$	identity matrix/tensor
h_{α}	element shape function at node α
${}^{t}\mathbf{K}^{*}$	element equivalent stiffness matrix
${}^{t}\mathbf{K}^{A}$	element convective stiffness matrix due to ALE
${}^{t}\mathbf{K}^{A1}$	element first convective stiffness matrix due to ALE
${}^{t}\mathbf{K}^{A2}$	element second convective stiffness matrix due to ALE
${}^{t}\mathbf{K}^{L}$	element Lagrangian stiffness matrix

${}^{t}\mathbf{K}^{L1}$	element Lagrangian material stiffness matrix
${}^{t}\mathbf{K}^{L2}$	element Lagrangian geometric stiffness matrix
${}^{t}\mathbf{M}^{A}$	element convective mass matrix due to ALE
${}^{t}\mathbf{M}^{L}$	element Lagrangian mass matrix
N	number of element nodal points
${}^{t}n_i$	components of unit normal to boundary surface
${}^{t}\mathbf{S}^{A1}$	element stress matrix related to first ALE stiffness
${}^{t}\mathbf{S}^{L2}$	element stress matrix related to Lagrangian stiffness
s	line search parameter
t	time
$\mathbf{u}$	element incremental material displacement vector
$\mathbf{u}^{g}$	element incremental grid displacement vector
$\overline{\mathbf{u}}$	element nodal incremental material displacement vector
$\overline{\mathbf{u}}^{g}$	element nodal incremental grid displacement vector
u_i	components of incremental material displacement vector
u_i^{g}	components of incremental grid displacement vector
$\overline{u}_{ik}$	components of nodal incremental material displacement vector
$\overline{u}_{ik}^{g}$	components of nodal incremental grid displacement vector
$\overline{\mathbf{v}}$	element nodal incremental material velocity vector
$\overline{\mathbf{v}}^{g}$	element nodal incremental grid velocity vector
${}^{t}\overline{\mathbf{v}}$	element nodal material velocity vector

${}^{t}\bar{\mathbf{v}}^{g}$	element nodal grid velocity vector
v_{i}	components of incremental material velocity vector
v_{i}^{g}	components of incremental grid velocity vector
${}^{t}v_{i}$	components of material point velocity vector
${}^{t}v_{i}^{g}$	components of grid point velocity vector
${}^{t}\dot{v}_{i}$	components of material acceleration vector
${}^{t}v_{i}'$	components of material referential acceleration vector
${}^{t}W^{ext}$	external work
${}^{t}\mathbf{x}$	element coordinate vector
${}^{t}\bar{\mathbf{x}}$	element nodal coordinate vector
X_{i}^{m}	components of material point initial coordinates
X_{i}^{g}	components of grid point initial coordinates
${}^{t}x_{i}$	components of material and grid points position vector
${}^{t}x_{i}^{m}$	components of material point mapping function
${}^{t}x_{i}^{g}$	components of grid point mapping function
${}^{t}\bar{x}_{ik}$	components of nodal coordinate vector
${}^{t}y_{ijk}$	components of stress-velocity product tensor
$\boldsymbol{\alpha}$	mesh motion parameter vector
α_{j}	components of mesh motion parameter vector

$\mathbf{B}$	mesh motion parameter matrix
β	Newmark integration parameter
$\beta_{(j)}$	components of mesh motion parameter vector
Δt	time increment
δ	virtual or variation
ε_{eq}^{p}	equivalent strain
ϕ_i	boundary curve
γ	Newmark integration parameter
ν	Poisson's ratio
θ	angle of inclination of x' and y' axes to the global x and y axes
${}^{t}\rho$	material density
${}^{t}\boldsymbol{\sigma}$	element stress vector
σ_{eq}	equivalent stress
σ_y	initial yield stress
${}^{t}\sigma_{ij}$	components of Cauchy stress tensor
${}^{t}\sigma_{ij}^{T}$	components of Truesdell stress rate tensor
$\overline{\sigma}_{ij\alpha}$	nodal stress components at node α

Acknowledgement

Sincere thanks and appreciation are expressed to my supervisor, Professor Mohamed Gadala, for suggesting the area of this research and for his invaluable advice and continuous guidance. His knowledge and experience greatly influenced this thesis.

I am very glad for having the opportunity to study at the University of British Columbia. I would like to acknowledge the funding of my supervisor as well as of the Faculty of Graduate Studies. I am honored to be a recipient of the University Graduate Fellowship for four years.

I would also like to thank my professors and previous supervisors. Special thanks are due to Professors Salah Bayoumi, Mohsen El-Mahdi, Ahmed Mansour and Aly El-Shafei. I am greatly indebted to them all.

I am grateful to my parents, my brother and my friends for their steadfast support. Thanks are due to my friend Ahmed El-Bindari for being a constant source of help and encouragement.

Finally, a very special thank you goes to my wife, Ingy, for her patience, love and dedication.

Chapter 1

INTRODUCTION

1.1 MOTIVATION

The history of metal forming is based on a highly empirical, nearly artisan form of technology. Perhaps one of the most expensive and time-consuming stages of the metal forming design and manufacturing cycle is prototype development. Prior to 1980's, prototype development was mainly based on trial and error. This is in a large part due to the fact that metal forming problems were too complex to model using the available tools.

The decade of the 1980's resulted in revolutionary technological advancements for metal forming industries. The vast increase in computational capabilities together with the recent advances in numerical techniques made it possible to simulate metal forming processes in the design stage. The use of the finite element method in the metal forming design process is practiced by a wide range of industries including aerospace, automotive, medical, defense and other diverse production companies. Metal forming simulation can predict possible forming defects, potential operating problems and provide optimum tooling design and running conditions leading to the reduction or elimination of the costly tool prototyping. As a by-product of the simulation, a better understanding of the mechanical properties of forming processes can be acquired. Prediction of forming loads, final workpiece geometry, potential workpiece cracking or damage, incomplete die filling, residual stresses, excessive work hardening, sheet metal wrinkle, tearing and spring-back, optimum tooling curvatures, tool wear, and blank dimensions are examples of the areas that can be addressed by process simulation. Though metal forming simulation has significantly improved in recent years, it is noted, however, that the accuracy and utility of metal forming simulation are still questionable.

1.2 PROBLEM DESCRIPTION

Metal forming processes are complex problems that generally involve large deformation, nonlinear material behavior, large boundary motion and interaction of the workpiece with the forming tools and dies. Finite element simulation of metal forming problems necessitates the use of a large strain formulation that can handle material and geometric nonlinearities.

1.2.1 Traditional large strain formulations

Foundations of large strain analysis of elastic-plastic solids can be traced back to the early work of Hill [1]. The first finite element formulation for large strain problems, known as the Total Lagrangian formulation, was introduced by Hibbitt et al. [2]. Later on, McMeeking and Rice [3] pioneered the use of the Updated Lagrangian formulation. The conceptual difference between the two formulations is the reference configuration that is used for the linearization of the incremental equations of motion. In the Total Lagrangian formulation the initial configuration is used as a reference, whereas in the Updated Lagrangian formulation, the reference configuration corresponds to the last calculated configuration. Being referential in nature, both formulations should give the same results [4], provided that the two formulations are consistently driven.

Finite element computer codes that are currently being used in metal forming simulation can be classified into two categories: general-purpose finite element codes [5-10] and codes that are developed specifically for metal forming simulation [11-16]. Both types of codes mainly depend on the Lagrangian (referential) formulation to simulate the large deformation behavior that is inherent in metal forming problems. In the Lagrangian formulation, the finite element mesh, or reference configuration, is fixed to the material points of the deforming body and the grid moves with the same velocity as the material. In the case of large strain problems, this can lead to excessive distortion of the finite element mesh and presents a major drawback of the formulation. Distorted meshes reduce the accuracy of numerical integrations and may ultimately result in singular matrices and computation termination. Another drawback of the Lagrangian approach is the difficulty to model non-material-associated boundary conditions. Figure 1.1 shows a punch indentation problem that has been analyzed using a general-purpose finite element package, NISA [7], which is based on the Lagrangian formulation. Because of symmetry, only half of the geometry is being modeled. The punch movement has been described as a downward constant velocity applied at the nodes right below the punch. It is required to estimate the forming loads for a 60 % reduction in the height of the workpiece. The figure shows that as the material moves under the punch load, the finite element mesh, being attached to the material, also moves with the effect of increasing the punch diameter. It is also clear from the figure that starting at approximately 50 % reduction in

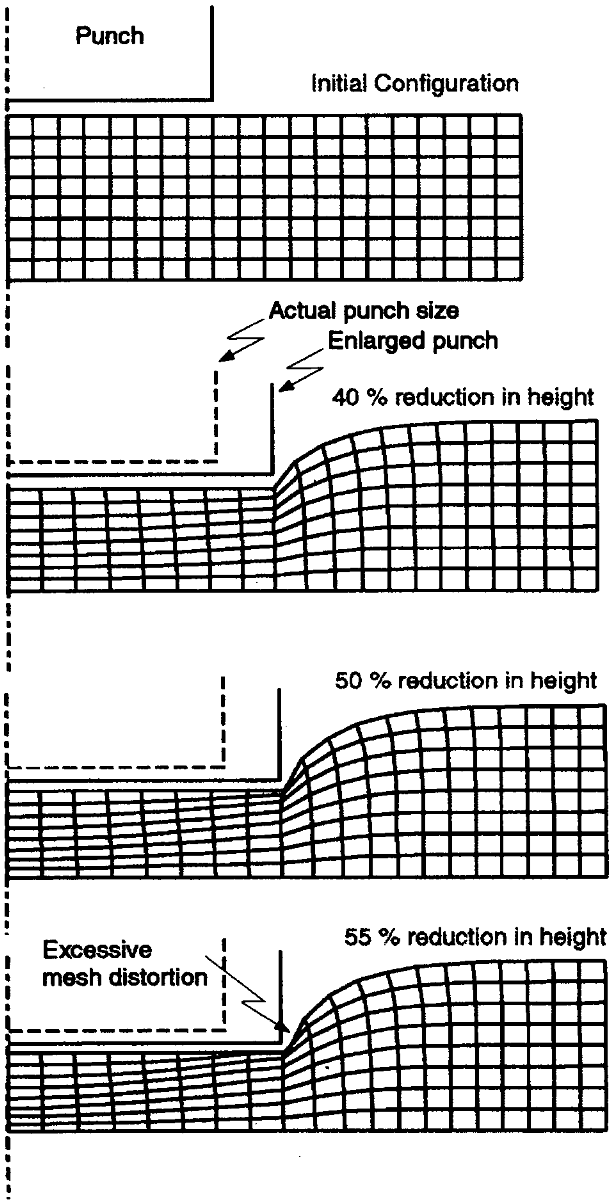


Figure 1.1. Simulation of punch indentation using the Lagrangian approach (NISA).

height, the finite element mesh was excessively distorted and the run was terminated at a 55 % height reduction. A more detailed description of the problem is given in Chapter 6.

Using general-purpose codes, the only remedy for the mesh distortion problem is to remesh the material domain with a new finite element mesh whenever needed [17]. The material properties for the new mesh are found from the corresponding old material points by interpolation between the old and new Gauss points, where certain approximations have to be introduced. Computations are then restarted from the last converged step. Remeshing, besides being very time consuming, sometimes requires user intervention. The simulation process may need frequent remeshing to reach the required deformation level. The analyst must either run the code interactively or have enough knowledge of the solution to write a detailed set of remeshing instructions ahead of time. Only a few general-purpose codes have automatic remeshing capabilities. On the other hand, metal forming codes are usually equipped by the auto-remesh capability [11]. Automatic remeshing, though still very time consuming, can be adjusted to take place either at a specified number of incremental load steps or when excessive mesh distortion occurs. However, the problem of incorrect modeling of non-material-associated boundary conditions will persist between remeshes in addition to the approximations of the assumed interpolation.

Meanwhile, several attempts have been aiming at the adaptation of Eulerian (spatial) formulation, widely used in fluid flow simulations, to large strain and metal forming problems [19]. Some special metal forming codes use the Eulerian formulation for the simulation of steady state forming processes such as extrusion and drawing [20]. In the

Eulerian description, the finite element mesh is fixed in space. The obvious drawbacks of the Eulerian description are the difficulty to model changing material boundaries and the difficulty to model material-associated boundary conditions. This limits the applicability of the Eulerian description to the analysis of a small class of metal forming problems of the steady-state type. An extensive bibliography of the application of finite element techniques in metal forming simulation is given in [21].

1.2.2 The Arbitrary Lagrangian Eulerian (ALE) formulation

The above discussion indicates the need for a formulation that is more suited for the simulation of large deformation problems with large boundary motions. The Arbitrary Lagrangian Eulerian (ALE) formulation has emerged in recent years as a technique that can alleviate many of the drawbacks of the traditional Lagrangian and Eulerian formulations [22, 23]. In the ALE formulation, the finite element mesh, or reference configuration, need not adhere to the material nor be fixed in space but can move arbitrarily. As the material deforms, the finite element mesh is continuously moved to meet any preset criterion (e.g. optimize elements shape) and the simulation should be completed without user intervention. Combining the merits of both the Lagrangian and Eulerian formulations, ALE can easily describe all types of boundary conditions and prevent mesh distortion. A proper ALE formulation should reduce to both the Lagrangian and Eulerian formulations as necessary.

Different approaches have been adopted to develop the ALE formulation. Since the pure Lagrangian and the pure Eulerian viewpoints have complementary virtues, researchers have tried to produce a formulation that can combine the best of both. The differences among the available ALE formulations depend on the intended application, assumptions made in deriving the ALE equations and details of implementation. These factors determine the limitations and accuracy of the resulting formulation. A survey of the main features of the various ALE formulations available in the literature is given in Chapter 2.

ALE is usually termed a coupled formulation since material deformation and convective effects are coupled in the same equations. However, two distinct techniques are being used to implement the ALE equations. The first technique is referred to as an 'operator split' or a 'fractional step' approach. Virtually all ALE analyses, with very few exceptions, are based on this strategy. In this approach, material deformation and convective effects are treated separately. Thus each time step may be divided into two steps: a regular Lagrangian step followed by an Eulerian step. In the Lagrangian step, the grid moves with the material, whereas in the Eulerian step, the Lagrangian solution is mapped to the reference grid and stresses are updated using convective effects. Time advances only during the Lagrangian step and there is no time associated with the Eulerian step. Thus it is not necessary to perform an Eulerian step in every time step. The alternative ALE approach, which has been used by fewer researchers, is known as the 'fully coupled' or the 'unsplit' approach. In this approach, the governing ALE equations are implemented and solved without disruption. A solution algorithm that can handle convective effects simultaneously with material deformation must be used.

The main advantage of the operator split over the fully coupled approach is the reduction in the cost of implementation of ALE into current Lagrangian codes as the Lagrangian step is unchanged and only the Eulerian step algorithm needs to be added. Moreover, the decoupling of the Lagrangian and Eulerian steps results in simpler equations to be solved and simpler algorithms to be used. However, from the theoretical point of view, the fully coupled ALE approach represents a true kinematic description that employs a more rigorous scheme in considering equilibrium in each step relative to a moving reference configuration. Therefore, the operator split solution, which stems from computational convenience, is expected to be less accurate than the fully coupled solution. Although most of the ALE literature is based on the operator split methods, the author believes that as computers become faster, unsplit methods will probably dominate because of their theoretical accuracy advantage.

As shall be inferred from the ALE literature review section, no attempt has been made to develop and implement a true fully coupled ALE formulation for general quasi-static and dynamic solid mechanics analyses. In addition, most of the available ALE formulations for solid mechanics applications are based on a generalization of the Eulerian formulation in which velocities are used as the independent variables. This treatment is not consistent with the traditional Lagrangian formulation that is more suited for solid mechanics applications and in which displacements are more natural to use as independent variables. The author believes that this treatment is one of the reasons for the misconception that ALE is more suited for fluid mechanics applications and for the delay in the wide use of ALE in solid mechanics applications.

In deriving the ALE equations, the relationship between the material time derivative, grid time derivative and spatial derivatives of arbitrary quantities is substituted into the governing equations. This substitution gives rise to convective terms in the ALE equations which account for the transport of material through the grid. Convective terms are the terms that involve the spatial gradients of quantities such as stresses. Since the values of these quantities are available at the integration points, and not at the nodal points connecting the elements, these quantities are generally discontinuous across element edges and their gradients cannot be reliably computed on the element level. The problem of evaluating convective terms presents a difficult task in the ALE formulation and different numerical assumptions and treatments are being used. As a result, the treatment of convective terms is a key factor that distinguishes the different ALE formulations. Details of the derivation of the equations governing the ALE formulation are given in Chapter 3.

Using the ALE formulation, the finite element mesh can be moved arbitrarily to maintain a homogeneous mesh and properly represent boundary conditions throughout the deformation process. A mesh motion scheme is necessary to continuously adapt the positions of internal nodes. On the boundaries, however, mesh motion must satisfy the boundary constraint that prevents the relative motion between the material points and mesh points in the direction normal to the boundary. This constraint ensures that the material and mesh configurations have the same boundary at all times. Implementation of the boundary constraint is not a straightforward task and some ALE developers have sidestepped its implementation completely by simply assuming Lagrangian boundaries and remeshing between load increments.

The main drawback of the ALE technique lies in the fact that the number of unknowns is increased, being both the material and mesh displacements. In addition, ALE matrices are generally unsymmetric and require an unsymmetric equation solver. Fully coupled ALE computations are expected to be more time consuming than the corresponding Lagrangian ones especially for practical metal forming problems with large number of elements and complex contact conditions. This indicates the necessity of incorporating an efficient solution algorithm when using ALE. As indicated in Chapter 2, in a quasi-static ALE formulation, solution of equations is limited to implicit techniques in which iterations and convergence checks must be employed. However, using a dynamic ALE formulation, computations may be performed using implicit, explicit or mixed implicit-explicit calculations and solution is always much faster and easier to achieve.

1.3 SCOPE OF WORK

The objective of this research is to develop and employ a true fully coupled ALE formulation in the simulation of large deformation metal forming problems. The formulation is aimed to be applicable to both quasi-static and high-speed metal forming processes.

The scope of work may be summarized in the following:

- Derivation of fully coupled ALE virtual work equations from the basic principles of continuum mechanics for both quasi-static and dynamic analyses. Using displacements as the independent variables, a consistent ALE formulation is sought that can be easily related to standard Lagrangian formulations for solid mechanics applications.

- Introduction of a new method for the treatment of convective terms in the equilibrium equations that avoids the resort to unjustified assumptions in calculating spatial gradients of stresses. This treatment aims to maintain the consistency and generality of the developed formulation.
- Discretization of the ALE virtual work equations using isoparametric finite elements. An effort is required to put the different virtual work terms in an easy-to-compute matrix form consistent with standard nonlinear finite element calculations.
- Enhancement of the mesh motion scheme especially for boundary nodes. Boundary nodes should be allowed to move in the tangential direction to material boundaries to maintain a uniform distribution while satisfying the ALE boundary constraint.
- Implementation of the ALE formulation into a 2-D computer code for plane stress, plane strain and axisymmetric problems. The implementation is intended to be modular and convenient for future developments.
- Implementation of efficient time integration schemes that allow for implicit, explicit and mixed implicit-explicit calculations. Implementation of a line search technique to accelerate the convergence of implicit calculations.
- Implementation of a simple contact algorithm to allow the simulation of metal forming problems involving workpiece-tool interactions.

- Application of the developed code in the simulation of several large deformation solid mechanics and metal forming problems. Comparison of results with experimental measurements or with other well established numerical techniques.

Chapter 2

BACKGROUND

2.1 ALE LITERATURE REVIEW

The concept of ALE was first proposed for fluid mechanics applications using the finite difference analysis [22, 23]. ALE was later introduced into the finite element analysis of fluid flow problems [24]. In this work, Hughes et al. elaborated on one of the basic concepts in ALE, which is the relationship between the material time derivative, the grid time derivative and the spatial gradients of any arbitrary quantity. This relationship is very important in deriving the governing ALE equations as indicated in Chapter 3.

The ALE technique was also applied in fluid-structure interaction problems to model the fluid domain while the structure domain was handled using the usual Lagrangian description [25]. The reason for the delay in the application of ALE to solid mechanics problems has been primarily due to the complexities of updating history dependent properties at integration points for arbitrary moving meshes.

Huétink [26] introduced the first ALE formulation for the analysis of solid mechanics applications. Huétink used the convenience of the operator split technique in his finite element simulations of quasi-static metal forming problems. He proposed a method for updating variables at integration points. He first calculated the variables at each nodal point by averaging the integration point values from all elements that share this point. A continuous field for each variable is obtained by interpolating nodal point values using element shape functions. The gradients may then be computed using the derivatives of shape functions. This method was later refined by applying local and global smoothing of these variables to avoid numerical instabilities [27]. This technique is highlighted in Chapter 3.

Schreurs and co-workers [28] discussed a similar ALE procedure. In this work, only fundamental ideas of ALE were discussed and formulation details were not given. The same authors [29] presented algorithms for the control of mesh quality, an important aspect of any ALE formulation. In controlling mesh quality, attention is mainly focussed on optimizing the element shape, while the topology of the element mesh is not changed. This is the reason that this process is often called 'mesh motion' as opposed to 'remeshing' which is commonly used in Lagrangian analyses and which usually involves optimization of element shape as well as element size.

Haber [30] presented an uncommon form of the ALE formulation termed Eulerian Lagrangian Description (ELD). In this description, the total deformation in each increment is divided into separate Eulerian incremental displacements and Lagrangian incremental displacements. An Eulerian deformation gradient defines the mapping from the initial configuration to the reference configuration while a Lagrangian deformation gradient describes the mapping from the reference configuration to the current configuration. The total deformation gradient from the initial to the current configurations is then the product of the Eulerian and Lagrangian deformation gradients. The ELD was successfully extended to include dynamic effects [31] and applied to analyze dynamic crack propagation problems [32]. However, the introduction of two sets of displacements in this ALE formulation makes it difficult to relate to other formulations and difficult to implement into existing Lagrangian codes.

Liu, Belytschko and Chang [33] presented an ALE formulation for path-dependent materials using explicit time integration. A new method for the treatment of the convective term was proposed. Although they discussed ALE equations for implicit calculations, their implementation was explicit. Explicit calculations do not require the decompositions and linearization involved in implicit calculations. The method of finding a continuous stress field by interpolation developed by Huétink [26] results in implicit calculations that are not compatible with the explicit time integration used. By defining a stress-velocity product, the computation of stress gradients was circumvented. However, this method requires interpolation of the stress-velocity product using shape functions. Thus, mixed finite elements, as opposed to the more common displacement based finite elements, must be used. This method is briefly described in Chapter 3.

Liu et al. [34, 35] derived a fully coupled implicit quasi-static ALE formulation in rate form. The independent variables in this work are velocities as opposed to displacements. The stress-velocity product technique, previously developed by the same research group [33], was used to handle the convective effects. The resulting ALE equations are difficult to relate to the incremental form of the displacement based Lagrangian formulation commonly used in solid mechanics simulations.

Benson [36] proposed a Simple ALE (SALE) formulation with the aim of reducing the cost of analysis and implementation into current Lagrangian codes. The SALE formulation is an ALE formulation that is limited to a single material in each element as well as using Lagrangian boundaries. He used the operator split technique with the central difference explicit time integration scheme.

Huerta & Casadei [37] presented several ALE dynamic applications using explicit time integration. Since their calculations are explicit, they employed the same stress-velocity product technique developed by Liu et al. [33] to update stresses. They showed that ALE is a strong competitor to the classical Lagrangian formulation for fast-transient dynamic applications such as impact and coining.

Extended from fluid mechanics, an implicit ALE formulation for solid mechanics problems was given by Ghosh and Kikuchi [38]. They indicated that the majority of ALE formulations available in the literature were based on explicit time integration. Explicit methods suffer from the lack of generality of application due to the stringent stability conditions, which usually necessitate the use of very small time steps. Consequently, an implicit ALE formulation was sought. In this work, however, it was assumed that the motion of the material points is quasi-static with respect to grid points, i.e. the time rate of change of the material velocity for a fixed grid point may be neglected. This simplified the derivation of the ALE equilibrium equations by neglecting terms that partially contribute to the material point acceleration. This assumption is quite common in steady-state analyses based on the Eulerian formulation in which all partial derivatives with respect to time are neglected. However, in ALE, the grid is not fixed and its motion is arbitrary. A solution for the ALE equations based on steady-state assumptions is physically meaningless if the grid velocity is to be arbitrarily modified. For quasi-static problems, it is the total material point acceleration that may be neglected. This formulation is, therefore, incorrect and may not be warranted for quasi-static nor transient applications.

An ALE formulation for quasi-static applications was developed by Wang [39]. Fully coupled ALE equations were derived in a fashion similar to that of Liu et al. [35]. Starting with the usual virtual work, integrations over spatial coordinates were first transformed into integrations over referential coordinates. Then by taking the rate of change of the virtual work expression with the reference coordinates held constant and transforming it back to the spatial coordinates, the rate form of the ALE expression of virtual work, i.e. virtual power, was obtained. Many of the basic procedures related to ALE, such as the mesh motion scheme and the need for unsymmetric solvers, were addressed in this work. An in-house finite element code was developed based on this formulation. Several quasi-static metal forming problems were successfully simulated showing the effectiveness of ALE. However, the following comments could be made on this work:

- Although fully coupled ALE equations were derived, the implementation was not a strictly fully coupled one. In this work, stress calculation followed a Lagrangian procedure and the extra convective terms related to ALE were not handled within the iterations of each load increment. A separate stress updating routine was used to update material associated properties, such as stresses, strains, and current yield surface after iterations converge and before the start of a new load increment. The reason for this undertaking was mainly to avoid convergence problems. Since fully coupled stiffness equations were solved, this formulation is a mix between the fully coupled and the operator split techniques.

- The treatment of convective terms in the fully coupled equations was based on the assumption of a continuous stress field by shape function interpolation developed by Huétink [26]. The same assumption was used to update stresses in the separate stress updating routine.
- Similar to the ALE formulations developed for fluid mechanics applications, the ALE equilibrium equations used in this work were derived from the principle of virtual power in which velocities are used as independent variables. This fact, together with assumptions used in the treatment of convective terms, resulted in ALE equations that can not be easily correlated with the standard Lagrangian formulation.
- The equations developed in this work were limited to quasi-static applications. The absence of inertia effects precludes the simulation of fast transient metal forming processes. In addition, quasi-static formulations necessitate the use of an implicit solver in which convergence problems are always expected.
- The ALE boundary constraint was not implemented in this work. Within each load increment, boundary nodes were assumed to be pure Lagrangian. Although this treatment is theoretically correct, it does not ensure that boundary nodes are uniformly spaced on the boundaries at the end of each load increment. To overcome this problem, a remeshing routine was used to remesh material boundaries with new nodes between increments. Material properties for the new nodes were obtained from the old ones by interpolation. The remeshing of material boundaries between

increments is another reason for not considering this work a true fully coupled ALE implementation.

- The finite element code developed in this work had limited capabilities. Only two-dimensional 4-node plane strain isoparametric elements were incorporated. No contact analysis capabilities were implemented.

2.2 DYNAMIC EFFECTS

Dynamic effects play an important role in metal forming simulation. Including dynamic effects extends simulation capabilities to include fast transient metal forming processes. In addition, dynamic effects enhance the characteristics of the nonlinear finite element solution.

2.2.1 Fast transient metal forming

The need for a higher production rate is a worldwide trend that gave rise to increased metal forming speeds [40-43]. Steinmann et al. [44] indicated that in sheet metal forming processes, high-speed deformations are very likely to occur. The interest is therefore to simulate the short-term transient response to these loading conditions and thus a solution strategy that can efficiently trace dynamic deformation behavior is required.

Baillet et al. [15] developed a finite element program, based on the dynamic explicit method, for the simulation of the ironing process. It is shown that the higher the punch speed, the more the sheet lifts away from the die, thus changing the contact conditions and strain distribution. Similarly, in a study of the static and dynamic forming of circular plates [45], it is shown that, while in static deformation the contact between the ram and the plate is maintained throughout the deformation history, in the dynamic case the contact between the plate and the ram changes. Fontane and Gelin [46] simulated several high speed metal forming processes, namely rolling, upsetting and wire drawing. In the rolling process simulation, the roll pressure was found to increase with the increase in rolling speed. In the wire drawing process, the optimal die angle and drawing stress increased with the increase in drawing speed. In the upsetting test, it is seen that the shape of the deformed mesh strongly depends on the upsetting speed. They concluded that the inclusion of dynamic effects in such calculations leads to modifications in the plastic flow and load distribution during forming processes and thus it is of the utmost importance. Kapinsiki studied the influence of punch velocity on the material deformation during deep drawing [47]. He noticed that punch velocity significantly influences the stress and strain distributions and that the quasi-static assumption causes large discrepancies between theoretical and experimental results. He concluded that it is necessary to take inertial effects into account to be in agreement with experimental observations.

The above examples indicate that it is sometimes necessary to retain inertial terms in developing an ALE formulation for general metal forming simulation.

2.2.2 Solution of nonlinear equilibrium equations in finite elements

A common solution approach for the nonlinear finite element equilibrium equations is the step-by-step incremental procedure in which it is assumed that solution for all equilibrium positions for all time steps from time 0 to time t have been obtained, and the solution for the next equilibrium position at time $t+\Delta t$ is required. This process is applied repetitively until the complete solution path has been solved for. An implicit solution scheme is adopted if the solution for equilibrium at time $t+\Delta t$ is based on using equilibrium conditions at time $t+\Delta t$, whereas an explicit scheme is used if equilibrium at time $t+\Delta t$ is based on using equilibrium conditions at time t. In quasi-static analyses, solution of equilibrium equations is always based on the implicit scheme. On the other hand, it is possible to employ an implicit, an explicit or a mixed implicit-explicit solution scheme for dynamic problems.

Using an implicit solution scheme, it is necessary to employ iterations and check convergence for every time step. Bathe [48] indicated that in nonlinear static analysis, iteration might not converge under certain loading conditions, whereas convergence is always achieved in dynamic analysis provided that the time step is sufficiently small. In addition, for a dynamic analysis the inertia of the system renders its dynamic response more smooth than its static one, and convergence is in general more rapid. The same observation was confirmed by Choudhry and Lee [41]. They indicated that in metal forming processes, like sheet-metal bending and frictionless deep-drawing, large rigid-body motions might be involved, leading to convergence problems. Contributions from the dynamic terms help to stabilize the computations. As an example for a sheet-metal bending process with large rotations and relatively small strains, a flat sheet of 1 mm thickness is deformed with a cylindrical punch into a channel. The sheet deformation is primarily due to bending in the regions of contact with punch and die. The maximum punch displacement attained in modelling the bending process quasi-statically was 35 mm. However, a displacement of 53 mm was attainable when the dynamic effects were considered. They concluded that it was important to retain dynamic effects in the simulation of this problem. This problem is discussed in more details in Chapter 6.

The solution of dynamic equations of motion using explicit time integration simply corresponds to a forward marching in time without iterations. Explicit integration schemes do not require a factorization of the global stiffness matrix. Moreover, if the global mass matrix is diagonal, no matrix factorization is required at all. In this case it is not necessary to assemble the global mass and stiffness matrices and the solution may be carried out on the element level and a little high-speed storage is achieved. The main shortcoming of this scheme is that it is conditionally stable. The time step Δt has to be smaller than a critical value Δt_{cr}. For typical sheet metal forming operations, a time step in the order of one millisecond is normally sufficient [10, 49]. This may impose a stringent limitation on the efficiency of this method for the simulation of slow metal forming processes which occur in several seconds.

Comparative investigations into implicit, explicit, and iterative implicit-explicit schemes for the simulation of sheet metal forming processes have been the topic for extensive recent research [49-55]. It is shown that the difficulties associated with the convergence of the implicit methods have led to a renewed interest in the use of dynamic explicit methods, even when the forming problem is essentially quasi-static [54]. These difficulties are more pronounced in the case of complex problems with large number of elements, large rigid body motions and variable contact conditions.

Although the implicit approach employs a more reliable and rigorous scheme in considering equilibrium at each step, convergence problems may be expected. If the explicit approach is used for all time steps, computation time for slow metal forming processes with fine meshes will become excessive. Thus, it might be more efficient to shift between the two techniques during the deformation process. Therefore, implicit, explicit and mixed implicit-explicit solution techniques will be implemented in the current work.

2.3 SUMMARY

Survey of the literature on ALE leads to the following conclusions:

- Almost all of the previous ALE analyses are based on the easy-to-implement operator split technique as opposed to the theoretically more accurate fully coupled approach.
- Most of the previous ALE formulations where developed in a rate form similar to the Eulerian formulation, with velocities as independent variables, and can not be easily related to the Lagrangian formulation normally used for solid mechanics applications.
- All dynamic ALE formulations available in the literature are based on the explicit solution scheme, which suffer from the lack of generality of application due to the stringent stability conditions and which usually necessitate the use of very small time steps.
- No attempt has been made to develop a fully coupled ALE formulation for both quasi-static and dynamic solid mechanics applications based on the more reliable implicit solution scheme and consistent with the standard Lagrangian formulation.

- The two common treatments of convective terms, namely the assumption of a continuous stress field and the use of the stress-velocity product, introduce some sort of approximations and limitations.

Chapter 3

DERIVATION OF ALE GOVERNING EQUATIONS

3.1 PRELIMINARIES

An implicit time-stepping approach will be used in deriving the governing ALE virtual work equilibrium equations from the basic principles of continuum mechanics [56, 57]. In this approach, we assume that the solutions for all time steps from time 0 to time t have been solved, and that the solution for time $t+\Delta t$ is required next. This process is applied repetitively until the complete solution path has been solved for. Linearized virtual work equations for quasi-static and dynamic applications are derived. A new method for handling convective terms in the equilibrium equations is presented [58].

3.1.1 Notations

Throughout the derivation, standard indicial notations are adopted; right subscripts denote the components of a tensor and repeated subscripts imply summation. In addition, time and configuration notations similar to those used by Bathe [4] are adopted. Left superscripts indicate the configuration in which the quantity occurs whereas left subscripts indicate the configuration to which the quantity is referred. Left subscripts may be omitted be if the quantity occurs in the same configuration in which it is measured. A quantity with no left superscripts or subscripts indicates an incremental quantity from time t to $t+\Delta t$.

3.1.2 Kinematics

In the ALE description, the material configuration at any time t refers to the set of material particles, whereas the reference configuration consists of a set of arbitrarily moving grid points sharing a common boundary with the set of material particles. The material configuration is identified by a set of material point coordinates X_i^m while the reference, or grid, configuration is identified by an independent set of grid point coordinates X_i^g. Let ${}^t x_i^m(X_j^m,t)$ and ${}^t x_i^g(X_j^g,t)$ be the vector functions or the mappings that characterize the motion of the material point X_j^m and the grid point X_j^g in space, respectively. The position of X_j^m at time t is given by

$$ {}^t x_i = {}^t x_i^m(X_j^m,t) \tag{3.1} $$

The set of material particles is related to the set of grid points by requiring that the two configurations share the same space at all times. Any point within the common boundary is occupied by elements of the two sets. Thus, the position of the grid point X_j^g that occupies the same point in space at time t as X_j^m is also given by ${}^t x_i$ as

$$ {}^t x_i = {}^t x_i^g(X_j^g,t) \tag{3.2} $$

The ALE formulation requires that the inverse of (3.1) and (3.2) exist to ensure a one-to-one mapping between the two configurations. The material velocity ${}^t v_i$ and the grid point velocity ${}^t v_i^g$ at time t are given by

$$ {}^t v_i = \left.\frac{\partial {}^t x_i^m}{\partial t}\right|_{X_j^m} \tag{3.3} $$

$$^{t}v_i^g = \left.\frac{\partial\, ^{t}x_i^g}{\partial t}\right|_{X_j^g} \tag{3.4}$$

The boundary constraint, which ensures that the material and grid configurations have the same boundary at all times, can be expressed as

$$(^{t}v_i - {}^{t}v_i^g)\, ^{t}n_i \Big|_{\text{on the boundary}} = 0 \tag{3.5}$$

where $^{t}n_i$ is the unit normal to the boundary surface.

The governing ALE equations involve the material time derivative of several quantities. The material derivative of an arbitrary function ^{t}f is denoted by a superposed dot and is defined to be the rate of change of the function holding the material particle X_i^m fixed

$$^{t}\dot{f} = \left.\frac{\partial\, ^{t}f}{\partial t}\right|_{X_i^m} \tag{3.6}$$

However, the grid configuration is the computational configuration that tracks the history of all quantities. Thus, it is convenient to define a grid time derivative, which is the time derivative of the function ^{t}f holding the grid point X_i^g fixed, and denote it by a superposed prime

$$^{t}f' = \left.\frac{\partial\, ^{t}f}{\partial t}\right|_{X_i^g} \tag{3.7}$$

The relation between the two time derivatives is given by [24]

$$^{t}\dot{f} = {}^{t}f' + (^{t}v_i - {}^{t}v_i^g)\frac{\partial\, ^{t}f}{\partial\, ^{t}x_i} \tag{3.8}$$

The ALE formulation will be discretized using the isoparametric displacement based finite element method. We denote the incremental material displacements from time t to time $t+\Delta t$ by u_i and the corresponding incremental grid displacements by u_i^g. We have the following relation

$$ {}^{t+\Delta t}x_i = {}^{t}x_i + u_i^g \tag{3.9}$$

where ${}^{t+\Delta t}x_i$ is the position of the grid point in the configuration at time $t+\Delta t$.

3.1.3 Continuity

The local form of conservation of mass, continuity, at time t is given by

$$ {}^{t}\dot{\rho} = -{}^{t}\rho \frac{\partial {}^{t}v_i}{\partial {}^{t}x_i} \tag{3.10}$$

where ${}^{t}\rho$ is the material density. Using (3.8), the continuity equation with respect to an arbitrary moving grid point can be expressed as

$$ {}^{t}\rho' = -{}^{t}\rho \frac{\partial {}^{t}v_i}{\partial {}^{t}x_i} - ({}^{t}v_i - {}^{t}v_i^g)\frac{\partial {}^{t}\rho}{\partial {}^{t}x_i} \tag{3.11}$$

3.2 QUASI-STATIC ANALYSIS

For quasi-static analysis, such as in the case of low-speed metal forming processes, inertia effects may be neglected. Employing an implicit incremental approach, the governing equilibrium equations for ALE are established for the configuration at time $t+\Delta t$. Since the configuration of the body at time $t+\Delta t$ is yet unknown, an approximate solution can be obtained by referring all variables to the grid configuration at time t and linearizing the resulting equations. The solution is then refined by iterations.

