EULERIAN FORMULATION WITH VOLUME OF SOLID (VOS) APPLICATION TO METAL FORMING

by

Khaled S. Al-Athel
B.Sc., M.Sc. King Fahad University of Petroleum and Minerals (KFUPM), 2005

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

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate Studies
(Mechanical Engineering)

The University of British Columbia
(Vancouver)

September 2010

© Khaled S. Al-Athel 2010
ABSTRACT

The volume of fluid (VOF) method has been used extensively in computational fluid dynamics (CFD) applications. The method is used to track the unconstrained flow of the fluid at the free surface in an Eulerian formulations framework. In this work we propose the use of the VOF method in nonlinear solid mechanics and metal forming applications, therefore the new method is called the Volume of Solid (VOS) method. The VOS method is to be used with Eulerian finite element (FE) formulations. In Eulerian formulation, the mesh is fixed in space and the material flows freely within the domain. This type of formulation overcomes many difficulties that are associated with other FE formulations such as mesh distortion and the need for re-meshing or a proper mesh motion scheme. The main drawback of the Eulerian formulation is the difficulty in tracking the unconstrained flow of the material at the free boundaries; therefore an additional scheme is needed to overcome this problem. The derivation of the VOS equations and its implementation in an Eulerian formulation along with applications in metal forming and metal cutting are the main goals of this research.

In this work, we present the derivation of the Eulerian quasi-static and dynamic equilibrium equations. The complete development of the VOS method is presented for the commonly used uniform mesh using a piecewise linear interface construction method, and for a general case of a non-uniform mesh using a new filling pattern scheme. The formulation of the VOS method includes the calculation of the fractional volume value of the solid in each element and finding a directional vector to locate the interface of the free surface. The implementation of the VOS method is discussed by covering important topics that include the connectivity of the free surface interfaces in addition to introducing and modeling a new algorithm for the rigid-flexible contact problem. The formulation presented in this work is implemented in a VOS FE-code and used to simulate several metal forming and metal cutting applications. The VOS simulations are validated by comparing the results with other experimental and numerical work presented in the literature.
# TABLE OF CONTENTS

ABSTRACT ........................................................................................................................................ ii

TABLE OF CONTENTS ....................................................................................................................... iii

LIST OF TABLES .................................................................................................................................. vi

LIST OF FIGURES ............................................................................................................................... vii

NOMENCLATURE .............................................................................................................................. xi

ACKNOWLEDGEMENTS ......................................................................................................................... xvi

DEDICATION ........................................................................................................................................ xvii

CHAPTER 1 INTRODUCTION ................................................................................................................ 1
  1.1 Motivation ..................................................................................................................................... 1
  1.2 Metal Forming Processes .............................................................................................................. 2
  1.3 Numerical Methods in Metal Forming .......................................................................................... 3
    1.3.1 Traditional Large Strain Finite Element Formulations ......................................................... 3
    1.3.2 Arbitrary Lagrangian-Eulerian (ALE) Formulation ............................................................... 6
    1.3.3 Meshless Methods ................................................................................................................ 7
  1.4 Scope of Research ....................................................................................................................... 8

CHAPTER 2 SURFACE TRACKING METHODS .................................................................................... 11
  2.1 Traditional Free Boundary Methods ......................................................................................... 11
  2.2 Volume of Fluid (VOF) Method .................................................................................................. 13
    2.2.1 The VOF Method .................................................................................................................. 13
    2.2.2 VOF in the Literature .......................................................................................................... 14
  2.3 Transition from VOF to VOS ..................................................................................................... 21

CHAPTER 3 EULERIAN FORMULATION ............................................................................................ 23
  3.1 Quasi-Static Analysis .................................................................................................................. 24
    3.1.1 Incremental Decomposition ................................................................................................. 25
3.1.2 Linearization ........................................................................................................ 27
3.1.3 Treatment of Convective Terms ........................................................................ 28
3.1.4 Eulerian Equilibrium Equation for Quasi-static Analysis .............................. 30
3.2 Dynamic Analysis .................................................................................................. 31
  3.2.1 Virtual Work Done by Inertia Forces ............................................................. 32
  3.2.2 Decomposition of Velocity and Acceleration .............................................. 32
  3.2.3 Linearization of the Referential Inertia Term .............................................. 33
  3.2.4 Linearization of the Convective Inertia Term .............................................. 34
  3.2.5 Eulerian Equilibrium Equation for Dynamic Analysis .............................. 36
3.3 Discretization of the Eulerian FE Equations ....................................................... 37
  3.3.1 Discretization of the Quasi-Static Eulerian Equation ................................ 37
  3.3.2 Discretization of the Dynamic Eulerian Equation ....................................... 40

CHAPTER 4 VOLUME OF SOLID (VOS) METHOD ..................................................... 44
  4.1 VOS for a Uniform Mesh .................................................................................... 46
    4.1.1 Solution of the Transport Equation .......................................................... 46
    4.1.2 Free Surface Directional Vector ............................................................... 47
    4.1.3 Construction of the Free Surface using a Geometrical PLIC Scheme ...... 52
  4.2 VOS for a Non-Uniform Mesh .......................................................................... 56
    4.2.1 Modification to the Solution of the Transport Equation ......................... 56
    4.2.2 Generalization of Free Surface Directional Vector ................................. 57
    4.2.3 Construction of the Free Surface using a Filling Pattern Scheme .......... 59

CHAPTER 5 SOLUTION CONTROLS AND IMPLEMENTATION .......................... 62
  5.1 Time Step Size and Stability ............................................................................. 62
  5.2 Updating the Velocity Field ............................................................................... 63
  5.3 Tracking Material Point Properties .................................................................. 64
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4</td>
<td>Free Surface Connectivity</td>
<td>65</td>
</tr>
<tr>
<td>5.5</td>
<td>Contact Analysis</td>
<td>69</td>
</tr>
<tr>
<td>5.6</td>
<td>Heat Transfer Analysis</td>
<td>72</td>
</tr>
<tr>
<td>5.7</td>
<td>Solution Procedure</td>
<td>73</td>
</tr>
<tr>
<td>6.1</td>
<td>Punch Indentation Problem</td>
<td>76</td>
</tr>
<tr>
<td>6.2</td>
<td>Compression between Wedge-Shaped Dies</td>
<td>81</td>
</tr>
<tr>
<td>6.3</td>
<td>Sheet Metal Extrusion</td>
<td>86</td>
</tr>
<tr>
<td>6.4</td>
<td>Backward Extrusion</td>
<td>91</td>
</tr>
<tr>
<td>6.5</td>
<td>Local Loading of a Simply-Supported Plate</td>
<td>98</td>
</tr>
<tr>
<td>6.6</td>
<td>V-Bending</td>
<td>104</td>
</tr>
<tr>
<td>6.7</td>
<td>Three-Point Bending</td>
<td>111</td>
</tr>
<tr>
<td>7.1</td>
<td>Experimental Work and Numerical Setup</td>
<td>121</td>
</tr>
<tr>
<td>7.2</td>
<td>VOS Simulation of Metal Cutting</td>
<td>122</td>
</tr>
<tr>
<td>7.3</td>
<td>Results and Discussion</td>
<td>124</td>
</tr>
<tr>
<td>8.1</td>
<td>Summary and Conclusions</td>
<td>132</td>
</tr>
<tr>
<td>8.2</td>
<td>Future Work</td>
<td>134</td>
</tr>
</tbody>
</table>

REFERENCES ........................................................................ 137
LIST OF TABLES

Table 4.1 Summary of the free surface interface parameters calculations ............... 55
Table 6.1 Contact pressure and strain comparison between VOS and [80] ............... 97
Table 6.2 von-Mises stress and strain comparison between VOS and ANSYS for the
  V-bending problem ........................................................................................................ 110
Table 7.1 List of experimental and numerical metal cutting setups ......................... 121
Table 7.2 Material properties for the workpiece and tool used in metal cutting ...... 122
LIST OF FIGURES

Figure 1.1 Initial and final configuration of a punch indentation problem using (a) Lagrangian formulation and (b) Eulerian Formulation 4
Figure 2.1 Distribution of ‘f’ at free surface and neighboring elements 13
Figure 2.2 Free surface averaging using (a) SLIC and (b) PLIC 15
Figure 2.3 Schematic of regular and hanging nodes in adaptive grid refinement 18
Figure 4.1 Definition of free surface element parameters 47
Figure 4.2 Definition of the free surface directional vector 51
Figure 4.3 Construction of fictitious elements at the right end of the domain 51
Figure 4.4 Four quadrant categorization of the free surface 52
Figure 4.5 Free surface shape possibilities in a uniform mesh 54
Figure 4.6 3x3 non-uniform stencil with outward facial normals 57
Figure 4.7 3x3 non-uniform stencil with unit vectors 59
Figure 4.8 Free surface sectioning in the filling pattern scheme 60
Figure 5.1 Velocity field update for newly added nodes 63
Figure 5.2 Tracking material points at the boundary using the free surface directional vectors 65
Figure 5.3 Default free surface connectivity scheme 66
Figure 5.4 Contact-modified free surface connectivity 68
Figure 5.5 Effect of weighting factor on the free surface connectivity process 68
Figure 5.6 Penetration detection and surface repositioning in contact analysis 71
Figure 5.7 Flowchart of the VOS FE-code solution procedure 75
Figure 6.1  Geometrical setup for the punch indentation problem ....................... 78
Figure 6.2  Evolution of the workpiece deformed shape in the punch indentation
problem using VOS .................................................................................. 78
Figure 6.3  von-Mises plastic strain distribution for the punch indentation problem79
Figure 6.4  von-Mises stress distribution for the punch indentation problem ........ 80
Figure 6.5  Setup for the compression between wedge-shaped dies problem ....... 82
Figure 6.6  Evolution of the workpiece deformed shape in the compression between
wedge-shaped dies problem using VOS .......................................................... 83
Figure 6.7  von-Mises strain distribution in the compression between wedge-shaped
dies problem .............................................................................................. 84
Figure 6.8  Load-strain comparison curves for the compression between wedge-
shaped dies problem .................................................................................. 85
Figure 6.9  Geometrical setup for the sheet metal extrusion problem ............... 88
Figure 6.10 Evolution of the billet deformed shape in the sheet metal extrusion
problem using VOS .................................................................................. 89
Figure 6.11 von-Mises strain distribution for the sheet metal extrusion problem .... 89
Figure 6.12 Longitudinal stresses at different lateral positions for the sheet metal
extrusion problem ....................................................................................... 90
Figure 6.13 Longitudinal stresses along the thickness of the billet in the sheet metal
extrusion problem ....................................................................................... 90
Figure 6.14 Geometrical setup for the backward extrusion problem .................. 93
Figure 6.15 Evolution of the workpiece deformed shape in the backward extrusion
problem using VOS .................................................................................. 94
Figure 6.16 Load-Stroke curve for the backward extrusion problem.......................... 95
Figure 6.17 Contact pressure (MPa) between workpiece and die at different strokes in the backward extrusion problem ................................................................. 95
Figure 6.18 von-Mises strain distribution for the backward extrusion problem....... 96
Figure 6.19 Geometrical setup for the plate indentation problem ....................... 100
Figure 6.20 Deformed shape of the plate indentation problem using VOS ........... 101
Figure 6.21 Load-Displacement curve for the flat plate indentation problem....... 101
Figure 6.22 Load-Displacement curve for the hemispherical plate indentation problem .......................................................................................................................... 102
Figure 6.23 Comparison of the load-displacement curves of the flat and the hemispherical indenters .................................................................................................. 102
Figure 6.24 von-Mises strain distribution using the flat-face indenter in the plate indentation problem .................................................................................................. 103
Figure 6.25 von-Mises strain distribution using the hemispherical indenter in the plate indentation problem .................................................................................... 103
Figure 6.26 Geometrical setup for the V-bending problem................................. 107
Figure 6.27 Evolution of the deformed shape from Experiment and VOS in the V-bending problem ................................................................................................. 108
Figure 6.28 Load-Stroke curve for the V-bending problem ................................. 108
Figure 6.29 von-Mises stress distribution for the V-bending problem ................. 109
Figure 6.30 von-Mises strain distribution for the V-bending problem ............... 110
Figure 6.31 Geometrical setup for the three-point bending problem ................. 113
| Figure 6.32 | Evolution of the three-point bending deformed shape from the Experiment and VOS | 114 |
| Figure 6.33 | Load-Stroke curve for the three-point bending problem | 115 |
| Figure 6.34 | von-Mises stress distribution for the three-point bending problem | 115 |
| Figure 6.35 | von-Mises strain distribution for the three-point bending problem | 116 |
| Figure 7.1  | Schematic view of metal cutting process with chamfered tool | 120 |
| Figure 7.2  | Evolution of chip deformation in the VOS metal cutting simulation | 124 |
| Figure 7.3  | Velocity fields for sharp tool, -10° chamfered tool, -25° chamfered tool and a blunt tool | 127 |
| Figure 7.4  | Effective stress distribution for sharp tool, -10° chamfered tool, -25° chamfered tool and a blunt tool | 128 |
| Figure 7.5  | Plastic strain rate distribution for sharp tool, -10° chamfered tool, -25° chamfered tool and a blunt tool | 129 |
| Figure 7.6  | Comparison of maximum normal and tangential stresses between VOS and ALE simulations | 130 |
| Figure 7.7  | Comparison between the cutting forces obtained using VOS, ALE and experimental results | 130 |
| Figure 7.8  | Comparison between the thrust forces obtained using VOS, ALE and experimental results | 131 |
NOMENCLATURE

\( t \) ( ) Quantity occurring at time \( t \)

\( t \) ( ) Quantity referred to time \( t \)

\( a \) Element incremental material acceleration vector

\( a_i \) Components of referential material acceleration vector

\( a_i \) Components of incremental referential material acceleration vector

\( B^{A1} \) Element shape function derivative matrix related to first convective stiffness

\( B^{A2} \) Element shape function derivative matrix related to second convective stiffness

\( B^{E1} \) Element shape function derivative matrix related to Eulerian material stiffness

\( B^{E2} \) Element shape function derivative matrix related to Eulerian geometrical stiffness

\( C \) Element elastic-plastic material constitutive matrix

\( C^{A1} \) Element first convective velocity-stiffness matrix

\( C^{A2} \) Element second convective velocity-stiffness matrix

\( C^{A3} \) Element third convective velocity-stiffness matrix

\( C^{A4} \) Element fourth convective velocity-stiffness matrix

\( C_{ijkl} \) Components of fourth order material constitutive tensor

\( c \) Specific heat

\( D_{kl} \) Components of rate of deformation tensor
\( \vec{D} \)  
Effective strain rate

\( \vec{D}^p \)  
Effective plastic strain rate

\( 'dS \)  
Elemental surface area

\( 'dS^C \)  
Contact surface area

\( ds \)  
Thickness of small area in the filling pattern scheme

\( 'dV \)  
Elemental volume

\( E \)  
Young’s modulus

\( E_T \)  
Tangent modulus

\( _t e \)  
Element strain vector

\( _t e_{ij} \)  
Components of strain tensor

\( 'f \)  
Element internal force vector

\( 'f^{\text{ext}} \)  
Element internal force vector

\( f \)  
Fractional volume value

\( f_{cr} \)  
Critical fractional volume value

\( 'f \)  
Arbitrary function at time \( t \)

\( 'f' \)  
Material time derivative of arbitrary function

\( 'f'' \)  
Grid derivative of arbitrary function

\( 'f^B_i \)  
Components of body force per unit mass

\( 'f^S_i \)  
Components of surface traction vector
$f_i^C$ Components of contact traction vector

$H$ Element interpolation matrix

$h_k$ Element shape function at node $k$

$h$ Convection heat transfer or film coefficient

$K^A$ Element convective stiffness matrix

$K^{A1}$ Element first convective stiffness matrix

$K^{A2}$ Element second convective stiffness matrix

$K^E$ Element Eulerian stiffness matrix

$K^{E1}$ Element Eulerian material stiffness matrix

$K^{E2}$ Element Eulerian geometrical stiffness matrix

$k$ Conductivity coefficient

$M^A$ Element convective mass matrix

$M^E$ Element Eulerian mass matrix

$N$ Number of nodal points

$N^I$ Nodal shape function at free surface node $I$

$n$ Unit outward normal vector

$n_i$ Components of unit normal vector

$q$ Rate of heat generation per unit volume

$r_i$ Free surface directional vector

$S^{A1}$ Element stress matrix related to first convective matrix

$S^{E2}$ Element stress matrix related to second convective matrix

$S$ Total thickness of free surface area in the filling pattern scheme
\( S_y \)  Yield stress

\( T \)  Temperature

\( t \)  Time

\( u \)  Element incremental material displacement vector

\( u^g \)  Element incremental grid displacement vector

\( u_i \)  Components of incremental material displacement vector

\( u_i^g \)  Components of incremental grid displacement vector

\( V \)  Element total volume

\( v \)  Element incremental material velocity vector

\( v^g \)  Element incremental 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

\( ^t \nu_i \)  Components of material acceleration vector

\( ^t \nu_{i,k} \)  Components of nodal incremental material velocity vector

\( \overline{v}_i \)  Average velocity

\( v^p \)  Penetration vector

\( v^K \)  Corrective velocity constraint at free surface point \( K \)

\( ^t W^{ext} \)  External work

\( w \)  Width of small area in the filling pattern scheme
$X_i^m$  Components of material point initial coordinates

$X_i^g$  Components of grid point initial coordinates

$'x_i$  Components of material point position vector

$'\bar{x}_i$  Components of nodal coordinate vector

$\Delta s$  Element edge length

$\Delta t$  Time increment

$\delta$  Virtual or variation

$\varepsilon$  Equivalent strain

$\nu$  Poisson’s ratio

$\theta$  Free surface angle

$'\rho$  Material density

$'\sigma$  Element stress vector

$'\sigma_{ij}$  Components of Cauchy stress tensor

$'\sigma_{ij}^T$  Components of Truesdell stress rate tensor

$\bar{\sigma}$  Effective stress
ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation for my advisor Dr. Mohamed S. Gadala for his continuous support, encouragement and guidance throughout the stages of this research and my course work. I would be in debt for his invaluable support for all of my future academic life.

I would also like to thank the members of my examining committee, Dr. Yusuf Altintas, Dr. Farrokh Sassani and Dr. Akram Alfantazi for their helpful discussion and their valuable time in reviewing this work.

A special thank for Dr. A. M. Arif for his support and his knowledge which sparked my interest in the area of Finite Element Method (FEM). I also express my gratitude for Dr. M. Sunar who served as my advisor during my Master’s degree who influenced me greatly during the early stages of my academic career. I am also grateful for the support of Dr. Y. Al-Nassar and for his encouragement.
DEDICATION

To my mother for her continuous love and support throughout my whole life. To my father who never stopped believing in me and for his guidance and help during all of my academic life.

I will always be in debt to them and will try to make them proud for the rest of my life…
CHAPTER 1

INTRODUCTION

1.1 Motivation

Manufacturing, in its broadest sense, is the process of converting raw materials into products. It encompasses the design and fabrication of goods by means of various production methods and techniques. Manufacturing processes includes a wide variety of metal forming and machining applications, each with a different problem definition and running conditions.

Before numerical methods became an important tool in solving problems in the field of engineering mechanics, attaining an exact solution for metal forming applications was quite difficult to accomplish due to the complexity of these applications. Pairing theoretical solutions with empirical relations did help in the progress of this field but the problem was that these solutions were limited to basic cases with simple geometries and material models. Even if some of those solutions were tabulated for a range of different material properties and geometrical variables, they were not very practical, especially in prototype design stages. Moreover, empirical relations are usually derived based on experiments performed under certain conditions and it is not clear how changing some of the parameters would impact the accuracy of these relations. Therefore it was clear that additional tools are needed in order to obtain broad and reliable results for metal forming applications.

The vast evolution in numerical techniques and Finite Element Methods (FEM) in the early 1980’s resulted in an innovative advancement in the metal forming industries. The ability to simulate different metal forming processes while covering different operating conditions, geometrical setups and material models reduced the time and cost of having to perform a different experiment for each set. Once the solution from a numerical simulation is verified by experimental or available theoretical/empirical results, it can be used in the prototype design by studying the effect of the process.
conditions on the final product in addition to predicting potential forming defects. Numerical simulations can also be used to achieve a better and more comprehensive understanding of the process mechanical properties, effect of the forming loads on the final geometry of the workpiece and the development of residual stresses in addition to different damage and failure concerns depending on the process.

Even with the ongoing significant improvement in numerical techniques and FEM’s, metal forming simulations still face some problems in terms of accuracy and convergence problems for certain applications. Additionally, it is still very challenging to develop a general FEM approach that can be used to simulate different processes with the same level of reliability without having to simplify the problem on the expense of the accuracy.

1.2 Metal Forming Processes

Three classes of manufacturing processes are of interest in this work; bulk deformation processes, sheet metal forming and metal cutting. In bulk deformation applications (extrusion, compression between dies, coining…etc), the shape of the workpiece is changed by plastic deformation under loads enforced by tools. Usually the workpiece is confined within a set of dies which controls the final shape of the workpiece. Sheet metal forming (plate bending, deep drawing…etc) applications are usually performed for workpieces with high ratio of area to thickness. The tool, in these types of applications, is what controls the final shape of the workpiece by pressing against it forcing it to take a certain shape. Metal cutting is a process that involves removing material from the workpiece. In some cases, the final manufactured products often require additional processing and machining to improve certain aspects such as the surface finish or dimensional accuracy.

All metal forming processes are considered to be complex problems because they involve both geometrical and material nonlinearities. Large deformations are usually considered to be one of the main sources of difficulty when it comes to simulating metal forming. Large boundary motion is also expected in such processes and requires accurate treatment to maintain the shape integrity of the workpiece. Modeling the interaction
between the workpiece and the tools/dies is very important to produce realistic operating conditions. The nonlinear material behavior is also very important in simulating metal forming processes and sometimes affects the interaction between the workpiece and the tool depending on the values of the material properties. Therefore, finite element simulations require large strain formulations that can handle both material and geometrical nonlinearities as well as nonlinear boundary conditions.

1.3 Numerical Methods in Metal Forming

1.3.1 Traditional Large Strain Finite Element Formulations

Two numerical formulations are commonly used in finite element (FE) simulation of large deformation solid mechanics problems: the Lagrangian and the Eulerian formulations. Each has its own advantages and disadvantages. One of the earliest large strain finite element formulations can be traced back to Hibbitt et al. [1] which was referred to as Total Lagrangian formulation. The method was later developed into a modified version named Updated Lagrangian formulation by McMeeking and Rice [2]. The basic difference between the two formulations is in the reference configuration. In Total Lagrangian, the initial configuration of the problem is used in the linearization of the incremental equation of motion. On the other hand, Updated Lagrangian uses the last calculated configuration as a reference for the calculation at the next incremental step. Generally, both formulations should provide the same results. The choice between the two formulations mostly depends on their relative effectiveness and numerical efficiency [3]. In Lagrangian formulation, the finite element mesh is attached to the material points and possesses the same velocity of the material, therefore undergoing the same deformation. Tracking the free surface is easily achieved since the material is attached to the mesh point but this leads to mesh distortion and convergence problems in applications that undergo large deformations of the material. Figure 1.1 shows a simple punch indentation problem where a rigid tool presses against the workpiece. The deformation of the material using Lagrangian formulation can be seen in Figure 1.1a.
To remedy the mesh distortion problem in Lagrangian formulation, researchers and commercial software programs have relied on ‘Re-Meshing’ procedures [4]. In this technique, the whole domain (or part of it) of the problem is re-meshed after a specified number of solution steps and/or after some mesh distortion criteria has been reached. Then, the program transfers problem parameters such as stress and strain to the new mesh through interpolation schemes. This re-meshing approach has other major drawbacks. One problem is that it is time consuming because of the data transfer process and the calculation of all element and global equations for the new mesh. Another problem is the accuracy degradation due to the transfer of the material point properties and solution variables unless the interpolation scheme is chosen properly to ensure accurate and consistent data transfer from the old mesh to the new one. Another problem in using Lagrangian formulation is the difficulty in implementing nonlinear boundary conditions that change during deformation.

