Incremental Sheet Forming Process: Control and Modelling

Hao Wang
St. Edmund Hall

A thesis submitted to
The University of Oxford
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
in Engineering Science

Oxford, United Kingdom, October, 2013

To my parents

Wu Hong-Mei

Wang Shan-Jun

献给我的父母　吴红梅 王善俊

Abstract

Incremental Sheet Forming (ISF) is a progressive metal forming process, where the deformation occurs locally around the point of contact between a tool and the metal sheet. The final work-piece is formed cumulatively by the movements of the tool, which is usually attached to a CNC milling machine. The ISF process is dieless in nature and capable of producing different parts of geometries with a universal tool. The tooling cost of ISF can be as low as 5–10% compared to the conventional sheet metal forming processes. On the laboratory scale, the accuracy of the parts created by ISF is between ± 1.5 mm and ± 3 mm. However, in order for ISF to be competitive with a stamping process, an accuracy of below ± 1.0 mm and more realistically below ± 0.2 mm would be needed. In this work, we first studied the ISF deformation process by a simplified phenomenal linear model and employed a predictive controller to obtain an optimised tool trajectory in the sense of minimising the geometrical deviations between the targeted shape and the shape made by the ISF process. The algorithm is implemented at a rig in Cambridge University and the experimental results demonstrate the ability of the model predictive controller (MPC) strategy. We can achieve the deviation errors around ± 0.2 mm for a number of simple geometrical shapes with our controller. The limitations of the underlying linear model for a highly nonlinear problem lead us to study the ISF process by a physics based model. We use the elastoplastic constitutive relation to model the material law and the contact mechanics with Signorini's type of boundary conditions to model the process, resulting in an infinite dimensional system described by a partial differential equation. We further developed the computational method to solve the proposed mathematical model by using an augmented Lagrangian method in function space and discretising by finite element method. The preliminary results demonstrate the possibility of using this model for optimal controller design.

Acknowledgements

From the very first day at Oxford till the correction of the last typo of this thesis, I am deeply in debt to my supervisor, Professor. Stephen R. Duncan, who brought me into the academic world, patiently guided me through the very early days, and supported me in my darkest days.

I am also indebted to our collaborators at Cambridge University, Dr. Omer Music and Dr. Julian Allwood, without whom the practical experiment results would never happen.

My gratitude goes to my financial sponsors, Chinese Ministry of Education, Henry Lester Trust, and the University of Oxford as well.

I would also like to thank various people that have helped me during my DPhil years. Friends, colleagues, professors, and my family. What is important is not put their names on this page but what they have left with me as a person.

Contents

Chapter 1

Introduction

As this thesis describes a novel manufacturing process, this chapter gives a brief background introduction to the sheet metal forming processes, before presenting a detailed account of the incremental sheet forming process (ISF). Further analysis of essential mechanical properties of this process together with the challenges of the ISF process are described for subsequent chapters. We conclude this chapter with a summary of the contributions of the thesis.

1.1 Sheet Metal Forming Processes

Manufacturing has long been an important sector of the economy and even following the modern information era, it maintains its position. In the UK, manufacturing contributes to 11% of GDP as reported in 2010 [1]. Among the materials used in manufacturing processes, a large proportion of steel and aluminium is used as shown in Table 1.1. Metal products range from car to aircraft, from heavy machinery to window frames, from beverage cans to needles (see Figure 1.1), are almost everywhere in human society. The processing of the metal has to start from the very form of ore, then go through first the furnace to become metal liquid, and a further steelmaking process to remove any impurities and add in any alloying elements. Subsequent casting processes then shape the metal into the bulk form, after which a rough product is created.

Material	Unit(MTons)
Iron(steel)	768
Aluminium	18
Copper	11
Zinc	7
Lead	5
Nickel	0.7
Polymers	85

Table 1.1 Material used worldwide, data from [2]

Once a piece of unformed metal is produced, depending on its dimension, we may adopt either bulk-metal forming or sheet-metal forming process to obtain a finer geometrical shape. Bulk metal forming processes are usually one of the following: forging; rolling; extrusion and drawing as shown in Figure 1.2 (more detail can be found in [3]). In this research, we focus on the sheet-metal forming, where the metal has a much smaller dimension in one direction. The incremental sheet forming process we are interested actually has roots in some common sheet-metal forming processes, a selection of which are described in the following sections.

Figure 1.1 Various metal products

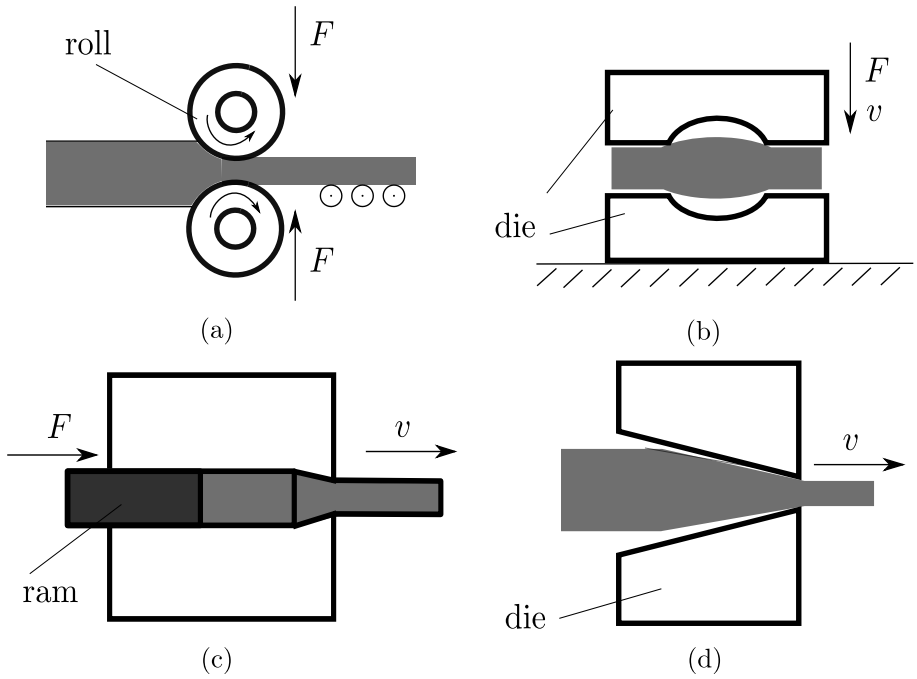


Figure 1.2 Four basic bulk-metal forming processes: (a) rolling, (b) forging, (c) extrusion, (d) drawing.

1.1.1 Bending

Bending is one of the most important forming operations in industry, as it forms the basis of some major products: automobile bodies, appliance etc. It is characterised by the elongation of the outer surface and compression of the inner surface by pre-designed dies. Figure 1.3 (a) depicts one of the possible operations, in which the work part is forced into a V-shape by a punch, illustrating that both stretching and compressive forces are present. One prominent phenomenon is the *springback* effect following the removal of the punch, as shown in Figure 1.3 (b), where the angle of the shape becomes larger than that is when the punch is in contact. In practice to compensate springback, one of the following methods are applied: first, an "overbend" is applied to the sheet, where the amount of the overbend is determined by trial and error; second we may coin the bending area with a high compressive force exerted on the tip of the punch when in contact with the metal surface, which is known as *bottoming*.

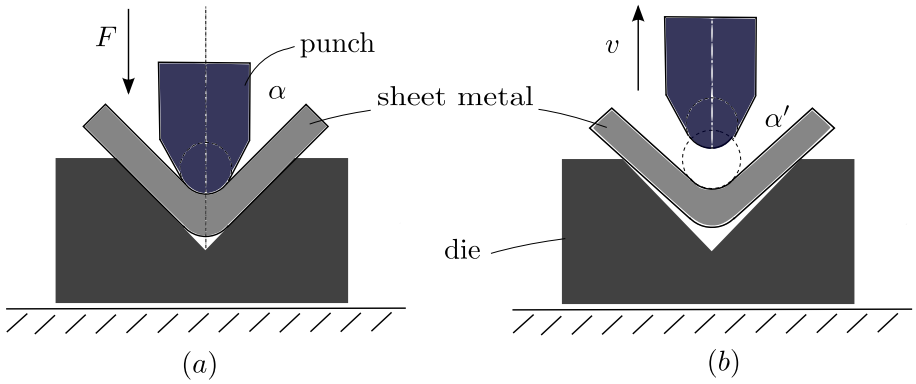
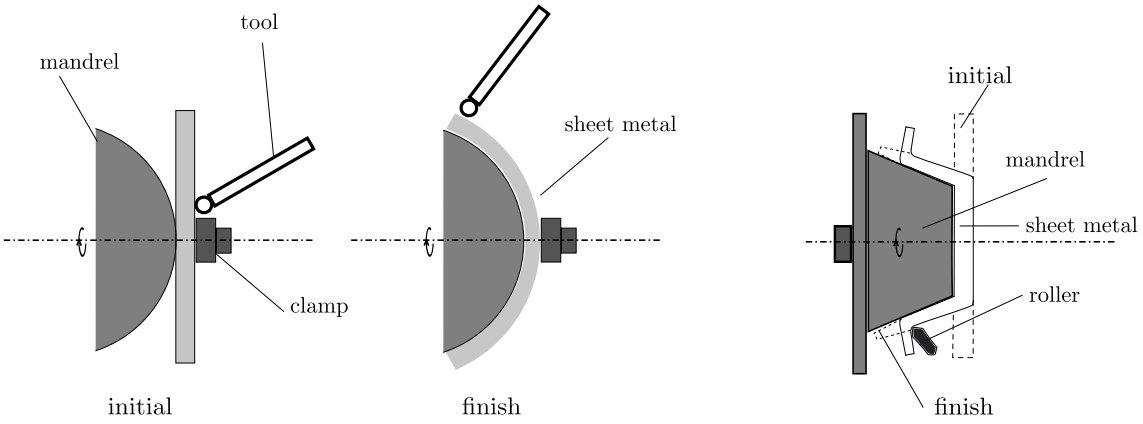


Figure 1.3 Sketch of a typical bending process: (a) the punch in action (b) upon removal of the punch, springback occurs

1.1.2 Spinning

In the spinning process, a disc or tube of metal is rotated at high speed and formed into an axially symmetric part by tools, which, in many cases, allows the production of symmetric products without requiring expensive dies. In Figure 1.4, we illustrate both the conventional (manual) spinning and shear spinning. In the former, a blank of sheet metal is held against a rotating mandrel (shown in Figure 1.4) and deformed by a tool that is moved manually over the mandrel. The thickness of the metal remains more or less unchanged during the processing. This process is particularly suitable for conical and curvilinear shapes. Shear spinning on the other hand can achieve the same goal in a relatively shorter period of time by using an automated controlled roller. Since shearing is the dominant force involved in this process, the thickness of the sheet metal becomes thinner towards the edge compared to the centre (the assumed starting point). In addition, a cooling system may be necessary to remove the heat generated due to friction. The connection between spinning and incremental sheet forming process will become apparent in the thesis.



(a) conventional spinning (b) shear spinning

Figure 1.4 Two spinning processes illustration: a) conventional spinning, manual, multi-pass; b) shear spinning, computer aided, one pass.

1.1.3 Deep Drawing

Deep drawing is one of the stamping processes, which is performed by pushing sheet metal into a die cavity with a punch, shown in Figure 1.5. The sheet metal usually is held against the die by the blankholder and it undergoes significant amount of drawing by the mechanical action of the punch. The mechanics of this process are much more complex than bending or spinning.

As the punch proceeds, the metal around the corners of the die first experience bending action (see Figure 1.5 (ii)), and it then becomes straight further down along the wall (Figure 1.5 (iii)). Together with the compressive force acting on the blankholder and a shorter perimeter of the metal piece, in the later stage of this process, the flange thickens (Figure 1.5 (iii)) before the process finishes (Figure 1.5 (iv)). Usually, the metal flows unevenly from the peripheral region towards the drawn area. For example, for a square flat sheet, more metal flow occurs around the side edges than that of the corners. As a result, to produce parts with better smoothness, in practice quite often an initial trial-and-error optimisation of metal sheet shape is required. A more detailed account of the metal flow in this process can be found in [4]. Commercial applications of this forming method often involve complex geometries with straight sides and radii, such as automotive bodies, aircraft components, utensils, etc.

1.1.4 Dies and Stamping

The metalworking process that uses punches and dies is quite often referred as *stamping process*, which could be one of the previously described processes such as bending, deep drawing or a combination of those processes. The sheet metalworking operations are performed on a machine called *stamping press* (to distinguish from the forging and extrusion press used in bulk-metal forming processes), where the tooling is a punch-and-die

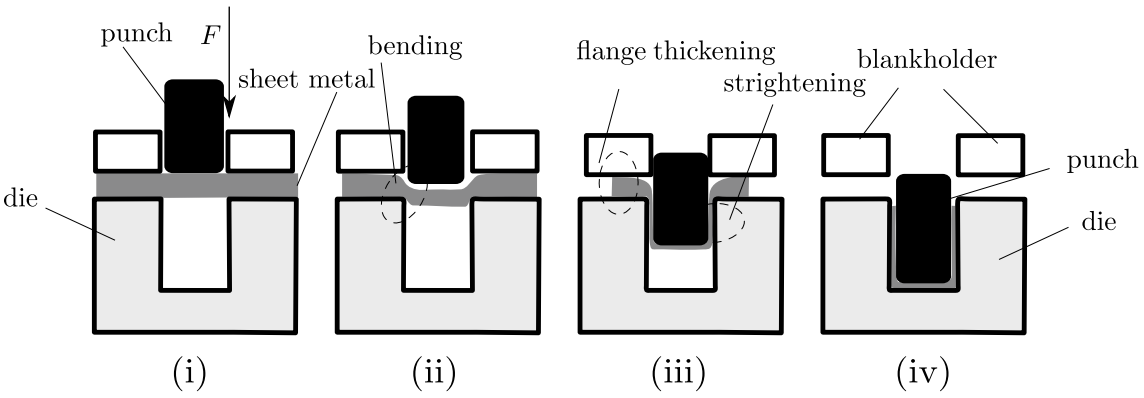


Figure 1.5 Deep drawing process illustration: (i) initial configuration, (ii) bending action in the corners, (iii) flange thickening and straightening in the wall, (iv) final configuration.

assembly. Apart from the difference in geometries of the dies, due to the nature of the different stamping processes, dies can be different in terms of how they function in a single press. The *simple die* performs a single operation with each stroke of the press. However *compound*, *combination* and *progressive* dies can be more complex and may involve more than one work station, see [4] for a detailed account. Although it is not our interest to investigate every possible configurations of the dies and punches, we do need to point out the cost of designing a die. In [3] takes an example of making a cup-shaped part to illustrate the economical considerations behind the choice of a proper metal forming process. As this part can be made from either deep drawing, or conventional spinning. The difference of die costs between these two processes are significant and a die cost-per-part in drawing will be high for a few parts but the time needed is much shorter and the breakeven number suggested in [3] would be 700.

1.2 Incremental Sheet Forming Process

Having described some alternative metal forming operations, in this section, we introduce the incremental sheet forming (ISF) process and point out the technical advantages over the conventional forming methods, mainly stamping process, and the challenges faced by ISF before it can be widely adopted by industry.

1.2.1 Process Description

Incremental sheet forming is a progressive metal forming process, where the deformation occurs locally around the point of contact between a tool and the metal sheet. The final work-piece is formed cumulatively by the movements of the tool, which is usually attached to a CNC milling machine. The basic elements of a single point ISF scheme are shown in Figure 1.6. The sheet metal is clamped by two blank holders and the tool moves with a feed velocity and forming force, whose trajectory is determined by a higher level planning unit. The "single" point refers to the sole forming tool employed in the configuration, as

we shall see shortly there are possibilities to use more. Figure 1.7 shows a single point ISF machine used for research at Cambridge University [5].

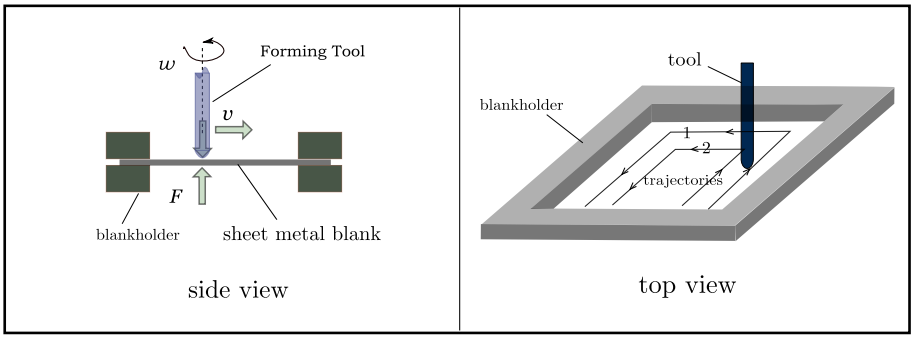


Figure 1.6 Single point ISF process illustration with both side and top views: F is the metal forming force, v is the tool feed and ω is the spindle rpm; the tool motion is usually described and controlled in the Cartesian coordinates. x-y plane is usually labelled as the sheet plane. [6, 7]

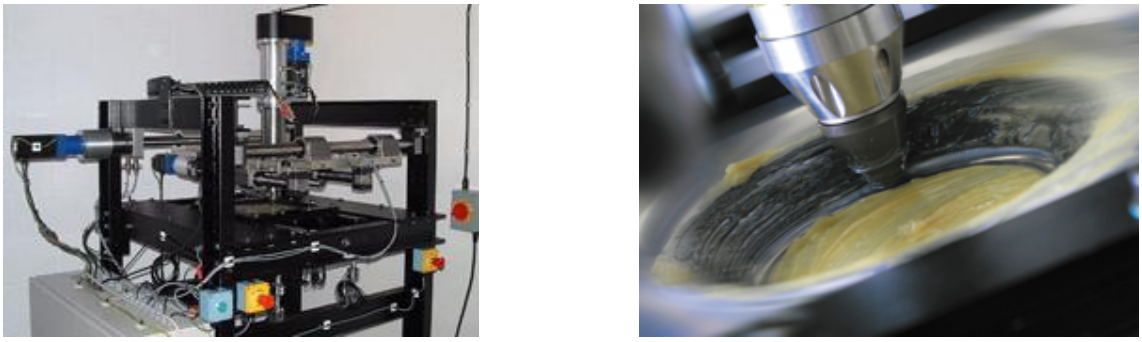


Figure 1.7 Single point ISF machine at Cambridge University: (a) the whole mechanical system, (b) the tool in action

1.2.2 Historical Review

As described in previous sections, most metal forming methods require a set of punch and dies or mandrels, which are both costly and time consuming to make. Also, the additional tooling change-over increases the total lead time in the whole manufacturing process and thus compromises its efficiency. The ISF process, however, is dieless in nature and capable of producing different parts of geometries with a universal tool. The tooling cost of ISF alone can be as low as 5–10% compared to the conventional sheet metal forming processes [8]. The earliest invention that resembles the concepts of ISF is believed to be a patent by Leszak [9] in 1967 as shown in Figure 1.8. In that patent, it is specifically stated the object of the invention is to form the ductile sheet metal into "the desired configurations of conic section shapes of revolution utilizing simple inexpensive tooling. [9]"

There are a number of possible variants of the single point ISF processes, such as two point forming tools (partially aim to speed up the process) and in some cases, supports

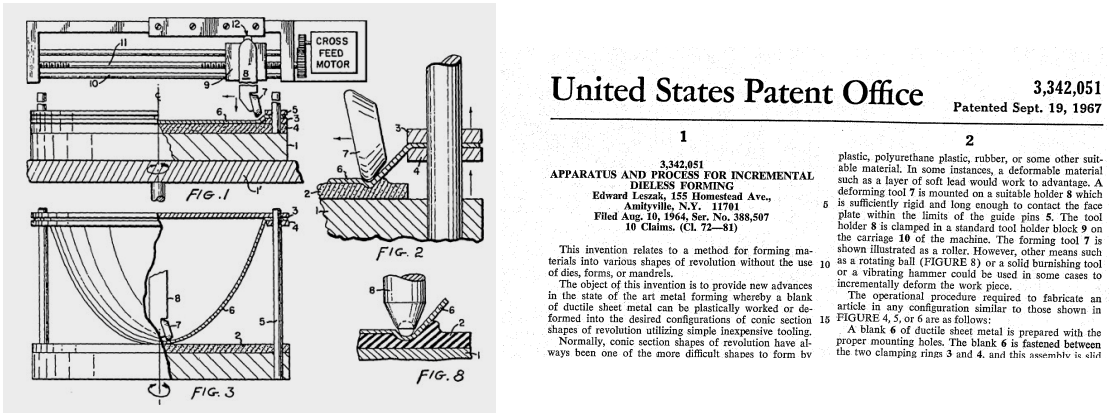


Figure 1.8 Leszak apparatus [9]

or partial dies can be provided to improve the final finish, see Figure 1.9. [7] provides a good account of the ISF development history and its other variants.

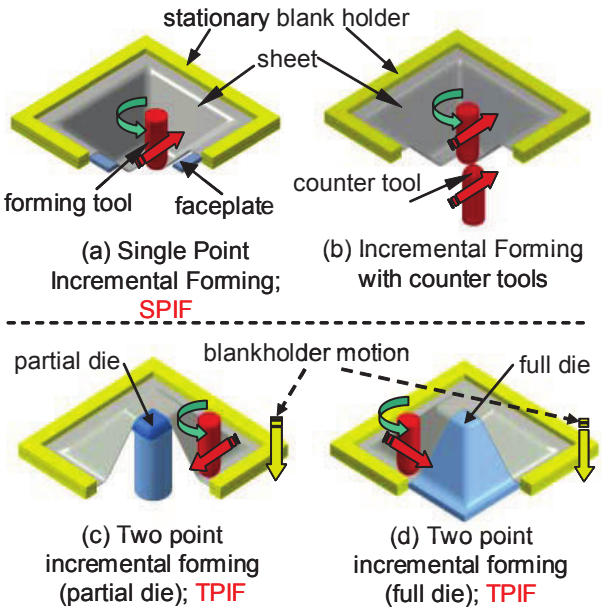


Figure 1.9 variants of ISF processes (from [10])

Like the conventional spinning process, ISF can follow conic paths and produce symmetric parts, but ISF is also sufficiently flexible to make complex asymmetric geometries and the ISF process is regarded as one of the sustainable manufacturing methods [11]. Since it is suitable for customised production, ISF can also be used as part of a flexible manufacturing system [12]. In the case of iterative operations in the design phase or rather the rapid prototyping, when the errors have been found and changes have to be made, ISF can play an important role. From a wider perspective, as documented by [13] the challenge of sustainable material manufacturing in a world in wake of rapid population growth and limited resources is urgent, and ISF does have something to offer in this respect.

On the other hand, we should also note that the deformation time alone in ISF is

longer than that of the conventional operations since it is a progressive process. In fact, the stamping process may take just seconds to complete a single stroke action whilst the ISF may well take a couple of minutes to hours. This is part of the reason that ISF is more suitable for small batch production, while stamping is preferred for mass production. However, the real challenge of ISF process, is that the geometric accuracy of the shapes made through ISF process is relatively poor. In the following sections we will extensively address this issue starting from an introduction to the mechanical behaviour of materials.

1.3 Mechanics of Materials

In this section, we aim to discuss the mechanical behaviours and properties of the materials used in ISF process: the concept of plastic deformation, formability issues, and springback. The general behaviour of the material under various loading conditions is studied in the context of *material mechanics* [14], where the objective is to understand the mechanics of materials by determining the stress, strain and displacement states of the material given the external loading condition.

1.3.1 Concepts of Stress and Strain

Stress and strain are the two basic concepts for mechanical analysis. Stress is a measure of the exerted force acting on a body and is defined as the intensity of force, F, at a point [15]:

$$\sigma = \partial F/\partial A, \text{ as } \partial A \to 0,$$

where σ denotes the stress and A represents the area over which the force acts. Note that we are considering the one-dimensional (uniaxial) case. Strain describes the deformation state compared to its original dimension. Assume a body deforms with initial distance between two points l_0 that has changed to l ($l \neq l_0$), then the *engineering strain* ϵ can be defined as

$$\epsilon = \frac{l - l_0}{l_0} = \frac{\Delta l}{l_0}.$$

Alternatively, the natural or *true strain* defined from strain increment as

$$\mathrm{d}\varepsilon = \mathrm{d}l/l,$$

and upon integration, we can write as

$$\varepsilon = \int_{l_0}^{l} \mathrm{d}\varepsilon = \int_{l_0}^{l} \mathrm{d}l/l = \ln\frac{l}{l_0}.$$

In case of three-dimensional stress and strain, a formal introduction is given in Section 3.1, here we assume already know their components in Cartesian coordinates as

$$\left[\varepsilon\right] = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix}, \quad \left[\sigma\right] = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}, \tag{1.1}$$

where for the indices, the first one indicates the direction of the source is exerting and the second index indicate the direction of the effect of this source (c.f. Section 3.1). A transformation of coordinates of ε, σ can be applied so that only the diagonal components are non-zero, which are called the *principal* stress and strain. The largest and smallest values are correspondingly termed *major* and *minor* stress/strain.

The relation between stress and strain is usually characterised by a *constitutive equation* or *constitutive modelling*, whose exact form depends on the nature of the deformation and the underlying material properties. We shall elaborate on this later. In section 3.1, we will revisit these two concepts through the view of theoretical physics by *tensors* and more details will be provided there.

1.3.2 Material Behaviour under Loading

For simplicity, consider the typical mechanical behaviour of a metal specimen under uniaxial tension. The strain-stress relation can be idealised as a strain-stress curve shown in Figure 1.10. At first, the material behaves linearly as the load is increased before reaching point A. This linear region is called the *elastic region*, where the material will retrieve to the original state O upon removal of the load. The relation between stress and strain is well known as *Hooke's law*,

$$\sigma = E\varepsilon, \tag{1.2}$$

where E is the *Young's modulus*, whose value can be simply determined from the slope of linear portion of the stress-strain curve, OA. Further along the stress-strain curve,

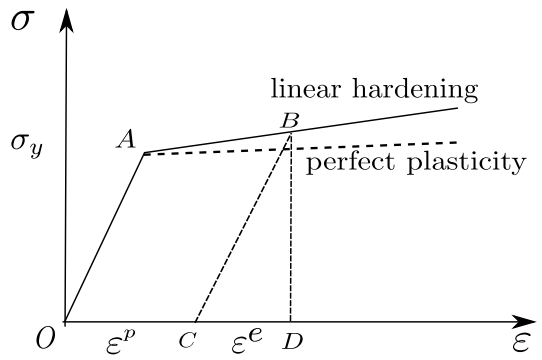


Figure 1.10 Idealised stress strain curve of the elasto-plastic material behaviour in uniaxis case. OA elastic deformation, after A plastic deformation in which the solid line represents linear hardening and dashed line represents perfect plasticity.

permanent deformation occurs after the point A . If at point B the load is removed, a

plastic strain ε^p (OC in the diagram) will remain in the specimen, whilst the counterpart *elastic strain* ε^e (CD in the diagram) is fully recovered. The total strain ε at state B is the sum of the two:

$$\varepsilon = \varepsilon^p + \varepsilon^e. \tag{1.3}$$

This is the classic additive decomposition of strain [16] and the strain achieved at this state is given by

$$\sigma = E\varepsilon^e = E(\varepsilon - \varepsilon^p). \tag{1.4}$$

The stress at point A, σ_y, is called the *yield stress* as the plastic deformation of the material begins. In reality, the deformation in the plastic region increases in time at constant stress. The material may be regarded as *strengthened* or *hardened* by the plastic deformation. The increase in stress during the plastic deformation is termed *work-hardening* or *strain-hardening*. To simplify understanding of the plasticity behaviour, sometimes a *perfect plasticity* model where the hardening effect is neglected is used, which is shown as the dashed line in Figure. 1.10. As the force is further applied, the stress can reach the largest value, which is called the *ultimate stress*. In a force-controlled test, the specimen will break at this point, whilst the specimen will experience a period of stress decrease after this point before it reaches *fracture* stress in a displacement-controlled test.

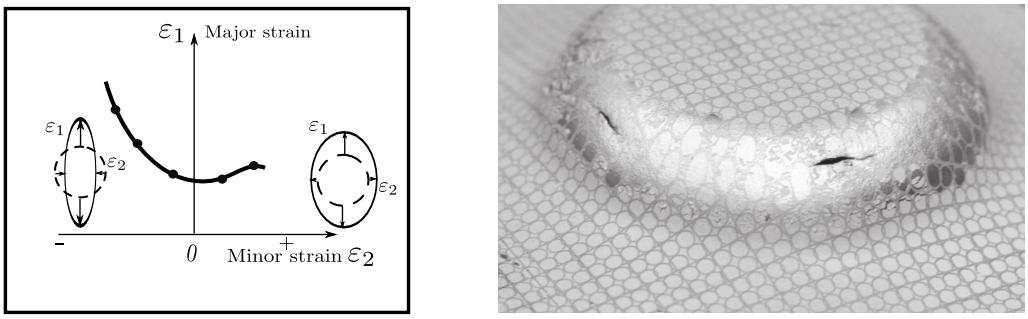
In the context of deformation, *elastic theory* studies the behaviour of the specimen as it undergoes only elastic deformation while *plastic theory* studies the behaviour of the plastic deformation. *Elasto-plastic theory* deals with the case when both elastic and plastic deformations are involved. In a metal forming process, it is the plastic deformation that allows the final shape to be made. The *springback* phenomenon, which accounts the retreat of the deformation state upon removal of the external force/tooling, and causes the geometrical defects, is the direct effect of elastic deformation.

1.4 Issues Concerning the ISF Process

1.4.1 Formability

As illustrated in previous section, excessive force would cause fracture, and in the metal forming process, this is known as the *formability* issue, which refers to the ability of sheet-metal material to be made into desired shape without breakdown or "losing its integrity", i.e. excessive thinning and fracture [17]. The formability of metal forming is investigated through strain analysis and usually a Forming Limit Diagram (FLD) is employed. The FLD provides a basic measurement for formability criteria in which the major and minor strain is recorded at the occurrence of material failure. Generally, a series of tests is needed to indicate different major-and-minor strain ratios. An example of FLD is drawn in Figure 1.11 (a), where the minor and major strain are measured relative to their original values and plane strain is assumed. Experimentally, a popular method is the circle grids making method, in which the sheet is marked with circle grids before deforming and the forming process continues until the sheet fracture occurs and the strain distribution can

be seen from the circle elongation. Figure 1.11 (b) demonstrates one of these results.



(a) (b)

Figure 1.11 (a) sketch of a common metal sheet forming limit digram (FLD) (modified from [18]), (b) incremental sheet metal formability test by circle grids marking method, from[19]

Usually, manufacturing with the ISF process starts from desired shapes in the form of CAD data, the formability study will allow the high level planning unit to decide whether the shape can be made or not.

Studies in [20, 21] have identified that plane strain happens close to the corners and intersections between flat surface, and the formability of the ISF process can be described with the index *maximum wall angle*. A table of material with initial thickness and maximum draw angle is listed in [6]. The investigations of the ISF formability issue lead to the general conclusion that the achievable strain is far larger than that has been shown by the normal metal forming limit curves, although the reason for this has been attributed very differently. [22] provides an overview, but [15] believes it is the high normal force on the tool and that each element undergoes bending and reverse bending so the strain path is not monotonic; [23] attributes the effect to the dominance of plane-strain. In [21], five sheet metal mechanical properties were studied, namely strength coefficient, strain hardening coefficient, normal anisotropy index, ultimate tensile strength and percentage elongation. The study claims that the material formability in incremental forming depends substantially on strain hardening coefficient and the percentage elongation.

1.4.2 Geometrical Accuracy

On the laboratory scale, the accuracy of the parts created by ISF is between ± 1.5 mm and ± 3 mm [6]. However, in order for ISF to be competitive with a stamping process, an accuracy of below ± 1.0 mm and more realistically below ± 0.2 mm would be needed. Geometrical accuracy of the shape made by the ISF process has been studied in a number of papers, such as [24].

The major factor with ISF accuracy, as with most other forming processes, is the springback issue. Springback, as explained earlier, is the effect that the formed sheet tends to return partially to its original shape upon the removal of the tool from the contact point of the sheet, i.e. unloading. The springback problem becomes more severe

with the use of higher strength and lower density (i.e. lower modulus) materials. There is extensively both analytical [25] and experimental work [26] in this area. As it is essential to have some ability to predict the springback given an initial product design, we turn to analytic methods.

Unfortunately, the analytical model in the field of applied mechanics, such as [27], is confined to very simple geometries, and the forcing actions considered are limited. For example, in [27] just pure bending and stretching are considered independently. However, the finite element analysis has been used in [28, 29, 30], and in [30] a truncated cone in single point ISF with finite element simulation is studied. Quite often the term "finite element modelling/analysis" in engineering literature has lost its accurate mathematical meaning. As will be explained later, the finite element method is in fact a numerical scheme to solve the partial differential equations, but the prevalence of this method is associated with the application of the commercial finite element software such as Abaquas, and the work done in this respect is less to do with the underlying mathematical/analytical model, but more to do with the selection of suitable elements and the improvements made towards a better, sometimes rather sophisticated integration scheme, such as [31]. Although, we can draw a large amount of information from the simulation result, in this thesis we will carefully study the mathematical model behind the method.

The second issue for geometrical accuracy is the wall thinning. As the metal is deformed to towards the final shape, the thickness of the material is reduced and the sheet thickness could vary at different locations at the final state. The wall thinning issue matters in the sense that the variations occur at different locations could exceed the required ± 0.2 mm geometric accuracy tolerance.

Surface smoothness is another issue that is often required for final products, as generally a smooth surface is favourable, such as a car body panel. In ISF processes, the final sheet surface usually has unwanted ridges, as illustrated in Figure 1.12. In one case, the tool incremental vertical step depth, denoted by Δz in Figure 1.12, has an effect on surface roughness [6]. It is reported that with a step depth of 0.13 mm, which is 1% of the tool radius, the ridge virtually disappears . Factors such as the spindle rotating speed, tool radius and wall angle are also claimed to have effects on the final surface roughness [6].

One possible method to improve the geometrical accuracy of a shape made by ISF is to optimise the tool trajectory. The tool trajectory, or path, directly decides how the tool moves and has a direct effect on final geometric accuracy. The simplest toolpath type is a *contour toolpath* in which the desired shape is first scanned or computed into a series of contours in sheet plane, denoted by $x - y$ plane in Cartesian coordinates, and then the tool moves along the outermost contour on the sheet, followed by a vertical step depth Δz movement in z direction. See Figure 1.13 for reference. An example of the contours of an irregular shape is shown in Figure 1.14, and this type of contours can be easily computed from a CAM software. Other types of specific tool trajectories are also being studied [32]. The *spiral* toolpath proposed in [20] can eliminate the ridges on the product surface and

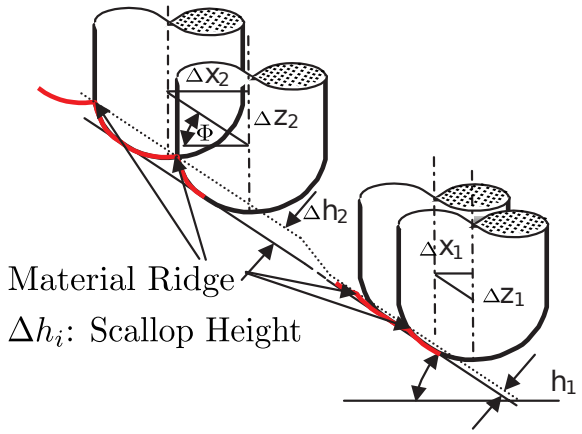


Figure 1.12 Illustration of how ridges formed [6]

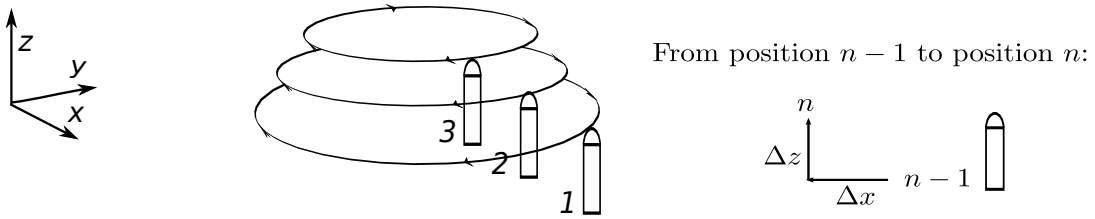


Figure 1.13 Contour Toolpath Illustration. Δx denotes the movement from one outer contour to an inner one. Δz denotes the vertical step depth.

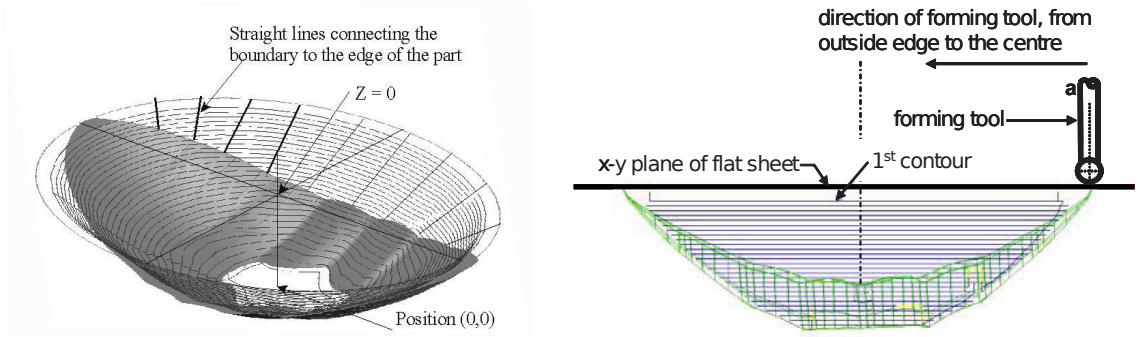


Figure 1.14 An example of contours of an irregular surface shape [6]

thus produce a better surface roughness. A *constant scallop height* trajectory, shown in Figure 1.15, can enhance the final sheet surface quality. Note that the scallop height is shown in Figure 1.12. To maintain a constant scallop, the vertical step Δz is constantly changing and [33] claims that a variable step depth is important to geometric accuracy. To this end, an *extra correction algorithm* could be developed. Under this framework,

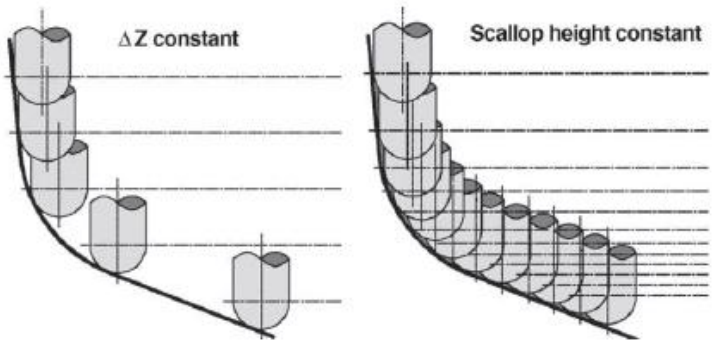


Figure 1.15 Constant scallop height toolpath [33], where the scallop height is explained in Figure 1.12

the regular contour toolpath is used to create a part and the formed part is measured. A deviation vector can be obtained between the target shape and the actual measured shape, which is then used to correct the original tool path. This process can be applied for several times until reaching an acceptable tolerance. An extension may be a *multipass* or *multi-stage forming*, where such intermediate shapes are designed beforehand. As an example, in [34] a method based on a simplified strain model is developed. The intermediate shape is chosen so that the final shape strain is more uniform, based on an analysis of the desired shape stains and the predicted strain computed from the simplified strain model.

In addition to improving tool trajectories, other possible routes towards a better geometrical accuracy could be: using supporting dies, cut-out technique at the end of the production [35], with multiple points acting together and etc. The work in this thesis focuses only on the single point incremental sheet forming process, without relying on other mechanical improvements, we aim to push the boundary of the control and enlarge our knowledge about this fundamentally difficult process.

1.5 Structure and Contributions of the Thesis

Our lines of research lay at the control of the process, with a focus on optimising the tool trajectory to achieve minimal deviations between desired shape and the final outcome. We start from Chapter 2 with a closer look at two control methods, namely optimal control and the model predictive control, then we use a simplified model of the deformation process in ISF to design a constrained model predictive controller. The implementation of the controller and the experimental results are presented at the end of this Chapter. From Chapter 3, we develop a physics based ISF process model and its corresponding

computational method to obtain a solution. In Chapter 3, we develop the mathematical ISF model using elasto-plasticity constitutive relation and contact mechanics. In Chapter 4, using methods in finite element and optimisation in function space, we develop a computational method for the mathematical model proposed in Chapter 3. In Chapter 5, we document the implementation details and produce the numerical results. Finally, we review and conclude our work in Chapter 6. The main body of the thesis can be regarded as two parts:

1. the control methodology with a simple linear model (Chapter 2);
2. modelling for ISF with material law and contact mechanics, and the corresponding computational method(Chapter 3 - Chapter 5).

The structure of the thesis also reflects the contributions of the thesis: the control methodology on the ISF process; the physics based modelling by casting ISF process as a contact problem; the computational method for the ISF model. For more summarised findings of this thesis, refer to the Chapter 6. Part of this work has been submitted as conference papers:

- Wang, H. and Duncan, S.R. Model predictive based incremental sheet forming toolpath optimisation. *Proc. UKACC International Conference on Control, Coventry, UK, pp. 1172-1177* . 2010.
- Wang, H. and Duncan, S.R. Constrained model predictive control of an incremental sheet forming process. *IEEE Multi-Conference on Systems and Control, Denver, USA, pp. 1288-1293.* 2011.
- Wang, H. and Duncan, S.R. Optimization of tool trajectory for incremental sheet forming using closed loop control. *IEEE International Conference on Automation Science and Engineering, Trieste, Italy, pp. 779 - 784* . 2011.

"Nothing is built on stone; all is built in sand. But we must build as if the sand were stone."

Jorge Luis Borges

Chapter 2

Constrained Model Predictive Control of ISF

In this chapter, we begin to study the deformation of sheet metal material in the ISF process using a simplified phenomenal model. We employ a predictive controller to obtain an optimised tool trajectory in the sense of minimising the geometrical deviations between the targeted shape and the shape made by the ISF process. We first present some background of the model predictive control (MPC) and the closed related optimal control problem, before addressing modelling the ISF process. The implementation of the controller and the experimental results are treated in the remainder of the chapter.

2.1 Background on Control Theory

In control theory, we are interested in the *states* of the control system, which consists of a number of "interconnected components that provide a desired response"[36], and devise methods to manipulate these states with *controls*. For example, the evolution of a finite-dimensional dynamical system may be modelled by the following ordinary differential equation (ODE):

$$\dot{x}(t) = f\left(t, x(t), u(t)\right), \quad x(t_0) = x_0, \tag{2.1}$$

where vector function $x(t)$ denotes the system states at time instant t, $x(t_0)$ denotes the initial system condition, and $u(t)$ denotes the control inputs. The control problem is then to seek a control law that achieves certain system objectives, such as tracking some reference signals, satisfying required terminal states, obtaining desired transient dynamical performances and etc. We start by considering two possible control strategies: optimal control and model predictive control.

2.1.1 Optimal Control

The *optimal control* has deep connection with *the calculus of variations* (see historical remarks in [37]), where the problem of minimising an integral over all curves are studied,

such as the classical *isoperimetrical problem* in which we are to find greatest possible area formed by a curve of given length , and the most well known one perhaps is Bernoulli's *brachystochrone problem*, also known as fastest descent problem (See Figure 2.1), both of which can be solved using optimal control theory.

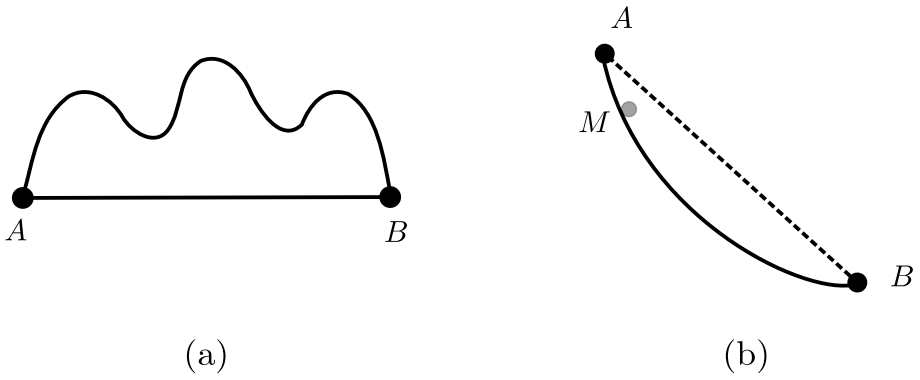


Figure 2.1 (a) The classical isoparametric problem (b) The brachystochrone problem illustration

The optimal control problem in general consists three parts: first, a *performance index* or *objective function*, typically, it may be written as

$$J = \phi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t), t)dt, \tag{2.2}$$

where ϕ represents the terminal state costs, and L is responsible for other system performance. Second, a state equation describing the behaviour of the system, typically in the form of the ODE equation (2.1). Third, various possible constraints, for example the terminal states are constrained to some prescribed values or as a function:

$$\psi\left(x(t_f), t_f\right) = 0; \tag{2.3}$$

or the control action $u(t)$ can be also limited within certain set as $u(t) \in U$. The control objective is then to find a control law that minimise the performance index subject to both the state equation and the constraints. The existence or the sufficient conditions for an optimal control are rather cumbersome (refer to [38]) and quite often not of very practical use. Instead, we focus on the *necessary conditions* for the optimal solution here, which is known as the *Pontryagin maximum principle*[1]. It is the discovery of this principle that underpins optimal control. A detailed historical account of the development of optimal control can be found in [39].

To express the necessary condition for the optimal control problem (2.2), we need to introduce a function called *Hamiltonian* as

$$H(x, u, t, \lambda) = L(x, u, t) + \sum_{i=1}^{n} \lambda_i f_i(x, u, t) = L(x, u, t) + \lambda(t) \cdot f(x, u, t), \tag{2.4}$$

[1]In our work, we reformulate the maximum principle as minimum principle.

where $\lambda(t)$ can be viewed either as a Lagrange multiplier or the *co-state*. Differentiate hamiltonian function H with respect to λ, we can retrieve the state equation (2.1), which is

$$H_\lambda = \frac{\partial H}{\partial \lambda} = f = \dot{x}(t). \tag{2.5}$$

To simplify the discussion, we assume that there are no terminal and control variable constraints (in the presence of constraints, additional Lagrange multiplier will be needed in the Hamiltonian) and suppose $u^*(t)$ is the optimal control and $x^*(t)$ is the corresponding state, then first we have co-state equation

$$-\dot{\lambda}^T(t) = H_x(x^*(t), u^*(t), \lambda(t), t) = \frac{\partial H}{\partial x}, \tag{2.6}$$

and the terminal condition

$$\lambda^T(t_f) = \phi_x(x(t_f)). \tag{2.7}$$

Pontryagin minimum principle[2] requires

$$H(x^*(t), u^*(t), \lambda(t), t) \leq H(x^*(t), u(t), \lambda(t), t), \forall t \text{ in } [t_0, t_f]. \tag{2.8}$$

Quite often a weaker condition to (2.8) is used in practice [40] as

$$H_u(x^*(t), u^*(t), \lambda(t), t) = 0, \text{ or } \frac{\partial H}{\partial u} = 0, \tag{2.9}$$

In practice, following the above necessary conditions to find the optimal control is limited to simple cases as it is difficult to find the costate that statisfies both the initial condition and the terminal condition, so more often the optimal control problem is tackled by the *dynamic programming* method[3], whose core is the principle that regardless of the initial decisions, states, the remaining decision must remain optimal with the states resulting from the initial decisions. We define value function $V(x, t)$ as

$$V(x, t) = \min_u \left\{ \phi(x(t_f), t_f) + \int_t^{t_f} L(x, u, \tau)\, \mathrm{d}\tau \right\},$$

and the dynamic programming principle yields Hamilton-Jacobian-Bellman equation

$$-\frac{\partial V}{\partial t} = H^*(x, \frac{\partial V}{\partial x}, t), \tag{2.10}$$

where

$$H^*(x, \frac{\partial V}{\partial x}, t) = \min_u \left\{ L(x, u, t) + \frac{\partial V}{\partial x} f(x, u, t) \right\}.$$

More details can be found in [41]. The above is a sketch of the dynamic programming method for continuous time system, whilst discrete system problems can be found in [42].

[2]We have reformulated the Pontryagin maximum principle as a minimum one

[3]Dynamic programming was originally conceived by Richard Bellman as an effective computational method for dealing with optimal decision making in discrete-time processes.

To curcumvent the computational difficulties brought with high dimension, a method of approximated dynamic programming is introduced in [43].

2.1.2 Model Predictive Control

The earliest control concept that is close to what is known as model predictive control (MPC) nowadays most likely to have developed in the engineering practice, and the earliest paper in this regard is believed to be [44], where the term of *model predictive heuristic control* is used, other ideas like *dynamic matrix control* [45], *general predictive control* [46] have all contributed to the MPC development. The early success in the industry practice promoted the theoretical study of the MPC and the subject advanced from the results in the receding horizon control as the early concern with the stability issue was finally resolved [47]. The historical development can be found in [48] and applications in process industries are well documented in [49]. [50] provides a good survey on the stability and optimality issue.

An illustration of the components of MPC is shown in Figure 2.2. At the heart of the

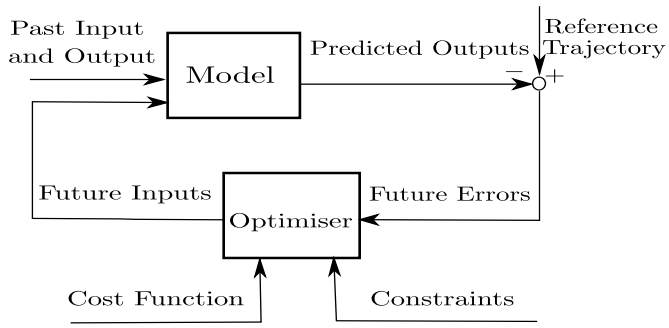


Figure 2.2 Illustration of the MPC structure

MPC is an inner process model, which takes account of not only past inputs and outputs, but also the future inputs. Indeed the MPC breaks away from the conventional PID control by using an explicit model. The other important part of the MPC is the optimiser, where an optimal control problem is solved except that the optimisation horizon of the problem usually is finite. The popularity of the MPC is its ability to handle constraints in a systematic manner, in which constraints of either system states or control actions can be elegantly included in the optimal control. The prediction horizon of the MPC is illustrated in Figure 2.3 as a discretised time interval N, which is moving as time elapses. Model predictive control allows the current control action to be determined not only by the usual current states but also on future predicted states, subject to various possible system constraints.

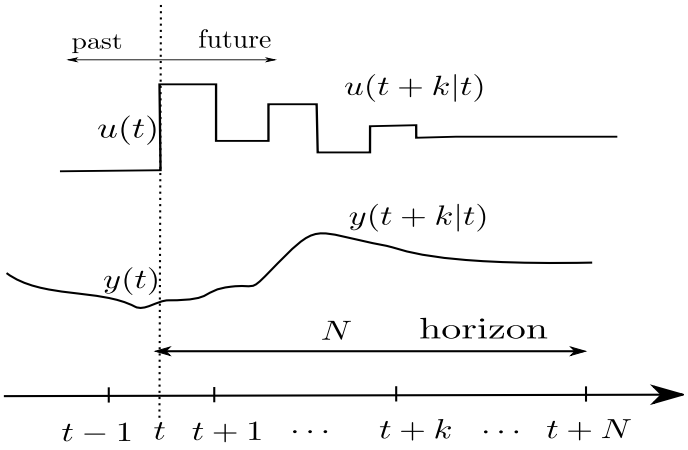


Figure 2.3 MPC approach with moving horizon.

2.2 ISF Modelling from the Perspective of Optimal Control

In this section, we take a closer look into the ISF process and propose a linearised deformation model that will form the basis of our MPC controller.

Consider making an axisymmetric part, whose final geometry can be captured by a single profile as illustrated in Figure 2.4. The figure also highlights deviations from the targeted shape: the flank around the blankholder becomes bent and the minor base part bends inward, the resulting surface resembles the shape of a pillow and is termed as "pillow effect" in [24], besides, the wall in-between tends to contract towards the centre. As we have mentioned before, this deviation is caused by elastic recovery when the tool moves away from the contact region. A further deviation will occur if the clamped blank holders are removed.

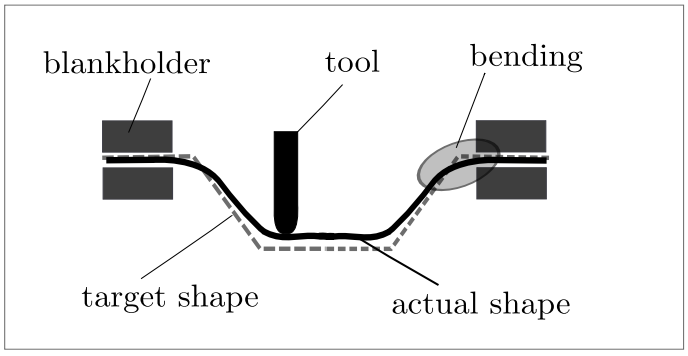


Figure 2.4 Final deformation in ISF with target and actual shape outlined

Since the ISF is a progressive process, the intermediate geometrical shape due to deformation depends on the type of trajectory applied. Here, we consider the the standard contour tool trajectory, referred as *contour following* method. As an example, a number of contours of a conic shape is illustrated in Figure 2.5, where the shape of contours are circles horizontally and placed vertically with a constant distance between the consecutive two. We treat each horizontal circle as a "layer", which is determined by the tool movement Δr,

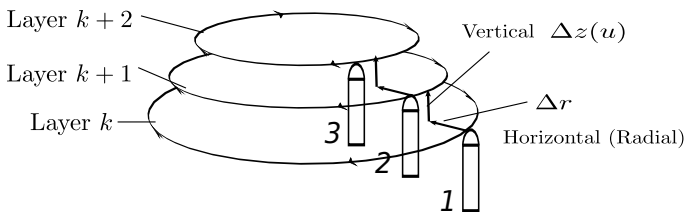


Figure 2.5 Contour toolpath of a cone: consecutive layers. Incremental steps in both radial and vertical directions (Δr and Δz respectively) are constants in every consecutive layers.

whilst the vertical tool movements Δz will determine the distance between each layer. This breakdown is the start point for modelling the ISF process, which amounts to discretise a continuous tool trajectory into some separated segments. Therefore, the tool movement in the current configuration starts from the outer side of the sheet and moves in a circle with initial radius r_0, next step inside Δr, and step up Δz and moves in another circle with radius $r_0 - \Delta r$, which is shown in Figure 2.5 (tool position numbered with $1, 2, 3$). In the open loop contour following approach, both the radial incremental inputs Δr and the vertical incremental inputs Δz stay constant for different layers.

To work with both Δz and Δr for a model that serves the purpose of building a controller is not easy. Partially for the sake of simplicity, we fix the horizontal incremental input Δr as a known variable and work with the vertical incremental step Δz as our controller input, denoted by u. If each layer is regarded as one state of the ISF process (system), then we can regard the ISF process, which in essence deforms a flat sheet to the desired shape, as a system evolving from the initial state to a desired state. In practice, for reasons associated with the physical limitations of the material deformation, the target state cannot be reached, so we require the system to be driven to a region that is close to the final state as possible. Therefore, this progressive process can be first cast as an finite horizon optimal control problem [51]:

$$\begin{aligned} &\min_{u_k:k=0,\ldots,N-1} \|x_N - \hat{x}_N\|_2 \,, \\ &\text{subject to } x_{k+1} = f(x_k, u_k), \quad u_k \in \mathbb{R}^1. \end{aligned} \tag{2.11}$$

where N, is the number of the layers (or steps), $x_k \in \mathbb{R}^m$ is the state of the sheet which is the geometrical deformation, i.e. the shape profile at layer (step) k in our case, m indicates the number of points used to discretise a continuous profile, $\hat{x}_N$ denotes the desired final state, i.e. the target shape profile, u_k denotes the system vertical inputs, which shall construct the tool trajectory together with predetermined Δr and f denotes a deformation function that models the physical system.

