The Pennsylvania State University

The Graduate School

College of Engineering

Incipient Plastic Phenomena in Machining of Copper

A Thesis in

Industrial Engineering

by

Fei Du

© 2013 Fei Du

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science

August 2013

The thesis of Fei Du was reviewed and approved* by the following:

Christopher J. Saldana

Assistant Professor of Industrial and Manufacturing Engineering

Thesis Adviser

Robert C. Voigt

Professor of Industrial and Manufacturing Engineering

Paul Griffin

Professor of Industrial and Manufacturing Engineering

Department Head of Industrial and Manufacturing Engineering

*Signatures are on file in the Graduate School

Acknowledgements

I would like to give my deepest gratitude to my advisor Dr. Christopher Saldana. As an international student, I met more difficulties in the entire research process than the other ones. However, Dr. Saldana was always incredibly patient with me and provided me very important guidance over the past year, no matter on academia or on my life. I cannot successfully accomplish this project without him. I also want to thank my nice labmates – Josh Norman, Cesar Moreno and Miguel Iturralde – who helped a lot in my daily work; thank the technical supports from Dan Supko, Randy Wells and David Shelleman during the experimental designs and operations of this research. In addition, I am grateful to Dr. Robert Voigt, whose review and amendments made the thesis further improved.

I would also like to thank my parents for their continuous supports in my school days, and thank my girlfriend Mengxin Niu whose encouragement and loving care accompanied me all the time. Last but not least, I feel very lucky for making lots of new friends at Penn State who provide me a fun and positive environment to live and study. Thank you all.

Table of Contents

List of Tables

List of Figures

Abstract

Past studies on the topic of material removal have emphasized study of deformation phenomena at steady-state; models and analyses in this regime are well established. In contrast, the plasticity associated with the incipient or transient regime has received far less attention. While transient phenomena are not typically critical in understanding the deformation field of conventional machining, recently-developed manufacturing technologies (e.g., vibration-assisted machining, surface texturing) make use of this regime to improve process/product performance. In this regard, detailed understanding of incipient plastic phenomena is needed to better understand potential for process-level improvements.

In this thesis, the transition from incipient stage to steady-state cutting was investigated in plane strain deformation of a model material system - pure copper. To measure the time-variant deformation in the incipient cutting regime, a numerical method based on a particle image velocimetry (PIV) algorithm was developed and applied to quantitatively measure deformation fields during chip formation. Characterization of the evolution to steady-state was made by PIV, hardness mapping and cutting load analysis, and all were shown to be capable of accurately predicting the length of incipient cutting stage. The incipient plastic phenomena in chip formation were analyzed under different cutting conditions and the effects of cutting parameters – tool rake angle and depth of cut – on the length of incipient cutting stage were investigated. A linear relationship between the incipient regime length and the two controllable parameters was found and was rationalized by considering the effects of each on the length of the shear plane in deformation. These results are potentially useful as a basis for designing parameters in processes that make use of transient phenomena in deformation.

Nomenclature

α	Rake angle, degree
a_0	Depth of cut, μm
a_c	Deformed chip thickness, μm
V_0	Cutting speed, mm/s
V_c	Chip formation speed, mm/s
λ	Chip thickness ratio (a_c/a_0), dimensionless
ϕ	Shear angle, degree
Δ	Width of deformation zone, μm
l	Length of deformation zone, μm
ε	Strain, dimensionless
$\dot{\varepsilon}$	Strain rate, sec^{-1}
$\bar{\varepsilon}$	Effective strain, dimensionless
$\dot{\bar{\varepsilon}}$	Effective strain rate, sec^{-1}
γ/ε_{xy}	Shear strain, dimensionless
$\dot{\gamma}/\dot{\varepsilon_{xy}}$	Shear strain rate, sec^{-1}
ε_{xx}	Normal strain on the horizontal axis, dimensionless
ε_{yy}	Normal strain on the vertical axis, dimensionless
$\dot{\varepsilon_{xx}}$	Normal strain rate on the horizontal axis, sec^{-1}
$\dot{\varepsilon_{yy}}$	Normal strain rate on the vertical axis, sec^{-1}
PIV	Particle Image Velocimetry
FEM	Finite Element Method
HV	Vickers Hardness

Chapter 1: Introduction

The mechanics of material removal in machining-based processes is a major subject of study in the metal cutting community, as these behaviors influence important system parameters that include temperature at the tool-chip interface, cutting load, energy consumption, tool life, and workpiece surface finish. Understanding how controllable cutting parameters alter the underlying mechanics of the cutting process is important for developing a process-level understanding that can facilitate practical improvement of the machining performance and associated cost effectiveness. For example, if one is able to better understand the effect that tool geometry has on cutting performance and micro-mechanics, new tools can be developed that extend the processing envelope in terms of speed and quality.

The classical theory underlying plastic phenomena in metal cutting is well established and often a simplified model is used to relate process variables to deformation parameters. In this regard, the orthogonal cutting was first proposed by Merchant in 1940s [1] is considered to be one of the most important. This model has been applied to many different machining configurations [2] of industrial interest. In orthogonal cutting, the cutting tool generates a plane surface parallel to the original surface of the material being cut and the cutting edge is orthogonal to the direction of travel. The chip is formed by shear in a narrow region extending from the cutting edge to the work surface, called the primary deformation zone or shear plane [1-4]. A secondary deformation zone extends some small distance into the chip along the chip-tool interface, where additional deformation arises as a result of friction [4,5]. The orthogonal cutting model by Merchant has been used to better understand deformation mechanics in terms of cutting parameters, such as cutting speed, depth of cut, cutting tool geometry, friction condition, etc. [6-9].

An overwhelming majority of the past research investigations in metal cutting have placed emphasis on understanding the deformation zone characteristics in what is called the steady cutting state regime. This regime occurs when the chip is

fully-developed, a case that persists more often than not as machining is known to produce long chips especially in ductile metals. In contrast, far little work has been done to better understand the origins of chip formation in the incipient cutting regime, where the plastic zone is not fully developed. The incipient cutting stage starts from the first contact between the cutting tool and the workpiece and ends with the strain in the deformed material reaching a steady level [10]. While the incipient cutting regime may not play a significant role in situations wherein the chip thickness is constant (e.g., continuous cutting), its importance is clear in processes wherein discontinuous chip formation occurs.

For example, in milling, the cutter flutes often enter and leave the workpiece constantly as the tool moves through the work. This results in a time-varying undeformed chip thickness that can be in an incipient state depending on the machining conditions used. In this case, tool wear, chatter and energy consumption all are influenced by the deformation conditions in the incipient stage [11,12]. Another industrial example where incipient chip formation is important is in that of machining of brittle materials, where discontinuous chips form due to the inherent nature of the material. In such cases, transient chip formation directly impacts surface integrity [13]. More recently, new machining technologies centered on the use of vibration have been developed that affect transient chip formation to improve the speed and capability of machining processes. An example of such a method is vibration-assisted drilling, which utilizes fast tool motions to impose a time-varying chip thickness that effectively causes discrete chip formation. The disruption of the tool-chip contact enables chip evacuation, increases drill life and reduces size errors [14]. This is indeed similar to the reason why high pressure coolant is used in deep hole drilling applications [15]. Vibration-assisted machining has also been applied to create component surfaces with novel attributes (texture, wear resistance) that enhance the product life cycle, through discretizing chip formation to small dimensions of controlled size and shape [16]. In these methods, higher vibration frequencies lead to smaller lengths of resulting chip geometries, and in the limit where the oscillation of the tool

occurs on the length scale of the depth of cut, the overall plastic flow mechanisms in machining may be modified [17]. An understanding of how transient plastic phenomena occur for these many industrial machining configurations can positively influence the process and tool design for these technologies.

It is unsurprising that the deformations occurring in the incipient plastic regime have received somewhat less attention. This regime is more complicated a scenario to study than that of the steady state, as the underlying deformation parameters are no longer time-invariant, including the shape of the deformation zone, local strains/stresses and overall forces [18]. However, there have been some topics associated with this domain that have attracted attention. Determining the boundary between the two cutting states has been studied. Several studies utilized finite element methods (FEM) to visualize the deformation fields [5,19]; others used measurements of cutting load as the deformation indicator to evaluate the change in cutting state [10,11]. Using FEM, the initial chip structure was observed of different cutting materials including aluminum, brass and copper [20], and the variation of deformation distribution was measured under different tool rake angles and cutting speeds [19,21]. These previous studies revealed common phenomena in non-steady cutting processes, however, most of them failed to provide satisfactory identification of the incipient cutting stage due to the limited accuracy of analytical methods and the absence of proper measurement criteria. Moreover, the relationships between the important cutting parameters and the duration of incipient cutting stage were never formally investigated. Bridging of this knowledge gap is the focus of the present study.

1.1 Problem Statement

The transition to steady-state from the incipient stages of chip formation in material removal is less-than-adequately understood. Furthermore, a process-level understanding of this phenomenon has remained unaddressed by the community. The objective of the present research is to characterize the deformation in incipient stages of cutting using a range of experimental methods, including particle image

velocimetry (PIV), hardness mapping and load measurements, to characterize the length of incipient cutting stage as a function of controllable machining variables. Controllable machining variables of tool geometry (rake angle) and depth of cut were adjusted to elucidate the effects on the transition to steady-state cutting. The understanding that results from this research is useful for providing a process-level framework for describing the transition from incipient to steady-state chip formation.

Chapter 2: Background

Deformation behavior during the incipient stages of chip formation is more complicated than that occurring in the steady-state regime. In this chapter, the mechanics of chip formation in orthogonal cutting is first presented. Next, several experimental methods used to obtain deformation field measurements during the chip formation will be reviewed and the pros and cons of each method will be discussed. Furthermore, the state-of-the-art in understanding and characterizing the incipient cutting stage will be explored. The experimental methods used to study the incipient chip formation process will be grounded based on these earlier works to bridge the gap in understanding the role of process parameters on incipient cutting phenomena.

2.1 Plane-Strain Orthogonal Machining

Orthogonal cutting models are widely applied for studying deformation behaviors in chip formation, which can be depicted as Fig. 2.1 [22]. In the model, principal machining parameters affecting response include tool rake angle (α), tool relief angle (β), cutting speed (V_0) and undeformed chip thickness/depth of cut (a_0). During orthogonal cutting, chip formation occurs along the rake face and chip flow proceeds at a speed (V_c) and deformed chip thickness (a_c) that is dependent on the material of interest. The process itself is characterized by concentrated shear that occurs at a shear plane that extends from the cutting edge to the free surface of the workpiece at a shear angle (ϕ). Fig. 2.2 [22] shows a schematic used to explain how shear strain can be determined in this model. Shear strain (γ) is the ratio of the transverse displacement (Δs) of card-like elements in the material to their thickness (Δy). It can be estimated according to the following relationship for plane-strain orthogonal cutting [1,4,22,23].

$$\gamma = \frac{\Delta s}{\Delta y} = \frac{AB'}{CD} = \frac{AD}{CD} + \frac{DB'}{CD} = \tan(\phi - \alpha) + \cot(\phi) = \frac{\cos(\alpha)}{\sin(\phi)\cos(\phi - \alpha)} \qquad \text{(Eq. 2.1)}$$

where α is the rake angle and ϕ is the shear angle. The shear angle itself is not a controllable cutting parameter but is a result that is largely dependent on the material and the cutting configuration. The shear angle can be determined by relating the deformed chip thickness (a_c) and the undeformed chip thickness (a_0) to the length of shear plane as in the following equation:

$$\frac{a_c}{\sin(\phi)} = \frac{a_0}{\cos(\phi - \alpha)}$$

$$\tan(\phi) = \frac{\frac{a_0}{a_c}\cos(\alpha)}{1 - \frac{a_0}{a_c}\sin(\alpha)} \quad \text{(Eq. 2.2)}$$

Another important parameter to consider during the shear process is the rate at which the deformation occurs, e.g. the strain rate. The shear strain rate is determined as the ratio of the shear strain to the time taken for the material to pass through the card-like element. It can be derived from Eq. 2.1 [24,25] and is given by:

$$\dot{\gamma} = \frac{\gamma}{\frac{\Delta y}{V_0 \sin(\phi)}} = \frac{\cos(\alpha)}{\cos(\phi - \alpha)} \frac{V_0}{\Delta y} \quad \text{(Eq. 2.3)}$$

where, V_0 is the cutting speed and Δy is the width of shear plane. Effective strain is a monotonically increasing scalar value that is useful in comparing the deformation that occurs in machining to that of other machining configurations. Ignoring the much smaller normal strains, the approximate relation between effective strain and shear strain is $\bar{\varepsilon} = 1/\sqrt{3}\gamma$ ($\dot{\bar{\varepsilon}} = 1/\sqrt{3}\dot{\gamma}$) [24]. Substituting Eq. 2.2 for ϕ in Eq. 2.1, effective strain can be estimated as [25-27]:

$$\bar{\varepsilon} = \frac{1}{\sqrt{3}}\left(\frac{\lambda}{\cos(\alpha)} + \frac{1}{\lambda\cos(\alpha)} - 2\tan(\alpha)\right) \quad \text{(Eq. 2.4)}$$

where, λ is the chip thickness ratio (a_c/a_0) or velocity ratio (V_0/V_c). Based on Eq. 2.3 and empirical analyses on the geometry of shear plane, effective strain rate can be approximated as [26]:

$$\dot{\varepsilon} = \frac{5V_0\cos(\alpha)}{\sqrt{3}a_c} \quad \text{(Eq. 2.5)}$$

Understanding strain and strain rate in material removal processes is highly important as these same variables are well known to affect the micro-mechanisms of plastic flow in a wide range of deformation configurations. In the context of machining, the effects of strain and strain rate on material response play a role in determining thermo-mechanical loading characteristics on tooling. This includes cutting forces and thermal energy dissipation, both of which are known to affect tooling performance. The effect of strain on the cutting process is evident in Refs. [3,4,24] where significant rises in forces/specific energies are seen with increasing levels of plasticity. Strain also affects microstructure development in the chip and machined surface, due to the enhanced dislocation activity at higher levels of strain [4]. Strain rate also has a clear effect on the cutting process, as higher strain rates can lead to either rate-dependent hardening or strain rate softening [4,9,28], depending on the regime of interest. Furthermore, strain rate can modify the types of microstructures developed in the deformation zone. For example, both highly twinned and dynamically recrystallized microstructures have been observed at higher strain rates in machined chips and surfaces [22,26]. Temperature is another crucial factor affecting the strain field and flow stress through changing the friction condition and the microstructure of the material, whose distribution and effects have also been fully studied [9,24,25,29].