3.2.1 Principle of virtual displacements

Since the earliest ALE formulations were developed for fluid mechanics applications, the principle of virtual power, with velocities as independent variables, found wide appeal. ALE researchers [39, 59-63] continued to use the principle of virtual power for solid mechanics applications in which displacements are more natural to use. In this work, a procedure analogous to that used by Bathe [4] in obtaining the virtual work equations for the Updated Lagrangian formulation is chosen. Thus the principle of virtual displacements is employed to express the equilibrium of the body at time $t+\Delta t$. It can be written in the form

$$\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\sigma_{ij}\delta\, {}_{t+\Delta t}e_{ij}\, {}^{t+\Delta t}dV = \delta\, {}^{t+\Delta t}W^{ext} \tag{3.12}$$

where ${}^{t+\Delta t}\sigma_{ij}$ are the components of the Cauchy stress tensor at time $t+\Delta t$ and ${}_{t+\Delta t}e_{ij}$ is the conjugate strain tensor defined by

$${}_{t+\Delta t}e_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial\, {}^{t+\Delta t}x_j} + \frac{\partial u_j}{\partial\, {}^{t+\Delta t}x_i}\right) \tag{3.13}$$

The external virtual work, $\delta\, {}^{t+\Delta t}W^{ext}$, is given by

$$\delta\, {}^{t+\Delta t}W^{ext} = \int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\rho\, {}^{t+\Delta t}f_i^B \delta u_i\, {}^{t+\Delta t}dV + \int_{{}^{t+\Delta t}S} {}^{t+\Delta t}f_i^S \delta u_i\, {}^{t+\Delta t}dS \tag{3.14}$$

in which ${}^{t+\Delta t}f_i^B$ and ${}^{t+\Delta t}f_i^S$ are the components of the body force per unit mass and the surface traction at time $t+\Delta t$, respectively.

3.2.2 Incremental decompositions

In referring variables to the grid configuration, variables at time $t+\Delta t$ are assumed to be composed of their respective values at time t plus an increment given by the grid time derivative of the variable multiplied by the time increment Δt. This is in contrast to the Updated Lagrangian formulation where the incremental quantities are given by the material time derivative of the variable multiplied by the time increment Δt.

Material density at time $t+\Delta t$ can be decomposed into

$$^{t+\Delta t}\rho = {}^{t}\rho + {}^{t}\rho' \Delta t \tag{3.15}$$

which, upon substituting with (3.11), gives

$$^{t+\Delta t}\rho = {}^{t}\rho - {}^{t}\rho \frac{\partial u_k}{\partial {}^{t}x_k} - (u_k - u_k^g)\frac{\partial {}^{t}\rho}{\partial {}^{t}x_k} \tag{3.16}$$

Stress components at time $t+\Delta t$ can be expressed in terms of the stresses at time t for the same grid point plus a stress increment $\sigma'_{ij}\Delta t$

$$^{t+\Delta t}\sigma_{ij} = {}^{t}\sigma_{ij} + {}^{t}\sigma'_{ij}\Delta t \tag{3.17}$$

and using (3.8), we get

$$^{t+\Delta t}\sigma_{ij} = {}^{t}\sigma_{ij} + {}^{t}\dot{\sigma}_{ij}\Delta t - (u_k - u_k^g)\frac{\partial {}^{t}\sigma_{ij}}{\partial {}^{t}x_k} \tag{3.18}$$

The material rate of Cauchy stresses ${}^{t}\dot{\sigma}_{ij}$ is calculated from the material constitutive relation which is usually given in terms of an objective stress rate tensor such as the

Truesdell stress rate tensor defined by

$$ {}^{t}\sigma_{ij}^{T} = {}^{t}\dot{\sigma}_{ij} + \frac{\partial^{t} v_{k}}{\partial^{t} x_{k}} {}^{t}\sigma_{ij} - \frac{\partial^{t} v_{j}}{\partial^{t} x_{k}} {}^{t}\sigma_{ik} - \frac{\partial^{t} v_{i}}{\partial^{t} x_{k}} {}^{t}\sigma_{jk} \qquad (3.19)$$

The material constitutive relation in terms of the Truesdell stress rate is given by

$$ {}^{t}\sigma_{ij}^{T} = {}^{t}C_{ijkl} \, {}^{t}D_{kl} \qquad (3.20)$$

where ${}^{t}D_{ij}$ is the rate of deformation tensor given by

$$ {}^{t}D_{ij} = \frac{1}{2}\left(\frac{\partial^{t} v_{i}}{\partial^{t} x_{j}} + \frac{\partial^{t} v_{j}}{\partial^{t} x_{i}}\right) \qquad (3.21)$$

and ${}^{t}C_{ijkl}$ is the fourth order material constitutive tensor. It should be noted that the above stress and strain measures were chosen to allow the ALE formulation to be effective in large strain situations.

The variation in the strain components at time $t + \Delta t$ can be decomposed as

$$ \delta_{t+\Delta t} e_{ij} = \delta_{t} e_{ij} + \delta_{t} e'_{ij} \Delta t \qquad (3.22)$$

in which $\delta_{t} e'_{ij}$ is the grid time derivative of $\delta_{t} e_{ij}$ given by [Appendix A]

$$ \delta_{t} e'_{ij} = -\frac{1}{2}\left(\frac{\partial \delta u_{i}}{\partial^{t} x_{k}} \frac{\partial^{t} v_{k}^{g}}{\partial^{t} x_{j}} + \frac{\partial \delta u_{j}}{\partial^{t} x_{k}} \frac{\partial^{t} v_{k}^{g}}{\partial^{t} x_{i}}\right) \qquad (3.23)$$

Substitution in (3.22) gives

$$ \delta_{t+\Delta t} e_{ij} = \delta_{t} e_{ij} - \frac{1}{2}\left(\frac{\partial \delta u_{i}}{\partial^{t} x_{k}} \frac{\partial u_{k}^{g}}{\partial^{t} x_{j}} + \frac{\partial \delta u_{j}}{\partial^{t} x_{k}} \frac{\partial u_{k}^{g}}{\partial^{t} x_{i}}\right) \qquad (3.24)$$

Incremental decomposition of elemental volume at time $t + \Delta t$ in terms of the elemental volume at time t is given by [64]

$$ {}^{t+\Delta t}dV = {}^{t}dV + {}^{t}dV{}^{t}\Delta t = (1 + \frac{\partial u_k^g}{\partial {}^{t}x_k}){}^{t}dV \tag{3.25}$$

Similarly, incremental decomposition of elemental surface area is given by [64]

$$ {}^{t+\Delta t}dS = {}^{t}dS + {}^{t}dS{}^{t}\Delta t = [1 + \frac{\partial u_k^g}{\partial {}^{t}x_k} - \frac{1}{2}(\frac{\partial u_m^g}{\partial {}^{t}x_n} + \frac{\partial u_n^g}{\partial {}^{t}x_m}){}^{t}n_m{}^{t}n_n]{}^{t}dS \tag{3.26}$$

where ${}^{t}n_m$ is the unit outward normal to the surface at time t.

3.2.3 Linearization

Linearization is accomplished by expanding (3.12) using the previous incremental decompositions and neglecting higher orders in all incremental quantities. Substituting by (3.18), (3.24) and (3.25), the internal virtual work can be linearized as

$$\begin{aligned}\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\sigma_{ij}\delta\, {}_{t+\Delta t}e_{ij}\, {}^{t+\Delta t}dV &= \int_{{}^{t}V} {}^{t}\sigma_{ij}\delta\, {}_{t}e_{ij}\, {}^{t}dV + \int_{{}^{t}V} {}^{t}\dot{\sigma}_{ij}\Delta t\delta\, {}_{t}e_{ij}\, {}^{t}dV + \int_{{}^{t}V} \frac{\partial u_k^g}{\partial {}^{t}x_k}{}^{t}\sigma_{ij}\delta\, {}_{t}e_{ij}\, {}^{t}dV \\ &- \int_{{}^{t}V} \frac{\partial u_k^g}{\partial {}^{t}x_j}{}^{t}\sigma_{ij}\frac{\partial \delta u_i}{\partial {}^{t}x_k}{}^{t}dV - \int_{{}^{t}V}(u_k - u_k^g)\frac{\partial {}^{t}\sigma_{ij}}{\partial {}^{t}x_k}\delta\, {}_{t}e_{ij}\, {}^{t}dV\end{aligned} \tag{3.27}$$

Considering the external virtual work on the RHS of (3.12), the body force term can be referred to the grid configuration by using (3.16) and (3.25) to get

$$\begin{aligned}\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\rho\, {}^{t+\Delta t}f_i^B\delta u_i\, {}^{t+\Delta t}dV &= \int_{{}^{t}V} {}^{t}\rho\, {}^{t+\Delta t}f_i^B\delta u_i\, {}^{t}dV - \int_{{}^{t}V} {}^{t}\rho\, {}^{t+\Delta t}f_i^B(\frac{\partial u_k}{\partial {}^{t}x_k} - \frac{\partial u_k^g}{\partial {}^{t}x_k})\delta u_i\, {}^{t}dV \\ &- \int_{{}^{t}V} {}^{t+\Delta t}f_i^B(u_k - u_k^g)\frac{\partial {}^{t}\rho}{\partial {}^{t}x_k}\delta u_i\, {}^{t}dV\end{aligned} \tag{3.28}$$

Similarly, using (3.26), the traction force term of the external virtual work is expressed as

$$\int_{{}^{t+\Delta t}S} {}^{t+\Delta t}f_i^S\delta u_i\, {}^{t+\Delta t}dS = \int_{{}^{t}S} {}^{t+\Delta t}f_i^S[1 + \frac{\partial u_k^g}{\partial {}^{t}x_k} - \frac{1}{2}(\frac{\partial u_m^g}{\partial {}^{t}x_n} + \frac{\partial u_n^g}{\partial {}^{t}x_m}){}^{t}n_m{}^{t}n_n]\delta u_i\, {}^{t}dS \tag{3.29}$$

3.2.4 Treatment of convective terms

Convective terms, such as the last integral on the RHS of (3.27), involve the computation of the spatial derivatives of stresses. Since the stress values are computed at the integration points, and not at the nodal points, the stress field is generally discontinuous across element edges. Thus, stress gradients may not be reliably computed on the element level when evaluating element matrices. A method of finding a continuous stress field by interpolation was developed by Huétink [26] and used in the Eulerian step of the operator split technique. Some researchers used the same method to handle convective terms in the coupled equilibrium equations [39]. In this method, integration point stresses are first extrapolated by a least square approximation to get the nodal stresses. Nodal stresses computed from each element are then averaged. A continuous stress field is then assumed in the form

$$ {}^{t}\sigma_{ij} = \sum_{\alpha=1}^{N} h_{\alpha}\, {}^{t}\overline{\sigma}_{ij\alpha} \qquad (3.30)$$

where h_{α} is the element shape function evaluated at node α, $\overline{\sigma}_{ij\alpha}$ are the nodal stress components at node α and N is the number of element nodal points. Finally, the spatial derivatives of stresses can be computed on the element level as

$$\frac{\partial\, {}^{t}\sigma_{ij}}{\partial\, {}^{t}x_{k}} = \sum_{\alpha=1}^{N} \frac{\partial h_{\alpha}}{\partial\, {}^{t}x_{k}}\, {}^{t}\overline{\sigma}_{ij\alpha} \qquad (3.31)$$

This method is very popular in the ALE literature despite the least square approximation and assumption of a continuous stress field. Another method for treating convective terms was proposed by Liu et al. [33]. A stress-velocity product is defined in the form

$$ {}^{t}y_{ijk} = ({}^{t}v_k - {}^{t}v_k^g)\,{}^{t}\sigma_{ij} \tag{3.32} $$

Differentiating (3.32) with respect to space gives the convective term as

$$ ({}^{t}v_k - {}^{t}v_k^g)\frac{\partial\,{}^{t}\sigma_{ij}}{\partial\,{}^{t}x_k} = \frac{\partial\,{}^{t}y_{ijk}}{\partial\,{}^{t}x_k} - (\frac{\partial\,{}^{t}v_k}{\partial\,{}^{t}x_k} - \frac{\partial\,{}^{t}v_k^g}{\partial\,{}^{t}x_k})\,{}^{t}\sigma_{ij} \tag{3.33} $$

Equation (3.33) circumvents the computation of the stress gradients by computing the gradients of the stress-velocity product instead. However, this method also requires interpolation of the stress-velocity product using shape functions in a fashion similar to (3.30) and thus mixed finite elements, rather than the more common displacement based finite elements, must be used.

In this work, a new method for the treatment of the convective term that sidesteps the computation of the spatial gradients of stresses is used. Based on fundamental ALE relations, the new method offers an accurate treatment of convective terms while maintaining the convenience of using displacement based finite elements. This method involves a transformation from volume integrals to surface integrals as offered by the divergence theorem. Use is also made of the boundary constraint in (3.5). The last integral on the RHS of (3.27) can be rewritten as

$$ \begin{aligned} \int_{{}^{t}V}(u_k - u_k^g)\frac{\partial\,{}^{t}\sigma_{ij}}{\partial\,{}^{t}x_k}\delta_{t}e_{ij}\,{}^{t}dV &= \int_{{}^{t}V}\frac{\partial[(u_k - u_k^g)\,{}^{t}\sigma_{ij}\delta_{t}e_{ij}]_{t}}{\partial\,{}^{t}x_k}\,{}^{t}dV - \int_{{}^{t}V}(u_k - u_k^g)\,{}^{t}\sigma_{ij}\frac{\partial\delta_{t}e_{ij}}{\partial\,{}^{t}x_k}\,{}^{t}dV \\ &- \int_{{}^{t}V}(\frac{\partial u_k}{\partial\,{}^{t}x_k} - \frac{\partial u_k^g}{\partial\,{}^{t}x_k})\,{}^{t}\sigma_{ij}\delta_{t}e_{ij}\,{}^{t}dV \end{aligned} \tag{3.34} $$

No new assumptions are introduced in the calculation of the second and third integrals on the RHS of (3.34) since the spatial derivatives of displacements and strains can be computed on the element level from shape functions derivatives. Applying the divergence theorem to the first integral on the RHS of (3.34) and using (3.5), we get

$$\int_{^tV}\frac{\partial[(u_k-u_k^g)^t\sigma_{ij}\delta_t e_{ij}]_t}{\partial^t x_k}{}^t dV=\int_{^tS}(u_k-u_k^g)^t\sigma_{ij}\delta_t e_{ij}{}^t n_k{}^t dS=0 \tag{3.35}$$

Substituting in (3.27), the internal virtual work becomes

$$\begin{aligned}\int_{^{t+\Delta t}V}{}^{t+\Delta t}\sigma_{ij}\delta_{t+\Delta t}e_{ij}{}^{t+\Delta t}dV &= \int_{^tV}{}^t\sigma_{ij}\delta_t e_{ij}{}^t dV+\int_{^tV}{}^t\dot{\sigma}_{ij}\Delta t\delta_t e_{ij}{}^t dV+\int_{^tV}{}^t\sigma_{ij}\delta_t e_{ij}\frac{\partial u_k}{\partial^t x_k}{}^t dV\\ &-\int_{^tV}\frac{\partial u_k^g}{\partial^t x_j}{}^t\sigma_{ij}\frac{\partial\delta u_i}{\partial^t x_k}{}^t dV+\int_{^tV}(u_k-u_k^g)^t\sigma_{ij}\frac{\partial\delta_t e_{ij}}{\partial^t x_k}{}^t dV\end{aligned} \tag{3.36}$$

The same method can be applied to the convective body force term to avoid the computation of the spatial derivatives of density. The last integral on the RHS of (3.28) can be treated in the same manner as in (3.34) and (3.35), to give

$$\begin{aligned}\int_{^{t+\Delta t}V}{}^{t+\Delta t}\rho^{t+\Delta t}f_i^B\delta u_i{}^{t+\Delta t}dV &= \int_{^tV}{}^t\rho^{t+\Delta t}f_i^B\delta u_i{}^t dV+\int_{^tV}{}^t\rho\frac{\partial^{t+\Delta t}f_i^B}{\partial^t x_k}(u_k-u_k^g)\delta u_i{}^t dV\\ &+\int_{^tV}{}^t\rho^{t+\Delta t}f_i^B(u_k-u_k^g)\frac{\partial\delta u_i}{\partial^t x_k}{}^t dV\end{aligned} \tag{3.37}$$

3.2.5 Fully coupled ALE equilibrium equation

Substituting (3.36), (3.37) and (3.29) into (3.12), the principle of virtual displacements at time $t+\Delta t$ referred to time t can be written as

$$\begin{aligned}&\int_{^tV}{}^t\dot{\sigma}_{ij}\Delta t\delta_t e_{ij}{}^t dV+\int_{^tV}{}^t\sigma_{ij}\delta_t e_{ij}\frac{\partial u_k}{\partial^t x_k}{}^t dV\\ &+\int_{^tV}(u_k-u_k^g)^t\sigma_{ij}\frac{\partial\delta_t e_{ij}}{\partial^t x_k}{}^t dV-\int_{^tV}\frac{\partial u_k^g}{\partial^t x_j}{}^t\sigma_{ij}\frac{\partial\delta u_i}{\partial^t x_k}{}^t dV\\ &=\delta^{t+\Delta t}W^{ext}-\int_{^tV}{}^t\sigma_{ij}\delta_t e_{ij}{}^t dV\end{aligned} \tag{3.38}$$

where

$$\delta^{t+\Delta t}W^{ext} = \int_{{}^tV} {}^t\rho[{}^{t+\Delta t}f_i^B + \frac{\partial^{t+\Delta t}f_i^B}{\partial^t x_k}(u_k - u_k^g)]\delta u_i\,{}^t dV + \int_{{}^tV} {}^t\rho^{t+\Delta t}f_i^B(u_k - u_k^g)\frac{\partial \delta u_i}{\partial^t x_k}{}^t dV$$

$$+\int_{{}^tS} {}^{t+\Delta t}f_i^S[1 + \frac{\partial u_k^g}{\partial^t x_k} - \frac{1}{2}(\frac{\partial u_m^g}{\partial^t x_n} + \frac{\partial u_n^g}{\partial^t x_m}){}^t n_m\,{}^t n_n]\delta u_i\,{}^t dS \qquad (3.39)$$

The constitutive relations in (3.19) to (3.21) can now be introduced into the first integral in (3.38) to give

$$\int_{{}^tV} {}^tC_{ijkl\,t}e_{kl}\delta\,{}_te_{ij}\,{}^t dV + \int_{{}^tV} {}^t\sigma_{ij}\delta\,{}_t\eta_{ij}\,{}^t dV$$

$$+\int_{{}^tV}(u_k - u_k^g){}^t\sigma_{ij}\frac{\partial \delta\,{}_te_{ij}}{\partial^t x_k}{}^t dV + \int_{{}^tV}(\frac{\partial u_k}{\partial^t x_j} - \frac{\partial u_k^g}{\partial^t x_j}){}^t\sigma_{ij}\frac{\partial \delta u_i}{\partial^t x_k}{}^t dV$$

$$= \delta^{t+\Delta t}W^{ext} - \int_{{}^tV} {}^t\sigma_{ij}\delta\,{}_te_{ij}\,{}^t dV \qquad (3.40)$$

where

$${}_t\eta_{ij} = \frac{1}{2}\frac{\partial u_k}{\partial^t x_i}\frac{\partial u_k}{\partial^t x_j} \qquad (3.41)$$

Equation (3.40) is linear in the incremental displacements u_i and u_i^g. The first two integrals on the LHS of (3.40) correspond to the Lagrangian material and geometric stiffness matrices and are exactly the same as those obtained using an Updated Lagrangian formulation. The last two integrals on the LHS establish the convective stiffness matrices due to ALE. The last term on the RHS of (3.40) corresponds to the Lagrangian internal force vector.

Equation (3.40) represents the fully coupled ALE equilibrium equation. This equation can reduce to the Updated Lagrangian formulation if we choose to attach the grid to the material, i.e. $u_i^g = u_i$, and to the Eulerian formulation if we choose to fix the grid in space, i.e. $u_i^g = 0$, as limiting cases. The ALE equilibrium equation derived in this work shows that ALE can be considered as a logical extension to the Lagrangian formulation and the modifications to the equilibrium equation of current Updated Lagrangian codes are clearly identified.

3.3 DYNAMIC ANALYSIS

In dynamic analyses, inertia effects are included in the balance of momentum at time $t+\Delta t$. Inertia forces involve the material time derivative of material velocities, i.e. material accelerations ${}^{t+\Delta t}\dot{v}_i$. In the ALE formulation, we follow the grid point in its motion as our reference configuration. Therefore, the referential material acceleration, which is the grid time derivative of the material velocity ${}^{t+\Delta t}v_i'$, should be used. For clarity, we will denote the referential material acceleration ${}^{t+\Delta t}v_i'$ by ${}^{t+\Delta t}a_i$.

3.3.1 Virtual work done by inertia forces

Using the relation between the two time derivatives in (3.8), the virtual work done by inertia forces can be expanded as

$$\begin{aligned}\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\rho\, {}^{t+\Delta t}\dot{v}_i \delta u_i\, {}^{t+\Delta t}dV &= \int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\rho\, {}^{t+\Delta t}a_i \delta u_i\, {}^{t+\Delta t}dV \\ &+ \int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\rho ({}^{t+\Delta t}v_j - {}^{t+\Delta t}v_j^g) \frac{\partial^{t+\Delta t} v_i}{\partial^{t+\Delta t} x_j} \delta u_i\, {}^{t+\Delta t}dV \end{aligned} \tag{3.42}$$

Equation (3.42) is considered as an extra virtual work term due to inertia effects to be added to the LHS, or subtracted from the RHS, of the ALE virtual work equation, (3.40).

The first term on the RHS of (3.42) can be referred to as the referential inertia term whereas the second term is referred to as the convective inertia term.

3.3.2 Decomposition of velocities and accelerations

The velocities and accelerations at time $t+\Delta t$ can be related to their respective values at time t using the relations

$$ {}^{t+\Delta t}a_i = {}^{t}a_i + a_i \qquad (3.43)$$

$$ {}^{t+\Delta t}v_i = {}^{t}v_i + v_i \qquad (3.44)$$

$$ {}^{t+\Delta t}v_i^g = {}^{t}v_i^g + v_i^g \qquad (3.45)$$

The incremental quantities a_i, v_i and v_i^g depend on the implicit time integration scheme to be employed. To maintain a linear increment, higher orders in a_i, v_i, v_i^g and Δt will be neglected.

3.3.3 Linearization of the referential inertia term

Incremental decomposition of the variables in the referential inertia term, in a manner similar to quasi-static analysis, and linearization of the result gives

$$\begin{aligned} \int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\rho\, {}^{t+\Delta t}a_i \delta u_i \, {}^{t+\Delta t}dV &= \int_{{}^{t}V} {}^{t}\rho\, {}^{t+\Delta t}a_i \delta u_i \, {}^{t}dV - \int_{{}^{t}V} {}^{t}\rho \left(\frac{\partial^t v_k}{\partial^t x_k} - \frac{\partial^t v_k^g}{\partial^t x_k}\right) {}^{t+\Delta t}a_i \delta u_i \Delta t \, {}^{t}dV \\ &\quad - \int_{{}^{t}V} \frac{\partial^t \rho}{\partial^t x_k} ({}^{t}v_k - {}^{t}v_k^g)\, {}^{t+\Delta t}a_i \delta u_i \Delta t \, {}^{t}dV \end{aligned} \qquad (3.46)$$

The last integral on the RHS of (3.46) can be rewritten as

$$\int_{{}^tV} \frac{\partial^t \rho}{\partial^t x}({}^tv_k - {}^tv_k^g)^{t+\Delta t}a_i \delta u_i \Delta t\, {}^t dV = \int_{{}^tV} \frac{\partial[{}^t\rho({}^tv_k - {}^tv_k^g)^{t+\Delta t}a_i \delta u_i]}{\partial^t x_k} \Delta t\, {}^t dV$$
$$- \int_{{}^tV} {}^t\rho\left(\frac{\partial^t v_k}{\partial^t x_k} - \frac{\partial^t v_k^g}{\partial^t x_k}\right)^{t+\Delta t}a_i \delta u_i \Delta t\, {}^t dV$$
$$- \int_{{}^tV} {}^t\rho({}^tv_k - {}^tv_k^g)\frac{\partial({}^{t+\Delta t}a_i \delta u_i)}{\partial^t x_k} \Delta t\, {}^t dV \qquad (3.47)$$

Applying the divergence theorem to the first integral above and using the boundary constraint in (3.5), we get

$$\int_{{}^tV} \frac{\partial[{}^t\rho({}^tv_k - {}^tv_k^g)^{t+\Delta t}a_i \delta u_i]}{\partial^t x_k} \Delta t\, {}^t dV = \int_{{}^tS} {}^t\rho({}^tv_k - {}^tv_k^g)^{t+\Delta t}a_i \delta u_i n_k \Delta t\, {}^t dV = 0 \qquad (3.48)$$

Substituting in (3.46), we get

$$\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\rho\, {}^{t+\Delta t}a_i \delta u_i\, {}^{t+\Delta t}dV = \int_{{}^tV} {}^t\rho\, {}^{t+\Delta t}a_i \delta u_i\, {}^t dV + \int_{{}^tV} {}^t\rho({}^tv_k - {}^tv_k^g)\frac{\partial({}^{t+\Delta t}a_i \delta u_i)}{\partial^t x_k} \Delta t\, {}^t dV \qquad (3.49)$$

Using (3.43), the referential inertia term can be written as

$$\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\rho\, {}^{t+\Delta t}a_i \delta u_i\, {}^{t+\Delta t}dV = \int_{{}^tV} {}^t\rho\, {}^t a_i \delta u_i\, {}^t dV + \int_{{}^tV} {}^t\rho a_i \delta u_i\, {}^t dV$$
$$+ \int_{{}^tV} {}^t\rho({}^tv_k - {}^tv_k^g)\frac{\partial({}^t a_i \delta u_i)}{\partial^t x_k} \Delta t\, {}^t dV \qquad (3.50)$$

3.3.4 Linearization of the convective inertia term

The convective inertia term can be expanded into

$$
\begin{aligned}
&\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\rho({}^{t+\Delta t}v_j - {}^{t+\Delta t}v_j^g)\frac{\partial^{t+\Delta t}v_i}{\partial^{t+\Delta t}x_j}\delta u_i\,{}^{t+\Delta t}dV \\
&= \int_{{}^{t}V} {}^{t}\rho({}^{t+\Delta t}v_j - {}^{t+\Delta t}v_j^g)\frac{\partial^{t+\Delta t}v_i}{\partial^{t}x_j}\delta u_i\,{}^{t}dV \\
&- \int_{{}^{t}V} {}^{t}\rho(\frac{\partial^{t}v_k}{\partial^{t}x_k} - \frac{\partial^{t}v_k^g}{\partial^{t}x_k})({}^{t+\Delta t}v_j - {}^{t+\Delta t}v_j^g)\frac{\partial^{t+\Delta t}v_i}{\partial^{t}x_j}\delta u_i\Delta t\,{}^{t}dV \\
&- \int_{{}^{t}V} \frac{\partial^{t}\rho}{\partial^{t}x_k}({}^{t}v_k - {}^{t}v_k^g)({}^{t+\Delta t}v_j - {}^{t+\Delta t}v_j^g)\frac{\partial^{t+\Delta t}v_i}{\partial^{t}x_j}\delta u_i\Delta t\,{}^{t}dV \\
&- \int_{{}^{t}V} {}^{t}\rho\frac{\partial^{t}v_k^g}{\partial^{t}x_j}({}^{t+\Delta t}v_j - {}^{t+\Delta t}v_j^g)\frac{\partial^{t+\Delta t}v_i}{\partial^{t}x_k}\delta u_i\Delta t\,{}^{t}dV
\end{aligned}
\tag{3.51}
$$

As before the third integral above, which involves the spatial gradients of density, can be treated using the divergence theorem and the boundary constraint, to get

$$
\begin{aligned}
&\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\rho({}^{t+\Delta t}v_j - {}^{t+\Delta t}v_j^g)\frac{\partial^{t+\Delta t}v_i}{\partial^{t+\Delta t}x_j}\delta u_i\,{}^{t+\Delta t}dV \\
&= \int_{{}^{t}V} {}^{t}\rho({}^{t+\Delta t}v_j - {}^{t+\Delta t}v_j^g)\frac{\partial^{t+\Delta t}v_i}{\partial^{t}x_j}\delta u_i\,{}^{t}dV \\
&+ \int_{{}^{t}V} {}^{t}\rho({}^{t}v_k - {}^{t}v_k^g)(\frac{\partial^{t+\Delta t}v_j}{\partial^{t}x_k} - \frac{\partial^{t+\Delta t}v_j^g}{\partial^{t}x_k})\frac{\partial^{t+\Delta t}v_i}{\partial^{t}x_j}\delta u_i\Delta t\,{}^{t}dV \\
&+ \int_{{}^{t}V} {}^{t}\rho({}^{t}v_k - {}^{t}v_k^g)({}^{t+\Delta t}v_j - {}^{t+\Delta t}v_j^g)\frac{\partial}{\partial^{t}x_k}(\frac{\partial^{t+\Delta t}v_i}{\partial^{t}x_j}\delta u_i)\Delta t\,{}^{t}dV \\
&- \int_{{}^{t}V} {}^{t}\rho\frac{\partial^{t}v_k^g}{\partial^{t}x_j}({}^{t+\Delta t}v_j - {}^{t+\Delta t}v_j^g)\frac{\partial^{t+\Delta t}v_i}{\partial^{t}x_k}\delta u_i\Delta t\,{}^{t}dV
\end{aligned}
\tag{3.52}
$$

Using (3.44) and (3.45), the convective inertia term can be written as

$$\begin{aligned}
&\int_{^{t+\Delta t}V} {}^{t+\Delta t}\rho({}^{t+\Delta t}v_j - {}^{t+\Delta t}v_j^g)\frac{\partial^{t+\Delta t}v_i}{\partial^{t+\Delta t}x_j}\delta u_i {}^{t+\Delta t}dV \\
&= \int_{^tV} {}^t\rho({}^tv_j - {}^tv_j^g)\frac{\partial^t v_i}{\partial^t x_j}\delta u_i {}^tdV \\
&+ \int_{^tV} {}^t\rho({}^tv_j - {}^tv_j^g)\frac{\partial v_i}{\partial^t x_j}\delta u_i {}^tdV \\
&+ \int_{^tV} {}^t\rho(v_j - v_j^g)\frac{\partial^t v_i}{\partial^t x_j}\delta u_i {}^tdV \\
&+ \int_{^tV} {}^t\rho({}^tv_k - {}^tv_k^g)(\frac{\partial^t v_j}{\partial^t x_k} - 2\frac{\partial^t v_j^g}{\partial^t x_k})\frac{\partial^t v_i}{\partial^t x_j}\delta u_i \Delta t {}^tdV \\
&+ \int_{^tV} {}^t\rho({}^tv_k - {}^tv_k^g)({}^tv_j - {}^tv_j^g)\frac{\partial}{\partial^t x_k}(\frac{\partial^t v_i}{\partial^t x_j}\delta u_i)\Delta t {}^tdV
\end{aligned} \tag{3.53}$$

3.3.5 Fully coupled ALE equation of motion

Combining (3.40), (3.50) and (3.53), the ALE equation of motion can be written as

$$\begin{aligned}
&+ \int_{^tV} {}^t\rho a_i \delta u_i {}^tdV + \int_{^tV} {}^tC_{ijkl\,t}e_{kl}\delta {}_te_{ij} {}^tdV + \int_{^tV} {}^t\sigma_{ij}\delta {}_t\eta_{ij} {}^tdV \\
&+ \int_{^tV} {}^t\rho({}^tv_j - {}^tv_j^g)\frac{\partial v_i}{\partial^t x_j}\delta u_i {}^tdV + \int_{^tV} {}^t\rho(v_j - v_j^g)\frac{\partial^t v_i}{\partial^t x_j}\delta u_i {}^tdV \\
&+ \int_{^tV} (u_k - u_k^g){}^t\sigma_{ij}\frac{\partial \delta {}_te_{ij}}{\partial^t x_k}{}^tdV + \int_{^tV} (\frac{\partial u_k}{\partial^t x_j} - \frac{\partial u_k^g}{\partial^t x_j}){}^t\sigma_{ij}\frac{\partial \delta u_i}{\partial^t x_k}{}^tdV \\
&= \delta^{t+\Delta t}W^{ext} - \int_{^tV} {}^t\sigma_{ij}\delta {}_te_{ij} {}^tdV - \int_{^tV} {}^t\rho {}^ta_i \delta u_i {}^tdV \\
&- \int_{^tV} {}^t\rho({}^tv_k - {}^tv_k^g)\frac{\partial({}^ta_i\delta u_i)}{\partial^t x_k}\Delta t {}^tdV - \int_{^tV} {}^t\rho({}^tv_j - {}^tv_j^g)\frac{\partial^t v_i}{\partial^t x_j}\delta u_i {}^tdV \\
&- \int_{^tV} {}^t\rho({}^tv_k - {}^tv_k^g)(\frac{\partial^t v_j}{\partial^t x_k} - 2\frac{\partial^t v_j^g}{\partial^t x_k})\frac{\partial^t v_i}{\partial^t x_j}\delta u_i \Delta t {}^tdV \\
&- \int_{^tV} {}^t\rho({}^tv_k - {}^tv_k^g)({}^tv_j - {}^tv_j^g)\frac{\partial}{\partial^t x_k}(\frac{\partial^t v_i}{\partial^t x_j}\delta u_i)\Delta t {}^tdV
\end{aligned} \tag{3.54}$$

Equation (3.54) is a linear equation in the incremental displacements u_i and u_i^g, incremental velocities v_i and v_i^g, and incremental accelerations a_i and it represents the fully coupled ALE equation of motion. The first three terms on both the LHS and RHS of (3.54) are exactly the same as in the Updated Lagrangian formulation. The first term on the LHS corresponds to the Lagrangian mass matrix. The second, third, sixth and seventh terms on the LHS and the first and second on the RHS were defined for quasi-static analysis. The fourth and fifth terms give rise to convective velocity stiffness matrices due to ALE. The third term on the RHS of (3.54) corresponds to the Lagrangian inertia force vector whereas the last four terms on the RHS are convective inertia force vectors due to ALE.

The fully coupled ALE equation of motion derived in this work is consistent with the standard Lagrangian formulation. No assumptions or approximations, other than those for the purposes of linearization, have been made in the derivation. Incremental velocities and accelerations are retained in the equations for later time integration.

Chapter 4

FINITE ELEMENT DISCRETIZATION

4.1 ISOPARAMETRIC FINITE ELEMENTS

In this chapter, the virtual work equations for both quasi-static analysis and dynamic analysis will be discretized using two-dimensional isoparametric finite elements. Details of computer implementation of the finite element matrix equations are also discussed [65].

In the isoparametric finite element discretization, element coordinates ${}^{t}x_i$ and incremental displacements u_i and u_i^g are interpolated using

$$ {}^{t}x_i = \sum_{k=1}^{N} h_k \, {}^{t}\overline{x}_{ik} \tag{4.1}$$

$$ u_i = \sum_{k=1}^{N} h_k \overline{u}_{ik} \tag{4.2}$$

$$ u_i^g = \sum_{k=1}^{N} h_k \overline{u}_{ik}^g \tag{4.3}$$

where ${}^{t}\overline{x}_{ik}$, $\overline{u}_{ik}$ and $\overline{u}_{ik}^g$ are the nodal coordinates and incremental material and grid displacements of degree of freedom *i* of nodal point *k* at time *t*, h_k is the element shape function corresponding to nodal point *k*, and *N* is the number of element nodal points. In two-dimensions, equation (4.1) can be expanded as

$$ {}^{t}x = \sum_{k=1}^{N} h_k\,{}^{t}\bar{x}_k \;,\; {}^{t}y = \sum_{k=1}^{N} h_k\,{}^{t}\bar{y}_k \tag{4.4}$$

In matrix form

$$\begin{Bmatrix} {}^{t}x \\ {}^{t}y \end{Bmatrix} = \left[\begin{array}{c|cc|c} \cdots & h_k & 0 & \cdots \\ & 0 & h_k & \end{array}\right] \left\{\begin{array}{c} \vdots \\ \hline {}^{t}\bar{x}_k \\ {}^{t}\bar{y}_k \\ \hline \vdots \end{array}\right\} \tag{4.5}$$

or

$$ {}^{t}\mathbf{x} = \mathbf{H}\,{}^{t}\bar{\mathbf{x}} \tag{4.6}$$

where ${}^{t}\mathbf{x}$ is the element coordinate vector given by

$$ {}^{t}\mathbf{x} = \begin{Bmatrix} {}^{t}x \\ {}^{t}y \end{Bmatrix} \tag{4.7}$$

$\mathbf{H}$ is the element interpolation matrix in the form

$$ \mathbf{H} = \left[\begin{array}{c|cc|c} \cdots & h_k & 0 & \cdots \\ & 0 & h_k & \end{array}\right]_{2\times 2N} \tag{4.8}$$

and ${}^{t}\bar{\mathbf{x}}$ is the nodal coordinate vector at time t given by

$$ {}^{t}\bar{\mathbf{x}} = \left\{\begin{array}{c|cc|c} \cdots & {}^{t}\bar{x}_k & {}^{t}\bar{y}_k & \cdots \end{array}\right\}^{T}_{1\times 2N} \tag{4.9}$$

Similarly, equations (4.2) and (4.3) can be written in the form

$$ \mathbf{u} = \mathbf{H}\bar{\mathbf{u}} \tag{4.10}$$

$$ \mathbf{u}^{g} = \mathbf{H}\bar{\mathbf{u}}^{g} \tag{4.11}$$

where $\mathbf{u}$, $\mathbf{u}^{g}$, $\bar{\mathbf{u}}$ and $\bar{\mathbf{u}}^{g}$ are the incremental element and nodal material and grid displacement vectors, respectively. We also have

$$ \delta\mathbf{u} = \mathbf{H}\delta\bar{\mathbf{u}} \tag{4.12}$$

where $\delta\bar{\mathbf{u}}$ is the virtual nodal displacement vector.

4.2 DISCRETIZATION OF THE QUASI-STATIC ALE EQUATION

The ALE virtual work equilibrium equation for quasi-static analysis was derived in Chapter 3 in the form

$$\begin{aligned}
&\int_{{}^tV} {}^tC_{ijkl\,t}e_{kl}\delta_t e_{ij}\,{}^tdV + \int_{{}^tV} {}^t\sigma_{ij}\delta_t\eta_{ij}\,{}^tdV \\
&+\int_{{}^tV}(u_k - u_k^g)\,{}^t\sigma_{ij}\frac{\partial\delta_t e_{ij}}{\partial {}^tx_k}\,{}^tdV + \int_{{}^tV}\left(\frac{\partial u_k}{\partial {}^tx_j} - \frac{\partial u_k^g}{\partial {}^tx_j}\right){}^t\sigma_{ij}\frac{\partial\delta u_i}{\partial {}^tx_k}\,{}^tdV \\
&=\delta^{t+\Delta t}W^{ext} - \int_{{}^tV} {}^t\sigma_{ij}\delta_t e_{ij}\,{}^tdV
\end{aligned} \tag{4.13}$$

As indicated before, the first and second terms on the LHS of (4.13) correspond to the Lagrangian material and geometric stiffness matrices, respectively, whereas the third and fourth terms establish the convective stiffness matrices due to ALE. The last term on the RHS of (4.13) corresponds to the Lagrangian internal force vector.