The objective function employs a 2-norm form to ensure an efficient solution, although other measures could be used. For example, the ∞-norm will limit large deviations, whilst 2-norms tend to keep average deviations small but allow large peak deviation [52].

The problem with the optimal control (2.11) is that the system equation $x_{k+1} = f(x_k, u_k)$ is difficult to obtain. To curcumvent this difficulty and construct an approxim-

ated sheet deformation model, we use the information obtained in the open loop approach, i.e. apply the standard contour following trajectory, and take measurements about the deformation after each layer has been completed, so that the function f can be linearised around the set of known states and inputs $(\bar{x}_k, \bar{u}_k)$ by a first order approximation as following:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) \\ &= f(\bar{x}_k, \bar{u}_k) + \left.\frac{\partial f}{\partial x_k}\right|_{(\bar{x}_k, \bar{u}_k)} (x_k - \bar{x}_k) + \left.\frac{\partial f}{\partial u_k}\right|_{(\bar{x}_k, \bar{u}_k)} (u_k - \bar{u}_k) \\ &\quad + \text{Higher Order Terms}, \end{aligned} \tag{2.12}$$

where the given inputs $\bar{u}_k$ are the inputs for running a contour following trajectory and the states $\bar{x}_k$ are the recorded corresponding sheet deformations. Given the relation between the deformation states of two consecutive layers

$$\bar{x}_{k+1} = f(\bar{x}_k, \bar{u}_k), \tag{2.13}$$

then by rearranging and neglecting high order terms of (2.12), we have the approximation

$$x_{k+1} - \bar{x}_{k+1} = \left.\frac{\partial f}{\partial x_k}\right|_{(\bar{x}_k, \bar{u}_k)} (x_k - \bar{x}_k) + \left.\frac{\partial f}{\partial u_k}\right|_{(\bar{x}_k, \bar{u}_k)} (u_k - \bar{u}_k). \tag{2.14}$$

Using $\Delta x_k = x_{k+1} - \bar{x}_{k+1}$ to represent the difference, yields

$$\Delta x_{k+1} = A_k \Delta x_k + B_k \Delta u_k \tag{2.15}$$

where

$$A_k = \left.\frac{\partial f}{\partial x_k}\right|_{(\bar{x}_k, \bar{u}_k)}, \quad B_k = \left.\frac{\partial f}{\partial u_k}\right|_{(\bar{x}_k, \bar{u}_k)}. \tag{2.16}$$

As previously mentioned, the state x_k used here is the shape profile which can be represented as a vector and the u_k the vertical input is a scalar. Since $x_k \in \mathbb{R}^m$, then the function f is the map $(\mathbb{R}^m, \mathbb{R}^1) \to \mathbb{R}^m$, and the first order partial derivative $A_k \in \mathbb{R}^{m \times m}$ is a Jacobian matrix and correspondingly $B_k \in \mathbb{R}^{m \times 1}$ is the gradient vector. Given that the Jacobian matrix is not available, we apply a further approximation by assuming the deformation function is additive with the form

$$f(x_k, u_k) = x_k + h_k(u_k), \tag{2.17}$$

so that the Jacobian matrix becomes the identity matrix,

$$A_k = I. \tag{2.18}$$

and (2.16) can be transformed into

$$\Delta x_{k+1} = \Delta x_k + B_k \Delta u_k. \tag{2.19}$$

In the general open loop method, we set up the tool trajectory, run the process and the geometrical deformation information at each layer is stored as $\bar{x}_k$. The gradient B_k is then simply approximated by

$$B_k = \frac{\bar{x}_{k+1} - \bar{x}_k}{\bar{u}_k}, \tag{2.20}$$

so that

$$\bar{x}_{k+1} = \bar{x}_k + B_k \bar{u}_k, \tag{2.21}$$

and combining this with (2.19), we can retrieve the equation directly related to u_k, which is

$$x_{k+1} = x_k + B_k u_k. \tag{2.22}$$

Assuming that the initial condition is

$$x_0 = 0, \tag{2.23}$$

then after N steps we have

$$\boldsymbol{x}_N = \mathbf{B}\mathbf{u}, \tag{2.24}$$

where

$$\mathbf{B} = \begin{bmatrix} B_0 & B_1 & \cdots & B_{N-1} \end{bmatrix} \tag{2.25}$$

and

$$\mathbf{u} = \begin{bmatrix} u_0 & u_1 & \cdots & u_{N-1} \end{bmatrix}^T \tag{2.26}$$

To ensure the validity of the linearisation around the contour path, the input $\mathbf{u}$ is constrained to lie in a region that is close to the open loop path, referred as *trust region* in [53], which can be expressed by a 2-norm ball constraint

$$\|\mathbf{u} - \bar{\mathbf{u}}\|_2 \leq R \tag{2.27}$$

where the R is the radius of the ball, which is set to a small value and has the same unit with $\mathbf{u}$, i.e. mm (milimeter). On the basis of experimental work and approximations, such as the one reported in [54] on the linearity issue, the value is chosen as $R = 0.2$, so that the deformation of the sheet metal will remain in the formable range. Mathematically, this constraint is to ensure the validity of the Talyor series expansion in (2.12)

The optimal control problem can be written as

$$\min_{\|\mathbf{u}-\bar{\mathbf{u}}\|_2 \leq R} \|\mathbf{B}\mathbf{u} - \hat{x}_N\|_2 . \tag{2.28}$$

We now deal with the "tool contact" issue, which is directly linked to the gap between tool position and the sheet. In Figure 2.6, the two curves show two consecutive sheet profiles during the deformation process. As the tool moves across from position r_k to position r_{k+1} without any vertical incremental input, the tool will be suspended, which means it has no contact with the sheet, so the tool will need an additional vertical movement before

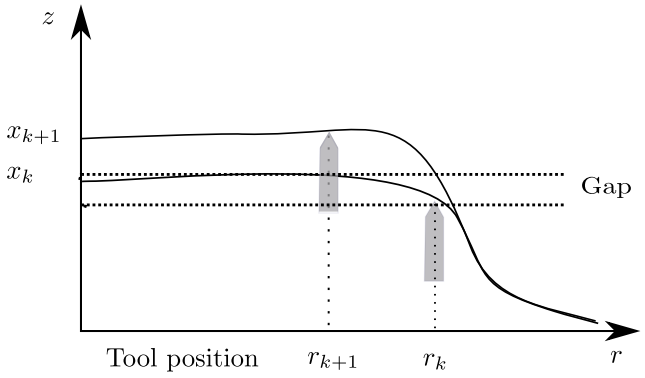


Figure 2.6 To include the "gap" into input value which will be fed to the ISF machine

any deformation occurs. If we denote the actual effective input on the sheet as $\tilde{u}_k$, then we have

$$\tilde{u}_k = u_k - (x_k|_{r_{k+1}} - x_k|_{r_k}), \tag{2.29}$$

where $x_k\,|_{r_k}$ represents the value of x_k at point r_k. *Note that it is u_k (without tilde) that we actually feed into the ISF machine, and following notations with tildes indicate the true amounts of that quanties come into effect in our system.* Define

$$J_k = [0, 0, \ldots, 1, 0, \ldots, 0]^T, \tag{2.30}$$

where only the k^{th} element equals to 1. So

$$J_k^T x_k = x_k\,|_{r_k}\,, \quad J_{k+1}^T x_k = x_k\,|_{r_{k+1}}\,. \tag{2.31}$$

Let

$$P_k^T = J_{k+1}^T - J_k^T, \tag{2.32}$$

so that the effective input is

$$\tilde{u}_k = u_k - P_k^T x_k. \tag{2.33}$$

Correspondingly, (2.22) becomes

$$\begin{aligned} x_{k+1} &= x_k + B_k\tilde{u}_k = x_k + B_k(u_k - P_k^T x_k) \\ &= (I - B_k P_k^T)x_k + B_k u_k, \end{aligned} \tag{2.34}$$

where $I \in \mathbb{R}^{m\times m}$ is an identity matrix of dimension m-by-m. Define matrix

$$E_k = I - B_k P_k^T, \tag{2.35}$$

so that a compact formula can be formed as

$$x_{k+1} = E_k x_k + B_k u_k. \tag{2.36}$$

With $x_0 = 0$ we have

$$x_N = B_{N-1}u_{N-1} + \sum_{k=0}^{N-2}\left(\prod_{i=k+1}^{N-1} E_i\right) B_k u_k \tag{2.37}$$

If we define nominally

$$\prod_{i=N}^{N-1} E_i = I, \tag{2.38}$$

(2.37) can be written in a closed form

$$x_N = \sum_{k=0}^{N-1} \tilde{B}_k u_k \tag{2.39}$$

where

$$\tilde{B}_k = \left(\prod_{i=k+1}^{N-1} E_i\right) B_k. \tag{2.40}$$

Let

$$\tilde{\mathbf{B}} = [\tilde{B}_0, \tilde{B}_1, \ldots, \tilde{B}_{N-1}] \tag{2.41}$$

The linear model then becomes:

$$x_N = \tilde{\mathbf{B}}\mathbf{u}. \tag{2.42}$$

Given B_k and definition (2.32), (2.35), (2.40) and (2.41), $\tilde{\mathbf{B}}$ can be computed off-line. For the constraints in (2.28), from (2.33) we have

$$\tilde{u}_k = u_k - P_k^T \sum_{i=0}^{k-1} \tilde{B}_i u_i. \tag{2.43}$$

Writing this in matrix form:

$$\begin{bmatrix}\tilde{u}_0 & \cdots & \tilde{u}_{N-1}\end{bmatrix}^T = L \begin{bmatrix}u_0 & \cdots & u_{N-1}\end{bmatrix}^T \tag{2.44}$$

where L is the lower triangular matrix with all diagonal entries set to 1 and the entry at i^{th} row, j^{th} column is

$$L_{ij} = -P_{i-1}^T \tilde{B}_{j-1}, (i > j, i > 0).$$

Thus, the ball constraint becomes

$$\left\| L\mathbf{u} - \widetilde{\tilde{\mathbf{u}}} \right\|_2 \leq R, \tag{2.45}$$

where $\widetilde{\tilde{\mathbf{u}}}$ is the actual effective input of the open loop contour tool trajectory. So the open

loop optimal control problem becomes

$$\begin{aligned} &\arg\min_{\mathbf{u}} \left\| \tilde{\mathbf{B}}\mathbf{u} - \widehat{x}_N \right\|_2 \\ &\text{subject to } \left\| L\mathbf{u} - \tilde{\tilde{\mathbf{u}}} \right\|_2 \leq R. \end{aligned} \tag{2.46}$$

2.3 Constrained Model Predictive Control of ISF

We have derived the general open loop model based on the initial state. A general model predictive control (MPC), however, is based on the current state which is measured as $\bar{x}_k$. In the sequel, we adopt the standard notation as in [55],

$$x(k|k) = \bar{x}_k. \tag{2.47}$$

The prediction at step k based on (2.36) becomes

$$x(k+1\,|k\,) = E_k x(k\,|k\,) + B_k u(k|k). \tag{2.48}$$

The prediction horizon is from $k+1 \to N$, and as k increases, the horizon is shrinking, which is different from the standard MPC strategy. We can predict the final output $x(N|k)$ by

$$\begin{aligned} x(N|k) = {} & E_{N-1}E_{N-2}\cdots E_k x(k|k) + E_{N-1}E_{N-2}\cdots E_{k+1}B_k u(k|k) \\ & + \cdots + B_{N-1}u(N-1|k). \end{aligned} \tag{2.49}$$

Let

$$H_k = E_{N-1}E_{N-2}\cdots E_k = \prod_{i=k}^{N-1} E_i \quad (0 \leq k \leq N-1) \tag{2.50}$$

and rewrite (2.49) in a compact form

$$x(N|k) = H_k x(k|k) + \tilde{\mathbf{B}}_k \mathbf{u}(k) \tag{2.51}$$

where

$$\tilde{\mathbf{B}}_k = \begin{bmatrix} \tilde{B}_k & \tilde{B}_{k+1} & \cdots & \tilde{B}_{N-1} \end{bmatrix} \tag{2.52}$$

and

$$\mathbf{u}(k) = \begin{bmatrix} u(k|k) & \cdots & u(N-1|k) \end{bmatrix}^T. \tag{2.53}$$

At step k, the objective function becomes

$$\min \left\| x(N|k) - \widehat{x}_N \right\|_2, \tag{2.54}$$

and substituting (2.51), results in

$$\min \left\| \tilde{\mathbf{B}}_k \mathbf{u}(k) - (\widehat{x}_N - H_k x(k|k)) \right\|_2. \tag{2.55}$$

The actual effective input $\tilde{\mathbf{u}}(k)$ at step k is

$$\begin{bmatrix} u(k|k) - P_k^T x(k|k) \\ u(k+1|k) - P_{k+1}^T x(k+1|k) \\ u(k+2|k) - P_{k+2}^T x(k+2|k) \\ \vdots \\ u(N-1|k) - P_{N-1}^T x(N-1|k) \end{bmatrix} = L(k)\mathbf{u}(k) - D(k)x(k|k), \tag{2.56}$$

where the lower triangular matrix $L(k)$ has a similar structure with L:

$$L(k)_{ij} = -P_{k+i-1}^T \left(\prod_{l=k+i-2}^{k+j} E_l \right) B_{k+j-1} \quad (i > j, 0 < i < N-k) \tag{2.57}$$

and

$$D(k) = \begin{bmatrix} P_k^T & P_{k+1}^T E_k & \cdots & P_{N-1}^T \left(\prod_{i=k}^{N-2} E_i \right) \end{bmatrix}^T. \tag{2.58}$$

The ball constraint can be expressed in the following form

$$\left\| L(k)\mathbf{u}(k) - D(k)x(k|k) - \widetilde{\overline{\mathbf{u}}} \right\|_2 \leq R, \tag{2.59}$$

thus the optimisation problem becomes

$$\begin{aligned} &\min \left\| \tilde{\mathbf{B}}_k \mathbf{u}(k) - (\widehat{x}_N - H_k x(k|k)) \right\|_2, \\ &\text{subject to } \left\| L(k)\mathbf{u}(k) - D(k)x(k|k) - \widetilde{\overline{\mathbf{u}}} \right\|_2 \leq R. \end{aligned} \tag{2.60}$$

This optimisation problem, with $\mathbf{u}(k)$ to be determined, is a constrained least square problem, which can be solved efficiently [56]. Only the first element $\mathbf{u}(k|k)$ will be applied in the process at each step before measuring the deformation state for the next step and repeating this optimisation. Note that the model predictive control method developed above is a variant of the standard strategy since the prediction horizon is shrinking rather than constant.

2.4 Implementation

The experimental rig, on which our algorithm is tested, is an CNC machine at Cambridge University, UK. The sheet metal used in the experiment is 140 mm×140 mm wide, 1mm thick Aluminium 5251-H22. A hemispherical tool is used, where the diameter of the tool is 7.5mm as shown in Figure 2.7. To implement the optimisation problem in (2.60), we need to obtain the online feedback information, which is $x(k|k)$ and the gradient B_k. A stereo camera is placed below the main frame (due to the mechanical configuration of the rig, see Figure 1.7) to obtain the measurements in the form of 3D data points of the sheet metal positions, and by image processing, we can obtain the profiles information online, as explained below. For the gradient B_k, we run an open loop contour following toolpath, record the deformation state at each step and compute the B_k by (2.20).

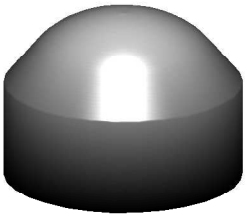


Figure 2.7 Close-up of the actual tool tip

2.4.1 Target Shapes

In our experimental work, two types of shapes are tested, namely *truncated cone* (TC) and the *truncated square pyramid* (TSP). A 3D truncated square pyramid is shown in Figure 2.8, in which marked positions A, B and C are to illustrate the profiles. The profile of a pyramid changes accordingly along the direction $A \rightarrow B \rightarrow C$. In the experiment,

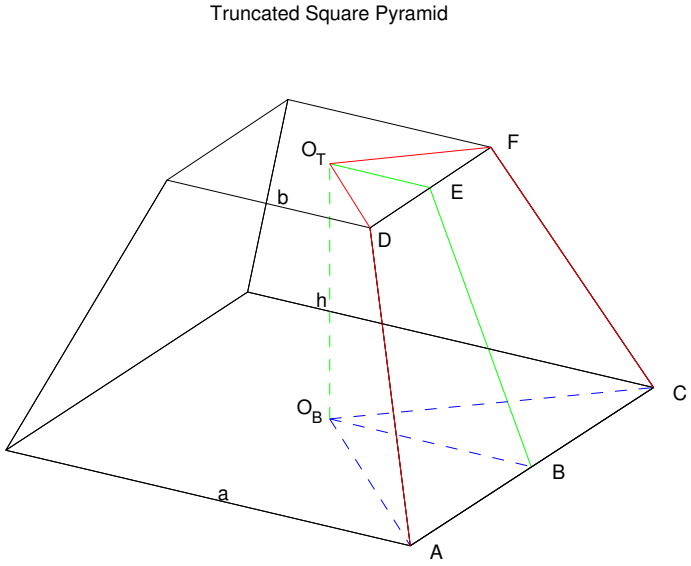


Figure 2.8 Truncated square pyramid: determined by three parameters, two edge-length a, b and the height h.

we choose $a = 100$mm, $b = 40$mm and $h = 30$mm, thus $\angle EBO_B = 45°$ and $\angle DAO_B \approx 35.3°$. Including the truncated cone, the detailed target shape specifications are shown in Figure 2.9, where the dashed lines denote the ideal target shapes whilst the solid lines take into account of the tool effect (with the tool tip radius being 7.5 mm).

The contour following toolpath used in open loop for making a truncated square pyramid is shown in Figure 2.10 and the top view of the contour toolpath is given in Figure 2.11. We can notice that the distances between two adjacent tool positions along radial directions have different values for different profiles. Denoted by variables Δ_1 and Δ_2, we have and $\Delta_2 = \sqrt{2}\Delta_1$.

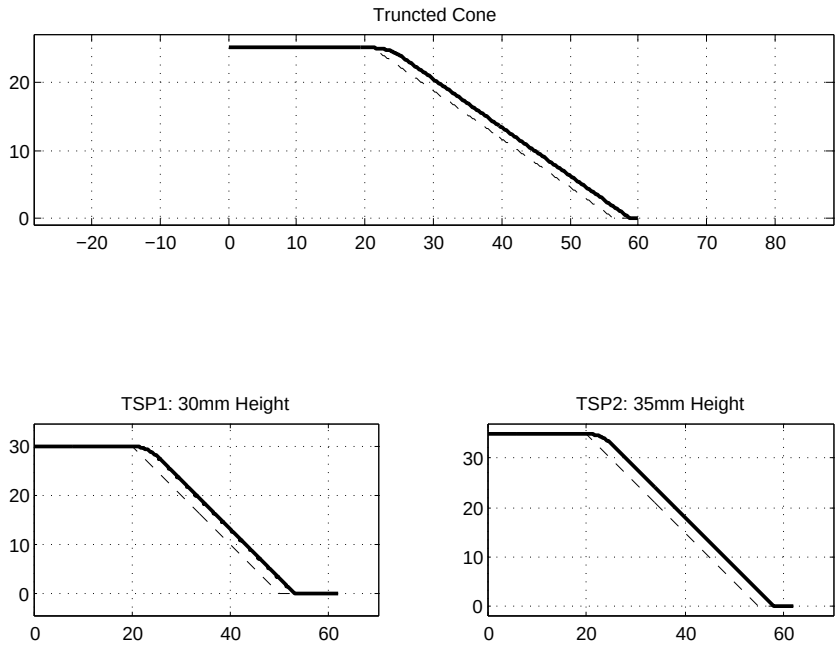


Figure 2.9 Target shapes illustration. Top figure shows the profile of a truncated cone and the lower two subfigures show the truncated square pyramids. The x axis is the radial direction and the y axis is the z (vertical) direction. The dashed lines denote the ideal target shapes whilst the solid lines take into account of the tool effect (for truncated square pyramid,we depicted the middle profiles only here.)

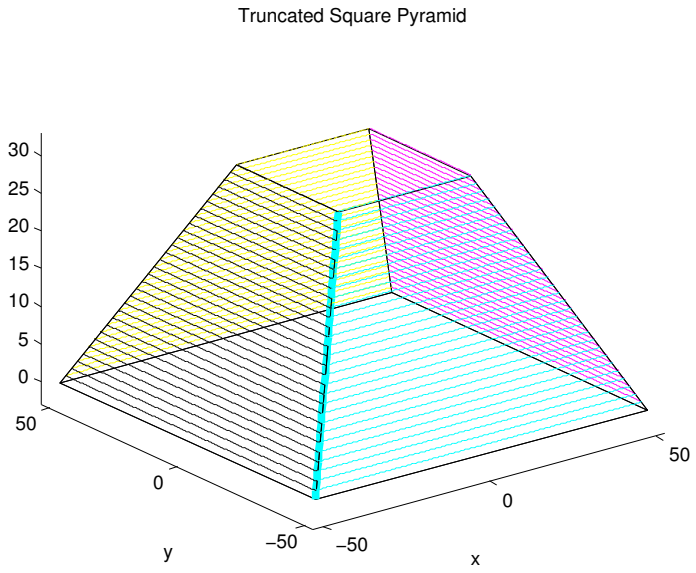


Figure 2.10 General contour toolpath

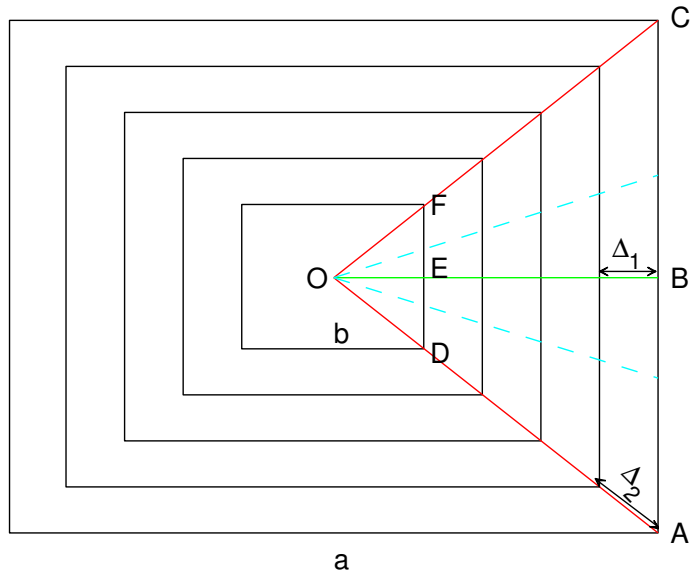


Figure 2.11 Top view of the contour toolpath, $\Delta_1 = \Delta r$ and $\Delta_2 = \sqrt{2}\Delta r$.

2.4.2 Image Processing

The measurements of the deformation are obtained using a stereo-camera and the first step towards implementation is the image processing technique such that the interested geometrical state of the sheet metal can be obtained. For example, a typical post-processed camera data of the truncated cone is visualised in Figure 2.12. To obtain the profile of this

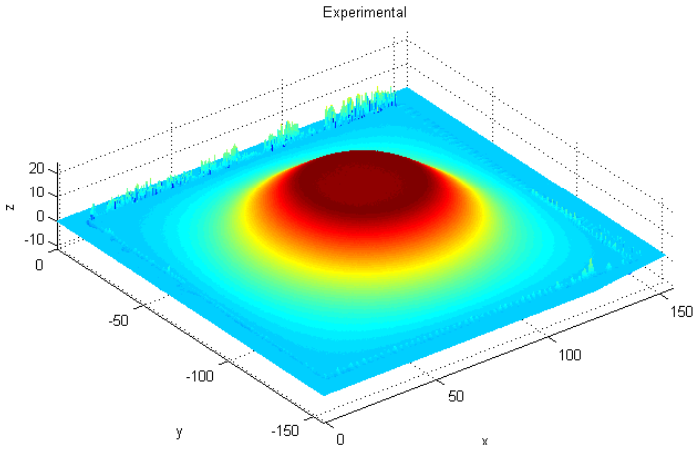


Figure 2.12 Visualised camera positional data for a truncated cone made in the laboratory

shape, we have to first identify the centre of the cone from the image data and based on the centre, we can construct a data matrix that is positioned with the geometrical centre and correspondingly obtain the profile information from two diagonal lines ($3-3'$ and $4-4'$)

and two middle lines ($1-1'$ and $2-2'$) as illustrated in Figure 2.13. Before introducing the

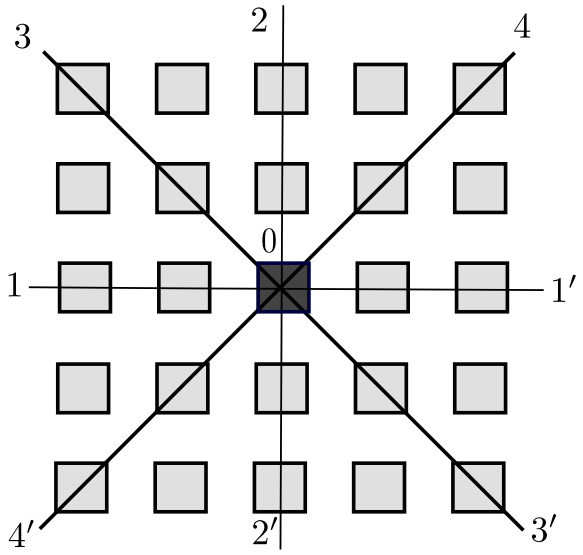


Figure 2.13 Illustration of profile computation. The diagram is an illustration of the square matrix from camera data. 0 is centre position, directions $1-1'$ and $2-2'$ represent the middle profiles, $3-3'$ and $4-4'$ are the diagonal profiles. The final profile is computed as an average of all four directions.

method to obtain the centre, we gives more information about how to obtain the square matrix as the original image from camera data looks like Figure 2.14 (a), which contains abundant information about the frame and other parts of the system. We attach three white stickers onto the frame, which roughly outlined the interested region (a square). The first task is to obtain their location from the camera data. The three points are first separated out from the rest by the pixe darkness, resulting Figure 2.14 (b), where some of noise is present. Next, we adopt a standard image processing technique by first erosion shown in Figure 2.14 (c) and then dilation as in Figure 2.14 (d). Alternatively, one can first label the all possible regions in binary image (b) in Figure 2.14. Since the white stickers occupy larger areas, we can identify their correct locations using functions (`bwlabel` and `regionprops`) in image processing toolbox in MATLAB. A post-processed image is shown as in Figure 2.14 (f). Also, the orientation of the image data, which is a rectangular frame, is not necessarily perfectly lined up as one sides parallel to the normal horizontal direction and the other to the vertical direction such as the case shown in Figure 2.15, which shows the camera data of a truncated square pyramid made by contour following toolpath. So an additional matrix rotation may needed.

The centre of the shape can now be found using an ellipse-fitting algorithm proposed by [57] and the profiles can be obtained. As an example, for a truncated square pyramid, we are interested at both middle and corner profiles, which are shown in Figure 2.16. For the same shape with two different camera samplings, i.e. two measurements, the difference looks like Figure 2.17. The camera data accuracy is known to be within $\pm 0.1mm$. Another fact regarding to the accuracy of the data is that the camera sampling spatio interval is around $0.4mm$ along the x and y axes and we interpolate the camera data values at the

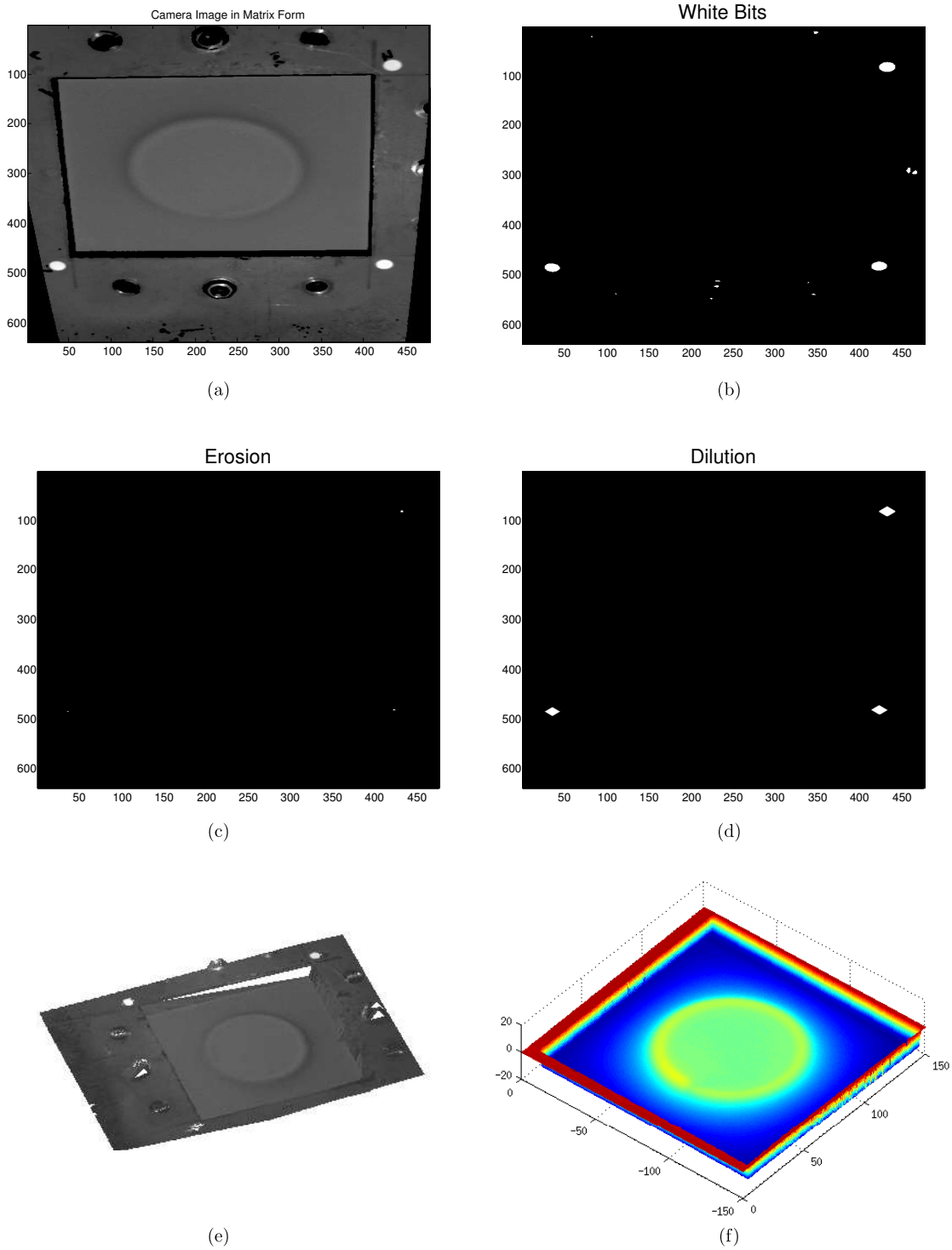
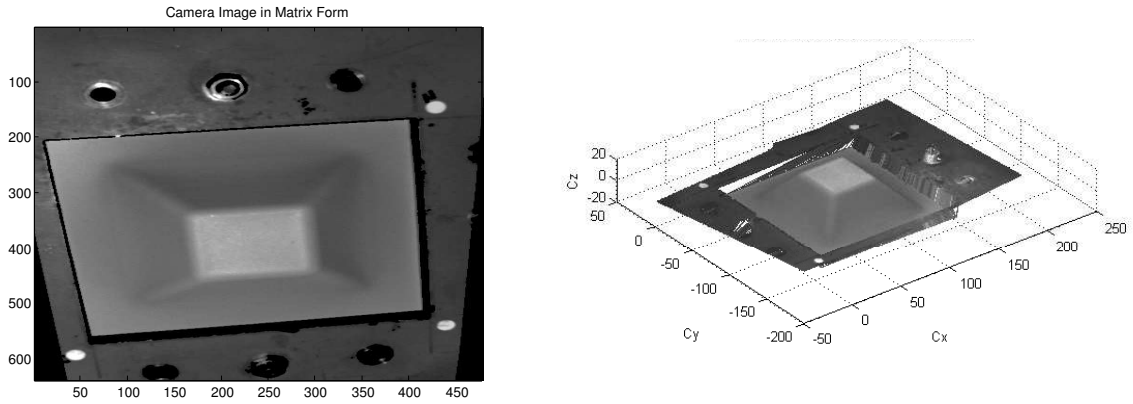


Figure 2.14 Visualised camera positional data for a truncated cone that is being forming. (a) original data (b) binary image containing the locations of the white stickers (c) erosion of the image to filter the noises (d) dilution of the interested white dots to obtain their centres (e) full version of the data camera (f) grid data for profile extracting



(a) (b)

Figure 2.15 Truncated square pyramid made by contour following method (a) original camera data (b) rotated view

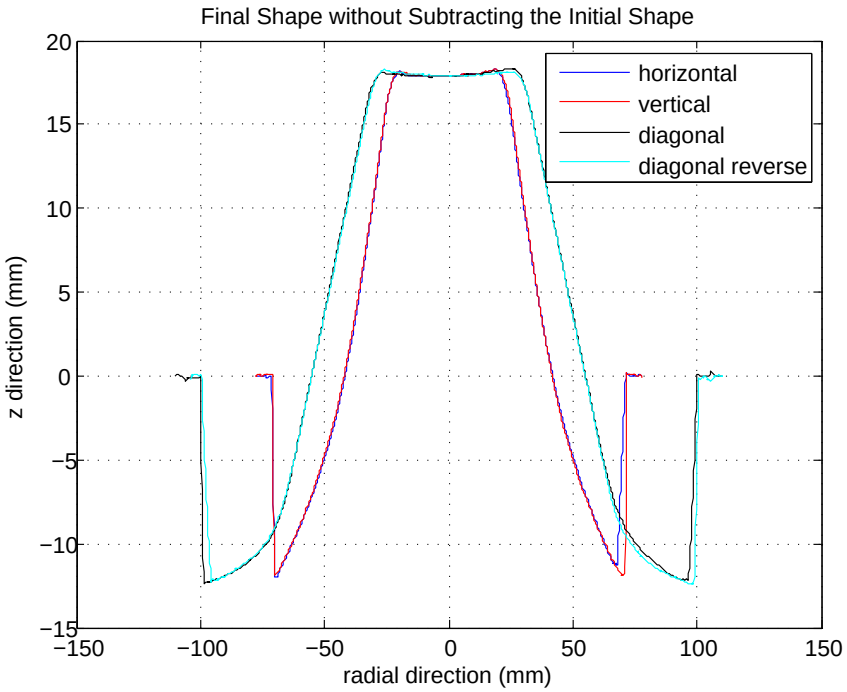


Figure 2.16 Shape profiles extracted from post-processed camera data for a truncated square pyramid.

the interval of $0.1mm$ linearly.

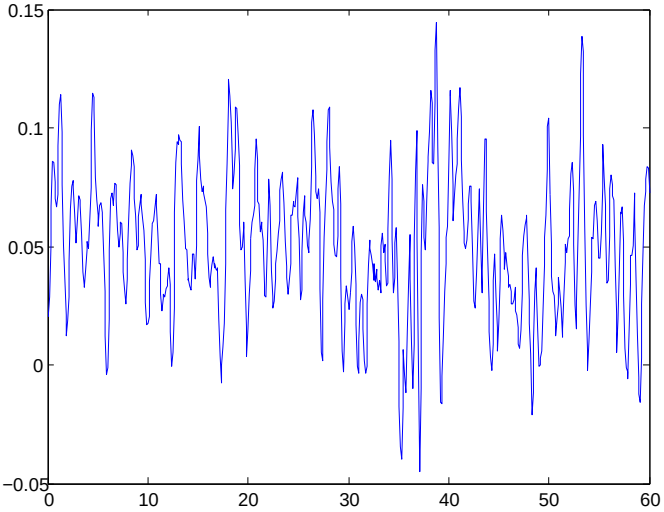


Figure 2.17 Difference of two measurements for the same shape

2.5 Experimental Results

The feedback constrained MPC control algorithm is first tested by manufacturing a truncated cone and comparing with the same cone made by contour following approach. First a truncated cone, we tried two incremental step lengths (Δ, or Δr) in radial direction:

1. $\Delta r = \sqrt{2}$ mm
2. $\Delta r = \frac{\sqrt{2}}{2}$ mm

and the final shapes are shown in Figure 2.18 and Figure 2.19, where the black lines indicate the target shape profiles, the red lines denote the profiles of the shape made by normal open loop contour following method and the blue lines denote the profile of the shapes made using our constrained MPC controller. Figure 2.20 and Figure 2.21 give the detailed error distributions of the profiles for these two cases.

Results shown in the diagrams are compared with corresponding contour following methods. Note that the effect of tool diameter on the shape has already been taken into consideration in the target shape. We set the number of the steps to be optimised as 25, when the fixed radial distance is $\sqrt{2}$mm. The error is brought down to a level close to $\pm$0.2mm. The actual input u_k fed to the machine is shown in Figure 2.22 which differs from the contour following input significantly.

We applied this strategy to a truncated square pyramid (TSP), where we set the number of the steps to be optimised as 35, and the fixed radial distance as 1mm. Instead of using one profiles as state of the system, we use both the corner and the middle profiles and run two online optimisations to compute two inputs: one at the corner and the other in the middle. The tool trajectory is obtained by applying a 4^{th} order polynomial

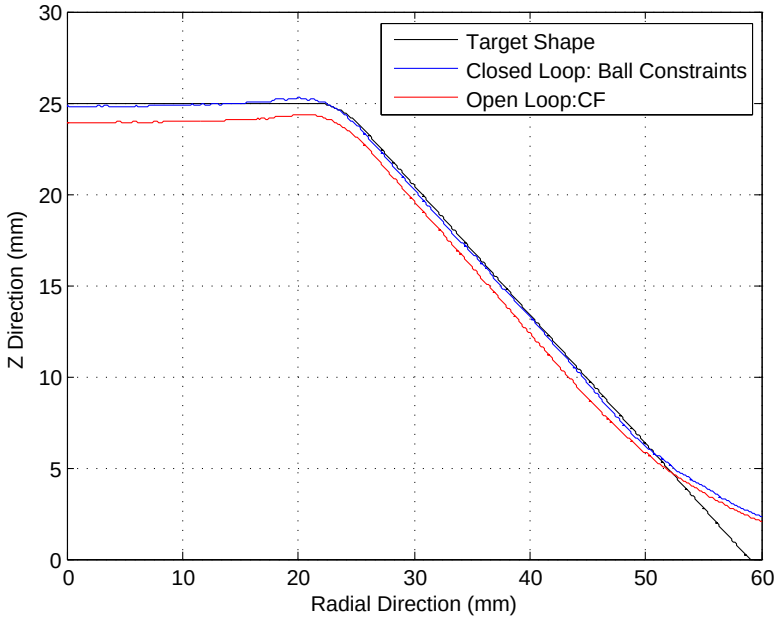


Figure 2.18 Final shape of a truncated cone, with $\Delta r = \sqrt{2}$ mm

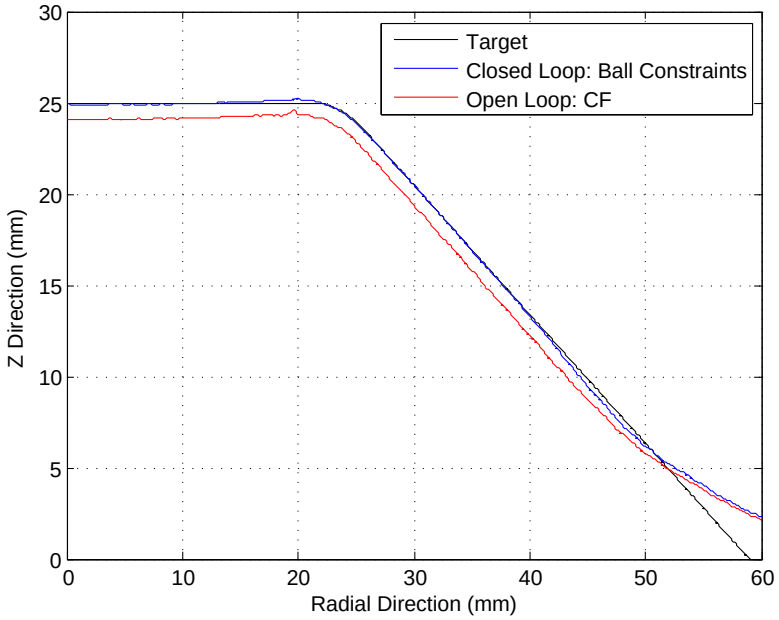


Figure 2.19 Final shape of a truncated cone, with $\Delta r = \frac{\sqrt{2}}{2}$ mm

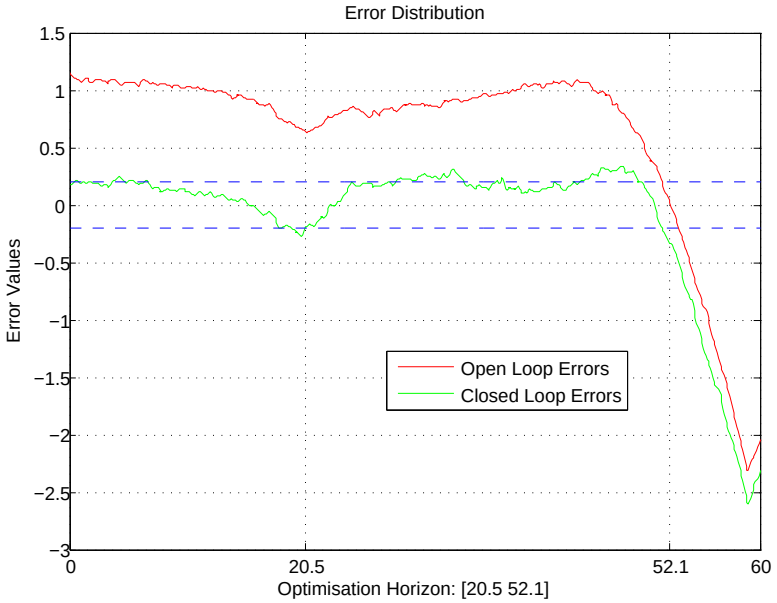


Figure 2.20 Error distribution of a truncated cone, with $\Delta r = \sqrt{2}$ mm

Figure 2.21 Error distribution of a truncated cone, with $\Delta r = \frac{\sqrt{2}}{2}$ mm

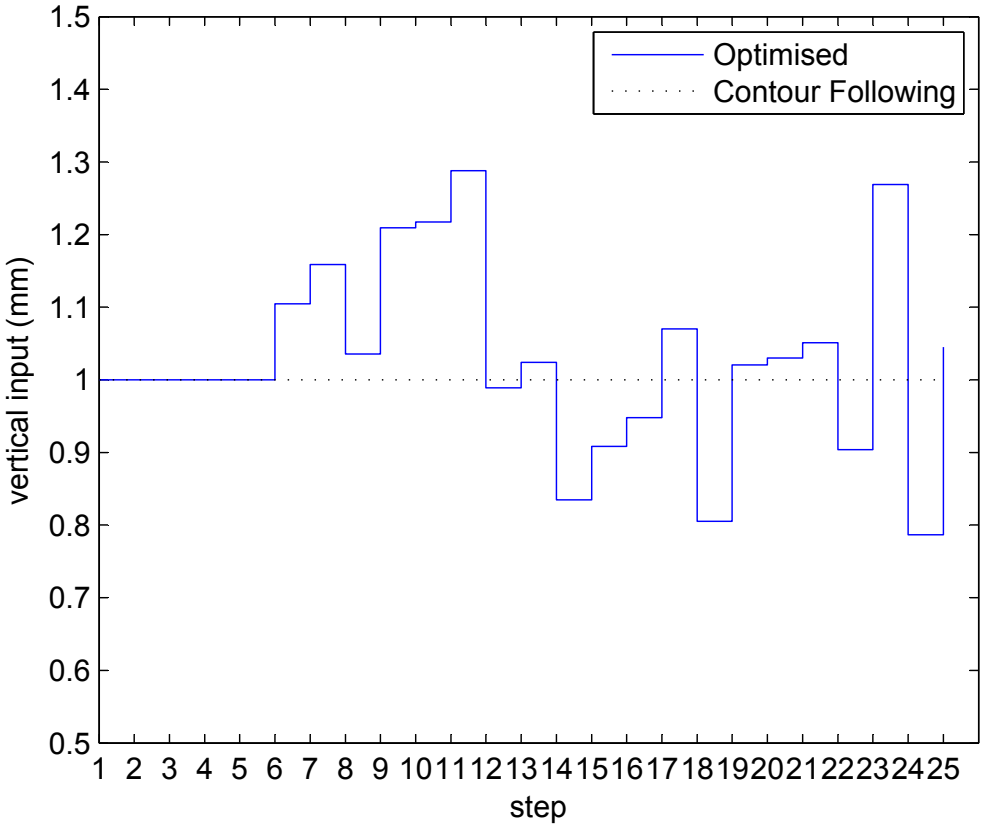


Figure 2.22 Inputs: u optimised vs contour following

interpolation to points between the corner and the middle profiles. A general fourth order polynomial can be expressed as $f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$. Since we require a symmetrical function, which means $f(x) = f(-x)$, then $a_3 = a_1 = 0$. The other three parameters can be determined by three conditions:

1. $f(0) = A$;
2. $f(x_0) = 0$;
3. $f'(x_0) = 0$.

By simple calculation, we have $a_4 = \frac{A}{x_0^4}$, $a_2 = -\frac{2A}{x_0^2}$ and $a_0 = A$. An example is shown in Figure 2.23.

The final shape produced for the truncated square pyramid is shown in Figure 2.24 and the detailed error distributions are shown in Figure 2.25. (Red lines show the results of contour following method and the blue lines show the results of constrained MPC method.) Since the truncated square pyramid is less symmetric than a cone shape, it is not surprisingly that the result for the TSP is worse than that for the cone. Two parts produced from the laboratory are shown in Figure 2.26 and Figure 2.27. An inappropriate control input that violates the necessary constraints (due to a programming error) may result in failure such as the one shown in Figure 2.28.

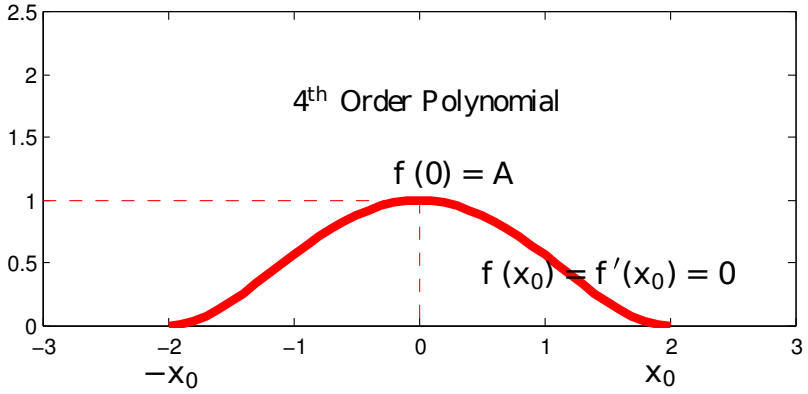


Figure 2.23 An example of possibly desired fourth order polynomial: $f(x) = 0.0625x^4 - 0.5x^2 + 1$.

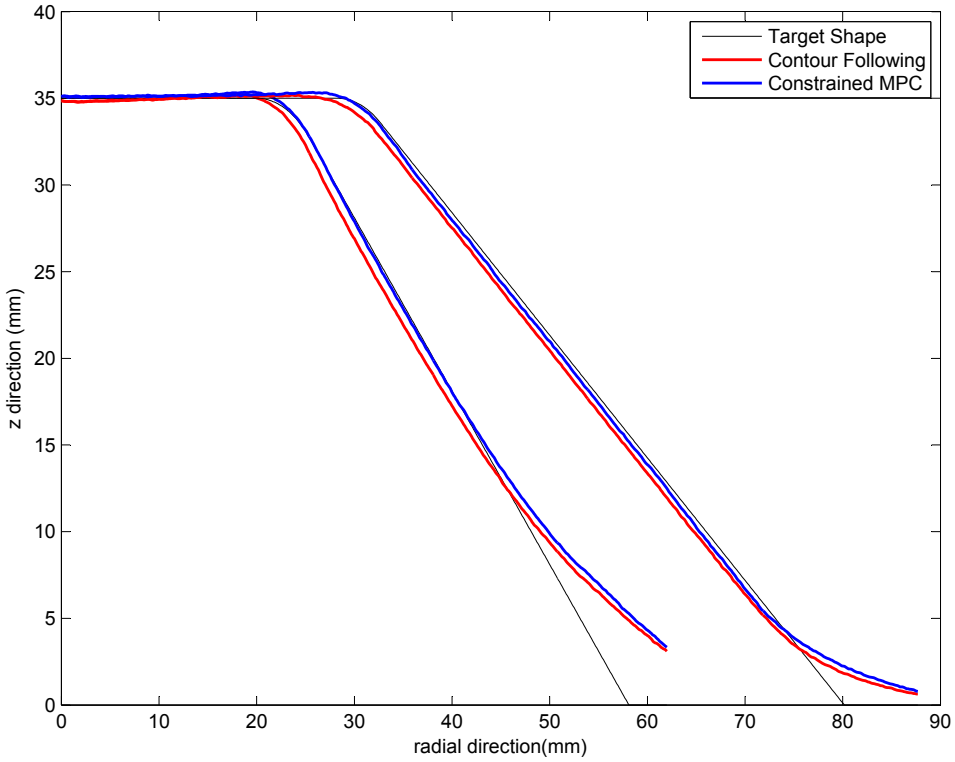


Figure 2.24 TSP 35mm height final shape under both contour following and constrained model predictive control

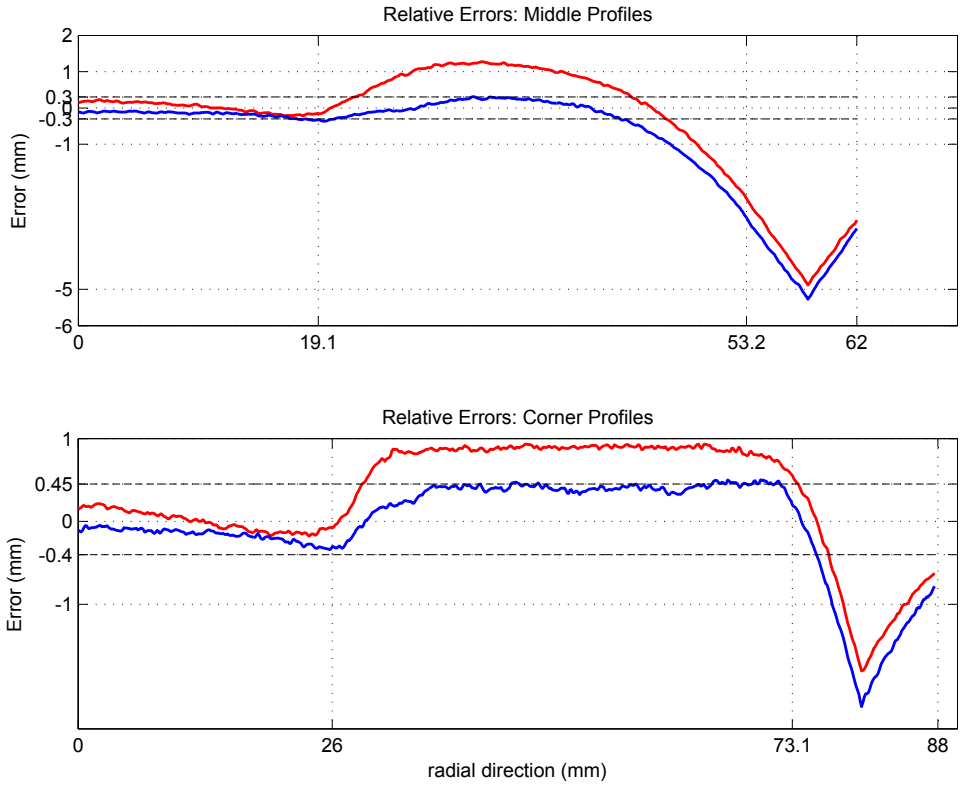


Figure 2.25 Error distribution of truncated square pyramid. The red line indicates the geometrical error of the shape by contour following method and the blue line indicates the geometrical error of the shape made by constrained MPC.