2.2 Deformation Field Characterization

Insight into the deformation field in machining can come from numerical simulations, in situ characterization and post-mortem characterization methods. In each of these methods, knowledge of the underlying deformation parameters (e.g., strain, strain rate and temperature) can come from computation, direct observation/measurement or subsequent measurement of effects on material characteristics, respectively. In this regard, each of these methods has associated advantages and disadvantages that we consider below.

Finite Element Method (FEM) based investigations have been popular use for simulation of deformation fields in chip formation since the 1980s due to the rapid development of computing technologies and software [30]. The primary motivation behind these techniques is that they can be used to cost-effectively simulate process results for a wide range of parameters including tool geometry, materials and cutting conditions [31]. FEM can be used to quantitatively obtain stress, strain and temperature fields by dividing the material into small elements and assembling them to reach a system of equations whose solutions yield the behavior of the total system [31,32]. Deformation fields obtained using FEM of continuous and discontinuous chip formation were presented in Refs. [31-34] and [13,32], respectively. Representative fields are shown in Fig. 2.3 [33]. The simulation results of several commonly used FE models were compared with each other under different cutting parameter settings in Ref. [35], and their application limits were analyzed. While FEM can be effectively used to generate stress/strain/temperature fields for a number of configurations, the FE method does suffer from a number of limitations and caution must be exercised in its application. Disadvantages of FEM include the considerable computational resources needed to conduct effective simulations and the effects of modeling assumptions on the overall results [30,32,34,35].

In situ characterization of the deformation field in machining is possible in the form of two general methods: grid-based deformation measurements and image correlation methods. In grid-based methods [36,37], an artificial grid is applied to the surface of the workpiece and the effect of the deformation on the grid shape is directly observed; parameters such as strain and strain rate are derived using classical plasticity constructs. Grid-based methods are useful in validating FEM simulations of the deformation in various materials. However, an inherent difficulty associated with these methods is that the spatial resolution attainable is limited. A second method for in situ characterization of deformation fields involves image correlation-based methods, e.g. digital image correlation (or particle image velocimetry).

Particle Image Velocimetry (PIV) is a computer-based image processing technique for measuring the displacement field in a fluid (or solid) flow. The displacement of an area of an imaged field can be obtained by using statistical image correlation between two successive images taken a specified time interval apart from one another [38]. An early application of this method for measuring deformation parameters (e.g., strain rate, strain) in plane strain machining was described by Lee et al. in 2006 [38,39]. Subsequent application of the method to machining operations proved to be accurate for various material systems such as lead, copper, aluminum, brass and titanium [38-41]. Further developments made in this method have extended the spatial resolution for measuring displacement fields up to a level of ~1/10th the pixel size under appropriate particle arrangement, which is an order of magnitude better than that of the earlier PIV measurements [42]. More recently, PIV has been widely applied to investigate the deformation behavior of various metal cutting configurations [25,26,42]. An example effective strain rate field for machining configurations as obtained using PIV is displayed in Fig. 2.4 [39]. A clear advantage of using PIV to characterize the deformation field is that it provides a direct approach for obtaining full-field displacement measurements at sub-pixel accuracy [40,42]. However, there are still challenges associated with the use of PIV methods that include, but are not limited to, the accuracy of results given experimental conditions (e.g., frame rate, image quality, particle distribution) [42] and the difficulty to apply to more complex scenarios, including discontinuous chip formation.

Post-mortem characterization of the deformation field can come from evaluation of local measurements of strength (e.g., hardness) or from evaluation of strain-dependent microstructure transformations. With regard to local measurements of strength, indentation hardness testing is an experimental method used to obtain the hardness of cold-worked parts. The indentation itself can be made over a size regime spanning over a range of length scales from 10^{-3} m to 10^{-9} m [43]. For strain-hardening materials, deformation induces a rise in material strength due to the associated dislocation activity. In this regard, hardness is a practical way of directly

characterizing strength and indirectly visualizing the strain distribution in a field-type measurement [44]. To correlate such hardness measurements to strain values, controlled experiments are required wherein known values of strain are applied and hardness of these specimens measured. Many prior investigations have made effort to establish an analytical relation between hardness and strain; these effects can be generally described as [44-47]:

$$HV = C_1 \bar{\varepsilon}^n + C_2 \qquad \text{(Eq. 2.6)}$$

where HV is the Vickers hardness, $\bar{\varepsilon}$ is the effective strain, C_1, C_2 and n are constants. The constants in Eq. 2.6 have been reported for several common metals, and errors associated with using such predicted hardness values have been evaluated in Ref. [45]. With regard to metal cutting, micro-hardness tests were used by Weinmann to investigate incipient deformation in metal cutting of 70/30 brass [18,43,48]. The subsequent hardness field in a chip obtained using micro-indentation is shown in Fig. 2.5 [18]. The hardness mapping approach is advantageous in that the obtained deformation fields are more accurate than that given by theoretical models [43], and also the method is easily-implemented. Further, as hardening in some materials is extremely non-linear at the low end of strain, the micro-hardness mapping approach may be more sensitive than even direct measurement methods. The disadvantages of using this approach includes incompatibility with non-strain hardening materials and also the fact that other important information, e.g. strain rate, cannot be obtained directly using this method.

Another post-mortem technique for visualizing deformation fields includes monitoring strain-dependent microstructure transformations. A good example is the effect that deformation has on the microstructure of the chip and machined surface. In this regard, the high strains imposed by machining are well known to cause significant grain refinement in chips and also cause white layer formation in the machined surface [4,6,49]. Another example of strain dependent microstructure effects that can reveal the deformation field is related to recrystallization phenomena. The

recrystallized grain size is strongly strain dependent and can be used to visualize deformation patterns. This method was also employed in Weinmann's investigations, but was shown to be less sensitive than hardness testing in highly deformed regions [18,43]. As with the hardness mapping method, a range of materials that do not show strain-dependent microstructure response may be incompatible with this method.

2.3 Incipient Chip Formation

When chip formation in machining occurs such that the chips that are produced are continuous and of constant chip thickness, it is considered to be at steady-state [4]. In this case, the deformation field in machining is considered time-invariant and is controlled by the relationships given by Eqs. 2.1-2.5. Consequently, the loading on the tooling vis-à-vis forces also is considered to be time-invariant [4,10,50]. In contrast, at the start of chip formation, the plastic zone must fully develop before chip thickness is constant. In this regime, the strain levels and force levels increase to plateau values until the chip thickens to its steady state limit.

Based on the statements above, a number of methods were proposed to distinguish the two regimes of deformation. Among them, the strain distribution in the chip is the most important criterion for evaluating the evolution of the chip to steady state from an incipient regime as strain directly reveals the deformation field. This analytical approach was used in Refs. [19,33,50] with the help of FEM. When the tools for characterizing deformation fields are not available, cutting force can be measured and the steady state can be identified from the force plateau [10,11]. Apart from these two common methods, Mhalsekar attempted to utilize the tool vibration signals captured during face cutting combining with a recurrence quantification analysis to differentiate the cutting states, which was also shown to be effective [12].

Although most of the related works were not concerned with the specific length of incipient cutting stage, the criteria introduced above can be applied to qualitatively analyze the length variation of incipient cutting stage based on the given deformation parameters. The effects of rake angle on initial cutting load, shear stress and shear

angle were studied in Refs. [18,21,43]. The equivalent stress fields presented in Ref. [21] showed that the regime in which the deformation distribution was non-uniform expanded with the decrease of rake angle. Ref. [19] exhibited the incipient cutting force plots for different cutting speeds. The duration of incipient deformation was almost unchanged, but the forces lowered with the increasing cutting speed. The cutting tests were performed on two different metals – lead and carbon steel under the same cutting conditions in Ref. [10]. The cutting force analyses indicated a rough trend that the incipient regime is generally longer when cutting softer materials. Moreover, the conclusion that the length of incipient cutting stage extends with the increasing chip thickness ratio (λ) was suggested by Ref. [5] based on the FEM results. In many of the references, the visual deformation fields were provided, which revealed the general contour of the distribution of deformation parameters in the incipient cutting stage. Fig. 2.6 [5] is a typical strain field in incipient regime of chip formation generated by a FE model. This can be a valuable tool for validating the results obtained in the present study.

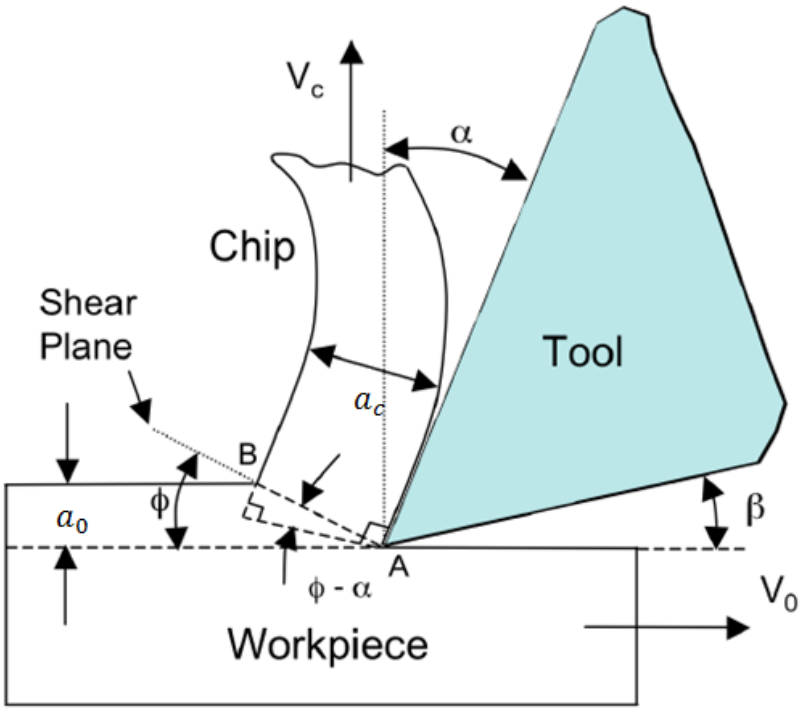


Figure 2.1: Schematic of orthogonal cutting model [22]

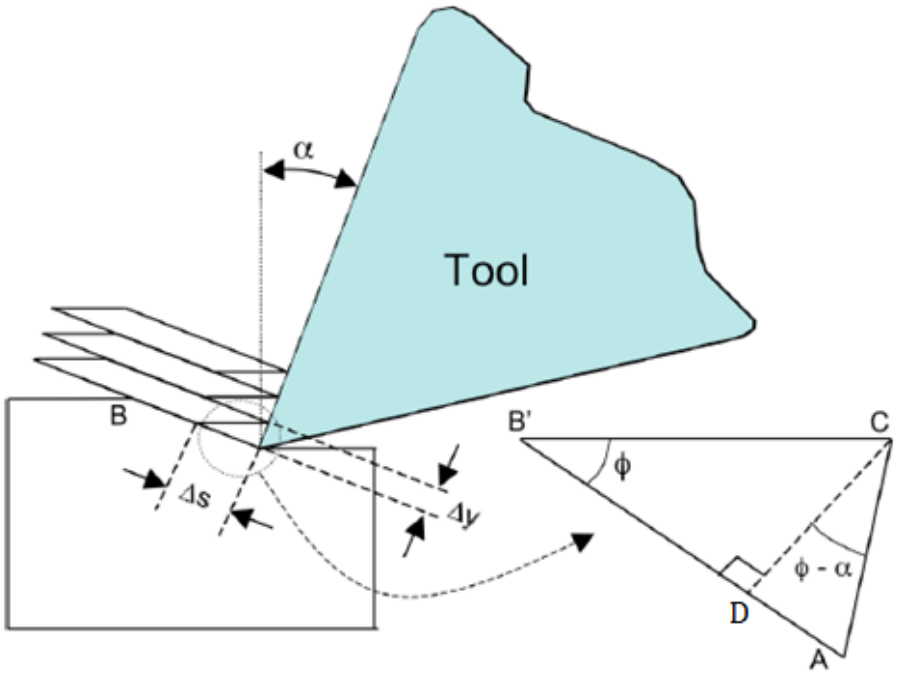


Figure 2.2: Shear strain determination in orthogonal cutting [22]