**Figure 1.1** Initial and final configuration of a punch indentation problem using (a) Lagrangian formulation and (b) Eulerian Formulation
The Eulerian formulation is suitable for problems with fluid-like material behavior, as in metal forming applications. The adaptation of this formulation, which is commonly used in the area of fluid mechanics, to large strain and metal forming problems has been used and presented in the literature [5-8]. Figure 1.1b shows the final configuration of the punch indentation problem using Eulerian formulation. The finite element mesh is spatially fixed and the material is allowed to flow freely within the fixed domain, hence no mesh distortion occurs. However, the Eulerian approach suffers from the difficulty in handling deformation dependent boundary conditions and in tracing history dependent properties as in the case of plasticity. Another drawback of this formulation is its incapability in tracking the free boundaries and modeling the unconstrained flow of the material within the fixed mesh causing elements to be partially occupied by the material. This drawback makes this approach limited to the simulation of steady-state processes, excluding the initiation of the process.

Different attempts were made to overcome this problem of the free boundaries by coupling the Eulerian formulation with a scheme that can handle the unconstrained flow of the material. One of the most popular methods in the literature is the pseudo-concentrations technique [9-13]. The pseudo-concentrations technique is based on assigning a pseudo-concentration ($c_i$ for $i = 1, \ldots, N$) for each node that defines the presence of the material in any node ($c_i > 0$ indicates the presence of the material at node $i$, $c_i < 0$ indicates the absence of the material at node $i$ and $c_i = 0$ means that node $i$ is on the free surface). The pseudo-concentrations are then interpolated between the nodes within each element to estimate the free surface which requires a smoothing algorithm. One of the drawbacks of this method is that the formulation depends on the number of free surfaces, which should be specified before deriving the finite element equations of this method. This means that a different formulation is required for each application with a different number of free boundaries. The method also cannot handle the contact between the material and another body within the Eulerian domain. The contact edge must be modeled prior to the simulation with the second contact body being modeled outside of the Eulerian domain as a rigid body or a Lagrangian mesh. Another drawback is the fact that each node with no present material is actually given an artificial low viscosity rather than excluding it from the analysis. In addition to the nodal interpolations
within each element, this may lead to numerical inaccuracies and convergence problems depending on the size of the problem domain. Other methods include an operator split method [14] and particle level set [15]. Both methods are limited to cases with only uniformly constructed mesh. In addition to that, both suffer from the difficulty in handling the contact within the Eulerian domain.

1.3.2 Arbitrary Lagrangian-Eulerian (ALE) Formulation

It is generally conceived that neither the Lagrangian nor the Eulerian approaches is well-suited for modeling deformation processes involving both large deformations and unconstrained flow of material with free boundaries. Each formulation excels at handling the weak points of the other method since the Lagrangian approach can handle the unconstrained flow of the material by definition, whereas the Eulerian approach does not suffer from the problem of mesh distortion. The Arbitrary Lagrangian-Eulerian (ALE) formulation emerged as a new formulation that combines the merits of both the Lagrangian and Eulerian ones [16, 17].

In ALE formulation, the finite element mesh is neither attached to the material (i.e. Lagrangian) nor fixed in space (i.e. Eulerian) but rather assigned an arbitrary motion independent of the material deformation [18, 19]. As the material deforms, the elements should move in a way that would optimize their shape to minimize mesh distortion while following the material flow on the boundaries to satisfy the boundary constraints that prevents the relative motion between material and mesh points normal to the boundary. This ensures that the mesh would have the same boundary as the material throughout the whole analysis.

The implementation of the ALE formulation in metal forming processes has become a major tool in the simulation of many applications. Different rolling and extrusion cases were tested using ALE and compared with results from the literature to validate the efficiency of the method once a proper mesh motion scheme is implemented [20-22]. Also ALE was used in simulating more complex and advanced applications such as the metal cutting process [23-25]. Some attempts have been made to employ ALE
formulation in 3D applications [26, 27], although these attempts utilize an operator split approach where the mesh motion is uncoupled into an Updated Lagrangian step and an Eulerian step rather than “true” coupled ALE approach [19-22].

In contrast to the Lagrangian re-meshing technique, in ALE, the material point variables are transferred from the grid points to the material points using proper continuum mechanics equations and therefore the accuracy of the results is not compromised by this transformation. The ALE formulation entails, however, some implementation difficulties and drawbacks. One of these difficulties is to devise a proper mesh motion scheme for each application since a generalized procedure for a class of applications is difficult to attain. This makes the efficiency of the ALE approach dependant of the mesh motion scheme advised by the user and the type of application used in the simulation. Other problems include the difficulty in modeling contact in metal forming simulations, the inherent convergence problems that comes from the complexity of implementing consistent and coupled ALE formulations especially in dynamic applications, and the fact that the number of unknowns is increased, enclosing both the material and mesh displacements [28].

1.3.3 Meshless Methods

A new family of numerical computational methods, called meshless methods [29], emerged in the last two decades to mainly overcome mesh related problems in simulating metal forming processes. These methods are based on a constructed domain of nodes that represents the material without the need for any elements. The analysis is carried by solving partial differential equations on the domain of the irregular distribution of nodes. Although these methods eliminate any difficulties in dealing with the mesh that arise from using finite element methods, they do suffer from several drawbacks.

The main issue with this approach is that the number and initial distribution of nodes have a great influence on the final shape of the workpiece. Following the unconstrained flow of the material without any elements is very difficult to accomplish, especially in problems with very large plastic deformations like the process of backward extrusion
Guan et al. [31] studied the effect of increasing the number of nodes on the final deformed shape of the workpiece. A major drawback observed from this method is the material volume loss during the simulation which means that a large number of nodes are required to minimize the volume loss. Since the method does not have any elements, a special interpolation process must be applied to present a smooth distribution of the data in addition to imposing essential boundary conditions to keep the outer boundary intact during the analysis [32]. Nodes must be laid out properly at the beginning of the simulation when dealing with curved tools and dies. Since there are no elements, there must be a node at each corner and center of a curve to have an accurate presentation of the shape [33, 34]. Different attempts were made to develop a coupled FE-meshless approach [35, 36]. The coupled approach starts with a regular finite element setup and when part of the mesh reaches a critical stage or gets distorted (based on a condition defined by the user) it gets converted into a meshless region and remains that way for the duration of the simulation. Although this approach is more efficient than complete meshless methods, it still shares many of the difficulties and drawbacks.

In summary, meshless methods are usually limited to either steady-state metal forming processes or very confined bulk deformations (e.g. closed die forging). That is due to the fact that the contact boundary conditions must be defined and applied prior to the simulation otherwise the method will not be able to properly follow the unconstrained flow of the metal. Additionally, the number of nodes not only affects the accuracy of the results but also the final shape and the volume loss of the material.

1.4 Scope of Research

Numerical simulation is a powerful and effective tool in simulating metal forming applications, particularly complex systems. The use of such tools can help in predicting design flaws and identifying problems associated with each process before reaching the prototype testing stage. A wide range of defects can be detected using numerical simulations. In bulk deformation processes, such defects include surface cracking, internal cracking, buckling and alligating among others. In sheet metal forming, factors
that can affect the process include the elongation of the yield point, development of residual stresses, springback effect and wrinkling. In metal cutting, the chip curl is a very important subject and requires many test/simulations to study. Excessive curling may cause the chip to fold into the uncut part of the workpiece or into the cutting tool which will cause problems due to the entanglement. Studying the temperature distribution is also very important for failure and fatigue analysis of the final products. All of the mentioned defects and factors can be studied and observed with the help of finite element analysis and numerical simulations.

It is clear that traditional finite element formulations proved to be incapable of simulating metal forming processes. That led to the derivation of the ALE formulation combining the strength of both approaches while eliminating most of their drawbacks. On the other hand, combining two different formulations did raise several difficulty issues when using ALE. Devising a proper and robust mesh motion scheme for each application is not an easy task to attain, especially for non-steady-state processes.

One of the very well established free surface tracking schemes in the literature is the Volume of Fluid (VOF) method [37]. The method was presented to be used with Eulerian formulation to capture the free surface of the fluid by calculating the fractional volume value of the fluid at each cell, then constructing the free surface based on these values. In this work we are proposing the use of the Eulerian formulation in simulating metal forming processes with the implementation of a tracking scheme that is based on the VOF method. The resulting method, to be called “Volume of Solid (VOS) method”, is to be used in simulating large deformation and metal forming applications. As mentioned above, the use of Eulerian approach has the advantage of not dealing with mesh distortion and eliminating the need of modeling a proper mesh motion scheme. The implementation of the VOS method will overcome the free boundary problem by properly tracking the location of the free surface of the solid.

A critical review of the VOF method is presented in Chapter 2 highlighting the important development of the method, since it was first presented by Hirt and Nichols [29], and the different approaches in constructing the free surface. This gives a better
understanding of the method which lays the foundation behind transforming the method from fluid mechanics to solid mechanics. Based on the principle of virtual work and continuum mechanics, a complete derivation of the Eulerian formulation is presented in Chapter 3 for quasi-static and dynamic analyses. Treatment of the convective terms is also included in the formulation. A section on heat transfer analysis is also included in this chapter.

The proposed Volume of Solid (VOS) method is discussed in details in Chapter 4. The discussion follows three major steps in the formulation of the VOS method; calculating the fractional volume value of the solid, defining the location and orientation of the free surface, and constructing the free surface. The method is first presented for uniformly constructed Eulerian grids. After that, a new scheme is presented to generalize the method for applications with non-uniform Eulerian grids. Other important features of the method are also discussed including tracking material point related properties and updating newly added nodes in the analysis.

Chapter 5 covers topics related to the implementation of the VOS method. A simple in nature yet effective averaging process is incorporated in the method to avoid any smearing of the data and to eliminate the need for any smoothing function. A contact algorithm is presented for rigid-flexible contact cases. The algorithm allows the modeling of a moving rigid body within the Eulerian domain and tracking the change of the contact boundary conditions at the free surface of the solid during the course of the simulation. A summary of the method and its implementation in a 2-D computer code is presented at the end of the chapter.

In Chapters 6 and 7, various metal forming and machining applications are simulated using the proposed VOS method. The results are compared to experimental measurements and/or numerical results from the literature. The cases simulated in these Chapters covers a wide range of processes to highlight different aspects of the VOS method. Finally, Chapter 8 concludes this work with a summary of the method and a discussion on its implementation in solid mechanics and metal forming problems.
CHAPTER 2
SURFACE TRACKING METHODS

The use of Lagrangian or ALE formulations in computational fluid dynamics (CFD) problems was a common choice of formulation due to the simple fact that the fluid surface is confined within the elements at all times. This led to relatively straightforward computations. However, these formulations still have difficulties dealing with mesh distortion and complexity in the shape of the outer surface in some problems due to the folding and merging of the surface. With the difficulties in using Lagrangian and ALE, Eulerian formulation became the main tool in modeling CFD problems.

Even though the fluid flows freely within a fixed mesh, it is customary to identify the fluid in an Eulerian mesh as elements on which computations are performed. The fluid moves between elements based on the velocity calculations performed at each step. Therefore, the fluid flow requires averaging of the flow properties for the fluid which may lead to some smearing and discontinuity in the free surface, and that is where the need for a special treatment of the free boundaries arises. Free surface tracking schemes are necessary to have an acceptable simulation of the fluid flow and representation of the free surface. The goals of these schemes are to be able to identify Eulerian cells that contain a free surface, describe the locations and shape of the free boundaries, enforce desired surface boundary conditions on the computational mesh and the newly constructed surfaces, and finally to compute the advancement of the free surface after each step.

2.1 Traditional Free Boundary Methods

When using an Eulerian mesh, the problem of the free surface is still a concern, but this type of formulation allows different free boundary capturing/tracking schemes to be incorporated into the main formulation to capture the free surface shape. At the same
time, this formulation can model the complex shape of the fluid free surface. Using Eulerian formulation, free boundary tracking schemes may be categorized into two groups: surface and volume tracking methods.

One of the earliest and simplest surface tracking methods is the Height Functions [38]. In this method, the free boundary is defined by its distance from a reference line as a position function along that reference line. This means that the method can only be used for a vertical or horizontal direction at a time. As a standalone scheme, Height Functions are very limited to simple shapes and small slope changes. A more generalized form of the Height Functions is the method of Line Segments [39]. This approach uses a series of short lines, or point, that are connected by line segments. Compared with Height Functions, Line Segments present a better resolution of the free surface but on the expense of the computational cost.

Another surface tracking method is the Marker Particles [40], which, instead of defining the free surface directly, defines particles at the fluid-occupied regions in the surface with each particle specified to move with the fluid velocity at its location. A free surface is identified when a mesh element contains markers but neighbors an element with no markers. Additional computation must be performed for the location and intersection of the free surface with the element.

The main problem with surface tracking methods occurs when there is an intersection between two or more surfaces. Additionally, these methods are not able to handle cases where the fluid exhibits severe surface folding and merging over on itself without undergoing rigorous mesh distortion. Another problem with surface tracking methods is that they require large computer storage to store all the point coordinates.

Volume tracking schemes are based on defining a tracer or a marker particle at each fluid region of the domain and not only at the surface. One of the most successful methods in this category was the Marker-and-Cell (MAC) method [41]. It was the first method to use pressure and velocity as the primary variables. The method is based on employing a distribution of marker particles all over the fluid region, and set free surface pressures at the centers of the surface cells. The method was improved with time to
include the pressure boundary condition at the location of the boundary within the cell that defines the free surface. The MAC method still requires large computer storage, plus it experiences a problem with non-natural creation of high and low marker number densities due to the irregularity of the flow field in some cases.

2.2 Volume of Fluid (VOF) Method

2.2.1 The VOF Method

The Volume of Fluid (VOF) was first introduced by Hirt and Nichols [37] based on the MAC method, but with major improvements. VOF is based on defining one new variable, fractional volume value $f$, for each element. The fractional volume is set to have a value of 1 when the mesh cell is fully occupied by the fluid, 0 when the cell is empty with no fluid and a value between 0 and 1 when the cell is partially occupied by the fluid which represents a free surface (Figure 2.1).

![Figure 2.1](image)

*Figure 2.1 Distribution of ‘f’ at free surface and neighboring elements*
The VOF method is based on three major steps. First is to calculate the fractional volume $f$ of each cell at each time step. These fractional values are stored for each element and are updated after each time step to check if the state of an element has changed (e.g. fully occupied to partially occupied cells). The second step is to predict the shape of the free surface at the boundary cells. It is very important to distinguish between the location and the direction for the free surface at this step because for each location and fractional value combination, more than one free surface shape possibility exists. Finally, to construct the free surface based on the values of the fractional volumes and the shape of the free surface. The natural computational domain for each element is a 3x3 stencil with the element of interest being in the center of that stencil. Different VOF schemes utilize the 3x3 stencil in different manners. Some of the less common schemes use a different stencil size which will be discussed in the next section. The beauty of this method is that it does not depend on specific variables such as the fluid pressure, which allows for different techniques to be used in the second and third steps when trying to predict and capture the shape and location of the free surface.

2.2.2 VOF in the Literature

With the use of the VOF method, the free surface problems became easier to deal with in fluid mechanics applications, but there are still some important points to consider with this approach. The main point is the smearing of $f$ due to the averaging process and discontinuity which leads to loosing the interface between the elements [37]. This was particularly noticeable in the early development stages of the VOF method when a simple line interface calculation (SLIC) was utilized for free surface construction. In SLIC, the free surface interface is reconstructed using a sequence of segments that are aligned with the grid (either vertically or horizontally).

The SLIC-VOF reconstruction method is relatively crude and would cause severe smearing of the fractional volume distribution. Therefore, the piecewise linear interface construction (PLIC) method was introduced to reduce the smearing and to smooth the representation of the free boundary. One of the critical aspects of the PLIC method is that
it does not perform the surface reconstruction as a series of joined segments for the whole free boundary, but as a discontinuous chain of segments with relatively small discontinuities. Figure 2.2 shows the averaging of the free surface using: (a) SLIC and (b) PLIC [42]. An improvement of the PLIC-VOF was suggested by Martinez et al. [43] to improve the curve of the constructed free surface segments. By incorporating the calculation of the free surface tension forces into the PLIC-VOF equations, a better curve estimation can be achieved. The use of this improved curvature technique eliminated the use of smoothed color functions to connect the chain of segments at the free surface.

Figure 2.2  Free surface averaging using (a) SLIC and (b) PLIC

Because of the smearing problem and the difficulty in defining exactly the free surface, different free surface capturing techniques have been developed and implemented in the VOF method. One of the common methods in literature is the donor-acceptor method which is used to overcome smearing and to represent the free surface in a smooth manner [37, 44-47]. The concept behind the donor-acceptor method is that for each interaction between neighboring elements, one acts as a donor element as the fluid flows out of that element and the other one as an acceptor as the fluid flows into it. This
donor-acceptor scheme is achieved by calculating the change in the fluid volume fluxes between each cell and its neighbors, then average those values so that a sharp interface at the free boundaries can be maintained. Based on the distribution of \( f \), and with the help of an orientation vector that points towards the free surface, the shape and location of the free surface may be determined.

The donor-acceptor method can be also implemented in conjunction with a SLIC or PLIC or any other surface construction scheme. Nobari et al. [48] worked on a modified definition of the donor-acceptor method called the defined donating and accepting region (DDAR) with a SLIC-VOF scheme. In their work they developed an algorithm that uses a fully multi-dimensional boundary flux integration technique. The algorithm predicts the VOF flux to not only vertical and horizontal neighboring elements, but also to diagonal neighboring elements. The results obtained from their work indicate that it is necessary to predict the VOF advection for all of the eight neighboring elements to achieve maximum accuracy. The modified SLIC-VOF 8-node advection calculation lead to accurate results compared to some of the conventional PLIC-VOF methods. A major observation of the donor-acceptor method is that it is very difficult to implement in non-uniform grids, which limits it mostly to cases with uniformly constructed grids.

Another interesting scheme when using the VOF method is the implementation of height functions in the construction of the free surface. Height functions have been used in 2D [49] and 3D [50] cases as well. In most VOF methods, a 3x3 stencil is used as the computational domain where the element of concern is located at the center and is neighbored by 8 elements. When height functions are employed, a 7x3 or 3x7 stencil is used for the calculation of the fractional volume and the free surface orientation. The idea is to calculate the normals of the free surface based on the horizontal and vertical fluxes, then, based on these values, if the vertical normal is bigger than the horizontal one, a 7x3 stencil is considered and vice versa. Fractional volumes are then summed along the direction that is more normal to the interface. Some methods start with the common 3x3 stencil to examine whether the initial estimate of the normal to the free surface is more horizontal or vertical before implementing a 3x7 or a 7x3 stencil [51]. This method has the advantage of having accurate curvatures and surface tension forces at contact lines in
addition to enforcing a contact angle boundary condition at a contact line because they utilize seven columns/rows instead of three. The use of the height functions is, however, more complex than using regular 3x3 stencils, and mesh refinement at the contact interface is usually needed to insure accurate results. In addition to that, height functions may lead to poor results if the stencil configuration includes a free surface other than the interface located at the element of interest. To overcome this problem, any partially occupied element (other than the element of interest) located in the 3x7/7x3 stencil is assumed to have a fractional volume value of one at that instance. After the computation is done for that stencil, the fractional volume value of that element is changed back to its original value. This scheme is also limited to uniformly constructed grids since elements of a 3x7/7x3 stencil will not be aligned on the same level in non-uniform mesh.

The use of mesh refinement at the free surface is utilized in many VOF methods to minimize the need for smoothing color functions or modified free surface calculations. The approach is an adaptive grid refinement. The idea is to refine the mesh at the free surface by dividing each free surface cell (parent cell) into four child cells (also called baby or leaf cells), then keep dividing each free surface child cell into another four child cells until a user specified mesh refinement criterion is met [51]. Using adaptive grid refinement leads to better shape and more accurate curves at the free surface interface, depending on the level of cell division used in the computation. Mesh refinement can also be used with any other scheme (e.g. VOF-PLIC, donor-acceptor…etc) but it suffers from the same problems of the pertinent method and it is difficult to implement it into a non-uniform mesh. The reason behind this is because the element division process is usually set prior to the computation to a certain size which cannot be set for irregular element since they have different sizes. Additionally it is very challenging to transform the free surface interface from the parent cell to the child cells for a random grid. The algorithm is usually developed in a systematic fashion to ensure the same level of accuracy along the interface [52]. One of the main drawbacks of using adaptive grid refinement is that with every division step we end up with what is called “hanging nodes” at the faces between neighboring elements of different sizes, in which a corner node from the child cell lies at the face-center of a larger cell (Figure 2.3). This is a problem because the connectivity of the FE is no longer satisfied. To overcome this issue, the larger cell is considered to have
two faces neighboring the child cells. Then, linear interpolation is used to estimate the values of the fluxes at the face-center in order to divide the fluid flow from the bigger cell into the smaller cells accurately. Although mesh refinement produces better results regarding the shape of the free surface, it is on the expense of the process variables computational accuracy since these variables must be interpolated back and forth between the original grid elements and the newly constructed ones[53].

Another approach presented in the literature to capture the evolution of the free surface interface with the VOF method is to use an Eulerian-Lagrangian advection scheme [54]. In this method, the advection of the free surface interface is carried out using a Lagrangian step, and then the fractional volume values are calculated for the Eulerian grid. Based on the values of the fractional volumes, the fluid is remapped from the Lagrangian grid to the Eulerian grid. Mashayek and Ashgriz [55] discretized all the
cell wetting possible cases between the element of interest and its neighboring elements. Based on these cases, the nodes at the outer boundary get altered in a Lagrangian fashion to capture the fluid surface. Although this approach, by definition, can be easily implemented into non-uniform grids, it is unfeasible to summarize all the possible cases for an irregular grid. Later, Ashgriz et al. [56] developed a more general approach by moving the free surface elements, and not just the surface nodes, in a Lagrangian sense. The mesh does not get altered in this approach; therefore the interface after the advection step has to be mounted on the Eulerian grid. The fractional volume values must be determined at each neighboring element after the advection step. The free surface interface is constructed geometrically by knowing the coordinates of the Lagrangian elements after the advection and the original Eulerian elements. This modification allows this approach to be used in cases with non-uniform grid.

Gueyffier et al. [57] developed a free surface interface tracking scheme for three dimensional flows using PLIC-VOF with a split Lagrangian advection scheme. They used a Lagrangian step for the advection of the free surface interface. Then, a PLIC technique is used to construct the free surface of the Eulerian grid based on the intersection between the Lagrangian and the Eulerian grids. Surface tension was implemented using the continuous surface stress or force method. This combination of the PLIC-VOF with the modified split Lagrangian scheme is very complex compared to other methods but can be upgraded to three dimensional problems without any significant increase in the difficulty.

An Unsplit Lagrangian Advection (ULA) scheme for the VOF method was presented by Yang et al. [58]. Instead of using the operator split method, where more than one interface reconstruction might be required at each step, a complete and true Lagrangian advection is utilized in this approach. When the interface from one cell moves into the neighboring cells, then the flux between those cells is evaluated geometrically. In the ULA scheme, instead of calculating the outflow and inflow fluxes for the elements separately, the outflow flux for any cell is considered as an inflow for another cell. This way the conservation of mass is not compromised during the simulation.
Janssen and Krafczyk [59] presented a hybrid VOF algorithm that combines the advantages of different schemes. They used a PLIC-VOF with mesh refinement in conjunction with an averaging process called cell fill level approach. The goal of the averaging process is to determine, without using geometrical surface reconstruction, the filled face level (i.e. the fraction of a cell face that is wetted by the fluid) of a cell by averaging the fill level of the two cells between which fluid mass exchange takes place. This approach ensures mass conservation between two free surface elements but not between free surface and fully occupied elements. To overcome this issue, a case distinction is used to set the face fill level value to one if either of the two cells is a fully occupied, otherwise and average value is used between them.

One of the common methods in computational physics is the level set (LS) method [60]. In recent years the coupling between the VOF and LS methods became very popular in CFD problems [61-63]. In the LS method, a smooth function (LS function) is used for the free surface interface representation. Generally, the LS function is defined as a signed distance function based on the shortest distance from a reference point to the interface where the sign indicates if the surface is continuous or dispersed. The function is reinitialized after each time step to update the location of the new interface. In the hybrid VOF and LS method, the LS function is used to calculate the interface normal and curvature, where VOF is used for the fluxes calculations and the free surface interface construction. The VOF method corrects the volume loss produced after reinitializing the distance after each step, where the LS function improves the accuracy of the interface curvature [64, 65].