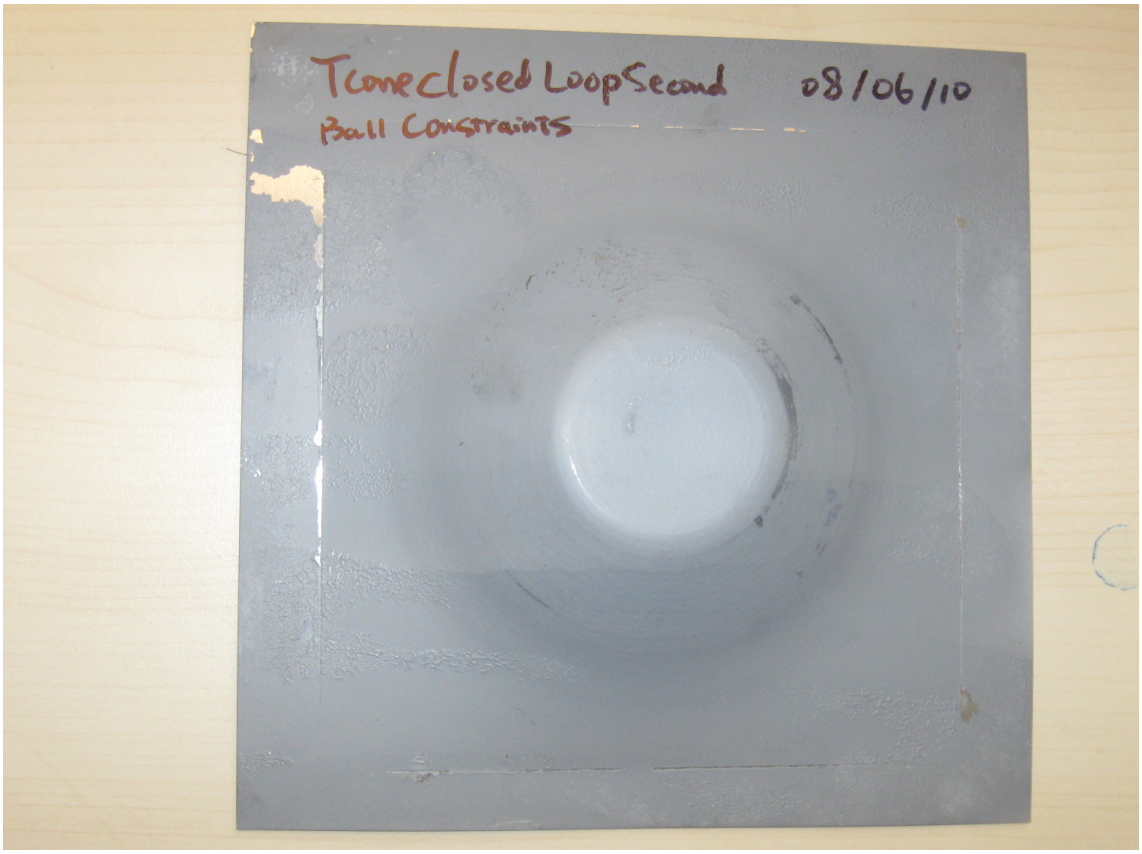


Figure 2.26 Truncated cone made by ISF

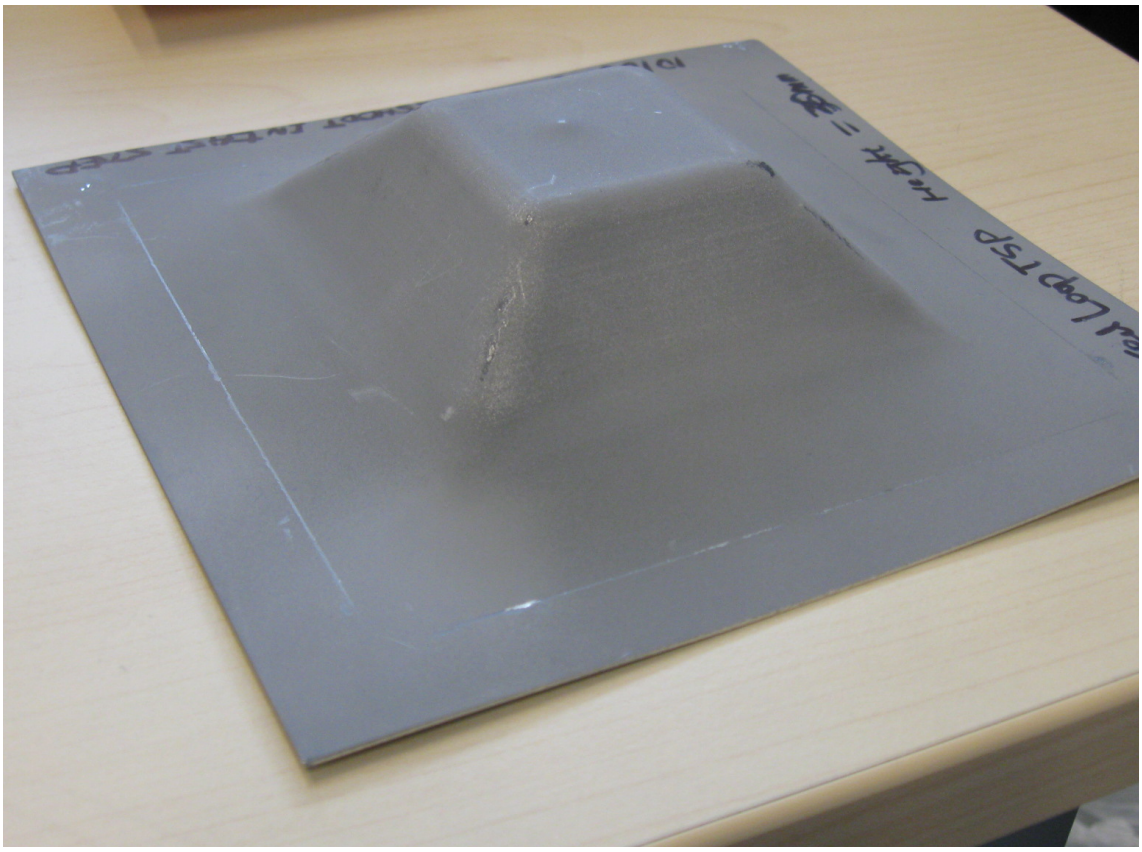


Figure 2.27 Truncated Square Pyramid made by ISF

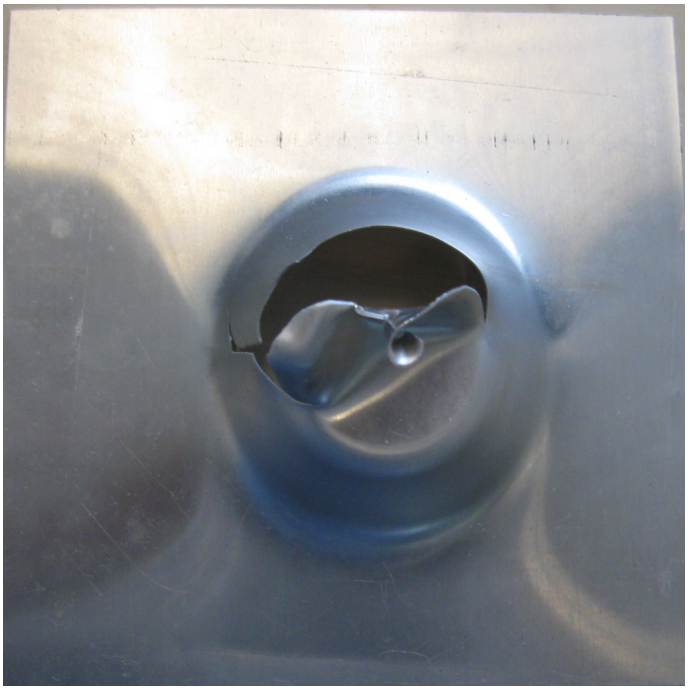


Figure 2.28 Cracked cone made in experiment

2.6 Discussion and Conclusion

As we have shown, the proposed constrained MPC successfully reduced the geometric errors of the shapes produced by ISF. In its best scenario, the deviations of the central part of the truncated cone can be limited to the target ± 0.2 mm, whilst for the truncated square pyramid the deviations also have been improved. The optimised tool trajectory implied by Figure 2.22 differs from that of the standard contour following method, which confirms the benefits of using MPC, although the shrinking horizon scheme can cause the last few inputs to behave aggressively.

On the other hand, we still have some reservations upon examining the profiles of the shapes produced using current algorithm. We may notice that the region that gets most improved is where it is neither too close to the symmetric axis of the shape nor to its edges. The deviations in these areas can be improved even more by a smaller radial incremental step as shown in Figure 2.20 and Figure 2.21, but not much around the flange (the region that is close to the edges) and neither the flat region. It may be easier to accept a less accurate flange improvement, given that we do not have any kind of support for the sheet metal near the blankholder. However, the "shoulder" region that connects the flat part and the slope for both truncated cone and truncated square pyramid behaves differently from the middle areas, which is the reason that the optimisation horizon (or the range of the sampling points used)—shown as a pair of dotted vertical lines in previous graphs—is carefully selected. Ideally, we should include the whole profile as our optimisation horizon, but the deformation results suggest otherwise. This feature in our algorithm is especially not welcome, given our objective for controlling ISF is not just about making simple geometries.

To analysis a possible cause for this effect, we turn to the computation of the gradient $\mathbf{B}$. As explained earlier, B_k is computed from

$$B_k = \frac{\bar{x}_{k+1} - \bar{x}_k}{\bar{u}_k}, \tag{2.61}$$

where $\bar{x}_k$ are the predetermined experimental profile at layer k, and $\bar{u}_k$ is the corresponding (vertical) input. As we set $\bar{u}_k = 1$, we also refer to B_k as *impulses* physically. In addition, B_k is smoothed using basic averaging technique as B_k essentially models the material behaviour. Comparing this gradient with experimental data, we can observe significant difference between the two as the number of steps grow, which are illustrated in Figure 2.29 and Figure 2.30. While the overall shape computed gradients look similar to the experimental data, the height of the gradient is much larger (refer to Figure 2.30). As a result, the final shape computed accumulatively from the model has a much higher profile indicated by the blue solid line in Figure 2.31. From Figure 2.32 to Figure 2.36, we illustrate the difference between the prediction and the experimental results over a number of steps from step 5 to step 24. The upper plots show the two consecutive profiles computed from the model at each step, i.e. the prediction (solid line) and the actual closed loop experiment results (dashed line), whilst the lower plots show the B_k from the model

(solid line) and the actual computed B_k from closed loop experiment (dashed line). We can observe that the model is actually operating outside the linear region defined by the ball constraints in (2.27) (notice how much difference the B_k illustrated in the lower plots) as the step increases. The "shoulder" region especially has larger deviation. The reason why the constrained MPC remain relatively effective has to be credited to the closed loop scheme.

Therefore, it can be drawn from our experimental results obtained here that a good deformation model will be required to further improve the geometrical accuracy of the shape made by ISF using model predictive control strategy, which motivate our physics based modelling in the following chapters.

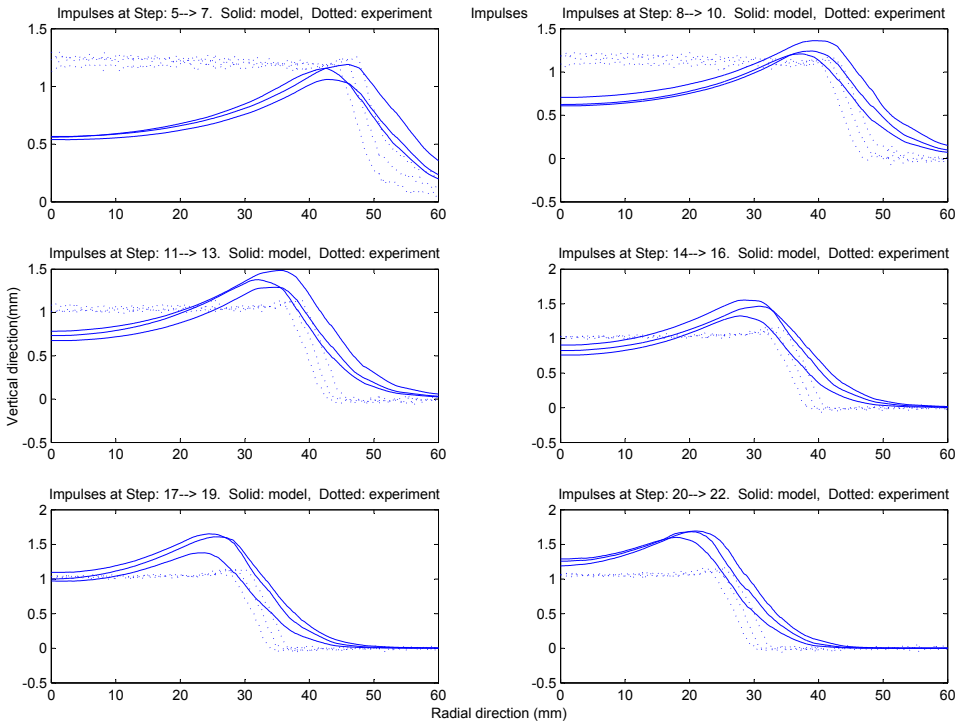


Figure 2.29 Gradient B_k: dotted lines show the unsmoothed experimental results which are of around 1mm height whereas the heights of the impulses in the model have much higher values shown in solid lines.

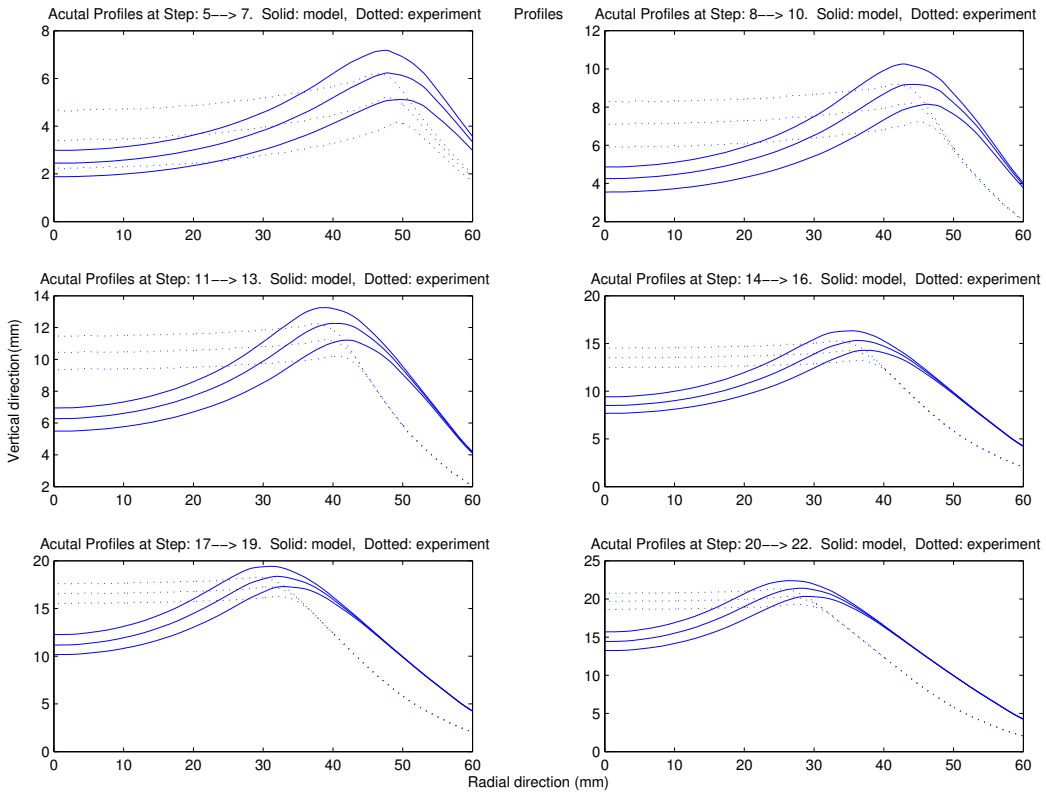


Figure 2.30 The profiles of the truncated cone, pre-computed (solid lines) vs measured (dotted lines).

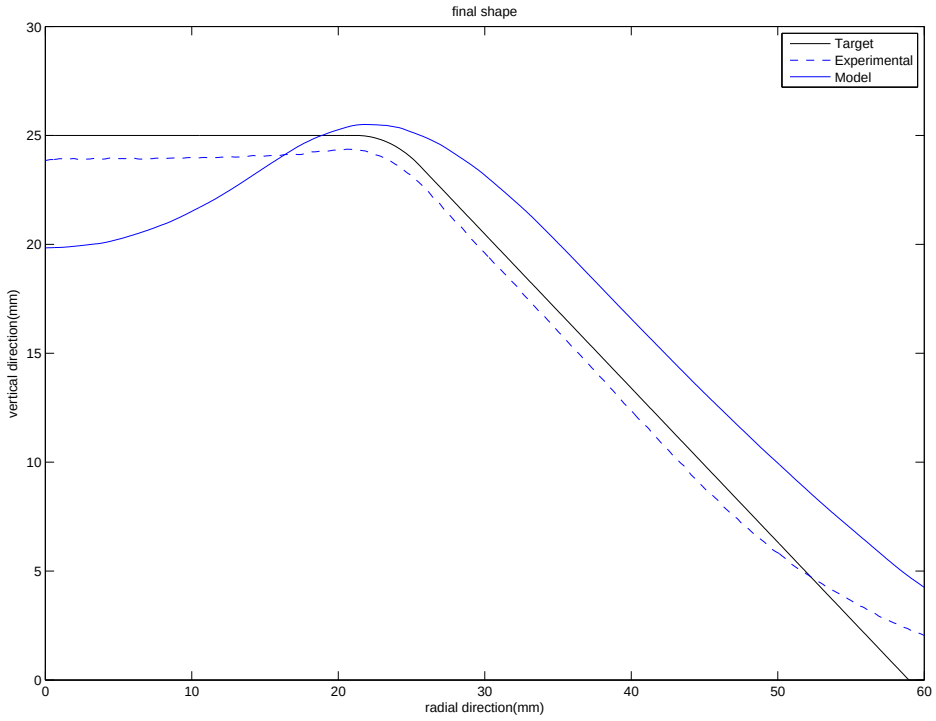


Figure 2.31 The final profiles of the truncated cone. Target profile (solid black line), experimental results (dashed blue line) and pre-computed (solid blue line)

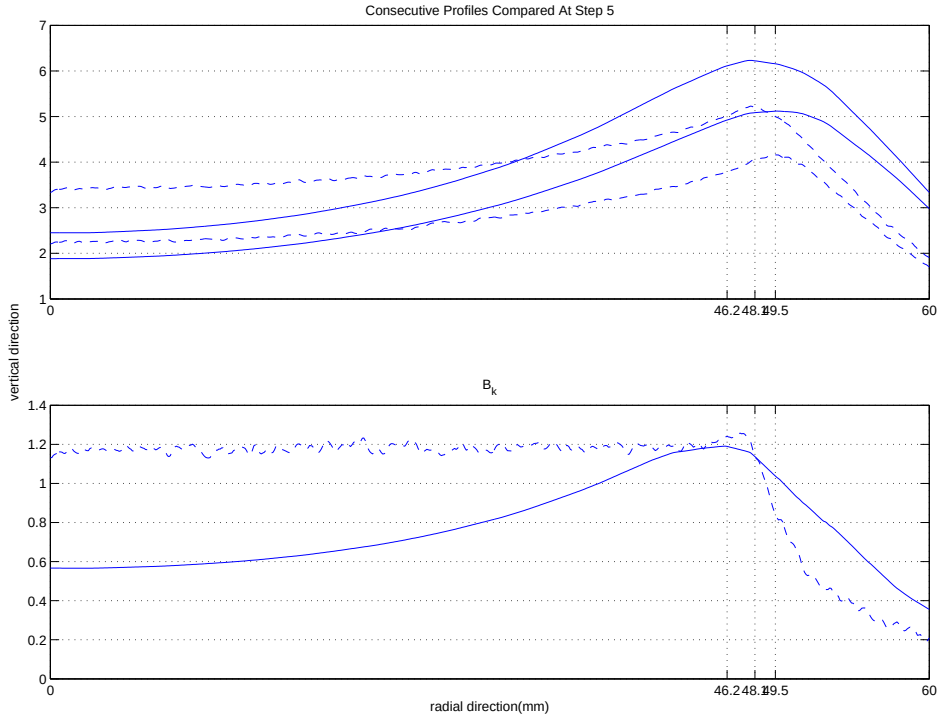


Figure 2.32 Two consecutive profiles and the corresponding gradient (impulses) computed at Step 5. Solid line (from model), dashed line (closed loop experiment result)

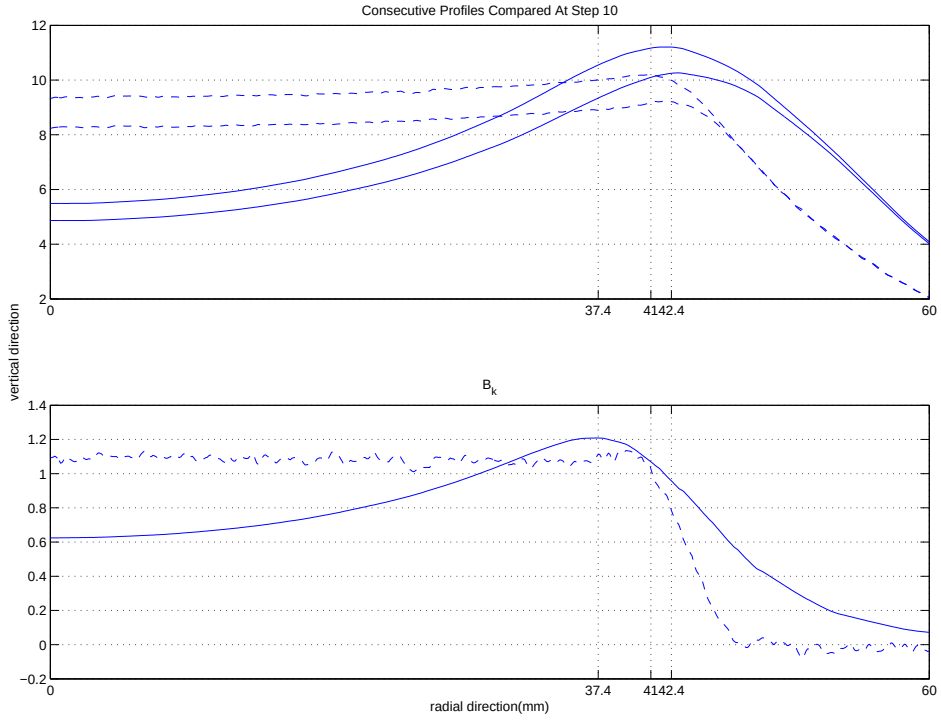


Figure 2.33 Two consecutive profiles and the corresponding gradient (impulses) computed at Step 10. Solid line (from model), dashed line (closed loop experiment result)

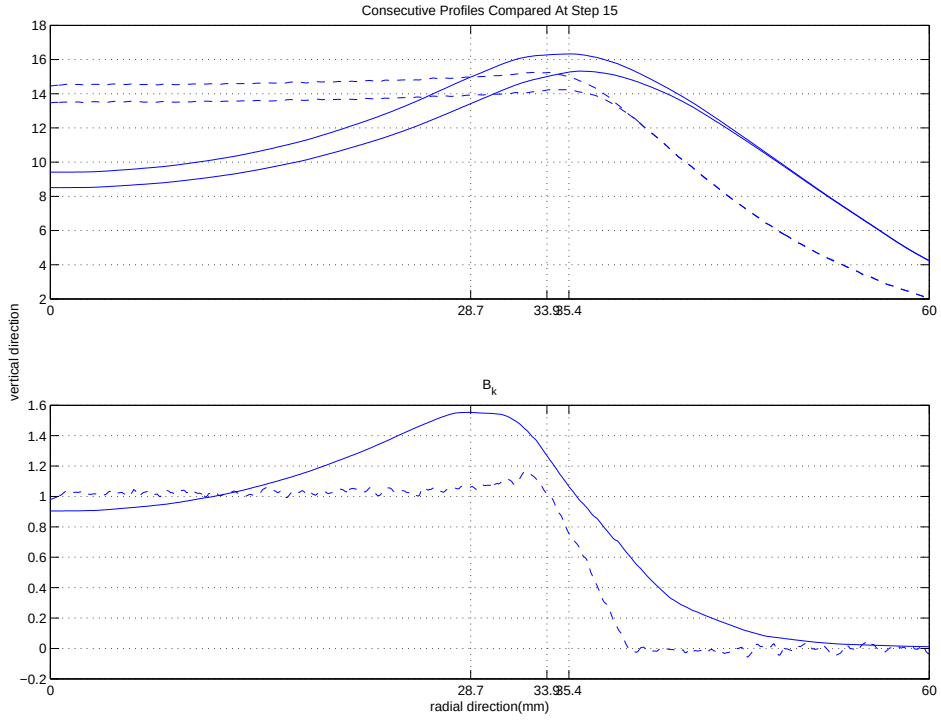


Figure 2.34 Two consecutive profiles and the corresponding gradient (impulses) computed at Step 15. Solid line (from model), dashed line (closed loop experiment result)

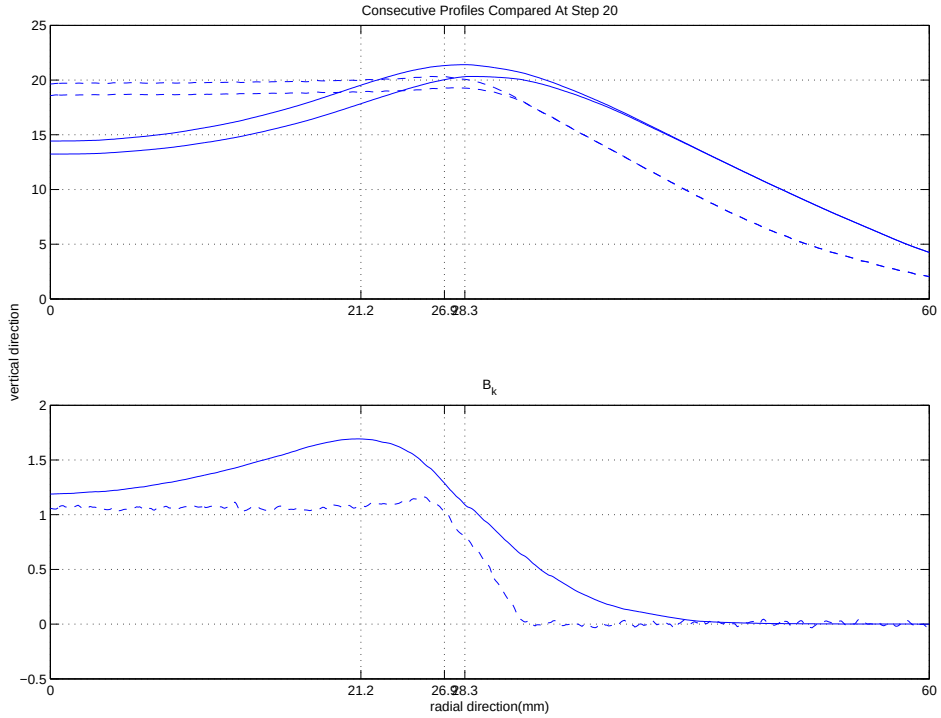


Figure 2.35 Two consecutive profiles and the corresponding gradient (impulses) computed at Step 20. Solid line (from model), dashed line (closed loop experiment result)

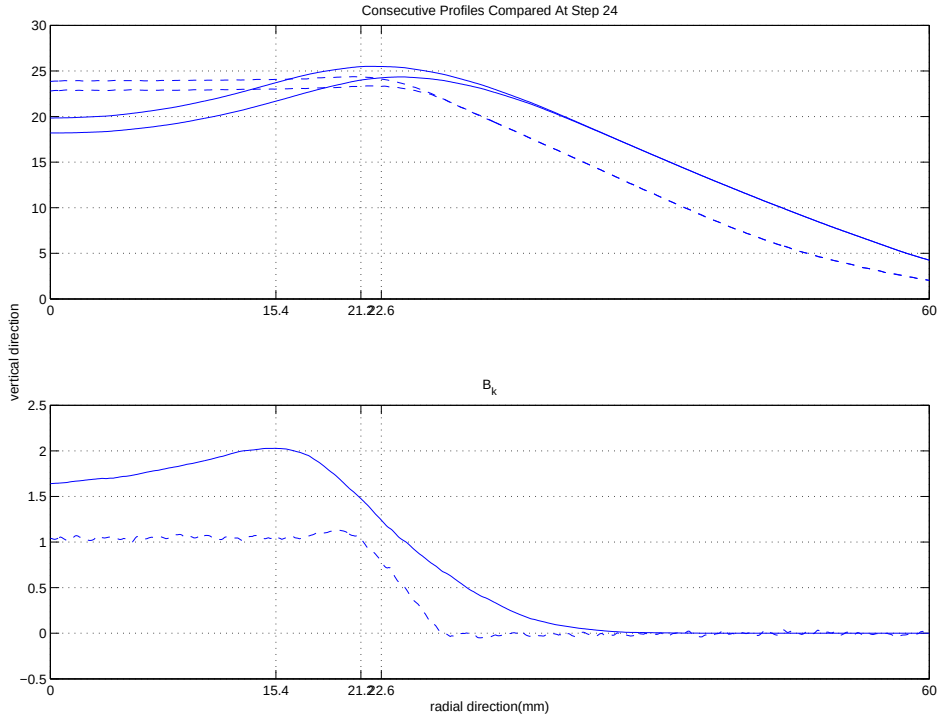


Figure 2.36 Two consecutive profiles and the corresponding gradient (impulses) computed at Step 24. Solid line (from model), dashed line (closed loop experiment result)

Chapter 3

Physics Based ISF Process Modelling

The motivation of constitutive modelling is twofold: so far our model is based on an approximation obtained from experimental data and the limitation of this model has been described in previous chapters. Since the nature of ISF deformation falls into the elasto-plastic category, a physics based model would first allow us to obtain a clearer picture of the deformation process and most importantly to obtain a better mathematical predictive model that could help the design of a controller. The elasto-plasticity material modelling is first introduced before ISF is cast as a contact problem later in this chapter.

The favour of the work in this chapter and the following is more mathematical, and some of the common notations used later are list here. For the tensor operations, more details can be found in Appendix A.

Symbols	Meanings
$\boldsymbol{\sigma}$	stress tensor
$\boldsymbol{\varepsilon}$	strain tensor
$\boldsymbol{I}$	second-order indentity tensor
I	fourth-order indentity tensor
$\mathsf{I_d}$	deviatoric projection tensor
$\mathsf{I_s}$	symmetric projection tensor
$\mathsf{I_t}$	transpositon tensor
p	hydrostatic pressure
$\boldsymbol{s}$	deviatoric stress
$\boldsymbol{\varepsilon}_d$	deviatoric strain
ε_v	volumetric strain
γ	plastic multiplier
$\boldsymbol{\varepsilon}^p$	plastic strain
$\boldsymbol{f}_B$	body force
$\boldsymbol{f}_N$	traction on boundary Γ_σ
:	tensor contraction operator
$\otimes$	tensor product

Table 3.1 List of notations

3.1 Stress and Strain Tensor

A deformable material under external loading will change the arrangement of the molecules and the initial state of equilibrium will be broken. To balance the state of equilibrium, internal forces arises which are called *internal stress*. Elasticity theory assumes that the force acts within a limited range, which implies that internal force originating from the surrounding parts exert on the body only on the surface of that part [58]. Let us consider the total force of some part of the body, so we sum up all these forces on all volume elements of that part of the body, which can be expressed as a volume integral $\int \boldsymbol{F}\,\mathrm{d}V$ where $\boldsymbol{F}$ is the force per unit volume and $\boldsymbol{F}\,\mathrm{d}V$ is the force on the volume element $\mathrm{d}V$. From the above assumption, these force can be expressed as the sum of forces acting on the surface element, i.e. an integral over the surface mathematically. The bridge between the force volume integral and the unknown surface integral is the *divergence theorem*, which states that an integral of a scalar (rank-zero tensor) over an arbitrary volume is equal to a vector (rank-one tensor) integral if the scalar is the divergence of the vector. In the case of a vector integrand $\boldsymbol{F}$ (tensor of rank one), it must be the divergence of a tensor of rank two, i.e. using summation convention[1]

$$F_i = \frac{\partial \sigma_{ik}}{\partial x_k}, \tag{3.1}$$

where $k = 1, 2, 3$ and x_k denotes the Cartesian coordinates. The force is then described as

$$\int F_i \,\mathrm{d}V = \int \frac{\partial \sigma_{ik}}{\partial x_k}\,\mathrm{d}V = \oint \sigma_{ik}\,\mathrm{d}S_k, \tag{3.2}$$

where $\mathrm{d}S_k$ are the components of the surface element vector $\mathrm{d}\boldsymbol{S}$, directed outward normal. σ_{ik} is called the *stress tensor* and $\sigma_{ik}\,\mathrm{d}S_k$ is the i^{th} component of the force, which is perpendicular to the surface element $\mathrm{d}S$. The directions of the stress tensor is illustrated in Figure 3.1 on an infinitesimal cube. Note that the dashed lines indicate the tensor on the shadow surfaces.

Next we consider the moment of force $\boldsymbol{F}$ on the whole volume, whose component M_{ik} points in a direction perpendicular to both that of x_i and x_k coordinates. We thus have the expression

$$M_{ik} = \int (F_i x_k - F_k x_i)\,\mathrm{d}V. \tag{3.3}$$

Substituting with force tensor equation (3.1), yields

$$\begin{aligned} M_{ik} &= \int \left(\frac{\partial \sigma_{il}}{\partial x_l} x_k - \frac{\partial \sigma_{kl}}{\partial x_l} x_i \right) \mathrm{d}V \\ &= \int \frac{\partial (\sigma_{il} x_k - \sigma_{kl} x_i)}{\partial x_l}\,\mathrm{d}V - \int \left(\sigma_{il} \frac{\partial x_k}{\partial x_l} - \sigma_{kl} \frac{\partial x_i}{\partial x_l} \right) \mathrm{d}V. \end{aligned} \tag{3.4}$$

[1] summation convention: when a index appears twice in a single term it implies summation of that term over all the values of the index. For Roman letter i, j, k the range is set as $\{1, 2, 3\}$ and Greek letter α, β the range is set to $\{1, 2\}$.

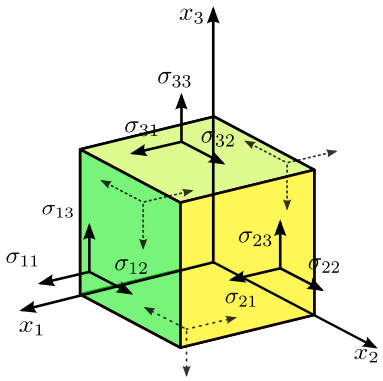


Figure 3.1 Illustration of stress tensor on an infinitesimal cube in an orthogonal coordinate.

Noting that $\dfrac{\partial x_k}{\partial x_l} = \delta_{kl}$ and $\dfrac{\partial x_i}{\partial x_l} = \delta_{il}$, and the multiplication gives $\sigma_{il}\delta_{kl} = \sigma_{ik}$, then the second integrand becomes

$$\int (\sigma_{ik} - \sigma_{ki})\ \mathrm{d}V.$$

The first integrand is a divergence, which can be rewritten as a surface integral from the divergence theorem. The moment M_{ik} is

$$M_{ik} = \oint (\sigma_{il}x_k - \sigma_{kl}x_i)\,\mathrm{d}S_l - \int (\sigma_{ik} - \sigma_{ki})\ \mathrm{d}V. \tag{3.5}$$

Just like the force, the moment on the total volume can be expressed as only the surface integral, which means the symmetric property of the stress tensor since the volume integral term in (3.5) has to vanish, so

$$\sigma_{ik} = \sigma_{ki}.$$

The stress tensor $\boldsymbol{\sigma}$ defined with respect to the Cartesian coordinate can be written as

$$\boldsymbol{\sigma} = \sigma_{ij}\boldsymbol{e}_i \otimes \boldsymbol{e}_j, \tag{3.6}$$

whose symmetric components in a matrix form are

$$(\sigma_{ij}) = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}. \tag{3.7}$$

If the stress occurs only in a plane, which is named *plane stress*, then

$$\sigma_{13} = \sigma_{23} = \sigma_{33} = 0.$$

Stress Invariants are the quantities of $\boldsymbol{\sigma}$ that are independent of the coordinate transformation, which are required for constitutive modelling in the later chapters. The

invariants come from the determinant

$$|\sigma_{ij} - \lambda\delta_{ij}| = \begin{vmatrix} \sigma_{11}-\lambda & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22}-\lambda & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33}-\lambda \end{vmatrix} = 0, \quad \delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j, \end{cases} \tag{3.8}$$

whose expanded form is a characteristic equation

$$|\sigma_{ij} - \lambda\delta_{ij}| = -\lambda^3 + I_1\lambda^2 - I_2\lambda + I_3 = 0. \tag{3.9}$$

I_1, I_2 and I_3 are called the *first, second* and *third* stress invariant, whose values for components defined in Cartesian coordinates are well known as

$$\begin{aligned} I_1 &= \sigma_{11} + \sigma_{22} + \sigma_{33} = \sigma_{ii}, \\ I_2 &= \sigma_{11}\sigma_{22} + \sigma_{22}\sigma_{33} + \sigma_{11}\sigma_{33} - \sigma_{12}^2 - \sigma_{23}^2 - \sigma_{31}^2 \\ &= \frac{1}{2}\left(\sigma_{ii}\sigma_{jj} - \sigma_{ij}\sigma_{ji}\right), \\ I_3 &= \det(\boldsymbol{\sigma}). \end{aligned} \tag{3.10}$$

Deviatoric and hydrostatic stresses It is often convenient to split the stress tensor $\boldsymbol{\sigma}$ into two parts

$$\boldsymbol{\sigma} = \boldsymbol{s} + p\boldsymbol{I}, \tag{3.11}$$

where the invariant

$$p \equiv \frac{1}{3}I_1(\boldsymbol{\sigma}) = \frac{1}{3}\text{trace}(\boldsymbol{\sigma}) \tag{3.12}$$

is the *hydrostatic pressure*, and

$$\boldsymbol{s} \equiv \boldsymbol{\sigma} - p\boldsymbol{I} = \mathsf{I}_\mathsf{d} : \boldsymbol{\sigma}, \tag{3.13}$$

where I_d is defined in (A.13), is a traceless tensor referred as the *deviatoric stress*, and $\boldsymbol{I}$ is a second-order identity tensor. A coordinate-independent definition for the trace of a second-order tensor $\boldsymbol{\sigma}$ is given by

$$\text{trace}(\boldsymbol{\sigma}) = \boldsymbol{I} : \boldsymbol{\sigma}. \tag{3.14}$$

As a second order tensor, the stress deviator $\boldsymbol{s}$ also has three invariants termed *deviatoric stress invariants*

$$\begin{aligned} J_1 &= s_{kk} = 0, \\ J_2 &= \frac{1}{3}I_1^2 - I_2 \\ &= \frac{1}{6}\left[(\sigma_{11}-\sigma_{22})^2 + (\sigma_{22}-\sigma_{33})^2 + (\sigma_{33}-\sigma_{11})^2\right] + \sigma_{12}^2 + \sigma_{23}^2 + \sigma_{31}^2, \\ J_3 &= \det(\boldsymbol{s}) \end{aligned} \tag{3.15}$$

Suppose that the position of a body is represented as a vector $\boldsymbol{r}$ in its original shape and $\boldsymbol{r}'$ after deformation, then the *displacement* due to the deformation is

$$\boldsymbol{u} = \boldsymbol{r}' - \boldsymbol{r},$$

whose individual components are $u_i = x_i' - x_i$. Let us consider the distance between two close points. Before the deformation, the distance is (summation rule applied)

$$\mathrm{d}l^2 = \mathrm{d}x_i^2$$

whereas after deformation the distance is

$$\mathrm{d}l'^2 = (\,\mathrm{d}u_i + \,\mathrm{d}x_i)^2.$$

Substituting with $\mathrm{d}u_i = \dfrac{\partial u_i}{\partial x_k}\,\mathrm{d}x_k$, we have

$$\begin{aligned} \mathrm{d}l'^2 &= \mathrm{d}l^2 + 2\frac{\partial u_i}{\partial x_k}\,\mathrm{d}x_k\,\mathrm{d}x_i + \frac{\partial u_i}{\partial x_k}\,\mathrm{d}x_k\frac{\partial u_i}{\partial x_l}\,\mathrm{d}x_l \\ &= \mathrm{d}l^2 + \left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i}\right)\mathrm{d}x_k\,\mathrm{d}x_i + \frac{\partial u_l}{\partial x_k}\frac{\partial u_l}{\partial x_i}\,\mathrm{d}x_k\,\mathrm{d}x_i. \end{aligned} \tag{3.16}$$

So $\mathrm{d}l'^2$ takes the final form

$$\mathrm{d}l'^2 = \mathrm{d}l^2 + 2\varepsilon_{ik}\,\mathrm{d}x_i\,\mathrm{d}x_k, \tag{3.17}$$

where the *strain tensor* $\boldsymbol{\varepsilon}$ is defined as

$$\varepsilon_{ik} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i} + \frac{\partial u_l}{\partial x_k}\frac{\partial u_l}{\partial x_i}\right). \tag{3.18}$$

In the case of the small deformation, u_l is small and the third term in equation (3.18) can be neglected as it is of second order. So for small deformation the strain tensor is given by

$$\varepsilon_{ik} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i}\right). \tag{3.19}$$

From equation (3.19), the symmetry of the strain tensor is implied by

$$\varepsilon_{ik} = \varepsilon_{ki}. \tag{3.20}$$

Similar to the stress tensor, the strain tensor $\boldsymbol{\varepsilon}$ defined with respect to Cartesian coordinates can be rewritten as

$$\boldsymbol{\varepsilon} = \varepsilon_{ij}\boldsymbol{e}_i \otimes \boldsymbol{e}_j, \tag{3.21}$$

whose components in Cartesian coordinate in a matrix form are

$$[\varepsilon_{ij}] = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix}. \tag{3.22}$$

Because of the symmetry, the strain tensor has six independent components to be determined. However, an arbitrary symmetric second order tensor does not necessarily make it a valid strain tensor since the strain ε_{ik} has to be determined by the admissible displacement $\boldsymbol{u}$. Thus, the strain tensor has to satisfy the *compatibility conditions* or *integrable conditions* (cf equation (3.19)). The derivation for the compatibility condition can be found in [59] and the compatibility equations are

$$\varepsilon_{im,jl} + \varepsilon_{jl,lm} - \varepsilon_{il,jm} - \varepsilon_{jm,il} = 0, \tag{3.23}$$

where $(\cdot)_{,i} = \partial(\cdot)/\partial x_i$. For example,

$$\varepsilon_{ij,kl} = \frac{\partial^2 \varepsilon_{ij}}{\partial x_k \partial x_l}.$$

Similar to the stress tensor decomposition outlined in (3.11), we can split the strain tensor $\boldsymbol{\varepsilon}$ into two parts, namely, *deviatoric* and *volumetric* (*mean*) strains

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_d + \frac{1}{3}\varepsilon_v \boldsymbol{I}, \tag{3.24}$$

where

$$\boldsymbol{\varepsilon}_d = \mathsf{I_d}\boldsymbol{\varepsilon}, \quad \varepsilon_v = \operatorname{trace}(\boldsymbol{\varepsilon}) = \boldsymbol{I} : \boldsymbol{\varepsilon}.$$

The stress-strain relation in continuum mechanics is described by *constitutive equations*, which describes the material properties.

It is more convenient to work with symmetric tensor in its *Voigt form* (or *Voigt notation*), which is a vector form to reduce the order to save computational resources. For example, the Voigt notation of stress tensor is given as[2]

$$\tilde{\boldsymbol{\sigma}} = \begin{bmatrix} \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 & \sigma_5 & \sigma_6 \end{bmatrix}^T \equiv \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{33} & \sigma_{12} & \sigma_{23} & \sigma_{13} \end{bmatrix}^T. \tag{3.25}$$

3.2 Elasto-Plastic Material Behaviour Modelling

In this section, we study the ISF deformation process in light of the elasto-plastic material behaviour. First we review the elastic theory before introducing the rate-dependent

[2]The standard Voigt form is given in a slightly different order as

$$[\boldsymbol{\sigma}] = \begin{bmatrix} \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 & \sigma_5 & \sigma_6 \end{bmatrix}^T \equiv \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{33} & \sigma_{23} & \sigma_{31} & \sigma_{12} \end{bmatrix}^T.$$

plasticity theory that relates to the metal forming process, and then we complete the elasto-plastic analysis for the ISF process by deriving a corresponding model.

3.2.1 Elastic Deformation Theory

The elastic theory in the current work is limited to a linear elasticbody where the stress and strain satisfy the following equation

$$\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{\varepsilon}, \tag{3.26}$$

where $\boldsymbol{C}$, called the *elasticity tensor*, is a fourth order tensor. $\boldsymbol{C}$ is a linear map from the space of the second order tensors into itself. If the density of the material and the elasticity tensor $\boldsymbol{C}$ are independent of position, the material is said to be *homogeneous*. Using summation rule, the component form in Cartesian coordinate can be expressed as

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl}. \tag{3.27}$$

The symmetry of the strain tensor $\boldsymbol{\varepsilon}$ implies:

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl} = C_{ijlk}\varepsilon_{lk},$$

i.e,

$$C_{ijkl} = C_{ijlk},$$

Hence

$$\sigma_{ij} = \frac{1}{2}\left(C_{ijkl} + C_{ijlk}\right)\varepsilon_{kl}.$$

Similarly, the symmetry of the stress tensor $\boldsymbol{\sigma}$ implies

$$\sigma_{ji} = \frac{1}{2}\left(C_{jikl} + C_{jikl}\right)\varepsilon_{kl} = \sigma_{ij}.$$

Therefore, the equation (3.27) can be expressed as

$$\sigma_{ij} = \frac{1}{4}\left(C_{ijkl} + C_{ijlk} + C_{jikl} + C_{jilk}\right)\varepsilon_{kl}.$$

So the symmetry properties

$$C_{ijkl} = C_{jikl} = C_{ijlk} \tag{3.28}$$

hold. From (3.28), it follows that $\boldsymbol{C}$ has only 36 independent components. Further reduction of the number of the independent components can be made by taking a thermodynamic perspective, where a free energy function is used to derive the elastic constitutive equation [59]. It is seen that the elastic tensor possesses the additional symmetry property

$$C_{ijkl} = C_{klij}. \tag{3.29}$$

Written in a compact matrix form, $\boldsymbol{C}$ becomes

$$[C_{ijkl}] = \begin{bmatrix} C_{1111} & C_{1122} & C_{1133} & C_{1112} & C_{1123} & C_{1113} \\ C_{2211} & C_{2222} & C_{2233} & C_{2212} & C_{2223} & C_{2213} \\ C_{3311} & C_{3322} & C_{3333} & C_{3312} & C_{3323} & C_{3313} \\ C_{1211} & C_{1222} & C_{1233} & C_{1212} & C_{1223} & C_{1213} \\ C_{2311} & C_{2322} & C_{2333} & C_{2312} & C_{2323} & C_{2313} \\ C_{1311} & C_{1322} & C_{1333} & C_{1312} & C_{1323} & C_{1313} \end{bmatrix}. \tag{3.30}$$

It becomes clear that the number of independent components reduce to 21 constants.[3] If we are limited to considering an *isotropic material*, which has the same elastic properties in all directions, then two independent elastic constants are sufficient to determine the isotropic elasticity tensor as

$$[C] = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}, \tag{3.31}$$

where λ and μ are called *Lamé* parameters. In component form, $\boldsymbol{C}$ can be expressed as

$$C_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}), \tag{3.32}$$

where δ_{ij}, as usual, denotes the Kronecker delta. Reformulate equation (3.32) into

$$C_{ijkl} = (\lambda + \frac{2}{3}\mu)\delta_{ij}\delta_{kl} + 2\mu\left[\frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) - \frac{1}{3}\delta_{ij}\delta_{kl}\right],$$

and let

$$K = \lambda + \frac{2}{3}\mu,$$

with definitions in (A.11), (A.12) and (A.13), we can write the isotropic elasticity tensor $\boldsymbol{C}$ in tensorial form (coordinate-independent) as

$$\boldsymbol{C} = 2\mu\mathsf{I}_\mathsf{d} + K(\boldsymbol{I}\otimes\boldsymbol{I}), \tag{3.33}$$

where K is called the *bulk modulus*, I_d is the fourth-order deviatoric projection tensor and $\boldsymbol{I}$ is the second-order identity tensor. Sometimes μ is replaced by an equivalent parameter G, the *shear modulus*.

[3] [59] mentions the long controversy about this number of independent components of $\boldsymbol{C}$. Poission and Cauchy claimed this number to be 15, based on a simplified molecular model of an elastic body. G. Green obtained the number 21. Much later, M. Born proved this number should be 21, using modern results of molecular theory.

For computational convenience, we may want to invert the component matrix, resulting

$$[C_{ij}]^{-1} = \frac{1}{\mu(3\lambda+2\mu)} \begin{bmatrix} \lambda+\mu & -\lambda/2 & -\lambda/2 & 0 & 0 & 0 \\ -\lambda/2 & \lambda+\mu & -\lambda/2 & 0 & 0 & 0 \\ -\lambda/2 & -\lambda/2 & \lambda+\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 3\lambda+2\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 3\lambda+2\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 3\lambda+2\mu \end{bmatrix}. \tag{3.34}$$

Young's modulus E in equation (1.2) and Poisson's ratio ν are related to the *Lamé* parameters in the following manner

$$E = \frac{\mu(3\lambda+2\mu)}{\lambda+\mu}, \quad \nu = \frac{\lambda}{2(\lambda+\mu)}. \tag{3.35}$$

With the elasticity tensor expressed in form (3.33), we can establish the relations between decomposed stress and strain components defined in (3.11) and (3.24) as

$$\boldsymbol{\sigma} = 2\mu \mathsf{I}_{\mathsf{d}}\boldsymbol{\varepsilon} + K(\boldsymbol{I}\otimes\boldsymbol{I})\boldsymbol{\varepsilon} = 2\mu\boldsymbol{\varepsilon}_d + 3K\varepsilon_v\boldsymbol{I},$$

where (A.16) is applied, and

$$\boldsymbol{\sigma} = \boldsymbol{s} + p\boldsymbol{I}.$$

Therefore, in the elastic range, we obtain

$$\boldsymbol{s} = 2\mu\boldsymbol{\varepsilon}_d, \quad p = 3K\varepsilon_v. \tag{3.36}$$

3.2.2 Plasticity Theory Foundations

The nonlinearity of the plastic deformation behaviour makes the study of the constitutive relations more complex than that of the elastic theory. The plastic deformation behaviour of a metal material is regarded as *rate-independent* in the sense that the rate at which metal deformation takes place is slow, so it can be regarded as a *quasi-static* process. The plasticity theory applied here is thus a rate-independent one.

In this section, we first introduce the rate-independent plasticity theory before going further with detailed modelling. In accordance with the one-dimensional elasto-plastic behaviour described in Chapter 1, additive decomposition of the total strain is assumed. Recall equation (1.3)

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p,$$

where $\boldsymbol{\varepsilon}^e$ denotes elastic strain and $\boldsymbol{\varepsilon}^p$ denotes irreversible plastic strain. The elastic strain is governed by the elastic law outlined in section 3.2.1, whereas plastic strain behaviour is modelled with *plastic flow rule*, which will be elaborated later.

Second, we make the *plastic incompressibility assumption*, which states that the volume of a body does not change in a plastic deformation [60]. As a result, the plastic strain rate

has to satisfy

$$\dot{\varepsilon}^p_{kk} = 0. \tag{3.37}$$

As we have pointed out earlier in the tensile test illustration, yielding starts when the stress reaches a threshold value and the state of the material then moves from elastic region to plastic region. An accurate account of the material yield behaviour can be expressed by postulating a yield function $\phi(\boldsymbol{\sigma})$

$$\begin{cases} \phi < 0, & \text{elastic} \\ \phi \geq 0, & \text{plastic} \end{cases} \tag{3.38}$$

where $\phi(\boldsymbol{\sigma}) = 0$ defines a yield surface, which is the boundary between elastic deformation region and plastic region.

The most significant postulate in plasticity for hardening material and perfect plastic material (softening material has a different expression) is **Drucker's Inequality**,

$$\dot{\sigma}_{ij}\dot{\varepsilon}^p_{ij} \geq 0. \tag{3.39}$$

This inequality is unchanged when multiplied by an infinitesimal time increment $\mathrm{d}t$, thus the inequality is a declaration about the work done in the course of incremental loading. The elastic work $\mathrm{d}\sigma_{ij}\,\mathrm{d}\varepsilon^e_{ij}$ is always positive, so the Drucker's inequality implies positive work done during the incremental loading and nonnegative work done during a full cycle of loading and unloading. Consider the cycle where we start with the initial stress state denoted as $\boldsymbol{\sigma}^*$, increase the load to the point where stress $\boldsymbol{\sigma}$ is at yield surface, the incremental stress $\mathrm{d}\boldsymbol{\sigma}$ producing an incremental strain field $\mathrm{d}\boldsymbol{\varepsilon}$, and finally elastic unload to the initial stress $\boldsymbol{\sigma}^*$. The work done by the external loading during this cycle is nonnegative, which can be expressed as

$$(\boldsymbol{\sigma} - \boldsymbol{\sigma}^*) : \dot{\boldsymbol{\varepsilon}}^p \geq 0. \tag{3.40}$$

The above implication from Drucker's postulate constitutes the *postulate of maximum dissipation* in its own right [16]. The plastic dissipation denoted as D_p, dependent on strain (and strain-like) variables, has the following property

$$D_p(\boldsymbol{\varepsilon}) = \max_{\boldsymbol{\sigma}^*} \boldsymbol{\sigma}^* \dot{\boldsymbol{\varepsilon}}^p. \tag{3.41}$$

Two consequences of this postulate are the *normality rule* and the *convexity* of the yield surface, which can be best illustrated in Figure 3.2. We now only consider the case where the yield surface is smooth, which indicates existence of a tangent plane everywhere on the yield surface. Given a stress state $\boldsymbol{\sigma}$ at the yield surface, inequality (3.40) has to hold for every possible initial stress $\boldsymbol{\sigma}$ within the elastic region, so plastic strain rate must directed along the outward normal at point $\boldsymbol{\sigma}$ shown in Figure 3.2 (a); this is known as the normality rule. Figure 3.2 (b) shows that the inequality is violated if any $\boldsymbol{\sigma}^*$ lies to the outward side of the tangent. As a result, the yield surface is convex.

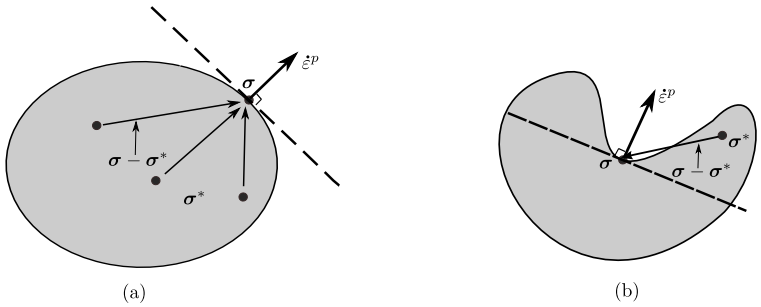


Figure 3.2 Properties of yield surface: convexity and normality. Shadow regions indicate elastic region, whilst outside region is the plastic region. The yield surface is the boundary. (a) shows a convex yield surface; (b) shows a non-convex yield surface violating the Drucker's inequality.

To account for the plastic reconstruction of internal structure, extra variables in addition to stress $\boldsymbol{\sigma}$ and strain $\boldsymbol{\varepsilon}$, called *internal variables* have to be introduced. Depending on the materials studied, the internal variables may take different forms. Essentially, the internal variables are of two kinds: kinematic variables and the force-like variables [61]. In the metal forming modelling, the plastic strain tensor $\boldsymbol{\varepsilon}^p$ and the hardening variable κ, or the *accumulated plastic strain* defined as

$$\kappa = \int \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^p : \dot{\boldsymbol{\varepsilon}}^p}\, \mathrm{d}t, \tag{3.42}$$

serve as the kinematic variables. An additional *hardening thermodynamic force variable* denoted by $\boldsymbol{\chi}$ is needed to describe the evolution of the hardening variable κ in the plastic theory.

3.2.3 Von Mises Yield Criterion

Various yield criteria with different critical values have been proposed, such as Tresca yield criterion, which assumes that plastic yielding begins when the maximum shear stress reaches a critical value. We choose the classic von Mises yield criterion to model the metal yield condition as it is less computational intensive than that of Tresca's. J_2 plasticity defined in (3.15) is used in the von Mises yield function

$$\phi(\boldsymbol{\sigma}) = J_2 - k^2, \tag{3.43}$$

where k is the yield stress of the material in pure shear and the J_2 stress deviator invariant can be expressed in terms of the stress deviator $\boldsymbol{s}$ as

$$J_2 = J_2\left(\boldsymbol{s}(\boldsymbol{\sigma})\right) = \frac{1}{2}\boldsymbol{s} : \boldsymbol{s} = \frac{1}{2}\left\|\boldsymbol{s}\right\|^2. \tag{3.44}$$

3.2.4 Flow Rule

One central question around plastic deformation is what happens after initial yielding, which is described by the theory of plastic flow rule. The plastic flow rule claims that there exists a *flow potential function* Ψ such that the following plastic rate equation holds

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\gamma}\frac{\partial \Psi}{\partial \boldsymbol{\sigma}}, \tag{3.45}$$

where γ is the *plastic multiplier*. In case of the metal material, an *associative law* (associated with the yield criterion) is used, where the potential function is the yield function, so that

$$\Psi = \phi,$$

and

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\gamma}\frac{\partial \phi}{\partial \boldsymbol{\sigma}}. \tag{3.46}$$

The plastic multiplier in the flow rule is obtained by solving the so-called *complementary conditions* or *load/unloading conditions*[1]

$$\begin{aligned} \dot{\gamma} &\geq 0, \\ \phi(\boldsymbol{\sigma}) &\leq 0, \\ \dot{\gamma}\phi(\boldsymbol{\sigma}) &= 0 \end{aligned} \tag{3.47}$$

and the *consistency condition* when $\phi = 0$

$$\begin{aligned} \dot{\gamma} &\geq 0, \\ \dot{\phi}(\boldsymbol{\sigma}) &\leq 0, \\ \dot{\gamma}\dot{\phi}(\boldsymbol{\sigma}) &= 0. \end{aligned} \tag{3.48}$$

We can see $\dot{\gamma} = 0$ if $\phi < 0$ or otherwise $\dot{\gamma} > 0$, if $\phi = 0$ and $\dot{\phi} < 0$. The loading and unloading condition (3.47) tells us that the plastic multiplier vanishes if only elastic deformation is involved, but may take any non-negative values during the plastic flow; consistency condition (3.48) complementarily tells us that during the plastic flow, the yield function remains the same (zero) and thus the rate of change for the yield function is zero. These two conditions are sufficient to capture the behaviour after yielding and to determine the plastic multiplier.

In all, $\dot{\gamma}$ only needs to be determined when $\phi = \dot{\phi} = 0$, and the exact form of $\dot{\gamma}$ is given in the section 3.2.6 after the hardening rule has been established.

[1]The resemblance of an optimisation problem of the plasticity multiplier γ from (3.46) is that it is a direct consequence of maximum dissipation posostulate, which is a minimisation formulation (cf. (3.41)).