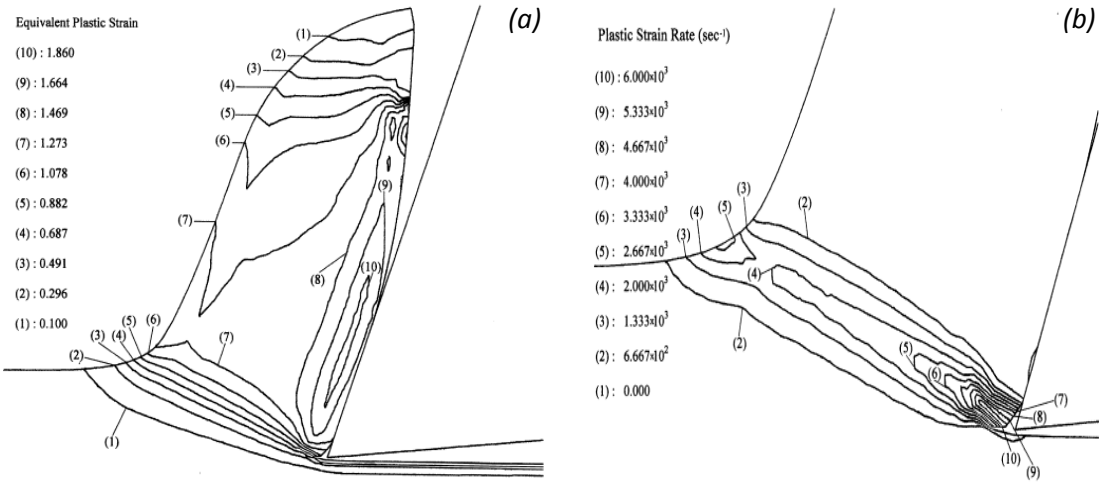


Figure 2.3: Deformation fields of a continuous chip formation process generated by finite element model: (a) equivalent plastic strain, (b) plastic strain rate [33]

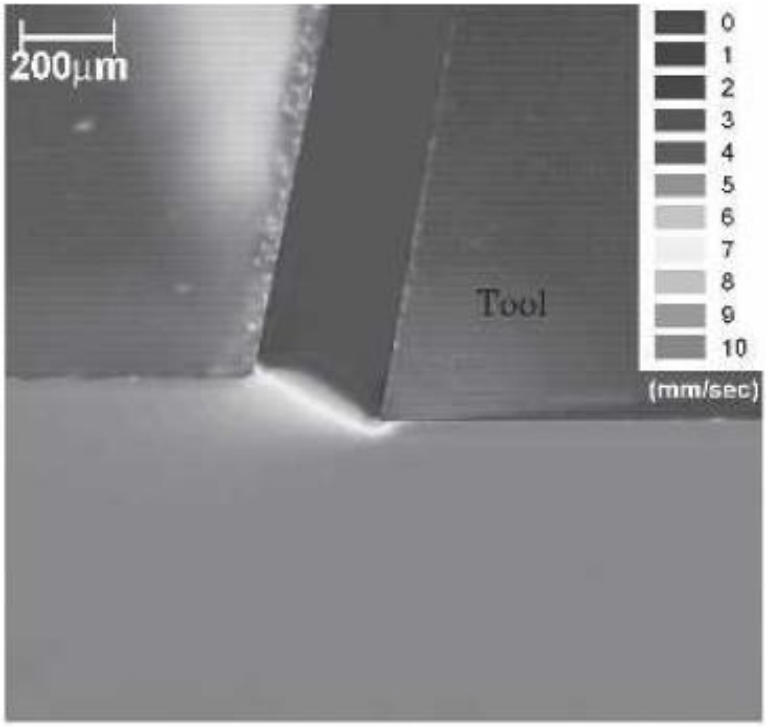


Figure 2.4: Strain rate field generated by PIV [39]

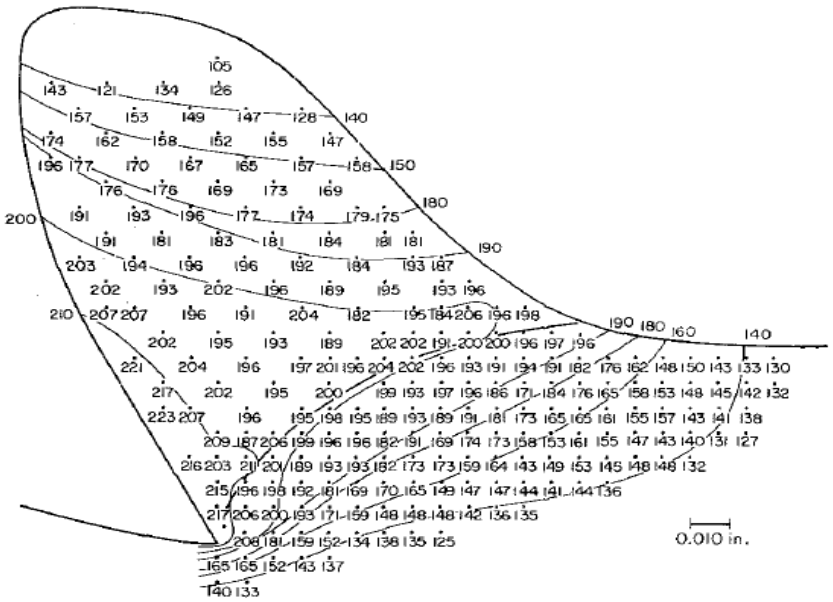


Figure 2.5: Hardness field in incipient chip formation generated by micro-hardness measurements [18]

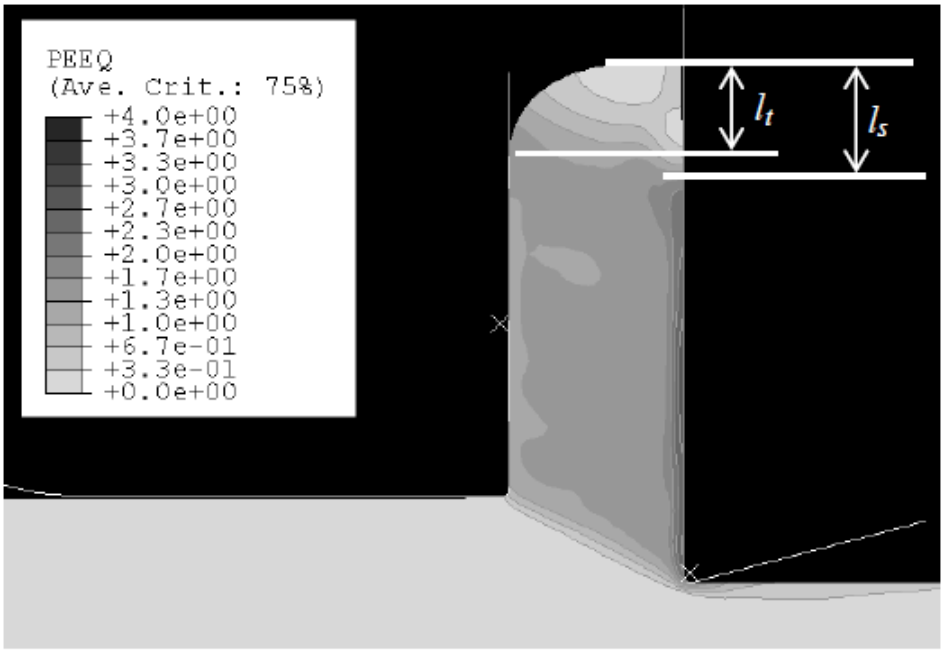


Figure 2.6: A typical strain field in incipient stage of chip formation (α=0°, V_0=10mm/s, a_0=250μm, λ=1.5) [5]

Chapter 3: Experimental Methods

In this chapter, the experimental procedure, equipment and testing conditions that were used are detailed. The overall experimental design is summarized in a flowchart shown in Fig. 3.1, including the following main steps. First, a model material and suitable conditions for orthogonal cutting were selected to understand the effects of the important parameters. The experimental equipment was then assembled to carry out the experiments. During these experiments, in situ measurements were made and were recorded for subsequent analyses. After the experiments, characterizations of the evolution to steady-state were made using three methods: PIV, load profile evolution, and hardness mapping. PIV was be done using images taken in situ during the chip formation process, cutting load analysis was made using in situ load cell data, and Vickers indentation hardness tests were carried out post-mortem on the workpiece materials. These multiple analyses are then combined to assemble a unified understanding of the length of the incipient cutting stage for certain machining parameters.

3.1 Experimental Conditions

The experimental material used in this research is annealed copper (Cu-110). Annealed copper is suitable for the studies of incipient cutting stage for several reasons. First, annealed copper is a pure metal whose hardening is purely driven by microstructure defects, heat treatment can remove the pre-existing defects and effectively homogenize the material. Second, annealed copper has low hardness and good ductility, which ensures that chip formation is continuous under a wide range of cutting conditions. Third, annealed copper significantly work hardens under deformation. The copper was purchased from MSC Industrial Supply Co. in half-hard, billet forms of 12.7 mm x 1828.8 mm x 12.7 mm and 25.4 mm x 1828.8 mm x 3.2 mm. These billets were annealed at 680 °C for 12 hours in an air furnace, followed by furnace cooling. The square bars were then machined to 12 mm x 38.1 mm x 5.1 mm workpieces for the large chip thickness experiments, while the rectangular bars were

machined to 25.4 mm x 25.4 mm x 3.2 mm plates for the small chip thickness experiments. Prior to the experiments, the undeformed hardness of the annealed copper bulks was measured and shown to be 61.7 HV with the standard deviation of 2.4 HV from 40 measurements. The original measurements are displayed in Table 3.1.

Tool rake angle, depth of cut and cutting speed are the primary parameters affecting the chip formation in metal cutting. While cutting speed is known to affect the temperature distribution and friction conditions at the tool-chip interface, its effect on chip geometry is expected to be less pronounced over small ranges of values when compared to rake angle (which determines strain) and depth of cut (determines chip thickness). In the present study, tool rake angle and depth of cut are chosen as the processing variables. Three levels of rake angle and six levels of depth of cut were investigated and are given by rake angle (α) of 30°, 15° and 5° and depth of cut (a_0) of 750μm, 500μm, 250μm, 200μm, 150μm and 100μm. Larger depths of cut and smaller rake angles were not chosen as cutting loads in excess of the machining platform capabilities would result. Smaller depths of cut were not considered due to the difficulty associated with maintaining accurate depth of cut levels in that range.

3.2 Experimental Platforms

Two sets of equipment were used for the orthogonal cutting experiments under different depths of cut, to make the experimental variables controllable and ensure that the experimental environment is appropriate for high-speed imaging (for PIV) and cutting load recording (for cutting load analysis).

3.2.1 Uniaxial Testing Machine

A set of experiments at large depths of cut (e.g., 750, 500 and 250 μm) were carried out on an apparatus consisting of a pre-fabricated die and a uniaxial testing machine operating in displacement-control compression mode. The workpiece motion was provided by a uniaxial testing machine – Instron Model 5866. The load frame is shown

in Fig. 3.2. The Instron was outfitted with compression platens. The bottom platen holds the cutting die and the top platen pushes a billet into the workpiece to produce the prescribed motions (see Fig. 3.3). In the experiment, the crossbar moved down at a predetermined constant speed, namely cutting speed (V_0) of 0.3 mm/s.

The orthogonal cutting configuration was set using a set of pre-fabricated cutting dies. The part drawing is shown in Fig. 3.4 and an image of the actual die is shown in Fig. 3.5. The cutting dies were made from heat-treated alloy steel with hardness greater than 43HRC and the channels were machined such that the cutting edge was sharp. To make the tool rake angle changeable, three cutting dies were manufactured with different rake angles – 30°, 15° and 5°. The tool relief angle was constant at 1°.

The depth of cut is controlled by a 500 μm width difference between the upper and lower channel. To test the different depths of cut, three workpiece shapes were prepared as in Fig. 3.6. The overall length of the workpiece was set to be ~40 mm to guarantee that the chip formation processes was long enough to reach the steady state. The thickness was set to be 5 mm. This provided sufficient width to approximate a state of plane strain cutting condition (chip width at least 10 times the depth of cut). The width of the upper portion of the workpiece was set to 12 mm so to fit the upper channel of the cutting dies. This produces a 500 μm depth of cut, according to the difference in the upper and lower channel widths. To generate two other cutting depths, the geometry of the billet in the vicinity of the cutting edge was modified. For the 250 μm experiments, the 12 mm-wide billet was reduced in thickness near the cutting edge such that a 250 μm depth of cut resulted. Similarly, for the 750 μm experiments, a step in the workpiece geometry was left such that the effective depth of cut was 750 μm. To guarantee that the chip formation is long enough to reach stability, the preset feeding distance (drop distance of the crossbar of the tensile tester) was 15 mm.