4.2.1 Discretization of the Lagrangian internal force term

The internal force vector obtained in ALE is the same as in a Lagrangian analysis. Thus it will be discretized exactly in the same manner [48]. We start by defining a stress vector in the form

$${}^t\boldsymbol{\sigma} = \begin{Bmatrix} {}^t\sigma_{xx} \\ {}^t\sigma_{yy} \\ {}^t\sigma_{xy} \\ {}^t\sigma_{zz} \end{Bmatrix} \tag{4.14}$$

We also define the incremental strain vector in the form

$$
{}_t\mathbf{e} = \begin{Bmatrix} {}_te_{xx} \\ {}_te_{yy} \\ 2{}_te_{xy} \\ {}_te_{zz} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u_x}{\partial^t x} \\ \dfrac{\partial u_y}{\partial^t y} \\ \dfrac{\partial u_x}{\partial^t y} + \dfrac{\partial u_y}{\partial^t x} \\ \dfrac{u_x}{{}^t x} \end{Bmatrix} \tag{4.15}
$$

where σ_{zz} and e_{zz} are the hoop stress and incremental strain in the case of axisymmetric problems. Using (4.2)

$$
{}_t\mathbf{e} = \begin{Bmatrix} \sum\limits_{k=1}^{N} \dfrac{\partial h_k}{\partial^t x}\bar{u}_{xk} \\ \sum\limits_{k=1}^{N} \dfrac{\partial h_k}{\partial^t y}\bar{u}_{yk} \\ \sum\limits_{k=1}^{N} (\dfrac{\partial h_k}{\partial^t y}\bar{u}_{xk} + \dfrac{\partial h_k}{\partial^t x}\bar{u}_{yk}) \\ \sum\limits_{k=1}^{N} \dfrac{h_k}{\sum\limits_{j=1}^{N} h_j {}^t\bar{x}_j}\bar{u}_{xk} \end{Bmatrix} = \left[\begin{array}{c|cc|c} & \dfrac{\partial h_k}{\partial^t x} & 0 & \\ & 0 & \dfrac{\partial h_k}{\partial^t y} & \\ \cdots & \dfrac{\partial h_k}{\partial^t y} & \dfrac{\partial h_k}{\partial^t x} & \cdots \\ & \dfrac{h_k}{\sum\limits_{j=1}^{N} h_j {}^t\bar{x}_j} & 0 & \end{array}\right] \begin{Bmatrix} \vdots \\ \bar{u}_{xk} \\ \bar{u}_{yk} \\ \vdots \end{Bmatrix} \tag{4.16}
$$

or

$$
{}_t\mathbf{e} = {}_t\mathbf{B}^{L1}\bar{\mathbf{u}} \tag{4.17}
$$

where

$$
{}_t\mathbf{B}^{L1} = \left[\begin{array}{c|cc|c} & \dfrac{\partial h_k}{\partial^t x} & 0 & \\ & 0 & \dfrac{\partial h_k}{\partial^t y} & \\ \cdots & \dfrac{\partial h_k}{\partial^t y} & \dfrac{\partial h_k}{\partial^t x} & \cdots \\ & \dfrac{h_k}{\sum\limits_{j=1}^{N} h_j {}^t\bar{x}_j} & 0 & \end{array}\right]_{4\times 2N} \tag{4.18}
$$

We also have

$$\delta_t\mathbf{e} = {}_t\mathbf{B}^{L1}\delta\bar{\mathbf{u}} \tag{4.19}$$

Using (4.14) and (4.19), the internal force term can thus be written as

$$\begin{aligned}\int_{{}^tV} {}^t\sigma_{ij}\delta_t e_{ij}\,{}^t dV &= \int_{{}^tV} (\delta_t\mathbf{e})^{T\,t}\boldsymbol{\sigma}\,{}^t dV \\ &= \int_{{}^tV} ({}_t\mathbf{B}^{L1}\delta\bar{\mathbf{u}})^{T\,t}\boldsymbol{\sigma}\,{}^t dV \\ &= (\delta\bar{\mathbf{u}})^T \int_{{}^tV} ({}_t\mathbf{B}^{L1})^{T\,t}\boldsymbol{\sigma}\,{}^t dV \\ &= (\delta\bar{\mathbf{u}})^{T\,t}\mathbf{f}\end{aligned} \tag{4.20}$$

where ${}^t\mathbf{f}$ is the internal force vector given by

$${}^t\mathbf{f} = \int_{{}^tV} ({}_t\mathbf{B}^{L1})^{T\,t}\boldsymbol{\sigma}\,{}^t dV \tag{4.21}$$

4.2.2 Discretization of the Lagrangian material stiffness term

The Lagrangian material stiffness virtual work term can be rewritten in matrix form as

$$\begin{aligned}\int_{{}^tV} {}^tC_{ijkl\,t}e_{kl}\delta_t e_{ij}\,{}^t dV &= \int_{{}^tV} (\delta_t\mathbf{e})^{T\,t}\mathbf{C}\,{}_t\mathbf{e}\,{}^t dV \\ &= \int_{{}^tV} ({}_t\mathbf{B}^{L1}\delta\bar{\mathbf{u}})^{T\,t}\mathbf{C}({}_t\mathbf{B}^{L1}\bar{\mathbf{u}})\,{}^t dV \\ &= (\delta\bar{\mathbf{u}})^T \int_{{}^tV} ({}_t\mathbf{B}^{L1})^{T\,t}\mathbf{C}\,{}_t\mathbf{B}^{L1\,t} dV\bar{\mathbf{u}} \\ &= (\delta\bar{\mathbf{u}})^{T\,t}\mathbf{K}^{L1}\bar{\mathbf{u}}\end{aligned} \tag{4.22}$$

where ${}^t\mathbf{C}$ is the elastic-plastic material constitutive matrix and ${}^t\mathbf{K}^{L1}$ is the incremental material stiffness matrix given by [48]

$${}^t\mathbf{K}^{L1} = \int_{{}^tV} ({}_t\mathbf{B}^{L1})^{T\,t}\mathbf{C}\,{}_t\mathbf{B}^{L1\,t} dV \tag{4.23}$$

4.2.3 Discretization of the Lagrangian geometric stiffness term

The Lagrangian geometric stiffness virtual work term can be rewritten as

$$\int_{{}^tV} {}^t\sigma_{ij}\delta_t\eta_{ij}\,{}^tdV = \int_{{}^tV} {}^t\sigma_{ij}\frac{\partial u_k}{\partial {}^tx_j}\frac{\partial \delta u_k}{\partial {}^tx_i}\,{}^tdV \tag{4.24}$$

By expanding the summations over *i*, *j* and *k*, we get

$$\begin{aligned}\int_{{}^tV} {}^t\sigma_{ij}\delta_t\eta_{ij}\,{}^tdV = \int_{{}^tV} (& {}^t\sigma_{xx}\frac{\partial u_x}{\partial {}^tx}\frac{\partial \delta u_x}{\partial {}^tx} + {}^t\sigma_{xy}\frac{\partial u_x}{\partial {}^ty}\frac{\partial \delta u_x}{\partial {}^tx} + {}^t\sigma_{yx}\frac{\partial u_x}{\partial {}^tx}\frac{\partial \delta u_x}{\partial {}^ty} + {}^t\sigma_{yy}\frac{\partial u_x}{\partial {}^ty}\frac{\partial \delta u_x}{\partial {}^ty} \\ & + {}^t\sigma_{xx}\frac{\partial u_y}{\partial {}^tx}\frac{\partial \delta u_y}{\partial {}^tx} + {}^t\sigma_{xy}\frac{\partial u_y}{\partial {}^ty}\frac{\partial \delta u_y}{\partial {}^tx} + {}^t\sigma_{yx}\frac{\partial u_y}{\partial {}^tx}\frac{\partial \delta u_y}{\partial {}^ty} + {}^t\sigma_{yy}\frac{\partial u_y}{\partial {}^ty}\frac{\partial \delta u_y}{\partial {}^ty} \\ & + {}^t\sigma_{zz}\frac{u_x}{{}^tx}\frac{\delta u_x}{{}^tx})\,{}^tdV\end{aligned} \tag{4.25}$$

Using (4.10) and (4.12), we get

$$\begin{aligned}\int_{{}^tV} {}^t\sigma_{ij}\delta_t\eta_{ij}\,{}^tdV &= (\delta\bar{\mathbf{u}})^T \int_{{}^tV} ({}_t\mathbf{B}^{L2})^{T\,t}\mathbf{S}^{L2}\,{}_t\mathbf{B}^{L2\,t}dV\bar{\mathbf{u}} \\ &= (\delta\bar{\mathbf{u}})^{T\,t}\mathbf{K}^{L2}\bar{\mathbf{u}}\end{aligned} \tag{4.26}$$

where

$${}_t\mathbf{B}^{L2} = \left[\begin{array}{c|cc|c} & \frac{\partial h_k}{\partial {}^tx} & 0 & \\ & \frac{\partial h_k}{\partial {}^ty} & 0 & \\ \cdots & 0 & \frac{\partial h_k}{\partial {}^tx} & \cdots \\ & 0 & \frac{\partial h_k}{\partial {}^ty} & \\ & \frac{h_k}{\sum_{j=1}^{N} h_j\,{}^t\bar{x}_j} & 0 & \end{array}\right]_{5\times 2N} \tag{4.27}$$

$$
{}^t\mathbf{S}^{L2} = \begin{bmatrix} {}^t\sigma_{xx} & {}^t\sigma_{xy} & 0 & 0 & 0 \\ {}^t\sigma_{xy} & {}^t\sigma_{yy} & 0 & 0 & 0 \\ 0 & 0 & {}^t\sigma_{xx} & {}^t\sigma_{xy} & 0 \\ 0 & 0 & {}^t\sigma_{xy} & {}^t\sigma_{yy} & 0 \\ 0 & 0 & 0 & 0 & {}^t\sigma_{zz} \end{bmatrix}_{5\times 5} \tag{4.28}
$$

and the incremental nonlinear geometric stiffness matrix ${}^t\mathbf{K}^{L2}$ is given by [48]

$$
{}^t\mathbf{K}^{L2} = \int_{{}^tV} ({}_t\mathbf{B}^{L2})^{T\,t}\mathbf{S}^{L2}\,{}_t\mathbf{B}^{L2\,t}dV \tag{4.29}
$$

4.2.4 Discretization of the first convective stiffness term

The first convective stiffness virtual work term due to ALE can be expanded as

$$
\begin{aligned}
\int_{{}^tV} (u_k - u_k^g)\,{}^t\sigma_{ij} \frac{\partial \delta\, {}_te_{ij}}{\partial {}^tx_k}\,{}^tdV = \int_{{}^tV} [(u_x - u_x^g)[{}^t\sigma_{xx} \frac{\partial^2 \delta u_x}{\partial {}^tx^2} + {}^t\sigma_{yy} \frac{\partial \delta u_y}{\partial {}^tx \partial {}^ty} \\
+ {}^t\sigma_{xy} (\frac{\partial^2 \delta u_x}{\partial {}^tx \partial {}^ty} + \frac{\partial^2 \delta u_y}{\partial {}^tx^2}) + {}^t\sigma_{zz} (\frac{1}{{}^tx} \frac{\partial \delta u_x}{\partial {}^tx} - \frac{\delta u_x}{{}^tx^2})] \\
+ (u_y - u_y^g)[{}^t\sigma_{xx} \frac{\partial^2 \delta u_x}{\partial {}^tx \partial {}^ty} + {}^t\sigma_{yy} \frac{\partial \delta u_y}{\partial {}^ty^2} \\
+ {}^t\sigma_{xy} (\frac{\partial^2 \delta u_x}{\partial {}^ty^2} + \frac{\partial^2 \delta u_y}{\partial {}^tx \partial {}^ty}) + {}^t\sigma_{zz} \frac{1}{{}^tx} \frac{\partial \delta u_x}{\partial {}^ty}]]\,{}^tdV
\end{aligned} \tag{4.30}
$$

It should be noted that this term involves second derivatives of displacements and consequently second derivatives of shape functions. Substituting by (4.10) and (4.12) and defining the matrices

$$
{}_{t}\mathbf{B}^{A1} = \left[\cdots \begin{array}{|cc|} \dfrac{\partial^2 h_k}{\partial^t x^2} & 0 \\ \dfrac{\partial^2 h_k}{\partial^t y^2} & 0 \\ \dfrac{\partial^2 h_k}{\partial^t x \partial^t y} & 0 \\ 0 & \dfrac{\partial^2 h_k}{\partial^t x^2} \\ 0 & \dfrac{\partial^2 h_k}{\partial^t y^2} \\ 0 & \dfrac{\partial^2 h_k}{\partial^t x \partial^t y} \\ 0 & (\dfrac{1}{{}^t x}\dfrac{\partial h_k}{\partial^t x} - \dfrac{h_k}{{}^t x^2}) \\ 0 & \dfrac{1}{{}^t x}\dfrac{\partial h_k}{\partial^t y} \end{array} \cdots \right]_{8\times 2N} \tag{4.31}
$$

and

$$
{}^t\mathbf{S}^{A1} = \begin{bmatrix} {}^t\sigma_{xx} & 0 & {}^t\sigma_{xy} & {}^t\sigma_{xy} & 0 & {}^t\sigma_{yy} & {}^t\sigma_{zz} & 0 \\ 0 & {}^t\sigma_{xy} & {}^t\sigma_{xx} & 0 & {}^t\sigma_{yy} & {}^t\sigma_{xy} & 0 & {}^t\sigma_{zz} \end{bmatrix}^T_{2\times 8} \tag{4.32}
$$

we get

$$
\begin{aligned}
\int_{{}^tV} (u_k - u_k^g)^t\sigma_{ij} \frac{\partial\delta\, {}_t e_{ij}}{\partial^t x_k}{}^t dV &= (\delta\overline{\mathbf{u}})^T \int_{{}^tV} ({}_t\mathbf{B}^{A1})^{T\,t}\mathbf{S}^{A1}\mathbf{H}^t dV(\overline{\mathbf{u}} - \overline{\mathbf{u}}^g) \\
&= (\delta\overline{\mathbf{u}})^{T\,t}\mathbf{K}^{A1}(\overline{\mathbf{u}} - \overline{\mathbf{u}}^g)
\end{aligned} \tag{4.33}
$$

where the first convective stiffness matrix due to ALE is given by

$$
{}^t\mathbf{K}^{A1} = \int_{{}^tV} ({}_t\mathbf{B}^{A1})^{T\,t}\mathbf{S}^{A1}\mathbf{H}^t dV \tag{4.34}
$$

4.2.5 Discretization of the second convective stiffness term

The second convective stiffness virtual work term due to ALE can be expanded as

$$
\begin{aligned}
\int_{^tV}(\frac{\partial u_k}{\partial {}^tx_j}-\frac{\partial u_k^g}{\partial {}^tx_j}){}^t\sigma_{ij}\frac{\partial \delta u_i}{\partial {}^tx_k}{}^tdV = \int_{^tV}[&(\frac{\partial u_x}{\partial {}^tx}-\frac{\partial u_x^g}{\partial {}^tx})({}^t\sigma_{xx}\frac{\partial \delta u_x}{\partial {}^tx}+{}^t\sigma_{yx}\frac{\partial \delta u_y}{\partial {}^tx}) \\
&+(\frac{\partial u_x}{\partial {}^ty}-\frac{\partial u_x^g}{\partial {}^ty})({}^t\sigma_{xy}\frac{\partial \delta u_x}{\partial {}^tx}+{}^t\sigma_{yy}\frac{\partial \delta u_y}{\partial {}^tx}) \\
&+(\frac{\partial u_y}{\partial {}^tx}-\frac{\partial u_y^g}{\partial {}^tx})({}^t\sigma_{xx}\frac{\partial \delta u_x}{\partial {}^ty}+{}^t\sigma_{yx}\frac{\partial \delta u_y}{\partial {}^ty}) \\
&+(\frac{\partial u_y}{\partial {}^ty}-\frac{\partial u_y^g}{\partial {}^ty})({}^t\sigma_{xy}\frac{\partial \delta u_x}{\partial {}^ty}+{}^t\sigma_{yy}\frac{\partial \delta u_y}{\partial {}^ty}) \\
&+\frac{1}{{}^tx^2}(u_x-u_x^g){}^t\sigma_{zz}\delta u_x]{}^tdV \qquad (4.35)
\end{aligned}
$$

Substituting by (4.10) and (4.12) and defining the matrix

$$
{}_t\mathbf{B}^{A2} = \left[\begin{array}{c|cc|c}
 & \frac{\partial h_k}{\partial {}^tx} & 0 & \\
 & 0 & \frac{\partial h_k}{\partial {}^tx} & \\
\cdots & \frac{\partial h_k}{\partial {}^ty} & 0 & \cdots \\
 & 0 & \frac{\partial h_k}{\partial {}^ty} & \\
 & \frac{h_k}{\sum_{j=1}^{N} h_j {}^t\bar{x}_j} & 0 &
\end{array}\right]_{5\times 2N} \qquad (4.36)
$$

we get

$$
\begin{aligned}
\int_{^tV}(\frac{\partial u_k}{\partial {}^tx_j}-\frac{\partial u_k^g}{\partial {}^tx_j}){}^t\sigma_{ij}\frac{\partial \delta u_i}{\partial {}^tx_k}{}^tdV &= (\delta\bar{\mathbf{u}})^T\int_{^tV}({}_t\mathbf{B}^{A2})^{T\,t}\mathbf{S}^{L2}\mathbf{B}^{L2\,t}dV(\bar{\mathbf{u}}-\bar{\mathbf{u}}^g) \\
&= (\delta\bar{\mathbf{u}})^{T\,t}\mathbf{K}^{A2}(\bar{\mathbf{u}}-\bar{\mathbf{u}}^g) \qquad (4.37)
\end{aligned}
$$

where the second convective stiffness matrix due to ALE is given by

$$
{}^t\mathbf{K}^{A2} = \int_{^tV}({}_t\mathbf{B}^{A2})^{T\,t}\mathbf{S}^{L2}\mathbf{B}^{L2\,t}dV \qquad (4.38)
$$

4.2.6 Discretized Quasi-Static ALE Equation

Substituting (4.20), (4.22), (4.26), (4.33) and (4.37) into (4.13), we get, for a single element or a group of elements

$$({}^{t}\mathbf{K}^{L1}+{}^{t}\mathbf{K}^{L2})\overline{\mathbf{u}}+({}^{t}\mathbf{K}^{A1}+{}^{t}\mathbf{K}^{A2})(\overline{\mathbf{u}}-\overline{\mathbf{u}}^{g})={}^{t+\Delta t}\mathbf{f}^{ext}-{}^{t}\mathbf{f} \tag{4.39}$$

or

$${}^{t}\mathbf{K}^{L}\overline{\mathbf{u}}+{}^{t}\mathbf{K}^{A}(\overline{\mathbf{u}}-\overline{\mathbf{u}}^{g})={}^{t+\Delta t}\mathbf{f}^{ext}-{}^{t}\mathbf{f} \tag{4.40}$$

where

$${}^{t}\mathbf{K}^{L}={}^{t}\mathbf{K}^{L1}+{}^{t}\mathbf{K}^{L2} \tag{4.41}$$

$${}^{t}\mathbf{K}^{A}={}^{t}\mathbf{K}^{A1}+{}^{t}\mathbf{K}^{A2} \tag{4.42}$$

${}^{t}\mathbf{K}^{L}$ and ${}^{t}\mathbf{K}^{A}$ are the Lagrangian and ALE stiffness matrices, respectively. Equation (4.40) represents the finite element matrix equation for quasi-static ALE analysis.

4.3 DISCRETIZATION OF THE DYNAMIC ALE EQUATION

The ALE virtual work equation of motion for dynamic analysis was derived in Chapter 3 in the form

$$
\begin{aligned}
&+\int_{{}^{t}V} {}^{t}\rho a_i \delta u_i \, {}^{t}dV + \int_{{}^{t}V} {}^{t}C_{ijkl\,t}e_{kl}\delta\,{}_{t}e_{ij}\, {}^{t}dV + \int_{{}^{t}V} {}^{t}\sigma_{ij}\delta\,{}_{t}\eta_{ij}\, {}^{t}dV \\
&+\int_{{}^{t}V} {}^{t}\rho({}^{t}v_j - {}^{t}v_j^g)\frac{\partial v_i}{\partial^{t}x_j}\delta u_i\, {}^{t}dV + \int_{{}^{t}V} {}^{t}\rho(v_j - v_j^g)\frac{\partial^{t}v_i}{\partial^{t}x_j}\delta u_i\, {}^{t}dV \\
&+\int_{{}^{t}V} (u_k - u_k^g)\,{}^{t}\sigma_{ij}\frac{\partial\delta\,{}_{t}e_{ij}}{\partial^{t}x_k}\, {}^{t}dV + \int_{{}^{t}V} \left(\frac{\partial u_k}{\partial^{t}x_j} - \frac{\partial u_k^g}{\partial^{t}x_j}\right){}^{t}\sigma_{ij}\frac{\partial\delta u_i}{\partial^{t}x_k}\, {}^{t}dV \\
&=\delta^{t+\Delta t}W^{ext} - \int_{{}^{t}V} {}^{t}\sigma_{ij}\delta\,{}_{t}e_{ij}\, {}^{t}dV - \int_{{}^{t}V} {}^{t}\rho\, {}^{t}a_i \delta u_i\, {}^{t}dV \\
&-\int_{{}^{t}V} {}^{t}\rho({}^{t}v_k - {}^{t}v_k^g)\frac{\partial({}^{t}a_i\delta u_i)}{\partial^{t}x_k}\Delta t\, {}^{t}dV - \int_{{}^{t}V} {}^{t}\rho({}^{t}v_j - {}^{t}v_j^g)\frac{\partial^{t}v_i}{\partial^{t}x_j}\delta u_i\, {}^{t}dV \\
&-\int_{{}^{t}V} {}^{t}\rho({}^{t}v_k - {}^{t}v_k^g)\left(\frac{\partial^{t}v_j}{\partial^{t}x_k} - 2\frac{\partial^{t}v_j^g}{\partial^{t}x_k}\right)\frac{\partial^{t}v_i}{\partial^{t}x_j}\delta u_i \Delta t\, {}^{t}dV \\
&-\int_{{}^{t}V} {}^{t}\rho({}^{t}v_k - {}^{t}v_k^g)({}^{t}v_j - {}^{t}v_j^g)\frac{\partial}{\partial^{t}x_k}\left(\frac{\partial^{t}v_i}{\partial^{t}x_j}\delta u_i\right)\Delta t\, {}^{t}dV
\end{aligned}
\tag{4.43}
$$

As indicated in Chapter 3, the first term on the LHS of (4.43) correspond to the Lagrangian mass matrix, whereas the fourth and fifth terms give rise to convective velocity stiffness matrices due to ALE. The third term on the RHS of (4.43) corresponds to the Lagrangian inertia force vector whereas the last four terms on the RHS are convective inertia force vectors due to ALE.

The discretization of velocities and accelerations is similar to that of displacements and coordinates given by (4.6), (4.10) and (4.11). Thus we can write,

$$ {}^{t}\mathbf{v} = \mathbf{H}\,{}^{t}\overline{\mathbf{v}} \tag{4.44} $$

$$ {}^{t}\mathbf{v}^{g} = \mathbf{H}\,{}^{t}\overline{\mathbf{v}}^{g} \tag{4.45} $$

$$ {}^{t}\mathbf{a} = \mathbf{H}\,{}^{t}\overline{\mathbf{a}} \tag{4.46} $$

$$ \mathbf{v} = \mathbf{H}\overline{\mathbf{v}} \tag{4.47} $$

$$ \mathbf{v}^{g} = \mathbf{H}\overline{\mathbf{v}}^{g} \tag{4.48} $$

$$\mathbf{a} = \mathbf{H}\bar{\mathbf{a}} \tag{4.49}$$

where ${}^t\bar{\mathbf{v}}$, ${}^t\bar{\mathbf{v}}^g$ and ${}^t\bar{\mathbf{a}}$ are the nodal material velocity, grid velocity and referential material acceleration at time t whereas $\bar{\mathbf{v}}$, $\bar{\mathbf{v}}^g$ and $\bar{\mathbf{a}}$ are the nodal incremental quantities from time t to $t + \Delta t$.

4.3.1 Discretization of the incremental mass term

The incremental Lagrangian mass term can be discretized using (4.49) to give

$$\begin{aligned}\int_{{}^tV} {}^t\rho a_i \delta u_i {}^t dV &= \int_{{}^tV} {}^t\rho (\mathbf{H}\delta\bar{\mathbf{u}})^T (\mathbf{H}\bar{\mathbf{a}}) {}^t dV \\ &= (\delta\bar{\mathbf{u}})^T \int_{{}^tV} {}^t\rho \mathbf{H}^T \mathbf{H} {}^t dV \bar{\mathbf{a}} \\ &= (\delta\bar{\mathbf{u}})^T \, {}^t\mathbf{M}^L \bar{\mathbf{a}}\end{aligned} \tag{4.50}$$

where ${}^t\mathbf{M}^L$ is the Lagrangian mass matrix given by

$${}^t\mathbf{M}^L = \int_{{}^tV} {}^t\rho \mathbf{H}^T \mathbf{H} {}^t dV \tag{4.51}$$

4.3.2 Discretization of the first convective velocity-stiffness term

The first convective velocity-stiffness virtual work term can be discretized using (4.44), (4.45) and (4.47) and expressed in the form

$$\int_{{}^tV} {}^t\rho ({}^tv_j - {}^tv_j^g) \frac{\partial v_i}{\partial {}^t x_j} \delta u_i {}^t dV = (\delta\bar{\mathbf{u}})^T \, {}^t\mathbf{C}^{A1} \bar{\mathbf{v}} \tag{4.52}$$

where ${}^t\mathbf{C}^{A1}$ is given by

$${}^t\mathbf{C}^{A1} = \left[\begin{array}{c|cc|c} & \vdots & & \\ \hline \cdots & {}^tC^{A1}_{2i-1,2j-1} & {}^tC^{A1}_{2i-1,2j} & \cdots \\ & {}^tC^{A1}_{2i,2j-1} & {}^tC^{A1}_{2i,2j} & \\ \hline & \vdots & & \end{array}\right]_{2N\times 2N} \tag{4.53}$$

in which *i* and *j* indicate node numbers from 1 to *N*, and

$$ {}^{t}C^{A1}_{2i-1,2j-1} = {}^{t}C^{A1}_{2i,2j} = \int_{{}^{t}V} {}^{t}\rho h_i [\frac{\partial h_j}{\partial^t x} \sum_{k=1}^{N} h_k ({}^{t}\overline{v}_{xk} - {}^{t}\overline{v}^{g}_{xk}) + \frac{\partial h_j}{\partial^t y} \sum_{k=1}^{N} h_k ({}^{t}\overline{v}_{yk} - {}^{t}\overline{v}^{g}_{yk})]^{t} dV \qquad (4.54) $$

$$ {}^{t}C^{A1}_{2i-1,2j} = {}^{t}C^{A1}_{2i,2j-1} = 0 \qquad (4.55) $$

4.3.3 Discretization of the second convective velocity-stiffness term

The second convective velocity-stiffness virtual work term is discretized using (4.44), (4.47) and (4.48) and expressed in the form

$$ \int_{{}^{t}V} {}^{t}\rho (v_j - v^{g}_j) \frac{\partial^t v_i}{\partial^t x_j} \delta u_i {}^{t}dV = (\delta \overline{\mathbf{u}})^{T}\, {}^{t}\mathbf{C}^{A2} (\overline{\mathbf{v}} - \overline{\mathbf{v}}^{g}) \qquad (4.56) $$

where ${}^{t}\mathbf{C}^{A2}$ is given by

$$ {}^{t}\mathbf{C}^{A2} = \left[\begin{array}{c|cc|c} & & \vdots & \\ \hline \cdots & {}^{t}C^{A2}_{2i-1,2j-1} & {}^{t}C^{A2}_{2i-1,2j} & \cdots \\ & {}^{t}C^{A2}_{2i,2j-1} & {}^{t}C^{A2}_{2i,2j} & \\ \hline & & \vdots & \end{array} \right]_{2N \times 2N} \qquad (4.57) $$

in which *i* and *j* indicate node numbers from 1 to *N*, and

$$ {}^{t}C^{A2}_{2i-1,2j-1} = \int_{{}^{t}V} {}^{t}\rho h_i h_j \sum_{k=1}^{N} \frac{\partial h_k}{\partial^t x} {}^{t}\overline{v}_{xk} {}^{t}dV \qquad (4.58) $$

$$ {}^{t}C^{A2}_{2i-1,2j} = \int_{{}^{t}V} {}^{t}\rho h_i h_j \sum_{k=1}^{N} \frac{\partial h_k}{\partial^t y} {}^{t}\overline{v}_{xk} {}^{t}dV \qquad (4.59) $$

$$ {}^{t}C^{A2}_{2i,2j-1} = \int_{{}^{t}V} {}^{t}\rho h_i h_j \sum_{k=1}^{N} \frac{\partial h_k}{\partial^t x} {}^{t}\overline{v}_{yk} {}^{t}dV \qquad (4.60) $$

$$ {}^{t}C^{A2}_{2i,2j} = \int_{{}^{t}V} {}^{t}\rho h_i h_j \sum_{k=1}^{N} \frac{\partial h_k}{\partial^t y} {}^{t}\overline{v}_{yk} {}^{t}dV \qquad (4.61) $$

4.3.4 Discretization of the inertia force term

The inertia force virtual work term is discretized using (4.46) to give

$$\begin{aligned}\int_{{}^tV} {}^t\rho\,{}^ta_i\delta u_i\,{}^tdV &= \int_{{}^tV} {}^t\rho(\mathbf{H}\delta\bar{\mathbf{u}})^T(\mathbf{H}\,{}^t\bar{\mathbf{a}})\,{}^tdV \\ &= (\delta\bar{\mathbf{u}})^T \int_{{}^tV} {}^t\rho\mathbf{H}^T\mathbf{H}\,{}^tdV\,{}^t\bar{\mathbf{a}} \\ &= (\delta\bar{\mathbf{u}})^T\,{}^t\mathbf{M}^L\,{}^t\bar{\mathbf{a}} \qquad (4.62)\end{aligned}$$

where ${}^t\mathbf{M}^L$ is the mass matrix given by (4.51).

4.3.5 Discretization of the first convective inertia force term

The first inertia force virtual work term can be discretized using (4.44), (4.45) and (4.46) and expressed in the form

$$\int_{{}^tV} {}^t\rho({}^tv_k - {}^tv_k^g)\frac{\partial({}^ta_i\delta u_i)}{\partial\,{}^tx_k}\Delta t\,{}^tdV = (\delta\bar{\mathbf{u}})^T\,{}^t\mathbf{M}^A\,{}^t\bar{\mathbf{a}} \qquad (4.63)$$

where ${}^t\mathbf{M}^A$ is given by

$${}^t\mathbf{M}^A = \begin{bmatrix} & \vdots & \\ \cdots & \begin{array}{cc} {}^tM^A_{2i-1,2j-1} & {}^tM^A_{2i-1,2j} \\ {}^tM^A_{2i,2j-1} & {}^tM^A_{2i,2j} \end{array} & \cdots \\ & \vdots & \end{bmatrix}_{2N\times 2N} \qquad (4.64)$$

in which *i* and *j* indicate node numbers from 1 to *N*, and

$$\begin{aligned}{}^tM^A_{2i-1,2j-1} = {}^tM^A_{2i,2j} &= \int_{{}^tV} {}^t\rho\Big[\Big(h_i\frac{\partial h_j}{\partial\,{}^tx} + h_j\frac{\partial h_i}{\partial\,{}^tx}\Big)\sum_{k=1}^N h_k({}^t\bar{v}_{xk} - {}^t\bar{v}^g_{xk}) \\ &+ \Big(h_i\frac{\partial h_j}{\partial\,{}^ty} + h_j\frac{\partial h_i}{\partial\,{}^ty}\Big)\sum_{k=1}^N h_k({}^t\bar{v}_{yk} - {}^t\bar{v}^g_{yk})\Big]\Delta t\,{}^tdV \qquad (4.65)\end{aligned}$$

$$ {}^tM^A_{2i-1,2j} = {}^tM^A_{2i,2j-1} = 0 \tag{4.66} $$

4.3.6 Discretization of the second convective inertia force term

The second convective inertia force virtual work term can be discretized using (4.44) and (4.45) and expressed in the form

$$ \int_{{}^tV} {}^t\rho({}^tv_j - {}^tv^g_j)\frac{\partial {}^tv_i}{\partial {}^tx_j}\delta u_i \, {}^tdV = (\delta\bar{\mathbf{u}})^T \, {}^t\mathbf{C}^{A1} \, {}^t\bar{\mathbf{v}} \tag{4.67} $$

where ${}^t\mathbf{C}^{A1}$ is given by (4.53) to (4.55).

4.3.7 Discretization of the third convective inertia force term

The third convective inertia force virtual work term can be discretized using (4.44) and (4.45) and expressed in the form

$$ \int_{{}^tV} {}^t\rho({}^tv_k - {}^tv^g_k)\left(\frac{\partial {}^tv_j}{\partial {}^tx_k} - 2\frac{\partial {}^tv^g_j}{\partial {}^tx_k}\right)\frac{\partial {}^tv_i}{\partial {}^tx_j}\delta u_i \Delta t \, {}^tdV = (\delta\bar{\mathbf{u}})^T \, {}^t\mathbf{C}^{A3} \, {}^t\bar{\mathbf{v}} \tag{4.68} $$

where ${}^t\mathbf{C}^{A3}$ is given by

$$ {}^t\mathbf{C}^{A3} = \left[\begin{array}{c:cc:c} & & \vdots & \\ \hdashline \cdots & {}^tC^{A3}_{2i-1,2j-1} & {}^tC^{A3}_{2i-1,2j} & \cdots \\ & {}^tC^{A3}_{2i,2j-1} & {}^tC^{A3}_{2i,2j} & \\ \hdashline & & \vdots & \end{array}\right]_{2N\times 2N} \tag{4.69} $$

in which *i* and *j* indicate node numbers from 1 to *N*, and

$$\begin{aligned}{}^{t}C^{A3}_{2i-1,2j-1}={}^{t}C^{A3}_{2i,2j} = \int_{{}^{t}V} {}^{t}\rho h_i[[&\frac{\partial h_j}{\partial^t x}\sum_{k=1}^{N}\frac{\partial h_k}{\partial^t x}({}^{t}\overline{v}_{xk} -2{}^{t}\overline{v}^{g}_{xk}) \\ &+\frac{\partial h_j}{\partial^t y}\sum_{k=1}^{N}\frac{\partial h_k}{\partial^t x}({}^{t}\overline{v}_{yk} -2{}^{t}\overline{v}^{g}_{yk})]\sum_{k=1}^{N}h_k({}^{t}\overline{v}_{xk}-{}^{t}\overline{v}^{g}_{xk}) \\ &+[\frac{\partial h_j}{\partial^t x}\sum_{k=1}^{N}\frac{\partial h_k}{\partial^t y}({}^{t}\overline{v}_{xk} -2{}^{t}\overline{v}^{g}_{xk}) \\ &+\frac{\partial h_j}{\partial^t y}\sum_{k=1}^{N}\frac{\partial h_k}{\partial^t y}({}^{t}\overline{v}_{yk} -2{}^{t}\overline{v}^{g}_{yk})]\sum_{k=1}^{N}h_k({}^{t}\overline{v}_{yk}-{}^{t}\overline{v}^{g}_{yk})]\Delta t{}^{t}dV\end{aligned} \tag{4.70}$$

$${}^{t}C^{A3}_{2i-1,2j}={}^{t}C^{A3}_{2i,2j-1} = 0 \tag{4.71}$$

4.3.8 Discretization of the fourth convective inertia force term

The fourth convective inertia force virtual work term can be discretized using (4.44) and (4.45) and expressed in the form

$$\int_{{}^{t}V} {}^{t}\rho({}^{t}v_k-{}^{t}v^{g}_{k})({}^{t}v_j-{}^{t}v^{g}_{j})\frac{\partial}{\partial^t x_k}(\frac{\partial^t v_i}{\partial^t x_j}\delta u_i)\Delta t{}^{t}dV = (\delta\overline{\mathbf{u}})^{T}\,{}^{t}\mathbf{C}^{A4}\,{}^{t}\overline{\mathbf{v}} \tag{4.72}$$

where ${}^{t}\mathbf{C}^{A4}$ is given by

$${}^{t}\mathbf{C}^{A4} = \left[\begin{array}{c:cc:c} & & \vdots & \\ \hdashline \cdots & {}^{t}C^{A4}_{2i-1,2j-1} & {}^{t}C^{A4}_{2i-1,2j} & \cdots \\ & {}^{t}C^{A4}_{2i,2j-1} & {}^{t}C^{A4}_{2i,2j} & \\ \hdashline & & \vdots & \end{array}\right]_{2N\times 2N} \tag{4.73}$$

in which *i* and *j* indicate node numbers from 1 to *N*, and

$$\begin{aligned}{}^{t}C^{A3}_{2i-1,2j-1}={}^{t}C^{A3}_{2i,2j} = \int_{{}^{t}V} {}^{t}\rho[(&\frac{\partial h_i}{\partial^t x}\frac{\partial h_j}{\partial^t x}+h_i\frac{\partial^2 h_j}{\partial^t x^2})[\sum_{k=1}^{N}\frac{\partial h_k}{\partial^t x}({}^{t}\overline{v}_{xk}-{}^{t}\overline{v}^{g}_{xk})]^2 \\ &+(\frac{\partial h_i}{\partial^t x}\frac{\partial h_j}{\partial^t y}+\frac{\partial h_i}{\partial^t y}\frac{\partial h_j}{\partial^t x}+2h_i\frac{\partial^2 h_j}{\partial^t x\partial^t y})\sum_{k=1}^{N}h_k({}^{t}\overline{v}_{xk}-{}^{t}\overline{v}^{g}_{xk})\sum_{k=1}^{N}h_k({}^{t}\overline{v}_{yk}-{}^{t}\overline{v}^{g}_{yk}) \\ &+(\frac{\partial h_i}{\partial^t y}\frac{\partial h_j}{\partial^t y}+h_i\frac{\partial^2 h_j}{\partial^t y^2})[\sum_{k=1}^{N}h_k({}^{t}\overline{v}_{yk}-{}^{t}\overline{v}^{g}_{yk})]^2]\Delta t{}^{t}dV\end{aligned} \tag{4.74}$$

$$ {}^{t}C^{A4}_{2i-1,2j} = {}^{t}C^{A4}_{2i,2j-1} = 0 \tag{4.75} $$

It should be noted that this term involves second order derivatives of shape functions.

4.3.9 Discretized Dynamic ALE Equation

Substituting into (4.43), we get, for a single element or a group of elements

$$ \begin{aligned} &{}^{t}\mathbf{M}\overline{\mathbf{a}} + {}^{t}\mathbf{C}^{A1}\overline{\mathbf{v}} + {}^{t}\mathbf{C}^{A2}(\overline{\mathbf{v}} - \overline{\mathbf{v}}^{g}) + ({}^{t}\mathbf{K}^{L1} + {}^{t}\mathbf{K}^{L2})\overline{\mathbf{u}} + ({}^{t}\mathbf{K}^{A1} + {}^{t}\mathbf{K}^{A2})(\overline{\mathbf{u}} - \overline{\mathbf{u}}^{g}) \\ &= {}^{t+\Delta t}\mathbf{f}^{ext} - {}^{t}\mathbf{f} - ({}^{t}\mathbf{M} + {}^{t}\mathbf{M}^{A}){}^{t}\overline{\mathbf{a}} - ({}^{t}\mathbf{C}^{A1} + {}^{t}\mathbf{C}^{A3} + {}^{t}\mathbf{C}^{A4}){}^{t}\overline{\mathbf{v}} \end{aligned} \tag{4.76} $$

Equation (4.76) represents the finite element matrix equation for dynamic ALE analysis.