3.2.5 Hardening Rules

In this section, we first introduce several hardening models before formulating a general hardening rule. As depicted in Figure 1.10, perfect plasticity is characterised by no hardening behaviour. Hardening occurs when the stress required to cause further plastic deformation increases, which means that the yield surface changes from the initial yield point. If the change of the yield surface is an expansion that is uniform in all directions, this kind of hardening is referred as *isotropic hardening*. Isotropic hardening is suitable for explaining the hardening behaviour under monotonically increasing load, but is not appropriate for reversed load. Instead *kinematic hardening*, where the yield surface preserves the shape and size of the original yield surface but translated in the stress space, can be used to account for the effect of the decreased resistance to plastic yielding in the opposite loading direction (known as *Bauschinger effect* [60]). These two types of hardening can be illustrated in a two dimensional case shown in Figure 3.3, where the shaded area shows the original yield surface at the onset of yielding.

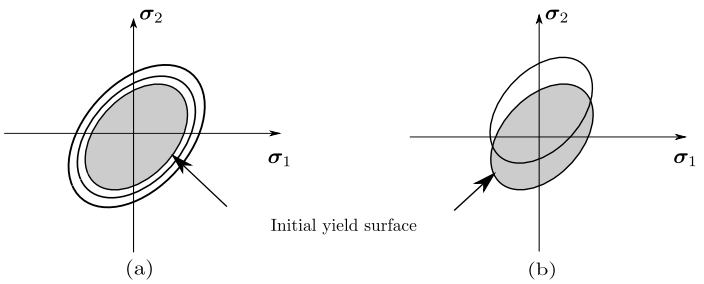


Figure 3.3 Illustration of two dimensional hardening behaviours: (a) isotropic hardening (b) kinematic hardening. Shadow area shows the original yield surface at the onset of yielding.

Hardening rules specify the dependence of the yield function ϕ on the internal variables and a hardening material implies an expanding yield surface. Thus the modelling of metal hardening reduces to choosing the appropriate internal variables as the yield stress is only affected by the internal variables. A general simple model [16] that we may adopt here is to take plastic strain $\boldsymbol{\varepsilon}^p$ and the hardening variable κ defined in (3.42) as parameters of our yield function, which is expressed as

$$\phi(\boldsymbol{\sigma}, \boldsymbol{\varepsilon}^p, \kappa) = F(\boldsymbol{\sigma} - \boldsymbol{\rho}(\boldsymbol{\varepsilon}^p)) - k(\kappa), \tag{3.49}$$

where $\boldsymbol{\rho}$ is a tensor function of plastic strain $\boldsymbol{\varepsilon}^p$. Both isotropic and kinematic hardening described by this model and the classification depending on the form of the $\boldsymbol{\rho}$ and k can be listed as

$$\begin{cases} \text{if } \boldsymbol{\rho} \equiv \mathbf{0} \text{ and } \frac{\mathrm{d}k}{\mathrm{d}\kappa} > 0, & \text{isotropic hardening;} \\ \text{if } \boldsymbol{\rho} \neq \mathbf{0} \text{ and } \frac{\mathrm{d}k}{\mathrm{d}\kappa} \equiv 0, & \text{kinematic hardening;} \\ \text{if } \boldsymbol{\rho} \equiv \mathbf{0} \text{ and } \frac{\mathrm{d}k}{\mathrm{d}\kappa} \equiv 0, & \text{perfect plasticity.} \end{cases} \tag{3.50}$$

As the von Mises yield criterion is used, the function F is the J_2 plasticity, and if *von*

Mises effective stress defined as $\sqrt{3J_2}$ is used, the yield function for a perfect plasticity can be described as

$$\phi(\boldsymbol{\sigma}) = \sqrt{3J_2\left(\boldsymbol{s}(\boldsymbol{\sigma})\right)} - \sigma_y. \tag{3.51}$$

where σ_y is a constant, namely the initial uniaxial yield stress. For isotropic hardening, the derivative $\dfrac{\mathrm{d}k(\kappa)}{\mathrm{d}\kappa}$ is non-zero and can be defined with a hardening modulus H, so

$$H = \frac{\mathrm{d}k(\kappa)}{\mathrm{d}\kappa}.$$

A *linear hardening rule* can be established as

$$k(\kappa) = \sigma_y + H\kappa. \tag{3.52}$$

So the linear isotropic hardening rule for a von Mises criterion has the following yield function

$$\phi(\boldsymbol{\sigma}, \kappa) = \sqrt{3J_2\left(\boldsymbol{s}(\boldsymbol{\sigma})\right)} - \sigma_y - H\kappa. \tag{3.53}$$

Similar to flow rule, the hardening evolution equation for the internal variable κ by associative law is

$$\dot{\kappa} = -\dot{\gamma}\frac{\partial \phi}{\partial \boldsymbol{\chi}}. \tag{3.54}$$

The hardening thermodynamic force variables $\boldsymbol{\chi}$ in this case is

$$\boldsymbol{\chi} \equiv k,$$

which results in Prandtl-Reuss flow equation [62]

$$\dot{\kappa} = -\dot{\gamma}\frac{\partial \phi}{\partial k} = \dot{\gamma}. \tag{3.55}$$

From equation (3.42), the rate equation for $\dot{\kappa}$ can be expressed alternatively as

$$\dot{\kappa} = \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^p : \dot{\boldsymbol{\varepsilon}}^p} = \sqrt{\frac{2}{3}}\left\|\dot{\boldsymbol{\varepsilon}}^p\right\|. \tag{3.56}$$

In the ISF process, because no reverse loading is involved, we will not include kinematic hardening modelling.

3.2.6 Constitutive Elasto-Plastic Deformation Model

In this section, we summarise the basic components of the plasticity theory derived so far and provide a general constitutive elasto-plastic deformation model for the ISF process.

As given in (3.50), for both perfect plasticity and linear isotropic hardening models with von Mises yield criterion, $\boldsymbol{\rho} = \mathbf{0}$, so a general yield function with von Mises effective

stress takes the form

$$\begin{aligned}\phi(\boldsymbol{\sigma},\kappa) &= F(\boldsymbol{\sigma}) - k(\kappa),\\ &= \sqrt{3J_2\left(\boldsymbol{s}(\boldsymbol{\sigma})\right)} - \left(\sigma_y + H\kappa\right).\end{aligned} \tag{3.57}$$

If $H = 0$, the yield function represents the perfect plasticity; otherwise this expression represents a linear isotropic hardening model. Substituting the yield function (3.57) into the flow rule defined in (3.46), we obtain

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\gamma}\frac{\partial}{\partial\boldsymbol{\sigma}}\sqrt{3J_2\left(\boldsymbol{s}(\boldsymbol{\sigma})\right)} = \sqrt{\frac{3}{2}}\dot{\gamma}\frac{\boldsymbol{s}}{\|\boldsymbol{s}\|}. \tag{3.58}$$

On the other hand, we may write the strain decomposition equation (1.3) in the following rate form

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^e + \dot{\boldsymbol{\varepsilon}}^p \tag{3.59}$$

Using the elastic law, yields

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{C}\left(\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}^p\right) = \boldsymbol{C}\left(\dot{\boldsymbol{\varepsilon}} - \dot{\gamma}\frac{\partial\phi}{\partial\boldsymbol{\sigma}}\right) \tag{3.60}$$

Recalling the consistency condition defined in (3.48)

$$\dot{\phi}(\boldsymbol{\sigma},\kappa) = \frac{\partial\phi}{\partial\boldsymbol{\sigma}} : \dot{\boldsymbol{\sigma}} + \frac{\partial\phi}{\partial\kappa}\dot{\kappa} = 0. \tag{3.61}$$

Substituting with the hardening law (3.55) and the rate form equation (3.60), the plastic multiplier $\dot{\gamma}$ can be deduced from equation (3.61) as

$$\dot{\gamma} = \frac{\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} : \boldsymbol{C}\dot{\boldsymbol{\varepsilon}}}{\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} : \boldsymbol{C}\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} - \dfrac{\partial\phi}{\partial\kappa}} \tag{3.62}$$

If we substitute the exact form of $\dot{\gamma}$ into equation (3.60), we can obtain a rate form

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{C}\left(\dot{\boldsymbol{\varepsilon}} - \frac{\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} : \boldsymbol{C}\dot{\boldsymbol{\varepsilon}}}{\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} : \boldsymbol{C}\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} - \dfrac{\partial\phi}{\partial\kappa}}\frac{\partial\phi}{\partial\boldsymbol{\sigma}}\right) = \boldsymbol{C}\dot{\boldsymbol{\varepsilon}} - \boldsymbol{C}\frac{\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}}\left(\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} : \boldsymbol{C}\dot{\boldsymbol{\varepsilon}}\right)}{\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} : \boldsymbol{C}\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} - \dfrac{\partial\phi}{\partial\kappa}}. \tag{3.63}$$

The symmetry property of the fourth-order tensor $\boldsymbol{C}$ implies

$$\frac{\partial\phi}{\partial\boldsymbol{\sigma}} : \boldsymbol{C}\dot{\boldsymbol{\varepsilon}} = \boldsymbol{C}\frac{\partial\phi}{\partial\boldsymbol{\sigma}} : \dot{\boldsymbol{\varepsilon}}. \tag{3.64}$$

Therefore, if we modify the numerator of the second term in (3.63) using (3.64), we can

separate $\dot{\boldsymbol{\varepsilon}}$ out by applying the relation given in (A.9), to give

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{C}\dot{\boldsymbol{\varepsilon}} - \frac{\boldsymbol{C}\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} \otimes \boldsymbol{C}\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}}}{\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} : \boldsymbol{C}\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} - \dfrac{\partial\phi}{\partial\kappa}}\dot{\boldsymbol{\varepsilon}}. \tag{3.65}$$

Note that the above expression is valid only when $\dot{\gamma} > 0$. In general, we have the stress-strain relation for both elastic and plastic deformation

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{C}^{ep}\dot{\boldsymbol{\varepsilon}}, \tag{3.66}$$

where $\boldsymbol{C}^{ep}$ ("ep" indicates elasto-plasticity) is defined as

$$\boldsymbol{C}^{ep} = \begin{cases} \boldsymbol{C} & \text{if } \dot{\gamma} = 0, \\ \boldsymbol{C} - \dfrac{\boldsymbol{C}\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} \otimes \boldsymbol{C}\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}}}{\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} : \boldsymbol{C}\dfrac{\partial\phi}{\partial\boldsymbol{\sigma}} - \dfrac{\partial\phi}{\partial\kappa}} & \text{if } \dot{\gamma} > 0. \end{cases} \tag{3.67}$$

We shall refer to (3.67) as the *continuum tangent moduli* (operator). Since

$$\frac{\partial\phi}{\partial\boldsymbol{\sigma}} = \sqrt{\frac{3}{2}}\frac{\boldsymbol{s}}{\|\boldsymbol{s}\|}$$

is a deviatoric tensor, we may define

$$\hat{\boldsymbol{n}} = \sqrt{\frac{2}{3}}\frac{\partial\phi}{\partial\boldsymbol{\sigma}} = \frac{\boldsymbol{s}}{\|\boldsymbol{s}\|} \tag{3.68}$$

whose trace is zero, i.e.

$$\text{trace}(\hat{\boldsymbol{n}}) = 0 \quad \text{ and } \quad \mathsf{I}_{\mathsf{d}}\hat{\boldsymbol{n}} = \hat{\boldsymbol{n}},$$

where I_{d} is defined in (A.13). Together with elasticity tensor described in form (3.33) and property (A.16), it follows that

$$\boldsymbol{C}\frac{\partial\phi}{\partial\boldsymbol{\sigma}} = (2\mu\mathsf{I}_{\mathsf{d}} + K\boldsymbol{I}\otimes\boldsymbol{I})\sqrt{3/2}\hat{\boldsymbol{n}} = \sqrt{3/2}\,(2\mu\hat{\boldsymbol{n}} + K\text{trace}(\hat{\boldsymbol{n}})\boldsymbol{I}) = 2\sqrt{3/2}\mu\hat{\boldsymbol{n}},$$

and

$$\frac{\partial\phi}{\partial\boldsymbol{\sigma}} : \boldsymbol{C}\frac{\partial\phi}{\partial\boldsymbol{\sigma}} = 3\mu\hat{\boldsymbol{n}} : \hat{\boldsymbol{n}} = 3\mu.$$

Hence, when $\dot{\gamma} > 0$, $\boldsymbol{C}^{ep}$ can be simplified as

$$\boldsymbol{C}^{ep} = \boldsymbol{C} - \frac{6\mu^2}{3\mu + H}\hat{\boldsymbol{n}}\otimes\hat{\boldsymbol{n}}. \tag{3.69}$$

So the continuum tangent moduli (the elasto-plasticity tensor) in (3.67) can be written as

$$\boldsymbol{C}^{ep} = \begin{cases} \boldsymbol{C} & \text{if } \dot{\gamma} = 0, \\ \boldsymbol{C} - \dfrac{6\mu^2}{3\mu + H}\hat{\boldsymbol{n}} \otimes \hat{\boldsymbol{n}} & \text{if } \dot{\gamma} > 0. \end{cases} \tag{3.70}$$

Note that the plastic multiplier expressed in (3.62) does not reveal the actual cause-effect relations as one would expect the plastic multiplier to be a result of stress changes, however, expression (3.62) does allow us to derive the general rate constitutive model given in (3.66) and also serves as a displacement based computational method. Nevertheless, we can get a form that is more physically straightforward by using the inverse relationship between stress and strain, as described in [63].

3.3 ISF Process Modelling: A Contact Problem

In the ISF process, the force exerted onto the metal sheet occurs through the contact between the sheet and the rigid tool, the modelling of which eventually falls within the subject of contact mechanics. We shall derive a mathematical model to describe the deformation after each incremental step. The classical formulation of the problem is given in this chapter, whereas a weak (or variational) form suitable for numerical solution will be derived in Chapter 4.

3.3.1 Contact Problem: From Hertz To Signorini

The modern contact problem is believed to be first studied by Hertz (Figure 3.4), where two elastic solid bodies (depicted in Figure 3.5) come in contact under external force F. The problem is to solve the amount of deformation for two bodies, which are u_z and u'_z respectively, given known relatively body movements h. Figure 3.5 (a) shows that the two surfaces near the contact point O share a common tangent plane. We take this tangent plane as the $xy-$plane. Kinematically, within the region of contact (shown in 3.5 (b)), we have

$$(z + u_z) + (z' + u'_z) = h, \tag{3.71}$$

and outside this region,

$$z + u_z + z' + u'_z < h.$$

Note that z and z' represent the equation of the surface, which can be expressed as a quadratic function. We may rewrite (3.71) as

$$Ax^2 + By^2 + u_z + u'_z = h, \tag{3.72}$$

where A and B are known quantities related to the radii of the surface curvatures. Next, we may introduce the normal pressure $P_z(x, y)$ between the two deformed bodies in the contact region. From equilibrium equations of the elastic body subject to a plane, we

Figure 3.4 German physicist Heinrich R. Hertz (22 February 1857 - 1 January 1894) [64]

obtain,

$$\begin{aligned} u_z &= \frac{1-\nu^2}{\pi E} \iint \frac{P_z(x',y')}{r}\,\mathrm{d}x'\,\mathrm{d}y' \\ u'_z &= \frac{1-\nu'^2}{\pi E'} \iint \frac{P_z(x',y')}{r}\,\mathrm{d}x'\,\mathrm{d}y' \end{aligned}, \tag{3.73}$$

where ν, ν' and E, E' are the Poisson's ratio and Young's moduli of the two elastic bodies respectively, and $r = \sqrt{x^2+y^2+z^2}$. Substituting (3.73) into (3.72), gives an integral equation

$$\frac{1}{\pi}\left(\frac{1-\nu^2}{E} + \frac{1-\nu'^2}{E'}\right) \iint \frac{P_z(x',y')}{r}\,\mathrm{d}x'\,\mathrm{d}y' = h - Ax^2 - By^2. \tag{3.74}$$

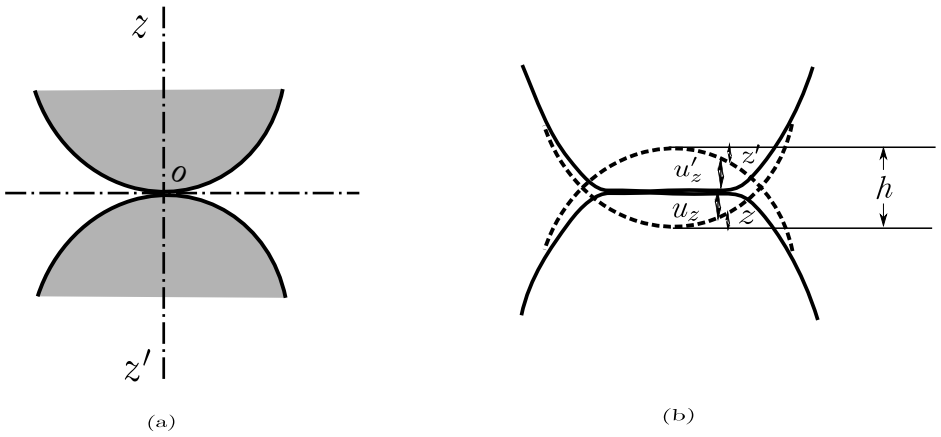


Figure 3.5 Illustration of two solid bodies contact problem: (a) initial state (b) after contact

Hertz solved this equation by using an analogy with the potential theory of electricity, of which a detailed modern treatment can be found in [58]. The region of contact turns

out to be bounded by an *ellipse* of the form

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1,$$

and the pressure function is of the form

$$P_z(x, y) = \text{constant} \times \sqrt{\left(1 - \frac{x^2}{a^2} - \frac{y^2}{b^2}\right)}.$$

The integral of pressure over the region of contact has to equal to the exerted force F, so that an exact form of $P_z(x, y)$ can be obtained. Therefore, the exact deformation can be determined by using (3.73).

In the case of two spheres in contact where h is the sum of the two radii of the spheres (or the distance between the centres of the spheres), the resulting region of contact is a circle, and F is proportional to $h^{3/2}$. We have to stress that Hertz's solution assumes the contact region is a plane, which is not always generally applicable for contact problem.

The next important advancement is from the Italian mathematician Antonio Signorini [65, 66], where the problem was formulated under the term "ambiguous boundary value problem". The well-posedness (existence and uniqueness) of Signorini's formulation was answered by fellow Italian Fichera [67]. The general mathematical theory of contact mechanics starts to take off from Duvant and Lions's work [68], followed by Necas [69] and many others. A good bibliography is available in Shillor et al's monograph (pp. 2-4, [70]).

There are many complex concepts in the contact problem, such as dynamic contact, friction, crack development, but we shall only focus on the frictionless contact in this work. Indeed the difference between *frictionless* or *frictional* contact assumption, resides in whether the tangential force in the contact region is regarded as zero, which shall be explained below. Friction is universal and exists even in the case of perfect lubrication, but the frictionless assumption allows a simpler formulation and quite often accurate enough for the underlying problem.

Let us consider Signorini's formulation for two linear elastic bodies in contact, with one of them a rigid foundation (depicted in Figure 3.6). As the contact process takes place at the surface of the body, so this type of problem is described by boundary conditions, as well as the body equilibrium equation. We shall distinguish the different sources of constraints/forces exerted on the boundary Γ (assumed to be Lipschitz) of the body. As the body is held fixed on Γ_u, we denote this homogeneous Dirichlet boundary condition as

$$\boldsymbol{u} = \boldsymbol{0}, \quad \text{on } \Gamma_u. \tag{3.75}$$

On the top boundary of Figure 3.6, the traction $\boldsymbol{f}_N$ on Γ_σ gives a Neumann condition, which can be expressed as

$$\boldsymbol{\sigma n} = \boldsymbol{f}_N, \quad \text{on } \Gamma_\sigma, \tag{3.76}$$

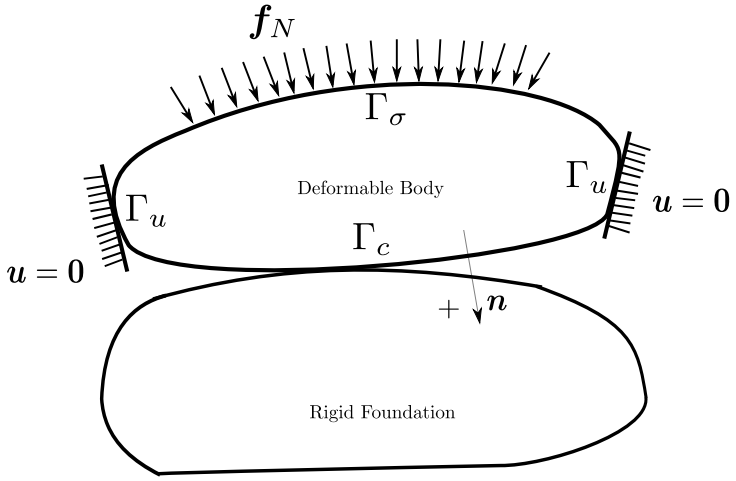


Figure 3.6 Illustration of two body contact problem with rigid foundation; Boundary is split into Dirichlet type Γ_u, Neumann type Γ_σ and contact boundary Γ_c.

where $\boldsymbol{\sigma}$ denotes the stress tensor whose normal and tangential components, denoted by σ_ν and $\boldsymbol{\sigma}_\tau$, are given by

$$\begin{aligned} \sigma_\nu &= (\boldsymbol{\sigma}\boldsymbol{n})\cdot\boldsymbol{n}, \\ \boldsymbol{\sigma}_\tau &= \boldsymbol{\sigma}\boldsymbol{n} - \sigma_\nu\boldsymbol{n}. \end{aligned} \tag{3.77}$$

From Hertz's solution it can be deduced that the region of contact is the most tricky part to work out. Indeed, one of those difficulties in a general contact problem is the exact region of contact unknown *a priori*, since the contact zone usually changes after contact takes place (i.e. a bounded zone of contact does exist). A further examination of the boundary of contact region, denoted by Γ_c in Figure 3.6, we may define a *gap function* $\ell(\boldsymbol{x})$ to represent the normal distance between the deformable body and the rigid foundation, whereas the displacement counterpart, the normal displacement denoted as $\boldsymbol{u}_\nu$, is defined as

$$\boldsymbol{u}_\nu = \boldsymbol{u}\cdot\boldsymbol{n},$$

where the positive direction of normal $\boldsymbol{n}$ is specified in Figure 3.6. A general *normal compliance* condition (p.18 [70]) on boundary Γ_c is given as

$$-\sigma_\nu = p_\nu(\boldsymbol{u}_\nu - \ell), \tag{3.78}$$

where $p_\nu(\cdot)$ is a nonnegative prescribed function that vanishes when the argument is non-positive. When $\boldsymbol{u}_\nu < \ell$, there is no contact and thus no reactive traction σ_ν; otherwise when contact takes place, $\boldsymbol{u}_\nu - \ell \geq 0$, σ_ν is negative since the nature of the traction is compressive. The function $p_\nu(\cdot)$ characterises the property of the contact surface and there are possible models (c.f [71]) available.

We may obtain the Signorini's contact condition from an idealised normal compliance

condition (3.78) in the following complementary form

$$\begin{aligned} \boldsymbol{u}_\nu - \ell &\leq 0 \\ \sigma_\nu &\leq 0 \\ \sigma_\nu \left(\boldsymbol{u}_\nu - \ell\right) &= 0 \end{aligned} \tag{3.79}$$

The sign of the traction σ_ν is the same as that from the normal compliance condition whether the two bodies are in contact or not. When the contact is first established or is about to be lost, i.e. $\boldsymbol{u}_\nu = \ell$, the normal pressure should vanish, but this need not be zero in Signorini's condition (3.79). Here we have to note that although this type of boundary conditions appear simple in their form, they are mathematically difficult, which shall become clearer later.

Next in the tangential direction, the simplest condition on Γ_c is the *frictionless* contact condition

$$\boldsymbol{\sigma}_\tau = \mathbf{0}. \tag{3.80}$$

In the case of dry contact, frictional contact is modelled with Coulomb law, which states the tangential traction $\boldsymbol{\sigma}_\tau$ has a *friction bound* H and once this bound has been reached, the surface can generate a relative slip motion. We can write this rule down as

$$\begin{aligned} \|\boldsymbol{\sigma}_\tau\| &\leq H, \\ \boldsymbol{\sigma}_\tau = -H\frac{\dot{\boldsymbol{u}}_\tau}{\|\dot{\boldsymbol{u}}_\tau\|} &\quad \text{if } \dot{\boldsymbol{u}}_\tau \neq \mathbf{0}, \end{aligned} \tag{3.81}$$

where $\dot{\boldsymbol{u}}_\tau$ is the relative tangential velocity.

To summarise, the boundary conditions for frictionless contact problem with rigid foundation in Signorini's formulation can be expressed as

$$\begin{aligned} &\Gamma_u: && \boldsymbol{u} = \mathbf{0} \quad \text{(Homogeneous Dirichlet)} \\ &\Gamma_\sigma: && \boldsymbol{\sigma}\boldsymbol{n} = \boldsymbol{f}_N \quad \text{(Neumann)} \\ &\Gamma_c: && \begin{cases} \boldsymbol{u}_\nu - \ell \leq 0 \\ \sigma_\nu \leq 0 \\ \sigma_\nu(\boldsymbol{u}_\nu - \ell) = 0 \\ \boldsymbol{\sigma}_\tau = \mathbf{0} \quad \text{(Frictionless)} \end{cases} \quad \text{(Contact constraints)} \end{aligned} \tag{3.82}$$

3.3.2 The ISF Model

Having introduced the Signorini's formulation, we model the ISF process at each incremental step in the similar manner. The ISF modelling at the initial step can be illustrated in Figure 3.7. To keep it simple, we will ignore the friction between the tool and the metal sheet in this work and assume that the dynamic effects of the process can be ignored as the ISF process is slow. We begin the ISF modelling as a quasistatic process.

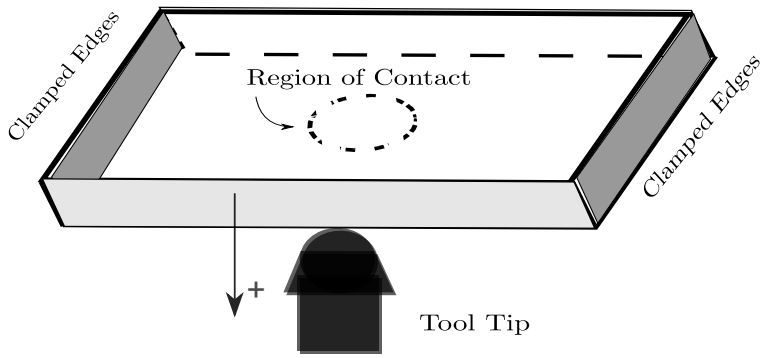


Figure 3.7 Illustration of the ISF modelling at initial step

Quasistatic Equilibrium Equation

The dynamic equation that governs the state evolution of the body based on the momentum conservation principle is

$$\rho \ddot{u}_i - \sigma_{ij,j} = f_{Bi}, \tag{3.83}$$

where ρ is the material density and f_{Bi} is the density body force, such as gravity. Note that the usual summation rule applies for index j. In the case of a slow process, such as ISF, the external forces act on the body in a way that the acceleration is small so the inertial term $\rho \ddot{u}_i$ in (3.83) can be neglected. Thus, (3.83) becomes the quasistatic state equilibrium equation

$$\text{div } \boldsymbol{\sigma} + \boldsymbol{f}_B = \mathbf{0} \tag{3.84}$$

where $\boldsymbol{f}_B$ denotes the body force and the stress $\boldsymbol{\sigma}$ has an elasto-plastic constitutive relation with the strain $\boldsymbol{\varepsilon}$ outlined in previous section.

In the sequel of this work, we shall refer to the problem of finding admissible displacements $\boldsymbol{u}$ from body governing equation (3.84) and boundary conditions in (3.82) with linear elasticity law

$$\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{\varepsilon},$$

as *Signorini's linear elasticity problem*, or simply *Signorini's problem.*

Boundary Conditions

Next, we regard the tool tip as a rigid surface and adopt Signorini's conditions (3.82) at the boundaries. For the clamped edges, the homogeneous Dirichlet boundary condition still hold. As there is no external traction exerted on the boundaries except the region of contact where the tool exerts a contact force, the Neumann boundary condition on Γ_σ is zero. For the contact constraints, the gap function in (3.82) needs extra treatment. Here the contact boundary condition will consider the kinematic constraints as well as the force requirements, which in our case is to meet the frictionless assumption. For every point $\boldsymbol{z}$ in the deformed sheet configuration in the contact region shown in Figure 3.8, we can

relate it to a point on the tool surface via a minimum distance expression

$$\min_{\overline{\xi}^{\alpha}} \left\| \boldsymbol{z} - \boldsymbol{z}_T(\overline{\xi}^{\alpha}) \right\|, \tag{3.85}$$

where $\boldsymbol{z}_T(\overline{\xi}^{\alpha})$ represents a parametric description of the tool tip. Let

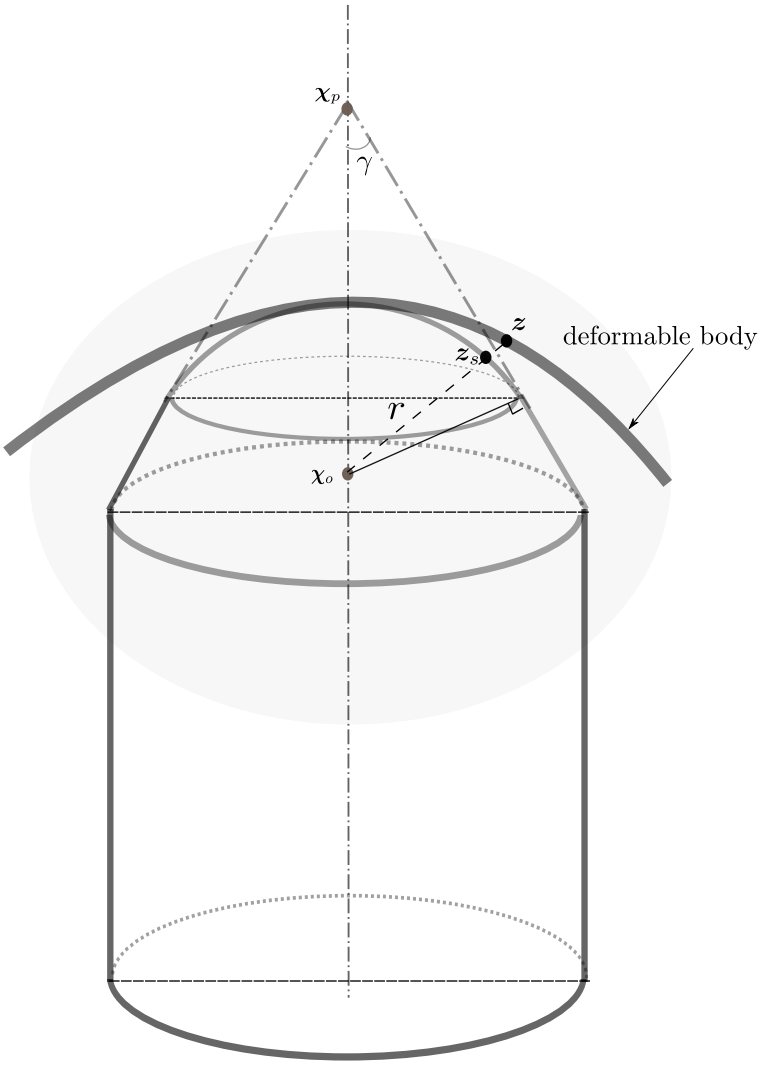


Figure 3.8 Geometry of the actual tool tip and the contact boundary treatment. $\boldsymbol{\chi}_0$ is the centre of the sphere, whose radius is denoted by r, whilst the vertex of the bottom frustum is illustrated as $\boldsymbol{\chi}_p$ (outside the tip).

$$f^d = \left\| \boldsymbol{z} - \boldsymbol{z}_T(\overline{\xi}^{\alpha}) \right\|,$$

then we can set

$$\frac{\mathrm{d}f^d}{\mathrm{d}\overline{\xi}^{\alpha}} = 0,$$

which becomes

$$\frac{\boldsymbol{z} - \boldsymbol{z}_T(\overline{\xi}^{\alpha})}{\left\| \boldsymbol{z} - \boldsymbol{z}_T(\overline{\xi}^{\alpha}) \right\|} \cdot \boldsymbol{z}_{T,\alpha}(\overline{\xi}^{\alpha}) = 0. \tag{3.86}$$

We shall recognise $\boldsymbol{z}_{T,\alpha}(\overline{\xi}^{\alpha})$ are the bases of the local curvilinear coordinates, which is further described in section 4.7.1. Also, we may deduce that

$$\frac{\boldsymbol{z}-\boldsymbol{z}_T(\overline{\xi}^{\alpha})}{\left\|\boldsymbol{z}-\boldsymbol{z}_T(\overline{\xi}^{\alpha})\right\|}=\boldsymbol{n}(\overline{\xi}^{\alpha}),$$

where $\boldsymbol{n}(\overline{\xi}^{\alpha})$ is the normal vector at point $\overline{\xi}^{\alpha}$. If we denote this point as $\boldsymbol{z}_s$ and the normal vector $\boldsymbol{n}_s$, then our gap function can be expressed as

$$\ell=(\boldsymbol{z}-\boldsymbol{z}_s)\cdot\boldsymbol{n}_s=(\boldsymbol{u}+\boldsymbol{z}_0-\boldsymbol{z}_s)\cdot\boldsymbol{n}_s, \tag{3.87}$$

where $\boldsymbol{z}_0$ denotes the initial configuration of the sheet, and our contact boundary condition becomes

$$\ell\geq 0, \tag{3.88}$$

which can be rearranged as a familiar form

$$\boldsymbol{u}\cdot\boldsymbol{n}_s\geq(\boldsymbol{z}_s-\boldsymbol{z}_0)\cdot\boldsymbol{n}_s. \tag{3.89}$$

Alternatively, we can have an equivalent boundary condition that states that the deformed configuration must lie outside the tool tip, or rather the points in the contact region must fall outside the tool tip surface. Suppose we express the tool tip surface as a function

$$\varsigma(\boldsymbol{x})=0,$$

whose exact form is

$$\varsigma(\boldsymbol{x})=\begin{cases}\|\boldsymbol{x}-\boldsymbol{\chi}_o\|-r & \text{if } \tilde{\boldsymbol{n}}\cdot\dfrac{\boldsymbol{x}-\boldsymbol{\chi}_o}{\|\boldsymbol{x}-\boldsymbol{\chi}_o\|}\geq\cos(\frac{\pi}{2}-\gamma)\\ \cos(\gamma)-\tilde{\boldsymbol{n}}\cdot\dfrac{\boldsymbol{\chi}_p-\boldsymbol{x}}{\|\boldsymbol{\chi}_p-\boldsymbol{x}\|} & \text{if } \tilde{\boldsymbol{n}}\cdot\dfrac{\boldsymbol{x}-\boldsymbol{\chi}_o}{\|\boldsymbol{x}-\boldsymbol{\chi}_o\|}<\cos(\frac{\pi}{2}-\gamma)\end{cases}, \tag{3.90}$$

where

$$\tilde{\boldsymbol{n}}=\frac{\boldsymbol{\chi}_p-\boldsymbol{\chi}_o}{\|\boldsymbol{\chi}_p-\boldsymbol{\chi}_o\|},$$

then the contact boundary condition can be expressed more compactly as

$$\varsigma(\boldsymbol{u}+\boldsymbol{x}_0)\geq 0. \tag{3.91}$$

However, kinematic conditions (3.89) and (3.91) on the contact boundary are derived from a general large deformation configuration and unfortunately both of them are nonconvex, which poses major difficulties for solving the complete model. Treatment of nonconvex contact constraints is demonstrated in [72, 73]. In this work, we would use a series of small incremental steps to simulate the physical process and the small deformation is our assumption. So we use the approximation approach that is similar to the one in [74] to obtain a small deformation boundary condition.

The rigid surface (i.e. the tool surface) can be modelled with

$$\varsigma(\boldsymbol{x}, \boldsymbol{o}_c) = 0, \tag{3.92}$$

where $\boldsymbol{x}$ denotes the position variable, and $\boldsymbol{o}_c$ denotes the centre of the tool, which is a constant parameter. Consider a point on the deformable body in the contact region, with initial position denoted as $\boldsymbol{x}_0$, which undergoes a deformation $\boldsymbol{u}$, then the current position would be $\boldsymbol{x}_0 + \boldsymbol{u}$. The kinematical constraint requires that the point has to be outside the surface ς after deformation, so mathematically

$$\varsigma(\boldsymbol{x}_0 + \boldsymbol{u}) \geq 0, \tag{3.93}$$

where the equality means the points are in perfect contact with the rigid surface. Note that we have omitted the constant parameter $\boldsymbol{o}_c$ as it is not related to the position of the point of interest. Under the small deformation assumption, we approximate $\varsigma(\boldsymbol{x}_0 + \boldsymbol{u})$ as

$$\varsigma(\boldsymbol{x}_0 + \boldsymbol{u}) \approx \varsigma(\boldsymbol{x}_0) + \varsigma_{\boldsymbol{x}}(\boldsymbol{x}_0) \cdot \boldsymbol{u}, \tag{3.94}$$

where $\varsigma_{\boldsymbol{x}}$ represent derivative with respect to $\boldsymbol{x}$. So the inequality (3.93) can be reformulated as

$$\varsigma(\boldsymbol{x}_0) + \varsigma_{\boldsymbol{x}}(\boldsymbol{x}_0) \cdot \boldsymbol{u} \geq 0. \tag{3.95}$$

The normal to the rigid surface can be expressed by

$$\boldsymbol{n} = \frac{\varsigma_{\boldsymbol{x}}}{|\varsigma_{\boldsymbol{x}}|}, \tag{3.96}$$

so dividing (3.95) by $|\varsigma_{\boldsymbol{x}}|$, we obtain

$$\boldsymbol{u} \cdot \boldsymbol{\nu} \geq -\frac{\varsigma(\boldsymbol{x}_0)}{|\varsigma_{\boldsymbol{x}}|}, \tag{3.97}$$

where $-\dfrac{\varsigma(\boldsymbol{x}_0)}{|\varsigma_{\boldsymbol{x}}|}$ is the gap function referred earlier (cf. (3.78)). The normal vector $\boldsymbol{\nu}$ is not easy to specify, but a further approximation can be made since the small deformation allows

$$\boldsymbol{n} = -\boldsymbol{\nu}. \tag{3.98}$$

Let

$$\ell(\boldsymbol{x}_0) = \frac{\varsigma(\boldsymbol{x}_0)}{|\varsigma_{\boldsymbol{x}}|}.$$

Substituting $\boldsymbol{\nu}$ with (3.98), we obtain

$$\boldsymbol{u} \cdot \boldsymbol{n}(\boldsymbol{x}_0) \leq \ell(\boldsymbol{x}_0). \tag{3.99}$$

Note that the omitted parameter $\boldsymbol{o}_c$ in ς is always set to the final configuration (i.e. the tool position after a movement has been made) and our notation $\boldsymbol{u}_\nu$ is associated with the

normal direction $\boldsymbol{n}(x_0)$, i.e.

$$\boldsymbol{u}_\nu = \boldsymbol{u} \cdot \boldsymbol{n}(\boldsymbol{x}_0).$$

Summary of ISF Model

So the frictionless ISF model with Signorini type of condition can be formally written as

$$\begin{aligned}
&\text{Find displacement } \boldsymbol{u}, \quad \text{such that} \\
V: \quad & \operatorname{div} \boldsymbol{\sigma} + \boldsymbol{f}_B = \boldsymbol{0} \\
\Gamma_u: \quad & \boldsymbol{u} = \boldsymbol{0} \\
\Gamma_\sigma: \quad & \boldsymbol{\sigma n} = \boldsymbol{f}_N = \boldsymbol{0} \\
\Gamma_c: \quad & \begin{cases} \sigma_\nu \leq 0 \\ \boldsymbol{u}_\nu \leq \ell \\ \sigma_\nu \left(\boldsymbol{u}_\nu - \ell\right) = 0 \\ \boldsymbol{\sigma}_\tau = \boldsymbol{0} \end{cases} \\
\text{and} \quad & \dot{\boldsymbol{\sigma}} = \boldsymbol{C}^{ep} \dot{\boldsymbol{\varepsilon}}, \\
& \boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right),
\end{aligned} \tag{3.100}$$

where V denotes the physical body of the sheet metal, excluding its boundaries Γ_σ, Γ_u and Γ_c.

"Mankind always sets itself only such problems as it can solve; since, looking at the matter more closely, it will always be found that the task itself arises only when the material conditions for its solution already exist or are at least in the process of formation."

Karl Marx, A contribution to the critique of political economy, 1859

Chapter 4

Computational Method for ISF Model

The ISF model developed in the previous chapter is a boundary value partial differential equation, which could be solved by two main approaches in general, namely the *finite difference method* or the *variational method*. The finite difference method is widely used to solve "simple" problems: simple geometry domain and simple boundary conditions. As our problem is beyond the limit of the finite difference method, we will adopt the latter approach, to which the finite element method belongs.

This chapter describes how the actual computational method, or the finite element method, can be used to solve the ISF model problem in the last chapter. We first present the mathematical theory of (linear) finite element method and develop the variational form for Signorini's linear elasticity problem (see page 76), then introduce the elasto-plasticity algorithm, where the time dimension has been first discretised, and finally reformulate the ISF model as a minimisation problem where the optimality conditions are given in order to solve it.

4.1 Finite Element Method: A Brief Introduction

The finite element method was first established by civil engineers around 1956 [75, 76], and the mathematical theory of this method catches up with engineering practice from early work of Courant [77] to a rigorous stage by Strang and Fix's work [78]. In this section, we give an introduction to the mathematical theory of the finite element method, so that this computational method for our ISF model can be established.

Quite often the *exact (classical) solution* to a partial differential equation requires high regularity restrictions and does not always exist; it is then possible to weaken the regularity requirements so that a solution to the original problem make sense. We refer to this type of solution as *weak solution*, which can be very useful to explain the real world phenomena. Sometimes the weak solution is the natural solution to a physical problem as it is derived from integral laws over some domain [79].

4.1.1 An Example

To demonstrate the procedure of the finite element method, we take an elliptic problem [80] as an example:

$$-\nabla \cdot (a_1 \nabla u) + a_0 u = f, \tag{4.1}$$

defined in a open bounded set $\Omega \subset \mathcal{R}^d$ with Lipschitz continuous boundary and imposed with homogeneous Dirichlet boundary conditions

$$u(\boldsymbol{x}) = 0, \text{ on } \partial\Omega.$$

The classical solution to this problem is a function u satisfying $u \in \mathcal{C}^2(\Omega) \cap \mathcal{C}^1(\bar{\Omega})$ (twice derivative continuous in the set Ω and first derivative continuous including the boundary) everywhere in the Ω and satisfying the boundary conditions at every point $\boldsymbol{x} \in \partial\Omega$.

To reduce the regularity requirements upon solution u, we first multiply problem (4.1) with a *test function* $v \in \mathcal{C}^\infty(\Omega)$, so that

$$-\nabla \cdot (a_1 \nabla u)v + a_0 uv = fv, \tag{4.2}$$

and integrate over Ω,

$$-\int_\Omega \nabla \cdot (a_1 \nabla u)v \,\mathrm{d}\boldsymbol{x} + \int_\Omega a_0 uv \,\mathrm{d}\boldsymbol{x} = \int_\Omega fv \,\mathrm{d}\boldsymbol{x}. \tag{4.3}$$

Note that test function v employs the same boundary condition with that of u, i.e. $v(\boldsymbol{x}) = 0$, on $\partial\Omega$. For the first integral term in (4.3), we apply the divergence related product rule

$$\nabla \cdot (v\boldsymbol{F}) = (\nabla v) \cdot \boldsymbol{F} + (\nabla \cdot \boldsymbol{F})v,$$

where $\boldsymbol{F}$ is a general vector, by replacing $\boldsymbol{F}$ with ∇u, therefore

$$\int_\Omega \nabla \cdot (\nabla u)v \,\mathrm{d}\boldsymbol{x} = \int_\Omega \nabla \cdot (\nabla u \; v) \,\mathrm{d}\boldsymbol{x} - \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}\boldsymbol{x}. \tag{4.4}$$

The first term of the right hand side of equation (4.4) can be further simplified using Green's formula, so that we obtain

$$\int_\Omega \nabla \cdot (\nabla u)v \,\mathrm{d}\boldsymbol{x} = \oint_{\partial\Omega} (\nabla u \; v) \cdot \boldsymbol{n} \,\mathrm{d}s - \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}\boldsymbol{x}, \tag{4.5}$$

where $\boldsymbol{n}$ is the outward normal vector to the boundary element $\mathrm{d}s$. Relation (4.5) is known as Green's identity, as a result, formulation (4.3) becomes

$$\int_\Omega a_1 \nabla u \cdot \nabla v \,\mathrm{d}\boldsymbol{x} + \int_\Omega a_0 uv \,\mathrm{d}\boldsymbol{x} = \int_\Omega fv \,\mathrm{d}\boldsymbol{x}, \quad \text{for every } v. \tag{4.6}$$

Note that the boundary term in (4.5) vanishes due to the zero boundary condition, i.e. $v = 0$ on $\partial\Omega$.

Because the solution to the problem (4.6) includes an integration compared to the original form, the regularity requirements is weakened to

$$u, v \in \mathcal{H}_0^1(\Omega),$$

where $\mathcal{H}_0^1(\Omega)$ denotes the *Sobolev space*, whose component is square-integrable and possesses all the partial spatial derivatives of the first order. (Note that the product of two square-integrable function is an integrable function but not necessarily square-integrable.) In an abstract form, equation (4.6) can be cast as

$$a(u, v) = l(v), \quad \text{for every } v \in \mathcal{V}, \tag{4.7}$$

where $a(u, v)$ is a bilinear function, $l(v)$ is a linear function and the Sobolev space $\mathcal{H}_0^1(\Omega)$ is denoted by $\mathcal{V}$.

It is often not easy to obtain the exact solution of this problem, so instead, an associated discrete problem is used to approximate the solution. The *Galerkin method* defines this discrete problem in the subspace of space $\mathcal{V}$, denoted by $\mathcal{V}_h$: find $u_h \in \mathcal{V}_h$ such that

$$\forall v_h \in \mathcal{V}_h, \quad a(u_h, v_h) = l(v_h). \tag{4.8}$$

The discretisation inevitably involves domain discretisation, which usually means that a mesh is needed, which will be explained more in the next section.

To solve (4.8) practically we choose some *basis function* of the finite element, denoted as w_k and $k = 1, 2, 3, \cdots m$, for both u_h and v_h in the space $\mathcal{V}_h$, so that

$$u_h = w_k u_k, \quad v_h = w_l v_l.$$

where u_k and v_l are the coefficients of the basis terms. In fact, the choice of the basis $(w_k)_{k=1}^m$ is of significant importance, to which we shall return later. The problem (4.8) then becomes

$$\begin{aligned} & a(w_k u_k, w_l v_l) = l(w_l v_l), \\ \text{or,} \quad & a(w_k, w_l) u_k v_l = l(w_l) v_l. \end{aligned} \tag{4.9}$$

Every coefficient of v_l must be identical between left and right hand side of equation (4.9) so that equation (4.8) can hold for any v_h in $\mathcal{V}_h$. Therefore, the solution to problem (4.9) is identical to solving the coefficients u_k for the following linear system

$$a(w_k, w_l) u_k = l(w_l), \text{ where } l = 1, 2, 3, \cdots, m. \tag{4.10}$$

The matrix $[a(w_k, w_l)]$ is usually called *stiffness matrix* and vector $[l(w_l)]$ is called *load vector* as historically (4.10) comes from elasticity problems.

We may expect the existence of approximation errors compared with the theoretical solution to the original problem and there are two major approaches to increase the accuracy of the finite element computations: one is *h version* finite element, in which

polynomials of fixed degree p are used and mesh refinement is used to increase the results accuracy, and the other, *p version* finite element, where the p is increased whilst using a fixed mesh [81]. A hybrid method of the two is of course admissible and it is called the *h-p version.*

4.1.2 Ingredients of Finite Elements

To generalise the discussion about the finite element method, suppose the original boundary value problem is posed over an open set Ω with boundary Γ and we consider the procedure to construct the subspace $\mathcal{V}_h$ in Galerkin method.

First, *triangulation* of the (closed) set $\bar{\Omega}$, i.e., the set $\bar{\Omega}$ is subdivided into a finite number of closed subsets K, called *finite elements*, in a way that $\bar{\Omega} = \cup K$, the interior of K is non-empty and non-overlapping between two elements and the boundary $\partial\Omega$ is Lipschitz-continuous. In practice, we usually set K as a *simplex* (*triangle* in 2 dimensions and *tetrahedron* in 3 dimensions). In this work, we will exclusively make triangles as our subset choice. Then, we define a finite element space $\mathcal{X}_h$ and finite dimensional space P_K spanned by restrictions $v_{h|K}$ of the functions $v_h \in \mathcal{X}_h$ to the finite element K as

$$P_K = \{v_{h|K}; \quad v_h \in \mathcal{X}_h\}.$$

The space P_K contains polynomials and is the key both to convergence and to obtaining a simple computational method to solve problem (4.8) [82]. We shall denote P_k (lower case k to differentiate from P_K) as the space of all polynomials of degree $\leq k$ of variables $x_i, i = 1, 2..., n$, i.e., a polynomial $p_{ol} \in P_k$ of the form

$$p_{ol}(x) = \sum_{\sum_{i=1}^{n} \alpha_i \leq k} \tilde{\eta}_{\alpha_1 \alpha_2 \cdots \alpha_n} x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n},$$

for appropriate coefficients $\tilde{\eta}_{\alpha_1 \alpha_2 \cdots \alpha_n}$. The dimension of the space P_k is given by

$$\dim\, P_k = \binom{n+k}{k}. \tag{4.11}$$

It is known that the function in space P_K can be uniquely determined by values and derivatives at the vertices and mid-points of the edges, termed as *degrees of freedom of the finite element* [82]. We then denote the corresponding *set of degrees of freedom* by Σ_K, whose exact form depends on degree k of a specific space P_k and the locations of the points (most commonly, vertices and mid-points) of interest.

A formal definition for a general finite element given in [82] is to regard finite element as a triples (K, P, Σ), where K, known as the finite element, has been defined earlier, P is a space of real-valued functions defined over set K, and Σ is a finite set of linearly independent linear forms $\psi_i, 1 \leq i \leq N$, defined over the space P. Most importantly, we

have

$$\forall p_{ol} \in P, \quad p_{ol} = \sum_{i=1}^{N} \psi_i(p_{ol}) p_{ol\,i}. \tag{4.12}$$

For the linear form ψ_i, members of set Σ, are the degrees of freedom of the finite element and the function $p_{ol\,i}$ are called the *basis functions of the finite element*. Thus, the bases $(\psi_i)_{i=1}^N$ and $(p_{ol\,i})_{i=1}^N$ can be viewed as dual bases in the algebraic sense. Given a finite element (K, P, Σ) and a smooth function $v : K \to \mathbb{R}$, we can define the *P-interpolant* of the function v as

$$\Pi v = \sum_{i=1}^{N} \psi_i(v) p_{ol\,i}. \tag{4.13}$$

4.1.3 Variational Formulation

The abstract form in (4.7) is actually a *variational problem*, which leads to the weak solution and the corresponding finite element approximation. We rewrite the problem by explicitly defining a linear space $\mathcal{V}$, so that the problem becomes to find $u \in \mathcal{V}$, such that

$$\forall v \in \mathcal{V}, \quad a(u, v) = l(v), \tag{4.14}$$

where $a(\cdot, \cdot) : \mathcal{V} \times \mathcal{V} \to \mathcal{R}$ as before denotes the bilinear form and $l(\cdot) : \mathcal{V} \to \mathcal{R}$ denotes the linear form. The space $\mathcal{V}$, a and l are assumed to satisfy the Lax-Milgram lemma (pp.8 [82]), which ensures problem (4.14) has a unique solution. So problem (4.1) does have a unique solution.

If we define a functional $J(v)$ as

$$J(v) \triangleq \frac{1}{2} a(v, v) - l(v), \tag{4.15}$$

then by a standard characterisation theorem (p.35 [83]), $J(v)$ attains its minimum over $\mathcal{V}$ if and only if (4.14) holds. The key to this proof is the introduction of function $J(u + tv)$, where $t \in R, v \in \mathcal{V}$:

$$J(u + tv) = \frac{1}{2} a(u + tv, u + tv) - l(u + tv) = J(u) + t\,(a(u, v) - l(v)) + \frac{1}{2} t^2 a(v, v). \tag{4.16}$$

Together with the positivity of $a(v, v)$, if (4.14) hold then $J(u)$ attains minimum and conversely, if $J(u)$ attains minimum, the derivative must vanish, i.e. (4.14) holds. In fact, the functional J is an energy expression for the underlying physical system, from which the abstract form (4.14) can be derived by a variational principle, which explains the term "variational". This important observation will allow us to treat complex contact boundary conditions of the ISF model in the finite element analysis context with some variational method. As a result, we also refer to the abstract form (4.14) as the variational formulation.

4.2 Variational Formulation of Signorini's Problem

In this section, we lay the foundation of a weak form or variational form of the ISF model developed in (3.100), so that a finite element method can be used. Compared to the linear finite element problem introduced in the previous section, the nature of our ISF model turns out to be much more complicated in the sense that both a complementary contact boundary condition and a history-dependent elasto-plasticity material law are used. Here we first analyse Signorini's linear elasticity problem (see page 76), where a linear elasticity material with a similar complementary contact boundary condition are included.