3.2.2 Linear Slide

A series of experiments at smaller depths of cut (e.g., 200, 150 and 100μm) were carried out on a servo-controlled, custom linear motion stage with a built-in precision cutting platform (Fig. 3.7). The workpiece was held by the linear motion stage, shown as Fig. 3.8, and the stationary cutting tool was mounted to a precision motion platform fixed to the work table. As in the previous experiments, the cutting speed was set to 0.3 mm/s. The high speed steel cutting tool assembly was mounted to an adjustable platform that allowed for the same rake angles to be used on this platform. In the experiments, the depth of cut was adjusted by a manual stage that controlled the vertical position of the cutter. The error tolerance of this adjustment was 5-10 μm, so the percentage of dimensional error of the 100 μm cutting experiments was smaller than 10%. Other experimental parameters of linear slide cutting included a feeding distance of 25.4 mm (the length of copper plates) and a relief angle of 5° for all the cutting tools.

3.3 Measurement Methods

3.3.1 High-speed Imaging

A suite of measurements were made during and after the cutting process to study the transition from incipient to steady state cutting. The first involves high-speed imaging of the cutting process using a high speed camera (Kodak Motion Corder Analyzer, Model SR-Ultra) mounted to a stereomicroscope (Nikon Model SMZ-2T) (Fig. 3.9) capable of producing a range of magnifications. The field of view ranged from 1.65 mm x 1.55 mm to 3 mm x 2.88 mm under different chip thicknesses and the image size was 480 px x 512 px, effectively giving an effective pixel size of 3.2-6 μm. The frame rate used in the imaging was 30 frames per second. The frame rate and image size of the high-speed camera can impact the accuracy of the PIV results. This will be discussed in the following chapter. The illumination during the process was controlled by a fiber optic assembly capable of delivery through a ring-light assembly. For the image correlation algorithms to work properly, micro-sized features on the surface of the workpieces were needed. To accomplish this, bead blasting was carried out on

the surface of each prepared workpiece. The surface of a workpiece (used on cutting die) before and after bead blasting are compared in Fig. 3.10. High-speed imaging was not used at the smaller depths of cut due to difficulty in imaging the region with the stereomicroscope assembly.

3.3.2 Cutting Force Measuring

The cutting load was captured in both sets of experiments using built-in dynamometry. The curvature of the cutting force traces were used to monitor the growth of the deformation zone during chip formation. For the cutting experiments performed on the tensile tester, the cutting load data were obtained by the Instron load cell with a maximum load of 10 kN, sampling frequency of 10 Hz (see Fig. 3.2) and saved in the connected computer. For the linear slide cutting experiments, the cutting load data were recorded by a force transducer (Kistler, Model 9027A) which was fixed to the cutter (see Fig. 3.7), and sent to computer through a data acquisition system (NI cDAQ). The load limit of the force transduce is 10 kN and the sampling frequency was preset to 100 Hz.

3.3.3 Hardness Testing

Micro-hardness testing was carried out after the cutting experiments on the deformed chips. The samples were first mounted in an air-curing epoxy (Allied, EpoxyMount resin & hardener), in the form of cylinders 38 mm in diameter and 25 mm in height. To prepare these samples for indentation, they were grinded and polished on a Struers Rotopol-22 polishing machine (see Fig. 3.11). The specific operation steps are listed in Table 3.2. Abrasive papers from 320 grit to 4000 grit were used with water flowing. Polishing of the specimens was done with MD-Mol polishing cloth and DiaPro Mol suspension. Final polish was done using OP-Chem polishing cloth with OP-S until the specimens were mirror finish ($Ra < 3\mu m$). The polished chip specimens are shown as Fig. 3.12. Micro-hardness testing was carried out using a Leco MHT 200 tester (see Fig. 3.13), outfitted with a Vickers pyramid indenter. The diagonals of each indentation were measured using an attached microscope and the Vickers hardness of each

measure point was calculated as is shown in Fig. 3.14. The applied load was 25 grams in most of the tests and the resulting impressions were typically 0.02-0.03 mm in diagonal length. On the 100 μm and 150 μm cutting experiments, a 10 gram load (resulting diagonal length ~ 0.015 mm) was used to allow for measurement on smaller thickness chips. The error tolerance of the measuring positions was about 2.5-5 μm. The distance between the centers of two adjacent indentations was set to be larger than 3 times the indentation size to avoid the interaction between the work-hardened regions.

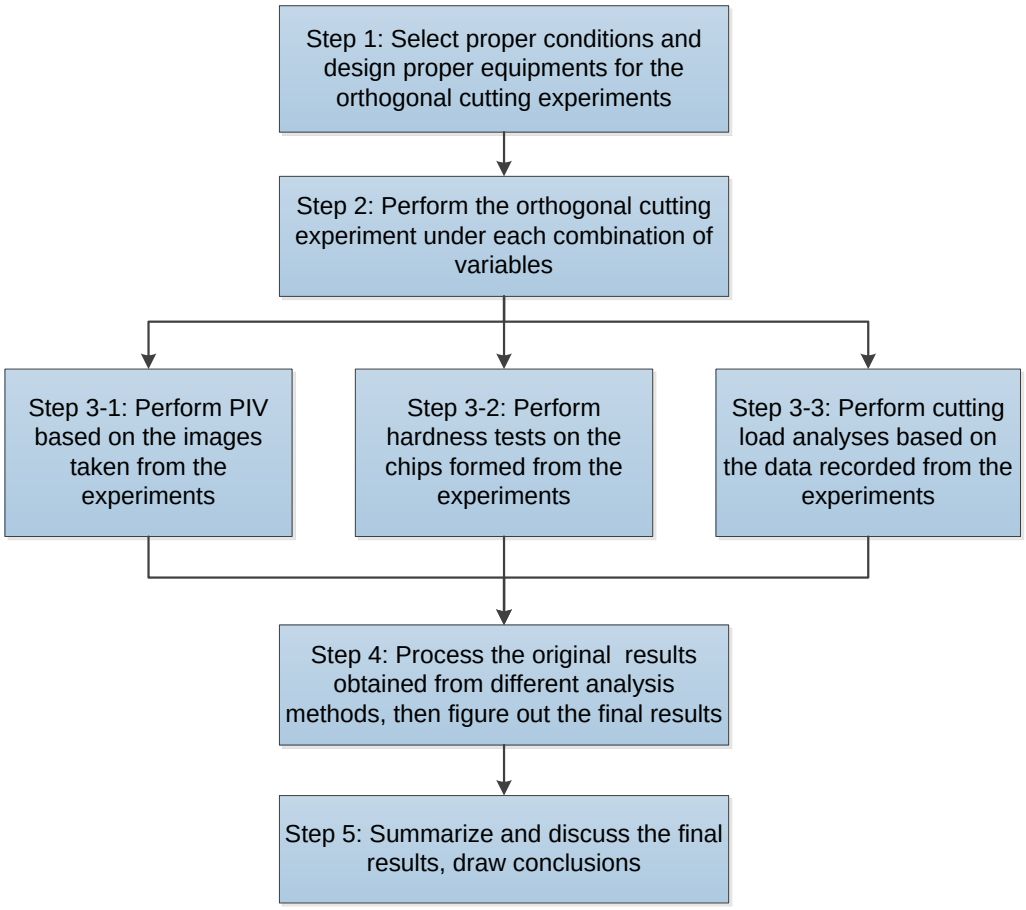


Figure 3.1: Flowchart of the experimental design

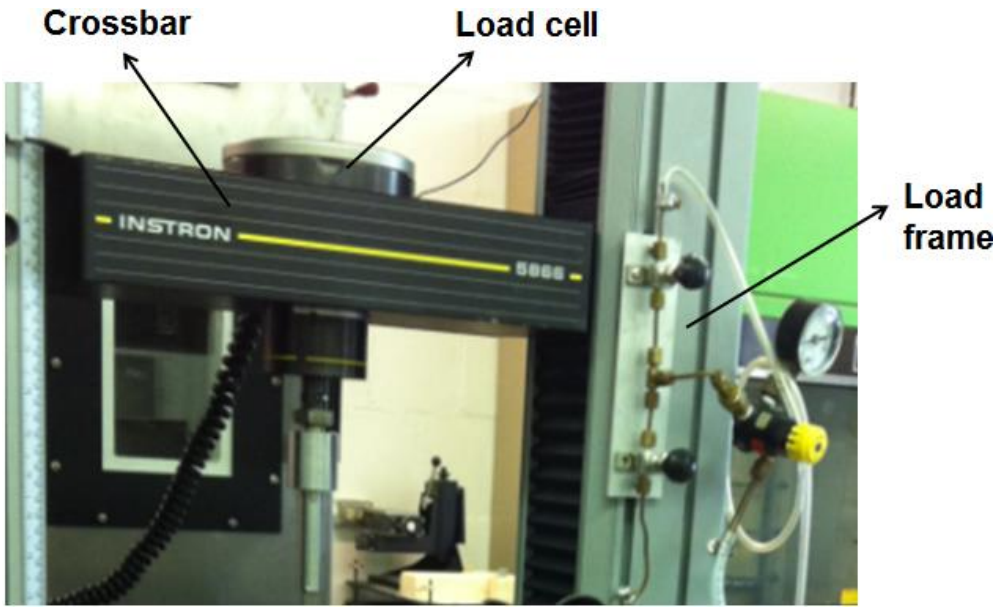


Figure 3.2: Load frame of the uniaxial testing machine used for cutting experiments

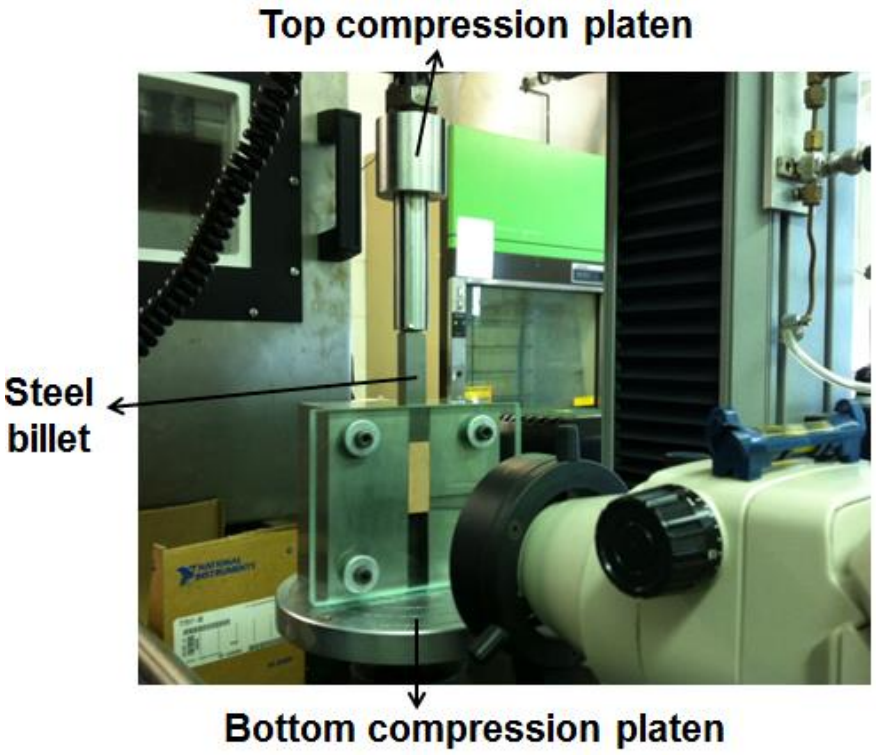


Figure 3.3: Implementation of the cutting action on the uniaxial testing machine

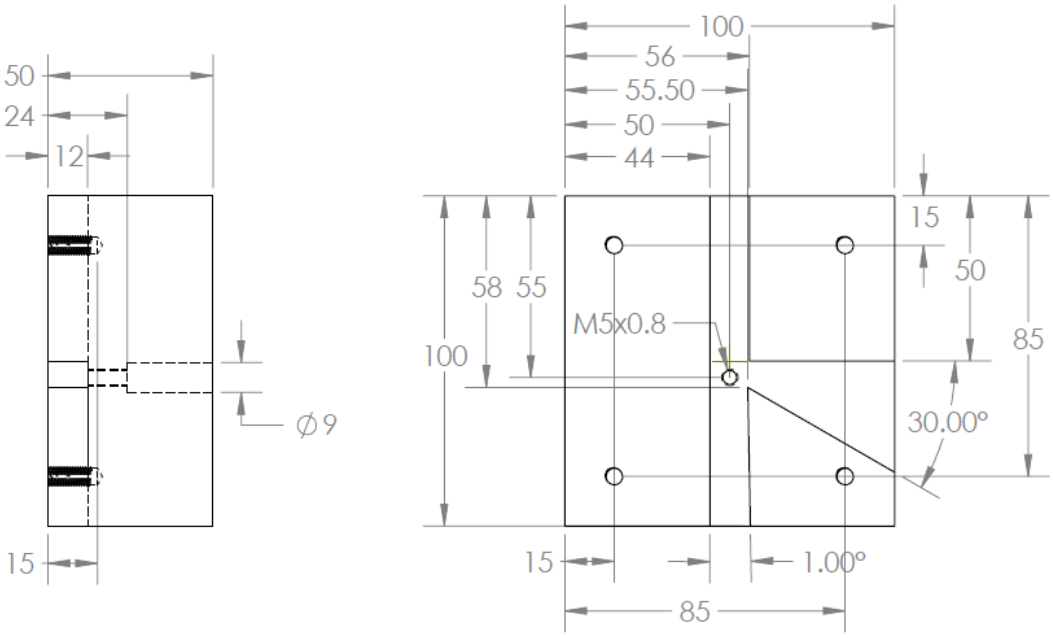


Figure 3.4: Part drawing of 30° cutting die used for cutting experiments (dimensions in mm)