Chapter 5

IMPLEMENTATION

5.1 MESH MOTION

Using the ALE formulation, the finite element grid points can be moved arbitrarily to maintain a homogeneous mesh and to properly represent boundary conditions throughout the deformation process. In this research, grid displacements are first related to material displacements through a set of arbitrary mesh motion parameters. This method is very efficient in controlling the motion of the grid in different parts of the domain. Specification of pure Lagrangian and pure Eulerian degrees of freedom using this method is a simple task. It is also very efficient in reducing the number of solution variables by condensing out grid displacements prior to solution. The choice of the arbitrary mesh motion parameters for interior degrees of freedom is handled by a special mesh motion scheme. Special treatment for mesh motion on free material boundaries is necessary to satisfy the ALE boundary constraint.

5.1.1 Grid displacement

In [39], grid displacements are related to material displacements using a relation of the form

$$u_j^g = \alpha_j + \beta_{(j)} u_{(j)} \Big|_{\text{no summation on } j} \tag{5.1}$$

where α_j and $\beta_{(j)}$ are two vectors of mesh motion parameters, and the brackets in the subscript (*j*) indicate no summation on *j*. Although this relation is quite efficient in many cases, it couples grid and material displacements at the same degree of freedom only, thus restricting the implementation of the ALE boundary constraint on material boundaries.

In this work, a better control over the mesh motion, especially on free material boundaries, is achieved by associating grid displacements with material displacements by the more general relation

$$\mathbf{u}^g = \boldsymbol{\alpha} + \mathbf{B}\mathbf{u} \tag{5.2}$$

where $\boldsymbol{\alpha}$ and $\mathbf{B}$ are a vector and a matrix of mesh motion parameters, respectively. Vector $\boldsymbol{\alpha}$ consists of appropriate grid displacements given by the mesh motion scheme while matrix $\mathbf{B}$ consists of factors that allow the coupling of grid and material displacements. $\mathbf{B}$ is usually chosen to be a diagonal matrix, i.e. grid and material displacements are coupled only at the same degree of freedom. On free material boundaries, however, it is sometimes necessary that all degrees of freedom of grid and material displacements be coupled at the same node. For two-dimensional problems, this would result in $\mathbf{B}$ being a tridiagonal matrix. As special cases to the general ALE motion, pure Lagrangian degrees of freedom can be obtained by setting $\boldsymbol{\alpha} = \mathbf{0}$ and $\mathbf{B} = \mathbf{I}$, whereas pure Eulerian degrees of freedom are obtained by using $\boldsymbol{\alpha} = \mathbf{0}$ and $\mathbf{B} = \mathbf{0}$.

5.1.2 Mesh motion for interior nodes

In this work, the transfinite mapping method [66, 67] is used as the mesh motion scheme for the degrees of freedom interior to any mesh region with four known boundary curves. This method provides a homogeneous mesh and matches the boundary of a given region at an infinite number of points. Another distinct advantage of the transfinite mapping method is that it allows the discrete representation of boundary curves, i.e. the coordinates and displacements of boundary nodes can be used to find the optimum position of the nodes internal to the region. It also allows for discontinuities in slope of boundary curves.

The transfinite mapping algorithm starts by partitioning the initial mesh into a number of regions of simpler forms. Considering the mapping of a typical distorted mesh region bounded by four curves $\phi_i(r,0)$, $\phi_i(r,1)$, $\phi_i(0,s)$ and $\phi_i(1,s)$ as shown in Figure 5.1, the new mesh coordinates can be obtained by mapping the region onto a unit square to give

$$\begin{aligned} {}^{t+\Delta t}x_i(r,s) &= (1-s)\phi_i(r,0) + s\phi_i(r,1) + (1-r)\phi_i(0,s) + r\phi_i(1,s) \\ &\quad -(1-r)(1-s)\phi_i(0,0) - (1-r)s\phi_i(0,1) - rs\phi_i(1,1) - r(1-s)\phi_i(1,0) \end{aligned} \tag{5.3}$$

where $0 \le r \le 1$ and $0 \le s \le 1$ are the normalized coordinates over the region. Thus, the mesh motion parameter α_i for degree of freedom i internal to a region can be given by the transfinite mapping method as

$$\alpha_i = {}^{t+\Delta t}x_i - {}^{t}x_i \tag{5.4}$$

5.1.3 Mesh motion on free material boundaries

For the transfinite mapping method to give a good quality mesh within a certain region, grid points on the boundaries of this region must be uniformly distributed. Consider point k located on a free material boundary as shown in Figure 5.2. Because of its location on a free material boundary, the ALE motion of point k is given by (5.2) as

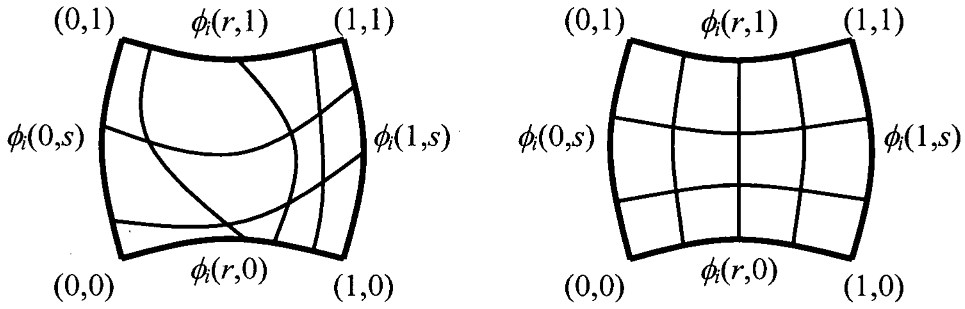


Figure 5.1. Transfinite mapping of a distorted mesh region.

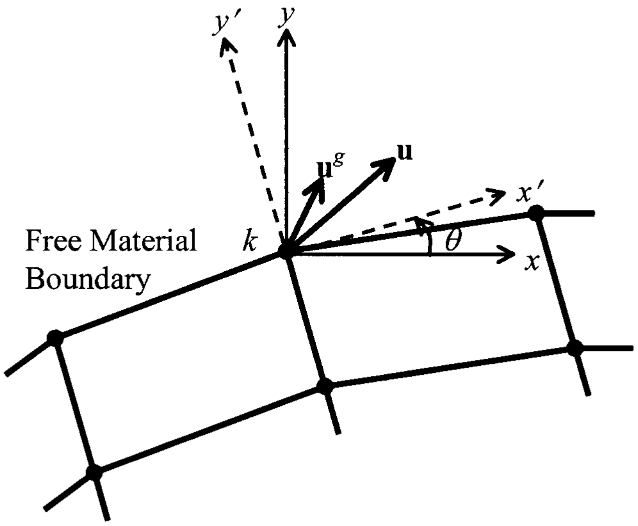


Figure 5.2. Mesh motion on free material boundaries.

$$u_x^g = \alpha_x + B_{xx} u_x + B_{xy} u_y \tag{5.5}$$

$$u_y^g = \alpha_y + B_{yx} u_x + B_{yy} u_y \tag{5.6}$$

where coupling between the *x* and *y* degrees of freedom is introduced in order that the grid motion of point *k* satisfies the boundary constraint given by equation (3.5). The question in this section is to find the mesh motion parameters α_x, B_{xx}, B_{xy}, α_y, B_{yx}, and B_{yy} for node *k* which satisfy the boundary constraint [68]. We define a set of local axes x' and y' at grid point *k* such that x' is tangent to the boundary and y' is its normal. Assume that the local x' and y' axes at point *k* are inclined to the global *x* and *y* axes by an angle θ. The components of the incremental material and grid displacement vectors in the global and local coordinate systems are related by

$$u_x' = u_x \cos\theta + u_y \sin\theta \tag{5.7}$$

$$u_y' = -u_x \sin\theta + u_y \cos\theta \tag{5.8}$$

$$u_x'^g = u_x^g \cos\theta + u_y^g \sin\theta \tag{5.9}$$

$$u_y'^g = -u_x^g \sin\theta + u_y^g \cos\theta \tag{5.10}$$

Meanwhile, the mesh motion equations referred to the local coordinate system can be written as

$$u_x'^g = \alpha_x' + B_{xx}' u_x' + B_{xy}' u_y' \tag{5.11}$$

$$u_y'^g = \alpha_y' + B_{yx}' u_x' + B_{yy}' u_y' \tag{5.12}$$

where α'_x, B'_{xx} and B'_{xy} are the ALE mesh motion parameters in the direction tangent to the boundary and which may be arbitrarily chosen, while α'_y, B'_{yx} and B'_{yy} are in the direction normal to the boundary controlled by the boundary constraint. In this work, cubic spline interpolation is used to find new coordinates for point *k* such that all grid points on this free material boundary are uniformly distributed. The difference between the old and new coordinates for point *k* establishes the mesh motion parameter α'_x. In this case B'_{xx} and B'_{xy} can be set to zero. The boundary constraint dictates that $u'^g_y = u'_y$ thus giving $\alpha'_y = 0$, $B'_{yx} = 0$ and $B'_{yy} = 1$. Substituting (5.7), (5.8) and (5.9) into (5.11) and letting $B'_{xx} = B'_{xy} = 0$, we get

$$u^g_x \cos\theta + u^g_y \sin\theta = \alpha'_x \tag{5.13}$$

Substituting (5.7), (5.8) and (5.10) into (5.12), and applying the boundary constraint $\alpha'_y = 0$, $B'_{yx} = 0$ and $B'_{yy} = 1$, we get

$$-u^g_x \sin\theta + u^g_y \cos\theta = -u_x \sin\theta + u_y \cos\theta \tag{5.14}$$

Solving (5.13) and (5.14) for u^g_x and u^g_y, we get

$$u^g_x = \alpha'_x \cos\theta + u_x \sin^2\theta - u_y \sin\theta\cos\theta \tag{5.15}$$

$$u^g_y = \alpha'_x \sin\theta - u_x \sin\theta\cos\theta + u_y \cos^2\theta \tag{5.16}$$

Comparing equations (5.15) and (5.16) with equations (5.5) and (5.6), we get

$$\alpha_x = \alpha'_x \cos\theta \tag{5.17}$$

$$B_{xx} = \sin^2\theta \tag{5.18}$$

$$B_{xy} = -\sin\theta\cos\theta \tag{5.19}$$

$$\alpha_y = \alpha'_x \sin\theta \tag{5.20}$$

$$B_{yx} = -\sin\theta\cos\theta \tag{5.21}$$

$$B_{yy} = \cos^2\theta \tag{5.22}$$

The above ALE mesh motion parameters ensure that mesh motion on free material boundaries is consistent with the boundary constraint given by equation (3.5).

5.2 SOLUTION OF NONLINEAR EQUILIBRIUM EQUATIONS

The ALE formulation derived in the previous chapters has been implemented into a nonlinear finite element code. Details of the computer solution of the nonlinear equilibrium equations for quasi-static and dynamic analysis are outlined in this section.

5.2.1 Quasi-static analysis

Due to the approximations involved in the linearization of equation (4.40), an iterative procedure must be used to ensure equilibrium. Employing the Newton-Raphson iterative scheme, equation (4.40) can be rewritten as

$$^{t+\Delta t}\mathbf{K}^{L(i-1)}\mathbf{u}^{(i)} + {}^{t+\Delta t}\mathbf{K}^{A(i-1)}(\mathbf{u}^{(i)} - \mathbf{u}^{g(i)}) = {}^{t+\Delta t}\mathbf{f}^{ext} - {}^{t+\Delta t}\mathbf{f}^{(i-1)} \tag{5.23}$$

where $^{t+\Delta t}\mathbf{K}^{L(i-1)}$ and $^{t+\Delta t}\mathbf{K}^{A(i-1)}$ are the tangent stiffness matrices in iteration $i-1$, $\mathbf{u}^{(i)}$ and $\mathbf{u}^{g(i)}$ are the corrections to the incremental material and grid displacement vectors in iteration i and which are used to update displacements according to

$$^{t+\Delta t}\mathbf{u}^{(i)} = {}^{t+\Delta t}\mathbf{u}^{(i-1)} + \mathbf{u}^{(i)} \tag{5.24}$$

$$^{t+\Delta t}\mathbf{u}^{g(i)} = {}^{t+\Delta t}\mathbf{u}^{g(i-1)} + \mathbf{u}^{g(i)} \tag{5.25}$$

Equation (5.23) corresponds to the full Newton iteration scheme. The calculation and factorization of the tangent stiffness matrices represents a major computational cost per iteration. The modified Newton iteration involves fewer stiffness reformations than the full Newton scheme. Using the modified Newton iteration, equation (5.23) is written as

$$ {}^{t}\mathbf{K}^{L}\mathbf{u}^{(i)}+{}^{t}\mathbf{K}^{A}(\mathbf{u}^{(i)}-\mathbf{u}^{g(i)})={}^{t+\Delta t}\mathbf{f}^{ext}-{}^{t+\Delta t}\mathbf{f}^{(i-1)} \tag{5.26}$$

However, convergence in the modified Newton iteration is in general slower. In this work, both methods have been implemented and the modified Newton iteration is found to be computationally more efficient. Rearranging equation (5.26) we get

$$({}^{t}\mathbf{K}^{L}+{}^{t}\mathbf{K}^{A})\mathbf{u}^{(i)}-{}^{t}\mathbf{K}^{A}\mathbf{u}^{g(i)}={}^{t+\Delta t}\mathbf{f}^{ext}-{}^{t+\Delta t}\mathbf{f}^{(i-1)} \tag{5.27}$$

5.2.2 Dynamic analysis

Using the modified Newton iteration, equation (4.76) for dynamic ALE analysis can be written in the form

$$\begin{aligned}&{}^{t}\mathbf{M}^{L}\mathbf{a}^{(i)}+{}^{t}\mathbf{C}^{A1}\mathbf{v}^{(i)}+{}^{t}\mathbf{C}^{A2}(\mathbf{v}^{(i)}-\mathbf{v}^{g(i)})+({}^{t}\mathbf{K}^{L1}+{}^{t}\mathbf{K}^{L2})\mathbf{u}^{(i)}+({}^{t}\mathbf{K}^{A1}+{}^{t}\mathbf{K}^{A2})(\mathbf{u}^{(i)}-\mathbf{u}^{g(i)})\\&={}^{t+\Delta t}\mathbf{f}^{ext}-{}^{t+\Delta t}\mathbf{f}^{(i-1)}-({}^{t}\mathbf{M}^{L}+{}^{t}\mathbf{M}^{A}){}^{t+\Delta t}\mathbf{a}^{(i-1)}-({}^{t}\mathbf{C}^{A1}+{}^{t}\mathbf{C}^{A3}+{}^{t}\mathbf{C}^{A4}){}^{t+\Delta t}\mathbf{v}^{(i-1)}\end{aligned} \tag{5.28}$$

in which $\mathbf{a}^{(i)}$, $\mathbf{v}^{(i)}$, $\mathbf{v}^{g(i)}$ are the corrections to the incremental material acceleration, material velocity and grid velocity vectors whereas ${}^{t+\Delta t}\mathbf{a}^{(i-1)}$ and ${}^{t+\Delta t}\mathbf{v}^{(i-1)}$ are the material acceleration and material velocity vectors at time $t+\Delta t$, iteration $i-1$. Next material acceleration, material velocity and grid velocity approximations are obtained using

$${}^{t+\Delta t}\mathbf{a}^{(i)}={}^{t+\Delta t}\mathbf{a}^{(i-1)}+\mathbf{a}^{(i)} \tag{5.29}$$

$${}^{t+\Delta t}\mathbf{v}^{(i)}={}^{t+\Delta t}\mathbf{v}^{(i-1)}+\mathbf{v}^{(i)} \tag{5.30}$$

$${}^{t+\Delta t}\mathbf{v}^{g(i)}={}^{t+\Delta t}\mathbf{v}^{g(i-1)}+\mathbf{v}^{g(i)} \tag{5.31}$$

Equation (5.28) can be rewritten in the form

$$\begin{aligned}&{}^t\mathbf{M}^L\mathbf{a}^{(i)} + ({}^t\mathbf{C}^{A1}+{}^t\mathbf{C}^{A2})\mathbf{v}^{(i)} - {}^t\mathbf{C}^{A2}\mathbf{v}^{g(i)} + ({}^t\mathbf{K}^L+{}^t\mathbf{K}^A)\mathbf{u}^{(i)} - {}^t\mathbf{K}^A\mathbf{u}^{g(i)} \\ &= {}^{t+\Delta t}\mathbf{f}^{ext} - {}^{t+\Delta t}\mathbf{f}^{(i-1)} - ({}^t\mathbf{M}^L+{}^t\mathbf{M}^A){}^{t+\Delta t}\mathbf{a}^{(i-1)} - ({}^t\mathbf{C}^{A1}+{}^t\mathbf{C}^{A3}+{}^t\mathbf{C}^{A4}){}^{t+\Delta t}\mathbf{v}^{(i-1)} \qquad (5.32)\end{aligned}$$

In this work, two methods are used to integrate the dynamic ALE equilibrium equations: implicit time integration using the Newmark scheme and mixed implicit-explicit time integration using the predictor-corrector algorithm. It is noted that the latter scheme might be advantageous in some cases to avoid convergence problems.

(i) Implicit time integration using the Newmark scheme

Using Newmark implicit time integration, the following assumptions are used [48]

$${}^{t+\Delta t}\mathbf{u}^{(i)} = {}^t\mathbf{u} + \Delta t\,{}^t\mathbf{v} + \Delta t^2(\frac{1}{2}-\beta)\,{}^t\mathbf{a} + \Delta t^2\beta\,{}^{t+\Delta t}\mathbf{a}^{(i)} \qquad (5.33)$$

$${}^{t+\Delta t}\mathbf{v}^{(i)} = {}^t\mathbf{v} + \Delta t(1-\gamma)\,{}^t\mathbf{a} + \Delta t\gamma\,{}^{t+\Delta t}\mathbf{a}^{(i)} \qquad (5.34)$$

where β and γ are parameters that control the accuracy and stability of integration. Rearranging (5.33) we get,

$${}^{t+\Delta t}\mathbf{a}^{(i)} = \frac{1}{\Delta t^2\beta}[{}^{t+\Delta t}\mathbf{u}^{(i)} - {}^t\mathbf{u} - \Delta t\,{}^t\mathbf{v} - \Delta t^2(\frac{1}{2}-\beta)\,{}^t\mathbf{a}] \qquad (5.35)$$

Using (5.24)

$$\begin{aligned}{}^{t+\Delta t}\mathbf{a}^{(i)} &= \frac{1}{\Delta t^2\beta}[{}^{t+\Delta t}\mathbf{u}^{(i-1)} + \mathbf{u}^{(i)} - {}^t\mathbf{u} - \Delta t\,{}^t\mathbf{v} - \Delta t^2(\frac{1}{2}-\beta)\,{}^t\mathbf{a}] \\ &= \frac{1}{\Delta t^2\beta}[{}^{t+\Delta t}\mathbf{u}^{(i-1)} - {}^t\mathbf{u} - \Delta t\,{}^t\mathbf{v} - \Delta t^2(\frac{1}{2}-\beta)\,{}^t\mathbf{a}] + \frac{1}{\Delta t^2\beta}\mathbf{u}^{(i)} \\ &= {}^{t+\Delta t}\mathbf{a}^{(i-1)} + \frac{1}{\Delta t^2\beta}\mathbf{u}^{(i)} \qquad (5.36)\end{aligned}$$

Comparing (5.36) with (5.29) we get

$$\mathbf{a}^{(i)} = \frac{1}{\Delta t^2 \beta}\mathbf{u}^{(i)} \tag{5.37}$$

Substituting (5.36) into (5.34)

$$\begin{aligned} {}^{t+\Delta t}\mathbf{v}^{(i)} &= {}^{t}\mathbf{v} + \Delta t(1-\gamma){}^{t}\mathbf{a} + \Delta t\gamma{}^{t+\Delta t}\mathbf{a}^{(i-1)} + \Delta t\gamma\frac{1}{\Delta t^2\beta}\mathbf{u}^{(i)} \\ &= {}^{t+\Delta t}\mathbf{v}^{(i-1)} + \frac{\gamma}{\Delta t\beta}\mathbf{u}^{(i)} \end{aligned} \tag{5.38}$$

Comparing (5.38) with (5.30)

$$\mathbf{v}^{(i)} = \frac{\gamma}{\Delta t\beta}\mathbf{u}^{(i)} \tag{5.39}$$

Similarly

$$\mathbf{v}^{g(i)} = \frac{\gamma}{\Delta t\beta}\mathbf{u}^{g(i)} \tag{5.40}$$

Substituting (5.37), (5.39) and (5.40) into (5.32) we get

$$\begin{aligned} &[\frac{1}{\Delta t^2\beta}{}^{t}\mathbf{M}^{L} + \frac{\gamma}{\Delta t\beta}({}^{t}\mathbf{C}^{A1} + {}^{t}\mathbf{C}^{A2}) + ({}^{t}\mathbf{K}^{L} + {}^{t}\mathbf{K}^{A})]\mathbf{u}^{(i)} - [\frac{\gamma}{\Delta t\beta}{}^{t}\mathbf{C}^{A2} + {}^{t}\mathbf{K}^{A}]\mathbf{u}^{g(i)} \\ &= {}^{t+\Delta t}\mathbf{f}^{ext} - {}^{t+\Delta t}\mathbf{f}^{(i-1)} - ({}^{t}\mathbf{M}^{L} + {}^{t}\mathbf{M}^{A}){}^{t+\Delta t}\mathbf{a}^{(i-1)} - ({}^{t}\mathbf{C}^{A1} + {}^{t}\mathbf{C}^{A3} + {}^{t}\mathbf{C}^{A4}){}^{t+\Delta t}\mathbf{v}^{(i-1)} \end{aligned} \tag{5.41}$$

(ii) Implicit-explicit predictor-corrector integration scheme

The implicit-explicit predictor-corrector integration scheme associated with the Newmark algorithm can allow the finite element mesh to contain two groups of elements: an implicit group and an explicit group [69, 70]. In addition, elements' integrations could be shifted between the two schemes as necessary. In the predictor phase, we set $i = 1$ and use the predictor values:

$$ {}^{t+\Delta t}\mathbf{u}^{(1)} = {}^{t+\Delta t}\mathbf{u}^{P} = {}^{t}\mathbf{u} + \Delta t\,{}^{t}\mathbf{v} + \Delta t^{2}(\frac{1}{2} - \beta)\,{}^{t}\mathbf{a} \tag{5.42} $$

$$ {}^{t+\Delta t}\mathbf{v}^{(1)} = {}^{t+\Delta t}\mathbf{v}^{P} = {}^{t}\mathbf{v} + \Delta t(1-\gamma)\,{}^{t}\mathbf{a} \tag{5.43} $$

$$ {}^{t+\Delta t}\mathbf{a}^{(1)} = \mathbf{0} \tag{5.44} $$

Iterations are then performed to satisfy the equilibrium equation in (5.41). In the corrector phase of each iteration, displacements, accelerations and velocities are updated according to:

$$ {}^{t+\Delta t}\mathbf{u}^{(i)} = {}^{t+\Delta t}\mathbf{u}^{(i-1)} + \mathbf{u}^{(i)} \tag{5.45} $$

$$ {}^{t+\Delta t}\mathbf{a}^{(i)} = ({}^{t+\Delta t}\mathbf{u}^{(i)} - {}^{t+\Delta t}\mathbf{u}^{P})/(\Delta t^{2}\beta) \tag{5.46} $$

$$ {}^{t+\Delta t}\mathbf{v}^{(i)} = {}^{t+\Delta t}\mathbf{v}^{P} + \Delta t\gamma\,{}^{t+\Delta t}\mathbf{a}^{(i)} \tag{5.47} $$

5.2.3 Elimination of grid displacements on the element level

Finite element equilibrium equations, equation (5.27) for quasi-static analysis and equation (5.41) for dynamic analysis, can both be written in the general form

$$ {}^{t}\mathbf{K}\mathbf{u}^{(i)} - {}^{t}\mathbf{K}^{g}\mathbf{u}^{g(i)} = \mathbf{f}^{(i)} \tag{5.48} $$

where ${}^{t}\mathbf{K}$ and ${}^{t}\mathbf{K}^{g}$ are equivalent stiffness matrices corresponding to $\mathbf{u}^{(i)}$ and $\mathbf{u}^{g(i)}$, respectively, while $\mathbf{f}^{(i)}$ is the incremental load vector for iteration *i*. The relation between the material and grid displacements in (5.2) is considered as a supplementary constraint equation to the finite element equilibrium equation. By introducing this constraint on the element level, grid displacements can be condensed out of element equilibrium equations prior to solution. Substituting (5.2) into (5.48), we get

$$({}^t\mathbf{K} - {}^t\mathbf{K}^g\mathbf{B})\mathbf{u}^{(i)} = \mathbf{f}^{(i)} + {}^t\mathbf{K}^g\boldsymbol{\alpha} \tag{5.49}$$

The only limitation to this procedure is that grid and material displacements may only be coupled at degrees of freedom within one element. Equation (5.49) can be rewritten in the form

$${}^t\mathbf{K}^*\mathbf{u}^{(i)} = \mathbf{f}^* \tag{5.50}$$

where ${}^t\mathbf{K}^*$ and $\mathbf{f}^*$ are the equivalent stiffness matrix and nodal force vector respectively. Conventional finite element assembly and elimination techniques may now be applied directly to solve for the material displacements.

5.2.4 Frontal solver

A frontal or a wavefront solver [71, 72] that can handle asymmetric matrices is implemented to solve the simultaneous linear equations in (5.50). The number of equations that are active after any element has been processed during solution is called the wavefront. The wavefront is determined by the sequence in which the elements are arranged or ordered. The computer time required for solution is proportional to the square of the mean wavefront. Therefore, it is crucial to be able to minimize the wavefront. A separate module was developed to reorder elements and minimize the wavefront. Elements are reordered such that the element for which each degree of freedom is first mentioned is as close as possible in sequence to the element for which it is mentioned last.

Once the element sequence has been optimized, wavefront solution starts by scanning all elements to determine which element is the last to use each degree of freedom.

Solution proceeds by adding the equations for degrees of freedom related to the current element and which occurs for the first time. The equation for a degree of freedom that occurs for the last time is algebraically solved in terms of the remaining unknowns and eliminated from the assembled matrix by Gauss elimination. The equation is then written to a file for later back-substitution and the remaining equations are modified. The assembled matrix expands and contracts as degrees of freedom make their first and last appearance in the element definitions.

Another feature that has been developed for the solver is the ability to handle multi-point constraint equations. These equations specify relations between two or more degrees of freedom during solution. In addition, a direct matrix transformation method is used to impose coupling of degrees of freedom at the same node.

5.2.5 Line search

A line search algorithm [8] has also been implemented to speed up the convergence. In some situations, the use of the full $\mathbf{u}^{(i)}$ as obtained by the solver leads to solution instabilities. The line search algorithm attempts to improve the Newton-Raphson solution $\mathbf{u}^{(i)}$ by scaling the solution vector by a scalar value s termed the line search parameter. Thus, equation (5.24) can be rewritten in the form

$$^{t+\Delta t}\mathbf{u}^{(i)} = {}^{t+\Delta t}\mathbf{u}^{(i-1)} + s\mathbf{u}^{(i)} \qquad (5.51)$$

where

$$0.05 < s < 1.0 \qquad (5.52)$$

The line search parameter s is determined by minimizing the norm of the residual force vector through iterations.

5.3 CONTACT ANALYSIS

Simulation of many metal forming problems necessitates the ability to model the contact phenomena that occurs between the forming tools and the workpiece. Contact analysis is a complex problem because of the requirement to track the motion of the bodies involved and to accurately represent the friction between their surfaces. The numerical objectives are to detect the contact of the bodies and apply enough constraints and boundary conditions to simulate the frictional behavior, avoid penetration and allow for separation when necessary. Several numerical techniques have been developed to perform these objectives. These methods include the Lagrange multipliers method, the penalty function method and the direct constraint method.

5.3.1 Lagrange multipliers

The Lagrange multipliers method is one of the most common techniques to treat the contact problem in the literature [18]. In performing contact analysis, we are basically solving a constrained minimization problem where the constraint is the no penetration constraint. In this method, the finite element equilibrium equations are augmented using the constraint equations by introducing an array of Lagrange multipliers as additional degrees of freedom. In contact analysis, Lagrange multipliers signify the contact pressure.

Lagrange multipliers technique has often been implemented in contact procedures using special interface elements known as gap elements. If the constraints are properly written, this method ensures that penetration does not occur. Unfortunately, the introduction of Lagrange multipliers leads to numerical difficulties as their inclusion results in a nonpositive definite mathematical system. This requires additional operations with high computational costs. Another problem with this method is that there is no mass matrix associated with the Lagrange multipliers degrees of freedom. This results in a global mass matrix that cannot be inverted. This precludes the use of the Lagrange multipliers technique in explicit dynamic simulations. In addition, the use of interface elements requires a prior knowledge of where contact occurs and puts a restriction on the amount of relative motion that can occur. This may not be feasible in the simulation of many manufacturing problems.

5.3.2 Penalty function

The penalty function method is an alternative procedure to numerically implement the contact constraints in an approximate way. Using this approach, some penetration occurs with the amount being determined by the penalty constant or function. The penalty approach can be considered as analogous to a nonlinear spring between the two bodies. The penalty method is relatively easy to implement and has been extensively used in explicit dynamic analysis. However, the choice of the penalty constant has a detrimental effect on the numerical stability and accuracy of the method. It is also possible to combine the penalty method with the Lagrange multipliers method in a hybrid fashion. In the hybrid methods, the contact element is derived from a complimentary energy principle by introducing the continuity on the contact surface as a constraint and treating the contact forces as additional elements [8].

5.3.3 Direct constraint method

In this method, the motion of the contact bodies is tracked, and when contact occurs, direct constraints are applied on the bodies as boundary conditions. Both kinematic constraints on transformed degrees of freedom and nodal forces may be applied. This procedure can be very accurate if the program can predict when contact occurs. The direct constraint method is the method that has been implemented in the developed finite element program. No special interface elements are required and complex changing contact conditions can be simulated since no knowledge of where contact occurs is required prior to the analysis. The procedure has been implemented as a direct node-to-surface contact only for the case of contact between the workpiece and one or more rigid bodies [62].

The procedure starts by reading the input file that defines the perimeters of the tools as sets of geometrical entities connected together, namely straight line and arc segments. In each incremental load step, the contact algorithm tracks the motion of boundary nodes of the workpiece to detect any penetration into any of the tools. Upon detection of such penetration, the algorithm determines the magnitude and direction of the penetration of the nodes and applies them as corrective prescribed displacements to reposition the nodes on the tool surface. Once the nodes are on the tool surface, appropriate contact boundary conditions (e.g. sticking, slipping with friction, etc.) are applied on the contact nodes. Both Coulomb and shear friction laws can be applied. A node is considered in contact as long as the nodal force normal to the contact surface is compressive. If the normal force becomes tensile, the node is separated from the surface and its nodal forces are set to zero. In each step, iterations are performed until no change of contact conditions is detected.

The contact algorithm implementation as it stands now suffers from major drawbacks and causes many numerical problems. Among these drawbacks is that the corrective prescribed displacements are applied on the nodes that have penetrated the contact surface. This intermediate position does not satisfy the contact conditions and any corrective action based on this position will be approximate. Another approach to overcome this problem is to define a small tolerance region on both sides of the contact surface in which the nodes are assumed to lie on the surface when penetration is detected. If the penetration of the node is within the tolerance region, contact boundary conditions are applied to the node in the next load step without any corrective action. If a node penetrates the entire tolerance region in one load step, then a smaller load step size should be used and the computation is resumed based on the equilibrium position prior to penetration.

5.4 PROGRAM STRUCTURE

The developed ALE formulation has been implemented into a finite element computer program. The program has been designed with emphasis on modularity and use of standard Updated Lagrangian calculations whenever possible. The program is divided into subroutines each with a specific function while the main program consists mainly of subroutine call statements. As discussed in Chapters 2 and 3, the version of the ALE formulation developed in this work can be considered an extension to the Updated Lagrangian formulation. Thus, the first step in the computer development in this work was to implement an Updated Lagrangian finite element formulation and test it. All the necessary routines related to the nonlinear large deformation Lagrangian mass, stiffness, and stress calculations were developed. In addition, the necessary solution schemes for the static and dynamic matrix equations were implemented and tested. The next step was to implement the necessary routines for the new ALE formulation. These routines were comprised mainly of calculations for the additional mass and stiffness matrices as well as mesh motion routines. The last step of development was to implement a simple contact algorithm to enable the program to simulate a wider class of metal forming and large deformation applications. In the following, the main computer routines called in the main program and the main flow charts developed in each of the program building stages are highlighted.

5.4.1 Updated Lagrangian routines

Figure 5.3 shows a flow chart of the main program for the Lagrangian calculations. The flow chart only shows the main routines needed for the Updated Lagrangian code and the function of each routine is briefly stated. The additional subroutines that are called from the main routines are not listed here for brevity. The program starts by reading the control variables that control the size of the program arrays. These variables are needed beforehand for dynamic memory allocation purposes. Next, all variables and arrays used in the program are initialized. Problem input data, such as nodal coordinates, element connectivity, applied loads and material properties is then read and stored in the appropriate arrays. Distributed loads are then converted into their equivalent nodal values and the total applied loads are divided into incremental load steps.

Newton equilibrium iteration starts by setting the boundary conditions for the current load step and current iteration. The predictor phase of the time integration algorithm is first preformed and elements lumped mass matrices are calculated (for dynamic analyses only). Elements stiffness matrices are then calculated and stored on the hard disc for later assembly. The stiffness matrices in the Lagrangian formulation include both nonlinear material and large deformation effects. The contribution of all elements to the global effective load vector is assembled prior to solution. The global effective load vector also includes out of balance internal forces from previous iterations. The frontal solution scheme is then used to solve for the displacements. Assembly of elements equivalent stiffness occurs during the solution process. The line search algorithm is then called to accelerate convergence. The corrector phase of the time integration scheme is then performed in which velocities and accelerations are computed (for dynamic analyses only). Stress integration is then performed to calculate the stresses at the Gauss (integration) points. Convergence of iterations is then checked. Three measures are used to check the convergence. These are: iterative displacements, residual (out of balance) forces and residual energy. Iterations are performed until convergence is achieved.

At the end of each load increment, extrapolation of Gauss point stresses to obtain the nodal stresses takes place. Finally the program outputs the requested results for the solved load increment and moves on to the next step. The modular structure of the program is apparent from Figure 5.3.

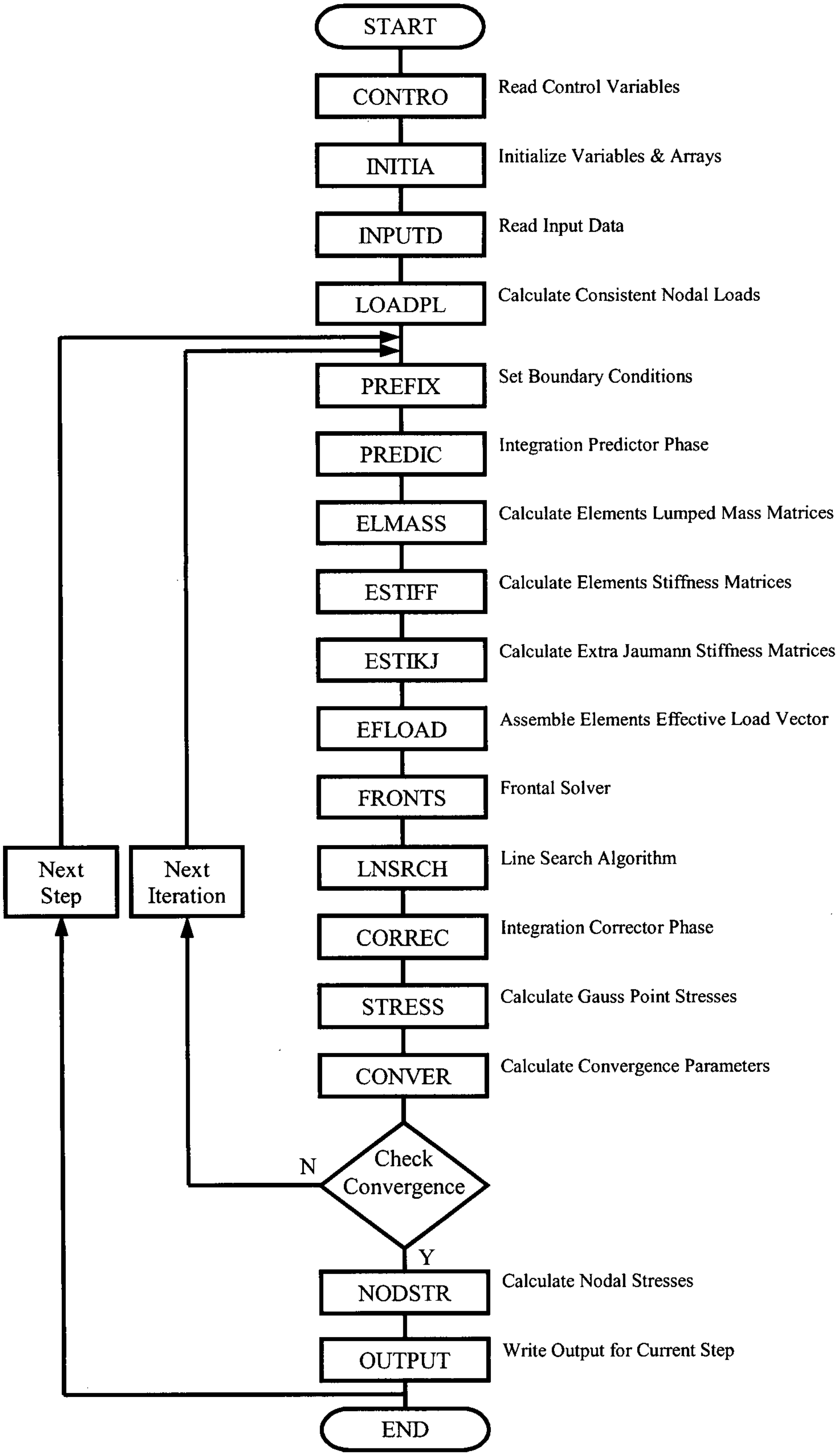

START
CONTRO
Read Control Variables
INITIA
Initialize Variables & Arrays
INPUTD
Read Input Data
LOADPL
Calculate Consistent Nodal Loads
PREFIX
Set Boundary Conditions
PREDIC
Integration Predictor Phase
ELMASS
Calculate Elements Lumped Mass Matrices
ESTIFF
Calculate Elements Stiffness Matrices
ESTIKJ
Calculate Extra Jaumann Stiffness Matrices
EFLOAD
Assemble Elements Effective Load Vector
FRONTS
Frontal Solver
LNSRCH
Line Search Algorithm
CORREC
Integration Corrector Phase
STRESS
Calculate Gauss Point Stresses
CONVER
Calculate Convergence Parameters
Check Convergence
N
Y
NODSTR
Calculate Nodal Stresses
OUTPUT
Write Output for Current Step
END
Next Step
Next Iteration

Figure 5.3. Flow chart of the Updated Lagrangian calculations.

5.4.2 ALE routines

The flow chart of the main program for the ALE calculations is shown in Figure 5.4. Comparing Figures 5.3 and 5.4, one can easily see the necessary ALE additions and changes to a Lagrangian code. One of the major changes to the Lagrangian code is the introduction of the mesh motion routines. The mesh motion routines decide how the mesh should be moved in the current iteration and choose the mesh motion parameters $\boldsymbol{\alpha}$ and $\mathbf{B}$ accordingly. The user may specify how the mesh should behave on some parts of the boundaries. Boundaries can be specified to be either pure Lagrangian or Lagrangian in the normal direction and ALE (or Eulerian) in the tangential direction. After the motion of the boundaries has been set for the current iteration, the transfinite mapping method determines how the interior regions of the mesh should be moved to preserve its uniformity.