To obtain a variational formulation for Signorini's problem, we start with the quasistatic state equilibrium equation (3.84) in Cartesian coordinates using standard techniques. Multiplying by test function $\boldsymbol{v}$ and integrating over body domain V, we have

$$\int_V -(\text{div } \boldsymbol{\sigma}) \cdot \boldsymbol{v} \, \mathrm{d}V = \int_V \boldsymbol{f}_B \cdot \boldsymbol{v} \, \mathrm{d}V, \tag{4.17}$$

where $\boldsymbol{v}$ is defined over linear space $\mathcal{V}$. To account for the boundary conditions in (3.82) we also require

$$\begin{aligned} \boldsymbol{v} &= \boldsymbol{0}, \quad \text{on } \Gamma_u \\ \boldsymbol{v}_\nu &\leq \ell, \quad \text{on } \Gamma_c. \end{aligned}$$

Define a convex space $\mathcal{K}$ as

$$\mathcal{K} = \{ \boldsymbol{v} \in \mathcal{V} \,|\, \boldsymbol{v} = \boldsymbol{0}, \text{ on } \Gamma_u \text{ and } \boldsymbol{v}_\nu \leq \ell \text{ on } \Gamma_c \}, \tag{4.18}$$

so that

$$\boldsymbol{v} \in \mathcal{K}.$$

Next, let us consider the relation $\text{div } (\boldsymbol{\sigma}\boldsymbol{v}) = (\sigma_{ij} v_j)_{,i}$, whose right hand side can be expanded using the chain rule as

$$\text{div } (\boldsymbol{\sigma}\boldsymbol{v}) = \sigma_{ij,i} v_j + \sigma_{ij} v_{j,i} = \sigma_{ji,i} v_j + \sigma_{ji} v_{j,i} = \text{div } (\boldsymbol{\sigma}) \cdot \boldsymbol{v} + \boldsymbol{\sigma} : \nabla \boldsymbol{v}. \tag{4.19}$$

Thus,

$$\begin{aligned} \int_V -\text{div } (\boldsymbol{\sigma}) \cdot \boldsymbol{v} \, \mathrm{d}V &= \int_V \left[\boldsymbol{\sigma} : \nabla \boldsymbol{v} - \text{div } (\boldsymbol{\sigma}\boldsymbol{v}) \right] \mathrm{d}V \\ = \int_V \boldsymbol{\sigma} : \nabla \boldsymbol{v} \, \mathrm{d}V - \int_V \text{div } (\boldsymbol{\sigma}\boldsymbol{v}) \, \mathrm{d}V. \end{aligned} \tag{4.20}$$

In the above derivation, the symmetry of the stress tensor $\boldsymbol{\sigma}$ is used. We also have the relation

$$(\boldsymbol{\sigma}\boldsymbol{v}) \cdot \boldsymbol{n} = \sigma_{ij} v_j n_i = \sigma_{ji} v_j n_i = (\sigma_{ji} n_i) v_j = (\boldsymbol{\sigma}\boldsymbol{n}) \cdot \boldsymbol{v}, \tag{4.21}$$

where $\boldsymbol{n}$ is an arbitrary vector for the moment. As for the divergence integration term in equation (4.20), using the divergence theorem

$$\int_V \text{div } (\boldsymbol{\sigma}\boldsymbol{v}) \, \mathrm{d}V = \oint_\Gamma (\boldsymbol{\sigma}\boldsymbol{v}) \cdot \boldsymbol{n} \, \mathrm{d}S, \tag{4.22}$$

where Γ is the boundary of the region V and $\boldsymbol{n}$ is the normal vector to Γ. Using relation (4.21), we have

$$\int_V \operatorname{div}(\boldsymbol{\sigma}\boldsymbol{v})\,\mathrm{d}V = \oint_\Gamma (\boldsymbol{\sigma}\boldsymbol{n})\cdot\boldsymbol{v}\,\mathrm{d}S. \tag{4.23}$$

So, combining equation (4.20) and (4.23), we obtain the variational form

$$\int_V \boldsymbol{\sigma} : \nabla\boldsymbol{v}\,\mathrm{d}V = \int_V \boldsymbol{f}_B\cdot\boldsymbol{v}\,\mathrm{d}V + \oint_\Gamma (\boldsymbol{\sigma}\boldsymbol{n})\cdot\boldsymbol{v}\,\mathrm{d}S \quad \forall\boldsymbol{v}\in\mathcal{V}. \tag{4.24}$$

Due to the symmetry property of the stress tensor $\boldsymbol{\sigma}$, $\sigma_{ij} = \sigma_{ij}$, we may introduce a symmetric gradient operator ∇^s, acting on $\boldsymbol{v}$ as

$$\nabla^s\boldsymbol{v} = \frac{1}{2}\left(v_{i,j} + v_{j,i}\right).$$

So

$$\boldsymbol{\sigma} : \nabla\boldsymbol{v} = \sigma_{ij}\boldsymbol{v}_{i,j} = \sigma_{ji}\boldsymbol{v}_{j,i} = \sigma_{ij}\frac{1}{2}\left(v_{i,j} + v_{j,i}\right) = \boldsymbol{\sigma} : \nabla^s\boldsymbol{v},$$

and equation (4.24) now takes the form

$$\int_V \boldsymbol{\sigma} : \nabla^s\boldsymbol{v}\,\mathrm{d}V = \int_V \boldsymbol{f}_B\cdot\boldsymbol{v}\,\mathrm{d}V + \oint_\Gamma (\boldsymbol{\sigma}\boldsymbol{n})\cdot\boldsymbol{v}\,\mathrm{d}S, \quad \forall\boldsymbol{v}\in\mathcal{K}. \tag{4.25}$$

For the surface integral in (4.25), substituting the boundary Γ with its components Γ_u, Γ_σ and Γ_c, yields,

$$\begin{aligned}\oint_\Gamma (\boldsymbol{\sigma}\boldsymbol{n})\cdot\boldsymbol{v}\,\mathrm{d}S &= \oint_{\Gamma_u} (\boldsymbol{\sigma}\boldsymbol{n})\cdot\boldsymbol{v}\,\mathrm{d}S + \oint_{\Gamma\sigma} (\boldsymbol{\sigma}\boldsymbol{n})\cdot\boldsymbol{v}\,\mathrm{d}S + \oint_{\Gamma_c} (\boldsymbol{\sigma}\boldsymbol{n})\cdot\boldsymbol{v}\,\mathrm{d}S \\ &= \oint_{\Gamma\sigma} \boldsymbol{f}_N\cdot\boldsymbol{v}\,\mathrm{d}S + \oint_{\Gamma_c} \sigma_\nu\boldsymbol{v}_\nu\,\mathrm{d}S.\end{aligned} \tag{4.26}$$

The normal reactive traction σ_ν in the contact constraints expressed in (3.82) is unknown and the test function $\boldsymbol{v}$ lies in space $\mathcal{K}$, which says $\boldsymbol{v}$ is confined on Γ_c as $v_\nu \le g$. So this constrained formulation causes a major hurdle for further analysis.

Alternatively, a *variational inequality* formulation can pose an elegant framework for Signorini's problem. The idea is to use test function $\boldsymbol{v}-\boldsymbol{u}$ instead of $\boldsymbol{v}$, so the variational form (4.25) can be expressed as

$$\int_V \boldsymbol{\sigma} : \nabla^s(\boldsymbol{v}-\boldsymbol{u})\,\mathrm{d}V = \int_V \boldsymbol{f}_B\cdot(\boldsymbol{v}-\boldsymbol{u})\,\mathrm{d}V + \oint_\Gamma (\boldsymbol{\sigma}\boldsymbol{n})\cdot(\boldsymbol{v}-\boldsymbol{u})\,\mathrm{d}S, \quad \forall\boldsymbol{v}\in\mathcal{K}, \tag{4.27}$$

where

$$\begin{aligned}\oint_\Gamma (\boldsymbol{\sigma}\boldsymbol{n})\cdot(\boldsymbol{v}-\boldsymbol{u})\,\mathrm{d}S &= \oint_{\Gamma_u} (\boldsymbol{\sigma}\boldsymbol{n})\cdot(\boldsymbol{v}-\boldsymbol{u})\,\mathrm{d}S + \oint_{\Gamma_\sigma} (\boldsymbol{\sigma}\boldsymbol{n})\cdot(\boldsymbol{v}-\boldsymbol{u})\,\mathrm{d}S + \oint_{\Gamma_c} (\boldsymbol{\sigma}\boldsymbol{n})\cdot(\boldsymbol{v}-\boldsymbol{u})\,\mathrm{d}S \\ &= \oint_{\Gamma_\sigma} \boldsymbol{f}_N\cdot(\boldsymbol{v}-\boldsymbol{u})\,\mathrm{d}S + \oint_{\Gamma_c} \left[\sigma_\nu(\boldsymbol{v}_\nu-\boldsymbol{u}_\nu) + \boldsymbol{\sigma}_\tau\cdot(\boldsymbol{v}_\tau-\boldsymbol{u}_\tau)\right]\mathrm{d}S.\end{aligned}$$

The contact boundary integral can be further treated as

$$\begin{aligned}\oint_{\Gamma_c} \sigma_\nu(\boldsymbol{v}_\nu - \boldsymbol{u}_\nu)\,\mathrm{d}S &= \oint_{\Gamma_c} \sigma_\nu\left(\boldsymbol{v}_\nu - \ell - (\boldsymbol{u}_\nu - \ell)\right)\,\mathrm{d}S \\ &= \oint_{\Gamma_c} \sigma_\nu(\boldsymbol{v}_\nu - \ell)\,\mathrm{d}S - \oint_{\Gamma_c} \sigma_\nu(\boldsymbol{u}_\nu - \ell)\,\mathrm{d}S, \qquad (4.28)\\ \oint_{\Gamma_c} \boldsymbol{\sigma}_\tau \cdot (\boldsymbol{v}_\tau - \boldsymbol{u}_\tau)\,\mathrm{d}S &= 0 \quad (\boldsymbol{\sigma}_\tau = \boldsymbol{0}).\end{aligned}$$

Given the complementary contact constraints (3.82), it is clear that

$$\oint_{\Gamma_c} \sigma_\nu(\boldsymbol{u}_\nu - \ell)\,\mathrm{d}S = 0, \qquad (4.29)$$

and since $\sigma_\nu \leq 0$ and $\boldsymbol{v}_\nu \leq \ell$, we have

$$\oint_{\Gamma_c} \sigma_\nu(\boldsymbol{v}_\nu - \ell)\,\mathrm{d}S \geq 0. \qquad (4.30)$$

Combining (4.29) and (4.30), we can see

$$\oint_{\Gamma_c} \sigma_\nu(\boldsymbol{v}_\nu - \boldsymbol{u}_\nu)\,\mathrm{d}S \geq 0, \qquad (4.31)$$

and that the equation (4.27) becomes the inequality

$$\int_V \boldsymbol{\sigma} : \nabla^s(\boldsymbol{v} - \boldsymbol{u})\,\mathrm{d}V - \int_V \boldsymbol{f}_B \cdot (\boldsymbol{v} - \boldsymbol{u})\,\mathrm{d}V - \oint_{\Gamma_\sigma} \boldsymbol{f}_N \cdot (\boldsymbol{v} - \boldsymbol{u})\,\mathrm{d}S \geq 0, \quad \forall \boldsymbol{v} \in \mathcal{K}, \quad (4.32)$$

which is termed variational inequality.

We shall refer to the problem: find an admissible displacement

$$\boldsymbol{u} \in \mathcal{K},$$

such that (4.32) holds, as the *variational formulation of the Signorini's problem.*

Theorem 4.2.1. *The variational form of Signorini's problem is equivalent to its original form.*

Proof. The necessity has been shown above, what is left is to demonstrate its sufficiency, i.e., we can retrieve the Signorini's problem from inequality (4.32) and $\boldsymbol{u} \in \mathcal{K}$.

First, construct the test function $\boldsymbol{v}$ with an arbitrary smooth function $\boldsymbol{\varphi}$ by

$$\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{\varphi}, \qquad \text{and } \boldsymbol{v} = \boldsymbol{u} - \boldsymbol{\varphi}.$$

Let $\boldsymbol{\varphi} \in \mathcal{V}$, over V, where we assume $\boldsymbol{u}$ is smooth enough, so that the property of linear space guarantees $\boldsymbol{v} \in \mathcal{V}$, over V, whilst the boundary condition imposed on $\boldsymbol{\varphi}$ has to be consistent with that of $\boldsymbol{v} \in \mathcal{K}$. First, consider some arbitrary $\boldsymbol{\varphi}$, with zero value boundary

conditions on the whole Γ. Substituting $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{\varphi}$, (4.32) becomes

$$\int_V \boldsymbol{\sigma} : \nabla^s \boldsymbol{\varphi} \, \mathrm{d}V \geq \int_V \boldsymbol{f}_B \cdot \boldsymbol{\varphi} \, \mathrm{d}V, \quad \forall \boldsymbol{\varphi} \in \mathcal{K}, \tag{4.33}$$

On the other hand, substituting with $\boldsymbol{v} = \boldsymbol{u} - \boldsymbol{\varphi}$, (4.32) becomes

$$\int_V \boldsymbol{\sigma} : \nabla^s \boldsymbol{\varphi} \, \mathrm{d}V \leq \int_V \boldsymbol{f}_B \cdot \boldsymbol{\varphi} \, \mathrm{d}V, \quad \forall \boldsymbol{\varphi} \in \mathcal{K}, \tag{4.34}$$

and from the two inequalities (4.33) and (4.34), we can see

$$\int_V \boldsymbol{\sigma} : \nabla^s \boldsymbol{\varphi} \, \mathrm{d}V = \int_V \boldsymbol{f}_B \cdot \boldsymbol{\varphi} \, \mathrm{d}V, \quad \forall \boldsymbol{\varphi} \in \mathcal{K}. \tag{4.35}$$

Recall the divergence theorem

$$\int_V \boldsymbol{\sigma} : \nabla^s \boldsymbol{\varphi} \, \mathrm{d}V = \int_V -\mathrm{div}(\boldsymbol{\sigma}) \cdot \boldsymbol{\varphi} \, \mathrm{d}V + \oint_\Gamma \boldsymbol{\sigma} \boldsymbol{n} \cdot \boldsymbol{\varphi} \, \mathrm{d}V, \tag{4.36}$$

so combining (4.35) with (4.36), we have

$$\int_V -\mathrm{div}(\boldsymbol{\sigma}) \cdot \boldsymbol{\varphi} \, \mathrm{d}V = \int_V \boldsymbol{f}_B \cdot \boldsymbol{\varphi} \, \mathrm{d}V, \tag{4.37}$$

from which we can deduce the equilibrium equation

$$\mathrm{div}(\boldsymbol{\sigma}) + \boldsymbol{f}_B = \boldsymbol{0}.$$

If we let the function $\boldsymbol{\varphi}$ only vanish on Γ_u and Γ_c ($\boldsymbol{v} \in \mathcal{K}$ remains valid), then with the same reasoning, we have

$$\int_{\Gamma_\sigma} (\boldsymbol{\sigma} \boldsymbol{n} - \boldsymbol{f}_N) \cdot \boldsymbol{\varphi} \, \mathrm{d}S = 0, \tag{4.38}$$

which leads to

$$\boldsymbol{\sigma} \boldsymbol{n} = \boldsymbol{f}_N, \quad \text{on } \Gamma_\sigma.$$

By virtue of the fact that $\boldsymbol{u} \in \mathcal{K}$, we have

$$\boldsymbol{u} = \boldsymbol{0}, \quad \text{on } \Gamma_u,$$

and

$$\boldsymbol{u}_\nu \leq \ell, \quad \text{on } \Gamma_c.$$

Finally, to obtain the other relations on the contact boundary Γ_c, we substitute (4.36) into (4.32), resulting in

$$\oint_\Gamma \boldsymbol{\sigma} \boldsymbol{n} \cdot (\boldsymbol{v} - \boldsymbol{u}) \, \mathrm{d}S \geq \oint_{\Gamma_\sigma} \boldsymbol{f}_N \cdot (\boldsymbol{v} - \boldsymbol{u}) \, \mathrm{d}S, \quad \forall \boldsymbol{v} \in \mathcal{K}. \tag{4.39}$$

Using the established results so far on boundary Γ_u and Γ_σ, we conclude that

$$\oint_{\Gamma_c} \boldsymbol{\sigma n} \cdot (\boldsymbol{v} - \boldsymbol{u}) \, \mathrm{d}S \geq 0, \quad \forall \boldsymbol{v} \in \mathcal{K}. \tag{4.40}$$

We let $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{\varphi}$ with $\boldsymbol{\varphi}_\nu = 0$ and $\boldsymbol{\varphi}_\tau = \pm\boldsymbol{\phi}$, so that $\boldsymbol{v} \in \mathcal{K}$. From (4.40) we can obtain

$$\oint_{\Gamma_c} (\sigma_\nu \boldsymbol{\varphi}_\nu \pm \boldsymbol{\sigma}_\tau \cdot \boldsymbol{\phi}) \, \mathrm{d}S \geq 0, \quad \forall \boldsymbol{\phi} \quad \Rightarrow \quad \oint_{\Gamma_c} \boldsymbol{\sigma}_\tau \cdot \boldsymbol{\phi} \, \mathrm{d}S = 0, \quad \forall \boldsymbol{\phi} \quad \Rightarrow \quad \boldsymbol{\sigma}_\tau = \mathbf{0}.$$

Let $\boldsymbol{\varphi}_\nu \leq 0$ and $\boldsymbol{\varphi}_\tau = \mathbf{0}$, still with $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{\varphi} \in \mathcal{K}$, yielding,

$$\oint_{\Gamma_c} \sigma_\nu \boldsymbol{\varphi}_\nu \, \mathrm{d}S \geq 0, \forall \boldsymbol{\varphi}_\nu \leq 0 \quad \Rightarrow \quad \sigma_\nu \leq 0. \tag{4.41}$$

Let $\boldsymbol{u}_\nu < \ell$ at a point $\boldsymbol{x} \in \Gamma_c$, then there exists a smooth function $\phi \geq 0$ on Γ_c such that

$$\phi(\boldsymbol{x}) > 0 \quad \text{ and } \boldsymbol{u}_\nu + \phi \leq \ell.$$

Let some $\boldsymbol{\varphi} \in \mathcal{V}$ satisfy

$$\boldsymbol{\varphi}_\nu = \phi, \quad \boldsymbol{\varphi}_\tau = \mathbf{0} \text{ on } \Gamma_c,$$

to ensure that $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{\varphi} \in \mathcal{K}$. Using the inequality

$$\oint_{\Gamma_c} \sigma_\nu \boldsymbol{\varphi}_\nu \, \mathrm{d}S \geq 0$$

and established result $\sigma_\nu \leq 0$ in (4.41), we can conclude

$$\sigma_\nu = 0$$

in the case of $\boldsymbol{u}_\nu < \ell$, which is the complementary condition on Γ_c. This completes our (sufficiency) proof. □

By further investigating the linear elastic law for the stress-strain relation and the kinematic equation, we have

$$\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{\varepsilon}(\boldsymbol{u}) = \boldsymbol{C}\nabla^s \boldsymbol{u}. \tag{4.42}$$

Substituting (4.42) into (4.32), we can rewrite the variational form for Signorini's problem as: find $u \in \mathcal{K}$, such that

$$\int_V \boldsymbol{C}\nabla^s \boldsymbol{u} : \nabla^s(\boldsymbol{v} - \boldsymbol{u}) \, \mathrm{d}V \geq \int_V \boldsymbol{f}_B \cdot (\boldsymbol{v} - \boldsymbol{u}) \, \mathrm{d}V + \oint_{\Gamma_\sigma} \boldsymbol{f}_N \cdot (\boldsymbol{v} - \boldsymbol{u}) \, \mathrm{d}S, \quad \forall \boldsymbol{v} \in \mathcal{K}. \tag{4.43}$$

The abstract form of this problem can be expressed as

$$\text{find } \boldsymbol{u} \in \mathcal{K}, \text{ such that } a(\boldsymbol{u}, \boldsymbol{v} - \boldsymbol{u}) \geq l(\boldsymbol{v} - \boldsymbol{u}), \quad \forall \boldsymbol{v} \in \mathcal{K}. \tag{4.44}$$

where a is an elliptic bilinear form and l is the linear form. The problem (4.44) is generally known as *elliptic variational inequality of the first kind* (EVI).

4.3 Elasto-plasticity Analysis: Predictor–Corrector Algorithm

Having developed the variational form for Signorini's problem, we turn to our ISF model (3.100) (cf. p.80). It would be ideal if we could find an equivalent formulation for our model. As we compare the two, it becomes clear that the material law stands in our way as the elasto-plastic constitutive relation cannot be expressed in the form of $\boldsymbol{C}\nabla^s\boldsymbol{u}$ using the previous formulation, thus a similar bilinear form $a(\boldsymbol{u},\boldsymbol{v})$ in (4.44) is not directly attainable.

However, it is possible to formulate the elasto-plastic model, which was developed in section 3.2 using linear isotropic hardening with von Mises yield criterion, as a variational inequality form, or to be accurate, *elliptic variational inequality (EVI) of the second kind* (see p.157 [61]). Therefore, it might turn out to be possible to combine both the EVI of the second kind for the material model and the EVI of the first kind for the contact mechanics as a unified variational inequality formulation. We shall drop this idea here out of computational considerations, instead we embark on considering the energy aspect of this system in section 4.4, before which we first consider the pseudo-time discretisation issue for the elasto-plasticity constitutive relation.

The elasto-plastic constitutive rate equation

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{C}^{ep}\dot{\boldsymbol{\varepsilon}}$$

introduces an additional time dimension. To simplify the computational effort for the ISF model, we first discretise the time dimension while leave with continuous spatial dimensions. The general practice is to use the Euler scheme, in which we employ discretisation among the pseudo-time interval $[t_n, t_{n+1}]$ as

$$\frac{\boldsymbol{\sigma}_{n+1}-\boldsymbol{\sigma}_n}{\Delta t_n} = \boldsymbol{C}^{ep}\frac{\boldsymbol{\varepsilon}_{n+1}-\boldsymbol{\varepsilon}_n}{\Delta t_n} = \frac{1}{\Delta t_n}\boldsymbol{C}^{ep}\Delta\varepsilon_n, \tag{4.45}$$

where Δt_n is the time interval $t_{n+1}-t_n$ and $\Delta\boldsymbol{\varepsilon}_n$ is the difference between two consecutive strain tensors. The strain tensor is a function of the displacement (cf. equation (3.19), p59), thus we can express this dependency in an *incremental form* as

$$\Delta\boldsymbol{\varepsilon}_n = \varepsilon(\boldsymbol{\Delta u}_n).$$

Therefore, equation (4.45) can be simply written as

$$\Delta\boldsymbol{\sigma}_n = \boldsymbol{C}^{ep}\Delta\boldsymbol{\varepsilon}_n. \tag{4.46}$$

In addition, recall that the exact form of $\boldsymbol{C}^{ep}$ in (3.70) is dependent on the stress tensor $\boldsymbol{\sigma}$, so the stress tensor in an explicit integration scheme can be expressed as

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_n + \boldsymbol{C}^{ep}\left(\boldsymbol{\sigma}_n\right)\varepsilon\left(\Delta\boldsymbol{u}_n\right). \tag{4.47}$$

We shall describe a *predictor-corrector* algorithm to obtain a discretised version of $\boldsymbol{C}^{ep}(\boldsymbol{\sigma}_n)$, which is termed as consistent tangent operator. In the light of the shell geometry described by general curvilinear coordinates in the upcoming section 4.7, without causing further complexity, we aim to provide a coordinate-independent formulation and defer the choice of coordinates until actual computation.

Suppose the strain incremental value $\Delta\boldsymbol{\varepsilon}_n$ at a pseudo time instant t_n is known, we employ Euler backward discretisation first by an *elastic trial predictor* step, in which the strain tensor and the accumulative strain are updated using

$$\boldsymbol{\varepsilon}_{n+1}^{e\,\mathrm{trial}} = \boldsymbol{\varepsilon}_n^e + \Delta\boldsymbol{\varepsilon}_n, \tag{4.48}$$

and

$$\kappa_{n+1}^{\mathrm{trial}} = \kappa_n. \tag{4.49}$$

The stress tensor is updated with elasticity law as

$$\boldsymbol{\sigma}_{n+1}^{\mathrm{trial}} = \boldsymbol{C}\boldsymbol{\varepsilon}_{n+1}^{e\,\mathrm{trial}}. \tag{4.50}$$

To exploit the particular properties of the strain and stress tensor, we use the decomposed versions (deviatoric and hydrostatic) as

$$\boldsymbol{\varepsilon}_{n+1}^{e\,\mathrm{trial}} = \boldsymbol{\varepsilon}_{d\,n+1}^{e\,\mathrm{trial}} + \frac{1}{3}\boldsymbol{\varepsilon}_{v\,n+1}^{e\,\mathrm{trial}}\boldsymbol{I}, \tag{4.51}$$

and from (3.36)

$$\begin{aligned} \boldsymbol{s}_{n+1}^{\mathrm{trial}} &= 2\mu\boldsymbol{\varepsilon}_{d\,n+1}^{e\,\mathrm{trial}} \\ p_{n+1}^{\mathrm{trial}} &= 3K\boldsymbol{\varepsilon}_{v\,n+1}^{e\,\mathrm{trial}} \end{aligned}, \tag{4.52}$$

the reason for which will become clearer later. To verify the assumed stress state actually lies in the linear elastic region, we compute the yield function

$$\phi^{\mathrm{trial}}\left(\boldsymbol{\sigma}_{n+1}^{\mathrm{trial}}, \kappa_{n+1}^{\mathrm{trial}}\right) = \sqrt{\frac{3}{2}}\left\|\boldsymbol{s}_{n+1}^{\mathrm{trial}}\right\| - \left(\sigma_y + H\kappa_n\right). \tag{4.53}$$

If $\phi^{\mathrm{trial}} \leq 0$, we accept the elastic trial step and update the corresponding states by

$$\begin{cases} \boldsymbol{\varepsilon}_{n+1}^e = \boldsymbol{\varepsilon}_{n+1}^{e\,\mathrm{trial}} \\ \boldsymbol{s}_{n+1} = \boldsymbol{s}_{n+1}^{\mathrm{trial}} \\ p_{n+1} = p_{n+1}^{\mathrm{trial}} \\ \kappa_{n+1} = \kappa_{n+1}^{\mathrm{trial}} \end{cases}; \tag{4.54}$$

Otherwise, the actual process involves plastic deformation and a corrector step is needed as shown in the following.

In the following *plastic corrector step* (or sometimes known as *return mapping*), first use additive strain decomposition relation between elastic and plastic strain and results

in (3.46), we obtain

$$\begin{aligned}\frac{\varepsilon^e_{n+1}-\varepsilon^e_n}{\Delta t} &= \frac{\varepsilon_{n+1}-\varepsilon_n}{\Delta t} - \frac{\gamma_{n+1}-\gamma_n}{\Delta t}\frac{\partial \phi}{\partial \boldsymbol{\sigma}}\left(\boldsymbol{\sigma}_{n+1}\right),\\ \boldsymbol{\varepsilon}^e_{n+1} &= \boldsymbol{\varepsilon}^e_n + \Delta\varepsilon_n - \Delta\gamma\frac{\partial \phi}{\partial \boldsymbol{\sigma}}\left(\boldsymbol{\sigma}_{n+1}\right),\end{aligned} \tag{4.55}$$

and

$$\boldsymbol{\varepsilon}^e_{n+1} = \boldsymbol{\varepsilon}^{e\,\text{trial}}_{n+1} - \sqrt{\frac{3}{2}}\Delta\gamma\hat{\boldsymbol{n}}_{n+1}, \tag{4.56}$$

$$\kappa_{n+1} = \kappa_n + \Delta\gamma, \tag{4.57}$$

$$\phi\left(\boldsymbol{\sigma}_{n+1}, \kappa_{n+1}\right) = 0, \tag{4.58}$$

where the deviatoric tensor $\hat{\boldsymbol{n}}$ is defined in (3.68) (cf. p70). The discretised plastic multiplier $\Delta\gamma$ is determined from incremental elastic strain relation, incremental internal hardening variable and yield condition, namely, (4.56), (4.57) and (4.58). To solve these three equations, we first use the deviatoric decomposition for the elastic strain formulated in (4.56)

$$\boldsymbol{\varepsilon}^e_{d\,n+1} = \mathsf{I}_{\mathsf{d}}\boldsymbol{\varepsilon}^e_{n+1} = \boldsymbol{\varepsilon}^{e\,\text{trial}}_{d\,n+1} - \sqrt{\frac{3}{2}}\Delta\gamma\hat{\boldsymbol{n}}_{n+1}, \tag{4.59}$$

where the deviatoric nature of the von-Mises yield criterion is applied. For the corresponding deviatoric stress component, we have

$$\begin{aligned}\boldsymbol{s}_{n+1} &= 2\mu\boldsymbol{\varepsilon}^e_{d\,n+1} - 2\mu\sqrt{\frac{3}{2}}\Delta\gamma\hat{\boldsymbol{n}}_{n+1}\\ &= \boldsymbol{s}^{\text{trial}}_{n+1} - 2\mu\sqrt{\frac{3}{2}}\Delta\gamma\frac{\boldsymbol{s}_{n+1}}{\|\boldsymbol{s}_{n+1}\|}\end{aligned} \tag{4.60}$$

Rearrange (4.60) as

$$\left(1 + 2\mu\sqrt{\frac{3}{2}}\frac{\Delta\gamma}{\|\boldsymbol{s}_{n+1}\|}\right)\boldsymbol{s}_{n+1} = \boldsymbol{s}^{\text{trial}}_{n+1}. \tag{4.61}$$

As we can see, $\boldsymbol{s}_{n+1}$ and $\boldsymbol{s}^{\text{trial}}_{n+1}$ are co-linear, which implies

$$\frac{\boldsymbol{s}_{n+1}}{\|\boldsymbol{s}_{n+1}\|} = \frac{\boldsymbol{s}^{\text{trial}}_{n+1}}{\|\boldsymbol{s}^{\text{trial}}_{n+1}\|}. \tag{4.62}$$

We also notice that

$$1 + 2\mu\sqrt{\frac{3}{2}}\frac{\Delta\gamma}{\|\boldsymbol{s}_{n+1}\|} > 1, \tag{4.63}$$

so stress tensor $\boldsymbol{s}_{n+1}$ is scaled relative to $\boldsymbol{s}^{\text{trial}}_{n+1}$. Substituting (4.62) into (4.60), yields

$$\boldsymbol{s}_{n+1} = \left(1 - 2\mu\sqrt{\frac{3}{2}}\frac{\Delta\gamma}{\|\boldsymbol{s}^{\text{trial}}_{n+1}\|}\right)\boldsymbol{s}^{\text{trial}}_{n+1}, \tag{4.64}$$

where the scaling factor satisfies

$$0 < 1 - 2\mu\sqrt{\frac{3}{2}}\frac{\Delta\gamma}{\|\boldsymbol{s}_{n+1}^{\text{trial}}\|} < 1$$

as a result of (4.63). The yield function then can be evaluated as

$$\begin{aligned}
\phi(\boldsymbol{\sigma}_{n+1}, \kappa_{n+1}) &= \sqrt{\frac{3}{2}}\,\|\boldsymbol{s}_{n+1}\| - (\sigma_y + H\kappa_{n+1}) \\
&= \sqrt{\frac{3}{2}}\left(1 - 2\mu\sqrt{\frac{3}{2}}\frac{\Delta\gamma}{\|\boldsymbol{s}_{n+1}^{\text{trial}}\|}\right)\left\|\boldsymbol{s}_{n+1}^{\text{trial}}\right\| - (\sigma_y + H\kappa_n + H\Delta\gamma) \\
&= \sqrt{\frac{3}{2}}\left\|\boldsymbol{s}_{n+1}^{\text{trial}}\right\| - (\sigma_y + H\kappa_n) - 3\mu\Delta\gamma - H\Delta\gamma \\
&= \phi^{\text{trial}} - (3\mu + H)\Delta\gamma.
\end{aligned} \tag{4.65}$$

The discretised plastic multiplier $\Delta\gamma$ is worked out by setting

$$\phi(\boldsymbol{\sigma}_{n+1}, \kappa_{n+1}) = 0,$$

giving

$$\Delta\gamma = \frac{\phi^{\text{trial}}}{3\mu + H}. \tag{4.66}$$

So the stress should be updated with

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{s}_{n+1} + p_{n+1}\boldsymbol{I} = \boldsymbol{s}_{n+1} + 3K\boldsymbol{\varepsilon}_{v\,\text{n+1}}\boldsymbol{I}, \tag{4.67}$$

where

$$p_{n+1} = p_{n+1}^{\text{trial}}$$

is unaffected in the corrector step. Also $\boldsymbol{s}_{n+1}$, $\boldsymbol{\varepsilon}_{n+1}^{e}$ and κ_{n+1} are updated substituting with values (4.66) into (4.64), (4.56) and (4.57) respectively.

Next we consider the elasto-plastic tensor $\boldsymbol{C}^{ep}$ as a derivative

$$\boldsymbol{C}^{ep} = \frac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\boldsymbol{\varepsilon}_{n+1}}.$$

In the elastic range, this value is the elastic tensor $\boldsymbol{C}$. In the plastic range, however, we need a bit further calculation. Note that the following relation

$$\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_{n+1}^{e\,\text{trial}} + \boldsymbol{\varepsilon}_{n+1}^{p} \tag{4.68}$$

holds, where $\boldsymbol{\varepsilon}_{n+1}^{p}$ is independent of $\boldsymbol{\varepsilon}_{n+1}^{e\,\text{trial}}$ (dependent on the internal variable instead). So we can compute $\boldsymbol{C}^{ep}$ by using

$$\boldsymbol{C}^{ep} = \frac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\boldsymbol{\varepsilon}_{n+1}} = \frac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\boldsymbol{\varepsilon}_{n+1}^{e\,\text{trial}}}.$$

Since the updated stress tensor can be expressed as

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{C}\left(\boldsymbol{\varepsilon}^{e\,\text{trial}}_{n+1} - \sqrt{\frac{3}{2}}\Delta\gamma\hat{\boldsymbol{n}}\right),$$

therefore we have the derivative

$$\frac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\boldsymbol{\varepsilon}^{e\,\text{trial}}_{n+1}} = \boldsymbol{C} - \sqrt{\frac{3}{2}}\Delta\gamma\left(\frac{\partial\boldsymbol{C}\hat{\boldsymbol{n}}}{\partial\boldsymbol{\varepsilon}^{e\,\text{trial}}_{n+1}}\right) - \sqrt{\frac{3}{2}}\frac{\partial\Delta\gamma}{\partial\boldsymbol{\varepsilon}^{e\,\text{trial}}_{n+1}} \otimes (\boldsymbol{C}\hat{\boldsymbol{n}}). \tag{4.69}$$

To evaluate $\dfrac{\partial\boldsymbol{C}\hat{\boldsymbol{n}}}{\partial\boldsymbol{\varepsilon}^{e\,\text{trial}}_{n+1}}$, with the aid of the directional derivative defined in (A.22) and the chain rule (A.20), it can be expressed as

$$\frac{\partial\boldsymbol{C}\hat{\boldsymbol{n}}}{\partial\boldsymbol{\varepsilon}^{e\,\text{trial}}_{n+1}} : \boldsymbol{\tau} = \frac{\partial\boldsymbol{C}\hat{\boldsymbol{n}}(\boldsymbol{s}^{\text{trial}}_{n+1})}{\partial\boldsymbol{s}^{\text{trial}}_{n+1}} : \left(\frac{\partial\boldsymbol{s}^{\text{trial}}_{n+1}}{\partial\boldsymbol{\varepsilon}^{e\,\text{trial}}_{n+1}} : \boldsymbol{\tau}\right). \tag{4.70}$$

With results in Appendix (A.23) we can obtain

$$\frac{\partial\boldsymbol{C}\hat{\boldsymbol{n}}}{\partial\boldsymbol{\varepsilon}^{e\,\text{trial}}_{n+1}} : \boldsymbol{\tau} = \left[\frac{\boldsymbol{C} - (\boldsymbol{C}\hat{\boldsymbol{n}}) \otimes \hat{\boldsymbol{n}}}{\|\boldsymbol{s}^{\text{trial}}_{n+1}\|}\right] : (2\mu\mathsf{I}_{\mathsf{d}} : \boldsymbol{\tau}), \tag{4.71}$$

where

$$\boldsymbol{C} : (2\mu\mathsf{I}_{\mathsf{d}}) : \boldsymbol{\tau} = (2\mu\mathsf{I}_{\mathsf{d}} + K\boldsymbol{I}\otimes\boldsymbol{I}) : (2\mu\mathsf{I}_{\mathsf{d}}) : \boldsymbol{\tau} = 4\mu^2\mathsf{I}_{\mathsf{d}} : \boldsymbol{\tau} + 2\mu K\boldsymbol{I}\otimes\boldsymbol{I} : (\mathsf{I}_{\mathsf{d}} : \boldsymbol{\tau}). \tag{4.72}$$

Using (A.16)

$$\boldsymbol{I}\otimes\boldsymbol{I} : (\mathsf{I}_{\mathsf{d}} : \boldsymbol{\tau}) = \boldsymbol{I}\text{trace}(\mathsf{I}_{\mathsf{d}} : \boldsymbol{\tau}) = \boldsymbol{0}.$$

Using (A.9)

$$\begin{aligned}(\boldsymbol{C}\hat{\boldsymbol{n}}) \otimes \hat{\boldsymbol{n}} : (2\mu\mathsf{I}_{\mathsf{d}}) : \boldsymbol{\tau} &= (\boldsymbol{C}\hat{\boldsymbol{n}})\,(\hat{\boldsymbol{n}} : (2\mu\mathsf{I}_{\mathsf{d}}) : \boldsymbol{\tau}) = (\boldsymbol{C}\hat{\boldsymbol{n}})\,((2\mu\mathsf{I}_{\mathsf{d}}) : \hat{\boldsymbol{n}} : \boldsymbol{\tau}) \\ &= (\boldsymbol{C}\hat{\boldsymbol{n}})\,(2\mu\hat{\boldsymbol{n}} : \boldsymbol{\tau}) \\ &= 2\mu(\boldsymbol{C}\hat{\boldsymbol{n}} \otimes \hat{\boldsymbol{n}}) : \boldsymbol{\tau},\end{aligned} \tag{4.73}$$

where

$$(\boldsymbol{C}\hat{\boldsymbol{n}}) \otimes \hat{\boldsymbol{n}} = [(2\mu\mathsf{I}_{\mathsf{d}} + K\boldsymbol{I}\otimes\boldsymbol{I})\hat{\boldsymbol{n}}] \otimes \hat{\boldsymbol{n}} = (2\mu\hat{\boldsymbol{n}} + \boldsymbol{0}) \otimes \hat{\boldsymbol{n}} = 2\mu\hat{\boldsymbol{n}} \otimes \hat{\boldsymbol{n}}.$$

Hence,

$$\frac{\partial\boldsymbol{C}\hat{\boldsymbol{n}}}{\partial\boldsymbol{\varepsilon}^{e\,\text{trial}}_{n+1}} = \frac{4\mu^2(\mathsf{I}_{\mathsf{d}} - \hat{\boldsymbol{n}} \otimes \hat{\boldsymbol{n}})}{\|\boldsymbol{s}^{\text{trail}}_{n+1}\|}. \tag{4.74}$$

Then, based on (4.66), we evaluate

$$\frac{\partial\Delta\gamma}{\partial\boldsymbol{\varepsilon}^{e\,\text{trial}}_{n+1}}$$

as

$$\frac{1}{3\mu + H}\frac{\partial\Phi^{\text{trial}}}{\partial\boldsymbol{\varepsilon}^{e\,\text{trial}}_{n+1}} = \frac{\sqrt{3/2}}{3\mu + H}\frac{\left\|\boldsymbol{s}^{\text{trial}}_{n+1}\right\|}{\partial\boldsymbol{\varepsilon}^{e\,\text{trial}}_{n+1}} = \frac{2\mu\sqrt{\frac{3}{2}}}{3\mu + H}\hat{\boldsymbol{n}}. \tag{4.75}$$

By collecting the above terms, eventually we have

$$\frac{\partial \boldsymbol{\sigma}_{n+1}}{\partial \boldsymbol{\varepsilon}^{e\,\mathrm{trial}}_{n+1}} = \boldsymbol{C} - \sqrt{\frac{3}{2}}\Delta\gamma 4\mu^2 \frac{\mathsf{I}_\mathsf{d} - \hat{\boldsymbol{n}}\otimes\hat{\boldsymbol{n}}}{\|\boldsymbol{s}^{\mathrm{trail}}_{n+1}\|} - \frac{6\mu^2}{3\mu + H}\hat{\boldsymbol{n}}\otimes\hat{\boldsymbol{n}}. \tag{4.76}$$

The whole predictor-corrector algorithm is summarised in Table 4.1.

Known variables from previous step: $\boldsymbol{\varepsilon}^e_n$, κ_n, $\boldsymbol{s}_n$, p_n

1. Given $\Delta\boldsymbol{\varepsilon}_n$, compute elastic predictor as:

$$\boldsymbol{\varepsilon}^{e\,\mathrm{trial}}_{n+1} = \boldsymbol{\varepsilon}^e_n + \Delta\boldsymbol{\varepsilon}_n, \quad \kappa^{\mathrm{trial}}_{n+1} = \kappa_n,$$

$$\boldsymbol{\varepsilon}^{e\,\mathrm{trial}}_{d\,n+1} = \mathsf{I}_\mathsf{d}\boldsymbol{\varepsilon}^{e\,\mathrm{trial}}_{n+1}, \quad \varepsilon^{e\,\mathrm{trial}}_{v\,n+1} = \boldsymbol{I}\colon\boldsymbol{\varepsilon}^{e\,\mathrm{trial}}_{n+1},$$
$$\boldsymbol{s}^{\mathrm{trial}}_{n+1} = 2\mu\boldsymbol{\varepsilon}^{e\,\mathrm{trial}}_{d\,n+1}, \quad p^{\mathrm{trial}}_{n+1} = 3K\boldsymbol{\varepsilon}^{e\,\mathrm{trial}}_{v\,n+1}.$$

2. Compute yield function

$$\phi^{\mathrm{trial}} = \sqrt{\frac{3}{2}}\left\|\boldsymbol{s}^{\mathrm{trial}}_{n+1}\right\| - (\sigma_y + H\kappa_n),$$

3. If $\phi^{\mathrm{trial}} \le 0$, update tensors

$$\boldsymbol{\varepsilon}^e_{n+1} = \boldsymbol{\varepsilon}^{e\,\mathrm{trial}}_{n+1},\ \boldsymbol{s}_{n+1} = \boldsymbol{s}^{\mathrm{trial}}_{n+1},\ p_{n+1} = p^{\mathrm{trial}}_{n+1},\ \kappa_{n+1} = \kappa^{\mathrm{trial}}_{n+1},$$

set

$$\boldsymbol{C}^{ep} = \boldsymbol{C},$$

and end of the programme;

4. Otherwise, run a plastic corrector by computing

$$\Delta\gamma = \frac{\phi^{\mathrm{trial}}}{3\mu + H},\ \hat{\boldsymbol{n}}_{n+1} = \frac{\boldsymbol{s}^{\mathrm{trial}}_{n+1}}{\|\boldsymbol{s}^{\mathrm{trial}}_{n+1}\|}$$

and update

$$\boldsymbol{s}_{n+1} = \left(1 - 2\mu\sqrt{\frac{3}{2}}\frac{\Delta\gamma}{\|\boldsymbol{s}^{\mathrm{trial}}_{n+1}\|}\right)\boldsymbol{s}^{\mathrm{trial}}_{n+1},\ p_{n+1} = p^{\mathrm{trial}}_{n+1},\ \kappa_{n+1} = \kappa_n + \Delta\gamma,$$

$$\varepsilon^e_{n+1} = \varepsilon^{e\,\mathrm{trial}}_{n+1} - \sqrt{\frac{3}{2}}\Delta\gamma\hat{\boldsymbol{n}}_{n+1},$$

set

$$\boldsymbol{C}^{ep} = \boldsymbol{C} - \sqrt{\frac{3}{2}}\Delta\gamma 4\mu^2 \frac{\mathsf{I}_\mathsf{d} - \hat{\boldsymbol{n}}\otimes\hat{\boldsymbol{n}}}{\|\boldsymbol{s}^{\mathrm{trail}}_{n+1}\|} - \frac{6\mu^2}{3\mu + H}\hat{\boldsymbol{n}}\otimes\hat{\boldsymbol{n}}.$$

Table 4.1 Elasto-plasticity analysis: predictor-corrector algorithm

4.4 ISF Model as A Minimisation Problem

The abstract elastic contact problem (4.44) is well known to be identical to a minimisation problem: if we define

$$\mathcal{J}(\boldsymbol{v}) = \frac{1}{2}a(\boldsymbol{v},\boldsymbol{v}) - l(\boldsymbol{v}),$$

$$\text{find } \boldsymbol{u} \in \mathcal{K}, \text{ such that } \mathcal{J}(\boldsymbol{u}) \leq \mathcal{J}(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in \mathcal{K}, \tag{4.77}$$

the proof of which can be found in most books on mathematical theory of variational inequality [69]. We may notice the similarity with (4.15) and the heart of the variational inequality lies in the convex set $\mathcal{K}$, upon which the problem is posed; the problem becomes a variational equation if $\mathcal{K}$ is a linear space. The physical meaning of the functional $\mathcal{J}(\boldsymbol{u})$ is the total system energy excluding the contribution from contact and the convex set $\mathcal{K}$ (defined in (4.18)) represents the contact boundary constraints.

Notice that our ISF model differs with Signorini's problem in the material law, so we can imagine a similar constrained optimisation formulation can be posed for a computational ISF model. Incorporating the results from section 4.3, we first consider the energy at pseudo-time instant t_{n+1}, excluding the contribution from the contact boundary, denoted by $\mathcal{J}$, can be expressed as

$$\mathcal{J}(\boldsymbol{u}_{n+1}) = \int_V \frac{1}{2}\boldsymbol{\sigma}_{n+1} : \boldsymbol{\varepsilon}_{n+1}\,\mathrm{d}V - \int_V \boldsymbol{f}_B \cdot \boldsymbol{u}_{n+1}\,\mathrm{d}V - \oint_{\Gamma_\sigma} \boldsymbol{f}_N \cdot \boldsymbol{u}_{n+1}\,\mathrm{d}S. \tag{4.78}$$

The contact boundary condition can be described with a convex set $\mathcal{K}$

$$\mathcal{K} = \{\boldsymbol{v} \in \mathcal{V} | \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_u, \boldsymbol{v}_\nu \leq \ell \text{ on } \Gamma_c\}, \tag{4.79}$$

where $\mathcal{V}$ is a linear space, so that we have

$$\boldsymbol{u}_{n+1} \in \mathcal{K}.$$

Given previously known displacement $\boldsymbol{u}_n$, the optimisation problem for ISF model becomes one to find the admissible displacement $\boldsymbol{u}_{n+1}$ that lies in $\mathcal{K}$ such that the functional $\mathcal{J}$ obtains the minimum, which can be expressed as

$$\arg \min_{\boldsymbol{u}_{n+1} \in \mathcal{K}} \mathcal{J}(\boldsymbol{u}_{n+1}). \tag{4.80}$$

Using the incremental forms

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \Delta\boldsymbol{u}, \quad \boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_n + \Delta\boldsymbol{\varepsilon}_n, \quad \text{and } \boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_n + \Delta\boldsymbol{\sigma}_n,$$

the functional $\mathcal{J}$ in (4.78) can be split into two parts

$$\mathcal{J}_{n+1} = \mathcal{J}(\boldsymbol{u}_{n+1}) = \mathcal{J}_n + F_n, \tag{4.81}$$

where $\mathcal{J}_n = \mathcal{J}(\boldsymbol{u}_n)$ and

$$F_n = \int_V \frac{1}{2}\Delta\boldsymbol{\sigma}_n : \Delta\boldsymbol{\varepsilon}_n \,\mathrm{d}V + \int_V \Delta\boldsymbol{\sigma}_n : \boldsymbol{\varepsilon}_n \,\mathrm{d}V - \int_V \boldsymbol{f}_B \cdot \Delta\boldsymbol{u}_n \,\mathrm{d}V - \oint_{\Gamma_\sigma} \boldsymbol{f}_N \cdot \Delta\boldsymbol{u}_n \,\mathrm{d}S. \quad (4.82)$$

Note that the symmetric property of tensor $\boldsymbol{C}^{ep}$ has been used from (4.78) to (4.82). Using (4.81), the optimisation problem (4.80) can be reduced to

$$\arg \min_{\Delta\boldsymbol{u}_n \in \mathcal{C}_n} F_n(\Delta\boldsymbol{u}_n), \quad (4.83)$$

where the convex set $\mathcal{C}_n$ takes the form

$$\mathcal{C}_n = \{\boldsymbol{v} \in \mathcal{V} | \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_u, \boldsymbol{v}_{\boldsymbol{\nu}} \leq \ell - (\boldsymbol{u}_n)_{\boldsymbol{\nu}} \text{ on } \Gamma_c\} \quad (4.84)$$

to ensure that $\boldsymbol{u}_{n+1}$ and $\boldsymbol{u}_n$ are in the same convex set $\mathcal{K}$.

A general constrained optimisation problem in the infinite dimensional space can be expressed as

$$\min F(x) \text{ over } x \in M, \quad (4.85)$$

where $F : M \subseteq X \to [-\infty, +\infty]$, $M \neq \varnothing$. The rigorous existence conditions given in [84] state that problem (4.85) has a solution, if i) X is a real reflexive Banach space. ii) M is bounded and weak sequentially closed, iii) F is weak sequentially lower semicontinuous on M. The uniqueness of problem (4.85), i.e. F has at most one minimum on M, can be guaranteed if i) M is a convex subset of the linear space X. ii) F is strictly convex, i.e., $F((1-t)u + tv) < (1-t)F(u) + tF(v)$ holds for all $u, v \in M$, $u \neq v$, and all $t \in (0, 1)$. The optimality condition to (4.85) is known as a variational inequality formulation

$$\langle F'(x^*), x - x^* \rangle \geq 0, \quad \forall x \in C, \quad (4.86)$$

where F' is the F-derivative or Fréchet derivative defined as

$$F(u_0 + h) = F(u_0) + \langle F'(u_0), h \rangle + o(\|h\|) \text{ as } h \to 0,$$

and x^* is the optimal solution. The variational inequality formulation is not computationally easy to handle for problem (4.83). Instead, we will describe an alternative method based on Lagrange multiplier theory in the next section.

4.5 Augmented Lagrangian Method

It is a common technique to use Lagrange multiplier to transform a constrained optimisation problem to an unconstrained one in finite dimension and the necessary optimality condition is given by the Karush-Kuhn-Tucker (KKT) conditions [85]. We also know that the *penalty method* for constrained optimisation in finite dimension offers a simpler structure, in which a penalty term that is positive when the current iteration violates

the constraint and zero otherwise is added to the original objective function. In infinite dimension, an equivalent Lagrangian formulation is also feasible (cf. [86]). In this work, however, we consider using the *Augmented Lagrangian Method* (ALM) for our constrained problem, which combines both the Lagrange multiplier method and the penalty method.

The Augmented Lagrangian method is first studied in the finite dimensional space and can reduce the possibility of ill-conditioning and has a faster convergence rate than Lagrange multiplier method, see [87, 88, 85], and later brought into the infinite dimensional space by a series of investigations as in [89, 90]. This method has appeared a number of times in the contact mechanics literature such as [91, 92] and etc, although the reference given here often derived in a finite dimensional flavour. In [93], the ALM has been shown as a powerful nonsmooth optimisation tool.

Rewrite the constrained optimisation problem (4.83) as

$$\min F_n(\Delta \boldsymbol{u}_n), \quad \text{subject to } \Delta \boldsymbol{u}_n \in \mathcal{C}_n, \tag{4.87}$$

where

$$\mathcal{C}_n = \{\boldsymbol{v} \in \mathcal{V} | \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_u, \boldsymbol{v}_{\boldsymbol{\nu}} \leq \ell_n \text{ on } \Gamma_c\},$$

and ℓ_n is defined as

$$\ell_n = \ell - (\boldsymbol{u}_n)_{\boldsymbol{\nu}}. \tag{4.88}$$

Let

$$C(\Delta \boldsymbol{u}_n) = \Delta \boldsymbol{u}_{n\boldsymbol{\nu}} - \ell_n, \tag{4.89}$$

and also we slightly abuse the notation by dropping all the incremental form, implied by notation Δ, and the pseudo-time subscript n, so the constrained problem (4.87) can be written as

$$\min F(\boldsymbol{u}), \quad \text{subject to } C(\boldsymbol{u}) \leq 0 \text{ on } \Gamma_c. \tag{4.90}$$

Note that the equality constraint on Γ_u, where $\boldsymbol{u} = \boldsymbol{0}$, can be prescribed in the actual computation. For the inequality constraint $C(\boldsymbol{u}) \leq 0$, we introduce the slack variable (functional) q, such that

$$C(\boldsymbol{u}) - q = 0, \qquad q \leq 0 \quad \text{on } \Gamma_c. \tag{4.91}$$

The augmented Lagrangian for (4.90), denoted by $\mathcal{L}_A$, can be expressed simply as

$$\mathcal{L}_A = F + \int_{\Gamma_c} \frac{1}{2\epsilon}(C(\boldsymbol{u}) - q)^2 + \lambda(C(\boldsymbol{u}) - q)\,\mathrm{d}S, \quad \text{s.t. } q \leq 0, \tag{4.92}$$

where ϵ is the penalty parameter (always positive) and λ denotes the Lagrange multiplier (a functional here). So the constrained problem (4.90) becomes

$$\min \mathcal{L}_A, \quad \text{over } q \leq 0. \tag{4.93}$$

The minimizer of (4.93) with respect to q can be obtained as

$$q^* = \min\{C(\boldsymbol{u}) + \epsilon\lambda, 0\}.$$

Let

$$\tilde{C}(\boldsymbol{u}) = C(\boldsymbol{u}) - q^* + \lambda\epsilon = \max\{0, C(\boldsymbol{u}) + \lambda\epsilon\}, \tag{4.94}$$

so that the Lagrangian $\mathcal{L}_A$ can be reformulated as

$$\mathcal{L}_A = F(\boldsymbol{u}) + G(\boldsymbol{u}), \tag{4.95}$$

where

$$G(\boldsymbol{u}) = \int_{\Gamma_c} \left[\frac{1}{2\epsilon} \tilde{C}(\boldsymbol{u})^2 - \frac{\epsilon\lambda^2}{2} \right] \mathrm{d}S. \tag{4.96}$$

The augmented Lagrangian method of the first order to solve the constrained problem (4.90) is the following iterative algorithm [90]:

- Initialise Lagrange multiplier (λ^0)
- Solve problem

$$\min \mathcal{L}_A(\boldsymbol{u}^{(i)}, \lambda^{(i)}) \tag{4.97}$$

  given in (4.95),
- Update the Lagrange multiplier with

$$\lambda^{(i+1)} = \max\{0, \lambda^{(i)} + C(\boldsymbol{u}^{(i)})/\epsilon\}. \tag{4.98}$$

If define

$$L(\lambda) = \min \mathcal{L}_A(\boldsymbol{u}, \lambda),$$

we can have

$$\frac{\mathrm{d}L}{\mathrm{d}\lambda} = \max\{0, C(\boldsymbol{u}) + \lambda\epsilon\},$$

so the above Lagrange multiplier update (4.98) represents a gradient method for maximising $L(\lambda)$. The unconditional convergence of the ALM algorithm is proofed in [90].

To complete the ALM algorithm, we still need a method to obtain the solution to the minimisation problem (4.97). The first order necessary condition for optimality requires the (Fréchet) derivative to be set as zero, which can be expressed as

$$\mathcal{L}'_A = F'(\boldsymbol{u}) + G'(\boldsymbol{u}) = \boldsymbol{0}, \tag{4.99}$$

where

$$G'(\boldsymbol{u}) = \int_{\Gamma_c} \frac{1}{\epsilon} \tilde{C}'(\boldsymbol{u}) \tilde{C}(\boldsymbol{u}) \, \mathrm{d}S.$$

To obtain a succinct expression for $F'(\boldsymbol{u})$, we introduce a third-order tensor $\boldsymbol{B}$ so that

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(\boldsymbol{u}) = \boldsymbol{B}\boldsymbol{u}, \quad \text{and} \quad \frac{\mathrm{d}\boldsymbol{\varepsilon}}{\mathrm{d}\boldsymbol{u}} = \boldsymbol{B}. \tag{4.100}$$

Using above equation (4.100) and the symmetry of the elasto-plasticity tensor $\boldsymbol{C}^{ep}$, we can obtain F' from (4.82) as (note the notational change introduced earlier cf. p100)

$$F'(\boldsymbol{u}) = \int_V \boldsymbol{B}\boldsymbol{C}^{ep}\boldsymbol{B}\boldsymbol{u} \, \mathrm{d}V + \int_V \boldsymbol{\varepsilon}_n \boldsymbol{C}^{ep}\boldsymbol{B} \, \mathrm{d}V - \int_V \boldsymbol{f}_B \, \mathrm{d}V - \oint_{\Gamma_\sigma} \boldsymbol{f}_N \, \mathrm{d}S. \tag{4.101}$$

From (4.94), we have

$$\tilde{C}'(\boldsymbol{u}) = \max\{0, C'(\boldsymbol{u})\}. \tag{4.102}$$

Note that 0 denotes an all-zero valued functional, i.e., $0(\boldsymbol{x})$, where $\boldsymbol{x}$ denotes the locations. Using (4.102), we can obtain

$$G'(\boldsymbol{u}) = \int_{\Gamma_c} C'(\boldsymbol{u}) \max\{0, \lambda + C(\boldsymbol{u})/\epsilon\} \, \mathrm{d}S. \tag{4.103}$$

Given that

$$C(\boldsymbol{u}) = \boldsymbol{u}_{\boldsymbol{\nu}} - \ell_n = \boldsymbol{u}_{\boldsymbol{\nu}} - (\ell - (\boldsymbol{u}_n)_{\boldsymbol{\nu}}), \tag{4.104}$$

so

$$C'(\boldsymbol{u}) = \boldsymbol{n}, \tag{4.105}$$

where $\boldsymbol{n}$ denotes the outward normal to the contact point.

We can see now the nonlinear operator $\mathcal{L}'_A$ is nonsmooth. To solve the equation

$$\mathcal{L}'_A = \mathbf{0},$$

in the next section, we will first discretise it using a finite element method before employing a semismooth Newton's method to obtain the numerical solution.