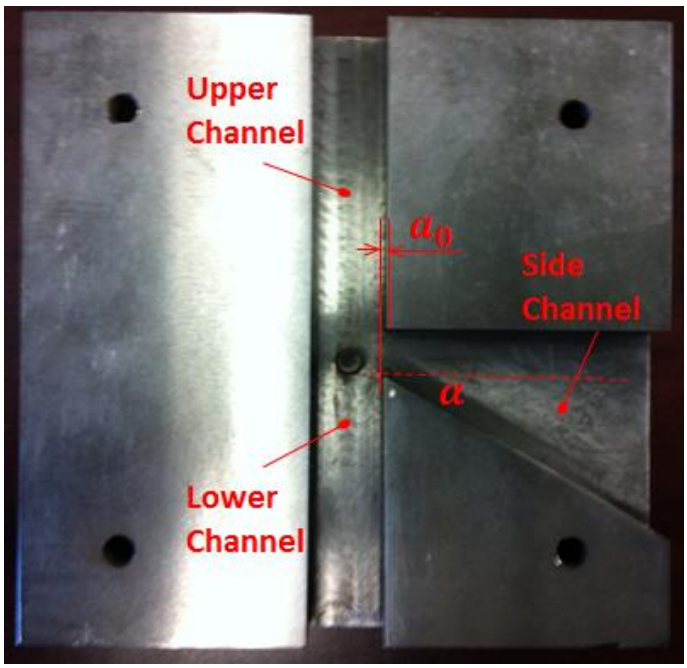


Figure 3.5: Picture of the actual 30° cutting die

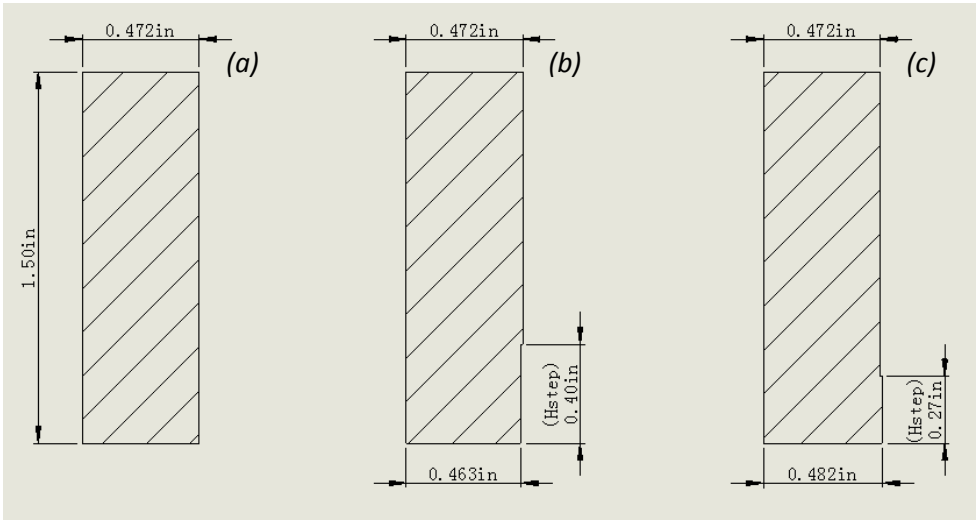


Figure 3.6: Three types of workpieces for the experiments performed on the cutting die and tensile tester: (a) for 500μm cutting experiments, (b) for 250μm cutting experiments, (c) for 750μm cutting experiments

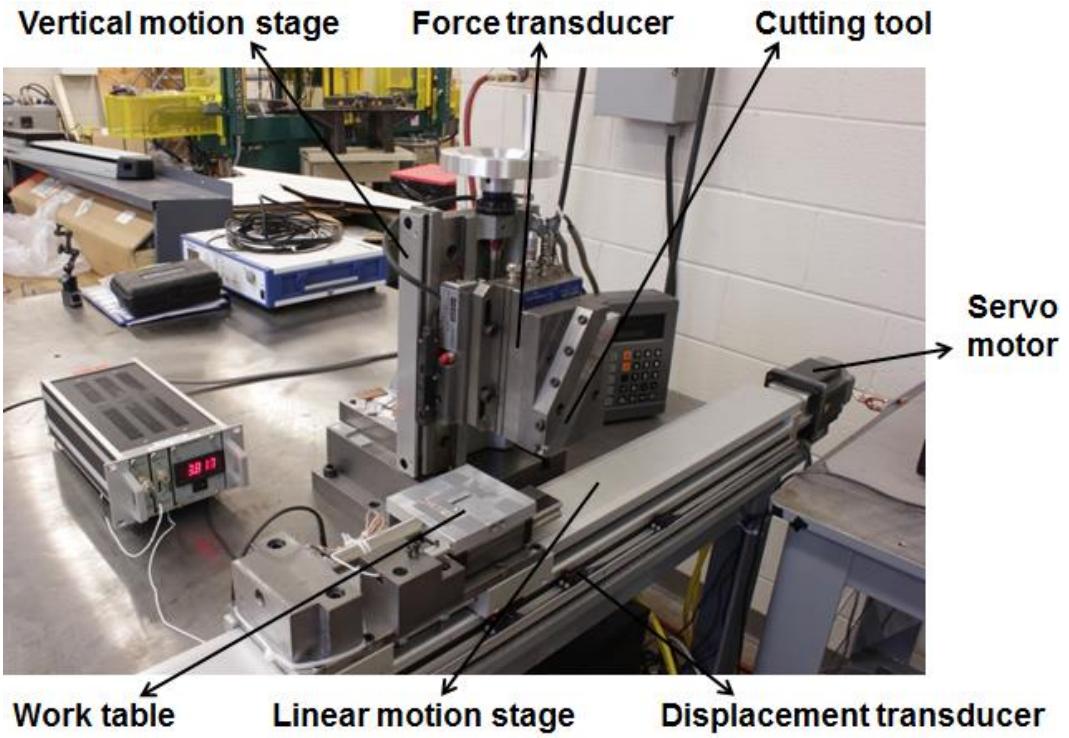


Figure 3.7: Mechanical structure of the linear slide cutting machine

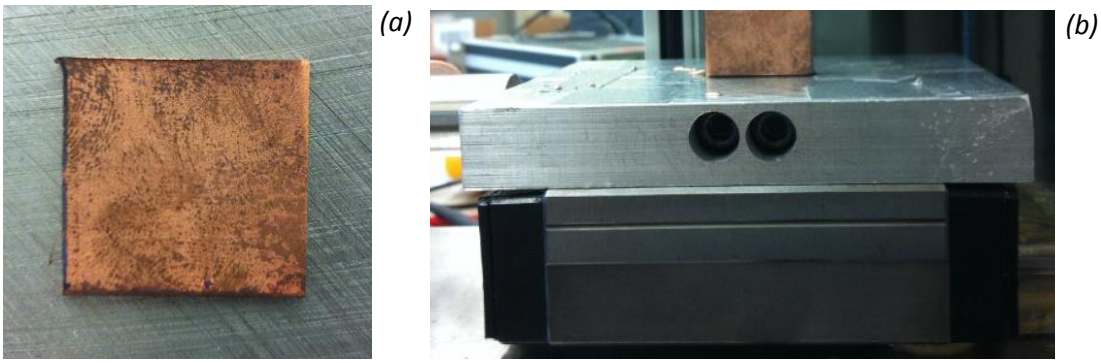


Figure 3.8: Workpiece for the experiments performed on the linear slide cutting machine: (a) machined to the required dimensions, (b) fastened on the work table

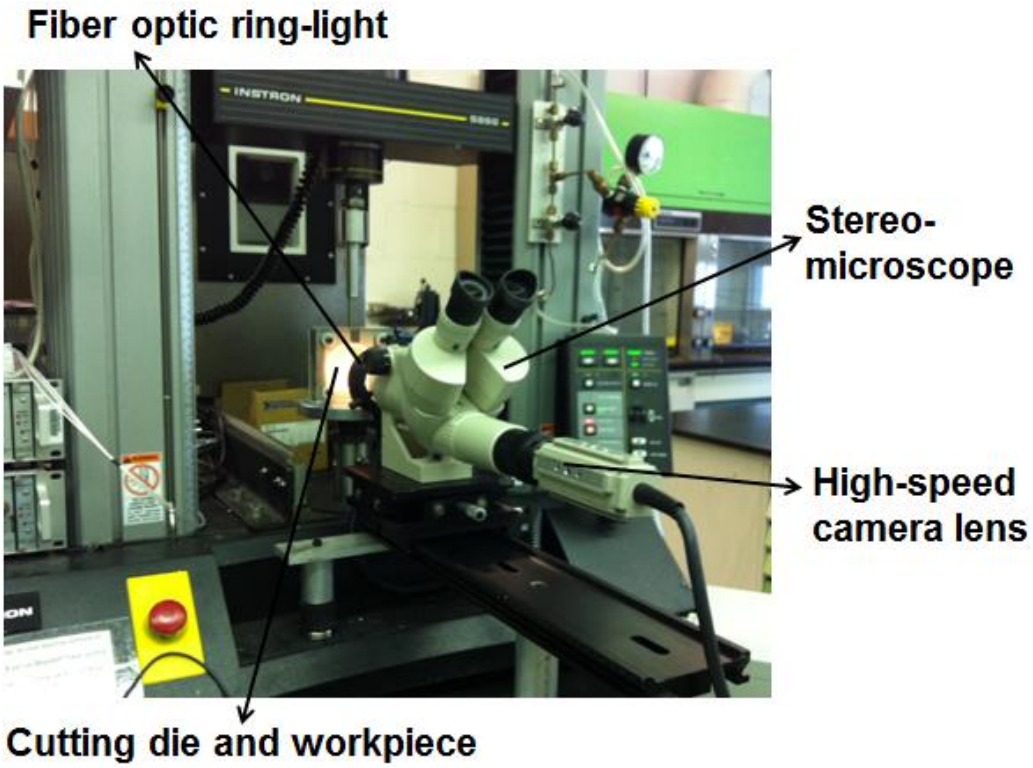


Figure 3.9: Image pick-up devices for the cutting experiments performed on the uniaxial testing machine

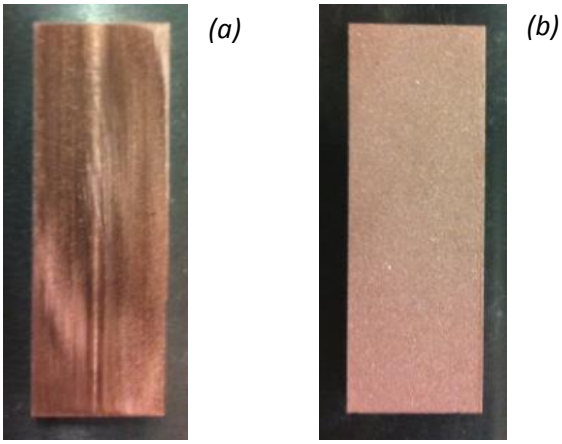


Figure 3.10: Comparison of workpiece surface: (a) before bead blasting, (b) after bead blasting

Figure 3.11: Struers Rotopol-22 polishing machine

(a)

(b)

Figure 3.12: Polished chip specimens: (a) chip formed from ($\alpha = 30°, a_0 = 500\mu m$) experiment, (b) chips formed from ($\alpha = 30°, a_0 = 200, 150, 100\mu m$) experiments