Another approach used in the VOF method is the filling pattern [44, 45]. In this method the filling pattern follows the estimated direction of the free surface and starts gradually filling the cell with the fluid until it reach the desired fractional volume value. The filling scheme is very popular in the simulation of mould filling type of problems [66] since the filling pattern is, to some extent, steady and follows the same path during the whole process. Jeong and Yang [67] generalized the filling pattern to be used in more complex problems. They classified the free surface interfaces into eight different cases that cover all general possibilities. Each case is assigned a unique pattern depending of
the shape of the free surface. The filling process then fills the elements according to its pattern until the element is filled with the desired amount of fluid. To improve the results, an adaptive mesh refinement was employed in addition to a free surface smoothing function. Since the filling pattern is only used for the construction of the interface, it does not prevent the use of other VOF schemes which makes this approach very easy to modify and improve.

### 2.3 Transition from VOF to VOS

One of the main reasons that the VOF method has been used extensively in the literature is because it allows for a wide range of schemes to be incorporated within its formulation. The main goal of this research is to apply a method that can handle the free surface of the unconstrained flow of the material in an Eulerian FE mesh for large deformation applications. The VOF method is used for the same purpose in CFD analyses; hence we propose the adaptation of the method into solid mechanics and therefore, would be named Volume of Solid (VOS) method.

Rather than having a literature review of the VOF method and its development in a chronological sense, a general study of the various schemes and modifications was presented in the previous section. Understanding the advantages and disadvantages of the VOF methods presented in the literature is the basic ground for this work. Some of the methods might be superior to others in the field of fluid dynamics but that is not necessary true when adapting VOF into solid mechanics. Due to the difference in the mechanics between metal forming applications and fluid dynamic problems, certain methods can be utilized in a more advantageous way. Understanding the dynamics of metal forming would also lead to new modifications in the VOF method that may improve the efficiency of the method.

In Chapter 4, a detailed discussion of the proposed Volume of Solid (VOS) method is presented. The calculation of the fractional volume values is presented in a similar way to the available VOF method in the literature and no major modification is needed when transitioning to VOS. A new method for identifying the direction and orientation of the
free surface interface is proposed in this work by fully utilizing a 3x3 stencil in a way similar to the height functions. The construction of the free surface is developed for a uniformly constructed Cartesian grid and for a general case of non-uniform grid. For the uniform grid case, a classification of all the possible free surface shapes is used with a PLIC scheme whereas a modified filling scheme is developed for the non-uniform case. The implementation of the VOS method is presented in Chapter 5 with a discussion on all related issues such as the connectivity of the free surface, handling the contact and tracking the material point related properties.
CHAPTER 3

EULERIAN FORMULATION

In this section we present the governing Eulerian virtual work equilibrium equations used in finite element analysis of quasi-static and dynamic systems. The derivation will follow the standard notation used by many researchers, e.g. [3]. The derivation presented in this work is based on the ALE formulations developed by Bayoumi and Gadala [68, 69].

In any finite element formulation there are two sets of configurations, one refers to the material particles and the other one refers to the grid points. The two sets share the same velocity in Lagrangian formulation and therefore one set of equations is required. In ALE formulation, since the mesh moves independently from the material, the two configurations have different velocities and, therefore, two sets of equations are required. Since we are using Eulerian formulation in this work where the grid is fixed in space, and hence has a zero velocity, we only consider the material velocity.

When deriving the Eulerian equations, time derivative of several quantities will be required. There are two different time derivatives in finite element analysis, one is the material time derivative and the other is the grid time derivative. The material time derivative is required to track the change in the function with time for the material point. For any arbitrary function $f$, the material time derivative is defined as the rate of change of the function while the material particle is fixed:

$$
\dot{f}_m = \frac{\partial f}{\partial t} \bigg|_{X_i^m}
$$

(3.1)

Another definition for the function rate of change is required as the grid point is held fixed. This is essential to track the value of the function within the grid as material particles move with time. The grid time derivative is defined as the rate of change of the function while the grid point is fixed:
\[ \dot{t} f = \frac{\partial t f}{\partial t} \bigg|_{X^t} \quad (3.2) \]

The relation between the material and grid time derivatives can be written by simply defining the progression of function \( f \) according to the following transport equation:

\[ \dot{t} f = \dot{t} f + \dot{t} v_i \frac{\partial t f}{\partial x_i} \quad (3.3) \]

where \( \dot{v}_i \) is the material point velocity at time \( t \). This relation is similar to the relation used to derive ALE equations [70] except for the grid velocity which is set to zero in Eq. (3.3). The formulation of the Eulerian equation in this work is divided into two sections. In the first section, the Eulerian equilibrium equation is derived for quasi-static analysis which concerns cases where inertia effects may be neglected as in low speed metal forming processes. In the second section, the derivation is extended to dynamic analysis where the effect of the inertia forces is included into the equilibrium equation.

### 3.1 Quasi-Static Analysis

In quasi-static analysis, the natural way to derive and write the equilibrium equation is to use the principle of virtual displacements since the inertia effects are not considered. The equilibrium of the body at time \( t + \Delta t \) using the principle of virtual displacements is written as:

\[ \int_{t+\Delta V}^{t+\Delta t} \sigma_{ij} \delta e_{ij} dV + \delta W^{ext} = \delta W^{int} \quad (3.4) \]

where \( \delta \) represents the virtual or the variation of a variable, \( t+\Delta \sigma_{ij} \) are the Cauchy stress tensor components at time \( t+\Delta t \) and \( t+\Delta e_{ij} \) is the conjugate strain tensor which is defined as:
\[ \tau^{t+\Delta t}_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial t^{t+\Delta t} x_j} + \frac{\partial u_j}{\partial t^{t+\Delta t} x_i} \right) \]  

(3.5)

The external virtual work in Eq. (3.4) is expanded by writing the components of the body force and the surface traction:

\[
\delta^{t+\Delta t} W^e = \int_{V} t^{+\Delta t} \rho^{t+\Delta t} f_i^B \delta u_i^{t+\Delta t} dV + \int_{S} t^{+\Delta t} f_i^S \delta u_i^{t+\Delta t} dS 
\]

(3.6)

where \( t^{+\Delta t} f_i^B \) and \( t^{+\Delta t} f_i^S \) are the components of the body force per unit mass and surface traction at time \( t + \Delta t \), respectively. Once the variables at time \( t + \Delta t \) are computed then Eq. (3.4) can be expanded to get the complete Eulerian equilibrium equation.

### 3.1.1 Incremental Decomposition

Since we are referring variables to the grid configuration, we assume that the value of any variable at time \( t + \Delta t \) is defined to be its value at time \( t \) plus an increment that is calculated by multiplying the grid time derivative by the time increment \( \Delta t \). Following the stated assumption, the material density is written as:

\[ t^{+\Delta t} \rho = t \rho + t \rho \Delta t \]

(3.7)

The conservation of mass at time \( t \) is given by:

\[ t \dot{\rho} = - t \rho \frac{\partial^{t} v_i}{\partial^{t} x_i} \]

(3.8)

Using Eq. (3.8) in Eq. (3.3), the continuity equation is written as:

\[ t^{\rho'} = - t \rho \frac{\partial^{t} v_i}{\partial^{t} x_i} - t v_i \frac{\partial^{t} \rho}{\partial^{t} x_i} \]

(3.9)
Now we substitute Eq. (3.9) in Eq. (3.7) to get the full decomposition of the material density:

\[ t + \Delta t \rho = t \rho - t \rho \frac{\partial u_k}{\partial x_k} - u_k \frac{\partial t}{\partial x_k} \]  \hspace{1cm} (3.10)

Similar to Eq. (3.7), the stress components at time \( t + \Delta t \) are expressed as:

\[ t + \Delta t \sigma_{ij} = t \sigma_{ij} + t \sigma_{ij}' \Delta t \] \hspace{1cm} (3.11)

Using Eq. (3.3), we can expand Eq. (3.11) to the following:

\[ t + \Delta t \sigma_{ij} = t \sigma_{ij} + t \sigma_{ij}' \Delta t - u_k \frac{\partial t}{\partial x_k} \] \hspace{1cm} (3.12)

The material rate of Cauchy stresses \( t \hat{\sigma}_{ij} \) is calculated from the material constitutive relation:

\[ t \hat{\sigma}_{ij} = t \hat{C}_{ijkl} t D_{kl} \] \hspace{1cm} (3.13)

where \( t \hat{C}_{ijkl} \) is the forth order tensor of the material properties, and \( t D_{kl} \) is the rate of deformation tensor given by:

\[ t D_{kl} = \frac{1}{2} \left( \frac{\partial t v_j}{\partial x_j} + \frac{\partial t v_j}{\partial x_i} \right) \] \hspace{1cm} (3.14)

\( t \hat{\sigma}_{ij} \) is an objective stress rate tensor (Truesdell, Green-Naghdi, Jaumann…etc). In our case the Truesdell stress rate will be used based on the comparison presented by Gadala and Wang [71]:

\[ t \hat{\sigma}_{ij} = t \hat{\sigma}_{ij} + \frac{\partial t v_k}{\partial x_k} t \sigma_{ij} - \frac{\partial t v_j}{\partial x_j} t \sigma_{ik} - \frac{\partial t v_i}{\partial x_i} t \sigma_{jk} \] \hspace{1cm} (3.15)
The variation in the strain components \( \delta_{t+\Delta t} e_{ij} \) can be derived in a similar way:

\[
\delta_{t+\Delta t} e_{ij} = \delta_{t} e_{ij} + \delta_{t} e_{ij}' \Delta t
\]

(3.16)

Since the grid in Eulerian analysis is fixed in space with zero velocity, the grid time derivative of the variation in the strain components \( \delta_{t} e_{ij}' \) is equal to zero, so Eq. (3.16) is reduced to:

\[
\delta_{t+\Delta t} e_{ij} = \delta_{t} e_{ij}
\]

(3.17)

The incremental decomposition of the elemental volume and elemental surface area at time \( t + \Delta t \) is derived exactly as the variation in the strain components as there is no change in volume or surface area in an Eulerian element:

\[
^{t+\Delta t} dV = \, \, ^{t} dV
\]

(3.18)

\[
^{t+\Delta t} dS = \, \, ^{t} dS
\]

(3.19)

### 3.1.2 Linearization

After expanding the variables using the incremental decomposition, substituting these decompositions into the equilibrium equation, Eq. (3.4), and neglecting all incremental quantities with higher orders; we can end up with the linearized equation. First we will expand the LHS of Eq. (3.4), by using Eq’s. (3.12), (3.17) and (3.18):

\[
\int_{^{t+\Delta t} V}^{t+\Delta t} \sigma_{ij} \delta_{t+\Delta t} e_{ij} \, \, ^{t+\Delta t} dV = \int_{^{t} V}^{t} \sigma_{ij} \delta_{t} e_{ij} \, \, ^{t} dV + \int_{^{t} V}^{t} \dot{\sigma}_{ij} \Delta t \delta_{t} e_{ij}' \, \, ^{t} dV - \int_{^{t} V}^{t} u_{k} \frac{\partial \sigma_{ij}}{\partial ^{t} x_{k}} \delta_{t} e_{ij}' \, \, ^{t} dV
\]

(3.20)

The external virtual work in Eq. (3.6) consists of two parts, body forces and surface tractions. The surface traction term of Eq. (3.6) can be expressed by Eq. (3.19):
The body force can be expanded and linearized using Eq’s. (3.10) and (3.18):

\[
\int_{S}^{t+\Delta t} f_i^S \delta u_i \cdot \int_{S}^{t+\Delta t} dS = \int_{S}^{t+\Delta t} \rho_i^B \delta u_i \cdot dV
\]

(3.21)

\[
- \int_{V}^{t+\Delta t} \rho_i^B \frac{\partial u_k}{\partial x_k} \delta u_i \cdot dV = \int_{V}^{t+\Delta t} f_i^B u_k \frac{\partial \rho}{\partial x_k} \delta u_i \cdot dV
\]

(3.22)

3.1.3 Treatment of Convective Terms

Convective terms refer to the spatial derivative of variables \( \frac{\partial f}{\partial x_j} \) as the last term in Eq. (3.3). The presence of convective terms in any equation complicates the numerical solution. One of the common methods used to deal with convective terms in Eulerian FEA is the operator split method. In operator split method, Eq. (3.3) would be divided into two equations, non-advective and advective equations. In this work we will modify the method presented by Bayoumi and Gadala [68, 69] for dealing with convective terms in ALE formulation to utilize it in Eulerian formulation. The advantage of this method is that it does not require two sets of equations to be solved separately as in the operator split method, in addition to avoiding the need to compute the spatial derivative of the variables. The first convective term that we have to deal with is the last term in Eq. (3.20) which requires the computation of the spatial derivative of the stresses. Because the stresses are usually evaluated at the integration points rather than the nodal points, the stress field is generally discontinuous along the element edges. This leads to the stress gradients not being reliably computed on the element level when evaluating the element matrices. That is where methods such as the operator split method are used to overcome this problem. In this formulation, we will use Gauss divergence theorem to deal with the convective term. First we re-write the last term in Eq. (3.20) as follows:
\[
\int_V u_k \frac{\partial^t \sigma_{ij}}{\partial^t x_k} \delta_t e_{ij} \, t \, dV = \int_V \delta \left[ u_k \frac{\partial^t \sigma_{ij}}{\partial^t x_k} \delta_t e_{ij} \right] \, t \, dV - \int_V u_k \frac{\partial \delta_t e_{ij}}{\partial^t x_k} \, t \, dV
\]

(3.23)

Since we use the VOS method to construct the interface at the free surface elements, then at that point the material in the new free surface elements will be given the velocity values through the surface nodes, thus making both, the material and the grid, share the same velocity at that instant. This is because in the VOS method we define the free surface within the element cell that contains the free surface, making the material and the grids share the same surface/boundary all times. So to ensure that the material and the grid have the same boundary at all times, we write the following boundary constraint by enforcing a fictitious grid boundary velocity:

\[
\left( t_{V_i} - t_{V_i}^g \right) n_i \bigg|_{\text{on the boundary}} = 0
\]

(3.24)

Now we include the grid displacement term in the first term on the RHS of Eq. (3.23) and apply the divergence theorem with the use of Eq. (3.24):

\[
\int_V \frac{\partial}{\partial^t x_k} \left( u_k - u_k^g \right) \sigma_{ij} \delta_t e_{ij} \, t \, dV = \int_S \left( u_k - u_k^g \right) \sigma_{ij} \delta_t e_{ij} \, t \, n_i \, t \, dS = 0
\]

(3.25)

Substituting Eq’s. (3.23) and (3.25) into (3.20), the internal virtual work is written as:

\[
\int_{t+\Delta t}^t \sigma_{ij} \delta_t e_{ij} \, t \, dV = \int_V^t \sigma_{ij} \delta_t e_{ij} \, t \, dV + \int_V^t \sigma_{ij} \Delta t \delta_t e_{ij} \, t \, dV
\]

(3.26)

Now the other convective term that we have to deal with is the convective body force term which is the last term in Eq. (3.12). Re-writing the body force convective term:
Applying the divergence theorem to the first term on the RHS of Eq. (3.27) and using the boundary constraint in Eq. (3.24) we get:

\[
\int_{\bar{V}} \frac{\partial}{\partial t} \left[ f_i^B \mathbf{u}_k \right] \cdot \mathbf{\tilde{d}} \mathbf{u}_i \, t \, dV = \int_{\bar{V}} \frac{\partial}{\partial t} \left[ f_i^B \mathbf{u}_k \right] \cdot \mathbf{\tilde{d}} \mathbf{u}_i \, t \\ t \, dV - \int_{\bar{V}} f_i^B \mathbf{u}_k \cdot \mathbf{\tilde{d}} \mathbf{u}_i \, t \, dV = 0 (3.28)
\]

Substituting Eq’s. (3.27) and (3.28) into (3.22), the body force is written as:

\[
\int_{\bar{V}} f_i^B \mathbf{u}_k \cdot \mathbf{\tilde{d}} \mathbf{u}_i \, t \, dV = \int_{\bar{V}} f_i^B \mathbf{u}_k \cdot \mathbf{\tilde{d}} \mathbf{u}_i \, t \, dV = \int_{\bar{V}} f_i^B \mathbf{u}_k \cdot \mathbf{\tilde{d}} \mathbf{u}_i \, t \, dV
\]

### 3.1.4 Eulerian Equilibrium Equation for Quasi-static Analysis

To write the final form of the Eulerian equilibrium equation, we substitute the internal virtual work (3.26), the body force (3.29) and the surface traction (3.21) into Eq. (3.4). This gives us the final form of the principle of virtual displacements at time \( t + \Delta t \) referred to time \( t \):

\[
\int_{\bar{V}} t^{\sigma_{ij} \Delta t} \mathbf{e}_{ij} \cdot \mathbf{\hat{d}} \mathbf{u}_i \, t \, dV + \int_{\bar{V}} t^{\sigma_{ij} \Delta t} \mathbf{e}_{ij} \cdot \mathbf{\hat{d}} \mathbf{u}_i \, t \, dV + \int_{\bar{V}} t^{\sigma_{ij} \Delta t} \mathbf{e}_{ij} \cdot \mathbf{\hat{d}} \mathbf{u}_i \, t \, dV
\]

where the external virtual work is:

\[
\int_{\bar{V}} t^{\sigma_{ij} \Delta t} \mathbf{e}_{ij} \cdot \mathbf{\hat{d}} \mathbf{u}_i \, t \, dV = 0 (3.30)
\]
\[ \delta^{t+\Delta t}W_{\text{ext}} = \int_V \rho \left[ \dot{t} B_i + \dot{\sigma}_{ij} \dot{e}_{ij} \right] \delta u_i^t dV + \int_{\partial V} \rho \dot{t} B_i u_k \frac{\partial \delta u_i^t}{\partial \dot{t} x_k} dV + \int_{\partial \Gamma} \dot{t} S \delta u_i^t dS \] (3.31)

Finally we use the constitutive relations introduced in Eq’s. (3.13), (3.14) and (3.15) to the first integral in Eq. (3.30) gives us the final form of the equilibrium equation:

\[ \int_V C_{ijkl} e_{kl} \delta_{ij} \eta_{ij}^t dV + \int_V \dot{t} \sigma_{ij} \delta_{ij} \eta_{ij}^t dV + \int_{\partial V} \frac{\partial u_k}{\partial \dot{t} x_i} \sigma_{ij} \frac{\partial \delta u_i^t}{\partial \dot{t} x_j} dV = \delta^{t+\Delta t}W_{\text{ext}} - \int_V \dot{t} \eta_{ij} \delta_{ij} e_{ij}^t dV \] (3.32)

where

\[ \dot{t} \eta_{ij} = \frac{1}{2} \frac{\partial u_k}{\partial \dot{t} x_i} \frac{\partial u_k}{\partial \dot{t} x_j} \] (3.33)

Eulerian equation (3.32) is linear in the incremental displacement \( u_i \) and is referred to an Eulerian grid since the velocity and displacement of the grid does not show in the equation.

### 3.2 Dynamic Analysis

In this section we expand the derivation of the quasi-static Eulerian equilibrium equation to dynamic analysis by including the inertia effects. By including the inertia forces we will have to deal with the material time derivative of the material velocity \( \dot{t} + \Delta t \nu_i \), and the grid time derivative of the material velocity \( \dot{t} + \Delta t \nu_i' \). As the Eulerian mesh has zero velocity, we end up with the time derivative of just the material velocity, and for
clarity we will refer to the referential material acceleration (grid time derivative of the material velocity) as $t^{+\Delta t}a_i$.

### 3.2.1 Virtual Work Done by Inertia Forces

Using the relation between the material and grid time derivatives in Eq. (3.3), we can write the virtual work done by the inertia forces as:

$$
\int_{V} t^{+\Delta t} \rho t^{+\Delta t} v_i \delta u_i t^{+\Delta t} dV = \int_{V} t^{+\Delta t} \rho t^{+\Delta t} a_i \delta u_i t^{+\Delta t} dV
$$

$$
+ \int_{V} t^{+\Delta t} \rho t^{+\Delta t} v_j \frac{\partial t^{+\Delta t} v_i}{\partial t^{+\Delta t} x_j} \delta u_i t^{+\Delta t} dV
$$

(3.34)

The first term on the RHS of Eq. (3.34) is the referential inertial term, where the second term is the convective inertia term. Once we linearize both terms we then can incorporate them in the Eulerian virtual work Eq. (3.32).

### 3.2.2 Decomposition of Velocity and Acceleration

The material velocity and acceleration at time $t + \Delta t$ can be related to the previous value at time $t$ by simply using the relations:

$$
t^{+\Delta t} a_i = t a_i + a_i
$$

(3.35)

$$
t^{+\Delta t} v_i = t v_i + v_i
$$

(3.36)

where the incremental acceleration and velocity $a_i$ and $v_i$ depend on the time integration scheme that is used in the analysis. Note that higher orders of these quantities will be neglected to uphold a linear increment in their values.
3.2.3 Linearization of the Referential Inertia Term

Following the same procedure used in quasi-static analysis, using the incremental decompositions and linearization gives us the following:

\[
\int_{i+\Delta t} \rho^{i+\Delta t} a_i \delta u_i^{i+\Delta t} dV = \int_{i} \rho^{i} a_i \delta u_i^{i} dV
\]

\[
- \int_{i} \rho \frac{\partial^l v_k}{\partial^l x_k} t^{i+\Delta t} a_i \delta u_i \Delta t^{i} dV - \int_{i} \frac{\partial^l \rho^i}{\partial^l x_k} v_k^{i+\Delta t} a_i \delta u_i \Delta t^{i} dV
\]

The last term on the RHS of Eq. (3.37) can be written as:

\[
\int_{i} \frac{\partial^l \rho^l v_k}{\partial^l x_k} t^{i+\Delta t} a_i \delta u_i \Delta t^{i} dV = \int_{i} \frac{\partial^l \rho^l v_k}{\partial^l x_k} t^{i+\Delta t} a_i \delta u_i \Delta t^{i} dV
\]

Applying Gauss divergence theorem to the first integral in RHS of Eq. (3.38) and using Eq. (3.24) we get:

\[
\int_{i} \frac{\partial^l \left[\rho^l \left(v_k - v^g_k\right)^{i+\Delta t} a_i \delta u_i\right]}{\partial^l x_k} \Delta t^{i} dV
\]

\[
= \int_{i} \rho^l \left(v_k - v^g_k\right)^{i+\Delta t} a_i \delta u_i n_k \Delta t^{i} dS = 0
\]

Substituting Eq’s (3.38) and (3.39) in Eq. (3.37), we get:

\[
\int_{i+\Delta t} \rho^{i+\Delta t} a_i \delta u_i^{i+\Delta t} dV = \int_{i} \rho^{i} a_i \delta u_i^{i} dV
\]

\[
+ \int_{i} \rho^l v_k \frac{\partial \left(t^{i+\Delta t} a_i \delta u_i\right)}{\partial^l x_k} \Delta t^{i} dV
\]

Using Eq. (3.35), we can write the referential inertia term as follows:
\[
\int_{t^i + \Delta t}^{t^i + 2\Delta t} \rho^{t^i + \Delta t} a_i \delta u_i^{t^i + \Delta t} \, dV = \int_{V}^{t^i} \rho^{t^i} a_i \delta u_i^{t^i} \, dV + \int_{V}^{t^i} \rho a_i \delta u_i^{t^i} \, dV \\
+ \int_{V}^{t^i + \Delta t} \frac{\partial}{\partial x_k} \left( t^i a_i \delta u_i \right) \Delta t \, dV
\]  
(3.41)

### 3.2.4 Linearization of the Convective Inertia Term

The convective term (the second term on the RHS) in Eq. (3.34) can be expanded into the following:

\[
\int_{t^i + \Delta t}^{t^i + \Delta t} \rho^{t^i + \Delta t} v_j \frac{\partial^{t^i + \Delta t} \delta u_i}{\partial x_j} \, dV = \int_{V}^{t^i} \rho^{t^i} v_j \frac{\partial^{t^i} \delta u_i}{\partial x_j} \, dV \\
- \int_{V}^{t^i} \rho^{t^i} \frac{\partial^{t^i} v_j}{\partial x_j} \frac{\partial^{t^i + \Delta t} \delta u_i}{\partial x_j} \, dV \\
- \int_{V}^{t^i} \frac{\partial^{t^i} \rho}{\partial t^i} v_k \frac{\partial^{t^i} v_j}{\partial x_j} \frac{\partial^{t^i + \Delta t} \delta u_i}{\partial x_j} \, dV
\]  
(3.42)