Among the important additional ALE routines are those that calculate the extra stiffness and mass matrices and load vectors. Assembly of those extra terms takes place in the solver routines according to the ALE equilibrium equations in which the mesh motion parameters are introduced. A final ALE change is required to the stress calculations routines to compute the additional stress terms due to convective effects. The consistency and simplicity of the approach used in deriving and implementing this version of the ALE formulation are highly appreciated by highlighting its implementation advantages in comparison with previous implementations [39]:

- No updating of material properties is required between load steps since the stresses calculated within the iterations already include convective effects.

- No remeshing or boundary nodes adjustment takes place between increments. This is due to new mesh motion scheme on free material boundaries that allow nodes to be Lagrangian in the normal direction and to move freely in the tangential direction.
- Nodal stresses are typically calculated for postprocessing purposes only and are not involved in any equilibrium calculations. This is due to the new treatment of convective effects in the equilibrium equations.
- The modularity of the implementation also helped in retaining the simplicity of the program structure and allowed for clear identification of the necessary ALE changes to Lagrangian codes.

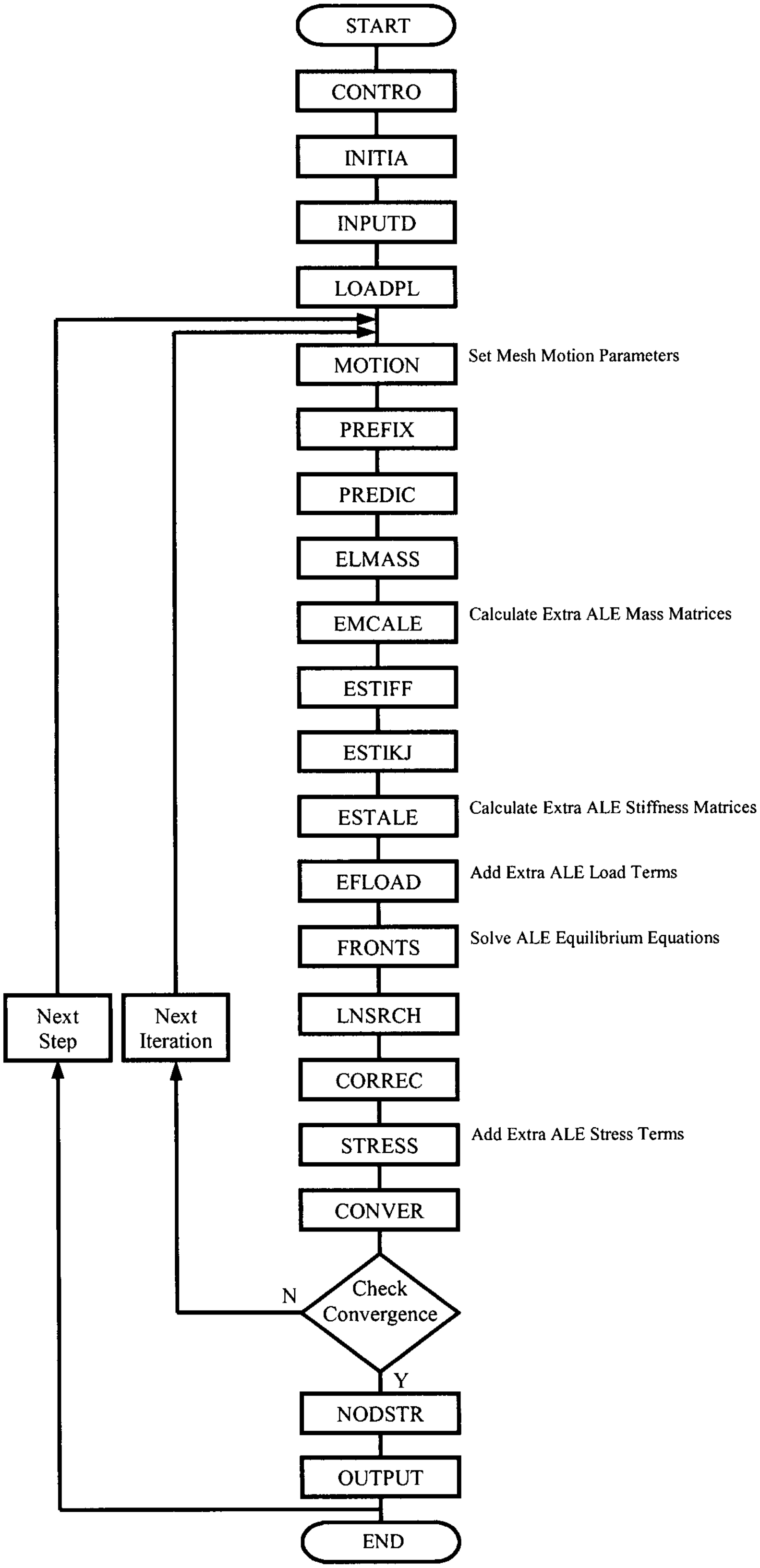
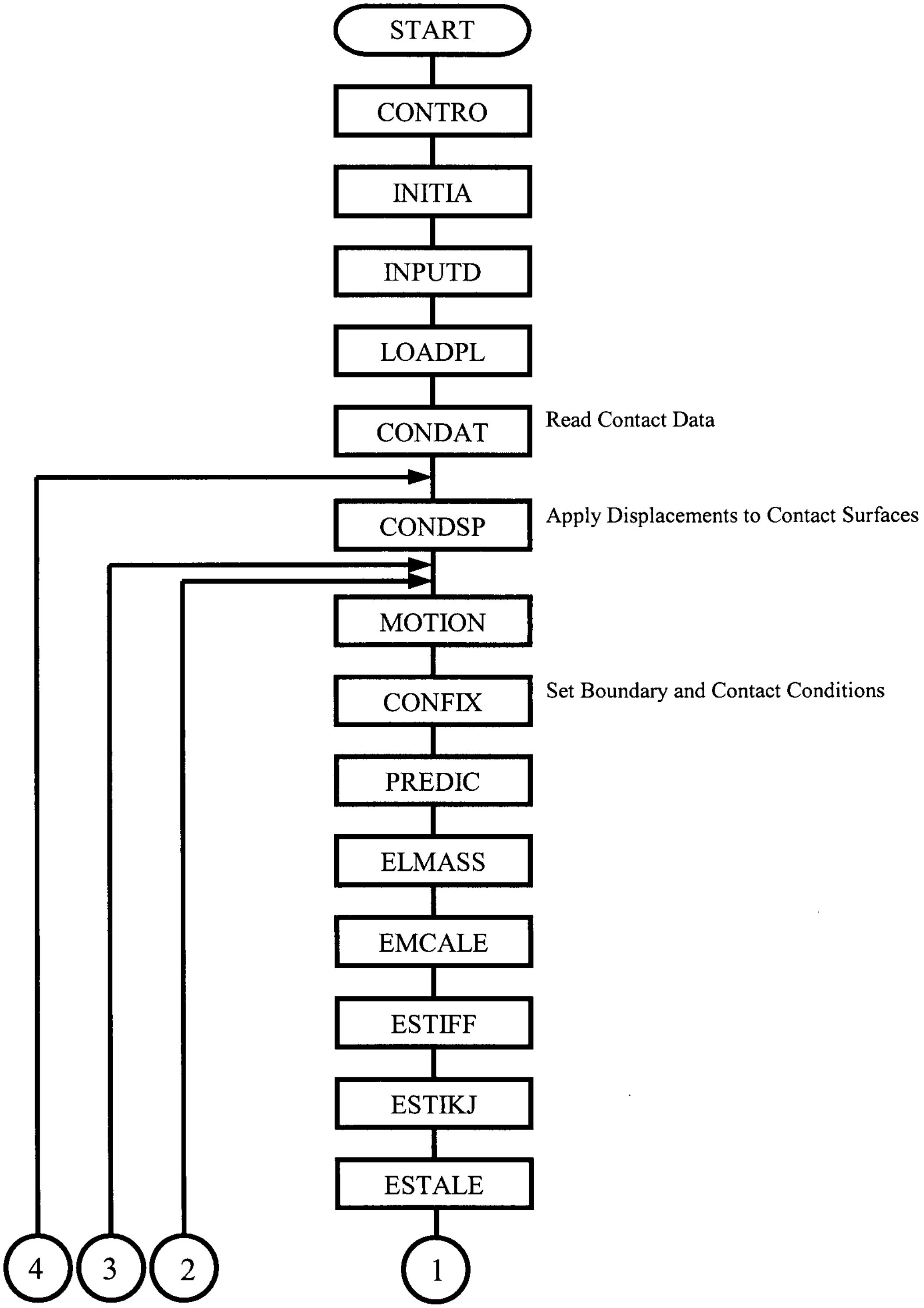


Figure 5.4. Flow chart of the ALE calculations.

5.4.3 Contact routines

As stated earlier, the last stage in the code development was to include contact analysis. Although ALE can alleviate the need for contact capabilities in some cases, however, contact analysis is essential in many metal forming and large deformation applications. In this work, contact analysis is based on the direct constraint method [62]. Figure 5.5 shows a flow chart of the ALE code with the contact routines included.

The first group of contact routines read the contact related data consisting of tool geometries, list of potential contact nodes on workpiece surface and list of nodes that are initially in contact. Another routine is called in the beginning of each load step to apply the necessary displacement boundary conditions on contact tools by updating the coordinates of the tools geometry. Within the contact and equilibrium iterations, a new routine is required to correctly set the contact-related and non-contact-related boundary conditions. After equilibrium iteration converges, the contact algorithm is called to check for any possible changes (penetrations or separations) in the contact conditions. If any changes are detected, corrective action is taken and another contact iteration is performed. This process is repeated until no change in contact condition is detected for the current load step after which a new load step is applied next. It is worth noting that the implemented contact algorithm suffers from convergence problems and future modifications are required.


START
CONTRO
INITIA
INPUTD
LOADPL
CONDAT
Read Contact Data
CONDSP
Apply Displacements to Contact Surfaces
MOTION
CONFIX
Set Boundary and Contact Conditions
PREDIC
ELMASS
EMCALE
ESTIFF
ESTIKJ
ESTALE
4
3
2
1

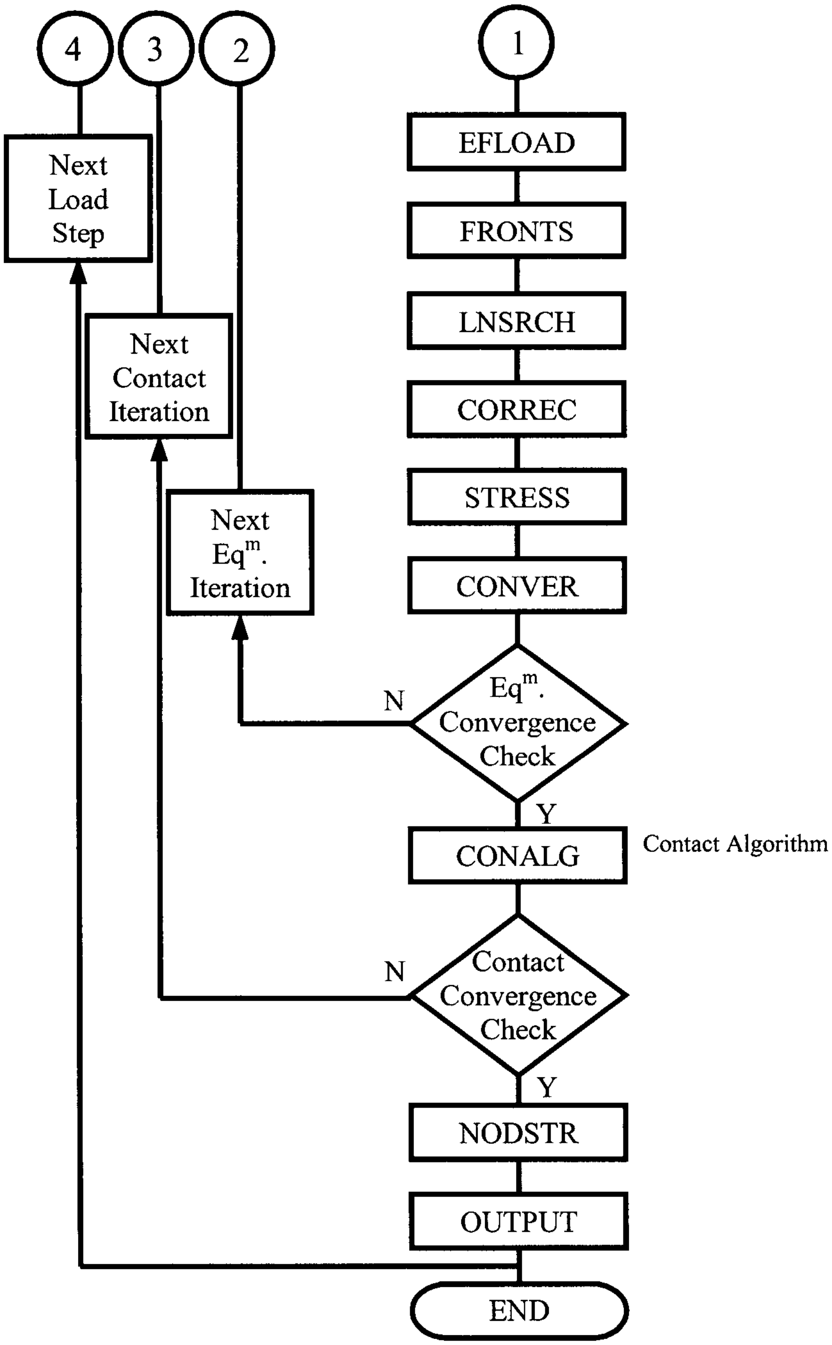


Figure 5.5. Flow chart with contact routines.

Chapter 6

APPLICATIONS

6.1 ONE-DIMENSIONAL STRESS WAVE PROPAGATION

The first application simulated using the developed finite element program is a wave propagation problem in a one-dimensional infinitely long elastic-plastic rod and is used to test the code with transient effects included. The same problem was reported in [33] and [37]. It should be noted that this problem doesn't require an ALE analysis and was selected because of the availability of analytical and numerical solutions. The infinitely long rod is discretized using 400 elements with a mesh size of 0.1 units as shown in Figure 6.1. The material properties of the rod are assumed to be: density ρ = 10000.0, Young's modulus E = 10000.0, plastic modulus E^p = 3333.33, yield stress σ_y = 75.0 and Poisson's ratio ν = 0.0. The rod is subjected to a compressive stress wave 100.0 in amplitude and 4.5 in width. The stress wave and the time interval under consideration are depicted in Figure 6.2. Any consistent system of units can be used to define the data for this problem.

Figures 6.3 and 6.4 compare the longitudinal stress distribution obtained using the presented ALE formulation with the analytical solution for both the elastic and elastic-plastic cases. ALE results are in good agreement with the analytical solution and with those of [33] and [37] (not shown here). Deviations from analytical solution are attributed to the difficulties associated with the numerical representation of step functions.

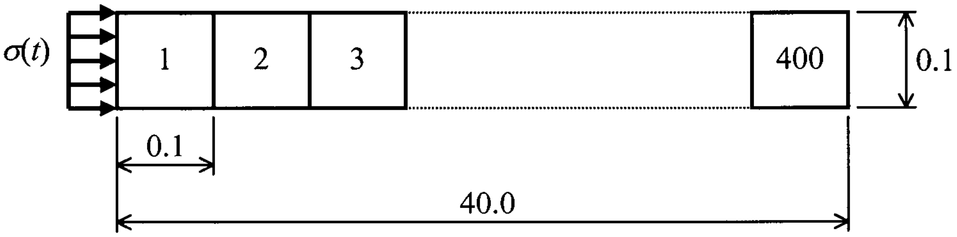


Figure 6.1. Mesh for the one-dimensional wave propagation problem.

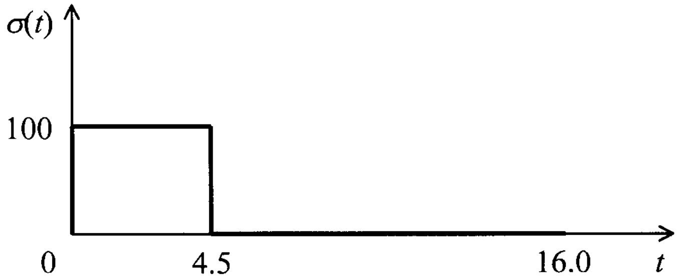


Figure 6.2. Stress wave amplitude and duration.

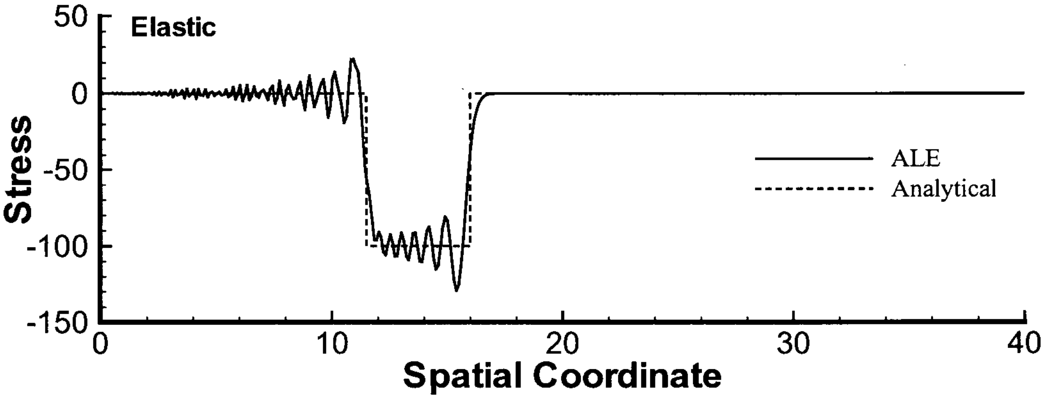


Figure 6.3. Longitudinal stress distribution comparison, elastic case.

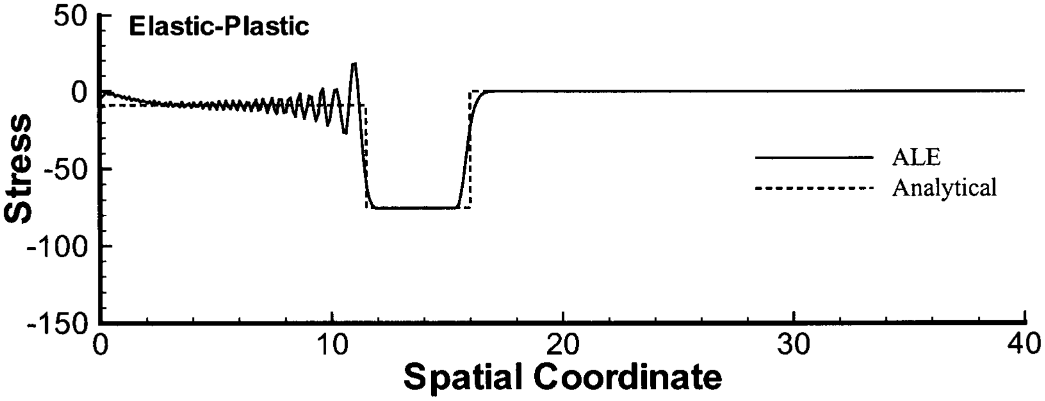


Figure 6.4. Longitudinal stress distribution comparison, elastic-plastic case.

6.2 BAR IMPACT

The bar impact problem has been investigated by many researchers [35, 73-75] and is considered a standard benchmark test for transient dynamic computer codes. In this problem, a cylindrical copper bar of initial radius 3.2 mm and length 32.4 mm strikes a rigid frictionless surface. The impact velocity is 227 m/s. The material is assumed to be elasto-plastic with Young's modulus E = 117 GPa, plastic modulus E^p = 100 MPa, Poisson's ratio $\nu = 0.35$, initial yield stress σ_y = 400 MPa, and density ρ = 8930 kg/m^3. A von Mises yield surface with linear isotropic hardening is assumed. Contact conditions are imposed by simply constraining the nodes in contact with the rigid surface. An axisymmetric mesh of 250 second order 8-noded elements is used. Computations are carried out up to a time of t = 80 μs.

Figure 6.5 compares the deformed shape and finite element mesh obtained using both the Lagrangian and the ALE formulations at different stages during the deformation process. In the ALE solution, boundary nodes are allowed to move in the tangential directions to the boundaries to maintain their uniform distribution while satisfying the ALE boundary constraint. The Lagrangian solution is obtained as a special case from the developed ALE formulation. It is clear that while the Lagrangian solution suffered severe mesh distortion, ALE was able to maintain a uniform mesh. It is worth mentioning, however, that the computation time for this problem using ALE is approximately twice that of the Lagrangian solution. This is mainly due to the fact that the ALE matrices are unsymmetric. Table 6.1 compares the final bar length and mushroom radius obtained using the developed code with numerical results obtained by other researchers. It is clear that the results obtained using the fully coupled implicit dynamic ALE formulation are in agreement with other well established numerical techniques and codes. Figure 6.6 shows the distribution of equivalent stress and equivalent plastic strain contours.

Table 6.1. Comparison of results for bar impact.

Reference (Code/Method)	Final length (mm)	Final mushroom radius (mm)
Kamoulakos [73]		
MARC	21.66	7.02
DYNA2D	21.47	7.13
DYNA3D	21.47	7.03
NIKE2D	21.47	7.07
Zhu & Cescotto [74]		
Lagrangian (different element types)	21.26 – 21.49	6.97 – 7.18
Camacho & Ortiz [75]		
Lagrangian (different remeshing schemes)	21.42 – 21.44	7.21 – 7.24
Liu et al. [35]		
Lagrangian	21.42	7.15
ALE (explicit)	21.53	6.87
Current Work		
Lagrangian	21.48	7.22
ALE (implicit)	21.69	6.90

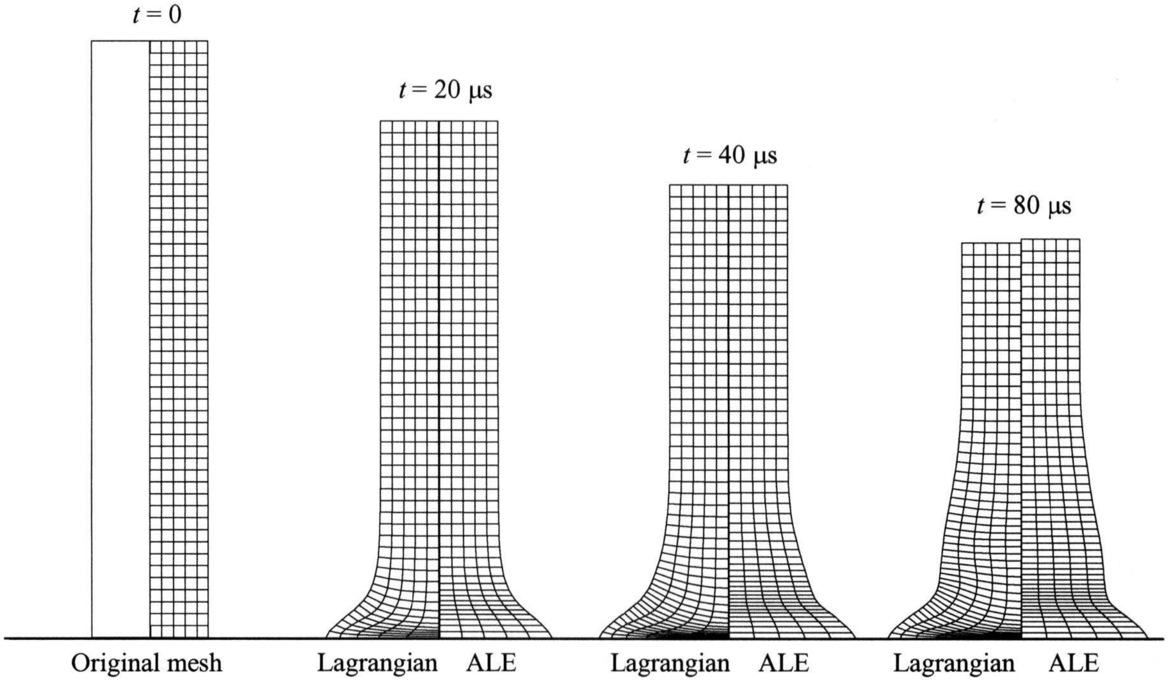


Figure 6.5. Comparison of the Lagrangian and ALE solutions for bar impact.

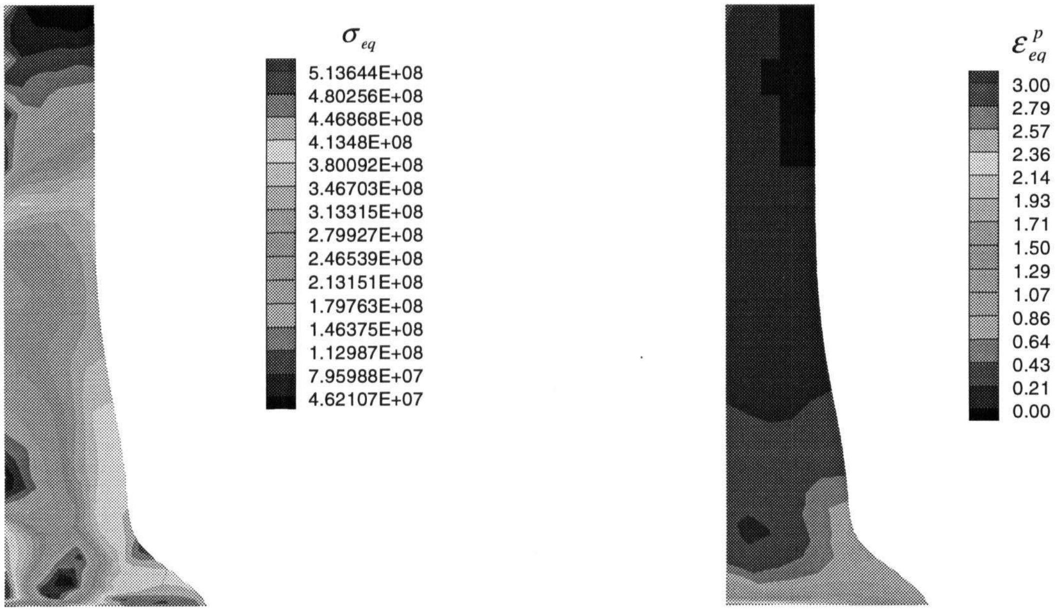


Figure 6.6. Equivalent stress and plastic strain contours for bar impact.

6.3 SHEET METAL EXTRUSION

Extrusion is a typical metal forming problem in which large strains are expected and in which remeshing and boundary condition updating are required if the traditional Lagrangian formulation is employed. In this problem, the extrusion die produces a 25% reduction in sheet thickness and is shaped in the form of a 5th order polynomial with zero curvature and slope at both ends. An aluminum billet of thickness $2a$ is forced through the die by a rigid piston pressing against the rear face of the billet and moving with a prescribed displacement. The length of the die is taken as $1.2a$. The same problem was previously solved using the Updated Lagrangian approach [76]. The material properties for the aluminum billet are taken as: Young's modulus = 10^4 ksi (68.95 GPa), Poisson's ratio = 0.3, initial yield stress = 57 ksi (393 MPa) and hardening parameter = 165 ksi (1.14 GPa). Figure 6.7 shows the initial geometry and mesh used in the simulation. Because of symmetry, only the upper half of the billet was analyzed.

Using the ALE formulation, all the nodes confined to the die area are set to be Eulerian during the course of deformation. The nodes on the boundaries are set as Lagrangian. The motion of all the other nodes is controlled by the ALE mesh motion scheme. Figure 6.8 shows the plastic strain distribution and the deformed shape after a piston displacement of $2a$ units. Contact between the billet and the die was set as boundary constraint equations. The ALE approach was able to eliminate the need for remeshing or boundary condition updating and the desired deformation level was reached without any user intervention or special contact treatments. Variations in the longitudinal stress component in the die region at different lateral positions from the mid-plane of the initial configuration are shown in Figure 6.9. The obtained stresses are in good agreement with those previously published [76].

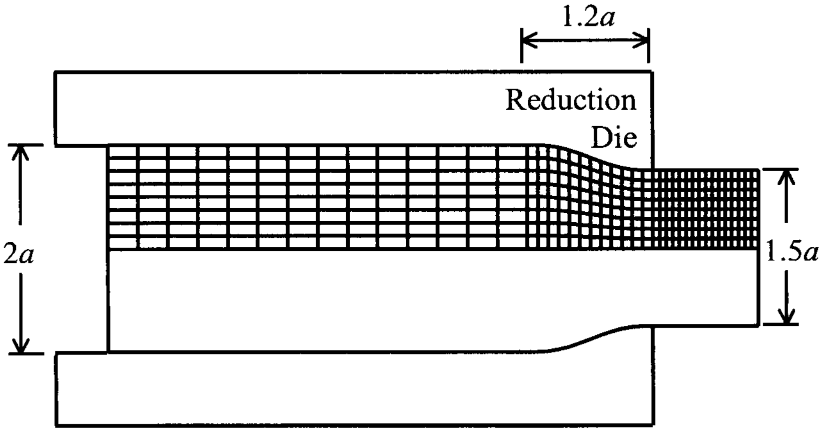


Figure 6.7. Geometry and mesh for extrusion process.

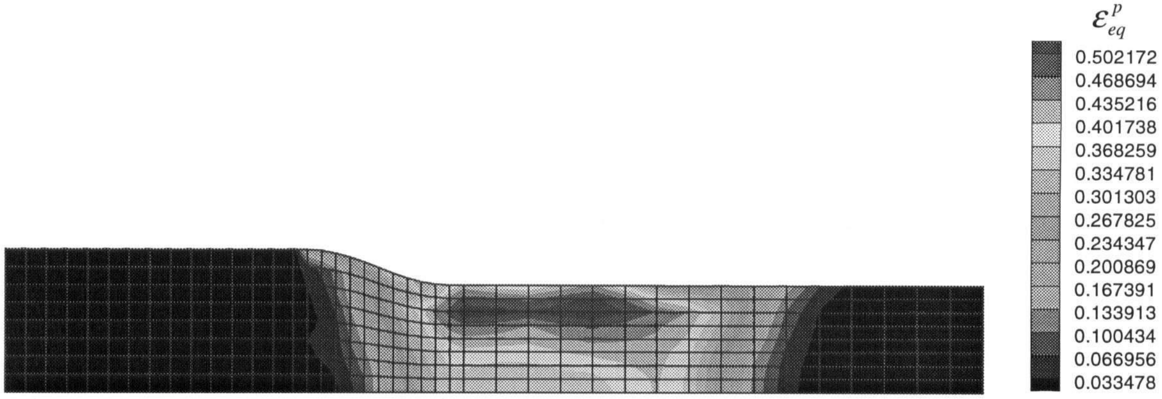


Figure 6.8. Plastic strain contours and deformed shape for extrusion problem.

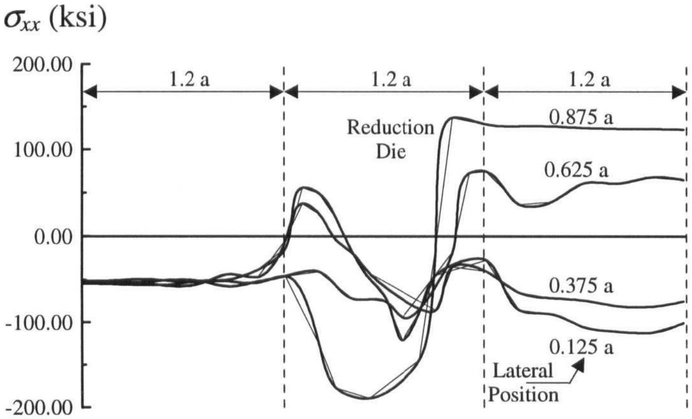


Figure 6.9. Longitudinal stress at different lateral positions.

6.4 QUASI-STATIC AND DYNAMIC COINING

In this example, a coining process, also known as punch indentation, is simulated to show the effectiveness of the ALE formulation in handling contact boundary conditions and in preventing mesh distortion. This process is simulated using both quasi-static and dynamic analyses to investigate the significance of dynamic effects. The problem is also solved using ANSYS [8] for comparison. Figure 6.10 shows the geometry and initial mesh of the body. The workpiece is placed between two rigid frictionless tools moving with constant velocity under plane strain conditions. Because of symmetry, only one quarter of the domain is modeled. The deformation process is continued up to a 60% reduction of the original workpiece height. The material is assumed to be elastic-plastic with a Young's modulus of 200 GPa, a Poisson's ratio of 0.3, an initial yield stress of 250 MPa and a hardening parameter of 1 GPa.

In the Lagrangian ANSYS simulation, it is noted that applying the downward motion of the punch as prescribed boundary conditions on the workpiece nodes initially in contact with punch would effectively cause the punch size to increase during the course of deformation as shown in Figure 6.11. When the Lagrangian formulation is employed, non-material-associated boundary conditions, such as the contact between the punch and the workpiece, is best handled using contact elements. However, contact elements are difficult to handle and may cause difficulties in achieving convergence. ANSYS, by default, uses a mixed penalty + Lagrange multipliers method in the formulation of its surface to surface contact elements. Two main parameters are used to define the characteristics of these elements. The first is the normal contact stiffness factor in the range of 0.001 to 100.0, with a default of 1.0. A smaller value of the normal contact stiffness factor provides for easier convergence but more penetration. The second parameter is a tolerance factor that determines if penetration compatibility is satisfied. Penetration compatibility is satisfied if penetration is within a clearance of the value of tolerance factor times the depth of the underlying element. The range of this factor is less than 1.0, with a default of 0.1.

Figure 6.12 shows the deformed shape of workpiece at the final stage of the process using the default settings of the contact elements in ANSYS. It is noticed that large penetration of the punch inside the workpiece was allowed when using the default values of the normal contact stiffness and tolerance parameters. Figure 6.13 shows the deformed shape when a contact stiffness of 100.0 and a tolerance of 0.00001 are used. It is noticed that the penetration is quite smaller in this case, though not completely eliminated. Reducing the tolerance parameter beyond this value did not affect results significantly. Convergence was more difficult to achieve in the case of these extreme contact parameters.

Figure 6.14 shows the evolution of the deformed shape obtained using the developed ALE formulation. The figure shows that the contact condition between the punch and the workpiece is accurately satisfied. This was easily achieved by allowing the degrees of freedom of the nodes directly under the punch to be Lagrangian in the vertical direction (to satisfy the boundary constraint) and Eulerian in the horizontal direction (to maintain the same punch size under deformation). No special contact algorithm was needed to handle the contact conditions. In addition, the transfinite mapping method was able to maintain a homogeneous mesh throughout the deformation history. In this analysis, the 60% reduction in height is solved using 750 incremental steps. Table 6.2 compares the stresses at the punch corner after the first load step of the total 750 steps of the analysis in the three cases: (a) ANSYS with default contact parameters; (b) ANSYS with extreme contact parameters; (c) ALE. It is clear that the stresses obtained by defining extreme contact parameters in ANSYS match the ALE analysis. However, stresses obtained by using the default values for the contact parameters, which provide for easier convergence, are inaccurate even at such early stages of the analysis.

Table 6.2 Comparison of stresses at punch corner after the first load step.
(a) ANSYS with default contact parameters.
(b) ANSYS with extreme contact parameters.
(c) Present ALE formulation.

Case	S_{xx} (MPa)	S_{yy} (MPa)	S_{xy} (MPa)	S_{zz} (MPa)
(a)	-3.05	-39.50	9.31	-12.76
(b)	-70.12	-234.18	23.35	-91.30
(c)	-70.43	-234.76	23.65	-91.56

Figure 6.15 compares the load-displacement curves obtained by ANSYS and the developed ALE formulation. It is clear that the erroneous load fluctuations associated with contact analysis in the Lagrangian solution can be avoided when using ALE.

The significance of dynamic effects is investigated by examining the indentation problem at different punch velocities. Figure 6.16 compares the final deformed shape and equivalent plastic strain distributions for the quasi-static case and for four different punch velocities. Dynamic effects are not significant for punch velocities less than 1 m/s since the deformed shape and the plastic strain distribution are quite similar to those of the quasi-static simulation. One can also notice that low punch velocities allow the workpiece to flow horizontally away from the punch, while at high velocities the workpiece tends to back extrude.

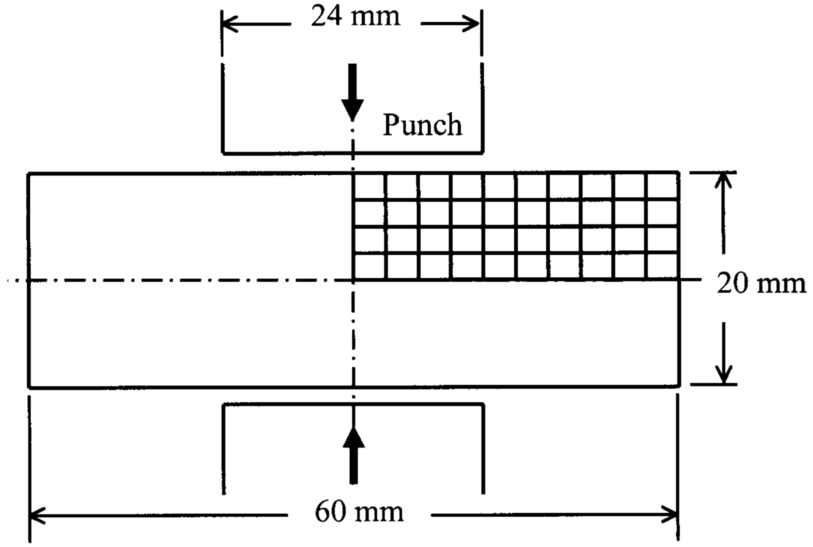


Figure 6.10. Geometry and initial mesh for coining process.

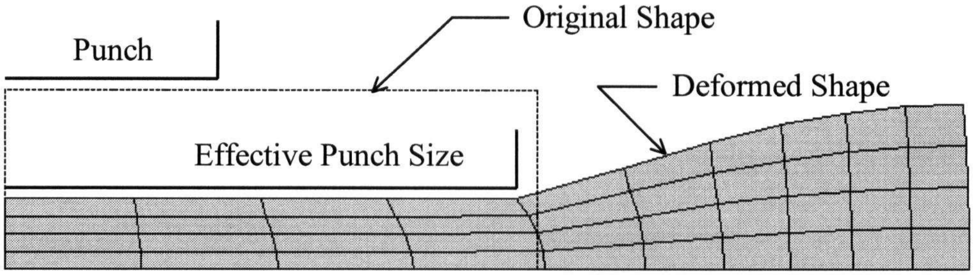


Figure 6.11. ANSYS solution with no contact elements.