4.6 Finite Element Discretisation and Semismooth Newton Method

In the last section, it becomes clear that the key to solving the ISF model problem (4.80) is equation (4.99). Since (4.99) is posed in the infinite dimension, we first discretise it into finite dimension using finite element method before a nonlinear programming method is proposed.

We first discretise the unknown variable $\boldsymbol{u}$, a first-order tensor, with a second-order tensor $\boldsymbol{N}$,

$$\boldsymbol{N}(\boldsymbol{x}) = \boldsymbol{N}_{ik}\boldsymbol{e}_i \otimes \tilde{\boldsymbol{e}}_k, \tag{4.106}$$

where $\boldsymbol{e}_i$ is a coordinate basis and $\tilde{\boldsymbol{e}}_k$ is a basis associated with an array (dyadic basis), so that

$$\boldsymbol{u} = \boldsymbol{N}\boldsymbol{u}_h, \tag{4.107}$$

where $\boldsymbol{u}_h$ is the unknown values at the mesh nodes. The exact form of $\boldsymbol{N}(\boldsymbol{x})$ will be determined later and it is known as the shape function in the finite element method. The third-order tensor $\boldsymbol{B}(\boldsymbol{x})$ defined in (4.100) thus takes the form

$$\boldsymbol{B}_{ijk} = \frac{1}{2}(\boldsymbol{N}_{ik,j} + \boldsymbol{N}_{jk,i}), \tag{4.108}$$

so that

$$\boldsymbol{B}\boldsymbol{u}_h = \nabla^s(\boldsymbol{N}\boldsymbol{u}_h),$$

and

$$\boldsymbol{\varepsilon} = \boldsymbol{B}\boldsymbol{u}_h. \tag{4.109}$$

Note that the third basis of tensor $\boldsymbol{B}$ is different from the previous two and denoted by a new basis direction as $\tilde{\boldsymbol{e}}$, tensor $\boldsymbol{B}$ can be expressed as

$$\boldsymbol{B} = \boldsymbol{B}_{ijk}\boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \tilde{\boldsymbol{e}}_k.$$

Using (4.107), we can have the discretised $F'(\boldsymbol{u})$ as

$$F'_h(\boldsymbol{u}) = \boldsymbol{f}^{\text{int}}(\boldsymbol{u}_h) - \boldsymbol{f}^{\text{ext}}, \tag{4.110}$$

where the subscript h indicates a discretised version,

$$\begin{aligned} \boldsymbol{f}^{\text{int}}(\boldsymbol{u}_h) &= \int_V \boldsymbol{B}\boldsymbol{C}^{ep}\boldsymbol{B}\boldsymbol{N}\boldsymbol{u}_h \,\mathrm{d}V, \\ \boldsymbol{f}^{\text{ext}} &= \int_v \boldsymbol{f}_B \,\mathrm{d}V + \oint_{\Gamma_\sigma} \boldsymbol{f}_N \,\mathrm{d}S - \int_V \boldsymbol{\varepsilon}_n \boldsymbol{C}^{ep}\boldsymbol{B} \,\mathrm{d}V. \end{aligned} \tag{4.111}$$

In addition, we can have the discretised $G'(\boldsymbol{u})$, denoted by $\boldsymbol{f}^{\text{c}}$:

$$\boldsymbol{f}^{\text{c}}(\boldsymbol{u}_h) = G'_h(\boldsymbol{u}_h) = \int_{\Gamma_c} \boldsymbol{n}\bar{\lambda}(\boldsymbol{u}_h) \,\mathrm{d}S, \tag{4.112}$$

where

$$\bar{\lambda}(\boldsymbol{u}_h) = \begin{cases} \lambda + C(\boldsymbol{u}_h)/\epsilon & \text{if} \quad \lambda + C(\boldsymbol{u}_h)/\epsilon > 0, \\ 0 & \text{if} \quad \lambda + C(\boldsymbol{u}_h)/\epsilon \leq 0. \end{cases} \tag{4.113}$$

For the sake of simplicity, we use the notation $\boldsymbol{f}$ to denote the discretised version of $\mathcal{L}'_A$. Based on (4.110) and (4.112), we obtain

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{u}_h) &= F'_h(\boldsymbol{u}_h) + G'_h(\boldsymbol{u}_h), \\ &= \boldsymbol{f}^{\text{int}}(\boldsymbol{u}_h) + \boldsymbol{f}^{\text{c}}(\boldsymbol{u}_h) - \boldsymbol{f}^{\text{ext}}. \end{aligned} \tag{4.114}$$

Note that the Lagrange multiplier λ is a functional on the boundary Γ_c, which requires

discretisation when it comes to computation. Here we briefly introduce a shape function $\boldsymbol{M}$ such that

$$\lambda = \boldsymbol{M}\boldsymbol{\lambda}_h, \tag{4.115}$$

where $\boldsymbol{\lambda}_h$ denotes the unknown values at the mesh nodes and the exact form of $\boldsymbol{M}$ is determined in the numerical implementation scheme. After the discretisations to both incremental displacement $\boldsymbol{u}$ and Lagrange multiplier λ, we can see that a key step in ALM algorithm, which solves the computational ISF model problem, is to solve the nonlinear equation

$$\boldsymbol{f}(\boldsymbol{u}_h) = \boldsymbol{0}. \tag{4.116}$$

Usually, *Newton's method* requires the gradient information about $\boldsymbol{f}$ to solve such a nonlinear equation. The gradient for $\boldsymbol{f}$ is a Jacobian matrix, denoted by K, we have

$$K = \frac{\mathrm{d}\boldsymbol{f}(\boldsymbol{u}_h)}{\mathrm{d}\boldsymbol{u}_h} = \frac{\mathrm{d}\boldsymbol{f}^{\mathrm{int}}(\boldsymbol{u}_h)}{\mathrm{d}\boldsymbol{u}_h} + \frac{\mathrm{d}\boldsymbol{f}^{\mathrm{c}}(\boldsymbol{u}_h)}{\mathrm{d}\boldsymbol{u}_h},$$

where

$$\frac{\mathrm{d}\boldsymbol{f}^{\mathrm{int}}(\boldsymbol{u}_h)}{\mathrm{d}\boldsymbol{u}_h} = \int_V \boldsymbol{B}\boldsymbol{C}^{ep}\boldsymbol{B}\boldsymbol{N}\,\mathrm{d}V.$$

For $\boldsymbol{f}^{\mathrm{c}}(\boldsymbol{u}_h)$, the derivative is piecewise

$$\frac{\mathrm{d}\boldsymbol{f}^{\mathrm{c}}}{\mathrm{d}\boldsymbol{u}_h} = \begin{cases} \int_{\Gamma_c} (\boldsymbol{N}\boldsymbol{n}) \otimes \boldsymbol{n}\,\mathrm{d}S & \text{if} \quad \lambda + C(\boldsymbol{u}_h)/\epsilon > 0, \\ \boldsymbol{0} & \text{if} \quad \lambda + C(\boldsymbol{u}_h)/\epsilon \leq 0. \end{cases} \tag{4.117}$$

Therefore, the nonlinear equation (4.116) is not continuously differentiable, i.e. nonsmooth, however, it satisfies *semismooth* condition. A rigorous definition of semismoothness can be found in p.27 [94]. We will solve the equation (4.116) by semismooth Newton method, before which we present the conventional Newton's method. Newton's method is an iterative algorithm to solve the nonlinear equation $\boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{0}$ by using the following updating scheme:

1. Initialise solution $\boldsymbol{u}^{(0)}$,
2. Compute Jacobian
$$K^{(i)} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}^{(i)}},$$
and the incremental step $\delta\boldsymbol{u}^{(i)}$ from
$$K^{(i)}\delta\boldsymbol{u}^{(i)} = -\boldsymbol{f}(\boldsymbol{u}^i), \tag{4.118}$$
3. Update
$$\boldsymbol{u}^{(i+1)} = \boldsymbol{u}^{(i)} + \delta\boldsymbol{u}^{(i)}$$
until convergence.

In the semismooth Newton method, whose updating scheme is similar to that of Newton's

method but the derivative information for a semismooth function needs special treatment. We follow the definitions given in [95] by first defining the differentiable set for function $\boldsymbol{f}$ as

$$D_{\boldsymbol{f}} = \{\boldsymbol{u} | \boldsymbol{f} \text{ is differentiable at } \boldsymbol{u}\}, \tag{4.119}$$

and the generalised Jacobian of $\boldsymbol{f}$ at $\boldsymbol{u}$ is defined by

$$\partial \boldsymbol{f}(\boldsymbol{u}) = \text{conv}\partial_B \boldsymbol{f}(\boldsymbol{u}), \tag{4.120}$$

where

$$\partial_B \boldsymbol{f}(\boldsymbol{u}) = \left\{ \lim_{\boldsymbol{u}^* \to \boldsymbol{u},\, \boldsymbol{u}^* \in D_{\boldsymbol{f}}} f'(\boldsymbol{u}^*) \right\}.$$

String `conv` in (4.120) stands for the convex hull. Therefore, a generalised semismooth Newton method for (4.116) becomes: having $\boldsymbol{u}_h^{(i)}$, compute $\boldsymbol{u}_h^{(i+1)}$ by

$$\boldsymbol{u}_h^{(i+1)} = \boldsymbol{u}_h^{(i)} - \bar{K}_i^{-1} \boldsymbol{f}(\boldsymbol{u}_h^{(i)}), \tag{4.121}$$

where

$$\bar{K}_i^{-1} \in \partial \boldsymbol{f}(\boldsymbol{u}_h^{(i)}).$$

This completes our discussion of the numerical method for solving the nonlinear equation (4.116).

From the perspective of finite element, all the current formulations derived in this section is element-wise, which means the integration domains V, Γ_c etc are applied to a single element. An assembly procedure is required in the numerical programme to combine all the contributions from each individual element together, which will be described in the next chapter. As we have mentioned before, the notation $\boldsymbol{u}$ actually stands for an incremental form Δu and h indicates discretisation. For mathematical convenience, we may drop the subscript h later when the context for discretised version is obvious.

4.7 Geometry Modelling with Shell Theory

To complete the finite element computation procedure, we need to determine the underlying geometry model of our problem. Since the shape of the metal deforms (changes) subject to the specific tool trajectory, we need to describe the geometry features consistently, so that evaluations of the quantities proposed earlier can be achieved. One possible geometry approximation is by modelling the deformed state as a shell, which is capable of accommodating arbitrary shapes. The behaviour of a shell is usually regarded as a combination of the plate bending and in-plane membrane actions, as shown in Figure 4.1. A simple flat facet shell element scheme with addictive plate element and membrane element was proposed [76], which shows good approximation accuracy but rigorous mathematical analysis, such as the work in [96, 97] strongly suggest the issue of convergence. Thus, here we opt to avoid this type flat facet shell element, but stick with the original math-

ematical derivations and we first review the necessary mathematical foundation for a shell description in the following section.

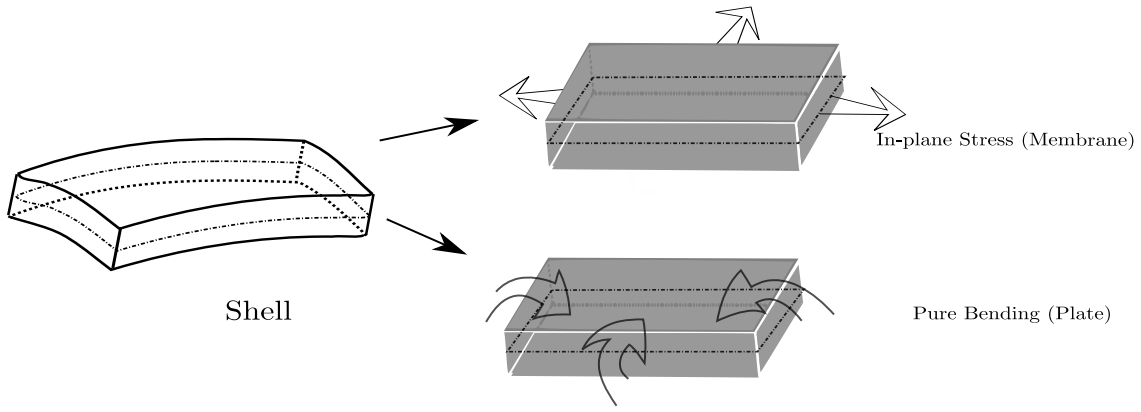


Figure 4.1 Shells regarded as plate bending and membrane action combined.

4.7.1 Definition of a Shell

Shells are naturally described by a general *curvilinear coordinate*, whose associated base vector changes throughout the space, in comparison to a Cartesian coordinates which have fixed base vectors. First we illustrate the use of the curvilinear coordinate to describe the middle surface of a shell $\mathcal{S}$ shown in Figure 4.2. We assume the middle surface can be mapped by $\vec{\phi} : \Omega \to \mathcal{S}$, where Ω is called a *reference domain* in two-dimensions and $\mathcal{S}$ sits in the usual Euclidean space. Assume the map $\vec{\phi}$ is regular, we can have linearly

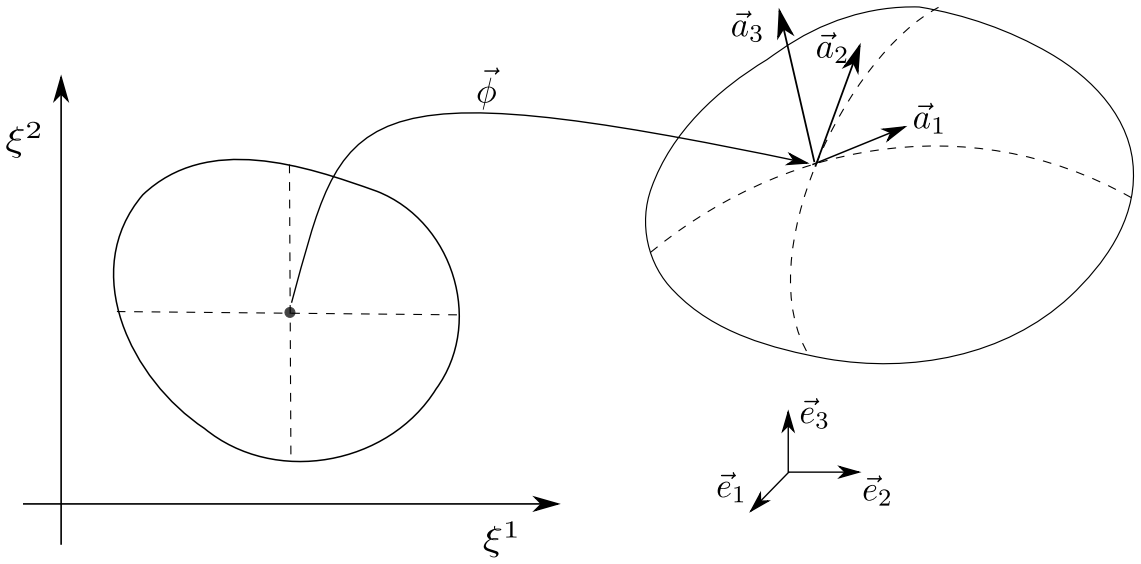


Figure 4.2 Definition of the middle surface of a thin shell

independent vectors

$$\vec{a}_\alpha = \vec{\phi}_{,\alpha} = \frac{\partial \vec{\phi}(\xi^1, \xi^2)}{\partial \xi^\alpha}, \quad (\alpha = 1, 2) \tag{4.122}$$

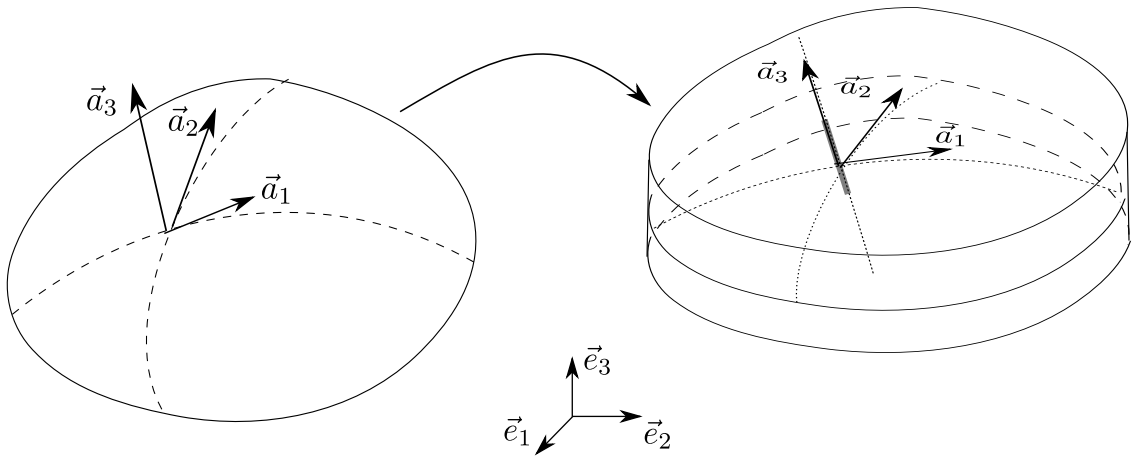


Figure 4.3 From thin shell to thick shell

and a normal vector to the tangent plane defined by $\vec{a}_\alpha$

$$\vec{a}_3 = \frac{\vec{a}_1 \times \vec{a}_2}{|\vec{a}_1 \times \vec{a}_2|}. \tag{4.123}$$

Vectors $\vec{a}_i, (i = 1, 2, 3)$ define a local coordinate frame for the middle surface, depicted in Figure 4.2, and are called *covariant basis* since it is associated with point $\vec{\phi}(\xi^1, \xi^2)$.

Having defined a thin shell, a general thick shell with thickness dimension shown in Figure 4.3 can be defined as mapping

$$\vec{\Phi}(\xi^1, \xi^2, \xi^3) = \vec{\phi}(\xi^1, \xi^2) + \xi^3 \vec{a}_3(\xi^1, \xi^2), \tag{4.124}$$

where a third coordinate ξ^3 is introduced to account for the thickness dimension. Similarly to the middle surface, we have a generised local basis expressed by

$$\vec{g}_m = \frac{\partial \Phi(\xi^1, \xi^2, \xi^3)}{\partial \xi^m}, \quad m = 1, 2, 3. \tag{4.125}$$

4.7.2 Differential Geometry of the Middle Surface

We first introduce the dual of the covariant basis $\vec{a_i}$, termed *contravariant basis* $\vec{a}^i$ and defined as

$$\vec{a}^i \cdot \vec{a}_j = \delta^i_j, \quad i, j = 1, 2, 3,$$

where δ^i_j is the Kronecker symbol, so that an orthogonal relation can be established. Note that $\vec{a}_3$ is a normalised vector perpendicular to the tangent plane at the point $\phi(\xi^1, \xi^2)$, so

$$\vec{a}^3 = \vec{a}_3.$$

As a result, a vector $\boldsymbol{u}$ can be expressed in either covariant bases or contravariant bases as

$$\boldsymbol{u} = u^i \vec{a}_i = u_i \vec{a}^j, \tag{4.126}$$

where u^i and u_i are called *covariant component* and *contravariant component*, or "coordinates" component. There are two fundamental forms corresponding to a shell surface: the *first fundamental form* or the *metric tensor*

$$a_{\alpha\beta} = \vec{a}_\alpha \cdot \vec{a}_\beta \tag{4.127}$$

and the *second fundamental form*

$$b_{\alpha\beta} = \vec{a}_3 \cdot \vec{a}_{\alpha,\beta} = -\vec{a}_\alpha \cdot \vec{a}_{3,\beta}. \tag{4.128}$$

The metric tensor can be used to evaluate the area over domain $\mathcal{S}$ by

$$\int_{\mathcal{S}} \mathrm{d}S = \int_\Omega \sqrt{(a_{\alpha\beta})}\, \mathrm{d}\xi^1\, \mathrm{d}\xi^2, \tag{4.129}$$

and the second fundamental form is geometrically related to how the surface sit in the Ω. One useful fact here is that the second fundamental form vanishes for the plane surface.

For computational reason, it may be convenient to define the *third fundamental form* by

$$c_{\alpha\beta} = b^\lambda_\alpha b_{\lambda\beta}. \tag{4.130}$$

We also have the relations

$$\vec{a}_\alpha = a_{\alpha\beta}\vec{a}^\beta, \quad \vec{a}^\alpha = a^{\alpha\beta}\vec{a}_\beta, \tag{4.131}$$

where

$$a^{\alpha\beta} = \vec{a}^\alpha \cdot \vec{a}^\beta.$$

Since the choice of the base vector for a tensor can be covariant, contravariant and mixed vectors, tensors have different component forms, which can be transformed with the help of the metric tensor (a). For example, we can have different forms for the second fundamental form tensor as

$$b^\beta_\alpha = a^{\beta\lambda}b_{\lambda\alpha}(\text{mixed}), \quad b^{\alpha\beta} = a^{\alpha\lambda}a^{\beta\nu}b_{\lambda\nu}(\text{contravariant}). \tag{4.132}$$

Next, we introduce Christoffel symbols[1], which characterise the varying behaviour of the basis vectors as the point changes and are defined as

$$\Gamma^\lambda_{\beta\alpha} = \vec{a}_{\beta,\alpha} \cdot \vec{a}^\lambda = \vec{a}_{\alpha,\beta} \cdot \vec{a}^\lambda = \Gamma^\lambda_{\alpha\beta}. \tag{4.133}$$

Also, the Gauss equation

$$\vec{a}_{\alpha,\beta} = \Gamma^\lambda_{\alpha\beta}\vec{a}_\lambda + b_{\alpha\beta}\vec{a}_3, \tag{4.134}$$

and Weingarten equation

$$\vec{a}_{3,\alpha} = \vec{a}^3_{,\alpha} = -b^\lambda_\alpha \vec{a}_\lambda. \tag{4.135}$$

[1]Also known as "Christo-awful" Symbols

Then we consider the covariant differentiation of the surface tensors $\vec{u} = u_\beta \vec{a}^\beta$:

$$\frac{\partial \vec{u}}{\partial \xi^\alpha} = u_{\beta,\alpha}\vec{a}^\beta + u_\beta \vec{a}^\beta_{,\alpha} = u_{\beta,\alpha}\vec{a}^\beta + u_\lambda \vec{a}^\lambda_{,\alpha}. \tag{4.136}$$

Substituting with (4.134) results in

$$\frac{\partial \vec{u}}{\partial \xi^\alpha} = (u_{\beta,\alpha} - \Gamma^\lambda_{\alpha\beta} u_\lambda)\vec{a}^\beta + b^\lambda_\alpha u_\lambda \vec{a}_3 \tag{4.137}$$

and define

$$u_{\beta|\alpha} = u_{\beta,\alpha} - \Gamma^\lambda_{\beta\alpha} u_\lambda, \tag{4.138}$$

so that we have

$$\frac{\partial}{\partial \xi^\alpha}\left(u_\beta \vec{a}^\beta\right) = u_{\beta|\alpha}\vec{a}^\beta + b^\lambda_\alpha u_\lambda \vec{a}_3. \tag{4.139}$$

4.7.3 Differential Geometry for Shell in 3D

As the shell is defined in three-dimension, we turn to the covariant base vectors defined in (4.125). Followed results in last section, we may derive

$$\vec{g}_\alpha = \frac{\partial \vec{\Phi}}{\partial \xi^\alpha} = \vec{a}_\alpha + \xi^3 \vec{a}_{3,\alpha} = \vec{a}_\alpha - \xi^3 b^\lambda_\alpha \vec{a}_\lambda, \tag{4.140}$$

and

$$\vec{g}_3 = \frac{\partial \vec{\Phi}}{\partial \xi^3} = \vec{a}_3. \tag{4.141}$$

Similar to the surface tensor, the metric tensor with covariant-covariant components can be defined as

$$g_{ij} = \vec{g}_i \cdot \vec{g}_j. \tag{4.142}$$

Based on (4.140) and (4.141), we can have,

$$\vec{g}^3 = \vec{g}_3,\ g_{33} = 1,\ g_{\alpha 3} = \vec{g}_\alpha \cdot \vec{g}_3 = 0, \tag{4.143}$$

and

$$g_{\alpha\beta} = a_{\alpha\beta} - 2\xi^3 b_{\alpha\beta} + (\xi^3)^2 c_{\alpha\beta}. \tag{4.144}$$

4.7.4 Shell Kinematics

Consider the gray material stick outlined in Figure 4.3 in the direction of $\vec{a}_3$

$$\vec{U}(\xi^1, \xi^2, \xi^3) = \vec{u}(\xi^1, \xi^2) + \xi^3 \theta_\lambda(\xi^1, \xi^2)\vec{a}^\lambda(\xi^1, \xi^2), \tag{4.145}$$

where $\vec{u}$ is the surface displacement and θ_λ is the rotational field, also we have

$$\vec{u}(\xi^1, \xi^2) = u_i(\xi^1, \xi^2)\vec{a}^i.$$

The *Reissner-Mindlin (RM) kinematical assumption* is used in (4.145) by taking into account of the rotation field θ_λ around that point, whilst in other popular kinematical assumptions such as Kichhoff-Love's, this line is assumed to stay normal to the mid-surface. The RM assumption says the normal line remain straight but not necessarily normal to the mid-surface and the first-order transverse shearing is considered, as illustrated in Figure 4.4. Note that the rotation around direction of the covariant basis $\vec{a}_\lambda$ actually contributes the rotational change for the material stick in the corresponding contravariant basis direction $\vec{a}^\lambda$. In Kirchhoff-Love type of kinematical assumption, the rotation can be inferred from the displacement field. The current shell model is studied in the name of "basic shell mathematical model" in [98, 96]. Using the differential geometry established

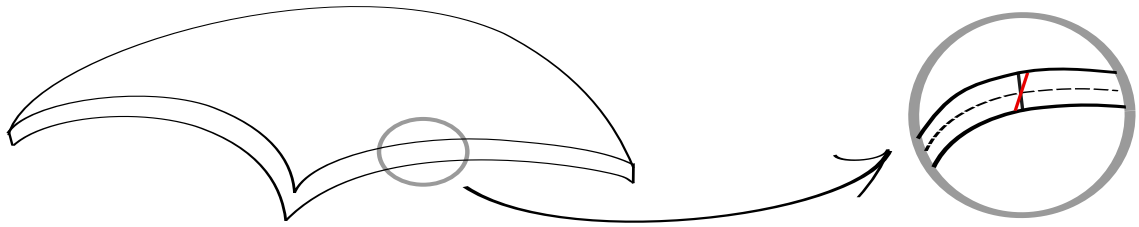


Figure 4.4 The states of the normal line before (black line) and after (red line) deformation

earlier, we proceed to derive the strain displacement relationship. Since that the basis of this curvilinear coordinates (ξ^1, ξ^2, ξ^3) has been defined as (4.125), the strain tensor thus can be expressed as

$$\boldsymbol{\varepsilon} = e_{ij}\vec{g}^i \otimes \vec{g}^j, \tag{4.146}$$

where a different notation for strain component e is used to distinguish from that in Cartesian coordinate. By some tensor manipulations, the strain tensor in a general curvilinear coordinates can be given through differentiating displacement vector $\vec{U}$ as

$$e_{ij} = \frac{1}{2}\left(\frac{\partial \vec{U}}{\partial \xi^i} \cdot \vec{g}_j + \frac{\partial \vec{U}}{\partial \xi^j} \cdot \vec{g}_i\right). \tag{4.147}$$

Using the results obtained from (4.139), we can write

$$\begin{aligned} \frac{\partial \vec{u}}{\partial \xi^\alpha} &= \frac{\partial}{\partial \xi^\alpha}\left(u_\lambda \vec{a}^\lambda + u_3 \vec{a}_3\right) \\ &= u_{\lambda|\alpha}\vec{a}^\lambda + b^\lambda_\alpha u_\lambda \vec{a}_3 + u_{3,\alpha}\vec{a}_3 + u_3 \vec{a}_{3,\alpha} \\ &= (u_{\lambda|\alpha} - b_{\lambda\alpha}u_3)\vec{a}^\lambda + (u_{3,\alpha} + b^\lambda_\alpha u_\lambda)\vec{a}_3, \end{aligned} \tag{4.148}$$

where a variant of (4.128) is used. Similarly to (4.139), we have

$$\frac{\partial}{\partial \xi^\alpha}\left(\theta_\lambda \vec{a}^\lambda\right) = \theta_{\lambda|\alpha}\vec{a}^\lambda + b^\lambda_\alpha \theta_\lambda \vec{a}_3, \tag{4.149}$$

so

$$\frac{\partial \vec{U}}{\partial \xi^\alpha} = \left(u_{\lambda|\alpha} - b_{\lambda\alpha}u_3 + \xi^3\theta_{\lambda|\alpha}\right)\vec{a}^\lambda + \left(u_{3,\alpha} + b^\lambda_\alpha u_\lambda + \xi^3 b^\lambda_\alpha \theta_\lambda\right)\vec{a}_3, \tag{4.150}$$

and

$$\frac{\partial \vec{U}}{\partial \xi^3} = \theta_\lambda \vec{a}^\lambda. \tag{4.151}$$

With all the necessary expressions ready for evaluating (4.147), we can obtain the covariant strain tensor component

$$\begin{cases} e_{\alpha\beta} = \gamma_{\alpha\beta}(\vec{u}) + \xi^3 \chi_{\alpha\beta}(\vec{u}, \vec{\theta}) - (\xi^3)^2 \kappa_{\alpha\beta}(\vec{\theta}), \\ e_{\alpha 3} = \zeta_\alpha(\vec{u}, \vec{\theta}) \\ e_{33} = 0 \end{cases} \tag{4.152}$$

where

$$\begin{cases} \gamma_{\alpha\beta}(\vec{u}) = \dfrac{1}{2}\left(u_{\alpha|\beta} + u_{\beta|\alpha}\right) - b_{\alpha\beta} u_3 \\ \chi_{\alpha\beta}(\vec{u}, \vec{\theta}) = \dfrac{1}{2}\left(\theta_{\alpha|\beta} + \theta_{\beta|\alpha} - b^\lambda_\beta u_{\lambda|\alpha} - b^\lambda_\alpha u_{\lambda|\beta}\right) - b^\lambda_\alpha b_{\lambda\beta} u_3 \\ \kappa_{\alpha\beta}(\vec{\theta}) = \dfrac{1}{2}\left(b^\lambda_\beta \theta_{\lambda|\alpha} + b^\lambda_\alpha \theta_{\lambda|\beta}\right) \\ \zeta_\alpha(\vec{u}, \vec{\theta}) = \dfrac{1}{2}\left(\theta_\alpha + u_{3,\alpha} + b^\lambda_\alpha u_\lambda\right) \end{cases} \tag{4.153}$$

4.7.5 Strain-Displacement Relations

We move on to derive an equivalent expression of the general strain-displacement relations for shell theory. In the curvilinear coordinates, we first approximate the covariant components by

$$u_i(\xi^1, \xi^2) = \Xi_{im}(\xi^1, \xi^2) u^h_m, \quad \text{and} \quad \theta_\alpha(\xi^1, \xi^2) = \Theta_{\alpha n}(\xi^1, \xi^2) \theta^h_n, \tag{4.154}$$

where $i = 1, 2, 3$, $m = 1, 2, \cdots, l_u$, $n = 1, 2, \cdots, l_\theta$ ($l_{u+\theta} = l_u + l_\theta$), u^h_m and θ^h_n are the unknown variables at the nodes in the finite elements. Next, we express the displacement defined in (4.145) as

$$\vec{U} = U_i \vec{g}^i,$$

whose covariant components can be calculated by multiplying the covariant bases as

$$U_i = \vec{U} \cdot \vec{g}_i = \left(u_j \vec{a}^j + \xi^3 \theta_\lambda \vec{a}^\lambda\right) \cdot \vec{g}_i. \tag{4.155}$$

Substituting with the explicit form of $\vec{g}_i$, we have the following

$$\begin{aligned} U_3 &= \left(u_j \vec{a}^j + \xi^3 \theta_\lambda \vec{a}^\lambda\right) \cdot \vec{g_3} = u_3; \\ U_\alpha &= \left(u_j \vec{a}^j + \xi^3 \theta_\lambda \vec{a}^\lambda\right) \cdot \left(\vec{a}_\alpha - \xi^3 b^\beta_\alpha \vec{a}_\beta\right) \\ &= u_j \delta^j_\alpha - \xi^3 u_j b^\beta_\alpha \delta^j_\beta + \xi^3 \theta_\lambda \delta^\lambda_\alpha - \left(\xi^3\right)^2 b^\beta_\alpha \theta_\lambda \delta^\lambda_\beta \\ &= u_\alpha - \xi^3 b^\beta_\alpha u_\beta + \xi^3 \theta_\alpha - \left(\xi^3\right)^2 b^\beta_\alpha \theta_\beta. \end{aligned} \tag{4.156}$$

Next we can proceed by deriving $\boldsymbol{N}$ defined in (4.106) as

$$\boldsymbol{N} = \boldsymbol{N}_{ik}\vec{g}^i \otimes \tilde{\boldsymbol{e}}_k. \tag{4.157}$$

Substituting in (4.156) with approximation (4.154), we can establish a relation between U_i, u_m^h and θ_n^h as

$$\begin{aligned} U_3 &= \Xi_{3m} u_m^h, \\ U_\alpha &= \left(\Xi_{\alpha m} - \xi^3 b_\alpha^\beta \Xi_{\beta m}\right) u_m^h + \left(\xi^3 \Theta_{\alpha n} - (\xi^3)^2 b_\alpha^\beta \Theta_{\beta n}\right) \theta_n^h. \end{aligned} \tag{4.158}$$

We use $\mathcal{U}$ to denote $\boldsymbol{u}_h$, which has the form

$$\mathcal{U} = \mathcal{U}_k^h \tilde{\boldsymbol{e}}^k (k = 1, 2, \cdots, l_u + l_\theta),$$

where

$$\mathcal{U}_k^h = \begin{cases} u_k^h & \text{if } k \le l_u, \\ \theta_{k-l_u}^h & \text{if } l_u < k \le l_{u+\theta}. \end{cases}, \tag{4.159}$$

so that we have

$$\vec{U} = \boldsymbol{N}\mathcal{U} = \boldsymbol{N}_{ik}\mathcal{U}_k^h \vec{g}^i, \tag{4.160}$$

which leads to

$$U_i = \boldsymbol{N}_{ik}\mathcal{U}_k^h,$$

and based on (4.158), we obtain

$$\begin{aligned} \boldsymbol{N}_{\alpha k} &= \begin{cases} \Xi_{\alpha k} - \xi^3 b_\alpha^\beta \Xi_{\beta k} & k \le l_u \\ \xi^3 \Theta_{\alpha\, k-l_u} - (\xi^3)^2 b_\alpha^\beta \Theta_{\beta\, k-l_u} & l_u < k \le l_u + l_\theta \end{cases} \\ \boldsymbol{N}_{3k} &= \begin{cases} \Xi_{3k} & k \le l_u \\ 0 & l_u < k \le l_u + l_\theta \end{cases} \end{aligned} \tag{4.161}$$

The tensor $\boldsymbol{B}$ in (4.108) is replaced with a slightly different notation $\mathcal{B}$ in order to distinguish it from the notation that will be used later (B-spline functions), so that we have

$$\mathcal{B} = \mathcal{B}_{ijk}\vec{g}^i \otimes \vec{g}^j \otimes \tilde{e}^k.$$

$$\boldsymbol{\varepsilon} = \mathcal{B}\mathcal{U}, \tag{4.162}$$

whose component form of equation (4.162) can be expressed as

$$\varepsilon_{ij} = \mathcal{B}_{ijk}\mathcal{U}_k^h = \mathcal{B}_{ijm} u_m^h + \mathcal{B}_{ijn+l_u}\theta_n^h. \tag{4.163}$$

In the following, we can have some of the evaluations:

$$u_3 = \Xi_{3m} u_m^h \tag{4.164a}$$

$$u_{\alpha|\beta} = u_{\alpha,\beta} - \Gamma_{\alpha\beta}^{\lambda} u_\lambda = \left(\Xi_{\alpha m,\beta} - \Gamma_{\alpha\beta}^{\lambda}\Xi_{\lambda m}\right) u_m^h \tag{4.164b}$$

$$u_{\beta|\alpha} = \left(\Xi_{\beta m,\alpha} - \Gamma_{\beta\alpha}^{\lambda}\Xi_{\lambda m}\right) u_m^h \tag{4.164c}$$

$$\theta_{\alpha|\beta} = \left(\Theta_{\alpha n,\beta} - \Gamma_{\alpha\beta}^{\lambda}\Theta_{\lambda n}\right) \theta_n^h \tag{4.164d}$$

$$\theta_{\beta|\alpha} = \left(\Theta_{\beta n,\alpha} - \Gamma_{\beta\alpha}^{\lambda}\Theta_{\lambda n}\right) \theta_n^h \tag{4.164e}$$

and defining

$$\Lambda_{\alpha\beta m}^{\gamma} = \frac{1}{2}\left(\Xi_{\alpha m,\beta} + \Xi_{\beta m,\alpha} - \left(\Gamma_{\alpha\beta}^{\lambda} + \Gamma_{\beta\alpha}^{\lambda}\right)\Xi_{\lambda m} - b_{\alpha\beta}\Xi_{3m}\right), \tag{4.165a}$$

$$\Lambda_{\alpha\beta n}^{\chi_\theta} = \frac{1}{2}\left(\Theta_{\alpha n,\beta} + \Theta_{\beta n,\alpha} - \left(\Gamma_{\alpha\beta}^{\lambda} + \Gamma_{\beta\alpha}^{\lambda}\right)\Theta_{\lambda n}\right), \tag{4.165b}$$

$$\Lambda_{\alpha\beta n}^{\chi_u} = -\frac{1}{2}\left(b_\beta^\lambda \Xi_{\lambda m,\alpha} + b_\alpha^\lambda \Xi_{\lambda m,\beta} - b_\beta^\lambda \Gamma_{\lambda\alpha}^{\nu}\Xi_{\nu m} - b_\alpha^\lambda \Gamma_{\lambda\beta}^{\nu}\Xi_{\nu m}\right) - b_\alpha^\lambda b_{\lambda\beta}\Xi_{3m}, \tag{4.165c}$$

$$\Lambda_{\alpha\beta n}^{\kappa} = \frac{1}{2}\left(b_\beta^\lambda\left(\Theta_{\lambda n,\alpha} - \Gamma_{\lambda\alpha}^{\nu}\Theta_{\nu n}\right) + b_\alpha^\lambda\left(\Theta_{\lambda n,\beta} - \Gamma_{\lambda\beta}^{\nu}\Theta_{\nu n}\right)\right), \tag{4.165d}$$

$$\Lambda_{\alpha n}^{\zeta_\theta} = \frac{1}{2}\Theta_{\alpha n}, \tag{4.165e}$$

$$\Lambda_{\alpha m}^{\zeta_u} = \frac{1}{2}\left(\Xi_{3m,\alpha} + b_\alpha^\lambda \Xi_{\lambda m}\right), \tag{4.165f}$$

(note that $\Gamma_{\alpha\beta}^{\lambda} = \Gamma_{\beta\alpha}^{\lambda}$) so that we can have

$$\gamma_{\alpha\beta} = \Lambda_{\alpha\beta m}^{\gamma} u_m^h, \tag{4.166a}$$

$$\chi_{\alpha\beta} = \Lambda_{\alpha\beta n}^{\chi_\theta}\theta_n^h + \Lambda_{\alpha\beta n}^{\chi_u} u_m^h, \tag{4.166b}$$

$$\kappa_{\alpha\beta} = \Lambda_{\alpha\beta n}^{\kappa}\theta_n^h, \tag{4.166c}$$

$$\zeta_\alpha = \Lambda_{\alpha n}^{\zeta_\theta}\theta_n^h + \Lambda_{\alpha m}^{\zeta_u} u_m^h, \tag{4.166d}$$

and correspondingly

$$\varepsilon_{\alpha\beta} = \left(\Lambda_{\alpha\beta m}^{\gamma} + \xi^3 \Lambda_{\alpha\beta m}^{\chi_u}\right) u_m^h + \left(\xi^3 \Lambda_{\alpha\beta n}^{\chi_\theta} - (\xi^3)^2 \Lambda_{\alpha\beta n}^{\kappa}\right)\theta_n^h, \tag{4.167a}$$

$$\varepsilon_{\alpha 3} = \Lambda_{\alpha m}^{\zeta_u} u_m^h + \Lambda_{\alpha n}^{\zeta_\theta}\theta_n^h. \tag{4.167b}$$

Comparing with equation (4.163), we obtain

$$\mathcal{B}_{\alpha\beta k} = \begin{cases} \Lambda_{\alpha\beta k}^{\gamma} + \xi^3 \Lambda_{\alpha\beta k}^{\chi_u} & \text{if } k \le l_u \\ \xi^3 \Lambda_{\alpha\beta k - l_u}^{\chi_\theta} - (\xi^3)^2 \Lambda_{\alpha\beta k - l_u}^{\kappa} & \text{if } l_u < k \le l_{u+\theta} \end{cases}, \tag{4.168}$$

and

$$\mathcal{B}_{\alpha 3 k} = \begin{cases} \Lambda_{\alpha k}^{\zeta_u} & \text{if } k \le l_u \\ \Lambda_{\alpha k - l_u}^{\zeta_\theta} & \text{if } l_u < k \le l_{u+\theta} \end{cases}, \mathcal{B}_{33k} = 0, \quad \mathcal{B}_{ijk} = \mathcal{B}_{jik}. \tag{4.169}$$

The quantities introduced in (4.165) are all dependent on ξ^1, ξ^2 only. The modularised expressions of (4.168) and (4.169) allow a simple matrix implementation in actual computations.

4.7.6 B-Spline Curves

Quite often the shell geometry cannot be expressed with exact analytic mapping function $\vec{\phi}$, but instead described by scattered data at finite points. One of the methods to get around this issue is to to use the *B-spline functions*. Spline originates from the ship building industry where engineers use a thin wood spline to pass a number of points and can obtain a curvature continuous curve. B(asic)-splines are piecewise polynomials and their appealing feature is that they are differentiable up to a prescribed order.

To generate the B-spline, let $\boldsymbol{t} := (t_j)$ be a non-decreasing sequence, which shall be referred to as a *knot sequence*, then a modern recursive definition of the basic spline functions of order k (degree $k-1$), $B_{j,k,t} \quad j = 1, 2, \cdots, n$ by Carl de Boor [99] states

$$B_{j,1,t}(\xi) = \begin{cases} 1 \text{ if } \xi \in [t_j, t_{j+1}), \\ 0 \text{ otherwise.} \end{cases}$$

For $k \geq 2$:

$$B_{j,k,t}(\xi) = \begin{cases} \dfrac{\xi - t_j}{t_{j+k-1} - t_j} B_{j,k-1,t}(\xi) + \dfrac{t_{j+k} - \xi}{t_{j+k} - t_{j+1}} B_{j+1,k-1,t}(\xi), & j = 1, 2, \cdots, n-1 \\ \dfrac{\xi - t_j}{t_{j+k-1} - t_j} B_{j,k-1,t}(\xi), & j = n. \end{cases} \tag{4.170}$$

In case of knot points multiplicity, i.e., $t_j = t_{j+k-1}$, we set the corresponding term

$$\frac{\xi - t_j}{t_{j+k-1} - t_j} B_{j,k-1,t}(\xi) = 0.$$

To ensure B-spline functions cover the whole interval $[0, 1]$, we need to modify $B_{n,k,t}(1) = 1$, rather than 0. From the definition of a B-spline function, we know that a degree k B-spline function is constructed from $k-1$ order piecewise polynomials, and thus it is differentiable up to order $k-2$.

The selection of the knot sequences must satisfy the *Schoenberg-Whitney conditions*:

$$\tau_i < t_i < \tau_{i+k}, \forall i \tag{4.171}$$

where τ_i are the interpolation points, and usually an optimal knot sequence by Michelli, Rivlin and Winograd is applied [100]. The sequence has to be determined by setting the first k and the last k values as

$$t_1 = \cdots = t_k = 0, \qquad t_{n+1} = \cdots = t_{n+k} = 1, \tag{4.172}$$

and the rest $(n-k)$ points are chosen as the discontinuity points of the step function,

denoted as $h(x)$, which satisfies

$$\begin{cases} |h(x)| = 0, x \in [0,1], h(0^+) = 1; \\ \int_0^1 f(x)h(x)\,\mathrm{d}x = 0, \forall; \\ h(x) \text{ has at most } n-k \text{ changes of sign within interval } [0,1]. \end{cases} \tag{4.173}$$

In practice, these requirements are achieved numerically [99]. Some notable properties of B-Spline are summarised here:

- Positivity: $B_{j,k,t} \geq 0, \forall \xi \in \mathbb{R}$,
- Small support: $B_{j,k,t} > 0$ only on $[t_j, t_{j+k}]$, zero otherwise.
- Partition of unity: $\sum_j B_{j,k,t} = 1$,

Given known k and knot sequence $\boldsymbol{t}$, we may use short notation B_j for $B_{j,k,t}$. In Figure 4.5, we illustrate an example of B-splines with interpolation points and knot sequence:

$$\boldsymbol{\tau} = \Big[0, \quad 0.0667, \quad 0.1333, \quad 0.2667, \quad 0.3333, \quad 0.4667, \quad 0.6000, \quad 0.8000, \quad 1.0000\Big],$$
$$\boldsymbol{t} = [0, 0, 0, 0, 0.1539, 0.2539, 0.3625, 0.4899, 0.6501, 1.0000, 1.0000, 1.0000, 1.0000].$$

For a 1D interpolation problem, we have

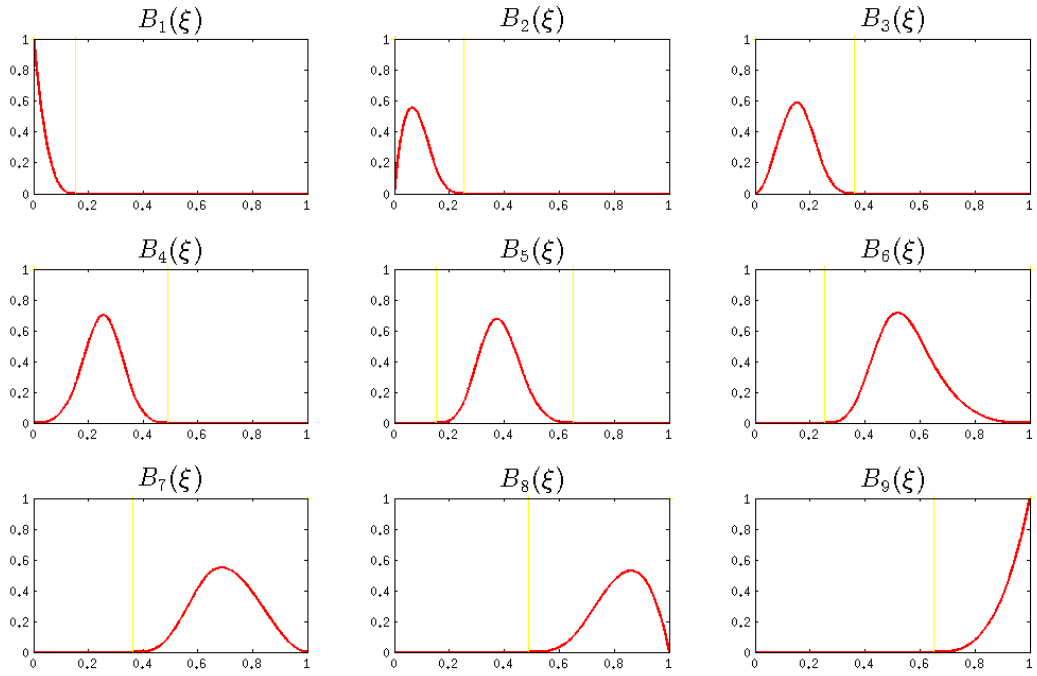


Figure 4.5 Basic spline functions (red line) B_j, $j = 1, 2, \cdots, 9$, and $k = 4$. The yellow lines highlight small support property.

$$\sum_{j=1}^{n} c_j B_j(\tau_i) = w(\tau_i), \quad i = 1, 2, \cdots, n, \tag{4.174}$$

where $w(\xi)$ is the function to be interpolated (its explicit form unknown) and the c_j are the relative coefficients to be determined. Equation (4.174) constructs a linear system, whose invertibility is guaranteed by equation (4.171), so that we can claim

$$(c)_{n\times 1} = (B_j(\tau_i))^{-1}_{n\times n} (w(\tau_i))_{n\times 1}. \tag{4.175}$$

An example shown in Figure 4.6 illustrate that a higher degree of differentiability incurs oscillation.

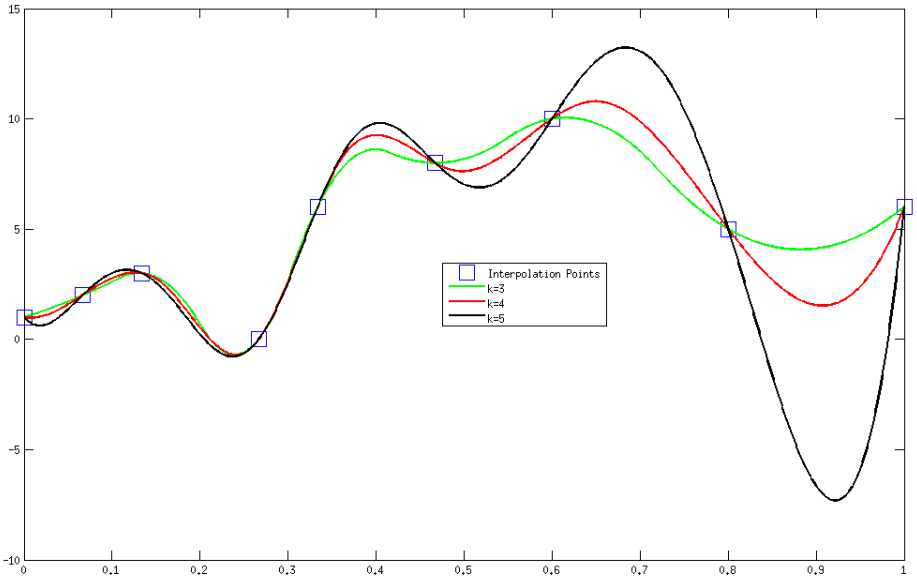


Figure 4.6 Basic spline functions interpolations with different orders. Blue squares are the interpolation points, green line: $k = 3$, red line: $k = 4$ and black line: $k = 5$.

Derivatives of B-Spline Function. The derivative of B-Spline function is given as [101]

$$B'_{j,k}(\xi) = \frac{k-1}{t_{j+k-1} - t_j} B_{j,k-1}(\xi) - \frac{k-1}{t_{j+k} - t_{j+1}} B_{j+1,k-1}(\xi), j = 1, 2, \cdots, n-1 \tag{4.176}$$

and the treatment with knot multiplicity and $j = n$ stay the same with that of B-spline definition in (4.170). Equation (4.176) basic shows that the derivative of a basis spline function is also a basis function with same knot sequence, but is one order lower. So the derivative of the B-Spline interpolated function

$$f(\xi) = \sum_{j=1}^{n} c_j B_{j,k,t}(\xi)$$

can be obtained from:

$$f' = \frac{\mathrm{d}f}{\mathrm{d}\xi} = \sum_{j=1}^{n} c_j B'_{j,k,t}(\xi) = (k-1)\sum_{j=1}^{n} \frac{c_j - c_{j-1}}{t_{j+k-1} - t_j} B_{j,k-1}, \tag{4.177}$$

where specifically $c_0 = 0$. Actually, a general derivative of order m can be written as [99]

$$D^m f = \sum_{j=1}^{n} c_j^{(m+1)} B_{j,k-m}(\xi), \tag{4.178}$$

where $c_j^{(m+1)}$ is defined recursively as

$$c_j^{(m+1)} = \begin{cases} c_j & m = 0; \\ \dfrac{c_j^{(m)} - c_{j-1}^{(m)}}{(t_{j+k-m} - t_j)/(k-m)} & m > 0. \end{cases} \tag{4.179}$$

So the explicit second order derivative can be evaluated as

$$f'' = \sum_{j=1}^{n} \frac{\dfrac{c_j - c_{j-1}}{(t_{j+k-1} - t_j)/(k-1)} - \dfrac{c_{j-1} - c_{j-2}}{(t_{j+k-2} - t_{j-1})/(k-1)}}{(t_{j+k-2} - t_j)/(k-2)} B_{j,k-2}(\xi). \tag{4.180}$$

4.7.7 B-Spline Surface Approximation

Having established the B-spline function for a single variable, we can have the bilinear B-spline interpolation for a function $w(\xi^1, \xi^2)$ as

$$\pi w(\xi^1, \xi^2) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} B^1_{i,k^1,t^1}(\xi^1) B^2_{j,k^2,t^2}(\xi^2), \tag{4.181}$$

where similar to (4.175) c_{ij} can be worked out as

$$(c)_{n^1 \times n^2} = \left(B^1_{i,k^1,t^1}(\tau_p^1)\right)^{-1}_{n^1 \times n^1} \left(w(\tau_p^1, \tau_q^2)\right)_{n^1 \times n^2} \left(B^2_{j,k^2,t^2}(\tau_q^2)\right)^{-1}_{n^2 \times n^2}. \tag{4.182}$$

The components $x^i(\xi^1, \xi^2), i = 1, 2, 3$ of the mapping

$$\vec{\phi}(\xi^1, \xi^2) = x^i(\xi^1, \xi^2)\vec{e}_i$$

thus can be approximated by (summation rule not applied here)

$$x^i(\xi^1, \xi^2) = \sum_{p=1}^{n_1}\sum_{q=1}^{n_2} c^i_{pq} B^i_{p,k^1,t^1}(\xi^1) B^i_{q,k^2,t^2}(\xi^2), \tag{4.183}$$

where k^1, k^2, t^1, t^2 indicate different choices of the B-spline order and different knot sequences in two dimension. In actual matrix computation, we stack the discretised basic

spline functions $B_i(\xi)$, column vectors, into a matrix, so that we have

$$[w(\xi^1,\xi^2)] = [B^1(\xi^1)][c_{ij}][B^2(\xi^2)]^T.$$

Next we turn to the evaluation of the surface tensors and let us consider the differentiation to (4.183) first. Similar to (4.177), we have

$$\frac{\partial x^i(\xi^1,\xi^2)}{\partial \xi^1} = (k^1-1)\sum_{p=1}^{n_1}\sum_{q=1}^{n_2}\frac{c^i_{pq}-c^i_{p-1\,q}}{t^1_{p+k^1-1}-t^1_p}B^i_{p,k^1-1,t^1}(\xi^1)B^i_{q,k^2,t^2}(\xi^2), \tag{4.184a}$$

$$\frac{\partial x^i(\xi^1,\xi^2)}{\partial \xi^2} = (k^2-1)\sum_{p=1}^{n_1}\sum_{q=1}^{n_2}\frac{c^i_{pq}-c^i_{p\,q-1}}{t^2_{q+k^2-1}-t^2_q}B^i_{p,k^1,t^1}(\xi^1)B^i_{q,k^2-1,t^2}(\xi^2), \tag{4.184b}$$

Hence, the first fundamental form

$$\vec{a}_\alpha(\xi^1,\xi^2) = \frac{\partial x^i}{\partial \xi^\alpha}\vec{e}_i = x^i_{,\alpha}\vec{e}_i, \alpha = 1,2 \tag{4.185}$$

can be obtained. Similarly, the second order derivative information can be deduced from equation (4.180).

"Everything is vague to a degree you do not realise till you have tried to make it precise."

Bertrand Russell

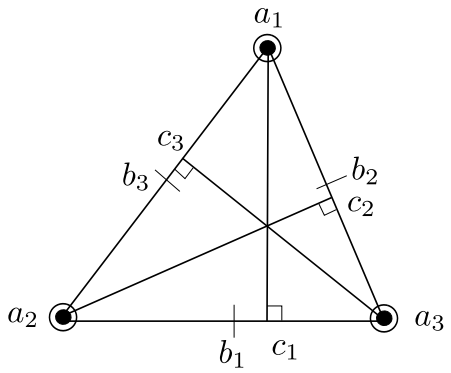
Chapter 5

Numerical Implementation

This chapter details several aspects of the numerical schemes required for the computational model set out in the previous chapters. At heart of this chapter is the numerical integrations and matrix based computational methods.