We-writing the third term on the RHS of Eq. (3.42):

\[
\int_{V}^{t^i + \Delta t} \frac{\partial^{t^i} \rho}{\partial t^i} v_k \frac{\partial^{t^i + \Delta t} \delta u_i}{\partial x_j} \, dV \\
= \int_{V}^{t^i} \frac{\partial^{t^i} \rho^{t^i} v_k}{\partial t^i} \frac{\partial^{t^i + \Delta t} \delta u_i}{\partial x_j} \, dV \\
- \int_{V}^{t^i} \frac{\partial^{t^i} \rho^{t^i} v_k}{\partial t^i} \frac{\partial^{t^i + \Delta t} \delta u_i}{\partial x_j} \, dV \\
- \int_{V}^{t^i + \Delta t} \frac{\partial^{t^i + \Delta t} v_k}{\partial x_j} \frac{\partial^{t^i + \Delta t} \delta u_i}{\partial x_j} \, dV \\
- \int_{V}^{t^i} \frac{\partial^{t^i} \rho^{t^i}}{\partial t^i} v_k \delta u_i \Delta t \, dV \\
- \int_{V}^{t^i} \frac{\partial^{t^i} \rho^{t^i}}{\partial t^i} v_k \delta u_i \Delta t \, dV
\]  
(3.43)
Applying Gauss divergence theorem and the boundary constrain to the first integral in RHS of Eq. (3.43) we get:

\[
\int_{V} \frac{\partial}{\partial t} \left[ t \rho^{t} \left( v_{k} - v_{k}^{g} \right) t^{+\Delta t} v_{j} \frac{\partial^{t+\Delta t} v_{i}}{\partial t x_{j}} \right] \Delta t \, dV
\]

(3.44)

\[
= \int_{S}^{t} \rho^{t} \left( v_{k} - v_{k}^{g} \right) t^{+\Delta t} v_{j} \frac{\partial^{t+\Delta t} v_{i}}{\partial t x_{j}} \delta u_{i} n_{k} \Delta t \, dS = 0
\]

Substituting Eq’s. (3.43) and (3.44) into Eq. (3.42) we get:

\[
\int_{V}^{t+\Delta t} \rho^{t+\Delta t} v_{j} \frac{\partial^{t+\Delta t} v_{i}}{\partial t x_{j}} \delta u_{i} t^{+\Delta t} \, dV = \int_{V}^{t} \rho^{t+\Delta t} v_{j} \frac{\partial^{t+\Delta t} v_{i}}{\partial t x_{j}} \delta u_{i} t^{+\Delta t} \, dV
\]

(3.45)

\[
+ \int_{V}^{t} \rho^{t} v_{k} \frac{\partial^{t} v_{j}}{\partial t x_{k}} \frac{\partial^{t+\Delta t} v_{i}}{\partial t x_{j}} \delta u_{i} \Delta t \, dV
\]

Using Eq. (3.36), we write the convective inertia term as follows:

\[
\int_{V}^{t+\Delta t} \rho^{t+\Delta t} v_{j} \frac{\partial^{t+\Delta t} v_{i}}{\partial t x_{j}} \delta u_{i} t^{+\Delta t} \, dV = \int_{V}^{t} \rho^{t} v_{j} \frac{\partial^{t} v_{i}}{\partial t x_{j}} \delta u_{i} t^{+\Delta t} \, dV
\]

(3.46)

\[
+ \int_{V}^{t} \rho^{t} v_{j} \frac{\partial^{t} v_{i}}{\partial t x_{j}} \delta u_{i} t^{+\Delta t} \, dV + \int_{V}^{t} \rho^{t} v_{j} \frac{\partial^{t} v_{i}}{\partial t x_{j}} \delta u_{i} t^{+\Delta t} \, dV
\]

\[
+ \int_{V}^{t} \rho^{t} v_{k} v_{j} \frac{\partial^{t} v_{i}}{\partial t x_{k}} \frac{\partial^{t} v_{i}}{\partial t x_{j}} \Delta t \, dV
\]

\[
+ \int_{V}^{t} \rho^{t} v_{k} v_{j} \frac{\partial^{t} v_{i}}{\partial t x_{k}} \frac{\partial^{t} v_{i}}{\partial t x_{j}} \Delta t \, dV
\]
3.2.5 Eulerian Equilibrium Equation for Dynamic Analysis

Incorporating Eq’s. (3.41) and (3.46) into Eq. (3.32) we get the Eulerian equation of motion that includes the inertia effects:

\[
\begin{align*}
\int_{V} t \rho \alpha_i \delta u_i \prime \prime dV + \int_{V} t C_{ijkl} e_{kl} \delta e_{ij} \prime \prime dV & + \int_{V} t \sigma_{ij} \delta \gamma_{ij} \eta \prime dV \\
+ \int_{V} t \rho \prime_{ij} v_j \frac{\partial v_i}{\partial t x_j} \delta u_i \prime dV & + \int_{V} t \rho \prime v_j \frac{\partial v^\prime_i}{\partial t x_j} \delta u_i \prime dV \\
+ \int_{V} t \rho \prime_{ij} v_k v_j \frac{\partial}{\partial t x_k} \left( \frac{\partial v_{ij}^\prime}{\partial t x_j} \delta u_i \prime \right) \Delta t dV & \\
+ \int_{V} u_k t \sigma_{ij} \frac{\partial \delta e_{ij} \prime}{\partial t x_k} dV & + \int_{V} \frac{\partial u_k}{\partial t x_j} t \sigma_{ij} \frac{\partial \delta u_i \prime}{\partial t x_k} dV \\
= \delta^{t+\Delta t} W_{ext} & - \int_{V} t \sigma_{ij} \delta e_{ij} \prime dV - \int_{V} t \rho \prime a_i \delta u_i \prime dV \\
- \int_{V} t \rho \prime_{ij} v_k \frac{\partial}{\partial t x_k} \left( \frac{\partial a_i}{\partial t x_j} \delta u_i \right) \Delta t dV & - \int_{V} t \rho \prime v_j \frac{\partial v^\prime_i}{\partial t x_j} \delta u_i \prime dV \\
- \int_{V} t \rho \prime v_k v_j \frac{\partial}{\partial t x_k} \frac{\partial v_{ij}^\prime}{\partial t x_j} \delta u_i \Delta t dV & \\
- \int_{V} t \rho \prime v_k v_j \frac{\partial}{\partial t x_k} \left( \frac{\partial v_{ij}^\prime}{\partial t x_j} \delta u_i \right) \Delta t dV
\end{align*}
\]

Equation (3.47) is referred to an Eulerian grid that is fixed in space and has zero velocity and acceleration. The equation is linear in the incremental displacements \( u_i \), incremental velocities \( v_i \) and incremental accelerations \( a_i \).
3.3 Discretization of the Eulerian FE Equations

3.3.1 Discretization of the Quasi-Static Eulerian Equation

The matrix form of the Eulerian quasi-static equilibrium equation (3.32) is written as follows:

$$(t^tK^E + t^tK^A)u^{(i)} = t^{+\Delta t}f^{\text{ext}} - t^{+\Delta t}f^{(i-1)}$$ (3.48)

where

$$t^tK^E = t^tK^{E1} + t^tK^{E2}$$ (3.49)

$$t^tK^A = t^tK^{A1} + t^tK^{A2}$$ (3.50)

$t^tK^{E1}$ and $t^tK^{E2}$ are the Eulerian material and geometrical stiffness matrices given by:

$$t^tK^{E1} = \int_V \left(t^tB^E1\right)^T t^tC^t t^tB^E1 dV$$ (3.51)

$$t^tK^{E2} = \int_V \left(t^tB^E2\right)^T t^tS^t t^tE^2 dV$$ (3.52)

$t^tK^{A1}$ and $t^tK^{A2}$ are the convective stiffness matrices given by:

$$t^tK^{A1} = \int_V \left(t^tB^A1\right)^T t^tS^t t^tH^t dV$$ (3.53)

$$t^tK^{A2} = \int_V \left(t^tB^A2\right)^T t^tS^t t^tE^2 dV$$ (3.54)

$t^{+\Delta t}f^{(i-1)}$ is the internal force vector given by:

$$t^{+\Delta t}f^{(i-1)} = \int_V \left(t^{+\Delta t}B^E1\right)^T t^{+\Delta t} t^{+\Delta t} \sigma^{(i-1)} dV$$ (3.55)
and $^\text{ext} \mathbf{f}$ is the vector of the externally applied loads. Matrix $^t \mathbf{C}$ is the elastic-plastic material constitutive matrix. Defining $h_k$ to be the element shape function corresponding to nodal point $k$ and $N$ is the number of element nodal points, matrices $\mathbf{H}$, $\mathbf{B}^E_1$, $\mathbf{B}^E_2$, $\mathbf{B}^A_1$, $\mathbf{B}^A_2$, $^t \mathbf{S}^E_2$ and $^t \mathbf{S}^A_1$ are given by:

$$\mathbf{H} = \begin{bmatrix} \cdots & h_k & 0 & \cdots \end{bmatrix}_{2 \times 2N}$$

$$^t \mathbf{B}^E_1 = \begin{bmatrix} \frac{\partial h_k}{\partial t x} & 0 \\ 0 & \frac{\partial h_k}{\partial t y} \\ \frac{\partial h_k}{\partial t y} & \frac{\partial h_k}{\partial t x} \\ \sum_{j=1}^{N} h_j^t \bar{x}_j \end{bmatrix}_{4 \times 2N}$$

$$^t \mathbf{B}^E_2 = \begin{bmatrix} \frac{\partial h_k}{\partial t x} & 0 \\ \frac{\partial h_k}{\partial t y} & 0 \\ 0 & \frac{\partial h_k}{\partial t x} \\ 0 & \frac{\partial h_k}{\partial t y} \\ \sum_{j=1}^{N} h_j^t \bar{x}_j \end{bmatrix}_{5 \times 2N}$$
\[ \begin{bmatrix}
\frac{\partial^2 h_x}{\partial' x^2} & 0 \\
\frac{\partial^2 h_x}{\partial' y^2} & 0 \\
\frac{\partial^2 h_x}{\partial' x \partial' y} & 0 \\
0 & \frac{\partial^2 h_x}{\partial' x^2} \\
0 & \frac{\partial^2 h_x}{\partial' y^2} \\
0 & \frac{\partial^2 h_x}{\partial' x \partial' y} \\
0 & \left( \frac{1}{t_x} \frac{\partial h_x}{\partial' x} - \frac{h_k}{t_x^2} \right) \\
0 & \frac{1}{t_x} \frac{\partial h_x}{\partial' y}
\end{bmatrix}^{8 \times 2N} \]

(3.59)

\[ \begin{bmatrix}
\frac{\partial h_k}{\partial' x} & 0 \\
0 & \frac{\partial h_k}{\partial' x} \\
\frac{\partial h_k}{\partial' y} & 0 \\
0 & \frac{\partial h_k}{\partial' y} \\
\frac{h_k}{\sum_{j=1}^{N} h_j t_j x_j} & 0
\end{bmatrix}^{5 \times 2N} \]

(3.60)
\[
{t}S_{E}^{2} = \begin{bmatrix}
{t}\sigma_{xx} & {t}\sigma_{xy} & 0 & 0 & 0 \\
{t}\sigma_{xy} & {t}\sigma_{yy} & 0 & 0 & 0 \\
0 & 0 & {t}\sigma_{xx} & {t}\sigma_{xy} & 0 \\
0 & 0 & {t}\sigma_{xy} & {t}\sigma_{yy} & 0 \\
0 & 0 & 0 & 0 & {t}\sigma_{zz}
\end{bmatrix}
\] 

(3.61)

\[
{t}S_{A1} = \begin{bmatrix}
{t}\sigma_{xx} & 0 & {t}\sigma_{xy} & 0 & {t}\sigma_{yy} & {t}\sigma_{zz} & 0 \\
0 & {t}\sigma_{xy} & {t}\sigma_{xx} & 0 & {t}\sigma_{yy} & 0 & {t}\sigma_{zz}
\end{bmatrix}^{T}
\] 

(3.62)

\[
{t+\Delta t}\sigma^{(i-1)}
\]

is the stress vector defined by:

\[
{t+\Delta t}\sigma^{(i-1)} = \begin{bmatrix}
{t+\Delta t}\sigma_{xx}^{(i-1)} & {t+\Delta t}\sigma_{yy}^{(i-1)} & {t+\Delta t}\sigma_{xy}^{(i-1)} & {t+\Delta t}\sigma_{xx}^{(i-1)} & {t+\Delta t}\sigma_{yy}^{(i-1)} & {t+\Delta t}\sigma_{zz}^{(i-1)}
\end{bmatrix}
\] 

(3.63)

### 3.3.2 Discretization of the Dynamic Eulerian Equation

The matrix form of the Eulerian dynamic equilibrium equation (3.47) is written as follows:

\[
{t}M_{E}a^{(i)} + \left( {t}C^{A1} + {t}C^{A2} \right)v^{(i)} + \left( {t}K^{E} + {t}K^{A} \right)u^{(i)} = {t+\Delta t}f^{\text{ext}} - {t+\Delta t}f^{(i-1)}
\]

\[
-\left( {t}M_{E} + {t}M^{A} \right){t+\Delta t}a^{(i-1)} - \left( {t}C^{A1} + {t}C^{A2} \right){t+\Delta t}v^{(i-1)}
\]

(3.64)

Matrix \( {t}M_{E} \) is the Eulerian mass matrix given by:

\[
{t}M_{E} = \int_{V} {t}\rho H^{T} H^{t} dV
\]

(3.65)
Matrix $t M^A$ is given by:

$$
t M^A = \begin{bmatrix}
  \cdots \\
  t M^A_{2i-1,2j-1} & t M^A_{2i-1,2j} \\
  t M^A_{2i,2j-1} & t M^A_{2i,2j} \\
  \cdots \\
\end{bmatrix}_{2N \times 2N} (3.66)
$$

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

$$
t M^A_{2i-1,2j-1} = t M^A_{2i,2j} = \int_V t \rho \left[ \left( h_i \frac{\partial h_j}{\partial t^l x} + h_j \frac{\partial h_i}{\partial t^l x} \right) \sum_{k=1}^N h_k \ t \bar{v}_{xk} + \left( h_i \frac{\partial h_j}{\partial t^l y} + h_j \frac{\partial h_i}{\partial t^l y} \right) \sum_{k=1}^N h_k \ t \bar{v}_{yk} \right] \Delta t^l dV (3.67)
$$

$$
t M^A_{2i-1,2j} = t M^A_{2i,2j-1} = 0 (3.68)
$$

Matrix $t C^A$ is given by:

$$
t C^A = \begin{bmatrix}
  \cdots \\
  t C^A_{2i-1,2j-1} & t C^A_{2i-1,2j} \\
  t C^A_{2i,2j-1} & t C^A_{2i,2j} \\
  \cdots \\
\end{bmatrix}_{2N \times 2N} (3.69)
$$

in which

$$
t C^A_{2i-1,2j-1} = t C^A_{2i,2j} = \int_V t \rho h_i \left[ \frac{\partial h_j}{\partial t^l x} \sum_{k=1}^N h_k \ t \bar{v}_{xk} + \frac{\partial h_j}{\partial t^l y} \sum_{k=1}^N h_k \ t \bar{v}_{yk} \right] t dV (3.70)
$$

$$
t C^A_{2i-1,2j} = t C^A_{2i,2j-1} = 0 (3.71)
$$
Matrix \( \mathbf{C}^{A_2} \) is given by:

\[
\mathbf{C}^{A_2} = \begin{bmatrix}
\vdots \\
' \mathbf{C}_{2i-1,2j-1}^{A_2} & ' \mathbf{C}_{2i,2j}^{A_2} \\
' \mathbf{C}_{2i,2j-1}^{A_2} & ' \mathbf{C}_{2i,2j}^{A_2}
\end{bmatrix}
\tag{3.72}
\]

in which

\[
' \mathbf{C}_{2i-1,2j-1}^{A_2} = \int_{V} \rho h_i h_j \sum_{k=1}^{N} \frac{\partial h_k}{\partial^t x} t' t_v x_k t' dV \\
' \mathbf{C}_{2i-1,2j}^{A_2} = \int_{V} \rho h_i h_j \sum_{k=1}^{N} \frac{\partial h_k}{\partial^t y} t' t_v x_k t' dV \\
' \mathbf{C}_{2i,2j-1}^{A_2} = \int_{V} \rho h_i h_j \sum_{k=1}^{N} \frac{\partial h_k}{\partial^t x} t' t_v y_k t' dV \\
' \mathbf{C}_{2i,2j}^{A_2} = \int_{V} \rho h_i h_j \sum_{k=1}^{N} \frac{\partial h_k}{\partial^t y} t' t_v y_k t' dV 
\tag{3.73}
\]

Matrix \( \mathbf{C}^{A_3} \) is given by:

\[
\mathbf{C}^{A_3} = \begin{bmatrix}
\vdots \\
' \mathbf{C}_{2i-1,2j-1}^{A_3} & ' \mathbf{C}_{2i,2j}^{A_3} \\
' \mathbf{C}_{2i,2j-1}^{A_3} & ' \mathbf{C}_{2i,2j}^{A_3}
\end{bmatrix}
\tag{3.74}
\]

in which

\[
' \mathbf{C}_{2i-1,2j-1}^{A_3} = ' \mathbf{C}_{2i,2j}^{A_3} \\
= \int_{V} \rho h_i \left[ \left( \frac{\partial h_j}{\partial^t x} \sum_{k=1}^{N} \frac{\partial h_k}{\partial^t x} t_v x_k + \frac{\partial h_j}{\partial^t y} \sum_{k=1}^{N} \frac{\partial h_k}{\partial^t x} t_v y_k \right) \sum_{k=1}^{N} h_k t_v x_k \\
+ \left( \frac{\partial h_j}{\partial^t y} \sum_{k=1}^{N} \frac{\partial h_k}{\partial^t y} t_v x_k + \frac{\partial h_j}{\partial^t x} \sum_{k=1}^{N} \frac{\partial h_k}{\partial^t y} t_v y_k \right) \sum_{k=1}^{N} h_k t_v y_k \right] \Delta t' dV 
\tag{3.75}
\]
\[ t^C_{A3}^{2i-1,j} = t^C_{A3}^{2i,2j-1} = 0 \]  

(3.76)

Matrix \( t^C_A \) is given by:

\[
\begin{bmatrix}
  \vdots \\
  t^C_{A4}^{2i-1,j-1} & t^C_{A4}^{2i-1,2j} \\
  t^C_{A4}^{2i,2j-1} & t^C_{A4}^{2i,2j} \\
  \vdots & \vdots \\
  t^C_{A4}^{2N-1,j-1} & t^C_{A4}^{2N-1,2j} \\
\end{bmatrix}_{2N \times 2N}
\]  

(3.77)

in which

\[
t^C_{A4}^{2i-1,j-1} = t^C_{A4}^{2i,2j} \\
= \int_V t^\rho \left[ \left( \frac{\partial h_i}{\partial t^j} \frac{\partial h_j}{\partial t^i} + h_i \frac{\partial^2 h_j}{\partial t^i \partial t^x} \right) \left( \sum_{k=1}^{N} \frac{\partial h_k}{\partial t^i} \right) \Delta t \right] \Delta t^i \text{dV} \\
= \sum_{k=1}^{N} h_k \Delta t^i \text{dV}
\]  

(3.78)

\[
t^C_{A4}^{2i-1,2j-1} = t^C_{A4}^{2i,2j-1} = 0 
\]  

(3.79)
CHAPTER 4

VOLUME OF SOLID (VOS) METHOD

Finite element Eulerian formulation in metal forming has mostly been used to simulate steady state problems [5-8]. Only limited trials in simulating unsteady state problems may be found [9-14]. In these trials, only the middle stage of the forming process is modeled and the expected shape of that stage is meshed with Eulerian grid. This pseudo analysis approach has been considered to deal with the problem of free surface detection and identification. The drawback is that it is limited to certain number of processes, and it does not give a full picture of the whole process. Using Eulerian formulation to simulate all stages of unsteady state metal forming process suffers from the detection of the free surface and the treatment of the convective terms in the analysis. In this section we discuss the development of the VOS theory and its application to problems in solid mechanics and metal forming. Also, we discuss the integration of the method in a FE context for a uniform grid.

First, we define a function \( f \) which represents the fractional volume of the solid in each cell that has a value between zero (at an empty cell) and one (at a cell fully occupied by solid). A value of \( f \) between zero and one signals a cell with a free surface (Figure 2.1). Hirt and Nichols [37] presented a transport equation for the evolution of the function \( f \) in a VOF method as:

\[
\frac{d}{dt} f + v_i \frac{\partial f}{\partial x_i} = 0
\]  (4.1)

The above equation states that \( f \) moves with the material and, in a Lagrangian formulation, it indicates that \( f \) remains constant in each cell. Also, it indicates that the material time derivative of \( f \) is zero as given by Eq. (3.3). Solving Eq. (4.1) gives the distribution of \( f \) and locates the elements in which free surface occurs. The next step would be to find the direction in which the free surface is located in each element. Different techniques were proposed in literature and they mostly present some
modification to the basic method of VOF-PLIC. Some work attempted solving equation (4.1) directly [31, 32] by considering $f$ as an unknown variable that is discretized on the computational grid. Other methods are based on tracking the change of $f$ through the fluid fluxes [37, 43]. One of the common methods used is the cubic-interpolated propagation method (CIP). This method is based on the fact that a variable and its spatial derivative propagates with the velocity and the spatial profile within each grid and is interpolated with a cubic polynomial [54]. Some methods use a semi-Lagrangian conservative scheme to solve the transport equation and to find the shape of the free surface. In addition to these techniques, mesh refinement and re-meshing techniques were also introduced to VOF solutions at the free surface elements [49-56].

The numerical solution of equation (4.1) tends to give some smearing problems; the reason being that the gradient of $f$ should be physically singular at the boundary but is finite in the numerical solution. So we need a method that can specify the shape and direction of the free surface in a consistent and continuous way along the element boundaries. Once the free surface is defined along the boundary, we need a scheme to update the current free surface to the new one in the Eulerian grid. The theory of the VOS method is presented through three major steps. First, equation (4.1) is solved in order to obtain the distribution of the fractional volume value $f$ on the domain. Next, a direction vector that is normal to the free surface interface and points toward the free boundary is calculated based on the distribution of $f$ that was obtained in the previous step. In the final step, the free surface interface is constructed by combining the free surface directional vector and the values of $f$ at each element.

The presentation in this chapter is divided into two parts. The first part presents the theory of the VOS method for a uniformly constructed grid, whereas the second part extends the method for a generalized non-uniform grid. The first step of the VOS method is almost the same between the two parts except for some minor differences. The calculation of the free surface directional vector in the second part follows the same approach of the uniform mesh but with a major modification in its equation in order to extend it for a generalized case of a non-uniform mesh. The construction of the surface is where the two parts differ. For the uniform mesh, we present a method based on the PLIC
scheme and a classification of all the possible interfaces. For the non-uniform mesh, we present a new scheme based on the filling pattern that can be used regardless of the shape of the element.

4.1 VOS for a Uniform Mesh

4.1.1 Solution of the Transport Equation

In order to find the location of the free surface we first need to calculate the fractional volume of each element. In what follows, this is calculated based on the volume change between each element and its neighboring elements. In solving Eq. (4.1), \( f \) is defined for the cell or the element, not the node. This means that in contrast to the discretized virtual work equation, another scheme is required to integrate the transport equation of \( f \). In the present analysis, the equation is first integrated over each element (cell) as follows:

\[
\int_V \left[ \frac{\partial f}{\partial t} + \nabla \cdot \mathbf{v} \right] dV = 0 \tag{4.2}
\]

Separating the integrals and applying the divergence theorem, Eq. (4.2) becomes:

\[
\left( \mathbf{f} \cdot \mathbf{v} \right)_i = -\int_A \left( \mathbf{f} \cdot \mathbf{v} \right) n_i dA \tag{4.3}
\]

Explicit time integration for Eq. (4.3) has been implemented in our analysis. Applying the central difference method, Eq. (4.3) becomes [44]:

\[
f_i^{\text{new}} = f_i^{\text{old}} + \frac{\Delta t}{V_i} \left[ -\sum_j \left( f_{\Gamma_j} \mathbf{v} \cdot \mathbf{n} \right) A_{\Gamma_j} \right] \tag{4.4}
\]

\[
f_{\Gamma_j} = \frac{A_{\text{solid}}^j}{A_j} \tag{4.5}
\]
where \( i \) is the element number and \( j \) is the element face number, \( f^\text{new}_i \) and \( f^{\text{old}}_i \) are the fractional volume values for the \( i^{\text{th}} \) element at the new and old time steps, \( V_i \) is the \( i^{\text{th}} \) element volume, \( A_j \) is the \( j^{\text{th}} \) face of the \( i^{\text{th}} \) element, \( \mathbf{v} \) and \( \mathbf{n} \) are the velocity and unit outward normal vectors at the \( j^{\text{th}} \) face of the \( i^{\text{th}} \) element. Figure 4.1 shows the definition of the parameters of Eq’s. (4.4) and (4.5). Since the elements here are considered as control volumes, the fractional volume calculation is based on the facial net change in each element, which is presented by the last term in Eq. (4.4). Also, \( f_{\Gamma_j} \), defined as the facial fractional value, indicates what part of the \( j^{\text{th}} \) face of the \( i^{\text{th}} \) element is in contact with the material and is defined by the fraction between the facial area \( A_{j}^{\text{solid}} \) in contact with the material and the total facial area \( A_j \) as shown in Figure 4.1.