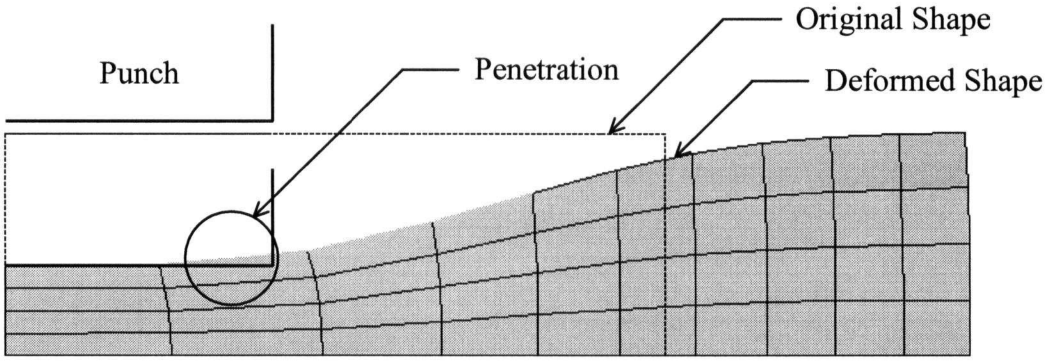


Figure 6.12. ANSYS solution with default contact parameters.

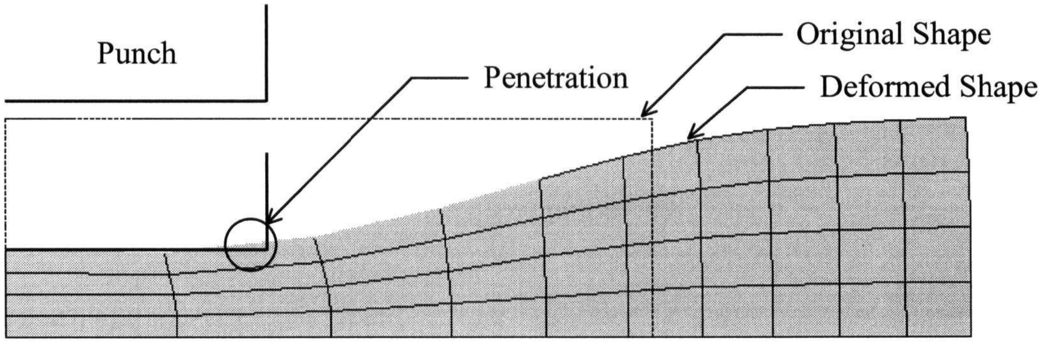


Figure 6.13. ANSYS solution with extreme contact parameters.

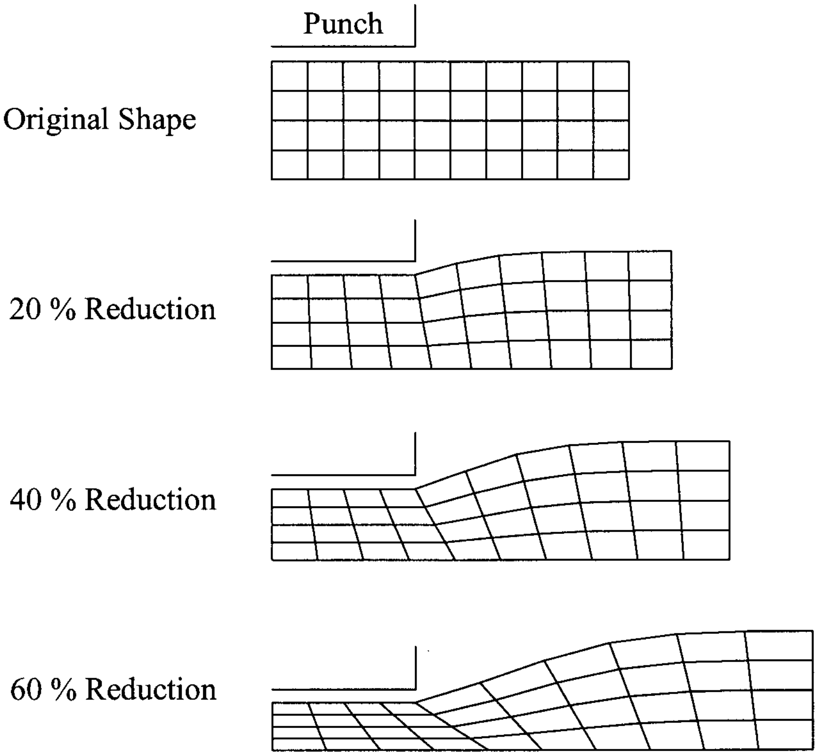


Figure 6.14. Evolution of the deformed shape during the coining process.

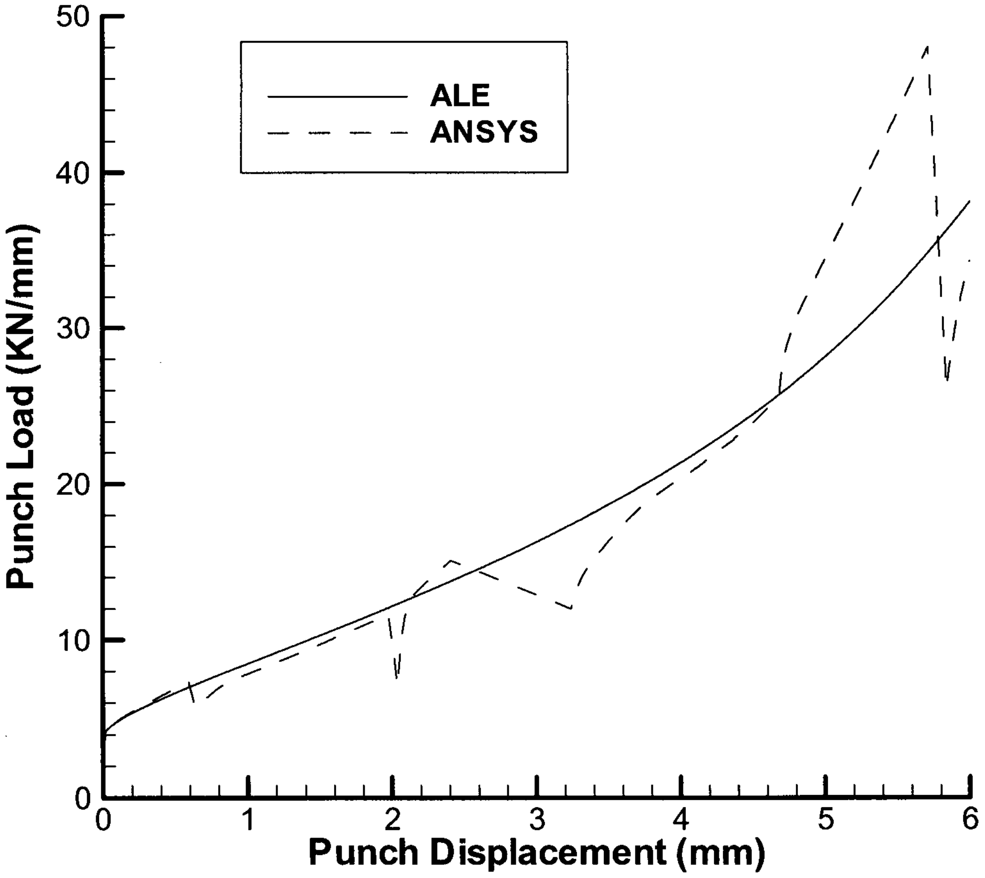


Figure 6.15. Punch load-displacement curve comparison.

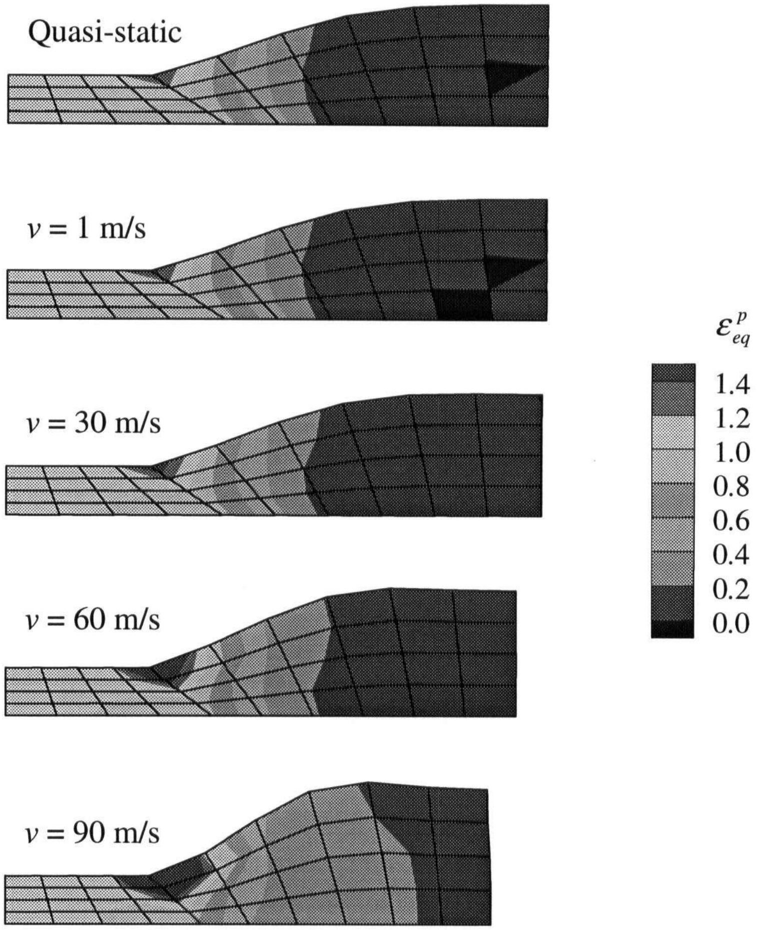


Figure 6.16. Final plastic strain distribution at different punch velocities.

6.5 V-BENDING

Although sheet metal bending processes, such as the V-bending process, might seem to be simple processes, precise numerical simulation using elastic-plastic finite elements with contact analysis is not an easy task. This example serves two purposes. The first objective is to elaborate on the power of ALE in overcoming the numerical difficulties associated with the simulation of metal forming processes with corner contact. The second objective is the ability to perform simple experiments for this type of process and thus verify the ALE predictions against experimental results [77].

Figure 6.17 shows an isometric sketch for the setting of the V-bending experiment. The punch and die are made of mild steel 1018 while the sheet material is Aluminum 5052-H32 with a Young's modulus of 70 GPa, Poisson's ratio of 0.33, yield strength of 195 MPa, tensile strength of 230 MPa and break elongation of 12 %. The sheet metal is 114.3 mm (4½ in.) long, 63.5 mm (2½ in.) wide and 6.35 mm (¼ in.) thick. The punch and die both have 90^o V angles and the punch has a nose radius of 9.525 mm (0.375 in.). The bending experiment was conducted using a Tinius Olsen UTM at a constant downward speed of 0.2 in/min. Figure 6.18 shows the actual experimental setup.

The V-bending process is simulated using the developed ALE code as well as using the Lagrangian FEA program MARC [18]. Accurate treatment of the tool-workpiece contact constitutes a major problem for simulations based on the Lagrangian formulation. Huang and his coworkers [78, 79] showed that it is very difficult to numerically satisfy the sharp corner contact condition that occurs between the die edge and the workpiece without penetration. The material behavior used in the ALE simulation is assumed to be bilinear elastic-plastic with kinematic work hardening. Figure 6.19 shows the finite element mesh used in the simulation.

Figures 6.20, 6.21 and 6.22 show the development of the deformed shape for the V-bending process at different stages obtained using the experiment, MARC and ALE simulations, respectively. Figure 6.23 shows the large corner penetration that occurs at an intermediate stage as obtained by MARC even though a relatively fine mesh was used. Unlike ANSYS, in which surface to surface contact can be defined, the contact algorithm in MARC is based on node to surface type of contact. In the ALE simulation, contact analysis is only introduced between the punch and the sheet. The sharp die corner contact with the sheet was handled in ALE by using Eulerian degrees of freedom for the node at the die corner [77]. Figure 6.24 compares the load-deflection curves obtained from the experiment with the simulations. It is clear from the figure that the sharp corner contact in MARC produces large fluctuations in the load prediction. It is also clear that the ALE predictions are smoother. The fluctuations in the ALE predictions are attributed to the contact analysis between the nose of the punch and the sheet that ALE could not eliminate.

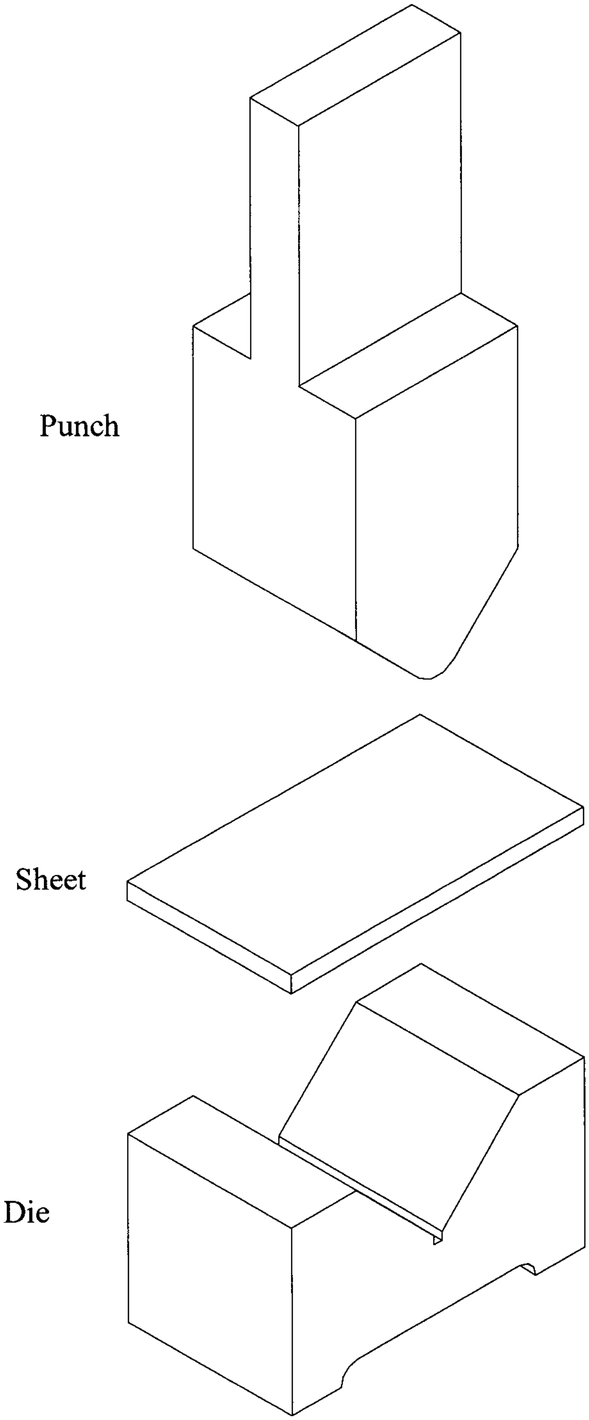


Figure 6.17. Isometric sketch for the setting of the V-bending experiment.

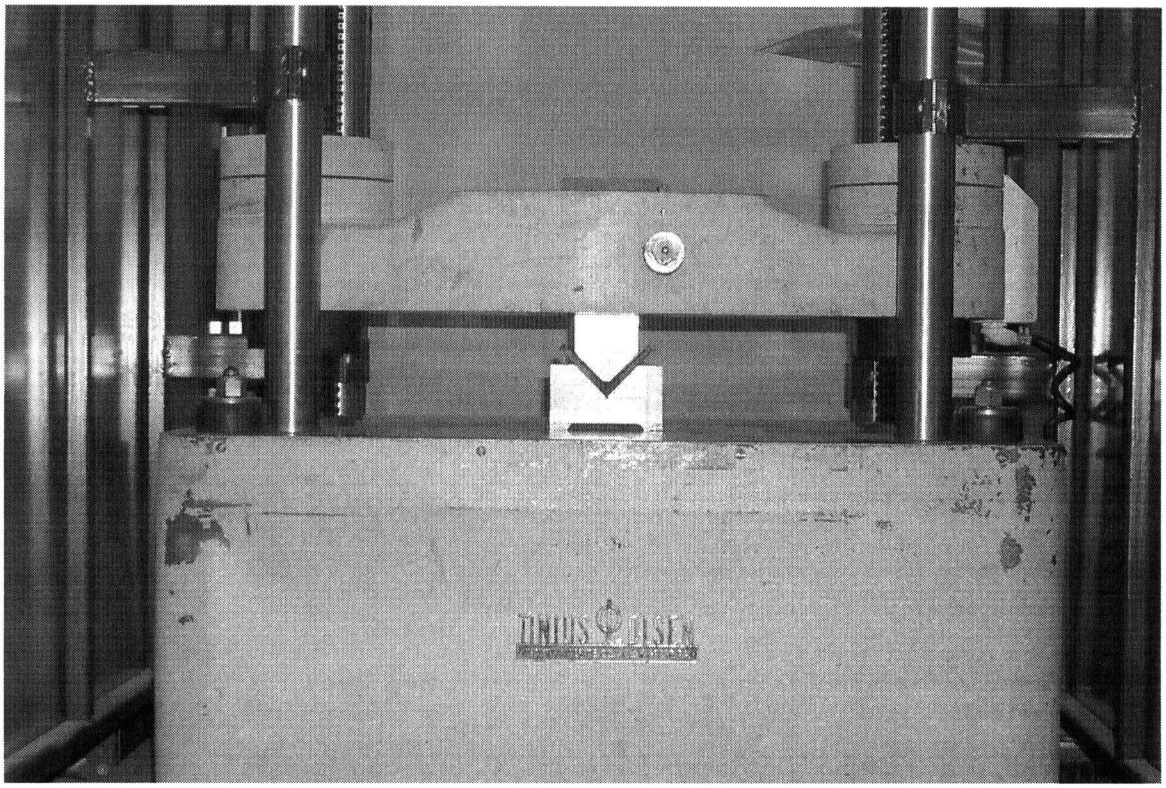


Figure 6.18. Tinus Olsen UTM used in the V-bending experiment.

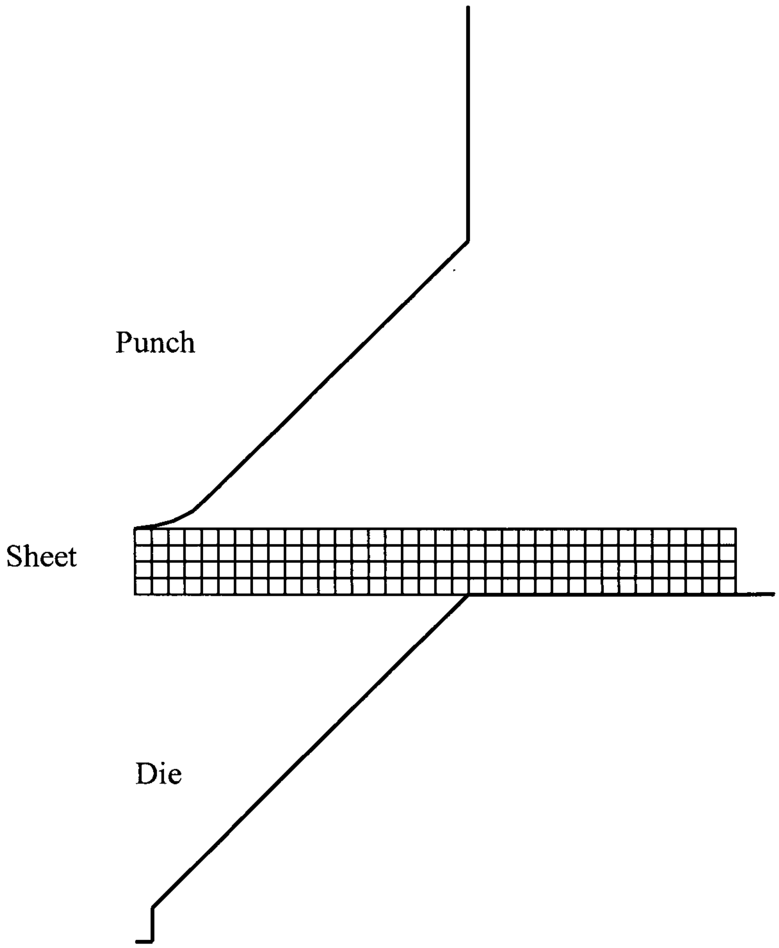


Figure 6.19. Finite element mesh used in the simulation of the V-bending process.

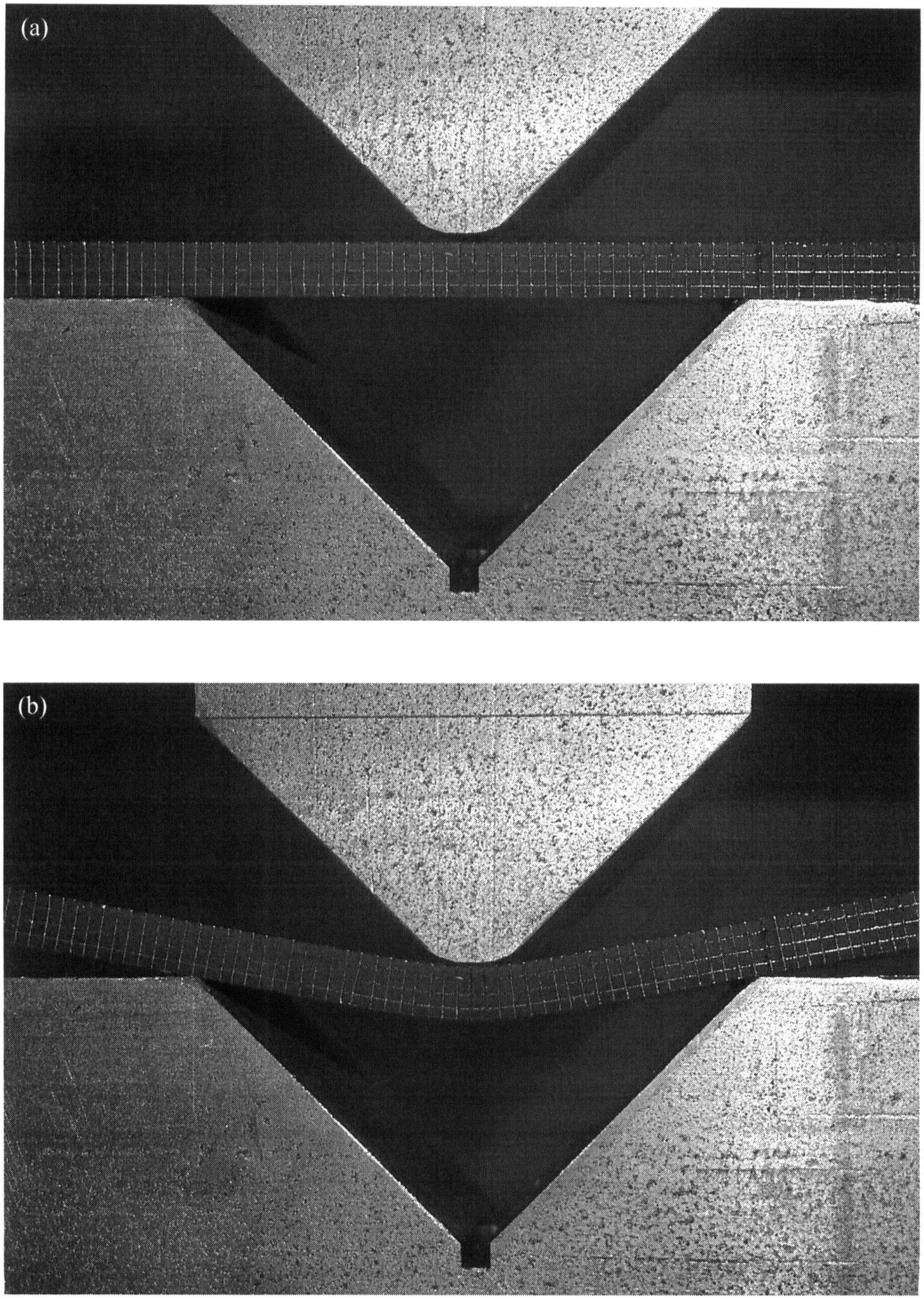


Figure 6.20 (a-b). Development of the deformed shape during the experiment.

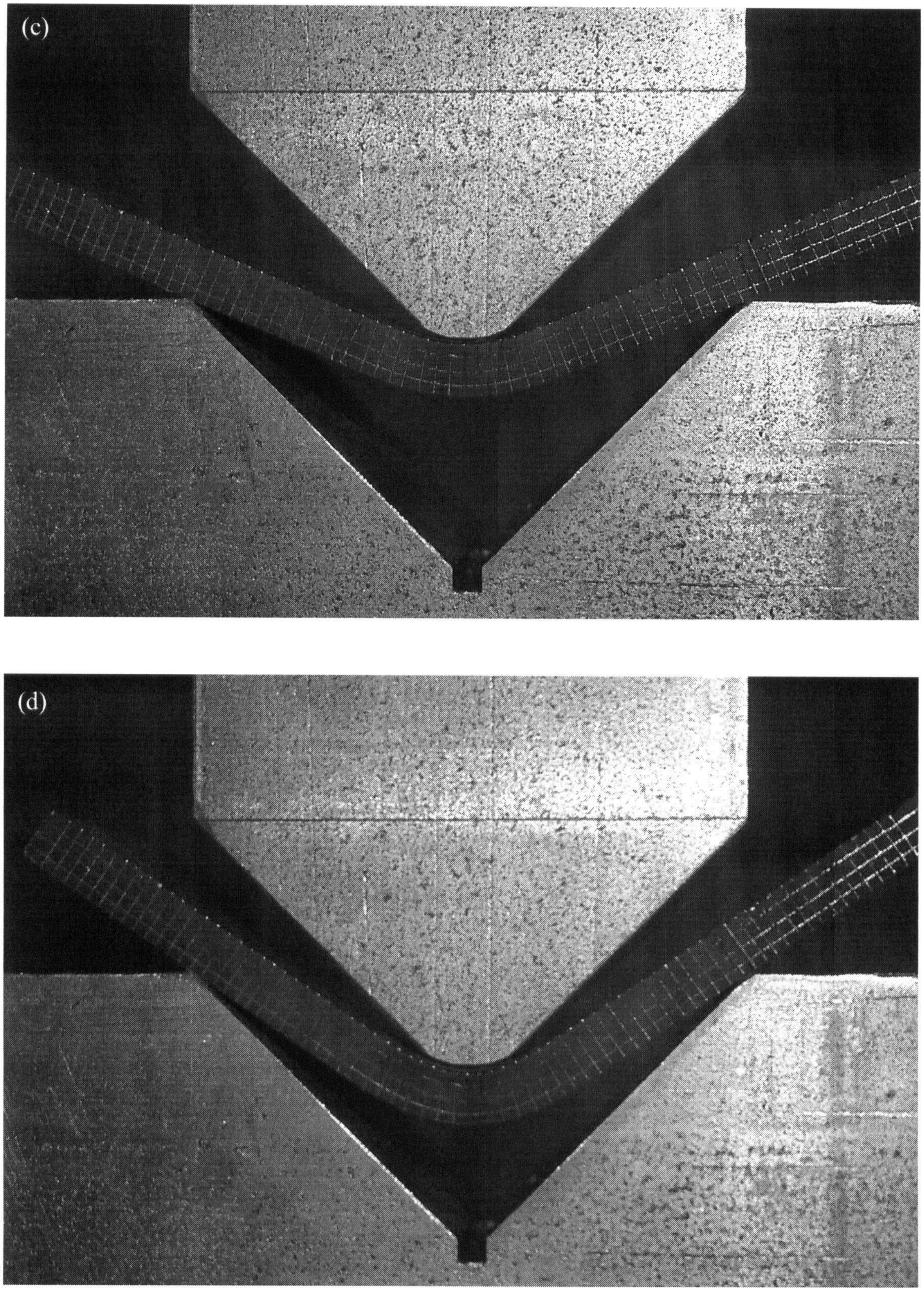


Figure 6.20 (c-d). Development of the deformed shape during the experiment.

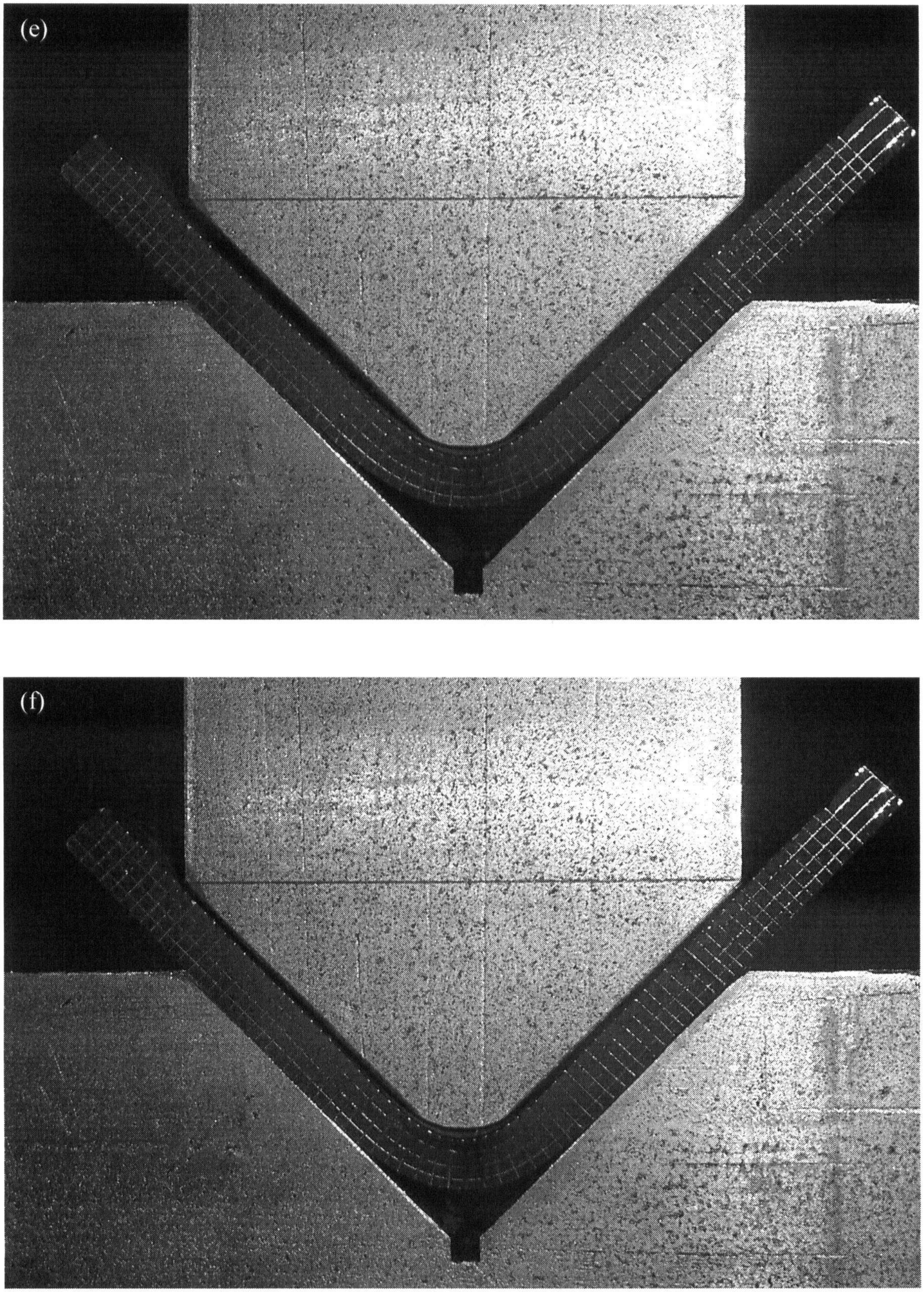


Figure 6.20 (e-f). Development of the deformed shape during the experiment.

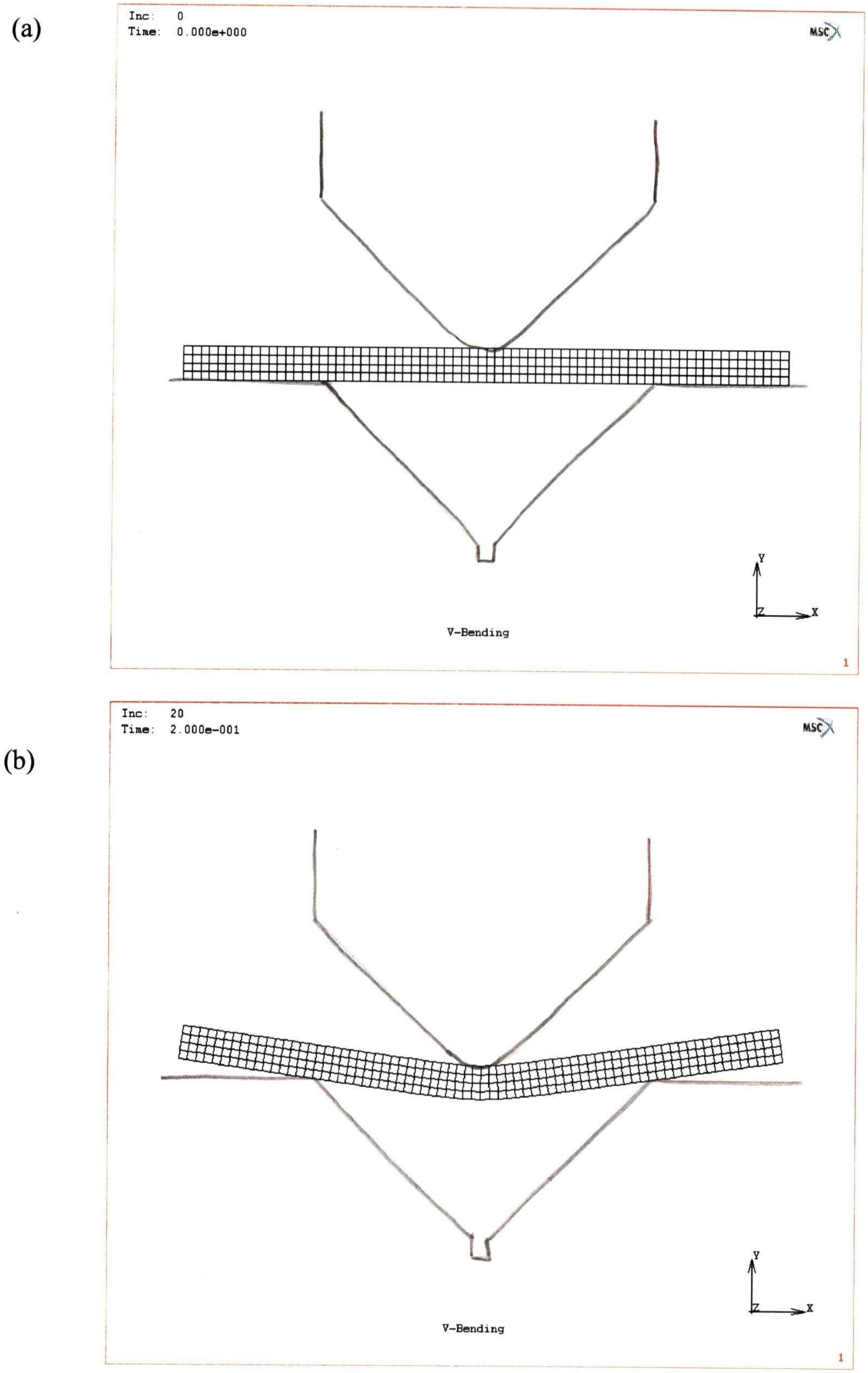


Figure 6.21 (a-b). Development of the deformed shape using MARC.

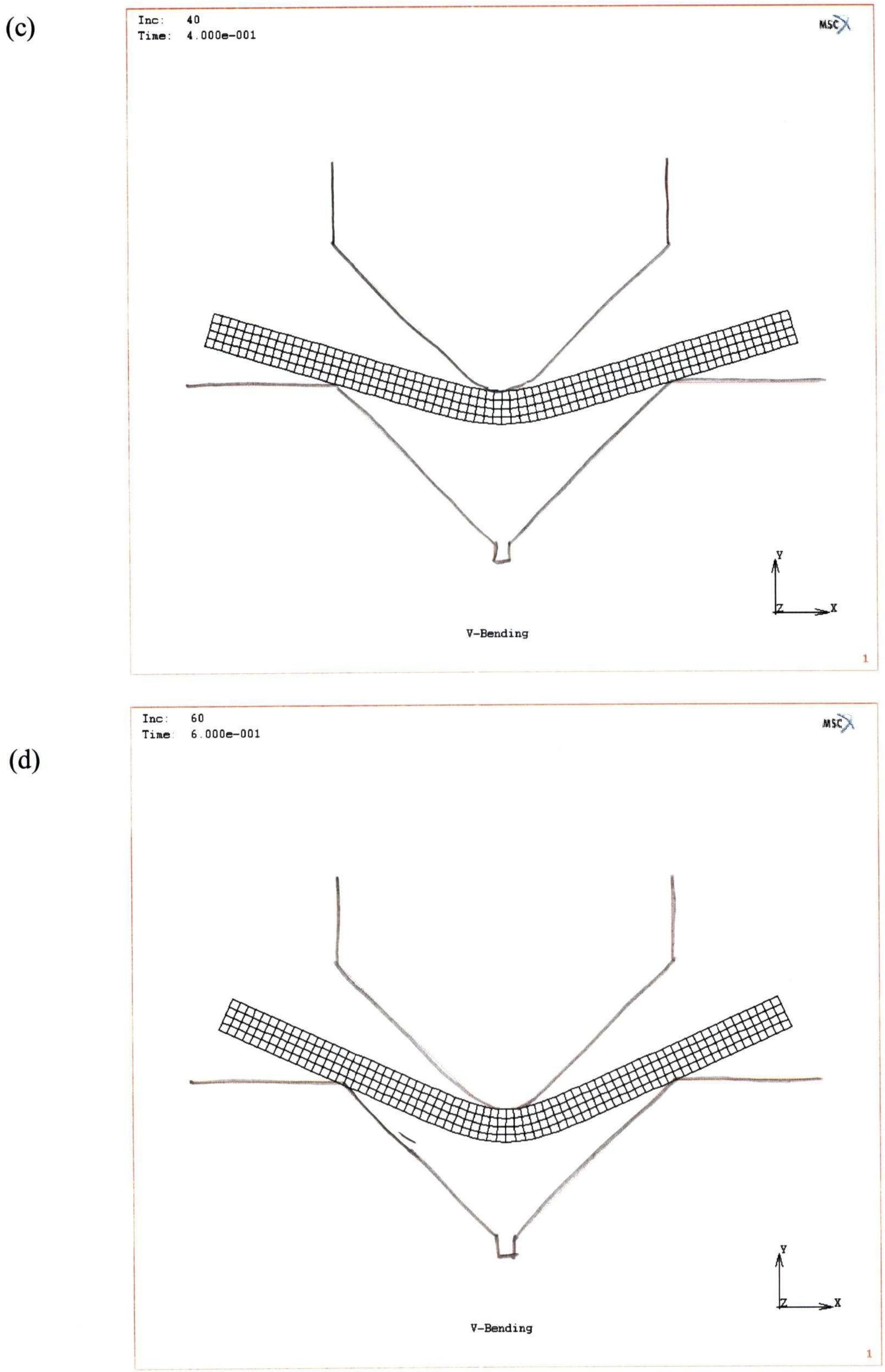
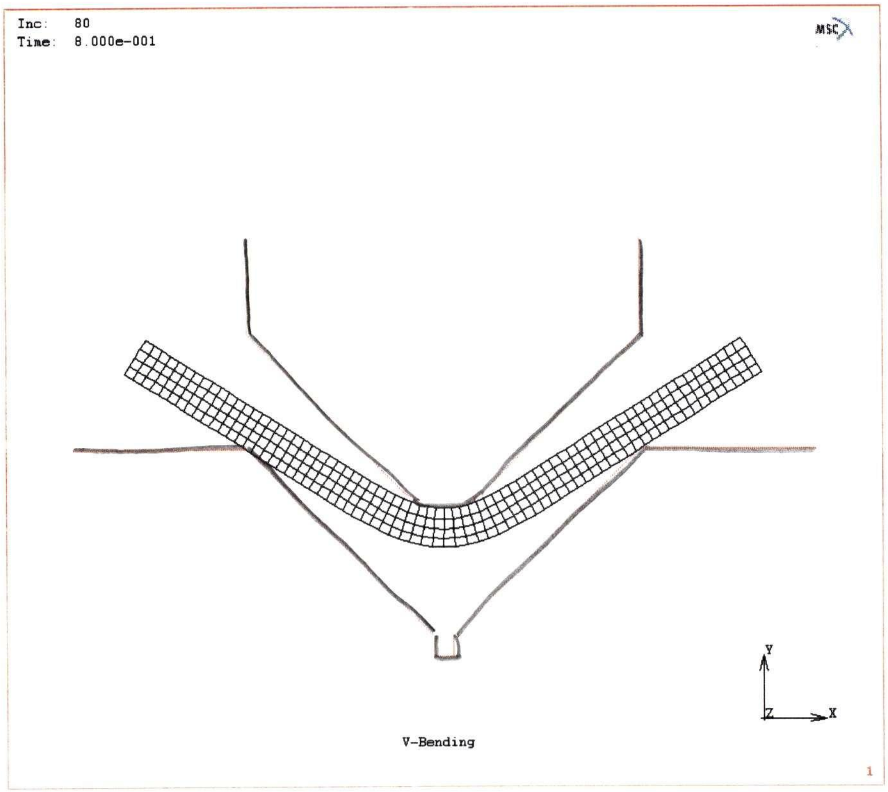


Figure 6.21 (c-d). Development of the deformed shape using MARC.