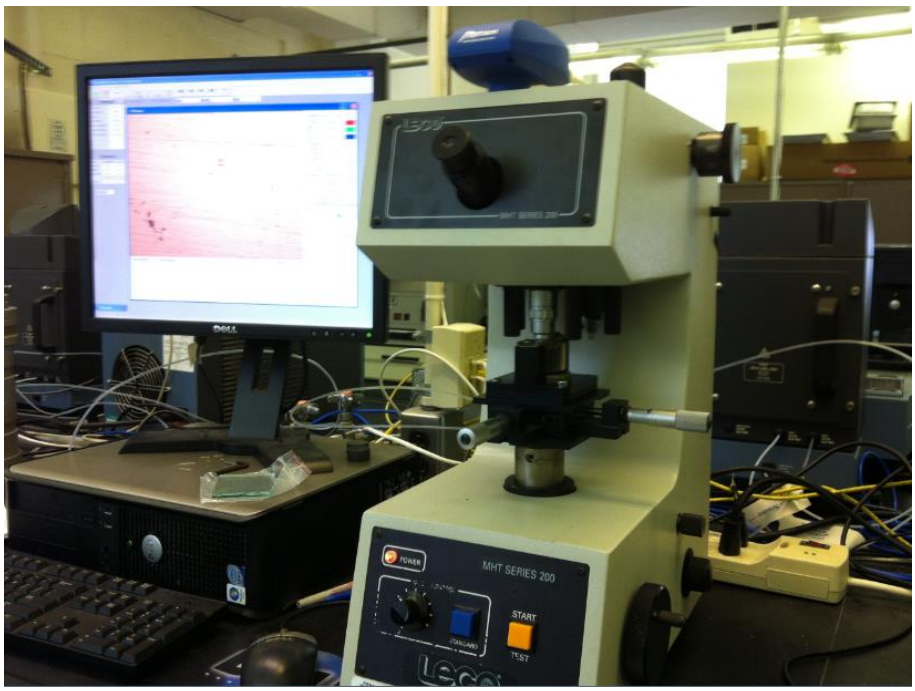
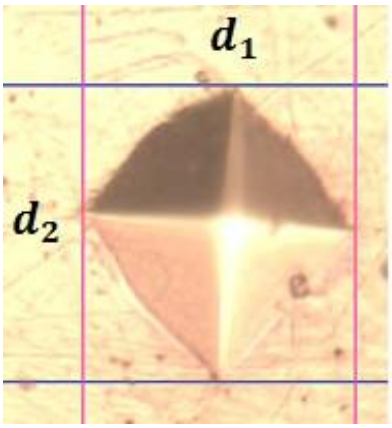


Figure 3.13: Leco MHT 200 hardness tester



$$HV = \frac{0.7564F}{(d_1+d_2)^2}$$

Figure 3.14: Indentation formed in the micro-hardness test and the calculation method for determining Vickers hardness

Table 3.1: Hardness measurements of the annealed copper billets

No.	1	2	3	4	5	6	7	8
HV	**64**	**63**	**64**	**63**	**63**	**63**	**62**	**62**
No.	9	10	11	12	13	14	15	16
HV	**60**	**62**	**62**	**62**	**62**	**54**	**61**	**60**
No.	17	18	19	20	21	22	23	24
HV	**60**	**67**	**63**	**62**	**63**	**61**	**63**	**60**
No.	25	26	27	28	29	30	31	32
HV	**60**	**60**	**59**	**58**	**58**	**63**	**63**	**60**
No.	33	34	35	36	37	38	39	40
HV	**60**	**65**	**66**	**64**	**63**	**62**	**58**	**64**
Mean = **61.73** HV Standard Deviation = **2.42** HV								

Table 3.2: Polishing steps for copper specimens

Grinding						
Step	Surface	Grit	Lubricant	rpm	Force (N)	Time (min)
PG	SiC-Paper	320	Water	300	25	1
FG1	SiC-Paper	800	Water	300	25	1
FG2	SiC-Paper	1200	Water	300	25	1
FG3	SiC-Paper	4000	Water	300	25	1
Polishing						
Step	Surface	Suspension	rpm	Force (N)	Time (min)	
DP	MD-Mol	DiaPro-Mol	150	25	4	
OP	OP-Chem	OP-S	150	15	1	

Chapter 4: Particle Image Velocimetry

Particle image velocimetry (PIV) is a computer-based image correlation technique that was developed for measuring two-dimensional velocity fields in fluid flows and subsequently used for in situ characterization of deformation fields in solid material flow [38,39]. In situ characterization of deformation at sub-pixel accuracy was used in the present study for understanding the deformation in plane-strain machining. In this chapter, a PIV algorithm is established and implemented in Matlab for computation of velocity fields, strain rate fields and strain fields. The basic principles, calibration method and results verification of this algorithm are presented in this chapter.

4.1 Overall Algorithm

The PIV algorithm consists of three main programs to measure three fields of interest: velocity (or displacement), strain rate and strain. First, instantaneous velocity fields are generated from successive high-speed images taken during the deformation process. Second, instantaneous strain rate fields are then estimated by considering the gradients in each velocity field. Finally, strain is computed by evaluating the strain rate fields as a function of time.

4.1.1 Velocity Field

The distribution of velocities in a flow field is measured through determination of the displacement of an area by statistical image correlation of consecutive image frames usually taken with a high-speed imaging system. For the image correlation to be effective, a sufficient number of markers are needed in the images. Thus, a suitable method is needed to introduce asperities onto the material whose deformation is going to be measured, such that the movements of these particles are reflective of the local material flow. The high-speed images are processed by the PIV algorithm according to the basic principle shown in Fig. 4.1 [39]. When analyzing an image pair, the images are first sectioned into square regions called interrogation windows or sub-windows, with side length of N pixels. In PIV, the number density of particles is

required to be high enough such that the mean number of particles within each interrogation window is far more than 1, to guarantee measurement accuracy [51]. The motion of each sub-window is measured between successive image frames using cross-correlation. The motion of the sub-window is indicative of the motion of all the particles contained within it. In Fig. 4.1, if (i,j) is the pixel location within the k^{th} sub-window, the light intensity (e.g., integers of 1-255 in binary images) of this pixel are $f_k(i,j)$ and $g_k(i,j)$ in the two images, the cross-correlation, $\Phi_k(m,n)$, is given by:

$$\Phi_k(m,n) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} f_k(i,j)g_k(i+m,j+n) \qquad (\text{Eq. } 4.1)$$

The vector given by (m,n) corresponds to the peak value of the cross-correlation and thus represents the displacement vector of the sub-window between the consecutive images. The computation time of this vector can be reduced by applying the fast Fourier transform (FFT) to Eq. 4.1 [39,52]. The velocity vector is then obtained by dividing the displacement vector by the time interval Δt between two images, which is determined by the frame rate of camera. By repeating this procedure for every sub-window in the imaged region, we can get the velocity vector for each sub-window in the image to generate a complete velocity field for each image-pair. The Matlab code developed for generation of velocity fields is attached in Appendix A.

A sub-function for masking out image regions that represent free boundaries was added to the main velocity measurement program. This was used to avoid noise caused by the imperfectly dark background and also save computing time from solving unnecessary cross-correlation functions. The principle of this masking sub-function can be explained based on Fig. 4.2. Fig. 4.2(a) is an image taken in a 30° cutting experiment. Fig. 4.2(b) is an artificial image created to cover the stationary cutting tool. Two Matlab built-in functions – “im2bw” and “imfilter” – were used to convert the original image (excluding the cutter zone) from gray-scale to binary and then smooth the binary image by blurring (see Fig. 4.2(c)). After finding the edge of

workpiece and eliminating the noise points on the background-side, the final masked region is shown as the white region in Fig. 4.2(d). For computation purposes, velocity vectors for sub-windows with centers located in the masked region are set to zero. The Matlab code developed for the masking sub-function is attached in Appendix B.

Another auxiliary sub-function that is important in the velocity field determination is that of data validation to remove outliers from the calculated velocity field. Fig. 4.3 [52] provides a typical output of a PIV analysis of a continuous and homogeneous flow field, wherein a number of spurious vectors are observed and must be removed. These vectors are results from interrogations in which the amplitude of the displacement-correlation peak did not exceed that of random correlation peaks [51]. To determine whether an individual vector v_0 is valid, a normalized median test was applied. The median of all neighboring vectors to vector v_0 in an R x S sub-windows region centered on vector v_0 (e.g., 3 x 3 in the red frame in Fig. 4.3) is referred to as the local median v_m. The median of the absolute differences between these neighboring vectors and the vector v_0 is referred to as the local residual median r_m. Vector v_0 is considered valid if the following criterion is fulfilled:

$$\frac{|v_m - v_0|}{r_m + 0.1} < T \qquad \text{(Eq. 4.2)}$$

where T is a user-defined threshold [51]. In the present data-validation method, vectors that did not pass the normalized median test were replaced by the respective local median v_m for the neighbors of those vectors. The Matlab code of this sub-function is attached in Appendix C. The modified velocity field can be shown as a color map by using the Matlab code in Appendix D.

4.1.2 Strain Rate Field

The strain rate field is derived from the measured flow (velocity, displacement) information. The strain rate components at a specific point in the field can be calculated by differentiating the corresponding velocity vector for that point as [38]:

$$\dot{\varepsilon}_{xx} = \frac{\partial u}{\partial x}, \quad \dot{\varepsilon}_{yy} = \frac{\partial v}{\partial y}, \quad \dot{\gamma} = 2\dot{\varepsilon}_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \qquad \text{(Eq. 4.3)}$$

where u and v are components of velocity vector in the x and y directions, respectively; ε_{xx} and ε_{yy} are the normal strains; γ is the shear strain. The central difference method was applied to approximate the value of the differential in the PIV algorithm, as the velocity vectors have been obtained for each sub-window in the velocity field. Fig. 4.4 shows a grid block that consists of 3 x 3 sub-windows, the strain rate components of the center window can be figured out by:

$$\dot{\varepsilon}_{xx} = \frac{u_3 - u_2}{2N}, \quad \dot{\varepsilon}_{yy} = \frac{v_4 - v_1}{2N}, \quad \dot{\gamma} = 2\dot{\varepsilon}_{xy} = \frac{v_3 - v_2 + u_4 - u_1}{2N} \qquad \text{(Eq. 4.4)}$$

where (u_1, v_1) and (u_4, v_4) are the velocity vectors of the adjacent sub-windows in the y direction, respectively; (u_2, v_2) and (u_3, v_3) are the velocity vectors of the adjacent sub-windows in the x direction, respectively; N is the window size. To avoid the difficulties in the following strain calculation associated with varying principle directions of the strain rate tensors [26], the effective strain rate (scalar) for a point was used in the PIV algorithm. The effective strain rate is related to the strain rate components as [38]:

$$\dot{\bar{\varepsilon}} = \sqrt{\frac{4}{9}\left(\frac{1}{2}\left[(\dot{\varepsilon}_{xx} - \dot{\varepsilon}_{yy})^2 + \dot{\varepsilon}_{xx}{}^2 + \dot{\varepsilon}_{yy}{}^2\right] + \frac{3}{4}\dot{\gamma}^2\right)} \qquad \text{(Eq. 4.5)}$$

The effective strain rate field can be obtained after determining the effective strain rate for each sub-window in the velocity field. It should be noted that, errors may occur in sub-windows adjoining the boundary between the wokpiece and background (e.g., masked region) because the masking sub-function sets all velocity vectors in the background to zero. This would lead to very large gradient values at free-boundaries. To mitigate this problem, a boundary identification sub-function was embedded to the program that modified the measurements in the strain rate field. This sub-function replaced the strain rate values for sub-windows in the boundary by the median of the strain rate values inside the cutting material (and not in the masked region). Further,

the strain rate values in sub-windows in the masked region were set to zero directly. The Matlab code developed for determination of the strain rate field is attached in Appendix E.

4.1.3 Strain Field

Utilizing the algorithm described, velocity fields and effective strain rate fields can be obtained for each image-pair during the entire material deformation process. Of relevance to deformation field characterization is the overall effective strain, e.g. the strain field. The total effective strain at a point in each strain rate field is estimated by summation of the effective strain increments along the deformation path that lead to and terminate at this point [26]. Using path 1 in Fig. 4.5 [26] as example, this path is determined by tracing the endpoint (e.g., point 10 in the image) based on the flow history given by all the velocity fields. This can be accomplished as follows. Let position i represents the position where point 10 was in the i^{th} frame. Since the strain rate distribution of all frames is known, the effective strain increment accrued between point i and point $i+1$ can be obtained by multiplying the local effective strain rate of point i by the time interval between two successive images Δt. Based on the principle introduced above, the effective strain of any point can be given by:

$$\bar{\varepsilon} = \sum_{i=1}^{F} \dot{\bar{\varepsilon}}_{k(i)}(i) \cdot \Delta t \qquad \text{(Eq. 4.6)}$$

where F is the number of frames wherein the measured point is in the field of view, $k(i)$ is the identification number of the sub-window where the measured point is located in the i^{th} frame, and $\dot{\bar{\varepsilon}}(i)$ is the effective strain distribution of the i^{th} frame. As hundreds or thousands of images are captured in a typical experiment, this calculation method takes a considerable amount of time. In the present PIV algorithm, only the effective strain of the center point of each sub-window was measured using Eq. 4.6, and the complete strain field was generated using interpolation. The Matlab code developed for strain measurement is attached in Appendix F.