![Figure 4.1 Definition of free surface element parameters](image)

4.1.2 Free Surface Directional Vector

One of the main problems in most methods used to find the location of the free surface is that they do not ensure a consistency of the free surface among all the free surface elements. This means that the free surface can be calculated accurately for each
element separately, but these free surfaces may not be connected. Solving Eq. (4.4) gives value for \( f \) in each element, but there are many cutting planes within an element that satisfies a given \( f \) value. Eliminating this problem is based on the idea that a normal vector together with the fractional volume value of each element determines a unique planar interface cutting the element (cell). Different methods have been implemented to calculate the free surface normal vector [44, 45, 51, 58, 66]. These methods differ with respect to the influence of the fractional volume distribution in the sub-domain around the element of interest. The most common sub-domains used in the literature are 3x3, 5x3, 3x5, 7x3 and 3x7 stencils. All of these, except of the 3x3 stencil, are mostly used with height function or line segment methods to calculate the vector normal to the free surface. In this study we use the 3x3 stencil but with a technique that captures the advantages of the height function and the regular 3x3 method of calculation.

To ensure optimum connectivity of the free surface, we calculate the location for all the free-surface elements simultaneously with the same level of accuracy. To do this, we first calculate the direction of a vector that is perpendicular to the free surface, and then construct the free surface for all the elements simultaneously. In most of the methods that use a 3x3 stencil, the directional vector is calculated by computing the horizontal and vertical fluxes of \( f \) between the element of interest and the neighboring elements in the horizontal and vertical directions, respectively. This is considered an acceptable approximation but it takes into account only two neighboring elements in each direction which may give rise to inaccurate indication of the directional vector. In methods that use the height function, a 3x7 or a 7x3 stencil is used, depending on the angle of the free surface, and the number of utilized neighboring elements increases. This bigger stencil gives better results for the interface curvature than the traditional 3x3 stencils but has the drawback of assuming that one component of the directional vector is a unit normal to the side with the 7 elements. Therefore, we need to develop a procedure that calculates the directional vector more accurately without such assumptions.

In this analysis we calculate the direction of the vector by taking into account the fractional volume values for each neighboring element (horizontally, vertically and diagonally). We present a new approach that computes the fluxes in a similar way to the
height function method but with some modifications. First we use a 3x3 stencil with the element of interest being at the center of the stencil. Second, we utilize the horizontal and vertical directions simultaneously without the assumption of the unit normal that is used in height functions. Therefore, we are measuring the fluxes in $f$ between neighboring rows/columns rather than neighboring cells. This gives a better picture for the distribution of the fractional volume values and their effect on calculating the directional vector while using a smaller stencil size than the height function in the same time. For our 3x3 stencil, we consider the element of interest is located in row $k$ and column $l$. The fractional volume difference between the three vertical elements on the $l+1$ column and the three on the $l-1$ column is used for the horizontal component of the direction vector, and vice versa. The free surface direction vector is calculated as follows:

$$
\mathbf{r}_i = \frac{\mathbf{R}_i}{|\mathbf{R}_i|}, \quad \mathbf{R}_i = -\left( \sum_{m=k-1}^{k+1} (f_{m,l+1} - f_{m,l-1}) \mathbf{n}_x + \sum_{n=l-1}^{l+1} (f_{k+1,n} - f_{k-1,n}) \mathbf{n}_y \right) \quad (4.6)
$$

where $\mathbf{r}$ is calculated for the $i^{th}$ element, $f$ is the fractional volume of the element in the $k^{th}$ row and $l^{th}$ column depending on the summation, and $\mathbf{n}_x$, $\mathbf{n}_y$ correspond to the facial normal vectors in the $x$- and $y$-direction, respectively. Note that Eq. (4.6) is carried out locally rather than globally, so that for each element we focus on the 3x3 stencil and perform the calculation within that stencil only. The process is then repeated for all the elements along the free boundary.

Because interior elements have a fractional volume that is larger than that of elements at the surface, making the free surface directional vector point to the inward direction, the negative sign is used to make the directional vector point to the outward direction of the interface. Equation (4.6) indicates that the directional vector is normalized and is based on the distribution of the fractional volume along the domain. Figure 4.2 shows the definition of the directional vector. This procedure is applied for a uniform grid with equal size square elements. The advantage is the ease of mesh generation and the ease of calculating $\mathbf{r}$ mathematically, since the normal vectors are set before we start the solution and defined as $i, j, -i$ or $-j$. 
In some situations, the free surface element might be located at one end of the grid domain. This means that the element is located next to a fictitious wall and might be in contact with a fixed rigid object, and in this case the 3x3 stencil will be missing three elements (either horizontal or vertical). To overcome this problem and to ensure having a 3x3 stencil all the time, we assume to have a fictitious element on the side adjacent to face \( j \) of the free surface element, where the grid domain ends, with two fictitious elements neighboring it in the other opposite directions. We assume that this fictitious element has a value of \( f \) that follows a linear variation with the adjacent elements. If the linear progression of \( f \) leads to an overshoot of the fractional volume value then an \( f \) value of one is used for that element and the amount of the overshoot is assigned to the fictitious element neighboring it in the interface progression direction. The other fictitious element is assigned an \( f \) value of one. If the linear progression of \( f \) leads to undershoot of \( f \) then a value of zero is used for both fictitious elements and the undershooting is applied to the remaining fictitious element that neighbors it in the direction if the interface progression. Considering the case where the free surface is adjacent to the right end of the grid domain, the construction of the fictitious elements is performed as follows:

\[
\begin{align*}
\text{if } f_{k,l} + (f_{k,l} - f_{k,l-1}) \geq 1 & \text{ then } \\
& 
\begin{cases}
    f_{k+1,l+1}^{\text{fic}} = f_{k,l} + (f_{k,l} - f_{k,l-1}) - 1 \\
    f_{k,l+1}^{\text{fic}} = 1 \\
    f_{k-1,l+1}^{\text{fic}} = 1
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\text{if } 0 < f_{k,l} + (f_{k,l} - f_{k,l-1}) < 1 & \text{ then } \\
& 
\begin{cases}
    f_{k+1,l+1}^{\text{fic}} = 0 \\
    f_{k,l+1}^{\text{fic}} = f_{k,l} + (f_{k,l} - f_{k,l-1}) \\
    f_{k-1,l+1}^{\text{fic}} = 1
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\text{if } f_{k,l} + (f_{k,l} - f_{k,l-1}) < 0 & \text{ then } \\
& 
\begin{cases}
    f_{k+1,l+1}^{\text{fic}} = 0 \\
    f_{k,l+1}^{\text{fic}} = 0 \\
    f_{k-1,l+1}^{\text{fic}} = 1 + f_{k,l} + (f_{k,l} - f_{k,l-1})
\end{cases}
\end{align*}
\]

Figure 4.3 shows the three different cases for the construction of the fictitious elements based on the linear progression of \( f \). The three cases correspond to the case
where a fictitious column is missing in right end of the domain. Other cases where fictitious rows/columns are needed at other sides of the domain follow the same rule used in Eq. (4.7) and shown in Figure 4.3.

Figure 4.2 Definition of the free surface directional vector

Figure 4.3 Construction of fictitious elements at the right end of the domain
4.1.3 Construction of the Free Surface using a Geometrical PLIC Scheme

Based on the value of $f$ and the free surface directional vector, we advance to locate the free surface. In the following analysis we divide the possible free surface configurations into four main categories, depending on the quadrant the free surface directional vector pointing to. Figure 4.4 shows the classification of the four categories based on the free surface directional vector $r$. In the following discussion we only cover the case with directional vector components, $r_x$ and $r_y$, having positive signs. The rest of the quadrants follow the same procedure. In any quadrant there are three interface shape cases. Two of those cases are unique and the third one is a general case that consists of several possibilities. These cases are identified by the angle $\theta = \tan^{-1}|r_y/r_x|$ as illustrated in Figure 4.4. The general case has $0^\circ < \theta < 90^\circ$ and the unique cases have $\theta$ equal to $0^\circ$ or $90^\circ$.

![Four quadrant categorization of the free surface](image)

Figure 4.4 Four quadrant categorization of the free surface
Based on the value of the angle \( \theta \) and the value of the fractional volume \( f \) we can define the location and shape of the free surface. Basically, we calculate the area of the element that is filled with the material and use the angle to specify which part of the element is filled. Consider the general case; there are six possible situations to fill the element as shown in Figure 4.5 (cases 1-3, 5-7). The lengths \( a_1 \) and \( a_2 \) are two unknown variables that locate where the free surface intercepts the element edges and are calculated for each case depending on \( \theta \) and \( f \).

Cases 1, 2 and 3 are considered when the free surface cuts through the element at an angle \( \theta \leq 45^\circ \). Cases 5, 6 and 7 are considered when the free surface cuts through the element at an angle \( 45^\circ < \theta < 90^\circ \). The difference between cases 1 & 3 and cases 5 & 7 is that in cases 1 & 3 \( a_1 > a_2 \), while cases 5 & 7 have the opposite. The difference between cases 2 and 6 is that in case 2 the free surface cuts through two element sides that are horizontally opposite to each other, while in case 6 the free surface cuts through the vertically opposite edges. Since we know which set of cases we have (1, 2, 3 or 5, 6, 7) from the identified angle, we need to specify which case among the set is applicable for each element. To do that, we define two critical fractional volumes \( f_{cr}^1 \) and \( f_{cr}^2 \) for the first set and two critical fractional volumes \( f_{cr}^3 \) and \( f_{cr}^4 \) for the second set, based on the area and geometry of the material within the cell:

\[
\begin{align*}
0^\circ < \theta \leq 45^\circ & \quad f_{cr}^1 = \frac{1}{2} \tan \theta, \quad f_{cr}^2 = 1 - \frac{1}{2} \tan \theta \\
45^\circ < \theta < 90^\circ & \quad f_{cr}^3 = \frac{1}{2} \cot \theta, \quad f_{cr}^4 = 1 - \frac{1}{2} \cot \theta 
\end{align*}
\] (4.8)

By comparing the actual fractional volume value \( f \) with the critical values defined in Eq. (4.8) we choose the specific case from Figure 4.5 as follows:

\[
\begin{align*}
\text{for } 0^\circ < \theta \leq 45^\circ & \quad \begin{cases} 
 f \leq f_{cr}^1 & \text{Case 1} \\
 f_{cr}^1 < f \leq f_{cr}^2 & \text{Case 2} \\
 f > f_{cr}^2 & \text{Case 3}
\end{cases}
\end{align*}
\] (4.9)
The last two special cases are straightforward, for \( \theta = 0^0 \) and for \( \theta = 90^0 \) (cases 4 & 8 in Figure 4.5). Now that we know the specific case for each element, we need to calculate the variables \( a_1 \) and \( a_2 \) to construct the free surface. This is done by calculating the area that is filled by the material and by relating the two variables through the angle \( \theta \).

If \( \Delta s \) is the element edge length, then by using \( f, \theta \) and \( \Delta s \) we can generate the free surface using a piecewise linear interface construction (PLIC). Considering cases 1 and 5 as an example, the \( a_1 \) and \( a_2 \) are calculated as follows:

\[
\begin{align*}
\frac{a_2}{a_1} &= \tan \theta \\
\frac{1}{2} a_1 a_2 &= (\Delta s)^2 f
\end{align*}
\]

\[
\Rightarrow \begin{cases} 
  a_1 = \Delta s \sqrt{2f \cot \theta} \\
  a_2 = a_1 \tan \theta
\end{cases}
\]

Similarly, the rest of the cases follow the same procedure. Table 4.1 summarizes the equations used in the calculation of \( a_1 \) and \( a_2 \) for all cases.

\[
\begin{align*}
\begin{cases} 
  f \leq f_{cr}^3 & \text{Case 5} \\
  f_{cr}^3 < f \leq f_{cr}^4 & \text{Case 6} \\
  f > f_{cr}^4 & \text{Case 7}
\end{cases}
\end{align*}
\]
<table>
<thead>
<tr>
<th>Case</th>
<th>Relation between parameters</th>
<th>Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\frac{a_2}{a_1} = \tan \theta$</td>
<td>$a_1 = \Delta s \sqrt{2 f \cot \theta}$, $a_2 = a_1 \tan \theta$</td>
</tr>
<tr>
<td></td>
<td>$\frac{1}{2}a_1 a_2 = (\Delta s)^2 f$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$\frac{a_2 - a_1}{\Delta s} = \tan \theta$</td>
<td>$a_1 = \frac{\Delta s}{2} (2 f - \tan \theta)$, $a_2 = a_1 + \Delta s \tan \theta$</td>
</tr>
<tr>
<td></td>
<td>$(a_1 + a_2) \frac{\Delta s}{2} = (\Delta s)^2 f$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$\frac{\Delta s - a_1}{\Delta s - a_2} = \tan \theta$</td>
<td>$a_1 = \Delta s \left(1 - \sqrt{2(1 - f) \tan \theta}\right)$, $a_2 = \Delta s \left(1 - \sqrt{2(1 - f) \cot \theta}\right)$</td>
</tr>
<tr>
<td></td>
<td>$\frac{1}{2}(\Delta s - a_1)(\Delta s - a_2) = (\Delta s)^2(1 - f)$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$a_1 = a_2$</td>
<td>$a_1 = \Delta s f$</td>
</tr>
<tr>
<td></td>
<td>$a_1 \Delta s = (\Delta s)^2 f$</td>
<td>$a_2 = a_1$</td>
</tr>
<tr>
<td>5</td>
<td>$\frac{a_2}{a_1} = \tan \theta$</td>
<td>$a_1 = \Delta s \sqrt{2 f \cot \theta}$, $a_2 = a_1 \tan \theta$</td>
</tr>
<tr>
<td></td>
<td>$\frac{1}{2} a_1 a_2 = (\Delta s)^2 f$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>$\frac{\Delta s}{a_2 - a_1} = \tan \theta$</td>
<td>$a_1 = \frac{\Delta s}{2} (2 f + \cot \theta)$, $a_2 = a_1 - \Delta s \cot \theta$</td>
</tr>
<tr>
<td></td>
<td>$(a_1 + a_2) \frac{\Delta s}{2} = (\Delta s)^2 f$</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>$\frac{\Delta s - a_1}{\Delta s - a_2} = \tan \theta$</td>
<td>$a_1 = \Delta s \left(1 - \sqrt{2(1 - f) \tan \theta}\right)$, $a_2 = \Delta s \left(1 - \sqrt{2(1 - f) \cot \theta}\right)$</td>
</tr>
<tr>
<td></td>
<td>$\frac{1}{2}(\Delta s - a_1)(\Delta s - a_2) = (\Delta s)^2(1 - f)$</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>$a_2 = a_1$</td>
<td>$a_2 = \Delta s f$</td>
</tr>
<tr>
<td></td>
<td>$a_2 \Delta s = (\Delta s)^2 f$</td>
<td>$a_1 = a_2$</td>
</tr>
</tbody>
</table>
4.2 VOS for a Non-Uniform Mesh

Most of the work done on the VOF method is based on a uniform Eulerian grid with rectangular elements. This is convenient due to the fact that the VOF method depends on calculating the amount of fluid moving through cell boundaries (fluxes) based on the fractional volume and the velocity field in each cell. Also, in most problems there is no need for a non-uniform mesh since the grid is fixed and the material is allowed to move freely within the domain. Nevertheless, some applications and problems need to use a non-uniform grid. Some researchers handled this problem through modified VOF techniques. These techniques are based on a mixed-two-step solution: a Lagrangian step followed by an Eulerian step. Although this method is efficient in most cases, it poses some challenges in problems where the mesh is too small in some regions. It also introduces more intense calculations since the solution consists of two steps rather than one. In addition to the difficulty in mapping parts of two or more Lagrangian elements into one Eulerian grid when needed [54-56].

In the following sections, we present a new technique to deal with non-uniform mesh and to calculate the fractional volume of each cell without the need for a Lagrangian step. The solution of the transport equation and the calculation of the free surface directional vector are similar to the ones presented for the uniform mesh whereas the construction of the free surface is where the two formulations differ.

4.2.1 Modification to the Solution of the Transport Equation

Similar to the development for the uniform mesh, we calculate the fractional volume of each cell using Eq. (4.4) by generalizing it for quadrilateral element. We modify the calculations of the element volume and the outward unit normal vector to each edge by defining a reference node ($O$) in each element and choose this node to always be at the bottom left corner. Referring to Figure 4.6, a typical cell would be $OBCD$. The cell volume is simply calculated by summing up volumes of two triangles ($OBC$ and $CDO$). Denoting a typical position vector between two grid points, e.g., $OB$, by $\mathbf{R}$ and its corresponding outward normal by $\mathbf{n}$, we may write the following relation:
\[ n = \pm \frac{R_y}{|R|} i \pm \frac{R_x}{|R|} j \]

where

\[ \begin{align*}
&+,- \text{ for } R_x \geq 0 \text{ and } R_y \geq 0 \\
&+,+ \text{ for } R_x \leq 0 \text{ and } R_y \geq 0 \\
&-,+ \text{ for } R_x \leq 0 \text{ and } R_y \leq 0 \\
&-,- \text{ for } R_x \geq 0 \text{ and } R_y \leq 0
\end{align*} \]

Figure 4.6 3x3 non-uniform stencil with outward facial normals

4.2.2 Generalization of Free Surface Directional Vector

We use the same general definition for the free surface direction vector and modify Eq. (4.6) to handle quadrilateral elements. In this case, each cell face will have both horizontal and vertical flux components and, therefore, we use the unit normal vectors for all the elements of the 3x3 stencil. The normal vectors will be calculated as inward vectors (Figure 4.7) for the element where the fractional volume \( f \) is considered. This will eliminate the problem of having to identify the sign for each component of the vectors.
when we calculate the difference in the fractional volume values, and we simply add the values of the fractional volumes multiplied by its inward normal vector. To get a rational picture for the distribution of the fractional volumes, we multiply each fractional volume by its element size. The modified version of Eq. (4.6) is written as:

\[
\mathbf{r}_i = \frac{\mathbf{R}_i}{|\mathbf{R}_i|},
\]

\[
\mathbf{R}_i = -\sum_{k=1}^{3} \left[ (f_{k,3} \cdot V_{k,3}) \mathbf{n}_{k,3} - (f_{k,1} \cdot V_{k,1}) \mathbf{n}_{k,1} \right]
- \sum_{l=1}^{3} \left[ (f_{3,l} \cdot V_{3,l}) \mathbf{n}_{3,l} - (f_{1,l} \cdot V_{1,l}) \mathbf{n}_{1,l} \right]
\]

where \( \mathbf{r} \) is calculated for the \( i^{th} \) element and \( f, V \) and \( \mathbf{n} \) are the fractional volume value, the volume and the unit normal vectors of the element in the \( k^{th} \) row and \( l^{th} \) column in a 3x3 stencil, respectively. The main difference between Eq. (4.6) and Eq. (4.13) is that the numerator in Eq. (4.13) is no longer separated into horizontal and vertical components prior to the calculation, but is separated into rows and columns with cells having both horizontal and vertical components. Note that corner elements have two unit normal vectors, so for the calculation of the fluxes between neighboring columns we use the unit vectors normal to the left face of the elements in the right column and vice versa. We do the same thing for the flux calculation between neighboring rows. Figure 4.7 shows the element numbering, in circles, for a 3x3 stencil.

In cases where one or more of the free surface elements are located at the end of the grid domain, we create a fictitious element beyond the domain with a reasonable \( f \) value that follows a linear progression similar to the scheme that was presented for the uniform mesh in Section 4.1.2. The constructed fictitious elements are considered to be uniform and Eq. (4.7) is used for the calculation of the fractional volume value of the fictitious elements.
4.2.3 Construction of the Free Surface using a Filling Pattern Scheme

It is not a straightforward task to find a general set of equations that may locate the free surface based on the value of the fractional volume and the free surface directional vector for a non-uniform mesh. Therefore, classifying the free surface interface shape possibilities, as in the case for the uniform mesh, is not practical and a different approach is needed for the free surface construction.

In this work, we present a filling technique that would incrementally fill the element using small strips through a specified pattern until it reaches the desired value of the fractional volume \( f \) that was calculated using Eq. (4.4). First, we choose a reference node for the start of the filling process. The reference node is chosen based on the signs of the free surface directional vector components defined in Eq. (4.13). Taking for example a typical element as in Figure 4.6 and Figure 4.8, we choose node \( O \) as the reference node if the both horizontal and vertical components of the directional vector \( (r_x \text{ and } r_y) \) are
positive (+ve). Similarly, we choose node B for -ve $r_x$ and +ve $r_y$; node C for -ve $r_x$ and -ve $r_y$, and node D for +ve $r_x$ and -ve $r_y$.

Each free surface element is then divided into three sections that are defined by drawing perpendicular lines to the free surface directional vector from the two nodes neighboring the reference node (Figure 4.8). We start filling each section with a small area of thickness $ds$ and width $w$. The thickness is defined by the user prior to the solution while the width is calculated at each time step based on the parameters defined in Figure 4.8. The total thickness of all the filled areas is defined as $S$.

Figure 4.8  Free surface sectioning in the filling pattern scheme
All the angles shown in Figure 4.8 are calculated using the dot product between the free surface directional vector and the vector connecting the reference node with the point where the strip is added:

\[
\alpha = \cos^{-1}\left(\frac{\mathbf{r} \cdot \mathbf{R}_{OB}^*}{|\mathbf{r}| |\mathbf{R}_{OB}|}\right) \quad \beta = \cos^{-1}\left(\frac{\mathbf{r} \cdot \mathbf{R}_{OD}^*}{|\mathbf{r}| |\mathbf{R}_{OD}|}\right)
\]

\[
\gamma = \cos^{-1}\left(\frac{\mathbf{r} \cdot \mathbf{R}_{OB^*}}{|\mathbf{r}| |\mathbf{R}_{OB}|}\right) \quad \delta = \cos^{-1}\left(\frac{\mathbf{r} \cdot \mathbf{R}_{OD^*}}{|\mathbf{r}| |\mathbf{R}_{OD}|}\right)
\]

(4.14)

Once the angles are known, the width of the small area may be calculated as follows:

\[
w = \begin{cases} 
S \left( \tan \alpha + \tan \beta \right) & \text{Section 1} \\
S \left( \tan \gamma + \tan \beta \right) & \text{Section 2} \\
S \left( \tan \gamma + \tan \delta \right) & \text{Section 3}
\end{cases}
\]

(4.15)

For each section, we multiply the width \(w\) by the thickness \(ds\) and divide by the cell volume to compute the fractional volume value of the added strip. Then, we compare the fractional volume of the filled area with the fractional volume value \(f\). If the filled area is less than \(f\) then we keep filling the element until we reach the desired value of \(f\). The total thickness of the filled area \(S\) is updated after each filling step by adding \(ds\) to the previous value. A bisectioning routine is employed if the filled area exceeds the desired value beyond the tolerance limit. In the routine, the thickness \(ds\) of the last filling step is reduced into a smaller thickness and the last filling step is re-initialized using the new \(ds\) value.
CHAPTER 5

SOLUTION CONTROL AND IMPLEMENTATION

5.1 Time Step Size and Stability

The choice of solution time step size is crucial for stability and convergence when using the VOS method. Based on the analysis presented in this work, the time increment must not allow the material to flow through more than one element in any direction. This condition is imposed for both cases of positive or negative net change for each element to ensure that an element will not be filled/emptied more that its volume. This is a very important stability concern since the VOS equations presented in this work are based on the assumption that fluxes occur only between adjacent elements.