(e)



(f)

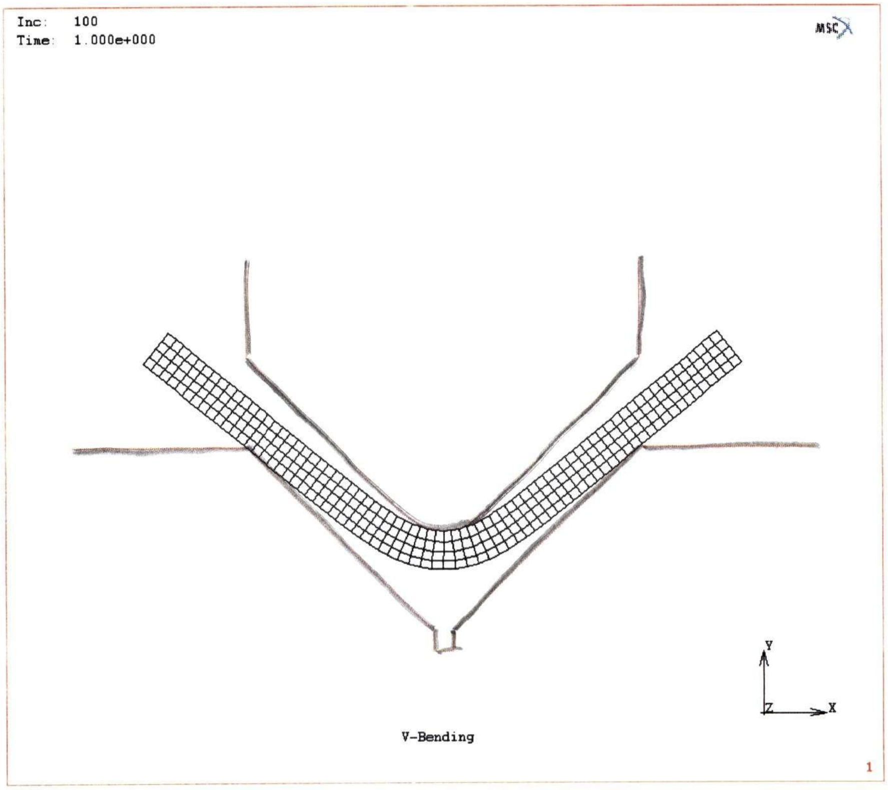


Figure 6.21 (e-f). Development of the deformed shape using MARC.

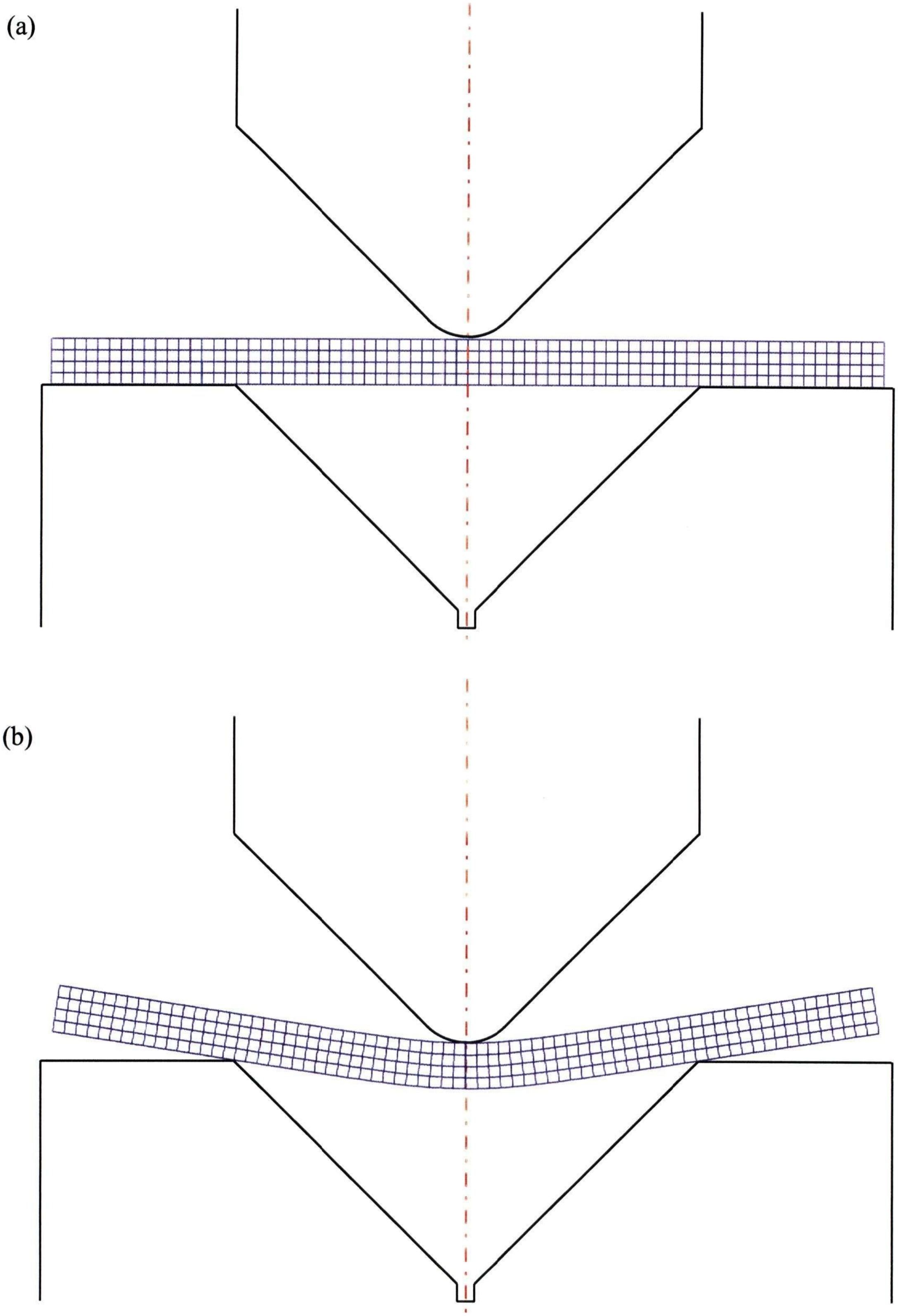


Figure 6.22 (a-b). Development of the deformed shape using ALE.

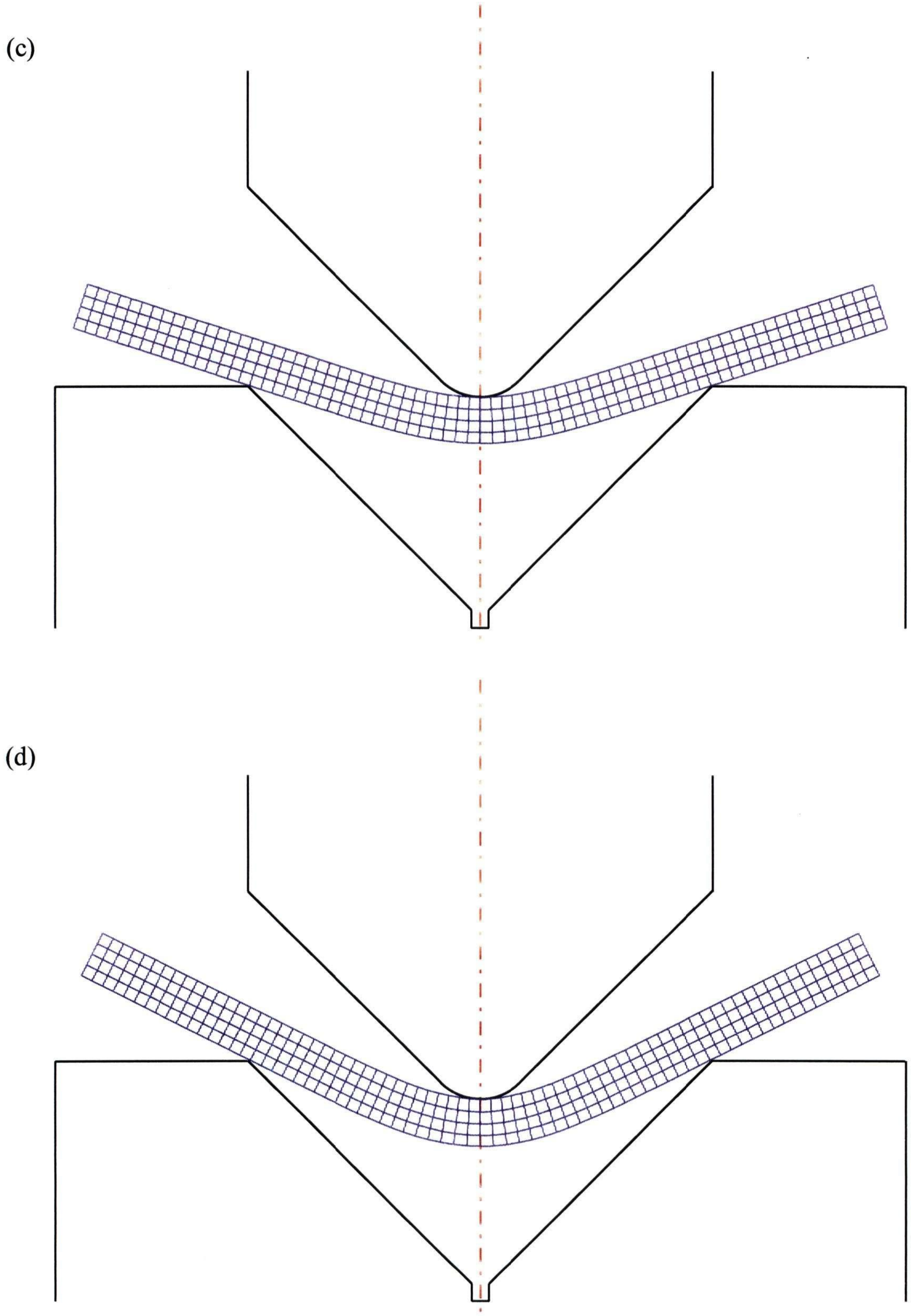


Figure 6.22 (c-d). Development of the deformed shape using ALE.

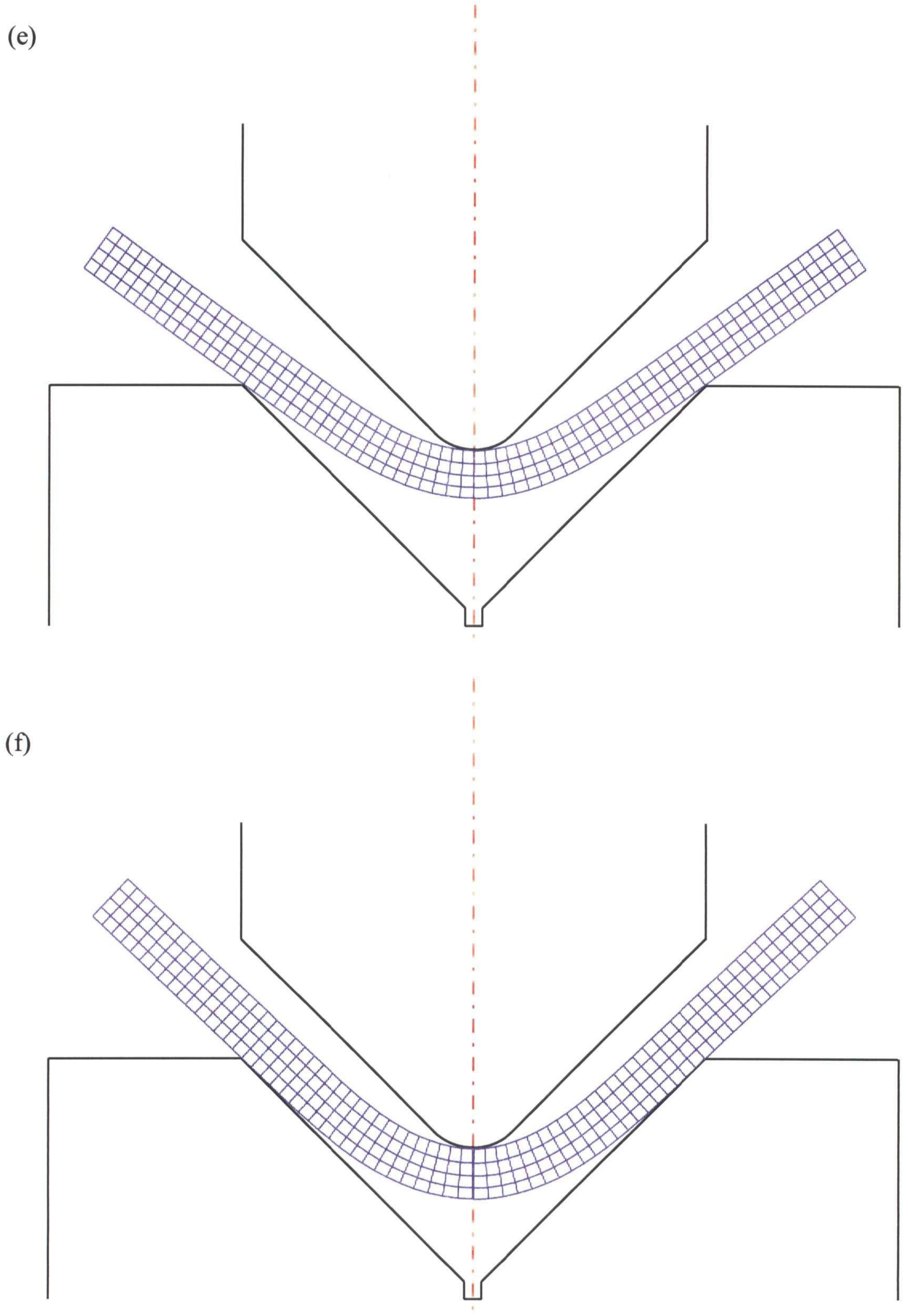


Figure 6.22 (e-f). Development of the deformed shape using ALE.

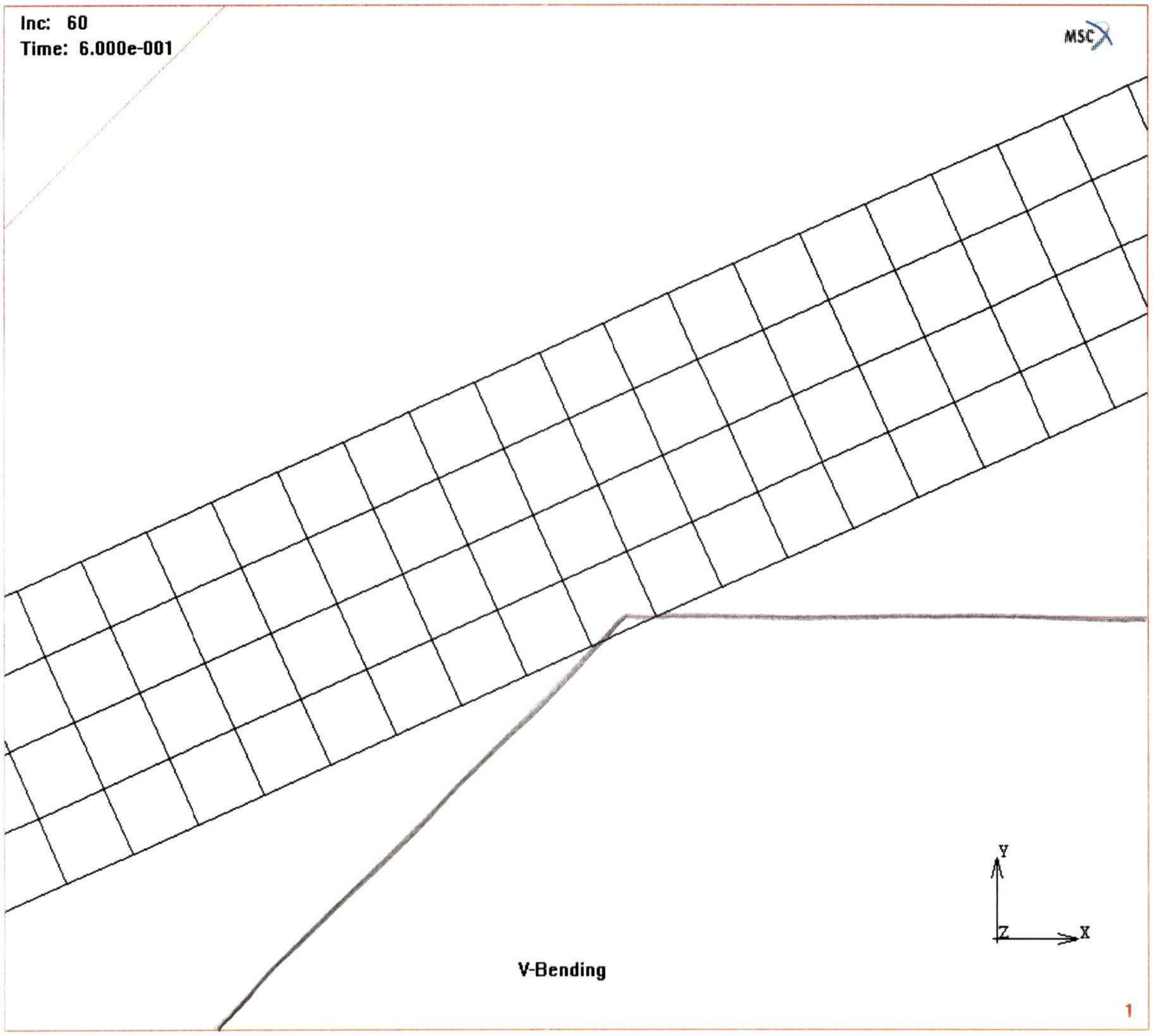


Figure 6.23. Deformed shape using MARC showing corner penetration.

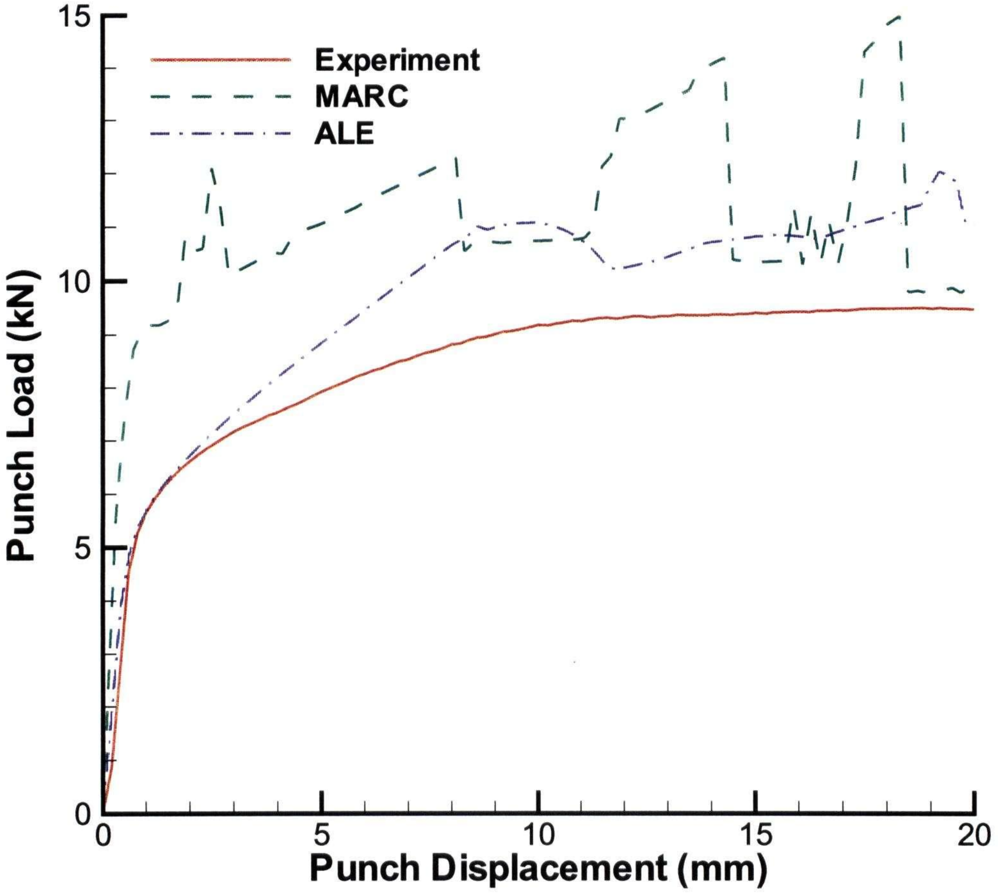


Figure 6.24. Comparison of load displacement curves for V-bending.

6.6 PLATE INDENTATION

Local loading of circular steel plates resting on a ring support is simulated using ALE and compared to experimental results [80]. The objective of this example is to compare the plate response when loaded using flat-faced and hemispherically-tipped indenters. In the case of the flat-faced indenter, ALE can easily handle the sharp corner contact condition whereas the contact algorithm will handle the case of the hemispherically-tipped indenter. The plate is 6 mm thick, 300 mm in diameter and is simply supported at a diameter of 248 mm. It is transversely loaded at its center at a constant rate of 3 mm/min using flat-faced as well as hemispherically-tipped cylindrical indenters of 12.7 mm diameter. The plate is made of mild steel with a Young's modulus of 235 GPa, yield strength of 305 MPa and ultimate tensile strength of 446 MPa. The material behavior is assumed to be bilinear elastic-plastic with a plastic modulus of 1 GPa. An axisymmetric finite element model of 300 elements was used in the simulation.

Figures 6.25 and 6.26 compare the load displacement curves obtained from the experiment and the ALE simulation for the two cases of the flat-faced and the hemispherically-tipped indenters, respectively. Experiments show that the plate fails by perforation at approximately 20 mm under the flat-faced indenter and at approximately 30 mm under the hemispherically-tipped indenter. Accordingly, utmost displacements of 20 mm and 30 mm were sought in the simulations, respectively. The figures show a very good agreement between the ALE predictions and experimental results. Deviations from the experimental results are more pronounced at higher deformation levels and may be attributed to discrepancies between the material model and actual material behavior at higher loads.

Figures 6.27 and 6.28 show the predicted deformed shape for the plate when loaded using the flat-faced and the hemispherically-tipped indenters, respectively. Figures 6.29 and 6.30 compare the deformed shape and equivalent plastic strain distribution in the indenter region. The figures show that the hemispherically-tipped indenter caused a greater degree of local indentation and bending around the indenter than the flat-faced indenter. For the flat-faced indenter, ALE eliminated the need for contact analysis between the sheet and the indenter. For the hemispherically-tipped indenter, ALE prevented mesh distortion at high levels of local deformation in the indenter region. The deformed shapes are in good agreement with those of the experiments [80].

Figure 6.31 shows that the flat-faced indenter produced a steeper load-displacement curve but with a smaller load and displacement to failure than the hemispherically-tipped indenter. The indenter profile plays an important role in the plate response. The degree of local indentation and the onset of perforation are highly dependent on the indenter profile. It is seen that the hemispherically-tipped indenter caused greater degree of local indentation and bending around the indenter than the flat-faced indenter. Although the local indentation caused plate thinning, the accompanied stretching and bending served to decrease shearing effects, thus delaying the onset of perforation. Thus, plates loaded by hemispherically-tipped indenters are able to withstand greater loads at higher deformations before perforation occurs than those loaded by flat-faced indenters.

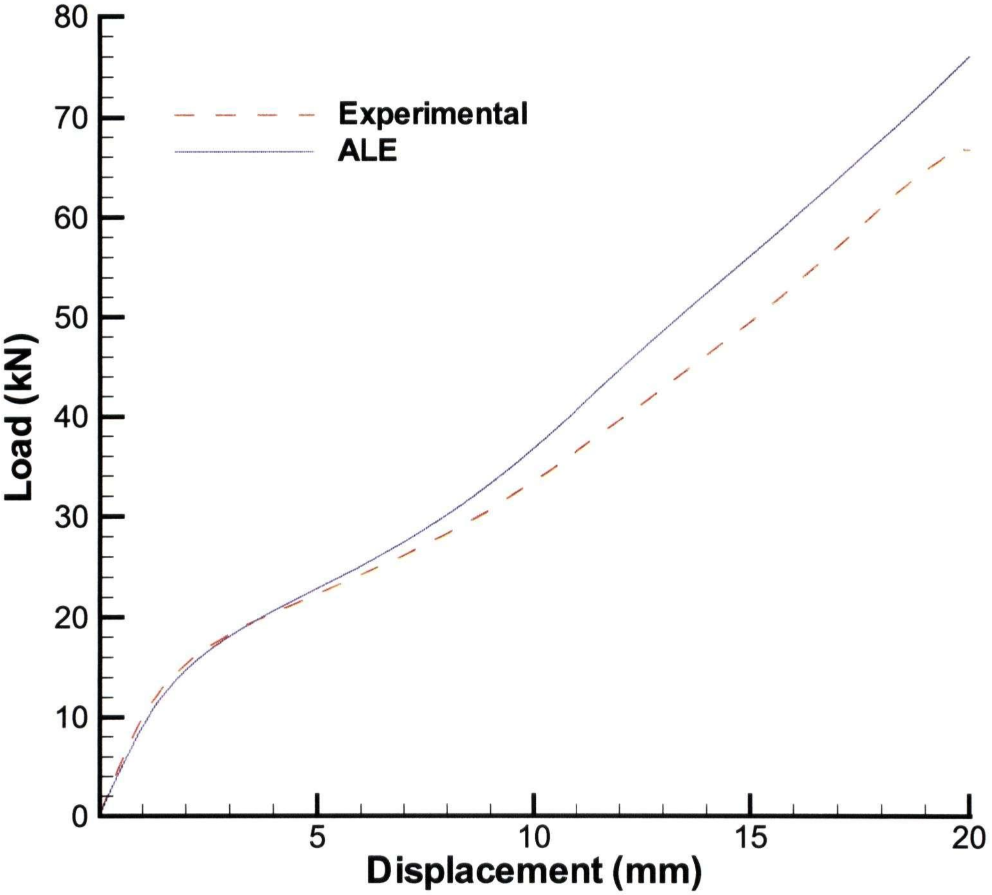


Figure 6.25. Comparison of loading curve for the flat-faced indenter.

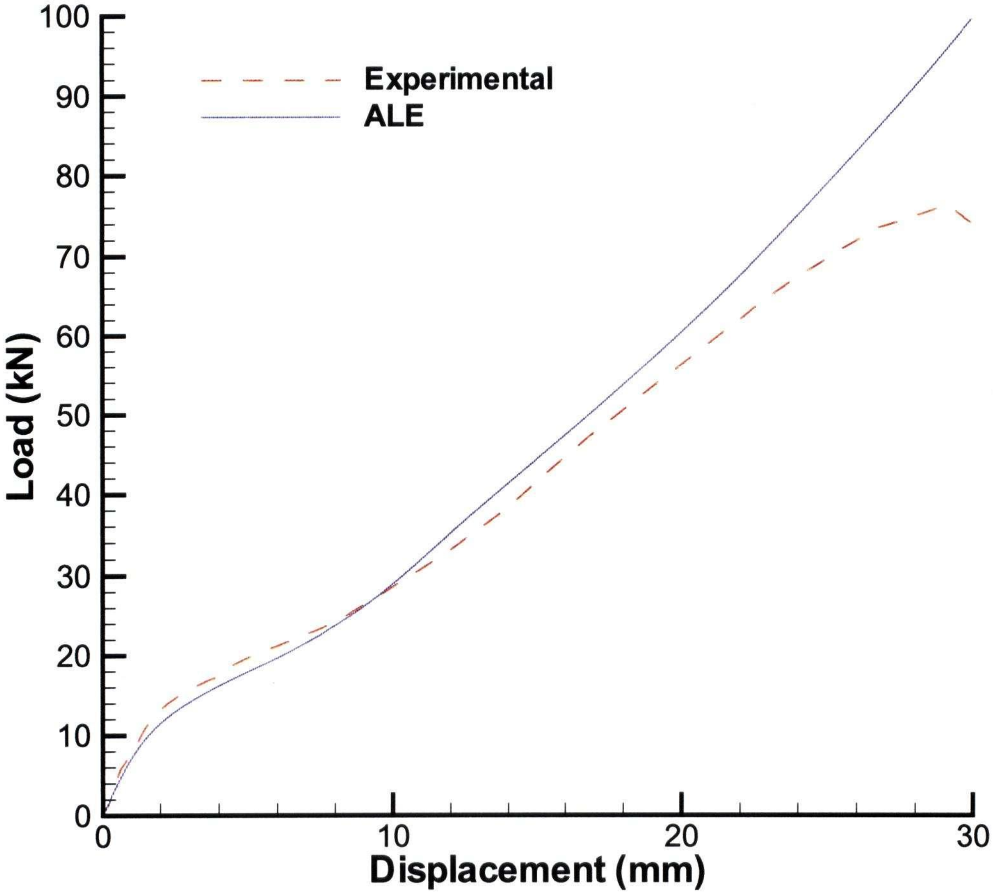


Figure 6.26. Comparison of loading curve for the hemispherically-tipped indenter.

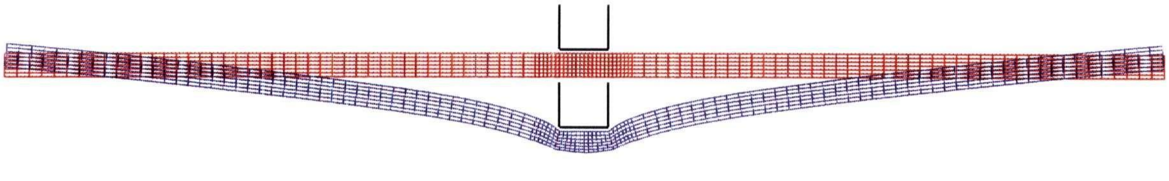


Figure 6.27. Plate deformed shape under flat-faced indenter.

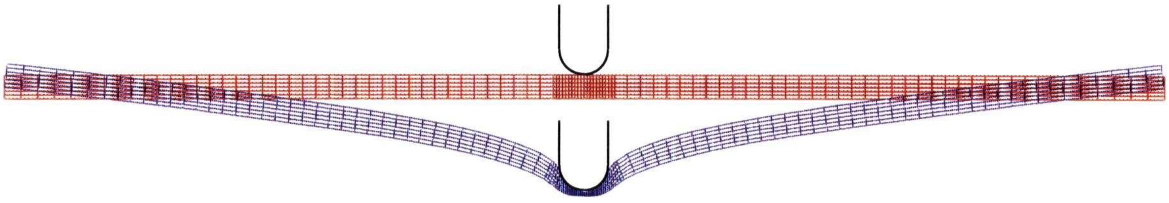


Figure 6.28. Plate deformed shape under hemispherically-tipped indenter.

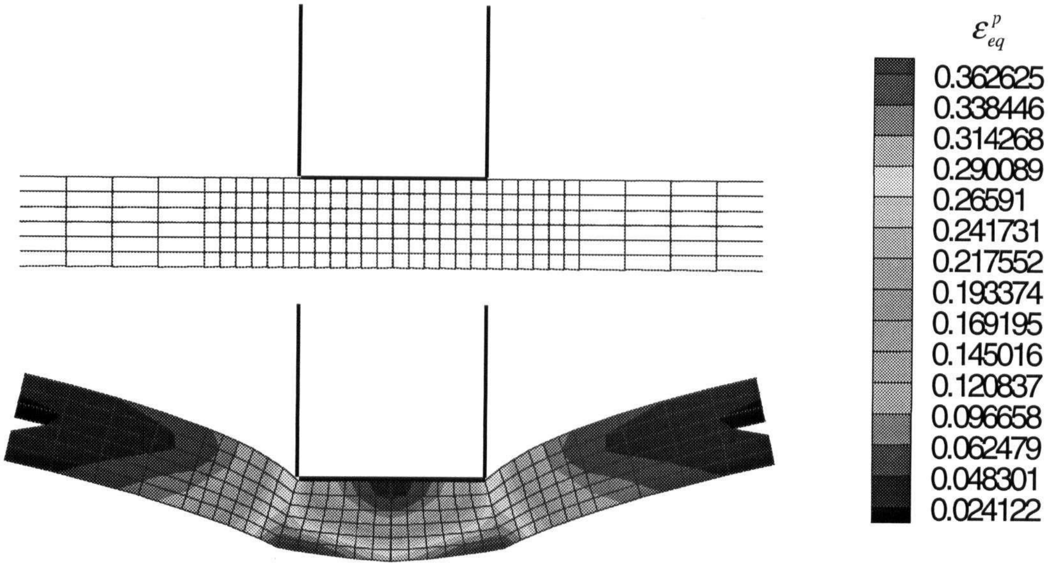


Figure 6.29. Plastic strain distribution for flat-faced indenter.

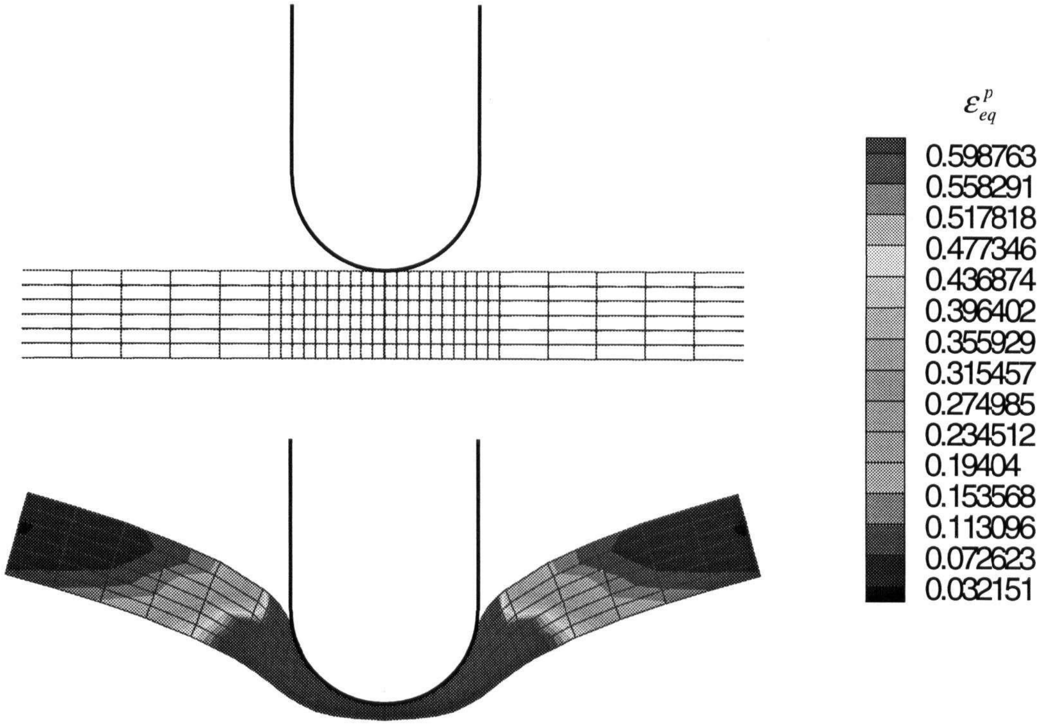


Figure 6.30. Plastic strain distribution for hemispherically-tipped indenter.

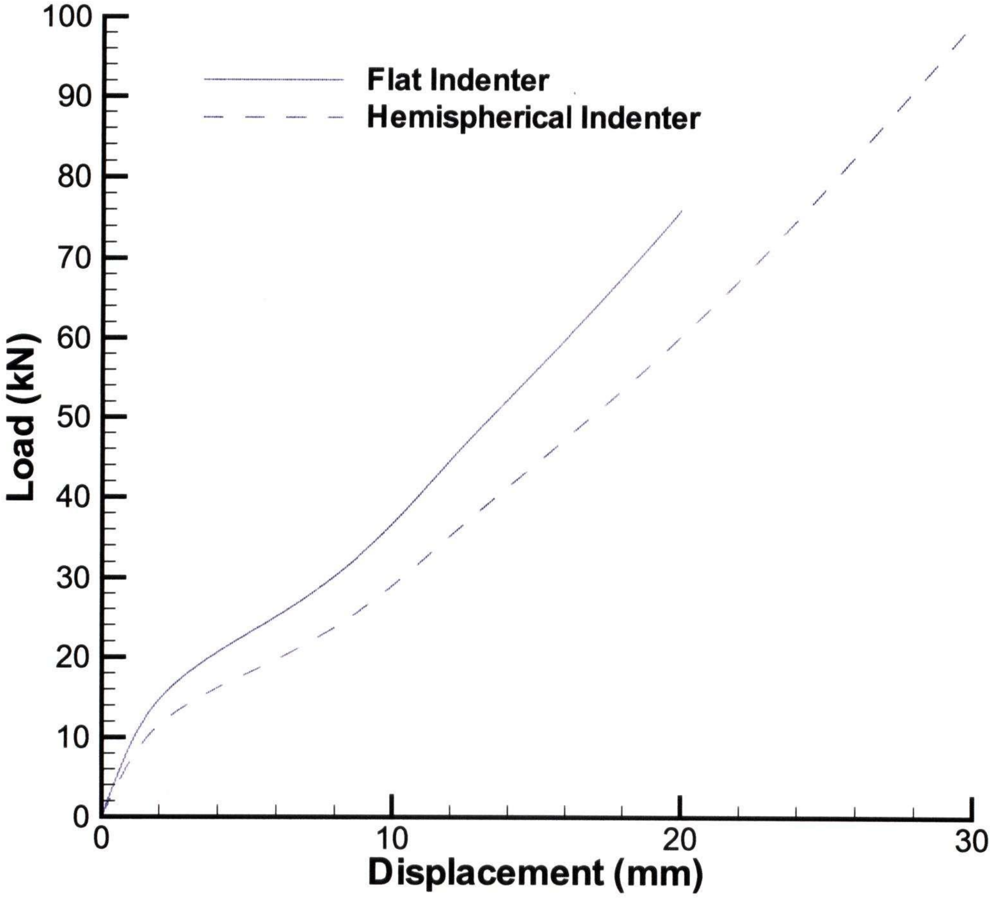


Figure 6.31. Comparison of the loading curve for flat and hemispherical indenters.