4.2 Calibration of the Algorithm

The PIV algorithm was tested to evaluate its performance and accuracy. Artificial images with varying imaging characteristics were utilized. In these images, the particle flow is preset and constant in each sub-window such that the deformation parameters can be predicted accurately. Fig. 4.6 is a typical artificial image that simulates the plane-strain orthogonal cutting condition. Particles with varying gray-scale intensity are randomly generated at a specified density level. The flow is akin to a cutting process, wherein some material moves undisturbed underneath a cutting tool (e.g., workpiece) and some material is deformed with a relative change in direction (e.g., chip). Between two successive images, all of the particles in the undeformed workpiece (yellow frame) move downwards v_w pixels; all of the particles in the deformation zone (red frame) move downwards v_d pixels and rightwards u_d pixels; all of the particles in the chip (green frame) move downwards v_c pixels and rightwards u_c pixels. The relative values for each of these inter-frame displacements are controllable and clearly determine the amount of deformation that occurs. The standard parameters of a group of artificial images are as follows:

1) Image size of 1000 px x 1200 px, 120 frames total;
2) Rake angle of 45°, perfectly sharp tool;
3) Window size of 20 px x 20 px;
4) Deformation zone thickness of 4 windows (80 px);
5) Particle size of 2 px x 2 px, density of 50%, intensity level (integer from 1 to 255) randomly assigned;
6) Inter-frame displacements of: $v_w = v_d = v_c = 4,\ u_d = 2,\ u_c = 4.$

It should be noticed that the movement behaviors set herein are a little different from those in the actual chip formation processes, where translational speed of the particles should decrease due to deformation. To facilitate the prediction of deformation, all particles share a constant vertical speed in the artificial images. However, such an unrealistic assumption will not influence the calibration, which is

only aiming to investigate the consistency between the PIV results and the expected results. Using the PIV algorithm described above to evaluate this group of images, the velocity, strain rate and strain fields were obtained and are given in Figs. 4.7-4.9. As the displacement vectors are pre-assigned, it is relatively straightforward to determine the velocity, strain rate and strain fields analytically using classical mechanics. These calculations are provided in Appendix G and were used to generate the same fields analytically for comparison with the results of the numerical methods used here. These two sets of fields were subtracted from one another to generate residual error fields that show the error present in the numerical PIV algorithm. These fields are shown in Figs. 4.10-4.12. For all imaged regions, including the workpiece, deformation zone and chip, the mean and variance of the PIV results were calculated based on the difference between the numerical and analytical velocity vectors and effective strain rate values. The detailed comparison between the PIV results and the expected results is shown in Table 4.1. The PIV algorithm clearly provides highly accurate results which indicate that the maximum velocity errors, strain rate errors and strain error are all ~2%. The strain field shows a higher variance, due to the fact that it is generated from a summation over multiple fields and is not a simple gradient developed from an individual field.

4.2.1 Optimal Particle Parameters

The sensitivity of the PIV results to changes in imaging and experimental parameters is important in understanding the effectiveness of the PIV algorithm. Such experimental parameters include: particle size, particle density and inter-frame displacement. The first variable that was considered was particle size. The remaining imaging/experimental parameters were kept constant at the levels described above and four different particle sizes were considered: 1 px x 1 px, 2 px x 2 px, 3 px x 3 px and 4 px x 4 px. This can be visualized in Fig. 4.13, which shows a small region (40 px x 40 px) in size for artificial images exhibiting the variation of particle size. The residual error fields for velocity, strain rate and strain for each of these particle sizes is shown in Figs. 4.14-4.16. Comparison of the four groups is given in Table 4.2, which

indicates that the use of smaller particles clearly is beneficial to ensure the accuracy of the numerical method. The mean error of the velocity measurements and the strain rate measurements was as low as 0.5% and 0.7% for the smallest particles, but respectively increased to 5.6% and 4.6% when the particle size is largest. Although the mean strain errors look similar in different groups, the variance of the strain field increases dramatically with the particle size, which means much more uncertainty and lower reliability of individual strain measurement.

Particle density was the next parameter that was varied, while all other parameters were held constant at the original levels. Five density levels - 20%, 35%, 50%, 65% and 80% - were used to create the artificial images. This can be visualized in Fig. 4.17, which shows a small region (40 px x 40 px) in size for artificial images exhibiting the variation of particle density. Comparison of the five groups is given in Table 4.3, which indicates that the use of larger particle densities is helpful for ensuring the accuracy of the numerical method. From the particle density of 20% to 80%, the mean velocity error decreased from 2.6% to 1.6%; the mean strain error decreased from 2.8% to 1.4%; meanwhile the variance of the stain measurements reduced by about 80%.

Finally, the effect of varying particle inter-frame displacement was investigated. Four different speeds were tested in the artificial images, given as: $(v_w = v_d = v_c = 2,\ u_d = 1,\ u_c = 2)$; $(v_w = v_d = v_c = 4,\ u_d = 2,\ u_c = 4)$; $(v_w = v_d = v_c = 6,\ u_d = 3,\ u_c = 6)$; and $(v_w = v_d = v_c = 8,\ u_d = 4,\ u_c = 8)$. These velocities were selected on the basis that all combinations guaranteed an equivalent strain rate distribution in the deformation zone, with progressively greater displacement between images. Comparison of the five groups is given in Table 4.4, which indicates that relatively lower inter-frame displacements are beneficial to ensure the accuracy of the numerical method. Under the smallest inter-frame displacements, the average velocity error and strain rate error were 2% and 1.9%; however, these errors respectively increased to 2.5% and 3.3% for the largest inter-frame displacements. Again, the strain measurements exhibited much stronger fluctuation with the decreasing accuracy of PIV. Because the sub-window size was 20 px x 20 px, the

analysis above suggested that the most appropriate inter-frame displacement of particles should be smaller than 20% of the sub-window size. In practice, this is accomplished by adjusting either the frame rate of the imaging or the speed of the deformation platform.

In summary, the numerical PIV method is more accurate with a dense distribution of fine particles in the image region. Further, lower cutting speeds and/or higher imaging frame rates act to improve accuracy. This understanding of the sensitivity and accuracy of the PIV method is helpful in experimental design and parameter selection. The selection of the particle-introducing method and the cutting speed in the orthogonal cutting experiments, described in Chapter 3, were guided by these error analyses.

4.2.2 Optimal Program Parameters

Apart from characteristics of the image (e.g., particle size, particle density) and imaging parameters (frame rate), parameters used in the numerical method also affect the quality of the results. Two program parameters, window size (size of the interrogation windows) and grid size, are key elements in this regard. As was discussed earlier, larger window sizes tend to improve the accuracy of the cross-correlation; however, increasing window size also leads to a smaller number of sub-windows and results in roughly-defined measurement fields. To mitigate this problem, overlapping sub-windows have been proposed [52]. The use of overlapping sub-windows results in the number of detected velocity vectors not depending on the window size and gives rise to the notion of a “grid size” that represents the distance between two adjacent vectors (e.g., centers of overlapping sub-windows). According to the principles introduced in the previous section, grid size may have a strong impact on PIV results due to the use of normalized median tests and central difference methods. To further examine the importance of these parameters, a sub-region, 200 px x 150 px in size, was taken from an image of actual orthogonal cutting experiment (Fig. 4.18). A constant displacement of 2 pixels per frame (similar

to the actual cutting speed) was applied to each pixel in this image and the effects of various program parameter settings on the ability to measure this displacement were studied.

First, the effect of window size was studied. The grid size was set constant at 10 px and six different window sizes – 12 px, 16 px, 20 px, 24 px, 28 px and 32 px – were tested. The resulting error in the velocity field measurements is shown in Table 4.5. From the results it is clear that a larger window size results in more accurate results. The error for a window size of 12 px was 25.6%, this falls to 9.5% for a window size of 32 px. However, this trend may not be valid for heterogeneous flows. For example, if the window size is set large such that fundamentally different flows occur in the same window, this obviously will have an adverse effect on the results. It should be noted that for real images, using window size 5 times the maximum inter-frame displacement may not be satisfactory. Even though for the window size of 32 px that is 16 times the displacement, the velocity error is still higher than in the case of the artificially generated images. This is likely due to the imperfect nature of the real images. For this reason, the window size of 32 px was used in all the PIV analyses for the entire thesis. A larger window size was not used because of the steep velocity gradients around the deformation zone.

Second, the effect of grid size was studied. The window size was set constant at 32 px and the grid size was varied over six different grid sizes - 6 px, 8 px, 10 px, 12 px, 14 px and 16 px. In addition to monitoring the errors in velocity, the error in strain rate was measured for this case as the grid size has a direct effect on the results of the central difference method calculations used for strain rate estimation. The results of these tests are shown in Table 4.6. As is clear from the table, the grid size had a marginal effect on velocity error and a clear trend in reducing the strain rate error. The noise level of strain rate for a small grid size of 6 px was 10.72×10^{-4}; this falls to 4.70×10^{-4} for a grid size of 16 px. It also can be seen that the strain rate error for a grid size of 14 px was lowest at 3.80×10^{-4}. This can be understood in that once the grid size is too large, loss of detail regarding material flow variations is possible. The

observation that smaller grid size leads to larger errors in strain rate field can be understood in considering Eq. 4.4. As the sub-windows are overlapping, *N* can be substituted by the grid size in the expression. As the magnitude of the velocity errors remains nearly unchanged for a constant window size, decreasing grid size is expected to increase errors in strain rate measurement.

In summary, for relatively homogeneous material flows, the numerical PIV method is more accurate with a larger window size and a moderate grid size. For a given material flow velocity and frame rate, the window size is better to be at least 10 times as large as the maximum inter-frame displacement, as indicated in the previous sub-section. The selection of grid size depends on the window size. The test results show that a high measurement accuracy can be achieved when the size of grids is about half of that of the sub-window size.

4.2.3 Results Verification

Lastly, a consistency check of the PIV results (under steady-state conditions) can come from comparison with the analytical upper-bound solutions using Eqs. 2.4 and 2.5. The high-speed images taken during orthogonal cutting of copper under conditions of ($\alpha = 30°, a_0 = 500\mu m$) were used for this consistency check. High-speed images were obtained of the steady-state portion (non-varying chip thickness) of the chip formation process (Fig. 4.19). To ensure a high degree of accuracy, a window size of 32 px and a grid size of 16 px were used in the subsequent analysis. The velocity, strain rate and strain fields obtained by the PIV algorithm are shown in Fig. 4.20-4.22. The steady cutting speed was 0.3 mm/s. In the velocity field, the mean of the vertical components of the velocity vectors in the undeformed workpiece was 0.277 mm/s; this yielded an error of ~7.7%. With regard to strain rate, the approximate effective strain rate at the shear plane is 0.82 from Eq. 2.5, under the rake angle of 30°, cutting speed of 0.3 mm/s and deformed chip thickness of ~910 μm. The PIV result is 0.75; this yielded an error of ~8.5% With regard to strain, the strain in the chip using Eq. 2.4 was 0.912 for a rake angle of 30° and chip thickness ratio of 1.82, while

that obtained with PIV was ~1. This resulted in an error of ~8.8%. Further, comparison with Fig. 2.6 shows that the strain distribution through the deformation zone is consistent with that expected for chip formation in machining. The close nature of the PIV measurements and the analytical solutions demonstrate that the PIV algorithm developed here is a suitable tool for analyzing the deformation behaviors in chip formation processes.