5.1 Displacement Discretisation and Implementations

This section describes the implementation of the strain-displacement relations. We choose Ganev triangles to approximate $u_i(\xi^1, \xi^2)$ and $\theta_\alpha(\xi^1, \xi^2)$, as shown in Figure 5.1.



$$P_K = P_4(K); \dim P_K = 15;$$

$$\Sigma_K = \{p_{ol}(a_i), Dp_{ol}(a_i)(a_{i-1} - a_i), Dp_{ol}(a_i)(a_{i+1} - a_i), p_{ol}(b_i), Dp_{ol}(b_i)(a_i - c_i), 1 \leq i \leq 3\}.$$

Figure 5.1 Ganev triangle illustration

Following equation (4.13), we write the p-interpolant for Ganev triangle in the matrix form

$$\Pi_G v = \overline{\boldsymbol{d}}_G^T p_G, \tag{5.1}$$

where $\overline{\boldsymbol{d}}_G$ denotes the set of *local* degrees of freedom for Ganev triangle, which is Σ_K in

Figure 5.1, and p_G is the *basis function* of the Ganev triangle. If expressed in barycentric coordinate (as indicated by vector $\boldsymbol{\zeta}_G$), p_G has the form

$$p_G = A_G \boldsymbol{\zeta}_G, \tag{5.2}$$

where exact forms of A_G and ζ_G are given in Appendix (B.11). The extra parameters η_i in (B.11) is called the *eccentricity parameter*, first introduced by Argyris, Fried and Scharpf [102], to handle normal derivative transformation. These parameters are defined as

$$\eta_i = 2\frac{\overline{c_i b_i}}{\overline{a_{i-1}a_{i+1}}}, i = 1, 2, 3. \tag{5.3}$$

In order to assemble the finite elements, a *global* degrees of freedom, denoted as $\boldsymbol{d}_G$, is needed. First, we state that the coordinates ξ^1, ξ^2 is replaced with x and y in the usual two dimensional Euclidean space, so the Barycentric coordinate defined in equation (B.1) for any arbitrary point (x, y) on a triangle with vertices (x_1, y_1), (x_2, y_2) and (x_3, y_3) becomes

$$\begin{cases} \zeta_1 x_1 + \zeta_2 x_2 + \zeta_3 x_3 = x \\ \zeta_1 y_1 + \zeta_2 y_2 + \zeta_3 y_3 = y \\ \zeta_1 + \zeta_2 + \zeta_3 = 1 \end{cases} . \tag{5.4}$$

Define scalars

$$x_{ij} = x_i - x_j \text{ and } y_{ij} = y_i - y_j \quad i, j = 1, 2, 3,$$

as the difference between vertices a_i and a_j. The local degrees of freedom can be then transformed into the global coordinates by the following relations

$$\begin{aligned} Dp_{ol}(a_i)(a_{i-1} - a_i) &= \partial_x p_o(a_i) x_{i-1\,i} + \partial_y p_o(a_i) y_{i-1\,i} \\ Dp_{ol}(a_i)(a_{i+1} - a_i) &= \partial_x p_o(a_i) x_{i+1\,i} + \partial_y p_o(a_i) y_{i+1\,i} \end{aligned} \tag{5.5}$$

where p_o is also the global evaluation corresponding to the local expression p_{ol}. Using the coordinates of triangle vertices, we can also have eccentricity parameters as (cf. 5.3) by

$$\eta_i = \frac{l_{i+2}^2 - l_{i+1}^2}{l_i^2}, \quad l_i = \sqrt{x_{i+1\,i-1}^2 + y_{i+1\,i-1}^2} \tag{5.6}$$

We also have [97]

$$|\overrightarrow{c_i a_i}| = \sqrt{{x_{ca}}_i^2 + {y_{ca}}_i^2},$$

where

$$\begin{aligned} x_{cai} &= x_i - \frac{1}{2}(1 - \eta_i)x_{i+1} - \frac{1}{2}(1 + \eta_i)x_{i-1} \\ y_{cai} &= y_i - \frac{1}{2}(1 - \eta_i)y_{i+1} - \frac{1}{2}(1 + \eta_i)y_{i-1} \end{aligned} , i = 1, 2, 3.$$

For the normal derivatives, we have

$$Dp_{ol}(b_i)(a_i - c_i) = \partial_\nu p_o(b_i) n_i, \tag{5.7}$$

where n_i is determined from

$$n_i = \begin{cases} |\overrightarrow{c_i a_i}| \dfrac{x_{cai}}{|x_{cai}|} & \text{if } x_{cai} \neq 0 \\ |\overrightarrow{c_i a_i}| \dfrac{y_{cai}}{|y_{cai}|} & \text{if } x_{cai} = 0 \end{cases} \qquad i = 1, 2, 3. \tag{5.8}$$

Formulating the local and global degree of freedom explicitly as follows

$$\begin{aligned} \overline{\boldsymbol{d}}_G = [&v(a_1), v(a_2), v(a_3), Dv(a_1)(a_3 - a_1), Dv(a_1)(a_2 - a_1), Dv(a_2)(a_1 - a_2), \\ &Dv(a_2)(a_3 - a_2), Dv(a_3)(a_2 - a_3), Dv(a_3)(a_1 - a_3), v(b_1), v(b_2), v(b_3), \\ &Dv(b_1)(a_1 - c_1), Dv(b_2)(a_2 - c_2), Dv(b_3)(a_3 - c_3)]^T; \end{aligned} \tag{5.9}$$

$$\begin{aligned} \boldsymbol{d}_G = [&v(a_1), v(a_2), v(a_3), \partial_x v(a_1), \partial_y v(a_1), \partial_x v(a_2), \partial_y v(a_2), \\ &\partial_x v(a_3), \partial_y v(a_3), v(b_1), v(b_2), v(b_3), \partial_\nu v(b_1), \partial_\nu v(b_2), \partial_\nu v(b_3)]^T, \end{aligned} \tag{5.10}$$

together with (5.5) and (5.7), the relations between the local and the global degrees of freedom can be finally established as

$$\overline{\boldsymbol{d}}_G^T = \boldsymbol{d}_G^T D_G, \tag{5.11}$$

where D_G is a block diagonal matrix defined by

$$D_G = \mathbf{diag}(1, 1, 1, d_1, d_2, d_3, 1, 1, 1, n_1, n_2, n_3), \tag{5.12}$$

where

$$d_i = \begin{bmatrix} x_{i-1i} & x_{i+1i} \\ y_{i-1i} & y_{i+1i} \end{bmatrix}, \quad i = 1, 2, 3.$$

To obtain the derivatives information, we may transform equation (5.4) as

$$\begin{cases} \zeta_1(x_1 - x_3) + \zeta_2(x_2 - x_3) = x - x_3 \\ \zeta_1(y_1 - y_3) + \zeta_2(y_2 - y_3) = y - y_3 \end{cases} \tag{5.13}$$

and correspondingly

$$\begin{bmatrix} \frac{\partial x}{\partial \zeta_1} & \frac{\partial x}{\partial \zeta_2} \\ \frac{\partial y}{\partial \zeta_1} & \frac{\partial y}{\partial \zeta_2} \end{bmatrix} = \begin{bmatrix} x_{13} & x_{23} \\ y_{13} & y_{23} \end{bmatrix} = \boldsymbol{J}, \quad \begin{bmatrix} \frac{\partial \zeta_1}{\partial x} & \frac{\partial \zeta_1}{\partial y} \\ \frac{\partial \zeta_2}{\partial x} & \frac{\partial \zeta_2}{\partial y} \end{bmatrix} = \boldsymbol{J}^{-1}. \tag{5.14}$$

With displacement in the general form of

$$u_i(\zeta_1,\zeta_2,\zeta_3) = u_i\left(\xi^1(\zeta_1,\zeta_2,\zeta_3),\xi^2(\zeta_1,\zeta_2,\zeta_3)\right)$$

we can write the derivative with respect to the barycentric coordinate as

$$\frac{\partial u_i}{\partial \zeta_j} = \frac{\partial u_i}{\partial \xi^\alpha}\frac{\partial \xi^\alpha}{\partial \zeta_j}, \tag{5.15}$$

and

$$u_{i,\alpha} = \frac{\partial u_i}{\partial \xi^\alpha} = \frac{\partial u_i}{\partial \zeta_k}\frac{\partial \zeta_k}{\partial \xi^\alpha}, \quad (i,k,j=1,2,3,\ \alpha=1,2), \tag{5.16}$$

where the first-order derivative coordinates transformation $\dfrac{\partial \zeta_k}{\partial \xi^\alpha}$ and $\dfrac{\partial \xi^\alpha}{\partial \zeta_k}$ is list in (5.14).

Using Ganev triangle for both approximations of u_i and θ_α, we have

$$u_i(\xi^1,\xi^2) = u_i(\zeta_1,\zeta_2,\zeta_3) = [\boldsymbol{u_h}_i^T]_{1\times 15}[D_G]_{15\times 15}[A_G]_{15\times 15}[\boldsymbol{\zeta}_G]_{15\times 1}, \tag{5.17a}$$

$$\theta_\alpha(\xi^1,\xi^2) = \theta_\alpha(\zeta_1,\zeta_2,\zeta_3) = [\boldsymbol{\theta_h}_\alpha^T]_{1\times 15}[D_G]_{15\times 15}[A_G]_{15\times 15}[\boldsymbol{\zeta}_G]_{15\times 1}, \tag{5.17b}$$

where $\boldsymbol{u_h}_i, i=1,2,3$ and $\boldsymbol{\theta_h}_\alpha, \alpha=1,2$, denote the global degrees of freedom, which take the form of $\boldsymbol{d}_G$, i.e. replacing v with $\boldsymbol{u_h}_i$ or $\boldsymbol{\theta_h}_\alpha$ in formula (5.10), and also we have

$$\left[u_m^h\right] = \begin{bmatrix}\boldsymbol{u_h}_1\\ \boldsymbol{u_h}_2\\ \boldsymbol{u_h}_3\end{bmatrix}, \quad \left[\theta_n^h\right] = \begin{bmatrix}\boldsymbol{\theta_h}_1\\ \boldsymbol{\theta_h}_2\end{bmatrix}, \tag{5.18}$$

So the basis matrices Ξ and Θ (also known as *shape function*) in equation (4.154) substantialise as

$$\Xi = \begin{bmatrix}\boldsymbol{\varrho} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\varrho} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{\varrho}\end{bmatrix}, \quad \Theta = \begin{bmatrix}\boldsymbol{\varrho} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\varrho}\end{bmatrix} \tag{5.19}$$

where

$$\boldsymbol{\varrho} = (D_G A_G \boldsymbol{\zeta}_G)^T. \tag{5.20}$$

Note that $\boldsymbol{\varrho}$ are row vectors rather than the common column vectors. So that the complete basis matrix $[\boldsymbol{N}_{ik}]$ referred in (4.161) can be constructed as

$$\begin{aligned}[\boldsymbol{N}_{ik}] &= \begin{bmatrix}\Xi_{1\cdot} - \xi^3\left(b_1^1\Xi_{1\cdot} + b_1^2\Xi_{2\cdot}\right) & \xi^3\Theta_{1\cdot} - (\xi^3)^2\left(b_1^1\Theta_{1\cdot} + b_1^2\Theta_{2\cdot}\right)\\ \Xi_{2\cdot} - \xi^3\left(b_2^1\Xi_{1\cdot} + b_2^2\Xi_{2\cdot}\right) & \xi^3\Theta_{2\cdot} - (\xi^3)^2\left(b_2^1\Theta_{1\cdot} + b_2^2\Theta_{2\cdot}\right)\\ \Xi_{3\cdot} & \mathbf{0}\end{bmatrix},\\ &= \left(\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} - \xi^3\begin{bmatrix}b_1^1 & b_1^2 & 0\\ b_2^1 & b_2^2 & 0\\ 0&0&0\end{bmatrix}\right)\begin{bmatrix}\Xi_{1\cdot} & \xi^3\Theta_{1\cdot}\\ \Xi_{2\cdot} & \xi^3\Theta_{2\cdot}\\ \Xi_{3\cdot} & \mathbf{0}\end{bmatrix}\end{aligned} \tag{5.21}$$

and the unknown nodal values of $\mathcal{U}$ are

$$\left[\mathcal{U}_k^h\right] = \begin{bmatrix} u_m^h \\ \theta_n^h \end{bmatrix}. \tag{5.22}$$

5.2 Lagrange Multiplier Discretisation

Here we choose a quadratic shape function for the Lagrange multiplier $\boldsymbol{\lambda}$ as

$$\lambda = \boldsymbol{M}\boldsymbol{\lambda}_h, \tag{5.23}$$

where

$$\boldsymbol{M} = \begin{bmatrix} \zeta_3(2\zeta_3 - 1) & \zeta_1(2\zeta_1 - 1) & \zeta_2(2\zeta_2 - 1) \\ 4\zeta_1\zeta_3 & 4\zeta_1\zeta_2 & 4\zeta_2\zeta_3 \end{bmatrix}.$$

5.3 Computations in Curvilinear Coordinates

In contrast to the various quantities introduced in the curvilinear coordinate, the ultimate rule for actual computation of the tensor algebra is to stick to contravariant-covariant component multiplication. In the following sections, we shall illustrate this point with various evaluations. The local covariant bases are $\vec{g}^i, (i = 1, 2, 3)$ and its dual bases are $\vec{g}_i$.

5.3.1 Tensors Components Evaluation Basics

Second order identity tensor

$$\boldsymbol{I} = \delta^i{}_j \vec{g}_i \otimes \vec{g}^j = g^{ij}\vec{g}_i \otimes \vec{g}_j = g_{ij}\vec{g}^i \otimes \vec{g}^j, \tag{5.24}$$

Fourth order identity tensor

$$\mathsf{I} = \delta^i{}_k \delta^j{}_l \vec{g}_i \otimes \vec{g}^k \otimes \vec{g}_j \otimes \vec{g}^l = g_{ik}g_{jl}\vec{g}^i \otimes \vec{g}^k \otimes \vec{g}^j \otimes \vec{g}^l \tag{5.25}$$

Fourth order symmetric tensor

$$\mathsf{I}_{\mathsf{S}}^{ijkl} = \frac{1}{2}\left(g^{ik}g^{jl} + g^{il}g^{jk}\right) \tag{5.26}$$

and tensor

$$\boldsymbol{I} \otimes \boldsymbol{I} = g^{ij}g^{kl}\vec{g}_i \otimes \vec{g}_j \otimes \vec{g}_k \otimes \vec{g}_l \tag{5.27}$$

$$\mathsf{I}_{\mathsf{d}} = \mathsf{I}_{\mathsf{s}} - \frac{1}{3}\boldsymbol{I} \otimes \boldsymbol{I} \tag{5.28}$$

thus

$$\mathsf{I}_{\mathsf{d}}^{ijkl} = \frac{1}{2}\left(g^{ik}g^{jl} + g^{il}g^{jk}\right) - \frac{1}{3}g^{ij}g^{kl} \tag{5.29}$$

Euclidean Norm of a vector $\vec{v}$ in curvilinear coordinate is denoted as $\|\vec{v}\|$. Vector in

curvilinear coordinates:

$$\|\vec{v}\|^2 = \vec{v} \cdot \vec{v} = v_i g^{ij} v_j = v_i v^i, \tag{5.30}$$

Second-order tensor in curvilinear coordinates:

$$\|\boldsymbol{T}\|^2 = T_{ij} g^{ik} g^{jl} T_{kl}, \tag{5.31}$$

Sometimes, it is necessary to transform the tensor components from one form to another, which can be achieved with metric form g_{ij}. For example,

$$v^i = \vec{v} \cdot \vec{g}^i = (v_j \vec{v}^j) \cdot \vec{g}^i = g^{ij} v_j \tag{5.32}$$

and similarly

$$v_i = g_{ij} v^j \tag{5.33}$$

For a second order tensor

$$T_{ij} = g_{ik} T^k_{\cdot j} = g_{jk} T_i^{\cdot k} = g_{ik} g_{jl} T^{kl}. \tag{5.34}$$

5.3.2 Elasticity Law in Curvilinear Coordinates

The elasticity tensor $\boldsymbol{C}$ defined earlier has a different component form in the curvilinear system as

$$C^{ijkl} = L_1 g^{ij} g^{kl} + L_2 (g^{ik} g^{jl} + g^{il} g^{jk}), \tag{5.35}$$

where L_1 and L_2 are the Lame constants, formly noted by λ and μ. Note that the symmetric properties stated earlier stand true. The elasticity law therefore may be written as

$$\sigma^{ij} = C^{ijkl} \varepsilon_{kl}. \tag{5.36}$$

5.4 Permutation and Element Assembly

One key aspect of the finite element method is the element assembly procedure, which combines the individual element contribution into a whole. Technically, this procedure falls into the matrix assembly with the indexing function. Figure 5.2 shows a sparse pattern of a matrix undergoes an assembly process, where the order follows from top to bottom, left to right.

As the core task of element assembly is to determine where the the local stiffness (or force vector) matrix contribute to the the global matrix, i.e. to decide where the local stiffness matrix entry K_{ij} add up to the global matrix entry. Take an example of a linear finite element method problem, where the final problem to solve is the linear equation

$$\Big[K\Big] \Big[U\Big] = \Big[F\Big]. \tag{5.37}$$

The key to the understanding the order (or indexing, from the perspective of programming)

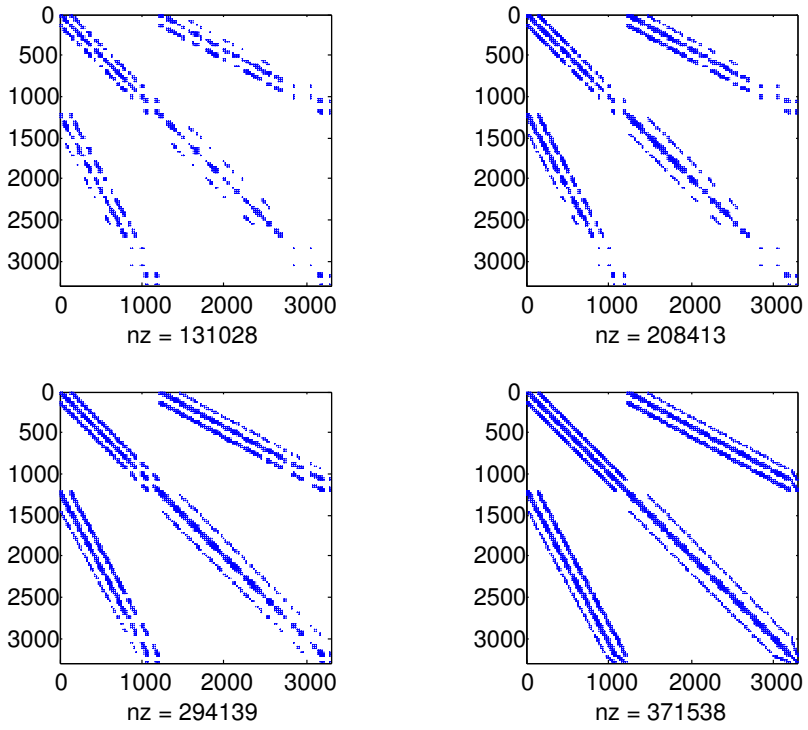


Figure 5.2 Illustration of an assembly result. "nz" stands for number of non-zero values in current matrix

is the association of two indices i, j of K_{ij} with both the unknown displacement variables and the right-hand side force variable. To simplify the indexing task, we first arrange local $\mathcal{U}_h$ in the order of first nodal contribution and then edge contributions in our numerical implementation, and then assembly from the element-wise (local) order to the global order (not to be confused with the terminology used in section 5.1) by a simple indexing function. The transformation of the orders of unknown is achieved by a permutation matrix outlined in (B.15). Mathematically, this transformation is expressed as

$$\boldsymbol{v} = P\bar{\boldsymbol{v}}, \tag{5.38}$$

where P is the orthonormal permutation matrix, $\bar{\boldsymbol{v}}$ is the unknown variables in its original order and $\boldsymbol{v}$ represents an order that suits a simple indexing function. Since a nonlinear finite element method boils down to a local variational form

$$\boldsymbol{f}(\bar{\boldsymbol{u}})^T\bar{\boldsymbol{v}} = 0,$$

a change of order for the local unknown variables (from $\bar{\boldsymbol{v}}$ to $\boldsymbol{v}$) means a permutation of $\boldsymbol{f}(\boldsymbol{u})$ at the local level is necessary. This can be seen from the following

$$\begin{matrix} \boldsymbol{f}(\bar{\boldsymbol{u}})^T\bar{\boldsymbol{v}} = 0 \\ \boldsymbol{f}(\bar{\boldsymbol{u}})^T P^T\boldsymbol{v} = 0 \end{matrix}, \tag{5.39}$$

so that we should permutate $\boldsymbol{f}$ by $P\boldsymbol{f}$ to get the global version. On the other hand, the

Jacobian matrix needed in the nonlinear solver is the one that computed based on the global order. To stress, this we put a superscript G for $\boldsymbol{f}$ and Jacobian $\boldsymbol{J}$

$$\boldsymbol{J}^G = \frac{\mathrm{d}\boldsymbol{f}^G}{\mathrm{d}u} = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{u}} P\boldsymbol{f}(\bar{\boldsymbol{u}}) = P\frac{\mathrm{d}\boldsymbol{f}(\bar{\boldsymbol{u}})}{\mathrm{d}\boldsymbol{u}} = P\frac{\mathrm{d}\boldsymbol{f}(\bar{\boldsymbol{u}})}{\mathrm{d}\bar{\boldsymbol{u}}}\frac{\mathrm{d}\bar{\boldsymbol{u}}}{\mathrm{d}\boldsymbol{u}} = PKP^T.$$

5.5 Tensor Vectorisation

To reduce the computational cost, we vectorise all relevant tensors based on their symmetry properties to save memory usage and computation operations. In light of the curvilinear coordinates, we have to consider about the appropriate component form, i.e. choose the covariant, contravariant or mixed form. From the perspective of tensors, the usual scalar and vector can be viewed as 0^{th} and 1^{st} order tensors, which do not need any special treatments. For the second order tensors $\boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}$, to ensure the elasticity law can be computed as

$$\sigma^{ij} = C^{ijkl}\varepsilon_{kl},$$

we convert them into

$$\begin{aligned}\left[\varepsilon_{ij}\right] &= \begin{bmatrix}\varepsilon_{11} & \varepsilon_{22} & \varepsilon_{33} & 2\varepsilon_{12} & 2\varepsilon_{23} & 2\varepsilon_{13}\end{bmatrix}^T \\ \left[\sigma^{ij}\right] &= \begin{bmatrix}\sigma^{11} & \sigma^{22} & \sigma^{33} & \sigma^{12} & \sigma^{23} & \sigma^{13}\end{bmatrix}^T\end{aligned}, \tag{5.40}$$

where the factor 2 accounts for the symmetry of the tensors. 3^{th} order tensor $\mathcal{B}$ defined in (4.168) and (4.169) can be cast into a matrix:

$$[\mathcal{B}_{ijk}] = \begin{bmatrix}\mathcal{B}_{11\cdot}\\ \mathcal{B}_{22\cdot}\\ \mathcal{B}_{33\cdot}\\ 2\mathcal{B}_{12\cdot}\\ 2\mathcal{B}_{23\cdot}\\ 2\mathcal{B}_{13\cdot}\end{bmatrix} = \begin{bmatrix}\mathcal{B}_{111} & \mathcal{B}_{112} & \mathcal{B}_{113} & \cdots & \mathcal{B}_{11n}\\ \mathcal{B}_{221} & \mathcal{B}_{222} & \mathcal{B}_{223} & \cdots & \mathcal{B}_{22n}\\ \mathcal{B}_{331} & \mathcal{B}_{332} & \mathcal{B}_{333} & \cdots & \mathcal{B}_{33n}\\ 2\mathcal{B}_{121} & 2\mathcal{B}_{122} & 2\mathcal{B}_{123} & \cdots & 2\mathcal{B}_{12n}\\ 2\mathcal{B}_{231} & 2\mathcal{B}_{232} & 2\mathcal{B}_{233} & \cdots & 2\mathcal{B}_{23n}\\ 2\mathcal{B}_{131} & 2\mathcal{B}_{132} & 2\mathcal{B}_{123} & \cdots & 2\mathcal{B}_{13n}\end{bmatrix} \tag{5.41}$$

so that

$$[\varepsilon_{ij}] = [\mathcal{B}_{ijk}]\mathcal{U}_k$$

holds. Due to the kinematical assumption made in (4.152), ε_{33} and $\mathcal{B}_{33\cdot}$ are all equal to zero, thus we may only reduce one dimension (row) in the actual implementations. The

symbols defined in (4.165) can also similarly vectorise as matrices without factor 2 as

$$\Lambda^{\gamma} = [\Lambda^{\gamma}_{\alpha\beta k}] = \begin{bmatrix} (\Lambda^{\gamma}_{11\cdot})^T & (\Lambda^{\gamma}_{22\cdot})^T & (\Lambda^{\gamma}_{12\cdot})^T \end{bmatrix}^T, \tag{5.42a}$$

$$\Lambda^{\chi_\theta} = [\Lambda^{\chi_\theta}_{\alpha\beta k}] = \begin{bmatrix} (\Lambda^{\chi_\theta}_{11\cdot})^T & (\Lambda^{\chi_\theta}_{22\cdot})^T & (\Lambda^{\chi_\theta}_{12\cdot})^T \end{bmatrix}^T, \tag{5.42b}$$

$$\Lambda^{\chi_u} = [\Lambda^{\chi_u}_{\alpha\beta k}] = \begin{bmatrix} (\Lambda^{\chi_u}_{11\cdot})^T & (\Lambda^{\chi_u}_{22\cdot})^T & (\Lambda^{\chi_u}_{12\cdot})^T \end{bmatrix}^T, \tag{5.42c}$$

$$\Lambda^{\chi_\kappa} = [\Lambda^{\chi_\kappa}_{\alpha\beta k}] = \begin{bmatrix} (\Lambda^{\chi_\kappa}_{11\cdot})^T & (\Lambda^{\chi_\kappa}_{22\cdot})^T & (\Lambda^{\chi_\kappa}_{12\cdot})^T \end{bmatrix}^T, \tag{5.42d}$$

$$\Lambda^{\zeta_\theta} = [\Lambda^{\zeta_\theta}_{\alpha k}] = \begin{bmatrix} (\Lambda^{\zeta_\theta}_{2\cdot})^T & (\Lambda^{\zeta_\theta}_{1\cdot})^T \end{bmatrix}^T, \tag{5.42e}$$

$$\Lambda^{\zeta_u} = [\Lambda^{\zeta_u}_{\alpha k}] = \begin{bmatrix} (\Lambda^{\zeta_u}_{2\cdot})^T & (\Lambda^{\zeta_u}_{1\cdot})^T \end{bmatrix}^T, \tag{5.42f}$$

so that we can have

$$\begin{aligned} \left[\mathcal{B}_{ijk}\right] &= \mathbf{diag}(I_{2\times 2}, 2I_{3\times 3}) \begin{bmatrix} \mathcal{B}^T_{11\cdot} & \mathcal{B}^T_{22\cdot} & \mathcal{B}^T_{12\cdot} & \mathcal{B}^T_{23\cdot} & \mathcal{B}^T_{13\cdot} \end{bmatrix}^T \\ &= \begin{bmatrix} I_{2\times 2} & 0 \\ 0 & 2I_{3\times 3} \end{bmatrix} \begin{bmatrix} \Lambda^{\gamma} + \xi^3 \Lambda^{\chi_u} & \xi^3 \Lambda^{\chi_\theta} - (\xi^3)^2 \Lambda^{\kappa} \\ \Lambda^{\zeta_u} & \Lambda^{\zeta_\theta} \end{bmatrix}. \end{aligned} \tag{5.43}$$

For those fourth-order tensors, the idea of exploiting the symmetry of the tensor is first illustrated for elasticity tensor $\boldsymbol{C}$ as expressed by a compact form in (3.30), which reduces the 81-component fourth-order symmetric tensor to a 6×6 matrix. In the curvilinear coordinate, the symmetry property stays true, except the contravariant components are expressed in (5.35). In particular, the normal stress component σ^{33} in the Reissner-Mindlin theory is also assumed to be zero, which unfortunately contradicts with general elasticity law since generally $\sigma^{33} = C^{33kl}\varepsilon_{kl} \neq 0$. As a result, in our implementation the vanishing dimension (indice 33) in strain and stress tensors allows a 5-by-5 matrix implementation for elasticity tensor:

$$\left[C^{ijkl}\right] = \begin{bmatrix} C^{1111} & C^{1122} & C^{1133} & C^{1112} & C^{1123} & C^{1113} \\ C^{2211} & C^{2222} & C^{2233} & C^{2212} & C^{2223} & C^{2213} \\ C^{3311} & C^{3322} & C^{3333} & C^{3312} & C^{3323} & C^{3313} \\ C^{1211} & C^{1222} & C^{1233} & C^{1212} & C^{1223} & C^{1213} \\ C^{2311} & C^{2322} & C^{2333} & C^{2312} & C^{2323} & C^{2313} \\ C^{1311} & C^{1322} & C^{1333} & C^{1312} & C^{1323} & C^{1313} \end{bmatrix}. \tag{5.44}$$

Let

$$C_A = \begin{bmatrix} (L_1 + 2L_2)g^{11}g^{11} & L_1 g^{11}g^{22} + 2L_2 g^{12}g^{12} & L_1 g^{11} & L_1 g^{11}g^{12} + 2L_2 g^{11}g^{12} \\ L_1 g^{22}g^{11} + 2L_2 g^{12}g^{12} & (L_1 + 2L_2)g^{22}g^{22} & L_1 g^{22} & L_1 g^{22}g^{12} + 2L_2 g^{12}g^{22} \\ L_1 g^{11} & L_2 g^{22} & L_1 + 2L_2 & L_1 g^{12} \\ L_1 g^{12}g^{11} + 2L_2 g^{11}g^{12} & L_1 g^{22}g^{12} + 2L_2 g^{12}g^{22} & L_1 g^{12} & L_1 g^{12}g^{12} + 2L_2 g^{11}g^{22} \end{bmatrix},$$

and

$$C_B = \begin{bmatrix} L_2 g^{22} & L_2 g^{12} \\ L_2 g^{12} & L_2 g^{11} \end{bmatrix},$$

we can have

$$\left[C^{ijkl}\right] = \begin{bmatrix} C_A & \mathbf{0} \\ \mathbf{0} & C_B \end{bmatrix}.$$

Therefore the matrix form of $[\mathcal{B}_{ijk}]^T[C^{ijkl}][\mathcal{B}_{ijk}]$ gives correct result for the expression $\mathcal{B}^T\boldsymbol{C}\mathcal{B}$, whose definition in component form is

$$\mathcal{B}^T\boldsymbol{C}\mathcal{B} = \mathcal{B}^T_{pij} C^{ijkl} \mathcal{B}_{klq} \tilde{\boldsymbol{e}}_p \otimes \tilde{\boldsymbol{e}}_q.$$

Similarly, the computation with matrix form for the tensor product $\mathcal{B}^T\boldsymbol{\sigma}$ also stay correct.

Christoffel symbols $\Gamma^\lambda_{\alpha\beta}$ can be stored as a 3×2 matrix due to their symmetry property ($\Gamma^\lambda_{\alpha\beta} = \Gamma^\lambda_{\beta\alpha}$)as

$$\left[\Gamma^\lambda_{\alpha\beta}\right] = \begin{bmatrix} \Gamma^1_{11} & \Gamma^1_{22} & \Gamma^1_{12} \\ \Gamma^2_{11} & \Gamma^2_{22} & \Gamma^2_{12} \end{bmatrix}^T \tag{5.45}$$

Similar to elasticity tensor $\boldsymbol{C}$, we can store $\mathsf{I}^{ijkl}_{\mathsf{d}}$ as

$$\left[\mathsf{I}^{ijkl}_{\mathsf{d}}\right] = \begin{bmatrix} \mathsf{I}^{1111}_{\mathsf{d}} & \mathsf{I}^{1122}_{\mathsf{d}} & \mathsf{I}^{1133}_{\mathsf{d}} & \mathsf{I}^{1112}_{\mathsf{d}} & \mathsf{I}^{1123}_{\mathsf{d}} & \mathsf{I}^{1113}_{\mathsf{d}} \\ \mathsf{I}^{2211}_{\mathsf{d}} & \mathsf{I}^{2222}_{\mathsf{d}} & \mathsf{I}^{2233}_{\mathsf{d}} & \mathsf{I}^{2212}_{\mathsf{d}} & \mathsf{I}^{2223}_{\mathsf{d}} & \mathsf{I}^{2213}_{\mathsf{d}} \\ \mathsf{I}^{3311}_{\mathsf{d}} & \mathsf{I}^{3322}_{\mathsf{d}} & \mathsf{I}^{3333}_{\mathsf{d}} & \mathsf{I}^{3312}_{\mathsf{d}} & \mathsf{I}^{3323}_{\mathsf{d}} & \mathsf{I}^{3313}_{\mathsf{d}} \\ \mathsf{I}^{1211}_{\mathsf{d}} & \mathsf{I}^{1222}_{\mathsf{d}} & \mathsf{I}^{1233}_{\mathsf{d}} & \mathsf{I}^{1212}_{\mathsf{d}} & \mathsf{I}^{1223}_{\mathsf{d}} & \mathsf{I}^{1213}_{\mathsf{d}} \\ \mathsf{I}^{2311}_{\mathsf{d}} & \mathsf{I}^{2322}_{\mathsf{d}} & \mathsf{I}^{2333}_{\mathsf{d}} & \mathsf{I}^{2312}_{\mathsf{d}} & \mathsf{I}^{2323}_{\mathsf{d}} & \mathsf{I}^{2313}_{\mathsf{d}} \\ \mathsf{I}^{1311}_{\mathsf{d}} & \mathsf{I}^{1322}_{\mathsf{d}} & \mathsf{I}^{1333}_{\mathsf{d}} & \mathsf{I}^{1312}_{\mathsf{d}} & \mathsf{I}^{1323}_{\mathsf{d}} & \mathsf{I}^{1313}_{\mathsf{d}} \end{bmatrix}. \tag{5.46}$$

and explicitly

$$\left[\mathsf{I}^{ijkl}_{\mathsf{d}}\right] = \begin{bmatrix} \frac{2}{3}\left(g^{11}\right)^2 & \left(g^{12}\right)^2 - \frac{1}{3}g^{11}g^{22} & -\frac{1}{3}g^{11} & \frac{2}{3}g^{11}g^{12} & 0 & 0 \\ \left(g^{12}\right)^2 - \frac{1}{3}g^{11}g^{22} & \frac{2}{3}\left(g^{22}\right)^2 & -\frac{1}{3}g^{22} & \frac{2}{3}g^{12}g^{22} & 0 & 0 \\ -\frac{1}{3}g^{11} & -\frac{1}{3}g^{22} & \frac{2}{3} & -\frac{1}{3}g^{12} & 0 & 0 \\ \frac{2}{3}g^{12}g^{11} & \frac{2}{3}g^{12}g^{22} & -\frac{1}{3}g^{12} & \frac{1}{2}g^{11}g^{22} + \frac{1}{6}\left(g^{12}\right)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}g^{22} & \frac{1}{2}g^{21} \\ 0 & 0 & 0 & 0 & \frac{1}{2}g^{12} & \frac{1}{2}g^{11} \end{bmatrix}. \tag{5.47}$$

Next, we consider implementing the devatoric decomposition of the strain tensor. The basic equations are:

$$\boldsymbol{\varepsilon}^{ij}_d = \mathsf{I}^{ijkl}_{\mathsf{d}} \varepsilon_{kl}, \quad \varepsilon_v = \boldsymbol{I}^{ij}\varepsilon_{ij} \tag{5.48}$$

In matrix form, we can have

$$\left[\boldsymbol{\varepsilon}_d^{ij}\right] = \left[\mathsf{I}_{\mathsf{d}}^{ijkl}\right]\left[\varepsilon_{kl}\right], \quad \varepsilon_v = \left[\boldsymbol{I}^{ij}\right]^T\left[\varepsilon_{ij}\right], \tag{5.49}$$

where

$$\left[\boldsymbol{I}^{ij}\right] = \begin{bmatrix} g^{11} & g^{22} & g^{33} & g^{12} & g^{23} & g^{13} \end{bmatrix}^T.$$

Note that in the above implementation, there is no factor 2 in matrix $\left[\boldsymbol{\varepsilon}_d^{ij}\right]$, from which the deviatoric elastic stress tensor can be computed as

$$\left[\boldsymbol{s}^{ij}\right] = 2\mu\left[\boldsymbol{\varepsilon}_d^{ij}\right]. \tag{5.50}$$

Also we want to compute

$$\|\boldsymbol{s}\|^2 = \boldsymbol{s}^{ij} g_{ik} g_{jl} \boldsymbol{s}^{kl}, \tag{5.51}$$

effectively. One vectorised approach to compute this value appears to first reconstruct the full matrix $\boldsymbol{s}^{ij}$ as $[\boldsymbol{s}]$, compute matrices

$$\left[\boldsymbol{s}_{\cdot l}^{i}\right] = \boldsymbol{s}g, \quad \left[\boldsymbol{s}_{i}^{\cdot l}\right] = g\boldsymbol{s}, \tag{5.52}$$

where g is the metric tensor matrix, and finally compute an inner product between fully vectorised matrices $\left[\boldsymbol{s}_{\cdot l}^{i}\right]$ and $\left[\boldsymbol{s}_{i}^{\cdot l}\right]$. To construct a fourth-order tensor from a symmetric second-order tensor $\hat{\boldsymbol{n}}$, we can simply use the following formula

$$\left[(\hat{\boldsymbol{n}} \otimes \hat{\boldsymbol{n}})^{ijkl}\right] = \left[\hat{\boldsymbol{n}}^{ij}\right]\left[\hat{\boldsymbol{n}}^{kl}\right]^T, \tag{5.53}$$

where no factor 2 is imposed in $\left[\hat{\boldsymbol{n}}^{ij}\right]$.

5.6 Mesh Generating

For our problem, the mesh is generated using MATLAB's `delaunay` function. The element for reference domain is a triangle *prism*, sketched in Figure 5.3, whose middle surface mesh is shown in Figure 5.4.

5.7 Numerical Integration Scheme

One aspect of the discretisation of the finite element method is the evaluation of various integrations (the other one is the discretisation of the continuous problem). The accuracy of the numerical integration directly affects the credibility of the numerical results. This section covers the basic numerical integration scheme used in our computational work.

The numerical integration is to evaluate function $f(\boldsymbol{x})$ over the domain Ω by the general form

$$I(f) = \int f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \approx \sum_i^N w_i f(\boldsymbol{x}_i), \tag{5.54}$$

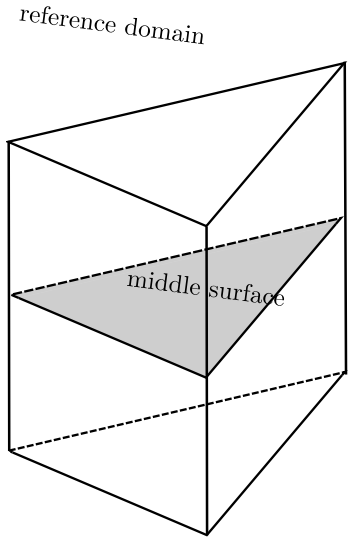


Figure 5.3 Sketch of the prism element where the middle surface is highlighted.

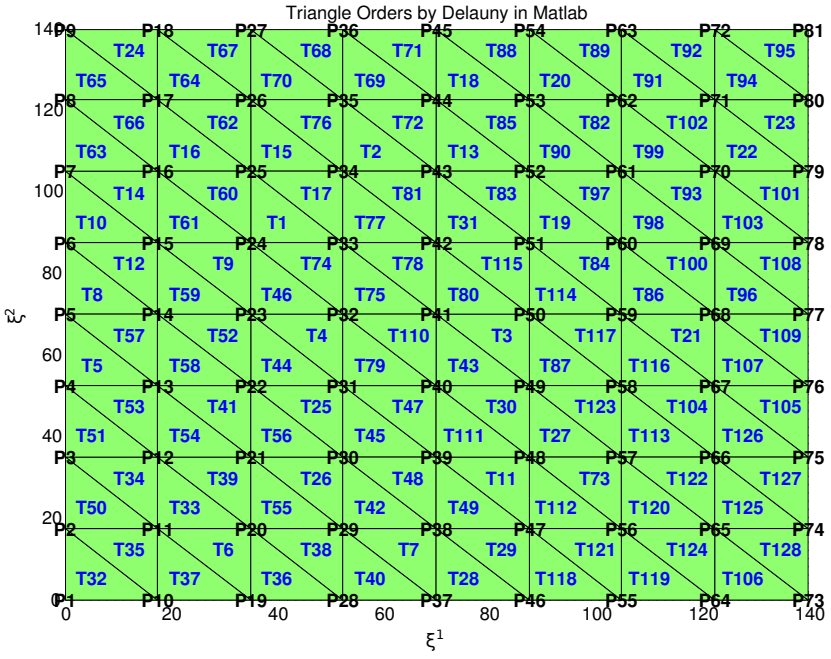


Figure 5.4 Middle surface mesh generated by `delaunay` algorithm

w_i	ζ_1	ζ_2	ζ_3	Multiplicity
0.05084 49063 70207	0.87382 19710 16996	0.06308 90144 91502	0.06308 90144 91502	3
0.11678 62757 26379	0.50142 65096 58179	0.24928 67451 70910	0.24928 67451 70911	3
0.08285 10756 18376	0.63650 24991 21399	0.31035 24510 33785	0.05314 50498 44816	6

Table 5.1 Numerical integration points from [108]

where w_i denotes the weight corresponding to integration point $\boldsymbol{x}_i$. The integration scheme is to determine the number of integration points of a given domain and the corresponding weights such that the numerical integration result computed by (5.54) is an appropriate approximation of the exact value. If the integral function f is a polynomial, the numerical scheme can be made exact by some integration points N. The numerical integration formulae is referred as of degree d, if it is exact for polynomial of order $\leq d$ but not for polynomial degree $\geq d+1$. The numerical integration in 1D is known as *quadrature* due to its geometrical meaning [103] and the well known Gaussian quadrature solves the numerical integration problem elegantly. In our case, the majority of the integrations are either surface integrations or volume integrations and the numerical integration scheme for these higher dimensions are sometimes described as *cubature*. The studies in this area have been extensive and the need for efficient integration schemes in finite element context has revives the interests [104, 105, 106, 107].

Here we consider the cases of triangle and triangular prism as the 2D and 3D integration domain in our computational work. For the surface integration, we mostly deal with integrand like $\boldsymbol{B}^T\boldsymbol{\sigma}$. Consider equations (4.168), (4.169) and (5.17), since the tensor $\boldsymbol{B}$ is of order 6, we can use a 12 points integration scheme given in [108], which is shown in Table 5.1. The curbature rule for Table 5.1 is

$$\iint_{\triangle} f \,\mathrm{d}S = A_e \sum_{i}^{m} w_i f(\zeta_1^{(i)}, \zeta_2^{(i)}, \zeta_3^{(i)}), \tag{5.55}$$

where A_e denotes the area of the triangle $\triangle$, and $\zeta_1, \zeta_2, \zeta_3$ are the barycentric coordinates of the sampling points within the triangle and the *multiplicity* indicates the symmetry of the sampling points. So a sampling point $(\alpha, \beta, \beta), (\alpha \neq \beta)$ would have a multiplicity factor of 3, since the other two symmetry points are (β, α, β) and (β, β, α). Similarly, a multiplicity factor of 6 implies six sampling points, namely $(\alpha, \beta, \gamma), (\alpha \neq \beta \neq \gamma), (\alpha, \gamma, \beta), (\beta, \alpha, \gamma), (\beta, \gamma, \alpha), (\gamma, \alpha, \beta)$ and (γ, β, α). As for the 3D integration, one of the simple numerical integration scheme is by using the product rule, i.e. we can integrate the third dimension after we have obtained the surface integral. A standard 3-point Gaussian quadrature sampling points for integral $\int_{-1}^{1} f(x)\,\mathrm{d}x$ is listed in Table 5.2.

$$\int_{-1}^{1} f(x)\,\mathrm{d}x = \sum_{i}^{3} w_i f(x_i) \tag{5.56}$$

w_i	x_i
$\sqrt{\frac{3}{5}}$	$5/9$
0	$8/9$
$-\sqrt{\frac{3}{5}}$	$5/9$

Table 5.2 Gaussian quadrature integration points for 1D problem $\int_{-1}^{1} f(x)\,\mathrm{d}x$

and

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} f(\xi)\,\mathrm{d}\xi = \frac{1}{2}\int_{-1}^{1} f\left(\frac{1}{2}x\right)\,\mathrm{d}x \tag{5.57}$$

We adopt the product rule for the numerical integration scheme, although more sophisticated numerical integration is available such as the one in [109]. In addition, the integration transformation between a physical domain and a reference domain is through

$$\int_{\Omega} \mathrm{d}V = \int_{\Omega} \sqrt{g}\,\mathrm{d}\xi^1\,\mathrm{d}\xi^2\,\mathrm{d}\xi^3.$$

5.8 Preliminary Results

5.8.1 Settings

To use the full piecewise description of the tool shape will incur nonsmoothness to the functional optimisation and we simplified to a hemisphere one as the contact is developed around only over this region. The material parameters we use are:

- Elasticity constant $E = 7.0e10$,
- Poisson's ration $\nu = 0.33$,
- Sheet thickness $h = 1$,
- Shear modulus $\approx 26e9$ (computed from elasticity constant),
- Bulk modulus $\approx 69e9$ (computed from elasticity constant),
- Yield stress $\sigma_Y = 130e6$,
- Hardening coefficient $H = 0$.

5.8.2 Deformation Results

We first set the tool position near the edge of the sheet, as shown in Figure 5.5, where red region hightlights the contact boundary that is used in the computation. A visualised displacement field is shown in Figure 5.6. We can observe that although the deformation is above the tool, which satisfies the contact boundary, the "ridges" around the contact region suggests some problem with the programme. For the results when the tool is the

middle area of the sheet, shown in Figure 5.7, where the red region is the contact area and the numerical results visualised is shown in Figure 5.8. A different orientation given in Figure 5.9 shows some positions are below the initial sheet. To do further debugging, a decompose of the component of the system and write test examples may be possible.

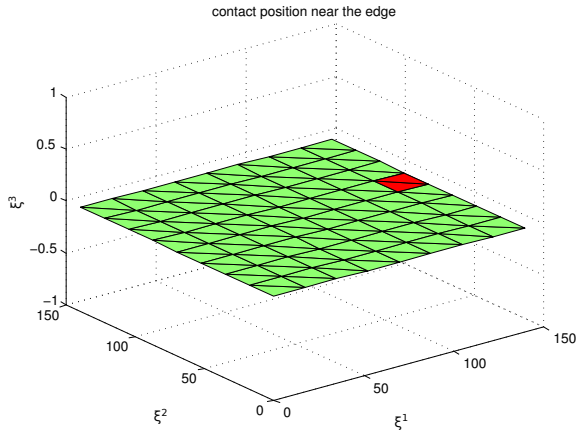


Figure 5.5 Illustration of the contact region near one of the edges (shown as red region)

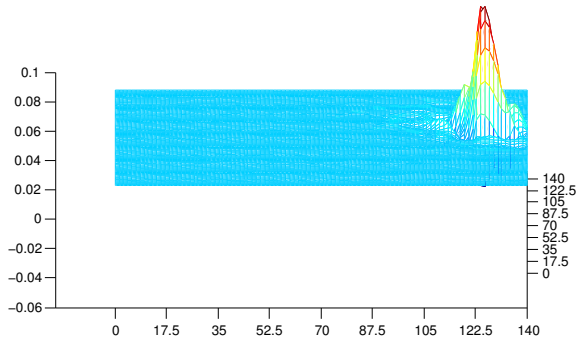


Figure 5.6 Visualised numerical result for a case where the tool position is near the edge

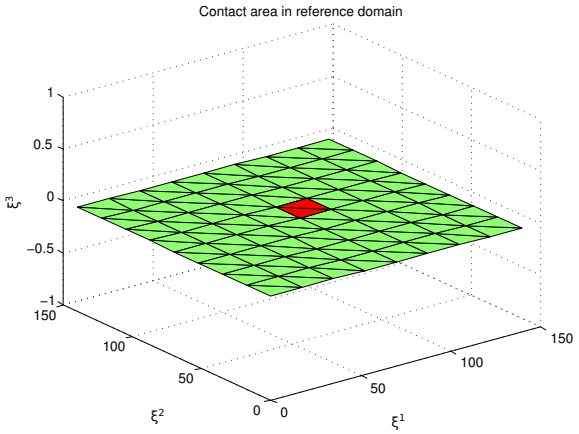


Figure 5.7 Illustration of the middle contact region (shown as red region)

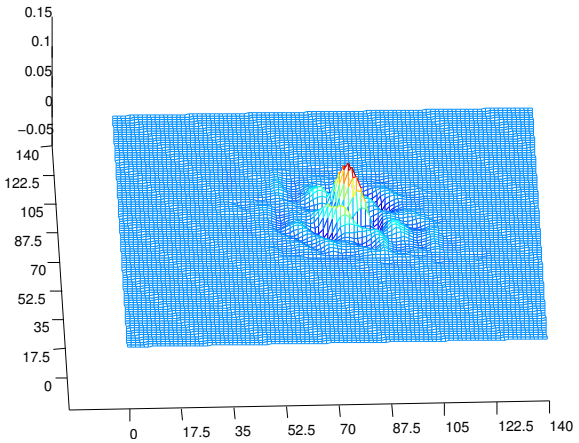


Figure 5.8 Visualised numerical result

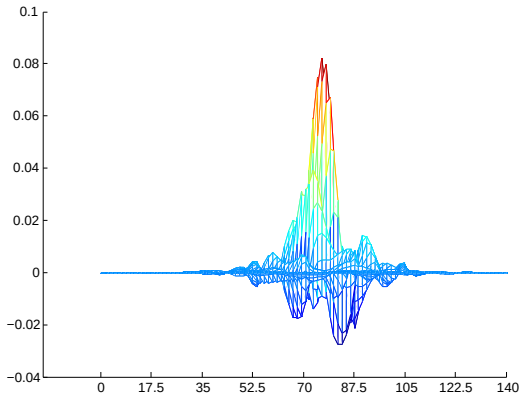


Figure 5.9 Same numerical result as in Figure 5.8 with a different angle

Chapter 6

Conclusions

6.1 Research Problem Recap

The research problem is about controlling of a relatively novel progressive metal forming process, incremental sheet forming (ISF), where the deformation occurs locally around the point of contact between a tool and the metal sheet. The tool and the sheet metal are mounted in a CNC milling machine, and the final work-piece is formed cumulatively by the movements of the tool. Unlike traditional stamping processes, where specific punches and dies are needed to produce the desired shape, the dieless ISF process is flexible enough to use one universal tool to deform the blank sheet to various required states subject to the prescribed tool trajectory. This cost efficient process would appeal to the one-off and low batch production, such as the automobile and aerospace industry. However, the geometrical accuracy of the final shape made by the ISF process is less accuracy than stamping processes due to the elastic springback phenomenon. On the laboratory scale, the accuracy of the parts created by ISF is between ± 1.5 mm and ± 3 mm, but in order for ISF to be competitive with a stamping process, an accuracy of below ± 1.0 mm and more realistically below ± 0.2 mm would be needed [6]. This motivates our research into the studies of the deformation mechanism and a proper control method to reduce the geometrical deviations between the target shape and the actual shape made by ISF.

Although other mechanical configurations of the ISF process, such as using multiple tools and support dies, may offer benefits towards a better geometrical accuracy, we focused exclusively on the single point ISF process in this work as to push the boundary of our control methodology as well as the fundamental understanding of the deformation process. In addition, our control effort is on optimising the tool trajectory, which has a direct effect on the final shape outcome. The tool movement is controlled by a servomotor, which can be programmed with movement parameters from our control algorithm.

The actual deformation process involves bending and stretching, elastic and plastic deformation and also the effects of tool movements, making the ISF process highly non-linear. This challenge poses difficulties in designing a practical controller that is both simple and effective to run on our rig. To circumvent this issue, we may first discretise

the deformation states of the underlying sheet metal into a number of intermediate layers by breaking down the supposedly continuous tool trajectory into vertical and horizontal parts.

6.2 Research Findings

When consider applying modern control theory, especially a model based one, to the ISF process—a deterministic multi-stage system, our starting decision was to juggle between the complexity of the ISF model and the accuracy required for the controller. The decision to model the deformation of sheet metal material in ISF process as a simplified linear phenomenal model is backed with the belief that the available feedback information about the state of the deformation from employing closed loop strategy would compensate the uncertainties in the linear model. As explained in Chapter 2, a stereo camera is installed below the main frame, where the sheet metal is hold, and the geometrical aspects of the deformed metal sheet thus becomes available.

In the end, the deformation model is linearised around a known trajectory, i.e. the conventional contour following tool-path. For every new type of shape, it is necessary to run the open loop contour following tool trajectory once to record measured data at each layer, and process those data as the parameters of our linearised model, notably $\mathbf{B}$ matrix (cf. p28). The resulting constrained model predictive control (MPC), where the online optimal control problem is solved at each vertical input, is implemented at the rig and the experimental results produced for both truncated cone and truncated square pyramids.

The geometrical deviations for the final truncated cones can be improved within around ± 0.2 mm for most of the part (see Figure 2.18 and Figure 2.19); for the truncated square pyramids the deviations understandably deteriorate to around ± 0.45 mm for the corner profile, whereas the middle profile is slightly better at the value ± 0.3 mm (see Figure 2.25). The results on both types of shapes have demonstrated the ability of the MPC scheme, especially the advantage over the conventional contour following type of tool trajectory.

One of the crucial components of predictive controller design is the choice of an appropriate system model. As we have demonstrated, the linear model adopted in this work functions well for making shapes with simple geometries, but a careful examination of the experimental data reveals the fact that the model is working outside theoretically valid region, which will limit its ability to make accurate complex shapes. The initial decision to adopt a simple linear model also takes into account the requirements of feasible online computations for MPC. The actual experiment time for producing a truncated cone or a truncated square pyramid in the lab, including the CNC machine working time and the computation time needed for MPC, is just around acceptable an hour.

Quite often the computational difficulties and the daunting complexity put off the idea of applying a physics based modelling. The second part of the thesis is an attempt to address this issue head-on by studying continuum mechanics of the ISF, as we try to understand the full scale of the nonlinearity character of the process and explore the

possibilities of applying modern control strategy.

In the end, the ISF process is treated as a quasi-static contact problem, where the loading is imposed by the tool movements and the contact condition is modelled by Signorin's type of boundary value conditions. The material deformation on the other hand is regarded as elasto-plastic and the material law is modelled by von Misers yield criterion. Before we finalise the model formulation, a thin shell theory is introduced to take care of the geometrical features of deformed sheet metal. The technique is popular in complex geometries modelling, such as deformation for wind turbines [97].

To solve the model like ours, generally it would require variational method, which has been shown in previous chapters. We further developed the computational aspects by adopting augmented Lagrangian method in function space and finite element method. The difficulties (nonlinearity) of the mathematical model becomes clearer, and can be partially explained in Figure 6.1. There are two major components in a contact problem model: the contact condition and material law, of which the former is involves nonlinearity (complementarity condition) and the latter differs in terms whether it is elastic or not. So the nonlinearity in ISF results from both the contact condition and nonlinear material behaviour. As a result, the discretised problem usually involves nonlinear programming.