We calculate the critical time step size for all the elements at each time step, and compare it with the minimum critical value given by the following expression:

\[
\Delta t_{cr} = \min \left\{ \frac{(1-f)V_i}{-\sum_j (f_{T_j} \cdot \mathbf{n}) A_j}, \frac{(0-f)V_i}{-\sum_j (f_{T_j} \cdot \mathbf{n}) A_j} \right\}
\]

Equation (5.1) has two different time steps, one of them computes the critical time step for a cell with a positive net change in the material volume and the other one is for a cell with a negative net change in the material volume. Note that the denominator in Eq. (5.1) represents the material net change in an element. This ensures that no element is filled with material more than its volume, and no filled (or partially filled) element is emptied more than its volume of material in one time increment.
5.2 Updating the Velocity Field

As the material flows within the fixed Eulerian grid, it is bound to occupy previously empty elements. These elements were not part of the computational domain in the previous time step but now must be included due to the recent presence of the material. The newly filled elements have no velocity values at their nodes and they need to be assigned a reasonable value that matches the velocity field distribution around them in order to be able to continue the simulation and advance the free surface. On the other hand, elements that are being emptied will be assigned a zero velocity at their nodes.

In this work, we use a volume weighted averaging process [74] for the velocity calculation at the new nodes by averaging the velocity field of the neighboring elements:

\[
v_{\text{new node}} = \frac{\sum \bar{v}_i V_i}{\sum V_i}
\]

(5.2)

where \( \bar{v}_i \) is the average velocity of the neighboring elements calculated by averaging the velocity of all nodes in that element and \( V_i \) is the volume of the neighboring elements. Figure 5.1 shows the newly added nodes at a free surface interface between times \( t \) and \( t+\Delta t \).

![Velocity field update for newly added nodes](image)

*Figure 5.1  Velocity field update for newly added nodes*
5.3 Tracking Material Point Properties

One of the main concerns in Eulerian and ALE formulations is how to deal with history dependant properties such as the stress and the strain. Because these physical parameters represent the history of the material during the course of the deformation, finding a proper formulation for them is essential. To do this, we apply Eq. (3.3) relating the grid and material time derivatives as follows; written for the equivalent strain $\bar{\varepsilon}$ as a typical quantity:

$$t' \hat{\varepsilon} = t' \varepsilon + t' v_i \frac{\partial^t \bar{\varepsilon}}{\partial^t x_i}$$  \hspace{1cm} (5.3)

The solution of Eq. (5.3) can be obtained globally by applying the Petrov-Galerkin approach and using the element shape functions of the Eulerian grid:

$$\int_V t' v_i \frac{\partial^t \bar{\varepsilon}}{\partial^t x_i} h_i dV = \int_V \left(t' \hat{\varepsilon} - t' \varepsilon\right) h_i dV = 0$$  \hspace{1cm} (5.4)

where $v_i$ is the material velocity, $h_i$ are the shape functions and $V$ is the volume of the elements. The equivalent strain rate $\hat{\varepsilon}$ can be obtained from the velocity calculations.

The solution of Eq. (5.3) could be achieved in a simpler manner at the free surface of the material if the trajectories are known. In Sections 4.1.2 and 4.2.2, the free surface location was calculated by finding the free surface directional vector which is a representation of the trajectory of the material points within the free surface element. Therefore, if the value of the equivalent strain is known at a certain material point $M_1$, then it can be calculated progressively along the directional vector following the same material point on the new updated free surface as shown in Figure 5.2. This is achieved using the following equation [75]:

$$t' \hat{\varepsilon} \left( x_i \right) \Delta t \equiv t' \varepsilon \left( x_i + \Delta x_i \right) - t' \varepsilon \left( x_i \right) \equiv \text{grad} \left( t' \bar{\varepsilon} \right) \cdot \Delta x_i$$  \hspace{1cm} (5.5)
5.4 Free Surface Connectivity

Even though the VOS method might ensure the connectivity of the free surface interfaces since they are calculated based on the fractional volume values of neighboring elements, it is not always guaranteed to have the surface connected between all elements at the free boundary. The objective here is to ensure connectivity of the free surface since such condition is not guaranteed. Therefore, we perform a connectivity step after each free surface capturing step is completed.

To guarantee the free surface to be connected at all times, we implement a default averaging scheme that connects the surface of two neighboring elements that share the same face. This is not the case for elements sharing a face with fictitious elements because their free surface remains the same from that side neighboring the fictitious element and no connectivity is required at that point. Once a connectivity step is performed, all subsequent surface interfaces must account for the change in neighboring free surface elements.

Based on the definition of facial fractional value in Eq. (4.5) and Figure 5.3, the default averaging process is performed as follows:
\[
\begin{align*}
    f_{\Gamma_j} &= \begin{cases} 
    (f_{\Gamma_j}^+ + f_{\Gamma_j}^-)/2 & \text{if } 0 \leq f_{\Gamma_j}^+ < 1 \text{ and } 0 \leq f_{\Gamma_j}^- < 1 \\
    1 & \text{if } f_{\Gamma_j}^+ = 1 \text{ or } f_{\Gamma_j}^- = 1 \\
    f_{\Gamma_j}^+ & \text{if } f_{\Gamma_j}^- \text{ is fictitious} \\
    f_{\Gamma_j}^- & \text{if } f_{\Gamma_j}^+ \text{ is fictitious}
    \end{cases} 
\end{align*}
\] (5.6)

where the + and - signs correspond to the opposite sides of the facial fractional volumes that share the same element side. If one facial value is equal to 1.0, we assume both values to be 1.0 so that a full element will not turn into a partially occupied one, and if one of the sides is a fictitious element we keep the value the same as it is for the true element.

The process starts from one element and is repeated for all other surface elements to average all the values and their after effect. The reason is because after connecting the free surface of two elements, the free surface of elements neighboring those two must account for the effect of the connectivity of those two elements to insure that the total volume of the solid is not changed during the simulation, i.e. to ensure that we do not lose any mass during the approximation (Figure 5.3).

![Figure 5.3 Default free surface connectivity scheme](image-url)
Originally, in fluid mechanics, the VOF method has been used in applications where the shape of the fluid free boundary is not exactly known in many cases and the VOF method was then used to predict the shape of the free boundaries. On the contrary to CFD applications, the expected final deformed shape of the workpiece in metal forming applications is known or at least partially known. Taking advantage of the pre-known deformed shape of the material in some regions can improve the quality and efficiency of the results obtained using the VOS method.

The first important modification we do to the default connectivity process is employed in the contact between the workpiece and rigid punch/dies. The connectivity process is modified so that it would not compromise free surface elements that are in contact with the punch/dies. The deformed shape of elements that are in contact with rigid objects will not be changed and will dominate the connectivity between them and neighboring elements that are outside of the contact region. Figure 5.4 shows the effect of having the contact-modified connectivity for a simple punch indentation problem compared to the default connectivity process. We know that the punch land area must remain flat due to the contact with the punch and not inclined, which is what happens with the default connectivity process as in Figure 5.4c. It is also clear that the default connectivity process will introduce some penetration between the punch and the material whereas the contact-modified process eliminates any penetration after effect (Figure 5.4d).

The second important modification to the default connectivity process is by introducing a weighting factor, with a value between zero and one, to the default averaging process in Eq. (5.6) to change the connectivity between two adjacent surfaces from a 50% averaging process to any other desired value. This can be used to modify the shape of curves and decide if a steep change in the curve is required towards the center or a gradual change is more suitable. Figure 5.5 shows the difference between two curves one with default connectivity and the other one with higher weighting factor value at both end of the curve. The weighting factor can also be used in other cases such as dealing with sharp corners of a workpiece during the deformation so that the corners will not lose their identity through the connectivity process.
Figure 5.4  Contact-modified free surface connectivity

Figure 5.5  Effect of weighting factor on the free surface connectivity process
5.5 Contact Analysis

One of the main features when simulating metal forming processes is modeling the contact between the workpiece and the tool. This is a very complex task due to the need for tracking the movement of both bodies, defining the state in which contact occurs between them, and applying constraints and boundary conditions that simulate the contact. A common assumption in metal forming processes is to model the contact set as a rigid-flexible pair, where the tool is modeled as a rigid body and the workpiece as a flexible body. In this analysis, we use the rigid-flexible contact assumption since flexible-flexible contact is beyond the scope of this research. Usually modeling the punch or the dies as a flexible entity is essential where the elastic deformation of the tool is of main interest. In this work, the focus is on the large deformations of the workpiece.

The two most common techniques used in the literature for the rigid-deformable contact are the Lagrange multiplier and the penalty methods. In the Lagrange multiplier method, a constrained minimization problem is solved where the constraint does not allow any penetration between the two contact bodies [28]. One of the problems with this method is that it increases the computational cost of the analysis because of the inclusion of the Lagrange multipliers into the system. Also, because special elements (gap elements) are used in this method, knowledge of where the contact will occur and a specification of the whole contact surface prior to the simulation are required [25]. The penalty is easier to employ but it allows some penetration between the contact bodies based on the penalty constant which also affects the accuracy and the stability of the numerical solution [28, 76].

In this work, a direct approach is used based on contact module presented for ALE analysis in [25, 28]. The contact module was used to track the contact state between a moving grid and a rigid tool, therefore it needs to be modified, so that it can track the movement of the material since the grid is fixed in space, and incorporated into the Eulerian formulation. This direct approach is originally based on a node-to-surface contact algorithm in which the tool is designed as geometrical segments, such as lines and arcs, which are connected together and dealt with as a whole surface. Since elements are fixed in an Eulerian grid, modifications are made to this contact module by adjusting
the way rigid segments are defined and changing the contact detection tracking scheme. The tool segments are now defined on the elemental level, which means that each tool segment is defined within an element by itself instead of defining all tool segments as one body. This allows the module to track the movement of the toll segments on the Eulerian grid. The contact is now detected at the free surface elements between rigid segments and free surface interfaces, and is now defined as a node-to-line which means the contact will be dealt with in each element separately rather than in the structure as a whole.

In each incremental time step, the algorithm first tracks the movement of the tool segments and records the coordinates of its final position. Then it tracks the movement of the boundary nodes, which are located at the intersection between the free surface interface and the element edges. The module then compares the coordinates of the tool and the free boundaries to detect if any penetration has occurred between them. Once penetration is detected, the algorithm calculates the depth and the direction of the penetration between the free surface interface and the rigid segments based on the free surface location that was calculated using the VOS method. If penetration occurs at more than one surface element, the algorithm determines the element at which penetration occurred first and starts with it first and then updates other contact elements accordingly before checking for penetration again. Once the penetration depth is calculated geometrically, a corrective velocity boundary condition, based on the value of the penetration depth, is enforced on the tool to reset the tool surface on the free surface interface. The corrective velocity should compensate for the free surface movement as the workpiece deforms during the process. Taking that into account, the corrective velocity constraint is calculated as follows (Figure 5.6):

\[ v^K = N^I v^I + N^J v^J + v^p \]  

(5.7)

where \( N^I \) and \( N^J \) are the shape functions corresponding to nodes \( I \) and \( J \), respectively, and \( v^p \) is the penetration vector that is calculated geometrically based on the depth of the contact penetration. Once the tool segments are repositioned, appropriate contact boundary conditions are applied to the boundary nodes that are in contact. The friction force of the contact is calculated based on Coulomb or shear friction laws, depending on
the problem, and applied to the contact nodes. The contributions of the contact forces are added to the Eulerian equilibrium equation:

\[ \int_{V} (t + \Delta t) \sigma_{ij} \delta t_{ij} e^{t + \Delta t} dV = \delta (t + \Delta t) W^{ext} + \int_{S^{C}} \delta u_{i}^{C} f_{i}^{C} (t + \Delta t) dS^{C} \quad (5.8) \]

where \( \delta u_{i}^{C} \) are the components of the virtual displacements on the contact surface defined at the boundary nodes and \( f_{i}^{C} \) are the components of the contact traction acting on the contact surface \( S^{C} \). For each boundary node, the contact algorithm will be performed for each surface element that includes that node separately. At the end, all the resulting forces and contact boundary conditions that were calculated for each element sharing that node are combined together to the shared node to simulate the effect of the contact. The algorithm also checks if the nodal force, normal to the contact surface, is compressive or tensile. If the force is compressive then the node is still considered to be in contact. However, if it is in tensile mode then the node is considered to be separated from the contact surface and these forces are set to zero. Figure 5.6 shows the penetration detection and the corrective repositioning steps.
5.6 Heat Transfer Analysis

The internal heat generation in metal cutting process plays an important role and cannot be neglected. Therefore we present in this section a heat transfer module that is used in the VOS simulation of metal cutting in Chapter 8. Most of the work dissipated due to the plastic deformation and friction is converted into heat that will raise the temperature of the workpiece and tool significantly. In this work we use an incremental staggered approach in which we solve a deformation problem followed by heat transfer problem in each increment. The deformation problem is performed as an isothermal stress analysis one with known temperature distribution at the beginning of each incremental step. The work dissipated in the stress analysis step is used as the source for the internal heat generation in the heat transfer analysis part at the end of that same incremental step. This approach is considered acceptable for small time steps and when the dependency of the material properties on the temperature is weak [25]. The general form of energy equation for fixed set of coordinates is written as:

$$\rho c T' + \rho cv \frac{\partial T}{\partial x} + (kT) = \dot{q}$$

(5.9)

in which \(v\) is the velocity, \(T\) is the temperature, \(\rho\) is the density, \(c\) is the specific heat, \(k\) is the conductivity coefficient and \(\dot{q}\) is the rate of heat generation per unit volume. The boundary conditions for this equation are:

$$q_i n_i = -q_s \left|_{S_1} + h \left( T_{m_1} - T_{m_2} \right) \right|_{S_2}$$

$$T \big|_{S_3} = T_{m_1}$$

(5.10)

which represents specified flux, convection and specified temperature on respective parts of boundary, \(S_1\) to \(S_3\). In Eq. (5.10), \(h\) represents the convection heat transfer or film coefficient, \(T\) is the temperature and \(n\) is the surface normal vector. The internal heat generation is defined by calculating the work done in the deformation step as follow:

$$\dot{q} = \sigma_{ij} \frac{\partial v_i}{\partial x_j} = \sigma_{ij} D_{ij} = \bar{\sigma} \bar{D}$$

(5.11)
where $\sigma$ and $\bar{D}$ represents the effective stress and effective strain rate, respectively. Practically, not all of the dissipated work is converted into heat and only the dissipated plastic work is counted for [77, 78]. Therefore we rewrite the internal heat generation as:

$$\dot{q} = \eta \bar{\sigma} \bar{D}$$

where $\bar{D}_p$ is the effective plastic strain rate and $\eta$ is a factor that represents the amount of the plastic work converted to heat which is found to range between 85-95%. Similarly, using the same definition for $\eta$, the heat generation due to friction between the workpiece and the tool can be written as:

$$q_s = \eta |\sigma_f v_f|$$

where $\sigma_f$ and $v_f$ are the frictional stress and the tangential velocity at the contact surface.

### 5.7 Solution Procedure

The VOS formulation presented in this work is implemented into a finite element code. The code is divided into three major steps. First we need to define the parameters of the process which includes all parts of the FE pre-processor stage. Next we set up the solution control parameters and apply connectivity modification where needed. After the problem setup is completed, the simulation is started by solving the Eulerian equilibrium equation and all related VOS equations. The solution procedure is summarized as follow (Figure 5.7):

1. **Problem Definition:**
   - Specify material properties
   - Generate mesh domain and specify initial distribution of $f$
   - Define tool geometry
   - Apply boundary conditions for the workpiece and the tool
(2) Solution Setup:

- Create contact pairing between the workpiece and the tool(s)
- Define contact parameters
- Apply connectivity weighting factor if needed
- Specify solution time step

(3) Solution Routine:

- Solve Eulerian equilibrium equation
- Obtain displacement and velocity fields
- Solve the transport equation and obtain the new distribution of the fractional volume values $f$
- Calculate the free surface directional vector
- Construct the free surface interface using the geometrical PLIC scheme (uniform mesh) or the filling pattern scheme (non-uniform mesh)
- Apply default connectivity process with specified weighting factors
- Apply contact-modified connectivity
- Update the velocity field for the newly added nodes at the free surface
- Obtain material point related properties

(4) **Repeat** the solution routine (3) until final solution time is reached.
Figure 5.7  Flowchart of the VOS FE-code solution procedure
CHAPTER 6

METAL FORMING APPLICATIONS

6.1 Punch Indentation Problem

The punch indentation problem is simulated to show the advantages of using the VOS method compared to the traditional Lagrangian formulation used in commercial FE software, e.g., ANSYS [76]. The main issues involved in simulating such processes are the mesh distortion due to large deformations, the penetration between the punch and the workpiece during the process, and the convergence problems, especially in achieving large penetration ratios.

The workpiece is compressed between two frictionless rigid dies moving with constant velocity until 50% height reduction is achieved. Due to symmetry, only one quarter of the problem will be modeled using plane strain conditions. Figure 6.1 shows the setup of the problem. The material model of the workpiece is assumed to be a bilinear elastic-plastic with Young’s modulus of elasticity $E = 200$ GPa, Poisson’s ratio $\nu = 0.3$, initial yield stress $S_y = 250$ MPa and tangent modulus $E_T = 1000$ MPa.

First, the problem is solved with ANSYS using Lagrangian formulation with 140 equally sized elements. For the VOS simulation, the whole domain is meshed into 600 elements with the original workpiece filling initially 300 elements. A simple rigid-flexible contact algorithm is used for this problem which detects the contact at the free surface or free boundary of the material. Contact-modified connectivity is used to keep elements under the punch land flat during the process. Figure 6.2 shows the evolution of the workpiece deformed shape from the VOS simulation. We notice that there is no penetration between the punch and the workpiece especially at the punch corner since the contact area between the punch and the workpiece is kept flat during the entire process with the help of the contact-modified connectivity process.
Figure 6.3 and Figure 6.4 show the von-Mises strain and stress distributions in ANSYS and the VOS model, respectively. The distribution and values of the strain and stress in the VOS simulation are similar to those obtained with ANSYS, however, the advantage of the VOS method is that the distribution of the values is smoother and does not possess high jumps. Also, by comparing the final deformed shape between the two models we can see how the punch land is completely flat in the VOS simulation where on the other hand slight penetration occurs between the punch corner and the workpiece in the ANSYS model.

The main purpose behind the simulation of this application was to validate the VOS FE-code by comparing with well-established commercial FE softwares such as ANSYS using a metal forming process that can be simulated using Lagrangian formulations without the need for any special treatment or re-meshing techniques. The results showed strong agreement between the VOS code and ANSYS. The VOS method is used to deal with the free surface tracking only for the elements at the boundary so that we do not have to deal with the mesh motion of the workpiece. Also, all of the empty cells are discarded from the solution until they are filled by the material, unlike traditional VOF methods where empty cells are considered to be filled by air and are included in the computational domain. Finally, by comparing the setup of the problem and the steps required at the preprocessor stage we see that in the VOS simulation no special contact options are required to be specified by the user. Also, in VOS there are no contact-tuning parameters to handle, such as the allowable penetration or the normal penalty stiffness parameters, which may need some tweaking when using ANSYS to get the optimum results.
**Figure 6.1** Geometrical setup for the punch indentation problem

**Figure 6.2** Evolution of the workpiece deformed shape in the punch indentation problem using VOS
Figure 6.3 von-Mises plastic strain distribution for the punch indentation problem
Figure 6.4  von-Mises stress distribution for the punch indentation problem
6.2 Compression between Wedge-Shaped Dies

Plane strain compression of a pre-shaped workpiece between two wedge-shaped dies is simulated using the VOS method. The same process was investigated experimentally by Chitkara and Johnson [79] and numerically by Gadala and Wang using an ALE formulation [22]. The purpose of this example is to demonstrate the capability of the VOS method in dealing with metal forming applications and the importance of using a non-uniform grid to setup the FE model of the problem. Since the workpiece is pre-shaped at the beginning of the process, then using a uniform mesh is not possible and a hybrid grid of uniform and non-uniform elements is essential to model the problem.

The process consists of two rough wedge-shaped dies pressing on two opposite sides of a pre-shaped workpiece. The geometrical setup of the problem is shown in Figure 6.5 with \( H \) defining the mean distance between the centers of both contact lines with a value of 24.07 mm. Only half of the problem is modeled due to the symmetry. The workpiece is made of half-hard aluminum and assumed to have bilinear elastic-plastic material model with Young’s modulus of elasticity \( E = 68.9 \) GPa, Poisson’s ration \( \nu = 0.33 \), yield stress \( S_y = 276 \) MPa and a tangent modulus \( E_T = 25.07 \) MPa.

The process is simulated up to 45% reduction in the initial height using a prescribed displacement. The experiment in [79] was conducted on a 300-ton Denison hydraulic testing machine with a load less than 3.45 MPa. A 121.6 x 15 mm window is used for the Eulerian grid in the VOS simulation with 778 elements, 436 of these elements are initially filled by the workpiece in its original configuration. Since the dies are rough, sticking contact condition is applied between the die and the workpiece.

The deformed shape of the workpiece is shown in Figure 6.6 for 15%, 30% and 45% reduction in height. The deformed shapes of the workpiece are very similar to the ones presented using ALE by Gadala and Wang [22]. We can see that by using the VOS method we avoided the mesh distortion problem that occurs in this type of applications when using traditional Lagrangian formulations. The plastic strain distribution during the deformation stages is shown in Figure 6.7. Comparing the values presented in here with the values obtained using ALE [22] we find that the strain values are very close to the
ones predicted using ALE. From the distribution of the strain at different stages, we can see that the plastic zone is formed around the die corners where the strain values are the highest. The central part of the interface has lower strain values, which is contributed to the sticking contact condition. The plastic zone starts forming under the corners and develops towards the center of the workpiece. No plastic zone occurs outside the dies and the strain values are very small. We conclude that the strain distribution is in agreement with the predicted behavior of such processes and is also similar to the behavior predicted by the ALE formulation.

By defining $h$ to be the mean height of the workpiece during the deformation, we identify the compressive strain as $(\ln H/h)$. Figure 6.8 shows the load variation on the die with respect to the compressive strain obtained from the VOS, ALE and experiment results. The load-strain curve from the VOS simulation matches very well with the curve obtained using the ALE formulation. Both, VOS and ALE, curves are very close to the experimental curve for strain values less than 0.35-0.4. A noticeable difference can be seen between the experimental and the VOS/ALE curves for higher strain values and this is due to the fact that a reduction in friction occurs in the experiment as the process continues, were in the numerical analysis the contact friction stays the same during the whole process.

![Figure 6.5](image)

**Figure 6.5**  Setup for the compression between wedge-shaped dies problem
Figure 6.6  Evolution of the workpiece deformed shape in the compression between wedge-shaped dies problem using VOS
(a) 15% Height reduction

(b) 30% Height reduction

(c) 45% Height reduction

Figure 6.7  von-Mises strain distribution in the compression between wedge-shaped dies problem
Figure 6.8  Load-strain comparison curves for the compression between wedge-shaped dies problem
6.3 Sheet Metal Extrusion

Sheet metal extrusion (forward extrusion) is one of the metal forming processes where large strains are expected due to the change of the sheet thickness. When using traditional Lagrangian formulations, remeshing is required to overcome the problem of the mesh entanglement at the region of the thickness reduction. Updating the boundary conditions is also important because the outer boundary of the workpiece is in contact with the die during the whole process. In this example, we use the VOS method to demonstrate the ease of simulating such applications without remeshing and updating the boundary conditions since we are using an Eulerian grid in which all boundary conditions are pre-defined for each region of the problem for the whole process.

In this application, a plane strain sheet metal extrusions is simulated for a 25% reduction in the sheet thickness. The die is shaped in the form of 5\textsuperscript{th} order polynomial curve with zero curvature and slope at both ends. An aluminum billet with a thickness of 2\textit{a} is forced between two frictionless rigid dies by a rigid piston pressing against the rear face of the billet. The piston is given a prescribed displacement of 4\textit{a}. The length of the reduction die is 1.2\textit{a}. Figure 6.9 shows the geometry of the problem at the undeformed stage. The material model of the aluminum billet is assumed to be bilinear elastic-plastic with Young’s modulus of elasticity \( E = 6.8947 \times 10^4 \) MPa, Poisson’s ration \( \nu = 0.3 \), yield stress \( S_y = 393 \) MPa and a tangent modulus \( E_T = 1137 \) MPa. Due to the symmetry, only one half of the process is considered. The same problem was solved by Gadala and Wang [20], and Bayoumi and Gadala [68] using ALE formulations with the same material model, and by Maniatty \textit{et al.} [7] using Eulerian formulation and a different material model. The difference between the listed references and this work is that they simulated the problem as steady state with part of the billet already being extruded. In our case we start from the undeformed state of the billet to show the capability of the VOS method to handle to process from the beginning.