6.7 QUASI-STATIC AND DYNAMIC BRAKE BENDING

The aim of this example is to demonstrate the significance of the dynamic formulation by comparing the results of simulation of a plane-strain brake bending process (also known as air bending) using quasi-static and dynamic analyses. Choudhry and Lee [41] pointed out that although the tooling and the process for this problem are rather simple, they simulated this problem using a dynamic Lagrangian formulation to prove the effectiveness of dynamic analysis even for sheet metal forming processes which are essentially quasi-static. It also provides a challenging test for the numerical stability of the solution algorithm due to the small strains and large rotations involved. The tooling geometry for the process is shown in Figure 6.32. The punch and the die are both 10 mm in radius. The sheet is 1 mm thick. The material behavior of the sheet is assumed to be bilinear elastic-plastic with a Young's modulus of 210 GPa, Poisson's ratio of 0.3, yield strength of 300 MPa, and a plastic modulus of 1 GPa. Free boundary conditions are assumed at the sheet edge.

Figures 6.33 and 6.34 show the punch load against the punch displacement for both the quasi-static and dynamic simulations, respectively. While the quasi-static formulation gives a rather smooth curve for the load-displacement curve, the dynamic formulation exhibits oscillations that agree with the static solution only in the average sense. The maximum punch displacement attained when modeling the process as a quasi-static process was 37 mm at which point the tangent stiffness matrix became extremely ill-conditioned and equilibrium iterations ceased to converge. However, a displacement of

52.5 mm was obtained when dynamic effects were considered. The results are in agreement with those previously reported [41].

Figure 6.35 shows the distribution of the equivalent plastic strain at different punch displacements. It is shown that the sheet deformation is primarily due to bending in the region in contact with the punch and virtually no strain occurs away from the punch. One is able to conclude that although inertia effects are negligible in this process, contributions from the dynamic terms seem to stabilize the convergence problems resulting from the large rigid body motions with little straining. Virtually identical strain distributions are obtained for both quasi-static and dynamic formulations when a convergent solution is obtained.

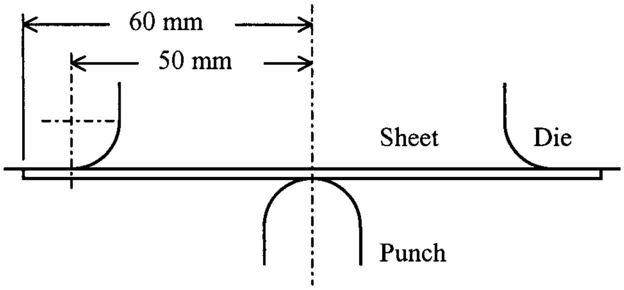


Figure 6.32. Tooling geometry for the brake bending process.

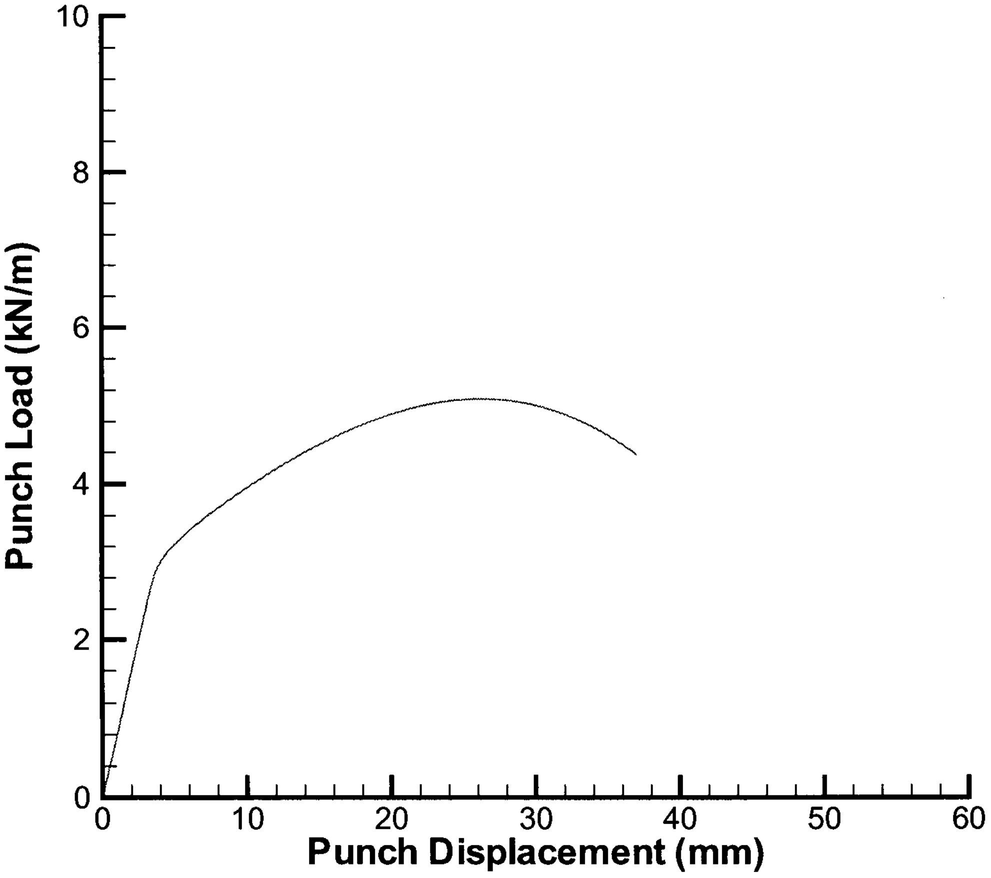


Figure 6.33. Quasi-static load-displacement curve for the brake bending process.

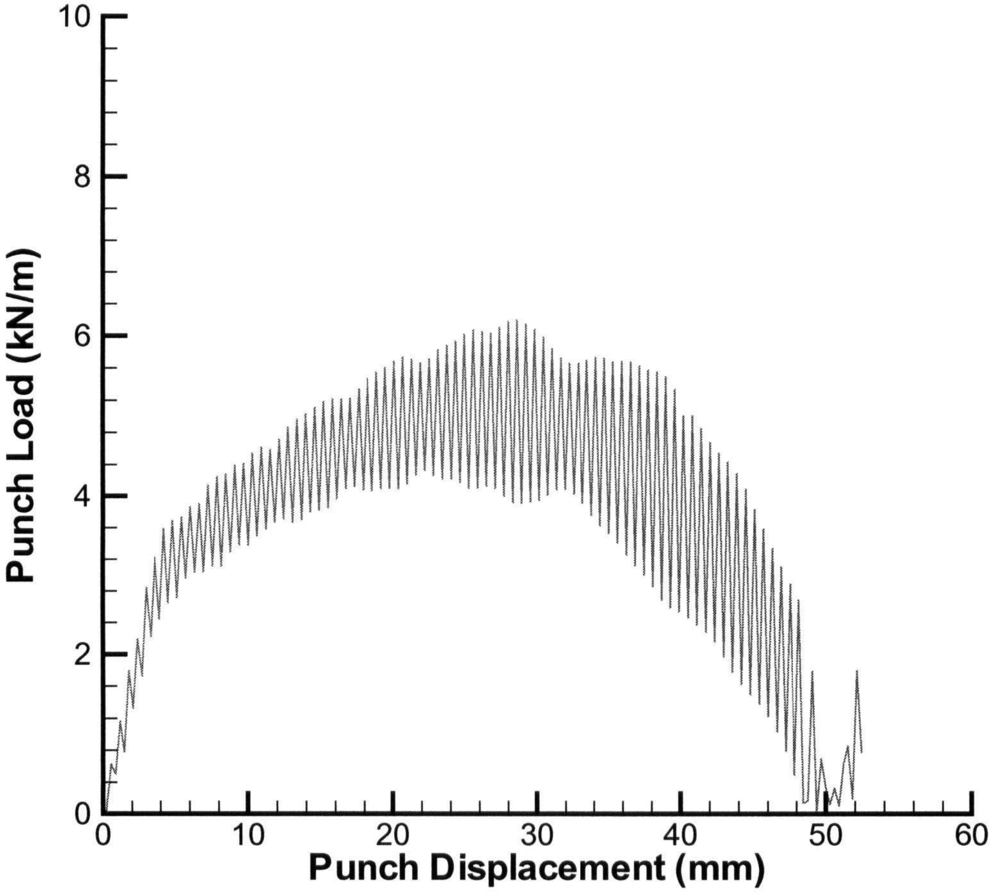


Figure 6.34. Dynamic load-displacement curve for the brake bending process.

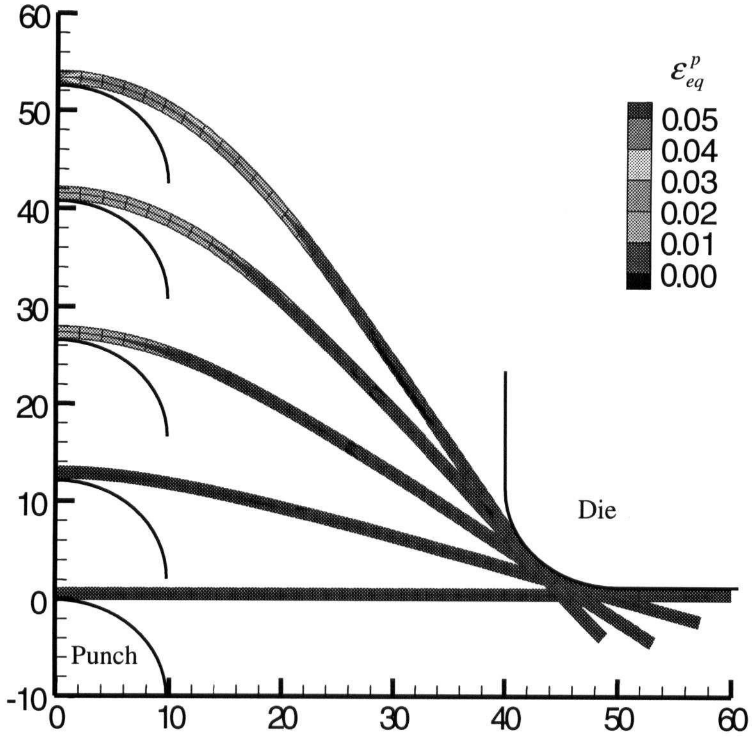


Figure 6.35. Plastic strain distribution for the brake bending process.

Chapter 7

CONCLUSIONS

7.1 SUMMARY

The accomplishments and conclusions of the present work may be summarized in the following:

- Derivation of fully coupled ALE virtual work equations from the basic principles of continuum mechanics for quasi-static and dynamic analyses. The fully coupled ALE virtual work equations derived in this work are unique as they are based on a new treatment for convective terms and because displacements are used as the independent variables as opposed to velocities which is a common practice in the ALE literature. The derivation introduces implicit fully coupled dynamic ALE calculations for the first time.
- Introduction of a new method for the treatment of convective terms in the ALE equilibrium equations that avoids the use of unjustified assumptions in calculating spatial gradients of stresses. This new method results in a consistent formulation and maintains its generality and accuracy.
- Discretization of the ALE virtual work equations using isoparametric finite elements. An effort is made to present the different ALE virtual work terms in matrix forms that are easy to compute while maintaining the resemblance of the calculations with the

Lagrangian formulation. Implementation of the ALE finite element equations into a 2-D computer code for plane stress, plane strain and axisymmetric problems with 4, 8 and 9 node elements. Special attention is made to the modularity of the code structure.

- Introduction of a new general relation that relates grid displacements with material displacements. Implementation of this relation on the element level during the assembly of element matrices eliminates grid displacements from the equations and offers a substantial saving in computation time. New treatment for mesh motion on material boundaries that allows boundary nodes to move in the tangential direction to the material boundaries in order to maintain a uniform distribution while satisfying the boundary constraint.
- Implementation of time integration schemes that allow for implicit, explicit and mixed implicit-explicit calculations. Implementation of a line search technique to accelerate the convergence of implicit calculations.
- Employment of a simple contact algorithm based on the direct constraint method. Including contact capabilities into the code allowed the simulation of more applications.
- Establishing a clear connection between the developed ALE formulation and the Lagrangian formulation. ALE is shown to be a logical extension to the Updated Lagrangian formulation and the necessary modifications to current Lagrangian codes are clearly identified. The author believes that a Lagrangian-consistent ALE

formulation will help spread the use of ALE in general-purpose commercial finite element codes in large deformation analyses.

- Simulation of several large deformation quasi-static and dynamic metal forming applications using the developed ALE code and other commercial Lagrangian-based codes. Experimental analysis was carried out for one of the simulated applications. Comparison of ALE predictions with analytical, numerical and experimental results is presented. ALE results are in good agreement with the other methods of analysis. The various simulations clearly show the power of the ALE formulation in preventing mesh distortion under high degrees of deformation. ALE was also employed to eliminate the need for contact analysis and to avoid corner penetration in some applications.

7.2 FUTURE WORK

The focus of the work in this thesis has been on the derivation of a consistent ALE formulation for quasi-static and dynamic solid mechanics applications, implementation into a 2-D finite element code and application in the simulation of large deformation and metal forming problems. Future developments may be directed towards the enhancement and extension of the formulation capabilities, improvement of the current code implementation and the simulation of more applications.

From the theoretical point of view, the ALE formulation presented in this work may be complemented in several areas:

- Investigation of more powerful mesh motion schemes. The transfinite mapping method, employed in this work as the mesh motion scheme, optimizes the mesh based solely on the geometry of the boundaries. Although this scheme is shown to be adequate for many applications, a better mesh can be generally produced if stress localization and strain gradient effects are taken into account when moving the mesh. This is essential if severe stress gradients are present. Isoparametric mapping can also be employed as an efficient mesh motion scheme especially for 3-D analyses.
- Extension of the formulation into the fluid mechanics area by deriving the virtual work equations for fluid elements. Material models for fluid elements can be introduced into the ALE governing equations following a similar procedure as that used for the constitutive equations of elastic-plastic materials.

On the implementation level, desirable improvements to the current code may include:

- Extension of the analysis dimensions to 3-D. The developed ALE equations are general and are directly applicable in 3-D. However, this extension will require the addition of 3-D brick and shell elements. It will also necessitate the implementation of a 3-D mesh motion scheme and a surface remeshing procedure, which are not simple tasks.
- Modification of the current contact algorithm or implementation of a more reliable one, possibly based on Lagrange multipliers and/or penalty function. The current contact algorithm causes lots of numerical problems in achieving convergence. It is

essential to use a superior and more stable contact algorithm especially if 3-D calculations are introduced.

On the applications side, the use of the developed ALE formulation and code can produce an impact in the simulation of several applications:

- Fracture mechanics is an application that is well suited for the ALE calculations. ALE offers an effective method for the simulation of crack tip advancement and control of mesh quality near the crack tip compared to the more approximate nodal separation and remeshing techniques used in the Lagrangian simulations. Using the dynamic ALE formulation developed in this thesis, dynamic fracture mechanics, in which high speed loading and high speed crack propagation take place, can be easily simulated. However, additional features to the current code implementation, such as a method of handling of newly created material surfaces and calculation of the J integral and strain energy density, are necessary.
- Fluid-structure interaction is another class of applications that can be efficiently simulated using ALE and the implemented solution schemes. Fluid mechanics problems are characterized with fine meshes and large material motions while material motion in structural problems is generally more restricted and require a relatively lower mesh density. ALE, with its flexible mesh motion scheme, can efficiently handle the distinct mesh motion requirements that occur in this kind of interaction problems. In addition, since mesh resolutions can be quite different at different regions in the problem, explicit integration for the relatively less dense mesh region (with larger critical time step size) and implicit integration for the more fine

mesh region (with restrictively small critical time step size) can be combined in the analysis leading to savings in computational time.

Bibliography

[1] R. Hill, "Some Basic Principles in the Mechanics of Solids Without a Natural Time", *Journal of the Mechanics and Physics of Solids*, Vol. 7, pp. 209-225, 1959.

[2] H.D. Hibbitt, P.V. Marcal and J.R. Rice, "A Finite Element Formulation for Problems of Large Strain and Large Displacement", *International Journal of Solids and Structures*, Vol. 6, pp. 1069-1086, 1970.

[3] R.M. McMeeking and J.R. Rice, "Finite Element Formulations for Problems of Large Elastic-Plastic Deformation", *International Journal of Solids and Structures*, Vol. 11, pp. 601-616, 1975.

[4] K.J. Bathe, E. Ramm and E.L. Wilson, "Finite Element Formulations for Large Deformation Dynamic Analysis", *International Journal for Numerical Methods in Engineering*, Vol. 9, pp. 353-386, 1975.

[5] J.C. Nagtegaal and N. Rebelo, "On the Development of a General Purpose Finite Element Program for Analysis of Forming Processes", *International Journal for Numerical Methods in Engineering*, Vol. 25, pp. 113-131, 1988.

[6] *ABAQUS User's Manual*, Hibbitt, Karlsson, & Sorensen Inc., 1992.

[7] *NISA II User's Manual*, Engineering Mechanics Research Corporation, 1992.

[8] *ANSYS User's Manual*, Swanson Analysis Systems Inc., 1989.

[9] P.C. Galbraith and J.O. Hallquist, "Shell Element Formulations in LS-DYNA3D: Their Use in the Modelling of Sheet Metal Forming", *Journal of Materials Processing Technology*, Vol. 50, pp. 158-167, 1995.

[10] M.J. Finn, P.C. Galbraith, L. Wu, J.O. Hallquist, L. Lum and T.L. Lin, "Use of a Coupled Explicit-Implicit Solver for Calculating Springback in Automotive Body Panels", *Journal of Materials Processing Technology*, Vol. 50, pp. 395-409, 1995.

[11] J. Tang, W.T. Wu and J. Walters, "Recent Development and Applications of Finite Element Method in Metal Forming", *Journal of Materials Processing Technology*, Vol. 46, pp. 117-126, 1994.

[12] A. Makinouchi and M. Kawka, "Process Simulation in Sheet Metal Forming", *Journal of Materials Processing Technology*, Vol. 46, pp. 291-307, 1994.

[13] M. Karima, "Practical Application of Process Simulation in Stamping", *Journal of*

Materials Processing Technology, Vol. 46, pp. 309-320, 1994.

[14] W. Kubli and J. Reissner, "Optimization of Sheet Metal Forming Processes Using the Special Purpose Program AUTOFORM", *Journal of Materials Processing Technology*, Vol. 50, pp. 292-305, 1995.

[15] L. Baillet, M. Brunet and Y. Berthier, "Experimental and Numerical Dynamic Modelling of Ironing Process", *Journal of Materials Processing Technology*, Vol. 60, pp. 677-684, 1996.

[16] M. Anon, "Numerical Simulation of Sheet Metal Forming Processes", *Sheet Metal Industries*, Vol. 32, No. 11, pp. 24-30, 1995.

[17] J. Cheng and N. Kikuchi, "A Mesh Rezoning Technique for Finite Element Simulations of Metal Forming Processes", *International Journal for Numerical Methods in Engineering*, Vol. 23, pp. 219-228, 1986.

[18] *MSC.MARC User's Guide*, MSC Software Corporation, 2000.

[19] M.S. Gadala, G.A. Oravas and M.A. Dokainish, "A Consistent Eulerian Formulation of Large Deformation Problems in Statics and Dynamics", *International Journal of Non-linear Mechanics*, Vol. 18, No. 1, pp. 21-35, 1983.

[20] K.A. Derbalian, E.H. Lee, R.L. Mallet and R.M. McMeeking, "Finite Element Metal Forming Analysis with Spatially Fixed Mesh", *Applications of Numerical Methods to Forming Processes*, ASME, AMD-Vol. 28, pp. 39-47, 1978.

[21] N. Brannberg and J. Mackerle, "Finite Element Methods and Material Processing Technology", *Engineering Computations*, Vol. 11, pp. 413-455, 1994.

[22] W.F. Noh, "A Time Dependent Two-Space-Dimensional Coupled Eulerian-Lagrangian Code", *Methods in Computational Physics*, Vol. 3, pp. 117-179, 1964.

[23] C.W. Hirt, A.A. Amsden and J.L. Cook, "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds", *Journal of Computational Physics*, Vol. 14, pp. 227-253, 1974.

[24] T.J.R. Hughes, W.K. Liu and T.K. Zimmermann, "Lagrangian-Eulerian Finite Element Formulation for Incompressible Viscous Flows", *Computer Methods in Applied Mechanics and Engineering*, Vol. 29, pp. 329-349, 1981.

[25] J.M. Kennedy and T.B. Belytschko, "Theory and Application of a Finite Element Method for Arbitrary Lagrangian-Eulerian Fluids and Structures", *Nuclear Engineering and Design*, Vol. 68, pp. 129-146, 1981.

[26] J. Huétink, "Analysis of Metal Forming Processes Based on a Combined Eulerian-Lagrangian Finite Element Formulation", *International Conference on Numerical Methods in Industrial Forming Processes*, pp. 501-509, 1982.

[27] J. Huétink, P.T. Vreede and J. van der Lugt, "Progess in Mixed Eulerian-Lagrangian Finite Element Simulation of Forming Processes", *International Journal for Numerical Methods in Engineering*, Vol. 30, pp. 1441-1457, 1990.

[28] P.J.G. Schreurs, F.E. Veldpaus and W.A.M. Brekelmans, "An Arbitrary-Eulerian-Lagrangian Finite Element Model for the Simulation of Geometrical Non-Linear Hyper-Elastic and Elasto-Plastic Deformation Processes", *International Conference on Numerical Methods in Industrial Forming Processes*, pp. 491-500, 1982.

[29] P.J.G. Schreurs, F.E. Veldpaus and W.A.M. Brekelmans, "Simulation of Forming Processes Using the Arbitrary Eulerian-Lagrangian Formulation", *Computer Methods in Applied Mechanics and Engineering*, Vol. 58, pp. 19-36, 1986.

[30] R.B. Haber, "A Mixed Eulerian-Lagrangian Displacement Model for Large-Deformation Analysis in Solid Mechanics", *Computer Methods in Applied Mechanics and Engineering*, Vol. 43, pp. 277-292, 1984.

[31] H.M. Koh and R.B. Haber, "Elastodynamic Formulation of the Eulerian-Lagrangian Kinematic Description", *Journal of Applied Mechanics*, Vol. 53, pp. 839-845, 1986.

[32] H.M. Koh, H.S. Lee and R.B. Haber, "Dynamic Crack Propagation Using Eulerian-Lagrangian Kinematic Descriptions", *Computational Mechanics*, Vol. 3, pp. 141-155, 1988.

[33] W.K. Liu, T. Belytschko and H. Chang, "An Arbitrary Lagrangian-Eulerian Finite Element Method for Path-Dependent Materials", *Computer Methods in Applied Mechanics and Engineering*, Vol. 58, pp. 227-245, 1986.

[34] W.K. Liu, H. Chang, T. Belytschko, and J.S. Chen, "Arbitrary Lagrangian-Eulerian Stress Update Procedures for Forming Simulations", *Advances in Inelastic Analysis*, ASME-AMD, Vol. 88, pp. 153-175, 1987.

[35] W.K. Liu, H. Chang, J.S. Chen and T. Belytschko, "Arbitrary Lagrangian-Eulerian Petrov-Galerkin Finite Elements for Nonlinear Continua", *Computer Methods in Applied Mechanics and Engineering*, Vol. 68, pp. 259-310, 1988.

[36] D.J. Benson, "An Efficient, Accurate, Simple ALE Method for Nonlinear Finite Element Programs", *Computer Methods in Applied Mechanics and Engineering*, Vol. 72, pp. 305-350, 1989.

[37] A. Huerta and F. Casadei, "New ALE Applications in Nonlinear Fast Transient Solid Dynamics", Engineering Computations, Vol. 11, pp. 317-345, 1994.

[38] S. Ghosh and N. Kikuchi, "An Arbitrary Lagrangian-Eulerian Finite Element Method for Large Deformation Analysis of Elastic-Viscoplastic Solids", *Computer Methods in Applied Mechanics and Engineering*, Vol. 86, pp. 127-188, 1991.

[39] J. Wang, *Arbitrary Lagrangian-Eulerian Method and Its Applications in Solid Mechanics*, PhD Dissertation, The University of British Columbia, January 1998.

[40] J. Tirosh and D. Iddan, "The Dynamics of Fast Metal Forming Processes", *Journal of Mechanics and Physics of Solids*, Vol. 42, pp. 611-628, 1994.

[41] S. Choudhry and J. K. Lee, "Dynamic Plane-Strain Finite Element Simulation of Industrial Sheet-Metal Forming Processes", *International Journal of Mechanical Sciences*, Vol. 36, No. 3, pp. 189-207, 1994.

[42] D.A. Schoch, "Critical Factors for Achievement of Successful High Speed Metal Forming Production", *Journal of Materials Processing Technology*, Vol. 46, pp. 409-414, 1994.

[43] J.C. Gelin, "Dynamic Loading, Viscoplasticity and Temperature Effects on the Evolution of Damage in Metal Forming Processes", *Journal of Materials Processing Technology*, Vol. 32, pp. 169-178, 1992.

[44] P. Steinmann, C. Miehe and E. Stein, "Fast Transient Dynamic Plane Stress Analysis of Orthotropic Hill-Type Solids at Finite Elastoplastic Strains", *International Journal of Solids and Structures*, Vol. 33, No. 11, pp. 1543-1562, 1996.

[45] K. Kormi, R.A. Etheridge and D.C. Webb, "FEM Simulation of the Static and Dynamic Forming of Circular Plates", *Journal of Materials Processing Technology*, Vol. 42, pp. 451-462, 1994.

[46] J.F. Fontane and J.C. Gelin, "A Finite Element Analysis of High-Speed Metal-Forming Processes", *Annals of the CIRP*, Vol. 40, pp. 277-280, 1991.

[47] S. Kapinski, "Influence of the Punch Velocity on Deformation of the Material in Deep-drawn Flange", *Journal of Materials Processing Technology*, Vol. 34, pp. 419-424, 1992.

[48] K.J. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, 1982.

[49] A. Makinouchi, "Sheet Metal Forming Simulation in Industry", *Journal of Materials Processing Technology*, Vol. 60, pp. 19-26, 1996.

[50] T. Huo and E. Nakamachi, "Evaluation of the Dynamic Explicit/Elastoviscoplastic

Finite Element Method in Sheet-Forming Simulation", *Journal of Materials Processing Technology*, Vol. 50, pp. 180-196, 1995.

[51] D. Zhou and R.H. Wagoner, "Development and Application of Sheet-Forming Simulation", *Journal of Materials Processing Technology*, Vol. 50, pp. 1-16, 1995.

[52] N. Rebelo, J.C. Nagtegaal, L.M. Taylor and R. Passmann, "Industrial Application of Implicit and Explicit Finite Element Methods to Forming Processes", *Numerical Methods for Simulation of Industrial Metal Forming Processes*, ASME, CED-Vol. 5, pp. 67-76, 1992.

[53] R.G. Whirley, B.E. Engelmann and R.W. Logan, "Some Aspects of Sheet Forming Simulation Using Explicit Finite Element Techniques", *Numerical Methods for Simulation of Industrial Metal Forming Processes*, ASME, CED-Vol. 5, pp. 77-83, 1992.

[54] D.Y. Yang, D.W. Jung, I.S. Song, D.J. Yoo and J.H. Lee, "Comparative Investigation into Implicit, Explicit and Iterative Implicit/Explicit Schemes for the Simulation of Sheet-Metal Forming Processes", *Journal of Materials Processing Technology*, Vol. 50, pp. 39-53, 1995.

[55] L. Taylor, J. Cao, A.P. Karafillis and M.C. Boyce, "Numerical Simulations of Sheet Metal Forming", *Journal of Materials Processing Technology*, Vol. 50, pp. 168-179, 1995.

[56] H.N. Bayoumi, M.S. Gadala and J. Wang, "Application of the ALE Formulation to Metal Forming Problems", *3rd EUROMECH Solid Mechanics Conference*, Stockholm, Sweden, August 1997.

[57] H.N. Bayoumi, M.S. Gadala and J. Wang, "Numerical Simulation of Metal Forming Processes", *Proceedings of the 6th International Conference on Numerical Methods in Industrial Forming Processes - NUMIFORM'98*, Enschede, The Netherlands, June 1998.

[58] H.N. Bayoumi and M.S. Gadala, "An Implicit Arbitrary Lagrangian Eulerian Formulation for Quasi-static and Dynamic Solid Mechanics Applications", *Computer Methods in Applied Mechanics and Engineering*, submitted for publication, June 2000.

[59] J. Wang and M.S. Gadala, "Formulation and Survey of ALE Method in Nonlinear Solid Mechanics", *Finite Elements in Analysis and Design*, Vol. 24, pp. 253-269, 1997.

[60] M.S. Gadala and J. Wang, "Simulation of Metal Forming Processes with Finite Element Methods", *International Journal for Numerical Methods in Engineering*, Vol. 44, pp. 1397-1428, 1999.

[61] M.S. Gadala and J. Wang, "Elasto-Plastic Finite Element Simulation of Rolling and Compression between Wedge-Shaped Dies", *Journal of Materials Processing Technology*, Vol. 97, pp. 132-147, 2000.

[62] M.R. Movahhedy, *ALE Simulation of Chip Formation in Orthogonal Metal Cutting Process*, PhD Dissertation, The University of British Columbia, January 2000.

[63] M.R. Movahhedy, M.S. Gadala and Y. Altintas, "Simulation of Orthogonal Metal Cutting Process Using an Arbitrary Lagrangian-Eulerian Finite Element Method", *Journal of Materials Processing Technology*, Vol. 103, pp. 267-275, 2000.

[64] L.E. Malvern, *Introduction to the Mechanics of a Continuous Medium*, Prentice Hall, 1969.

[65] H.N. Bayoumi and M.S. Gadala, "Finite Element Implementation of the Fully Coupled ALE Formulation", *International Journal for Numerical Methods in Engineering*, submitted for publication, December 2000.

[66] R. Haber, M.S. Shepard, J.F. Abel, R.H. Gallagher and D.P. Greenberg, "A General Two-Dimensional, Graphical Finite Element Preprocessor Utilizing Discrete Transfinite Mappings", *International Journal for Numerical Methods in Engineering*, Vol. 17, pp. 1015-1044, 1981.

[67] H.N. Bayoumi and M.S. Gadala, "Finite Element Analysis of Large Strain Solid Mechanics Problems", *Proceedings of the 1st Canadian Conference on Nonlinear Solid Mechanics - CanCNSM'99*, Victoria, BC, Canada, pp. 375-384, June 1999.

[68] H.N. Bayoumi and M.S. Gadala, "Simulation of Large Deformation Problems Using the Arbitrary Lagrangian Eulerian Formulation", *Proceedings of the European Conference on Computation Mechanics - ECCM'99*, Munich, Germany, pp. 1-14, September 1999.

[69] T.J.R. Hughes and W.K. Liu, "Implicit-Explicit Finite Elements in Transient Analysis: Stability Theory", *Journal of Applied Mechanics*, Vol. 45, pp. 371-374, 1978.

[70] T.J.R. Hughes and W.K. Liu, "Implicit-Explicit Finite Elements in Transient Analysis: Implementation and Numerical Examples", *Journal of Applied Mechanics*, Vol. 45, pp. 375-378, 1978.

[71] B. Irons and S. Ahmad, *Techniques of Finite Elements*, J. Wiley & Sons, 1980.

[72] H.N. Bayoumi, M.S. Gadala and J. Wang, "Finite Element Frontal Solver", *Technical Report*, Submitted to Forming Technologies Inc., Oakville, Ontario, Canada, July 1996.

[73] A. Kamoulakos, “A Simple Benchmark for Impact”, *Bench Mark*, pp. 31-35, 1990.

[74] Y.Y. Zhu and S. Cescotto, “Unified and Mixed Formulation of the 4-Node Quadrilateral Elements by Assumed Strain Method: Application to Thermomechanical Problems”, *International Journal for Numerical Methods in Engineering*, Vol. 38, pp. 685-716, 1995.

[75] G.T. Camacho and M. Ortiz, “Adaptive Lagrangian Modelling of Ballistic Penetration of Metallic Targets”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 142, pp. 269-301, 1997.

[76] E.H. Lee, R.L. Mallet and W.H. Yang, “Stress and Deformation Analysis of the Metal Extrusion Process”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 10, pp. 339-353, 1977.

[77] S. Ko, *Numerical Comparison of Various Finite Element Formulations in Nonlinear and Plasticity Problems*, Master of Engineering Thesis, The University of British Columbia, in preparation.

[78] Y. Huang and Y. Lu, “Elasto-Plastic Finite Element Analysis of V-Shape Sheet Bending”, *Journal of Materials Processing Technology*, Vol. 35, pp. 129-150, 1992.

[79] Y. Huang and D. Leu, “Finite Element Analysis of Contact Problems for a Sheet Metal Bending Process”, *Computers and Structures*, Vol. 57, pp. 15-27, 1995.

[80] G.G. Corbett and S.R. Reid, “Quasi-static and Dynamic Local Loading of Monolithic Simply-Supported Steel Plate”, *International Journal of Impact Engineering*, Vol. 13, pp. 423-441, 1993.

Appendix A

GRID TIME DERIVATIVE OF VIRTUAL STRAIN

The grid time derivative of $\delta\, {}_t e_{ij}$ is given by

$$\begin{aligned}\delta\, {}_t e'_{ij} &= \frac{\partial}{\partial t}\left|\frac{1}{2}\left(\frac{\partial \delta u_i}{\partial\, {}^t x_j} + \frac{\partial \delta u_j}{\partial\, {}^t x_i}\right)\right|_{X^g} \\ &= \frac{1}{2}\left(\frac{\partial}{\partial t}\left|\frac{\partial \delta u_i}{\partial\, {}^t x_j}\right|_{X^g} + \frac{\partial}{\partial t}\left|\frac{\partial \delta u_j}{\partial\, {}^t x_i}\right|_{X^g}\right)\end{aligned} \tag{A.1}$$

For clarity, we will use vector notation to derive $\frac{\partial}{\partial t}\left|\frac{\partial \delta u_i}{\partial\, {}^t x_j}\right|_{X^g}$. We can write $\frac{\partial \delta u_i}{\partial\, {}^t x_j}$ as

$$\frac{\partial \delta u_i}{\partial\, {}^t x_j} = \frac{\partial \delta \mathbf{u}}{\partial\, {}^t \mathbf{x}} \tag{A.2}$$

Defining **I** to be the identity tensor, we have

$$\frac{\partial \delta \mathbf{u}}{\partial\, {}^t \mathbf{x}}\left(\frac{\partial \delta \mathbf{u}}{\partial\, {}^t \mathbf{x}}\right)^{-1} = \mathbf{I} \tag{A.3}$$

Taking the grid time derivative of both sides of (A.3) and rearranging gives

$$\frac{\partial}{\partial t}\left|\frac{\partial \delta \mathbf{u}}{\partial\, {}^t \mathbf{x}}\right|_{X^g}\left(\frac{\partial \delta \mathbf{u}}{\partial\, {}^t \mathbf{x}}\right)^{-1} = -\frac{\partial \delta \mathbf{u}}{\partial\, {}^t \mathbf{x}}\frac{\partial}{\partial t}\left|\left(\frac{\partial \delta \mathbf{u}}{\partial\, {}^t \mathbf{x}}\right)^{-1}\right|_{X^g} \tag{A.4}$$

Multiplying both sides of (A.4) by $\frac{\partial \delta \mathbf{u}}{\partial\, {}^t \mathbf{x}}$ from the right gives

$$\frac{\partial}{\partial t}\left|\frac{\partial \delta \mathbf{u}}{\partial\, {}^t \mathbf{x}}\right|_{X^g} = -\frac{\partial \delta \mathbf{u}}{\partial\, {}^t \mathbf{x}}\frac{\partial}{\partial t}\left|\left(\frac{\partial \delta \mathbf{u}}{\partial\, {}^t \mathbf{x}}\right)^{-1}\right|_{X^g}\frac{\partial \delta \mathbf{u}}{\partial\, {}^t \mathbf{x}} \tag{A.5}$$

in which $\frac{\partial}{\partial t}\left|\left(\frac{\partial \delta \mathbf{u}}{\partial\, {}^t \mathbf{x}}\right)^{-1}\right|_{X^g}$ can be calculated as

$$\begin{aligned}
\frac{\partial}{\partial t}\left|\left(\frac{\partial\delta\mathbf{u}}{\partial^t\mathbf{x}}\right)^{-1}\right|_{X^g} &= \frac{\partial}{\partial t}\left|\frac{\partial^t\mathbf{x}}{\partial\delta\mathbf{u}}\right|_{X^g} \\
&= \frac{\partial\left|\partial^t\mathbf{x}/\partial t\right|_{X^g}}{\partial\delta\mathbf{u}} \\
&= \frac{\partial^t\mathbf{v}^g}{\partial\delta\mathbf{u}} \\
&= \frac{\partial^t\mathbf{v}^g}{\partial^t\mathbf{x}}\frac{\partial^t\mathbf{x}}{\partial\delta\mathbf{u}} \\
&= \frac{\partial^t\mathbf{v}^g}{\partial^t\mathbf{x}}\left(\frac{\partial\delta\mathbf{u}}{\partial^t\mathbf{x}}\right)^{-1} \qquad \text{(A.6)}
\end{aligned}$$

Substituting (A.6) into (A.5) gives

$$\begin{aligned}
\frac{\partial}{\partial t}\left|\frac{\partial\delta\mathbf{u}}{\partial^t\mathbf{x}}\right|_{X^g} &= -\frac{\partial\delta\mathbf{u}}{\partial^t\mathbf{x}}\frac{\partial^t\mathbf{v}^g}{\partial^t\mathbf{x}}\left(\frac{\partial\delta\mathbf{u}}{\partial^t\mathbf{x}}\right)^{-1}\frac{\partial\delta\mathbf{u}}{\partial^t\mathbf{x}} \\
&= -\frac{\partial\delta\mathbf{u}}{\partial^t\mathbf{x}}\frac{\partial^t\mathbf{v}^g}{\partial^t\mathbf{x}} \qquad \text{(A.7)}
\end{aligned}$$

Rewriting (A.7) in indicial notation gives

$$\frac{\partial}{\partial t}\left|\frac{\partial\delta u_i}{\partial^t x_j}\right|_{X^g} = -\frac{\partial\delta u_i}{\partial^t x_k}\frac{\partial^t v_k^g}{\partial^t x_j} \qquad \text{(A.8)}$$

Therefore

$$\delta\,{}_t e'_{ij} = -\frac{1}{2}\left(\frac{\partial\delta u_i}{\partial^t x_k}\frac{\partial^t v_k^g}{\partial^t x_j} + \frac{\partial\delta u_j}{\partial^t x_k}\frac{\partial^t v_k^g}{\partial^t x_i}\right) \qquad \text{(A.9)}$$