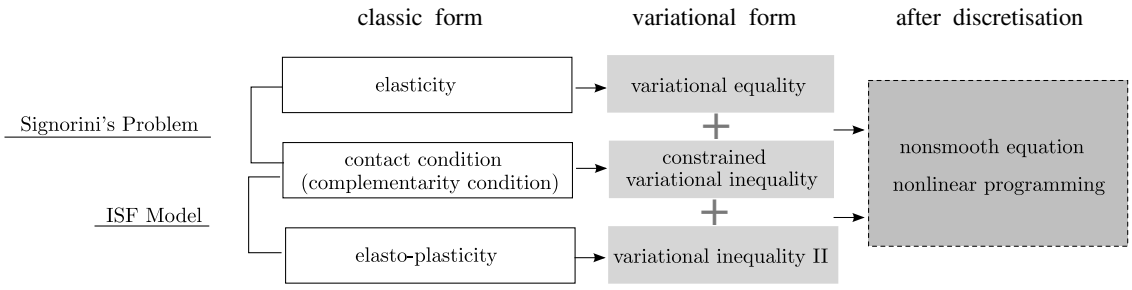


Figure 6.1 Illustration the mathematical framework of the Signorini's problem and our ISF model

Our model can be transferred to the well established mathematical tool, i.e. variational inequality and therefore, our work opens the door for modelling a general mechanical system that both contact and elasto-plastic deformation take place in an elegant mathematical framework, which allows further optimal control design.

6.3 Future Work

The constitutive ISF model developed in Chapter 3 can be cast as a variational inequality of the second kind (VI II) [110], and we may well use optimal control of VI or optimal control of partial differential equations [111] to carry out further study. Usually the necessary conditions to find the optimal solution is expressed as another variational inequality [112], whose computational method can be the augmented Lagrangian method (ALM) and the procedure of ALM has already been shown in previous chapters. In the inevitable numerical solution, the finite element method is also list in this work.

The closed-loop control is not new to metal forming process, such as the closed-loop

blankholder force control in [113], we may incorporate optimal control to achieve a finer version of MPC scheme as explained in the first part of this thesis.

One of the assumptions made in the elasto-plasticity modelling of the ISF process is the small deformation assumption, which has been reported to be accurate given our input is relatively small. Naturally, one might expect large deformations in the metal deformation field, but one should also be aware of the increase of the computational cost and the theoretical complexity in future work.

Having spent much time discussing applying optimal control strategy on an infinite system, other control concept, such as robust control method to deal with uncertainties, could also be explored, although we did not obtain a satisfying result (too conservative, not shown in the current thesis).

Appendix A

Vector and Tensor Algebra

Summation convention: Any index that occurs twice in a term implies summation, i.e.

$$a_i b_i = \sum_1^n a_i b_i, \quad \zeta_{ii} = \sum_1^n \zeta_{ii}.$$

In this work, we will use boldface italic letter to denote vector and tensors. Vectors are denoted by lowercase Latin letters, second-order tensors by lowercase Greek letter and fourth-order tensor by a uppercase Latin letter. Summation convention is used for repeated indices, unless elsewhere stated. As the scale of the work in this thesis is most often limited to three-dimensional space and we will make use of the Cartesian coordinate system whose orthonormal basis is represented as $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$, as a convention, the lowercase Greek letters such as α, β, γ take values of 1 and 2 as subscripts, whilst lowercase i, j, k take values of $1, 2$ and 3 as subscripts.

A second-order tensor $\boldsymbol{\tau}$ is a linear operator that maps vectors to vectors space and may be represented as a matrix. For a vector $\boldsymbol{a}$, $\boldsymbol{\tau a}$ is a vector such that the action of $\boldsymbol{\tau}$ on $\boldsymbol{a}$ is linear. Using the defined orthonormal basis, the mapping on vector $\boldsymbol{a}$ with tensor $\boldsymbol{\tau}$ can be expressed as:

$$\boldsymbol{\tau a} = \tau_{ij} a_j \boldsymbol{e}_i. \tag{A.1}$$

Dot product or **scalar product** between two vectors or tensors are defined as

$$\boldsymbol{a} \cdot \boldsymbol{b} = a_i b_i, \tag{A.2}$$

$$\boldsymbol{\sigma} : \boldsymbol{\tau} = \sigma_{ij} \tau_{ij}. \tag{A.3}$$

Tensor product Let V, W be two vector spaces, a new vector space from tensor product $V \otimes W$ has two properties:

- if $\boldsymbol{v} \in V$ and $\boldsymbol{w} \in W$ then there is a product $\boldsymbol{v} \otimes \boldsymbol{w} \in V \otimes W$;
- the product is bilinear.

The mapping of vector product between $\boldsymbol{a}$ and $\boldsymbol{b}$ is a second-order tensor defined by

$$(\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{c} = (\boldsymbol{b} \cdot \boldsymbol{c})\boldsymbol{a} \quad \forall \boldsymbol{c}. \tag{A.4}$$

The matrix form is

$$\boldsymbol{a} \otimes \boldsymbol{b} = \boldsymbol{a}\boldsymbol{b}^T.$$

Similarly, quantities $\boldsymbol{e}_i \otimes \boldsymbol{e}_j$ form the basis for the space of the second-order tensors, and thus any such tensor can be represented in the form:

$$\boldsymbol{\tau} = \tau_{ij}\boldsymbol{e}_i \otimes \boldsymbol{e}_j.$$

Suppose we have a second order tensor $\boldsymbol{\tau}$ that can be constructed with two vector

$$\boldsymbol{\tau} = \boldsymbol{c} \otimes \boldsymbol{d},$$

following (A.4), we can have the formula

$$\boldsymbol{a}(\boldsymbol{b} \cdot \boldsymbol{\tau}\boldsymbol{v}) = \boldsymbol{a}(\boldsymbol{b} \cdot \boldsymbol{c})(\boldsymbol{d} \cdot \boldsymbol{v}) = (\boldsymbol{b} \cdot \boldsymbol{c})(\boldsymbol{a} \otimes \boldsymbol{d})\boldsymbol{v}. \tag{A.5}$$

Provided the bases associated with above tensors are orthogonal, we may further derive the computational matrix format as

$$\left[\boldsymbol{a}\boldsymbol{b}^T\right][\boldsymbol{\tau}][\boldsymbol{v}]. \tag{A.6}$$

A fourth-order tensor $\boldsymbol{C}$ may be defined as a linear operator mapping spaces of second-order tensors into itself. The action on second-order tensor $\boldsymbol{\tau}$ with fourth-order tensor $\boldsymbol{C}$ is denoted as $\boldsymbol{C}\boldsymbol{\tau}$, having the component form $C_{ijkl}\tau_{kl}$. The canonical orthonormal basis is $\boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k \otimes \boldsymbol{e}_l$, where $1 \leq i, j, k, l \leq 3$.

In the light of the definitions in (A.1), (A.3) and (A.4), we may define the scalar tensor product operator ':' between two tensors of different orders as

$$\boldsymbol{\tau} : \boldsymbol{a} = \boldsymbol{\tau}\boldsymbol{a}, \tag{A.7}$$

where tensor $\boldsymbol{\tau}$ has a higher order than $\boldsymbol{a}$. Note that the scalar tensor product operator ':' and construct operator '$\otimes$' have lower order of operations than that of the mapping action, for example,

$$\begin{aligned} \boldsymbol{a} \otimes \boldsymbol{\tau}\boldsymbol{b} &= \boldsymbol{a} \otimes (\boldsymbol{\tau}\boldsymbol{b}) \\ \boldsymbol{\tau} : \boldsymbol{C}\boldsymbol{\beta} &= \boldsymbol{\tau} : (\boldsymbol{C}\boldsymbol{\beta}). \end{aligned} \tag{A.8}$$

Two Important Relations First, the product among second-order tensors (according

to multi-linear transformation definition):

$$\begin{aligned}(\boldsymbol{\alpha}\otimes\boldsymbol{\tau})\,\boldsymbol{\beta} &= \alpha_{ij}\tau_{kl}\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l\,[\beta_{mn}\boldsymbol{e}_m\otimes\boldsymbol{e}_n]\\ &= \alpha_{ij}\tau_{kl}\beta_{mn}\boldsymbol{e}_i\otimes\boldsymbol{e}_j(\boldsymbol{e}_k\cdot\boldsymbol{e}_m)(\boldsymbol{e}_l\cdot\boldsymbol{e}_n)\\ &= \alpha_{ij}\tau_{kl}\beta_{mn}\boldsymbol{e}_i\otimes\boldsymbol{e}_j\delta_{km}\delta_{ln}\\ &= \alpha_{ij}\tau_{kl}\beta_{kl}\boldsymbol{e}_i\otimes\boldsymbol{e}_j = (\alpha_{ij}\boldsymbol{e}_i\otimes\boldsymbol{e}_j)\tau_{kl}\beta_{kl}\\ &= \boldsymbol{\alpha}\,(\boldsymbol{\tau}:\boldsymbol{\beta})\,.\end{aligned} \tag{A.9}$$

Second, the property of a symmetric fourth-order tensor $\boldsymbol{S}$ can be expressed as

$$\boldsymbol{\tau}:\boldsymbol{S}:\boldsymbol{\beta} = \boldsymbol{S}:\boldsymbol{\tau}:\boldsymbol{\beta}. \tag{A.10}$$

This formula is useful to separate $\boldsymbol{\beta}$ out from the original term.

Identity Tensor and Projection Tensors Define *fourth order identity* tensor I as

$$\mathsf{I} = \delta_{ik}\delta_{jl}\,\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l, \tag{A.11}$$

symmetric projection tensor

$$\mathsf{I}_\mathsf{s} = \frac{1}{2}\left(\delta_{ik}\delta_{jl}+\delta_{il}\delta_{jk}\right)\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l, \tag{A.12}$$

and *deviatoric projection* tensor

$$\begin{aligned}\mathsf{I}_\mathsf{d} &= \left[\frac{1}{2}\left(\delta_{ik}\delta_{jl}+\delta_{il}\delta_{jk}\right)-\frac{1}{3}\delta_{ij}\delta_{kl}\right]\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l\\ &= \mathsf{I}_\mathsf{s}-\frac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I},\end{aligned} \tag{A.13}$$

where $\boldsymbol{I}$ is a second-order identity tensor. Also, define *transposition* tensor I_t

$$\mathsf{I}_\mathsf{t} = \delta_{il}\delta_{jk}\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l \tag{A.14}$$

then, any isotropic fourth-order tensor $\boldsymbol{Z}$ can be decomposed as a linear combination of the three basic isotropic tensors: I, I_t and $(\boldsymbol{I}\otimes\boldsymbol{I})$:

$$\boldsymbol{Z} = \alpha\mathsf{I}+\beta\mathsf{I}_\mathsf{t}+\gamma(\boldsymbol{I}\otimes\boldsymbol{I}), \tag{A.15}$$

where α,β and γ are some appropriate scalars. For tensor $(\boldsymbol{I}\otimes\boldsymbol{I})$, it can be verified that the action upon a second-order tensor $\boldsymbol{\tau}$ is

$$(\boldsymbol{I}\otimes\boldsymbol{I})\boldsymbol{\tau} = \delta_{ij}\delta_{kl}\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l\tau_{mn}\boldsymbol{e}_m\otimes\boldsymbol{e}_n = \delta_{ij}\tau_{kl}\delta_{kl}\boldsymbol{e}_i\otimes\boldsymbol{e}_j = \mathrm{trace}(\boldsymbol{\tau})\boldsymbol{I}. \tag{A.16}$$

Tensor Trace The trace of a second-order tensor $\boldsymbol{\tau}$ is defined as

$$\mathrm{trace}(\boldsymbol{\tau}) = \boldsymbol{I}\colon\boldsymbol{\tau} \tag{A.17}$$

If tensor $\boldsymbol{\tau}$ is constructed by

$$\boldsymbol{\tau} = \boldsymbol{a} \otimes \boldsymbol{b}$$

and we know for any identity tensor $\boldsymbol{I}$

$$\boldsymbol{I} : (\boldsymbol{a} \otimes \boldsymbol{b}) = \boldsymbol{a} \cdot \boldsymbol{b}, \tag{A.18}$$

we can then generalise the trace computation for a tensor in curvilinear coordinates as

$$\begin{aligned} \text{trace}(\boldsymbol{\tau}) &= \tau_{ij}\vec{g}^i \cdot \vec{g}^j = \tau^{ij}\vec{g}_i \cdot \vec{g}_j = \tau_i^j \vec{g}^i \cdot \vec{g}_j = \tau_j^i \vec{g}_i \cdot \vec{g}^j \\ &= \tau_{ij} g^{ij} = \tau^{ij} g_{ij} = \tau_i^i \end{aligned} \tag{A.19}$$

Gradient, divergence and derivatives. The gradient of a scalar field $\Phi(\boldsymbol{x})$ denoted by $\nabla\phi$ is defined as

$$\nabla\phi = \frac{\partial \phi}{\partial x_i}\boldsymbol{e}_i.$$

The gradient of a vector field $\boldsymbol{u}(\boldsymbol{x})$ is defined as

$$\nabla \boldsymbol{u}(\boldsymbol{x}) = \frac{\partial u_i}{\partial x_j}\boldsymbol{e}_i \otimes \boldsymbol{e}_j.$$

The divergence of $\boldsymbol{u}(\boldsymbol{x})$ is

$$\text{div } \boldsymbol{u} = \frac{\partial u_i}{\partial x_i}$$

and divergence of a second-order tensor $\boldsymbol{\tau}$ defined by

$$\text{div } \boldsymbol{\tau} = \frac{\partial \tau_{ij}}{x_j}\boldsymbol{e}_i$$

is a vector. For a scalar valued function $f(\boldsymbol{u})$ of a vector variable, its derivative with respect to $\boldsymbol{u}$ is a vector:

$$\frac{\partial f(\boldsymbol{u})}{\partial \boldsymbol{u}} = \frac{\partial f(\boldsymbol{u})}{\partial u_i}\boldsymbol{e}_i.$$

In case of a tensor variable, we have

$$\frac{\partial f(\boldsymbol{\tau})}{\partial \boldsymbol{\tau}} = \frac{\partial f(\boldsymbol{\tau})}{\partial \tau_{ij}}\boldsymbol{e}_i \otimes \boldsymbol{e}_j.$$

If $\boldsymbol{f}(\boldsymbol{\tau})$ is a matrix valued function of a second-order tensor, then its derivative with respect to $\boldsymbol{\tau}$ is a fourth-order tensor with components

$$\frac{\partial \boldsymbol{f}(\boldsymbol{\tau})}{\partial \tau_{ij}} = \frac{\partial f_{kl}(\boldsymbol{\tau})}{\partial \tau_{ij}}\boldsymbol{e}_k \otimes \boldsymbol{e}_l.$$

Chain Rule for tensor valued functions of second-order tensors: Let $\boldsymbol{\sigma}(\boldsymbol{\beta})$ be a second order tensor valued function of the second order tensor $\boldsymbol{\beta}$, then the derivative of $\boldsymbol{\sigma}$ with

respect to $\boldsymbol{\beta}$ in the direction τ is a fourth order tensor defined as

$$\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\beta}} : \boldsymbol{\tau} = D\boldsymbol{\sigma}(\boldsymbol{\beta})[\boldsymbol{\tau}] = \left[\frac{\mathrm{d}}{\mathrm{d}\alpha}\boldsymbol{\sigma}(\boldsymbol{\beta} + \alpha\boldsymbol{\tau})\right]_{\alpha=0}.$$

Quite often $\boldsymbol{\sigma}$ can be regarded as

$$\boldsymbol{\sigma}(\boldsymbol{\beta}) = \boldsymbol{\sigma}_1(\boldsymbol{\sigma}_2(\boldsymbol{\beta})),$$

then the chain rule claims

$$\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\beta}} : \boldsymbol{\tau} = \frac{\partial \boldsymbol{\sigma}_1}{\partial \boldsymbol{\sigma}_2} : \left(\frac{\partial \boldsymbol{\sigma}_2}{\partial \boldsymbol{\beta}} : \boldsymbol{\tau}\right) \tag{A.20}$$

The derivative of the unit norm

$$\hat{\boldsymbol{n}} = \frac{\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|}$$

is given by the formula

$$\frac{\mathrm{d}\hat{\boldsymbol{n}}}{\mathrm{d}\boldsymbol{\beta}} = \frac{1}{\|\boldsymbol{\beta}\|}\left(\mathsf{I} - \hat{\boldsymbol{n}} \otimes \hat{\boldsymbol{n}}\right) \tag{A.21}$$

To prove, use the directional derivative definition

$$\begin{aligned}
\frac{\partial \hat{\boldsymbol{n}}}{\partial \boldsymbol{\beta}} : \boldsymbol{\tau} &= \frac{\mathrm{d}}{\mathrm{d}\alpha}\left[\hat{\boldsymbol{n}}(\boldsymbol{\beta} + \alpha\boldsymbol{\tau})\right]\Big|_{\alpha=0} = \frac{\mathrm{d}}{\mathrm{d}\alpha}\left[\frac{\boldsymbol{\beta} + \alpha\boldsymbol{\tau}}{\|\boldsymbol{\beta} + \alpha\boldsymbol{\tau}\|}\right]\Big|_{\alpha=0} \\
&= \left(\frac{\boldsymbol{\tau}}{\|\boldsymbol{\beta} + \alpha\boldsymbol{\tau}\|} - \frac{\boldsymbol{\beta} + \alpha\boldsymbol{\tau}}{\|\boldsymbol{\beta} + \alpha\boldsymbol{\tau}\|^2}\frac{\mathrm{d}}{\mathrm{d}\alpha}\|\boldsymbol{\beta} + \alpha\boldsymbol{\tau}\|\right)_{\alpha=0} \\
&= \frac{\boldsymbol{\tau}}{\|\boldsymbol{\beta}\|} - \frac{1}{\|\boldsymbol{\beta}\|}\hat{\boldsymbol{n}}\left(\hat{\boldsymbol{n}} : \boldsymbol{\tau}\right)
\end{aligned} \tag{A.22}$$

with (A.9), the formula (A.21) can be obtained. If the normal $\hat{\boldsymbol{n}}$ in (A.22) is under action of a fourth-order tensor $\boldsymbol{C}$, then the derivative takes the form

$$\frac{\partial}{\partial \boldsymbol{\beta}}\left(\frac{\boldsymbol{C}\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|}\right) = \frac{\boldsymbol{C} - (\boldsymbol{C}\hat{\boldsymbol{n}}) \otimes \hat{\boldsymbol{n}}}{\|\boldsymbol{\beta}\|}. \tag{A.23}$$

Relation: Given $\boldsymbol{\tau}\left(\boldsymbol{\beta}\boldsymbol{a}\right)$, we want to separate $\boldsymbol{a}$ out. Suppose we have the following lower order tensor construction

$$\boldsymbol{\tau} = \boldsymbol{\tau}_1 \otimes \boldsymbol{\tau}_2, \qquad \boldsymbol{\beta} = \boldsymbol{\beta}_1 \otimes \boldsymbol{\beta}_2,$$

then we can apply (A.9) as

$$\begin{aligned}
\boldsymbol{\tau}\left(\boldsymbol{\beta}\boldsymbol{a}\right) &= \left(\boldsymbol{\tau}_1 \otimes \boldsymbol{\tau}_2\right)\left(\left(\boldsymbol{\beta}_1 \otimes \boldsymbol{\beta}_2\right)\boldsymbol{a}\right) \\
&= \left(\boldsymbol{\tau}_1 \otimes \boldsymbol{\tau}_2\right)\left(\boldsymbol{\beta}_1(\boldsymbol{\beta}_2 : \boldsymbol{a})\right) = \boldsymbol{\tau}_1(\boldsymbol{\tau}_2 : \boldsymbol{\beta}_1)(\boldsymbol{\beta}_2 : \boldsymbol{a}) \\
&= \left(\left(\boldsymbol{\tau}_1(\boldsymbol{\tau}_2 : \boldsymbol{\beta}_1)\right) \otimes \boldsymbol{\beta}_2\right)\boldsymbol{a} \\
&= \left(\left(\boldsymbol{\tau}\boldsymbol{\beta}_1\right) \otimes \boldsymbol{\beta}_2\right)\boldsymbol{a}
\end{aligned} \tag{A.24}$$

Appendix B

Finite Element Formulae

Barycentric Coordinates Consider a non-degenerate triangle with three vertices A, B, C whose positions in Cartesian coordinates are

$$\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c},$$

and any point P, $\boldsymbol{p}$ lies within this triangle, shown in Figure B.1, can be alternatively expressed with the *barycentric coordinates* $(\zeta_1, \zeta_2, \zeta_3)$ as the the follows

$$\begin{cases} \zeta_1 \boldsymbol{a} + \zeta_2 \boldsymbol{b} + \zeta_3 \boldsymbol{c} = \boldsymbol{p} \\ \zeta_1 + \zeta_2 + \zeta_3 = 1 \end{cases} . \tag{B.1}$$

The barycentric coordinates is defined as the area ratios:

$$\zeta_1 = \frac{\Delta PBC}{\Delta ABC}, \quad \zeta_2 = \frac{\Delta PCA}{\Delta ABC}, \quad \zeta_3 = \frac{\Delta PAB}{\Delta ABC}. \tag{B.2}$$

For a triangle in 3D with arbitrary orientation, we introduce the normal to the triangle (A, B, C) in counterclockwise order as

$$\boldsymbol{n} = (\boldsymbol{b} - \boldsymbol{a}) \times (\boldsymbol{c} - \boldsymbol{a}),$$

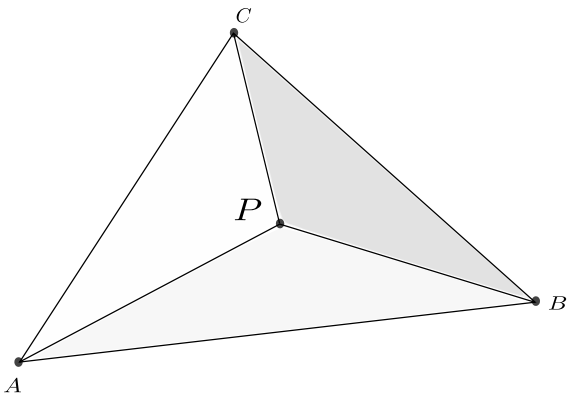


Figure B.1 Barycentric coordinates for triangles

and similarly we may define three normals to the sub-triangles (P, B, C), (P, C, A) and (P, A, B) as

$$\begin{aligned}\boldsymbol{n}_a &= (\boldsymbol{c} - \boldsymbol{b}) \times (\boldsymbol{p} - \boldsymbol{b}) \\ \boldsymbol{n}_b &= (\boldsymbol{a} - \boldsymbol{c}) \times (\boldsymbol{p} - \boldsymbol{c}) \,. \\ \boldsymbol{n}_c &= (\boldsymbol{b} - \boldsymbol{a}) \times (\boldsymbol{p} - \boldsymbol{a})\end{aligned}$$

We can verify that

$$\frac{\boldsymbol{n} \cdot \boldsymbol{n}_a}{\|\boldsymbol{n}\| \, \|\boldsymbol{n}_a\|} = 1, \frac{\boldsymbol{n} \cdot \boldsymbol{n}_b}{\|\boldsymbol{n}\| \, \|\boldsymbol{n}_b\|} = 1, \text{ and } \frac{\boldsymbol{n} \cdot \boldsymbol{n}_c}{\|\boldsymbol{n}\| \, \|\boldsymbol{n}_c\|} = 1,$$

if $\boldsymbol{p}$ is inside the triangle. As the area of each triangle is simply the norm of the normal, i.e.

$$\Delta ABC = \|\boldsymbol{n}\| \,, \Delta PBC = \|\boldsymbol{n}_a\| \,, \Delta PCA = \|\boldsymbol{n}_b\| \,, \Delta PAB = \|\boldsymbol{n}_c\| \,,$$

so we can derive the barycentric coordinates as follows

$$\zeta_1 = \frac{\|\boldsymbol{n}_a\|}{\|\boldsymbol{n}\|} = \frac{\|\boldsymbol{n}_a\|}{\|\boldsymbol{n}\|} \frac{\boldsymbol{n} \cdot \boldsymbol{n}_a}{\|\boldsymbol{n}\| \, \|\boldsymbol{n}_a\|} = \frac{\boldsymbol{n} \cdot \boldsymbol{n}_a}{\|\boldsymbol{n}\|^2}, \tag{B.3}$$

and similarly

$$\zeta_2 = \frac{\boldsymbol{n} \cdot \boldsymbol{n}_b}{\|\boldsymbol{n}\|^2}, \qquad \zeta_3 = \frac{\boldsymbol{n} \cdot \boldsymbol{n}_c}{\|\boldsymbol{n}\|^2}. \tag{B.4}$$

Next, we may work out the first order derivative transformation between the barycentric coordinates and the Cartesian coordinates. First we write barycentric coordinates as $\zeta_i(\boldsymbol{p})$, so the calculation of

$$\frac{\mathrm{d}\zeta_i}{\mathrm{d}\boldsymbol{p}} = \frac{1}{\|\boldsymbol{n}\|^2} \left(\frac{\mathrm{d}\boldsymbol{n}_i}{\mathrm{d}\boldsymbol{p}} \right)^T \boldsymbol{n} \tag{B.5}$$

boils down to the calculation of $\dfrac{\mathrm{d}\boldsymbol{n}_i}{\mathrm{d}\boldsymbol{p}}$, where $\boldsymbol{n}_i$ denotes $\boldsymbol{n}_a$, $\boldsymbol{n}_b$ and $\boldsymbol{n}_c$. Take $\boldsymbol{n}_a$ as an example, we have

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}x_1} \left[(\boldsymbol{c} - \boldsymbol{b}) \times (\boldsymbol{p} - \boldsymbol{b}) \right] &= (\boldsymbol{c} - \boldsymbol{b}) \times \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T = \begin{bmatrix} 0 & c_3 - b_3 & -(c_2 - b_2) \end{bmatrix}^T \\ \frac{\mathrm{d}}{\mathrm{d}x_2} \left[(\boldsymbol{c} - \boldsymbol{b}) \times (\boldsymbol{p} - \boldsymbol{b}) \right] &= (\boldsymbol{c} - \boldsymbol{b}) \times \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T = \begin{bmatrix} -(c_3 - b_3) & 0 & c_1 - b_1 \end{bmatrix}^T. \\ \frac{\mathrm{d}}{\mathrm{d}x_3} \left[(\boldsymbol{c} - \boldsymbol{b}) \times (\boldsymbol{p} - \boldsymbol{b}) \right] &= (\boldsymbol{c} - \boldsymbol{b}) \times \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T = \begin{bmatrix} c_2 - b_2 & -(c_1 - b_1) & 0 \end{bmatrix}^T\end{aligned}$$

So

$$\frac{\mathrm{d}\boldsymbol{n}_a}{\mathrm{d}\boldsymbol{p}} = \begin{bmatrix} 0 & -(c_3 - b_3) & c_2 - b_2 \\ c_3 - b_3 & 0 & -(c_1 - b_1) \\ -(c_2 - b_2) & c_1 - b_1 & 0 \end{bmatrix}, \tag{B.6}$$

and similarly,

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{n}_b}{\mathrm{d}\boldsymbol{p}} &= \begin{bmatrix} 0 & -(a_3-c_3) & a_2-c_2 \\ a_3-c_3 & 0 & -(a_1-c_1) \\ -(a_2-c_2) & a_1-c_1 & 0 \end{bmatrix}. \\ \frac{\mathrm{d}\boldsymbol{n}_c}{\mathrm{d}\boldsymbol{p}} &= \begin{bmatrix} 0 & -(b_3-a_3) & b_2-a_2 \\ b_3-a_3 & 0 & -(b_1-a_1) \\ -(b_2-a_2) & b_1-a_1 & 0 \end{bmatrix}\end{aligned} \tag{B.7}$$

On the other hand, if the third coordinate ζ_3 in (B.1) got replaced with $1-\zeta_1-\zeta_2$, we would have the form

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_1-c_1 & b_1-c_1 \\ a_2-c_2 & b_2-c_2 \\ a_3-c_3 & b_3-c_3 \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, \tag{B.8}$$

where $(x_1, x_2, x_3) = \boldsymbol{p}$. Define

$$\begin{bmatrix} a_1-c_1 & b_1-c_1 \\ a_2-c_2 & b_2-c_2 \\ a_3-c_3 & b_3-c_3 \end{bmatrix} = \boldsymbol{J},$$

whose components (J_{ij}) are the first order derivative $\dfrac{\partial x_i}{\partial \zeta_j}$. Quite often the more familiar operator transformation

$$\begin{bmatrix} \dfrac{\partial}{\partial \zeta_1} \\ \dfrac{\partial}{\partial \zeta_2} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial x_1}{\partial \zeta_1} & \dfrac{\partial x_2}{\partial \zeta_1} & \dfrac{\partial x_3}{\partial \zeta_1} \\ \dfrac{\partial x_1}{\partial \zeta_2} & \dfrac{\partial x_2}{\partial \zeta_2} & \dfrac{\partial x_3}{\partial \zeta_2} \end{bmatrix} \begin{bmatrix} \dfrac{\partial}{\partial x_1} \\ \dfrac{\partial}{\partial x_2} \\ \dfrac{\partial}{\partial x_3} \end{bmatrix} \tag{B.9}$$

and its counterpart

$$\begin{bmatrix} \dfrac{\partial}{\partial x_1} \\ \dfrac{\partial}{\partial x_2} \\ \dfrac{\partial}{\partial x_3} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \zeta_1}{\partial x_1} & \dfrac{\partial \zeta_2}{\partial x_1} \\ \dfrac{\partial \zeta_1}{\partial x_2} & \dfrac{\partial \zeta_2}{\partial x_2} \\ \dfrac{\partial \zeta_1}{\partial x_3} & \dfrac{\partial \zeta_2}{\partial x_3} \end{bmatrix} \begin{bmatrix} \dfrac{\partial}{\partial \zeta_1} \\ \\ \dfrac{\partial}{\partial \zeta_2} \end{bmatrix} \tag{B.10}$$

are needed. As we have shown in (B.5), $\dfrac{\partial \zeta_i}{\partial x_j}$ can be obtained from (B.6) and (B.7). Alternatively, if we recognise the transformation matrix in (B.9) is $\boldsymbol{J}^T$ we can work out the transform matrix in (B.10) by its pseudoinverse

$$\boldsymbol{J}\left(\boldsymbol{J}^T\boldsymbol{J}\right)^{-1},$$

which gives the same values as (B.6) and (B.7).

Basis Function of the Ganev Triangle (reproduced from p.104, [97])

$$
\begin{bmatrix} p_1^0 \\ p_2^0 \\ p_3^0 \\ p_{1,3}^1 \\ p_{1,2}^1 \\ p_{2,3}^1 \\ p_{2,3}^1 \\ p_{3,2}^1 \\ p_{3,1}^1 \\ p_4^0 \\ p_5^0 \\ p_6^0 \\ p_4^\perp \\ p_5^\perp \\ p_6^\perp \end{bmatrix} =
\left[\begin{array}{ccc|cccccc|ccc|ccc}
1 & 0 & 0 & 4 & 4 & 0 & 0 & 0 & 0 & 0 & -5 & -5 & $3(\eta_2-\eta_3)-4$ & $-3(\eta_3+\eta_2)$ & $3(\eta_3+\eta_2)$ \\
0 & 1 & 0 & 0 & 0 & 4 & 4 & 0 & 0 & -5 & 0 & -5 & $3(\eta_1+\eta_3)$ & $3(\eta_3-\eta_1)-4$ & $-3(\eta_1+\eta_3)$ \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 4 & 4 & -5 & -5 & 0 & $-3(\eta_2+\eta_1)$ & $3(\eta_2+\eta_1)$ & $3(\eta_1-\eta_2)-4$ \\
\hline
 & & & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & $-\frac{1}{2}(1-\eta_2)$ & $-\frac{1}{2}(1+\eta_2)$ & $\frac{1}{2}(1+\eta_2)$ \\
 & & & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & $-\frac{1}{2}(1+\eta_3)$ & $\frac{1}{2}(1-\eta_3)$ & $-\frac{1}{2}(1-\eta_3)$ \\
 & & & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & $\frac{1}{2}(1+\eta_3)$ & $-\frac{1}{2}(1-\eta_3)$ & $-\frac{1}{2}(1+\eta_3)$ \\
\hline
 & & & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & $-\frac{1}{2}(1-\eta_1)$ & $-\frac{1}{2}(1+\eta_1)$ & $\frac{1}{2}(1-\eta_1)$ \\
 & & & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & $-\frac{1}{2}(1+\eta_1)$ & $\frac{1}{2}(1+\eta_1)$ & $-\frac{1}{2}(1-\eta_1)$ \\
 & & & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & $\frac{1}{2}(1+\eta_2)$ & $-\frac{1}{2}(1-\eta_2)$ & $-\frac{1}{2}(1+\eta_2)$ \\
\hline
 & & & & & & & & & 16 & 0 & 0 & -16 & 16 & 16 \\
 & & & & & & & & & 0 & 16 & 0 & 16 & -16 & 16 \\
 & & & & & 0 & & & & 0 & 0 & 16 & 16 & 16 & -16 \\
\hline
 & & & & & & & & & & & & -4 & 4 & 4 \\
 & & & & & & & & & & & & 4 & -4 & 4 \\
 & & & & & & & & & & & & 4 & 4 & -4
\end{array}\right]
\begin{bmatrix} \zeta_1^4 \\ \zeta_2^4 \\ \zeta_3^4 \\ \zeta_1^3\zeta_3 \\ \zeta_1^3\zeta_2 \\ \zeta_2^3\zeta_1 \\ \zeta_2^3\zeta_3 \\ \zeta_3^3\zeta_2 \\ \zeta_3^3\zeta_1 \\ \zeta_2^2\zeta_3^2 \\ \zeta_3^2\zeta_1^2 \\ \zeta_1^2\zeta_2^2 \\ \zeta_1^2\zeta_2\zeta_3 \\ \zeta_1\zeta_2^2\zeta_3 \\ \zeta_1\zeta_2\zeta_3^2 \end{bmatrix}
\tag{B.11}
$$

Coordinates Transformation The basic relation for a local-global coordinates transformation illustrated in Figure B.2 can be expressed as

$$(x, y, z) = \bar{x}\bar{\boldsymbol{i}} + \bar{y}\bar{\boldsymbol{j}} + \bar{z}\bar{\boldsymbol{k}} + \boldsymbol{t}, \tag{B.12}$$

where the local coordinate directions $\bar{\boldsymbol{i}}, \bar{\boldsymbol{j}}, \bar{\boldsymbol{k}}$ can be related to the global coordinate directions $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ as

$$\begin{aligned}
\bar{\boldsymbol{i}} &= (\bar{\boldsymbol{i}} \cdot \boldsymbol{i})\boldsymbol{i} + (\bar{\boldsymbol{i}} \cdot \boldsymbol{j})\boldsymbol{j} + (\bar{\boldsymbol{i}} \cdot \boldsymbol{k})\boldsymbol{k} \\
\bar{\boldsymbol{j}} &= (\bar{\boldsymbol{j}} \cdot \boldsymbol{i})\boldsymbol{i} + (\bar{\boldsymbol{j}} \cdot \boldsymbol{j})\boldsymbol{j} + (\bar{\boldsymbol{j}} \cdot \boldsymbol{k})\boldsymbol{k} \,. \\
\bar{\boldsymbol{k}} &= (\bar{\boldsymbol{k}} \cdot \boldsymbol{i})\boldsymbol{i} + (\bar{\boldsymbol{k}} \cdot \boldsymbol{j})\boldsymbol{j} + (\bar{\boldsymbol{k}} \cdot \boldsymbol{k})\boldsymbol{k}
\end{aligned} \tag{B.13}$$

So the transformation becomes

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{i}} \cdot \boldsymbol{i} & \bar{\boldsymbol{j}} \cdot \boldsymbol{i} & \bar{\boldsymbol{k}} \cdot \boldsymbol{i} \\ \bar{\boldsymbol{i}} \cdot \boldsymbol{j} & \bar{\boldsymbol{j}} \cdot \boldsymbol{j} & \bar{\boldsymbol{k}} \cdot \boldsymbol{j} \\ \bar{\boldsymbol{i}} \cdot \boldsymbol{k} & \bar{\boldsymbol{j}} \cdot \boldsymbol{k} & \bar{\boldsymbol{k}} \cdot \boldsymbol{k} \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} + \boldsymbol{t} \tag{B.14}$$

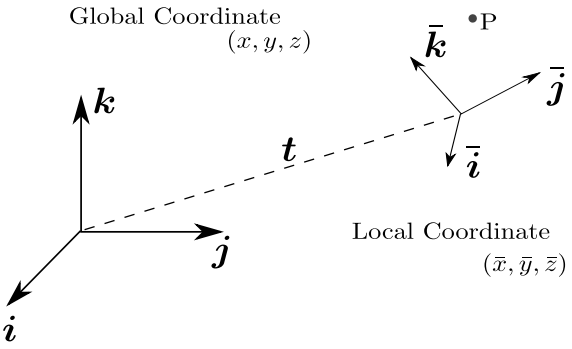


Figure B.2 Local-global coordinates transformation illustration

Permutation Matrix Quite often we have to sort out an appropriate order for the unknown variables to allow a simpler element assembly implementation. For example, we would like to permutate a column vector

$$\boldsymbol{v} = \begin{bmatrix} v_1 & v_2 & v_3 & \cdots & v_n \end{bmatrix}^T$$

into a different one

$$\bar{\boldsymbol{v}} = \begin{bmatrix} v_{\pi(1)} & v_{\pi(2)} & v_{\pi(3)} & \cdots & v_{\pi(n)} \end{bmatrix}^T,$$

where $\pi(j)$ is the new index, and the corresponding permutation matrix is an orthogonal one:

$$P_\pi = \begin{bmatrix} \boldsymbol{e}_{\pi(1)} \\ \boldsymbol{e}_{\pi(2)} \\ \vdots \\ \boldsymbol{e}_{\pi(n)} \end{bmatrix}, \tag{B.15}$$

where $\boldsymbol{e}_{\pi(j)}$ is the unit row vector, with only $\pi(j)^{\text{th}}$ component being 1. Therefore we have

$$\bar{\boldsymbol{v}} = P_\pi \boldsymbol{v}.$$

In MATLAB, P_π can be easily implemented by manipulating the identity matrix. Code:

```
I = eye(n);
P = I(piv,:);
```

where `piv` is the new index vectors as defined by $\pi(j)$.

Bibliography

[1] UK government, "Manufacturing in the UK: An economic analysis of the sector," Department for Business Innovation & Skill, Tech. Rep., December 2010.

[2] J. Beddoes and M. J. Bibby, *Principles of Metal Manufacturing Processes*. Arnold, 1999.

[3] S. Kalpakjian, S. Schmid, and H. Musa, *Manufacturing Engineering and Technology*. Pearson Prentice Hall, 2006.

[4] M. P. Groover, *Principles of Modern Manufacturing*. John Wiley & Sons, INC, 2011.

[5] http://www.lcmp.eng.cam.ac.uk/welcome/equipment/incremental-sheet-forming.

[6] J. Jeswiet, F. Micari, G. Hirt, A. Bramley, J. Duflou, and J. Allwood, "Asymmetric single point incremental forming of sheet metal," *CIRP Annuals-Manufacturing Technology*, vol. 34 Issue 2, pp. 88–114, 2005.

[7] W. Emmens, G. Sebastiani, and A. van den Boogaard, "The technology of incremental sheet forming: A brief review of the history," *Journal of Materials Processing Technology*, vol. 210, no. 8, pp. 981 – 997, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924013610000555

[8] O. Owodunni, S. Hinduja, and S. Mekid. (2005) Towards rapid sheet metal forming. University of Manchester. [Online]. Available: http://conference.iproms.org/presentation/186

[9] E. Leszak, "Apparatus and processes for incremental dieless forming," US Patent 3 342 051, 1967.

[10] M. Bambach, G. Hirt, and S. Junk, "Modelling and experimental evaluation of the incremental CNC sheet metal forming process," in *VII International Conference on Computational Plasticity, COMPLAS*, 2003.

[11] J.-L. Liow, V. P. Astakhov, R. K. Vinayak Gaitonde, J. P. Davim, and M. B. Silva, *Sustainable manufacturing*, J. P. Davim, Ed. ISTE Ltd and John Wiley & Sons, Inc, 2010.

[12] N. R. Greenwood, *Implementing Flexible Manufacturing Systems.* Macmillan Education LTD, 1988.

[13] J. M. Allwood and J. M. Cullen, *Sustainable Materials with Both Eyes Open:Future Buildings, Vehicles, Products and Equipment - Made Efficiently and Made with Less New Material.* UIT Cambridge, 2011.

[14] J. M. Gere and B. J. Goodno, *Mechanics of Materials.* Cengage Learning, 2009.

[15] W. F. Hosford and R. M. Caddell, *Metal Forming: mechanics and metallurgy.* Cambridge University Press, 2011.

[16] J. Lubliner, *Plasticity Theory.* Pearson Education, Inc., 2006.

[17] H. S. Valberg, *Applied Metal Forming including FEM analysis.* Cambridge University Press, 2010.

[18] E. Viatkina, W. Brekelmans, and M. Geers. Forming limit diagrams for sheet deformation processes. Technische Universiteit Eindhoven. [Online]. Available: http://www.mate.tue.nl/mate/pdfs/959.pdf

[19] M. Pohalk, J. Majak, and R. Kuttner, "Manufacutrability and limitations in incremental sheet forming," *Proc. Estonian Acad. Sci. Eng*, vol. 13, No 2, pp. 129–139, 2007.

[20] L. Filice, L. Fratini, and F. Micari, "Analysis of material formability in incremental forming," *CIRP Annuals-Manufacturing Technology*, vol. 51 Issue 1, pp. 199–202, 2002.

[21] L. Fratini, G. Ambrogio, R. Lorenzo, and F. Micari, "Influence of mechanical properties of the sheet material on formability in single point incremental forming," *CIRP Annals - Manufacturing Technology*, vol. 53 Issue 1, pp. 207–210, 2004.

[22] W. C. Emmens and A. H. van den Booggard, "An overview of stabilizing deformation mechanisms in incremental sheet forming," *Journal of Materials Processing Technology*, vol. 209, pp. 3688–3695, 2009.

[23] K. Jackson, J. Allwood, and M. Landert, "Incremental forming of sandwich panels," *Journal of Materials Processing Technology*, vol. v 204 n 1-3, pp. 290–303, 11 Aug. 2008.

[24] F. Micari, G. Ambrogio, and L. Filice, "Shape and dimensional accuracy in single point incremental forming: State of the art and future trends," *Journal of Materials Processing Technology*, vol. 191, pp. 390–395, 2007.

[25] M. G. Lee, J. H. Kim, K. Chung, S. J.Kim, R. H. Wagoner, and H. Y. Kim, "Analytical springback model for lightweight hexagonal close-packed sheet metal," *International Journal of Plasticity*, vol. 25, pp. 399–419, 2009.

[26] I. A. . Burchitz, "Improvement of springback prediction in sheet metal forming," Ph.D. dissertation, Universiteit Twente, 2008.

[27] N. He and R. H. Wagoner, "Springback simulation in sheet metal forming," in *NUMISHEET*, 1996, pp. 308–315.

[28] M. Tisza, "Numerical modelling and simulation in sheet metal forming," *Journal of Materials Processing Technology*, vol. 151, pp. 58–62, 2004.

[29] A. Makinouchi, "Sheet metal forming simulation in industry," *Journal of Materials*, vol. 60, pp. 19–26, 1996.

[30] S. Dejardin, S. Thibaud, J. Gelin, and G. Michel, "Experimental investigations and numerical analysis for improving knowledge of incremental sheet forming processes for sheet metal parts," *Journal of Materials Processing Technology*, vol. 210, pp. 363–369, 2010.

[31] T. Hama, T. Nagata, C. Teodosiu, A. Makinouchi, and H. Takuda, "Finite-element simulation of springback in sheet metal forming using local interpolation for tool surfaces," *International Journal of Mechanical Sciences*, vol. 50, pp. 175–192, 2008.

[32] M. Rauch, J.-Y. Hascoet, J.-C. Hamann, and Y. Plenel, "Tool path programming optimization for incremental sheet forming applications," *Computer-Aided Design*, vol. 41, pp. 877–885, 2009.

[33] A. Attanasio, E. Ceretti, and C. Giardini, "Optimization of tool path in two points incremental forming," *Journal of Materials Processing Technology*, vol. 177, pp. 409–412, 2006.

[34] T. Kim and D. Yang, "Improvement of formability for the incremental sheet metal forming process," *International Journal of Mechanical Sciences*, vol. 42, pp. 1271–1286, 2000.

[35] J. M. Allwood, D. Braun, and O. Music, "The effect of partially cut-out blanks on geometric accuracy in incremental sheet forming," *Journal of Materials Processing Technology*, vol. 210, pp. 1501–1510, 2010.

[36] R. C. Dorf and R. H. Bishop, *Modern Control System.* Pearson Education, Inc., 2005.

[37] E. McShane, "The calculus of variations from the beginning through optimal control theory," *SIAM Journal on Control and Optimization*, vol. 27, no. 5, pp. 916–939, 1989. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/0327049

[38] J. Macki and A. Stauss, *Introduction to Optimal Control Theory.* Springer-Verlag, 1982.

[39] H. Sussmann and J. Willems, "300 years of optimal control: from the brachystochrone to the maximum principle," *Control Systems, IEEE*, vol. 17, no. 3, pp. 32–44, 1997.

[40] F. L. Lewis, D. L. Vrabie, and V. L. Syrmos, *Optimal Control*. John Wiley & Sons, Inc, 2012.

[41] J. L. Speyer and D. H. Jacobson, *Primer on Optimal Control Theory*. SIAM Philadelphia, 2010.

[42] R. D. R. III, D. G.Wilson, G. R. Eisler, and J. E. Hurtado, *Applied Dynamic Programming for Optimization of Dynamic Systems*. SIAM Philadelphia, 2005.

[43] W. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley, 2011.

[44] J. Richalet, A. Rault, J. Testud, and J. Papon, "Model predictive heuristic control: Applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413 – 428, 1978. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0005109878900018

[45] C. R. Cutler and B. L. Ramaker, "Dynamic matrix control – a computer control algorithm," in *Joint American Control Conference*, 1980.

[46] D. W. Clarke, C. Mothtadi, and P. S. Tuffs, "Generalized predictive control–part i. the basic algorithm," *Automatica*, vol. 23, No. 2, pp. 137–148, 1987.

[47] D. Clarke, *Advances in Model-Based Predictive Control*. Oxford University Press, 1994, ch. Survey Papers, pp. 3–21.

[48] J. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2000.

[49] R. Haber, R. Bars, and U. Schmitz, *Predictive Control in Process Engineering: From the Basics to the Applications*. Wiley-VCH Verlag GmbH & Co. KGaA, 2011.

[50] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, 2000.

[51] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization,Estimation and Control*. Taylor & Francis, 1975.

[52] C. E. García, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice–a survey," *Automatica*, vol. 25, no. 3, pp. 335 – 348, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0005109889900022

[53] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[54] J. Allwood, O. Music, A. Raithathna, and S. R. Duncan, "Closed-loop feedback control of product properties in flexible metal forming processes with mobile tools," *CIRP Annals- Maufacturing Technology*, vol. 58, pp. 289–290, 2009.

[55] E. F. Camacho and C. Bordons, *Model Predictive Control.* Springer, 2007.

[56] P. E. Gill, W. Murrary, and M. H. Wright, *Practical Optimization.* Academic Press, London, UK, 1981.

[57] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *Pattern Analysis and Machine Learning*, vol. 21, No. 5, pp. 476–480, 1999.

[58] I. D. Landau and E. M. Lifshitz, *Theory of Elasticity*, 3rd ed., ser. Course of Theoretical Physics. Elsevier Butterworth-Heinemann, 1986, vol. 7.

[59] T. M. Atanackovic and A. Guran, *Theory of Elasticity for Scientists and Engineers.* Birkhauser, 2000.

[60] F. Dunne and N. Petrinic, *Introduction to Computational Plasticity.* Oxford University Press, 2006.

[61] W. Han and B. D. Reddy, *Plasticity: Mathematical Theory and Numerical Analysis.* Springer, 1999.

[62] E. de Souza Neto, D. Peric, and D. Owen, *Computational Methods For Plasticity: Theory and Applications.* John Wiley & Sons Ltd, 2008.

[63] A. S. Khan and S. Huang, *Continuum Theory of Plasticity.* Jon Wiley & Sons, 1995.

[64] H. Hertz, *Miscellaneous Papers.* Macmillan and Co. Ltd, 1896.

[65] A. Signorini, "Sopra alcune questioni di elastostatica," *Annli della Scuola Normale Superiore di Pisa, Classe di Scienze* 2^e *serie*, vol. 2, pp. 231–251, 1933.

[66] ——, "Questioni di elasticità non linearizzata e semilinearizzata (issues in non linear and semilinear elasticity)," *Rendiconti di Matematica e delle sue applicazioni*, vol. 18, pp. 95–139, 1959.

[67] G. Fichera, "Problemi elastostatici con vincoli unilaterali ii. problema di signorini con ambique condizioni al contorno," *Mem Accas. Naz. Lincei*, vol. VII,Sez I,5, pp. 91–140, 1964.

[68] G. Duvaut and J. L. Lions, *Inequalities in Mechanics and Physics.* Springer, Berlin, 1976.

[69] I. Hlaváček, J. Haslinger, J. Nečas, and J. Lovíšek, *Solution of Variational Inequalities in Mechanics.* Springer-Verlag, 1988.

[70] M. Shillor, M. Sofonea, and J. J. Telega, *Models and Analysis of Quasistatic Contact.* Springer, 2004.

[71] J. T. Oden and J. A. C. Martins, "Models and computational methods for dynamic friction phenomena," *Computer Methods in Applied Mechanics and Engineering*, vol. 52, Issue 1-3, pp. 527–634, 1985.

[72] O. Pantz, "A frictionless contact algorithm for deformable bodies," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 45, pp. 234–254, 2011.

[73] T. A. Laurens and J. C. Simo, "A continuum-based finite element formulations for the implicit solution of multibody, large deformation frictional contact problem," *Internal Journal for Numerical Methods in Engineering*, vol. 36, pp. 3451–3485, 1993.

[74] N. Kikuchi and J. T. Oden, *Contact Problems in Elasticity: A study of Variational Inequalities and Finite Element Methods.* SIAM, 1988.

[75] M. J. Turner, R. W. Clough, H. C. Martin, and L. P. Topp, "Stiffness and deflection analysis of complex structures," *Journal of Aeronautical Society*, vol. 23, p. 805–823, 1956.

[76] O. C. Zienkiewicz, R. L. Taylaor, and J. Z. Zhu, *The Finite Element Method : Its Basis and Fundamentals.* Elsvier Butterworth-Heinemann, 2005.

[77] R. Courant, "Variational methods for the solution of problems of equilibrium and vibrations," *Bulletin of American Mathematics Society*, vol. 49, pp. 1–23, 1943.

[78] G. Strang and G. Fix, *An Analysis of The Finite Element Method.* Prentice Hall, 1973.

[79] J. N. Reddy, *An Introduction to the Finite Element Method*, J. J. Corrigan and J. M. Morriss, Eds. McGraw-Hill, Inc., 1993.

[80] P. Solin, *Partial Differential Equations and the Finite Element Method.* John Wiley & Son, Inc., 2006.

[81] I. Babuska and M. Suri, "The p and $h-p$ versions of the finite element method: basic principles and properties," *SIAM Review*, vol. 36, pp. 578–632, Dec., 1994.

[82] P. G. Giarlet, *The Finite Element Method for Elliptic Problems.* SIAM, 2002.

[83] D. Braess, *Finite Elements: Thoery, Fast Solvers and Applications in Solid Mechanics.* Cambridge University Press, 2001.

[84] E. Zeidler, *Nonlinear Functional Analysis and its Applications III: Variational Methods and Optimization.* Springer-Verlag, 1984.

[85] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

[86] F. Botelho, *TOPICS ON FUNCTIONAL ANALYSIS, CALCULUS OF VARIATIONS AND DUALITY*. Academic Publications, 2011.

[87] M. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, vol. 4, pp. 303–320, 1968.

[88] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996.

[89] K. Ito and K. Kunisch, "The augmented Lagrangian method for equality and inequality constraints in Hilbert spaces," *Mathematical Programming*, vol. 46, pp. 341–360, 1990.

[90] ——, *Lagrange Multiplier Approach to Variational Problems and Applications*. SIAM, Philadelphia, 2008.

[91] F. Auricchio and E. Sacco, "Augmented lagrangian finite-elements for plate contact problems," *International Journal for Numerical Methods in Engineering*, vol. 39, pp. 4141–4158, 1996.

[92] G. Pietrzak and A. Curnier, "Large deformation frictional contact mechanics: Continuum formulation and augmented Lagrangian treatment," *Computer Methods in Applied Mechanics and Engineering*, vol. 177, pp. 351–381, 1999.

[93] K. Ito and K. Kunisch, "Novel concepts for nonsmooth optimization and their impact on science and technology," *Proceedings of the International Congress of Mathematicians*, pp. 3061–3090, 2010.

[94] M. Ulbrich, *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces*. SIAM and Mathematical Optimization Society, Philadelphia, 2011.

[95] L. Qi and D. Sun, "A survey of some nonsmooth equations and smoothing Newton methods," *Progress in Optimization*, vol. 30, pp. 121–146, 1999.

[96] D. Chapelle and K.-J. Bathe, *The Finite Element Analysis of Shells–Fundamentals*. Springer-Verlag, 2003.

[97] M. Bernadou, *Finite Element Methods for Thin Shell Problems*. John Wiley & Sons Ltd, 1996.

[98] P.-S. Lee and K.-J. Bathe, "Insight into finite element shell discretizations by use of the "basic shell mathematical model"," *Computers and Structures*, vol. 83, pp. 69–90, 2003.

[99] C. de Boor, *A Practical Guide to Splines*. Springer, 2001.

[100] C. A. Micchelli, T. J. Rivlin, and S. Winograd, "The optimal recovery of smooth functions," *Numerische Mathematik*, vol. 26, no. 2, pp. 191–200, 1976.

[101] C. de Boor, "B(asic) Spline Basics." [Online]. Available: http://www.cs.unc.edu/ dm/UNC/COMP258/Papers/bsplbasic.pdf

[102] J. H. Argyris, I. Fried, and D. W. Scharpf, "The TUBA family of plate elements for the matrix displacement method," *The Aeronautical Journal of the Royal Aeronautical Society*, vol. 72, pp. 701–709, 1968.

[103] V. I. Krylov, *Approximate Calculation of Integrals*. Dover Publications Inc, 1962.

[104] P. Hammer, O. Marlowe, and A. Stroud, "Numerical integration over simplexes and cones," *Mathematical Tables and Other Aids to Computation*, vol. 10, no. 55, pp. 130–137, 1956.

[105] G. E. Bartholomew, "Numerical integration over the triangle," *Mathematical Tables and Other Aids to Computation*, vol. 13, no. 68, pp. 295–298, 1959. [Online]. Available: http://www.jstor.org/stable/2002802

[106] P. Keast, "Moderate-degree tetrahedral quadrature formulas," *Computer Methods in Applied Mechanics and Engineering*, vol. 55, no. 3, pp. 339 – 348, 1986. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0045782586900599

[107] C. T. Reddy and D. J. Shippy, "Alternative integration formulae for triangular finite elements," *International Journal for Numerical Methods in Engineering*, vol. 17, no. 1, pp. 133–139, 1981. [Online]. Available: http://dx.doi.org/10.1002/nme.1620170111

[108] G. R. Cowper, "Gaussian quadrature formulas for triangles," *International Journal for Numerical Methods in Engineering*, vol. 7, no. 3, pp. 405–408, 1973. [Online]. Available: http://dx.doi.org/10.1002/nme.1620070316

[109] E. J. Kubatko, B. A. Yeager, and A. L. Maggi, "New computationally efficient quadrature formulas for triangular prism elements," *Computers & Fluids*, vol. 73, pp. 187–201, 2013.

[110] K. Krabbenhøft, "A variational principle of elastoplasticity and its application to the modelling of frictional materials," *International Journal of Solids and Structures*, vol. 46, pp. 464–479, 2009.

[111] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, D. Cox, S. G. Krantz, R. Mazzeo, and M. Scharlemann, Eds. American Mathematical Society, 2010.

[112] V. Barbu, *Optimal Control of Variational Inequalities*. Pitman Publishing Limited, 1984.

[113] K. Siegert, E. Dannenmann, S. Wagner, and A. Galaiko, "Closed-loop control system for blank holder forces in deep drawing," *CIRP Annals - Manufacturing Technology*, vol. 44, no. 1, pp. 251 – 254, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0007850607623191

List of Figures

List of Tables