An 11.5\textit{a} x 1.5\textit{a} window is meshed into 1202 elements for the Eulerian domain with 352 elements being initially filled by the material of the undeformed shape of the billet. In this application we take advantage of using a combination between uniform and non-uniform elements to mesh the domain. By doing that we can improve the shape of the
deformed billet under the die since it is shaped as a 5th order polynomial. Two contact sets were defined in this problem, the first one is between the fixed rigid die and the billet and the other one is between the rigid piston and the billet. Both contact sets are defined with no additional contact treatment. The deformed shape of the billet is controlled by the fixed rigid die during the whole simulation with the help of the non-uniform mesh under the die that fits the shape of the billet once it reaches the steady state stage.

Figure 6.10 shows the FE model of the problem in addition to the deformed shape when the billet enters the region of the extrusion die and the final deformed shape. It is clear from Figure 6.10b that the irregularity in the mesh under the reduction die helps the billet material to flow according to the shape of the 5th order polynomial accurately without any compromisation. The distribution of the von-Mises strains after a displacement of 4a is shown in Figure 6.11. The distribution and the values of von-Mises strain are in a very good agreement with the values obtained using ALE in [20, 68]. The values of the longitudinal stress component at different lateral positions from the mid-plane of the initial configuration are shown in Figure 6.12 starting from 1.5a before the reduction die up to 2a after the exit of the die. Generally, the values and the fluctuation behavior between compression and tension through the three regions of the problem are in agreement with the same figure that was presented in [20, 68]. The differences between the results are due to fact that in this work we simulated the process from the initial undeformed configuration which contributed to the slight difference in values at the region before the reduction die. Maniatty et al. [7] used an Eulerian formulation method for simulating steady state forming processes. In this work we used the VOS method with Eulerian formulation to simulate the process from the beginning rather than simulating only the steady state part.

Maniatty et al. [7] showed in their work that the ratio between the length of the reduction die and the mean thickness determines the pattern of the residual stresses. For low ratios, they showed that residual stresses should be compressive towards the center of the billet, whereas on the surface they tend to be in tension mode. In this problem we have a ratio of approximately 1.37 which is considered low based on their work, so we expect the same kind of behavior for the longitudinal stresses in our case. For \(b = 0.75a\),
Figure 6.13 shows the longitudinal stress along the thickness ($y$) of the billet at a distance $4a$ from the die exit. We can see that the pattern is as expected. Stresses are in compression towards the center of the billet and in tension towards the surface.

The above example shows the advantage of using a non-uniform mesh in obtaining an accurate shape of a deformed metal sheet. Compared to other methods, there is no need to jump to the steady state stage and ignore the initiation of the process in order to simplify the problem. The process is simulated from the beginning which allowed us to get better results that gave a clearer picture of the process and its behavior.

Figure 6.9  Geometrical setup for the sheet metal extrusion problem
(a) Initial configuration

(b) After 0.8a stroke

(c) Final Configuration

Figure 6.10  Evolution of the billet deformed shape in the sheet metal extrusion problem using VOS

Figure 6.11  von-Mises strain distribution for the sheet metal extrusion problem
Figure 6.12  Longitudinal stresses at different lateral positions for the sheet metal extrusion problem

Figure 6.13  Longitudinal stresses along the thickness of the billet in the sheet metal extrusion problem
6.4 Backward Extrusion

In backward extrusions, mesh distortion is a critical problem and it is not possible to use traditional Lagrangian formulation due to excessive distortion and folding of the elements unless an extensive re-meshing algorithm is used. Using the coupled Eulerian and VOS formulation eliminates the problem of the mesh distortion and easily overcomes the problems of contact and free boundary tracking. The following example shows that such processes can be smoothly simulated using an Eulerian formulation with the VOS method and a uniform mesh. This example is based on the same setup presented by Hur et al. [80] and the results from the VOS simulation will be validated by comparing them to the results presented by [80].

The geometrical setup is shown in Figure 6.14. The only difference in this model is that a rigid die is used instead of an elastic die, and a bilinear elastic-plastic material model is used. The workpiece is made from AISI-1010 steel with a flow stress relation: \( \bar{\sigma} = 715.95\varepsilon^{0.22} \). The bilinear material model has Young’s modulus \( E = 206754 \) MPa, Poisson’s ratio \( \nu = 0.3 \), yield stress \( S_y = 305 \) MPa and tangent modulus \( E_T = 780 \) MPa. In the simulation, the friction coefficient between the punch and the die with the workpiece is 0.1. The problem is simulated as an axisymmetric problem and the simulation is performed for a total punch stroke of 10 mm.

A total of 2,556 elements were used to mesh the entire domain, with 1,353 elements used for the initial configuration of the workpiece. The mesh used in the domain has a higher intensity in the bottom half of the domain to increase the accuracy of the curved workpiece shape around the contact area. Elements that are in contact with the punch land are defined to take the shape of the punch during the whole simulation with the help of the contact-modified connectivity process. The bottom edge of the workpiece is fixed in the vertical direction, but left free in the horizontal direction. The process is solved as an axisymmetric FE model by using only the right half of the model and applying axisymmetric boundary conditions on the left edge of the model.

Figure 6.15 shows the original undeformed shape of the workpiece and the deformed shape at 2, 4, 6, 8 and 10 mm punch strokes. Note that at the region where the punch
contacts the workpiece, the material flows smoothly around that corner and goes into the
groove between the punch and the die without any complications. Also, the curved part of
the punch land is accurately represented using the contact algorithm in our VOS code in
addition to the weighting factor that was used in the connectivity procedure.

The load-stroke diagram for the backward extrusion is shown in Figure 6.16. The
values are comparable to the ones presented in [80]. The differences between the two
curves could be attributed mainly to the difference in the FE formulation of the two
methods in addition to the approximation of the material model used in the VOS
simulation. Usually a flow stress relation, like the one used in [80], has a curved shape
stress-strain relation where in our model we used an elastic-plastic model that has one
slope for the elastic region and one for the plastic region. This difference in the material
model causes the results to differ at some regions, and come closer at others.

The contact pressure at the initial contact region between the workpiece and the rigid
die is shown in Figure 6.17 for different punch strokes. Compared to [80], the contact
pressure in both models starts increasing rapidly as we progress, until we reach the last
stages of the process where it starts decreasing gradually. We also notice that the region
with nonzero contact pressure starts decreasing as we progress through the stages of the
process. The reason behind this is that the material starts moving more smoothly as the
material of the workpiece enters the plastic region. With the difference in the material
model and the fact that a rigid die is used rather than an elastic one as in [80], the
maximum values are within an acceptable range.

The development of the von-Mises strain distribution is shown in Figure 6.18 for
various punch strokes. The development of the strains is very logical throughout the
simulation since the strains start to build up around the punch land corner where the
material experiences high rotations and large deformation. The bottom right side of the
workpiece, where it meets the die, has the lowest strain values during the whole process
and this is because the material gets pressed against the die in that region and stays at the
same state during the entire process. This is also one of the reasons why the contact
pressure values are high at that region. Compared to the strain distribution at the final
stage (punch stroke of 10mm) presented in [80], the values in our work are in very good
agreement, and the distribution is quite similar in the two cases. Table 6.1 compares the values of the maximum pressure at different punch strokes and the strain values at the final stage. Overall, values presented using VOS are very close to those in [80].

The VOS method was used in this application to demonstrate its ability in handling nonlinear problems with large deformation and rotations and. Backward extrusion is considered to be among the metal forming application that are difficult to simulate by means of regular finite element formulations. The process was simulated efficiently with no need for re-meshing techniques or any mixed formulation. The combination of the Eulerian formulation and the VOS method presented the expected deformed shape of the workpiece in such applications. The results obtained from the VOS method were very good compared to results from the literature, and the distribution of these values follows the common knowledge that is known from this application.

Figure 6.14 Geometrical setup for the backward extrusion problem
Figure 6.15  Evolution of the workpiece deformed shape in the backward extrusion problem using VOS
Figure 6.16  Load-Stroke curve for the backward extrusion problem

Figure 6.17  Contact pressure (MPa) between workpiece and die at different strokes in the backward extrusion problem
Figure 6.18  von-Mises strain distribution for the backward extrusion problem
<table>
<thead>
<tr>
<th>Method</th>
<th>Max. contact pressure (0.9mm)</th>
<th>Max. contact pressure (3.4mm)</th>
<th>Max. contact pressure (7.3mm)</th>
<th>Max. contact pressure (8.6mm)</th>
<th>Max. contact pressure (10mm)</th>
<th>Max. effective strain value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOS</td>
<td>648.86</td>
<td>891.33</td>
<td>1051.9</td>
<td>1066.5</td>
<td>1054.8</td>
<td>2.5</td>
</tr>
<tr>
<td>Hue et al. [80]</td>
<td>858.28</td>
<td>967.255</td>
<td>1037.95</td>
<td>1027.11</td>
<td>998.363</td>
<td>2.4</td>
</tr>
</tbody>
</table>
6.5 Local Loading of a Simply-Supported Plate

The plate indentation in this example is simulated using two different indenters; a flat-faced and hemispherically-tipped indenters. The process consists of a circular steel plate that is loaded locally while the plate is resting on a ring support. The VOS method is used in this example to show the capability of dealing with sharp corner contacts (flat indenter) and curved shape contacts (hemispherical indenter) without the need to change the contact options or parameters between the two cases. Avoiding mesh distortion while accurately representing the contact face without any penetration is accomplished easily using the VOS with the Eulerian formulations. The two cases are compared with experimental results [81] and numerical results using ALE formulation [28].

The plate is 6 mm thick, 300 mm in diameter and is simply supported at a diameter of 248 mm. Both indenters are 12.7 mm in diameter and are loading the plate at a rate of 3 mm/min. The plate is made of mild steel and is assumed to be bilinear elastic-plastic with Young’s modulus of elasticity $E = 235$ GPa, Poisson’s ration $v = 0.3$, yield stress $S_y = 305$ MPa and a tangent modulus $E_T = 1000$ MPa. The process was simulated for a total displacement of 20 mm for the flat-faced indenter and 30 mm for the hemispherically-tipped indenter. Figure 6.19 shows the setup with the flat and hemispherical indenters.

An axisymmetric finite element model was used for this process. A 150 x 30 mm window, meshed into 1191 elements, is used for the flat indenter case whereas a 150 x 40 mm window is used for the hemispherical indenter with 1566 elements. Both Eulerian grids had 262 elements initially filled with the initial configuration of the steel plate. The contact is defined between the rigid indenters and the plate with the contact-modified connectivity process that ensures an accurate shape of the indenters land shape. The contact is assumed to have no friction and no separation between the indenter and the plate. The mesh is refined at the top right corner of the grid domain to have a better prediction for the shape for the sharp corner of the steel plate since the corner is connecting two perpendicular lines and the VOS method uses a single free surface line at each boundary element. The smaller elements allow the sharp corner to progress through the simulation while keeping the geometry intact.
Figure 6.20 shows the final deformed shape of the finite element model for the circular steel plate when loaded using the flat-faced and the hemispherically-tipped indenters, respectively. Compared with the deformed shapes presented in [28], the deformed shapes using the VOS method are in a very good agreement with those obtained using ALE.

The load-displacement curves obtained from the VOS method, ALE [28] and the experiment [81] are shown in Figure 6.21 and Figure 6.22 for the two cases. The experimental results in [81] showed that the plate fails by perforation at approximately 20 mm under the flat-faced indenter and 30 mm under the hemispherically-tipped indenter which is why the simulation was performed for these two values. From the two figures it is clear that the results obtained from the VOS method are in a good agreement with the experimental and ALE results. Even compared with the ALE results, the results from the VOS method are closer to the experimental ones. Generally we can see that at higher deformation levels, the numerical curves starts to deviate from the experimental one, and that is due to the discrepancies between the actual material model and the bilinear elastic-plastic model used in the numerical simulations.

Figure 6.23 compares the load-displacement curves between the flat and the hemispherical indenters. The flat indenter produces a slightly steeper curve than the hemispherical indenter, but with a lower failure value. From the figure we can see clearly that the indenter profile affects the indentation process to a great deal. The hemispherical indenter causes a higher degree of local indentation and more bending around the contact region than the flat indenter which the sheering effect to decrease allowing the plate to withstand higher load values before perforation occurs.

Figure 6.24 and Figure 6.25 show the deformed shape and the von-Mises strain distribution around the contact region. The deformed shapes and the strain distribution values are very similar to those obtained using ALE [28] for the hemispherically-tipped indenter, whereas the flat-faced indenter has some differences in the strain distribution around the indenter corner. The reason is because we used a rigid indenter and modeled the contact between the indenter and the plate, where in ALE results the indenter was substituted with an equivalent boundary condition to simulate its effect. In the rest of the
region the values and the distribution were very similar. We can observe the indenter land
is flat in the first case which shows that the VOS method handled the contact with no
penetration. In the second case, the effect of the curved shape indenter was presented
accurately with the VOS method. 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. The VOS method eliminated the need to deal with the mesh
distortion under the hemispherically-tipped indenter.

![Figure 6.19 Geometrical setup for the plate indentation problem](image-url)

*Figure 6.19* Geometrical setup for the plate indentation problem
Figure 6.20  Deformed shape of the plate indentation problem using VOS

Figure 6.21  Load-Displacement curve for the flat plate indentation problem
Figure 6.22  Load-Displacement curve for the hemispherical plate indentation problem

Figure 6.23  Comparison of the load-displacement curves of the flat and the hemispherical indenters
Figure 6.24  von-Mises strain distribution using the flat-face indenter in the plate indentation problem

Figure 6.25  von-Mises strain distribution using the hemispherical indenter in the plate indentation problem
6.6 V-Bending

In this application, the V-bending process is simulated to demonstrate the capability of the VOS method in dealing with sharp corner contacts and in handling complex changes in nonlinear boundary conditions during the stages of the simulation. The simulation results from the VOS method are compared to experimental results [28] and numerical results from ANSYS [76].

The material used for the sheet metal in this application is Aluminum 5052-H32. A tensile test following the ASTM standards is performed using a Tinius Olsen Testing Press to obtain the stress-strain curve of the material. The material behavior is approximated to a bilinear elastic-plastic model with a Young’s modulus $E = 70$ GPa and Poisson’s ratio $v = 0.33$, initial yield stress $S_y = 120$ MPa and tangent modulus $E_T = 580$ MPa. The punch and the die are made from a high strength Mild Steel 1018 that is machined to the required dimensions. The process consists of a $90^\circ$ V-shaped punch pressing against an aluminum sheet into a $90^\circ$ V-shaped die with a total punch stroke of 24.5 mm in the downward direction. Figure 6.26 shows the setup of the V-bending application with the dimensions of each part. The experiment is performed using the Tinius Olsen Testing Press. All the parts shown in Figure 6.26 are machined for this test. The punch load and displacements are recorded during the experiment to be later compared with the numerical results. The same process is simulated using ANSYS with the workpiece meshed into 144 uniform 4-noded elements and treated as a plain strain problem.

Due to symmetry, the VOS simulation is performed for half of the model using a total number of 8,057 elements for the whole domain with 1,495 of these elements being initially filled by the metal sheet in its initial configuration. The Eulerian grid domain is meshed using a uniform grid at some regions and a non-uniform grid at other regions. This is done to optimize the number of elements at regions that are expected to be empty during the course of the deformation. Considering the right half of the V-bending model, the mesh intensity is set to be low at the top left and bottom right corners of the domain since those two regions are expected to be empty during the entire process. The die is modeled as a rigid body using line segments fixed in all directions. The punch is modeled
as a rigid body using three line segments, all fixed in the horizontal direction. Each line of the punch and the die is defined as smaller segments at the elemental level. Two contact pairs are defined in the simulation. The first one is between the punch and the workpiece and the second one is between the die and the workpiece with a coefficient of friction of 0.1 for both pairs. The contact algorithm in the VOS simulation defines the contact at the elemental level, and the contact is detected once the rigid line elemental segments contact a free surface interface. After the initiation of the contact at each free surface element, the coordinates of the free surface element line and the rigid line segment are monitored to detect any update in the contact condition. A connectivity weighting factor of 1 is given to the bottom boundary of the metal sheet at the region under the punch. This ensures that the sheet will maintain its thickness since the part in contact with the punch dominates the connectivity process in the upper boundary of the workpiece. All subsequent regions share the same weighting factor at both sides of the workpiece.

Figure 6.27 shows the evolution of the deformed shape for the metal sheet during the experiment and the VOS simulation. Comparing the deformed shapes of the metal sheet, it is possible to see that the deformed shape is accurately presented in VOS. The contact with the corner point remained during the entire process with hardly any penetration which is difficult to maintain when using traditional Lagrangian or ALE formulations. The contact algorithm is part of the VOS method which incorporates the contact boundary conditions as a part of the free surface detection scheme. This arrangement allows the method to eliminate any penetration during the simulation process.

Figure 6.28 shows the load-displacement curve obtained from the VOS simulation, ANSYS and the experimental results. In general, the load-stroke curve has a similar behavior in the three solutions. However, at the last stages of the process, the punch load starts to increase at a very high rate due to the fact that most of the workpiece boundary comes in contact with the punch and the die. The VOS and ANSYS curves have load values that are much higher than the experimental curve. This difference is attributed to the fact that the die in the experiment is designed with a stress relaxation gap to prevent overloading the load cell of the testing press at maximum punch displacement. Another
observation is that before the last stages of the process, the load values of the VOS method are slightly higher than in ANSYS and the experiment. This is due to the approximation of the material model that was used in the VOS simulation.

The development of the von-Mises stresses and strains at different punch strokes using the VOS method are shown in Figure 6.29 and Figure 6.30, respectively. It can be seen from the distribution that the maximum values for the stress and strain occur at the punch land towards the center line (line of symmetry). This distribution is very logical since that region is where the sheet undergoes the maximum compression (top boundary) and tension (bottom boundary). Table 6.2 compares the maximum values for the von-Mises stress and strain from the VOS method and ANSYS. The results from VOS are very close to the values obtained using ANSYS. The slight differences in values are due to the approximation of the material modeling and the difference in the formulation between the Lagrangian formulation used in ANSYS, and the Eulerian formulation used in VOS.

The VOS method was validated in the above example by comparison with experimental work [28] and numerical analysis using commercial FE software (ANSYS). The development of the sheet metal deformation was presented accurately using VOS, and the shape of the free boundary was very similar to the experiment. Even though there are no mesh distortion problems in such applications, simulating such process using the VOS method highlights the capability of the method in dealing with large rotations and change of free boundaries along the whole surface of a workpiece rather than just one or two sides. The VOS also showed how easily it can deal with sharp corner contacts without the need for any special contact algorithm as it deals with it using the default contact parameters. In addition to estimating the shape of the free surface, different results were validated by comparing VOS to ANSYS, showing that the VOS method can be used effectively to capture the free surface in an Eulerian grid in addition to calculating accurate results and distribution of the stresses and strains.
Figure 6.26  Geometrical setup for the V-bending problem
Figure 6.27  Evolution of the deformed shape from Experiment and VOS in the V-bending problem

Figure 6.28  Load-Stroke curve for the V-bending problem
Figure 6.29  von-Mises stress distribution for the V-bending problem.
Figure 6.30 von-Mises strain distribution for the V-bending problem

Table 6.2 von-Mises stress and strain comparison between VOS and ANSYS for the V-bending problem

<table>
<thead>
<tr>
<th></th>
<th>Punch Stroke 6.35mm</th>
<th>Punch Stroke 12.7mm</th>
<th>Punch Stroke 19.05mm</th>
<th>Punch Stroke 25.4mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress (VOS)</td>
<td>208.8 MPa</td>
<td>238. MPa</td>
<td>241.7 MPa</td>
<td>244.5 MPa</td>
</tr>
<tr>
<td>Stress (ANSYS)</td>
<td>216.4 MPa</td>
<td>231.6 MPa</td>
<td>236.6 MPa</td>
<td>242.3 MPa</td>
</tr>
<tr>
<td>Strain (VOS)</td>
<td>0.133763</td>
<td>0.23644</td>
<td>0.368352</td>
<td>0.516223</td>
</tr>
<tr>
<td>Strain (ANSYS)</td>
<td>0.130869</td>
<td>0.228923</td>
<td>0.359114</td>
<td>0.462803</td>
</tr>
</tbody>
</table>
6.7 Three-Point Bending

In this application, the three-point bending (U-bending) is simulated to show the capability of the VOS method in dealing with large curved contact areas. Since the punch in this example is a cylinder and the dies are rollers, we expect the contact regions to be highly curved throughout the simulation in which the VOS can accurately predict the deformed shape of the workpiece while keeping the free surface elements intact. The VOS simulation is compared to experimental results.

The process consists of a punch with a cylindrical tip that presses against a sheet metal that is supported by two rollers. The rollers are connected to a supporting frame that allows the rollers to rotate freely. The punch and the rollers are all made from mild steel 1018 with Young’s modulus $E = 205$ GPa, yield strength $S_y = 370$ MPa, ultimate strength $S_u = 440$ Mpa and density $\rho = 7870$ kg/m$^3$. The sheet is made from Aluminum 5052-H32 with Young’s modulus $E = 70$ GPa and Poisson’s ration $v = 0.33$. A tensile test following the ASTM standards was performed using a Tinius Olsen Testing Press to obtain the stress-strain curve of the material. The material behavior was approximated to a bilinear elastic-plastic model for the VOS simulation with yield stress $S_y = 195$ MPa and a tangent modulus $E_T = 280$ MPa. The geometrical setup of the problem is shown in Figure 6.31. The process was performed for a total punch stroke of 76.2 mm with a constant velocity of 6.35mm/min. The experiment was performed using the Tinius Olsen Testing Press with all the parts being machined for this test. The punch load was recorded during the experiment for different punch strokes.

The VOS simulation was performed for half of the model due to symmetry. An 80 x 105 window, meshed into 8766 elements, was used for the Eulerian domain with 539 elements filled by the initial configuration of the sheet. The punch and the roller were modeled as a quarter and a half of a circle using segments of rigid lines. The only boundary conditions assigned to the workpiece were the symmetry boundary conditions at the line of symmetry. The average coefficient of friction for aluminum used in this simulation is 0.1. Similar to the V-bending problem (Section 6.6), a connectivity weighting factor of 1 is given to the bottom boundary of the sheet at the region under the punch. This ensures that the sheet will maintain its thickness since the part in contact with
the punch dominates the connectivity process at the upper boundary of the workpiece. All subsequent regions share the same weighting factor at both sides of the workpiece.

Figure 6.32 shows the evolution of the deformed shape at different punch strokes during the VOS simulation and the experiment. It can be seen that the deformed shape from the VOS simulation is presented accurately compared to the deformed shape from the experiment. The punch load-stroke curve obtained from the experiment is compared to the one obtained from the VOS simulation in Figure 6.33. The curve from the VOS results matched very well with the experimental curve. At the beginning, the punch load increases with the punch stroke until it reaches a peak value when the workpiece bend flange becomes tangent to the die face. Then the load starts to decrease as the workpiece slides to the side of the rollers and the punch.

The development of the von-Mises stresses and strains at different punch strokes using the VOS method are shown in Figure 6.34 and Figure 6.35, respectively. The stresses and strains start to develop at the region under contact with the punch. As the process continues and the contact area with the punch increases, stresses and strains starts to generate and extend towards the end of the workpiece. The values are always higher under the punch land because that is where the workpiece undergoes the maximum tension (bottom boundary) and compression (top boundary).

The simulation of the three-point bending problem using the VOS method was compared to experimental results obtained using a similar setup to the V-Bending problem in Section 6.6 [28]. The evolution of the deformed shape was modeled accurately compared the recoded deformation shapes from the experiment. The VOS method handled the change in contact condition and the large rotations effectively. The change in the free boundary was tracked using the VOS method during the course of the simulation. The VOS method also showed how easily it can deal with sharp corner contacts without the need for any special contact algorithm as it deals with it using the default contact parameters.
Figure 6.31 Geometrical setup for the three-point bending problem
Figure 6.32  Evolution of the three-point bending deformed shape from the Experiment and VOS
Figure 6.33  Load-Stroke curve for the three-point bending problem

Figure 6.34  von-Mises stress distribution for the three-point bending problem
Figure 6.35  von-Mises strain distribution for the three-point bending problem
CHAPTER 7
METAL CUTTING APPLICATION

Metal cutting is considered to be one of the most complicated manufacturing processes. Even though the basic mechanics are almost the same for all metal cutting operations, each case is considered different due to the complexity that comes from the dependency of the process on many geometrical factors and material properties. Most of the research done in metal cutting used to be based on experimental work but since the advancements of the Finite Element Method (FEM) and numerical simulations in the last few decades there has been more work on studying the machining process by means of numerical analysis. Compared to experiments, presenting accurate results from numerical simulations is still considered a challenging task due to the coupling of high temperatures and high strain rates, and the difficulty in modeling the friction and the contact between the chip and the tool. Due to the large plastic work and high friction, massive amounts of thermal energy are generated causing the temperature to rise in the metal and the tool. This means that an accurate numerical simulation must consider appropriate heat transfer analysis along with the deformation and stress analyses [24, 77, 82].

The natural FE formulation choice in metal forming applications, including machining, is the Lagrangian approach because the mesh is attached to the material points and undergoes the same deformation. In metal cutting this takes care of an important issue which is the unconstrained material flow on the free boundaries during the formation of the chip throughout the machining process. This eliminates the need for a priori assumption for the shape of the chip at steady state stages. This is important since the shape of the chip is highly dependent on the friction coefficient between the chip and the tool, the cutting speed and the shape of the tool (rake angle, chamfered, blunt…etc) [77, 83, 84].

The most common way to simulate metal cutting with Lagrangian formulation is to use a nodal separation method in conjugation with a failure criterion [85-89]. In this
approach, the line of separation at the cut depth is defined prior to the analysis and the failure criterion is applied to the nodes/elements along that line. As the tool advances through the metal, the nodes start to detach in front of the tip of the tool based on the failure criterion. Usually the separation aspect of the analysis is modeled in a similar fashion to an elastic-plastic crack propagation problem. One of the drawbacks of this approach is that the material separation happens at a small distance in front of the tool tip rather than at the tip itself [86-88]. Physically, this is an unrealistic approach since such cracks are not observed in actual cutting of ductile materials. Additionally, the process will be modeled with discrete jumps of the crack tip from one node to the next one which requires a very fine mesh in order to minimize the error of the material deformation history results at the vicinity of the tool tip. Although this might be reasonable in some cases where the distance between the tool tip and crack tip is too small, it can only perform well for tools with finite nose radius and fails in applications with chamfered or blunt tools [25, 77]. With chamfered and blunt tools, there exists a dead metal zone under the chamfered length or the curved radius that cannot be modeled properly with nodal separation methods. In addition to that, one must pay attention to the unzipped nodes along the parting line since the line might be pushed out of its position as the tool advances. Another difficulty with this approach is the choice of a failure/damage criterion with the proper values. Most of these criterions are based on parameters that are obtained from experiments and once the experimental parameters or the material used in the process are changed, the criteria will not be valid. Different numerical simulations were carried out using different values and different criterions [88-90]. It is noteworthy to know that Huang and Black [91] concluded after studying different criteria that the type of criterion used in the analysis is not significant but the magnitude of the failure critical value has a great effect on the distribution of the stresses and strains.

A viable substitute for the nodal separation method could be achieved by simulating the cutting process as an indentation problem by having the tool pressing against the workpiece causing the material (elements) to flow around the tool [75, 84, 92, 93]. This approach requires extensive remeshing around the tool tip at very small time increments due to the enormous distortion of the elements around the tool. Although this approach overcomes most of the difficulties and problems related to the nodal separation method, it
does have its own drawbacks. The constant remeshing of the workpiece in addition to the fine mesh required around the tool leads to huge computational cost [93]. Another negative aspect of this approach is the interpolation of the material variables from the old mesh to the new mesh after each remeshing step. Because of the difference between the old and the new mesh configuration, the approximation that is used in the interpolation process can lead to error accumulation at each remeshing step which may cause deterioration of the results [24].

A less common choice for metal cutting simulations is the Eulerian formulation approach. This approach eliminates some of the problems that arise when using any of the Lagrangian approaches. Since the mesh is fixed in space and the material is allowed to flow freely around the tool, the cutting process can be modeled as an indentation problem without the need for a failure criterion. Additionally, since no mesh distortion occurs in an Eulerian grid, no remeshing steps are required. This makes the Eulerian approach computationally efficient and appropriate for modeling the flow around the tool tip. The main downside of this formulation is that it is not easily adaptable for modeling the unconstrained flow of the material at the free boundaries as the chip evolves during the process [25, 94-96]. To overcome this problem, a priori assumption for the shape of the chip at its steady state is required and the process is simulated only for the steady state stages to estimate the values of the material variables around the chip-tool area. A proper assumption of the chip shape is very difficult to obtain since it depends on many factors and it cannot be predicted accurately without performing experiments with the same cutting conditions.

An alternative approach is ALE formulation. Some of the work done with ALE in metal cutting focused on improving the nodal separation method by allowing a more natural treatment of the metal cracking [23]. This is achieved by minimizing the “unrealistic” distance between the tool tip and the separation nodes. ALE can also be used in conjunction with remeshing techniques to minimize the number of remeshing steps. The ALE can also be used to take full advantage of the Lagrangian and Eulerian methods and eliminate the need for remeshing and nodal separation. This requires a proper mesh motion scheme that allows the free flow of the material around the tool
edge, as in the Eulerian approach, and the unconstrained flow of the material at the free boundaries, as in the Lagrangian approach, to develop the chip shape naturally [24, 25, 77]. The ALE approach proved to be viable once a proper mesh motion scheme has been implemented in the analysis. Establishing a general and reliable mesh motion scheme for ALE applications in metal cutting is not, however, an easy task although some effort has been done in this direction [23-25].

In this chapter we present a study of the metal cutting process with one sharp tool, two chamfered tools and one blunt tool that are used under the same cutting conditions. The study is carried numerically using the VOS method and results are compared to the numerical work done by Movahhedy et al. [77] using ALE as well as experimental results by Ren and Altintas [97] that were performed for three of the four different tools to compare the forces acting in the process. Figure 7.1 shows the schematic of the metal cutting process for a general case with a chamfered tool. The thrust and cutting forces are defined by $F_t$ and $F_c$, respectively, where $F_s$ and $F_n$ represent the forces acting along and normal to the shear plane, respectively. The rake and chamfer angles are defined as $\alpha_0$ and $\alpha_f$. The length of the chamfered section is defined as $b_{cf}$.

![Figure 7.1 Schematic view of metal cutting process with chamfered tool](image)

Figure 7.1  Schematic view of metal cutting process with chamfered tool

120
7.1 Experimental Work and Numerical Setup

In this work we used P20 steel for the workpiece and carbide of ISO S10 class for the tools. Three sets of orthogonal cutting tests were performed using one sharp and two chamfered tools [97]. The P20 mold steel disks were turned in thrust orthogonal mode on a CNC turning center. All of the tests were performed with a 0° rake angle and the only differences between the three cases were in the chamfer angles and lengths. The cutting speed was set at 240 m/min with a chip load of 0.1 mm. The same three cases were simulated using the VOS method in addition to an extra case with a blunt (worn) tool with a radius of 0.075 mm. Table 7.1 lists the four cases used in the experiment and the simulation.

Table 7.1 List of experimental and numerical metal cutting setups

<table>
<thead>
<tr>
<th>Case</th>
<th>$\alpha_1$</th>
<th>$b_{cf}$ (mm)</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0°</td>
<td>-</td>
<td>Exp. &amp; VOS</td>
</tr>
<tr>
<td>2</td>
<td>-10°</td>
<td>0.0902</td>
<td>Exp. &amp; VOS</td>
</tr>
<tr>
<td>3</td>
<td>-25°</td>
<td>0.0863</td>
<td>Exp. &amp; VOS</td>
</tr>
<tr>
<td>4</td>
<td>blunt</td>
<td>Radius 0.075</td>
<td>VOS</td>
</tr>
</tbody>
</table>

The flow stress model used for the P20 steel workpiece is dependent on the strain rate and the temperature in the following form [77, 97]:

$$\bar{\sigma} = (A + B \bar{\varepsilon}^m)(1 + C \ln \dot{\varepsilon})(D - ET^* n)$$

with the variable $T^*$ defined as:

$$T^* = \frac{T - T_{room}}{T_{melting} - T_{room}}$$
where $A = 178.5$ MPa, $B = 462.4$ MPa, $C = 0.0438$, $D = 1.51$ and $E = 0.99$ are the coefficient for the flow stress model. The strain sensitivity parameter $m$ is equal to 0.169 and the temperature sensitivity parameter $n$ is equal to 0.666. The melting temperature for $P20$ is 1480°C. The rest of the $P20$ steel material properties are listed in Table 7.2 along with thermal properties of the carbide tool. All of the workpiece properties are listed for the $P20$ steel except for the convection heat transfer coefficient where we used a value for C45 steel [78] due to the difficulty in obtaining any values for $P20$ in the literature. The tool was considered rigid for the stress analysis part in the numerical simulation and only the thermal heat transfer analysis was performed for the tool.

It is difficult to apply perfect contact conditions in metal cutting simulations and different values and settings have been used in previous work done in the area. The general consensus in the literature is that a mean friction coefficient value can be obtained from the ratio of the experimental values of the thrust and cutting forces with the same conditions. The mean values obtained from the experimental work [97] fall within the range of values presented in the literature [27, 77, 78, 92, 98].

### Table 7.2  
**Material properties for the workpiece and tool used in metal cutting**

<table>
<thead>
<tr>
<th>Material Properties</th>
<th>$P20$ Workpiece</th>
<th>Carbide Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Young’s Modulus (GPa)</td>
<td>260</td>
<td>-</td>
</tr>
<tr>
<td>Poisson’s Ratio</td>
<td>0.3</td>
<td>-</td>
</tr>
<tr>
<td>Thermal Conductivity, $k$ (W m$^{-1}$°C$^{-1}$)</td>
<td>51.5</td>
<td>120</td>
</tr>
<tr>
<td>Specific Heat (J Kg$^{-1}$°C$^{-1}$)</td>
<td>470</td>
<td>343.3</td>
</tr>
<tr>
<td>Thermal Expansion Coefficient (°C$^{-1}$)</td>
<td>1.4x10$^{-6}$</td>
<td>5.2x10$^{-6}$</td>
</tr>
<tr>
<td>Thermal Convection Coefficient, $h$ (W m$^{-2}$°C$^{-1}$)</td>
<td>20</td>
<td>-</td>
</tr>
</tbody>
</table>

### 7.2 VOS Simulation of Metal Cutting

A combination of uniform and non-uniform mesh was used for the Eulerian domain. Regions outside of the deformation zone were constructed using coarse mesh to minimize
the number of elements that are not used in the analysis (Figure 7.2). The left region of the domain, where the velocity is applied to model the movement of the workpiece towards the tool, was constructed using a uniform grid with fairly sized elements. In a similar way, the right region where the material exits the machining section was also constructed with fairly sized uniform grid. This is because the material flow is steady in the horizontal direction and there is no change in flow movement or in the general shape. The region enclosing the tool was also meshed with relatively medium sized elements except toward the edges where the contact between the tool and the workpiece occurs. A fine mesh was used around the contact region to get better results in the thermal analysis. Finally, the middle part is the most important part of the domain since it is the region where the chip deformation is expected to take place. This part was constructed with very fine mesh to allow more accurate chip tracking results and better chip shape. Since the VOS method is based on constructing linear lines at the interface of the free surface elements, the finer the mesh the more accurate the free surface shape results become.

Figure 7.2 shows the evolution of the chip deformation through the cutting process for the sharp tool. In our VOS FE-code, the stress and the thermal analyses were performed in a staggered fashion. At first, the stress analysis is performed with the workpiece and tool being at room temperature. The temperature data calculated from the internal heat generation in stress analysis is then transferred and the thermal analysis is performed. After that, the new stress analysis step is performed using the temperature data calculated from the thermal analysis as an input. Although this approach does not represent the physical dynamics of the process, it is not expected to introduce any other approximations beyond those adapted in a staggered approach especially since the material properties dependency on the temperature is weak and the time step used is very small. In addition to that, the scheme is easy to implement and the results from this work and from [25, 77] proved to be stable and acceptable.
When studying metal cutting process with chamfered and blunt tools, one of the main points of interest is the existence of the dead metal zone under the chamfer or the worn part of the tool. Experimental and analytical studies stated that a dead metal zone is formed in those regions and that it is mostly independent of the cutting conditions and it depends only on the geometrical profile of the chamfered part [25, 97, 99, 100]. The explanation is that this zone is supposed to fill the part under the chamfer which makes the tool work like a sharp tool with dead metal zone acting as the effective cutting edge of the tool.

To examine this phenomenon, we study the velocity field of the material around the tool edge. Figure 7.3 shows the velocity profile of the four different tools used in the VOS simulations. For the sharp tool, it is clear that there is no dead metal zone except for a very small region where the material comes in contact with the tool corner causing the material flow to slow down. For the chamfered tools, it is clear that part of the material is

7.3 Results and Discussion

Figure 7.2 Evolution of chip deformation in the VOS metal cutting simulation
trapped under the chamfer, with low velocities, and almost completes the missing tool nose. The zone is clearer for the case where the chamfer angle is -25° since the area of the missing nose is bigger which shows that it is dependent on the geometry/chamfer angle. As for the blunt tool, a dead metal zone occurs but to a lesser extent since the curved part of the tool allows a smoother flow of the material.

Another interesting observation is the size of the region where material velocity starts to increase from relative stagnation to the working velocity of the workpiece under the tool flank. For the sharp tool, the material almost directly picks up the working velocity. For the chamfered cases, the material takes more time to get back to the working velocity again which also increases with increased chamfer angles. The blunt tool has the largest extent of the stagnation region because the curved part of the tool reaches beyond the corners of the chamfered tools. This occurs because of the dead metal zone which develops a tool nose that is not on the same level of the tool flank causing additional straining at that region.

The distribution of the effective stresses is demonstrated in Figure 7.4 for the four different tool geometries. The shear zone is clear in these contours which possesses the highest stress values and is concentrated between the beginning of the chip curve at the free surface to the edge of the tool or the dead metal zone. The stress value is almost constant for all cases along the shear plane. It is also clear that for blunt and higher chamfer angle tools; high stresses extend beyond the shear plane towards the flank of the tool because they allow a smoother flow of the material around the tool edges which also drags the high stresses out of the shear plane causing some stress relieving towards the center of the shear zone.

The same observations regarding the effect of the dead metal zone and the shear zone can be seen in the distribution of the plastic strain rate in Figure 7.5. The strain rates are concentrated at higher values along the shear plane than, where in the rest of the workpiece strain rates are much smaller and fairly constant. The distribution also has large values towards the flank of the tool due to the extrusion of the material under the dead metal zone. The effect of the chamfer angle or a blunt tool can also be observed here.
by noting how longer it takes for the material strain rate to drop to lower values after is
passes the tool flank area.

Although the chip evolution has a weak dependency on the chamfer angle or radius
of the blunt, these factors do have a significant effect on the tangential and, to a lesser
extent, on the normal forces. Figure 7.6 compares the maximum stresses in the tangential
and normal directions around the tool nose obtained from the VOS simulation and the
ones from ALE [77]. It is clear that the increase of the chamfer angle causes a slight
increase in the normal stresses acting along the cutting direction. On the other hand, the
stresses in the thrust direction, that is tangent to the tool rake face, show an evident
increase as we go from a sharp tool to a chamfered and a blunt tool. The increase of the
chamfer angle has a more dominant effect in this direction compared to the normal
direction. This is logical since the presence of a chamfer causes the buildup of the dead
metal zone which strains the material to a larger extent compared to the sharp tool.

Figure 7.7 and Figure 7.8 compares the cutting and thrust forces obtained from VOS,
ALE [1] and the experimental work, respectively. It is obvious that both forces increase
with the increase of the chamfer angle. As for the stresses, the force in the cutting
direction is less affected by the chamfer angle and the tool radius in contrast to the forces
in the thrust direction which display a more pronounced increase in values. The same
observation can be seen from the experimental results which exhibit similar behavior to
that of the VOS and ALE results. The forces calculated for the blunt tool are higher than
the ones for the sharp and chamfered tools. A rational explanation for this is that the
higher and lower portions of the worn tool curve may act as a chamfered edge with a
small and large chamfer angles, respectively. There is a notable difference between the
numerical and the experimental values which is attributed to the simplification of the
friction conditions used in the numerical simulations and the uncertainty of the material
model and its thermal properties used in the VOS and ALE simulations.

The results from the VOS simulation showed good agreement in values and behavior
with the ones obtained from the proposed ALE approach by Movahhedy et al. [77]. The
effect of the chamfer angle and blunt radius on the cutting and thrust forces in the VOS
and ALE simulation possesses the same behavior of the experimental results [97]. Some
differences in the values between two approaches and the experiment can be credited to the assumptions and simplifications made in the friction and material model. In general, the VOS shows a great promise in handling continuous metal cutting applications and the differences between the VOS and experimental results were due to the approximation in the friction and material modeling. It is important to note that the VOS method is only suitable for continuous chip formation and it cannot handle segmented chip formation in its current state.

Figure 7.3  Velocity fields for sharp tool, -10° chamfered tool, -25° chamfered tool and a blunt tool
Figure 7.4  Effective stress distribution for sharp tool, -10° chamfered tool, -25° chamfered tool and a blunt tool
Figure 7.5 Plastic strain rate distribution for sharp tool, -10° chamfered tool, -25° chamfered tool and a blunt tool.
Figure 7.6  Comparison of maximum normal and tangential stresses between VOS and ALE simulations

Figure 7.7  Comparison between the cutting forces obtained using VOS, ALE and experimental results
Figure 7.8  
Comparison between the thrust forces obtained using VOS, ALE and experimental results
8.1 Summary and Conclusions

The simulation of metal forming and machining processes is considered to be one of the most challenging topics in the area of solid mechanics. Both traditional, Lagrangian and Eulerian, formulations have many drawbacks that hinders their use in this field of research. Different schemes and formulations have been presented in the literature to tackle these drawbacks and overcome the difficulties that arise when using FE in simulating metal forming applications. Re-meshing proved to be very effective in dealing with problems related to mesh distortion but on the expense of accuracy, computational time and problem dependency. The ALE formulation on the other hand showed that it may be used effectively in overcoming all the drawbacks associated with other formulations and schemes. The main issue in ALE is that devising a general and proper mesh motion scheme that can be used in wide range of applications is very difficult to attain, hence most of the mesh motion schemes are limited to certain class of applications.

In the present work we proposed a new method that is based on the Volume of Fluid (VOF) method [37] to be used with Eulerian formulations to calculate the volume of the solid material, hence called the Volume of Solid (VOS) method, and to track the unconstrained flow of the material at the free boundaries. A general review of the current FE formulations used in metal forming numerical simulations was presented in Chapter 1. In Chapter 2, a comprehensive survey of the VOF method and the different schemes used in the literature was presented highlighting the drawback/advantages of each scheme. This critical study is the basic ground for formulating the VOS method since understanding the different aspects of the VOF method helps in recognizing the proper approach to be used in formulating the VOS method for metal forming applications.
The Eulerian finite element equations were presented in Chapter 3 for quasi-static and dynamic analyses. The complete formulation and discussion of the proposed VOS method was presented in Chapter 4 for the case of a uniformly constructed mesh and for the general case of a non-uniform mesh. All topics related to the implementation of the VOS method were discussed in Chapter 5. These topics included the time step stability, updating the velocity field, tracking material point related properties and connectivity of the free surface interfaces. A contact module was also presented in addition to a heat transfer analysis procedures. Several metal forming and metal cutting simulations were performed in Chapters 6 and 7. The results were validated with different results obtained from the literature, experimental work and commercial FE softwares. The contributions from this work may be summarized in the following points:

- A complete derivation of the Eulerian FE equations based on the principle of virtual work was presented for quasi-static and dynamic analyses.
- A new method was introduced for the treatment of the convective terms in the Eulerian equilibrium equations based on the work by Bayoumi and Gadala [28, 68]. Unlike the commonly used operator split method where the equations are uncoupled into an advection and a non-advection terms, the new approach utilizes the definition of the VOS method in dealing with these terms without uncoupling the solution.
- The complete formulation of the VOS method was developed for uniform and non-uniform grids. The formulation included the computation of the fractional volume values, calculation of the free surface directional vector and the construction of the free surface interface. For the uniform mesh case, a PLIC-VOS method is used in addition to a general classification of all the possible free surface interface shapes. A new filling pattern scheme is used for the general case of non-uniform mesh. The filling scheme is developed to construct the free surface with no dependency on the shape of the element.
- A new free surface connectivity process was employed in the VOS formulation to ensure that each free surface interface is connected to the interfaces at
neighboring elements. The process also accounts for every connectivity step to satisfy the conservation of mass and ensure no volume loss during the process.

- A contact algorithm was used to handle the contact between rigid and flexible bodies. The new module allows the modeling of a moving rigid body within the Eulerian domain and tracks the contact condition. Metal forming simulations have been limited mostly to applications that do not require having a moving punch/die. The new algorithm utilizes the VOS method by recording the movement of the free boundaries and the rigid tool which widens the range of applications that can be analyzed.

- Several metal forming applications were simulated using the developed VOS FE-code. The applications covered a wide range of applications highlighting the power of the VOS method overcoming problems like the mesh distortion and modeling the unconstrained flow of the material. Different shapes of tools were used in these simulations to demonstrate the effectiveness of the proposed VOS formulation in handling the deformed shape under these tools. Dealing with sharp angles was also handled easily with no penetration.

- A complex simulation for metal cutting analysis was performed using the VOS method. The simulation consisted of a heat transfer model that was coupled with the stress analysis to include the effects of the heat generation and the heat transfer between the chip and the cutting tool. The results from this model were compared to experimental and numerical results using ALE formulation. No material separation criterion was needed and the VOS method simulated the process in a more natural way as an indentation problem.

8.2 Future Work

Even though the VOS formulation presented in this work proved to be a reliable FE formulation in simulating applications with large strains and deformations, there is still room for improvement. The VOS method is new and there is a lot of work to be done
towards the future progress of the method either by improving the current scheme or by implementing new techniques.

Different topics could be explored for the future development of the VOS method in addition to implementation of new schemes and techniques. Such topics can be summarized as follows:

- Since the first introduction of the VOF method, different schemes were presented in the literature to improve the efficiency of the method in tracking the free surface of the fluid. A diverse range of free surface re-construction methods were used for different application. Since the VOS method presented in this work acts as an introduction of the method into solid mechanics, more exploration of the VOF methods could lead to better and improved VOS formulations.

- Extending the VOS method to handle two (or more) flexible bodies; rather than one flexible and one rigid; would introduce a new class of applications that could be analyzed with this method. One of the main advantages of the proposed VOS method in this work is that it defines cells that are not occupied by the material as “true” empty cells, rather than assigning them fictitious values. This makes the formulation of a modeling two or more flexible bodies much easier.

- Improvements could be made on the connectivity process which is considered an important topic in the VOF literature. Additional controlling parameters could be introduced to improve the efficiency of the current connectivity process. New connectivity schemes may also be developed to enhance the connectivity of the free surface interfaces. The connectivity could also be incorporated in the actual formulation of the VOS equations to ensure a more natural connectivity rather than having a separate connectivity step performed after each time step. If a formulation for two or more flexible bodies is developed, then a new connectivity process must be derived to account for both bodies if they come into contact and ensure the mass conservation of both.

- An extension of the current contact module can be done to include the contact between two flexible bodies. This would widen the range of analyses that could be performed including tool design and flex-flex dynamic impact problems. If the
contact problem is elastic-plastic contact then modifications on the current module can be made to track the movement of the elastic body. If both bodies are expected to deform within the plastic range, then a new contact algorithm must be employed. The free surface connectivity of both bodies must be coupled when contact occurs to ensure the integrity of the contact is not lost during the connectivity process of each body.

- Extending the current VOS formulation to 3D analysis is a topic of interest. This would require modifications mainly in the free surface construction step. It would be challenging to classify all possible interfaces at the free surface for a uniform mesh. On the other hand, the filling pattern might be an easier choice to implement in 3D uniform mesh. In the case of non-uniform mesh, it is a difficult task to apply the filling pattern scheme as it would be complicated to choose a reference node and to calculate the free surface directional vector.

- Implementations of the VOF method in applications that involve fluid-structure interaction have been widely discussed. In these applications the structure is always fixed and the fluid is the main interest in the analysis. The fluid-structure interface can be explored from the VOS method point of view with the solid being the main point of interest. It would be interesting to derive a set of VOS-VOF equations that could be paired together in such applications.
REFERENCES


144