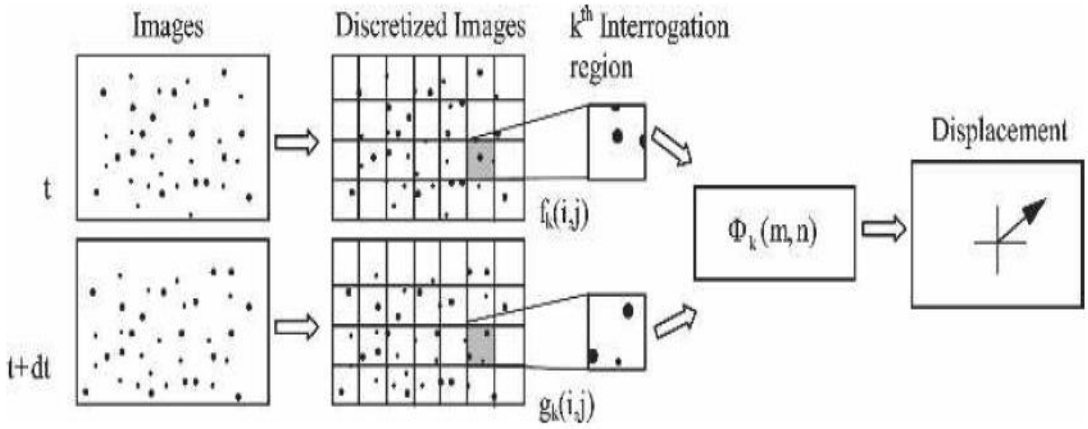


Figure 4.1: Principle for obtaining the velocity fields in the PIV algorithm [39]

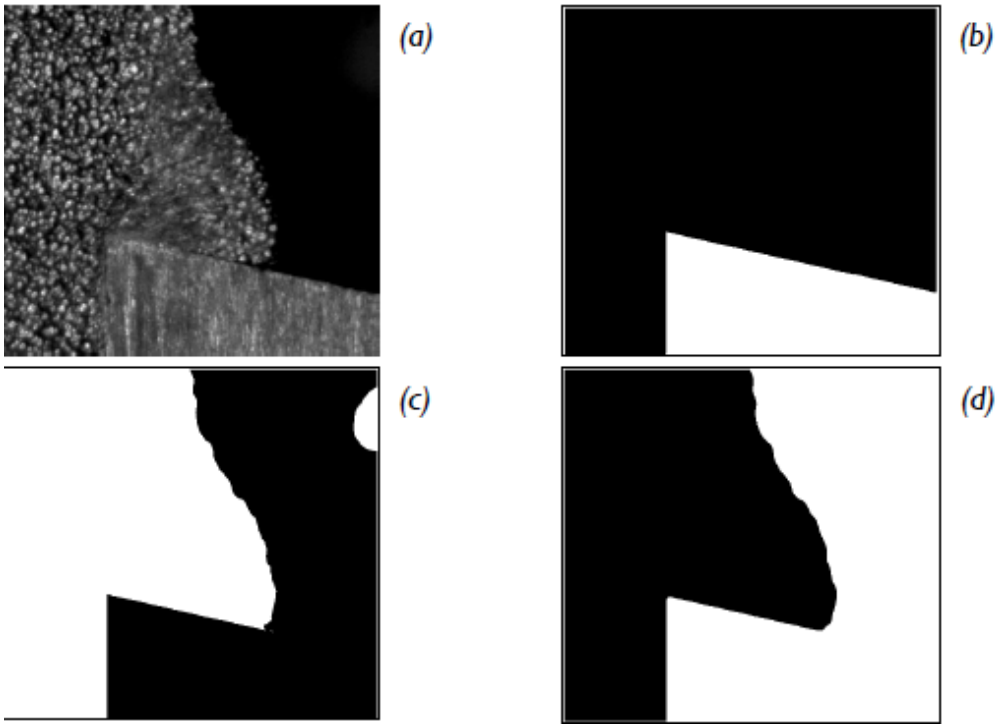


Figure 4.2: Principle of the masking sub-function: (a) original image, (b) artificial masking image for covering the cutter, (c) conversion of the original image from grayscale to binary, (d) ultimate masking image

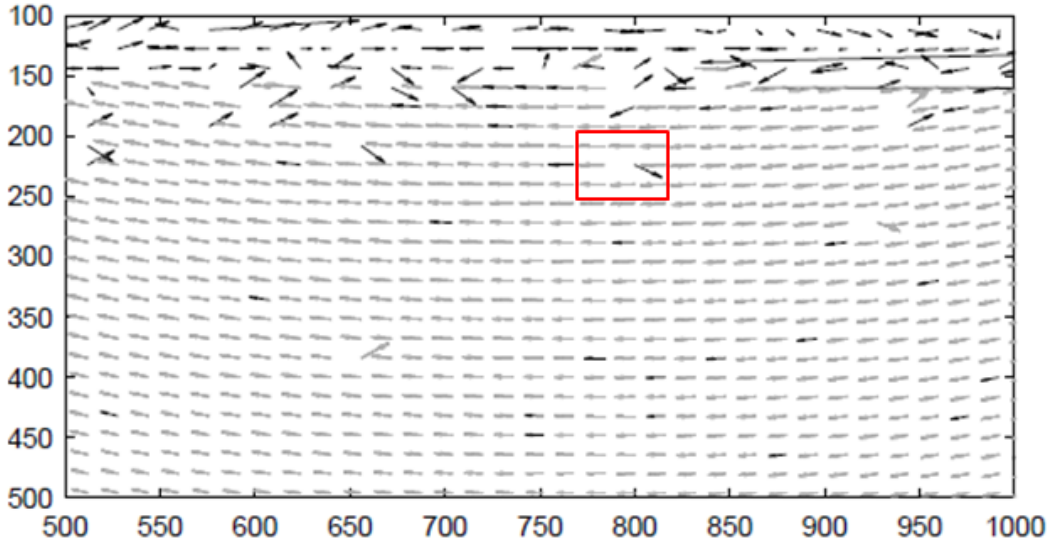


Figure 4.3: A typical PIV velocity field with invalid vectors [52]

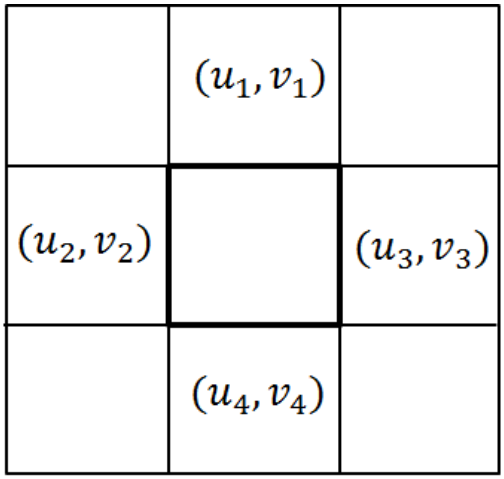


Figure 4.4: Principle of central difference method used for obtaining the strain rate field in the PIV algorithm

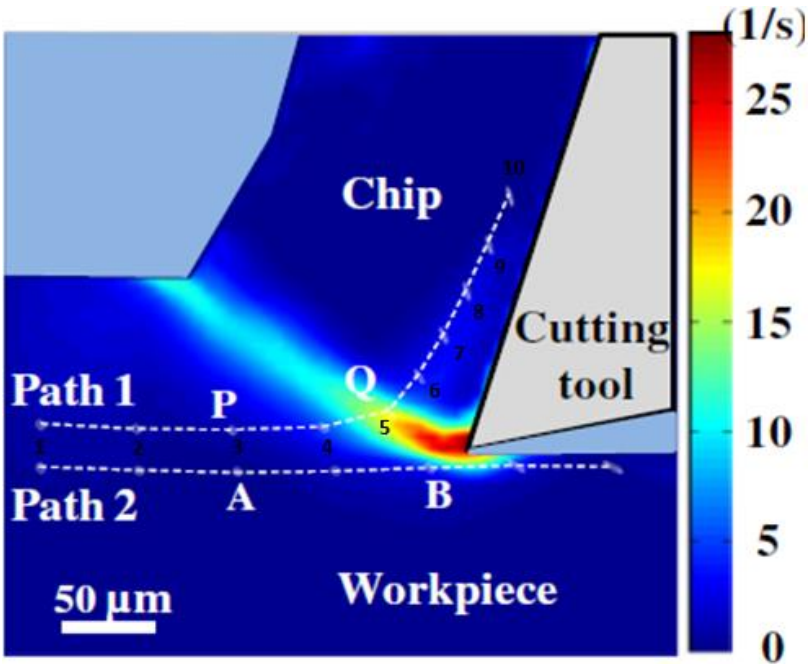


Figure 4.5: Principle for strain calculation in the PIV algorithm [26]

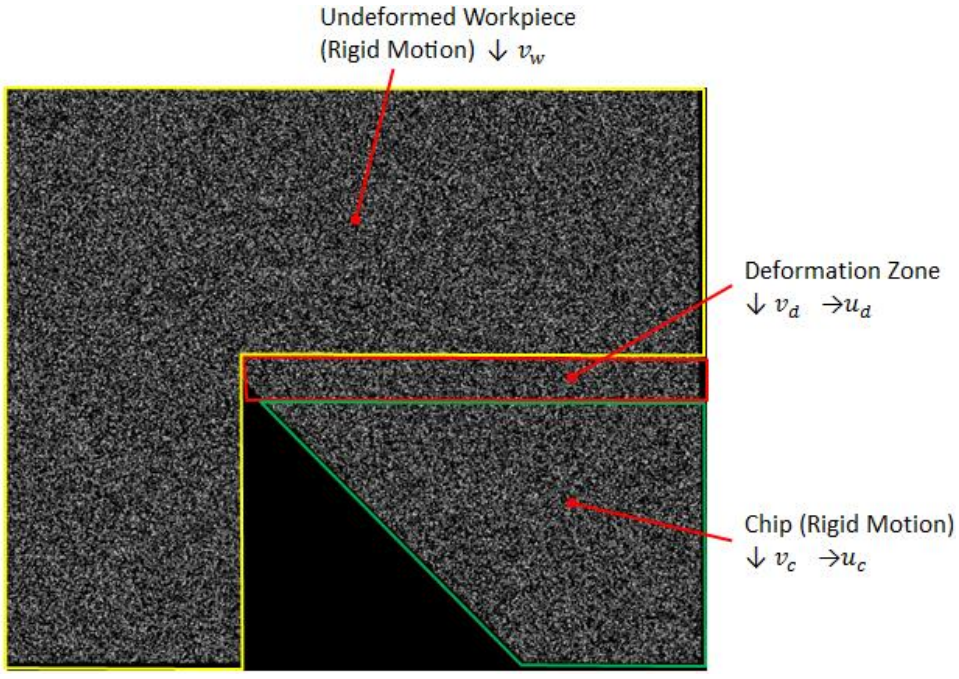


Figure 4.6: Setting of particle movements in artificial images used for calibration

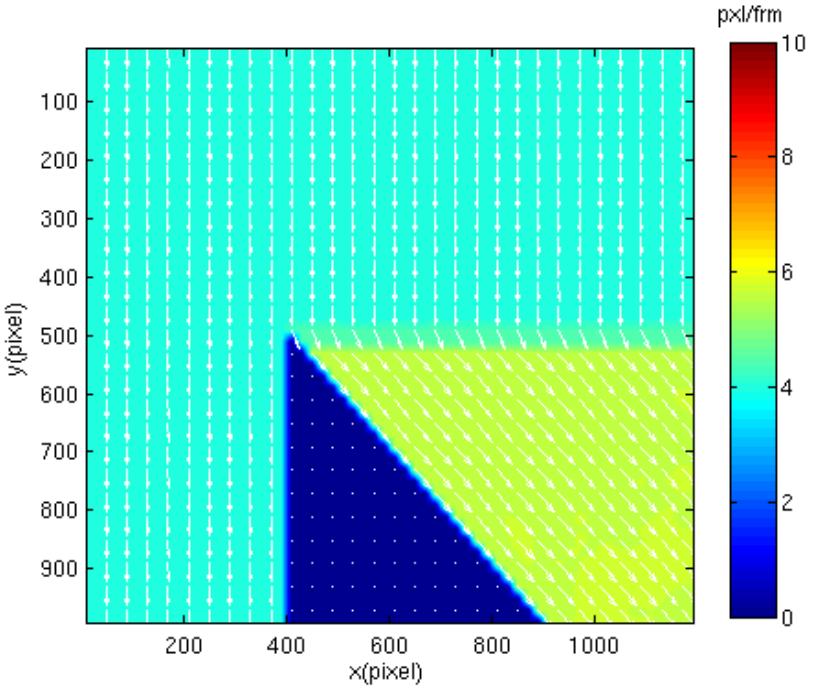


Figure 4.7: Velocity field by PIV for the standard calibration images

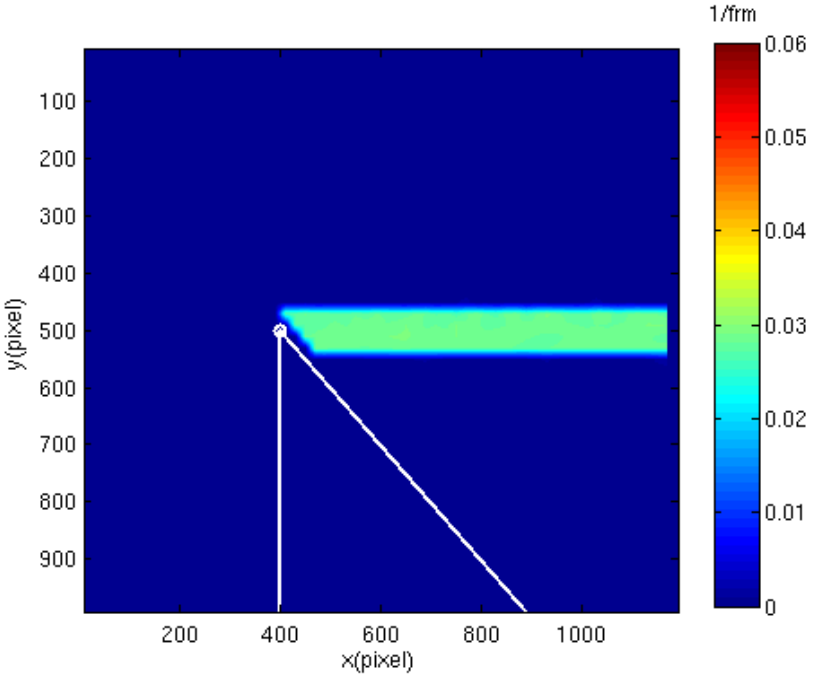


Figure 4.8: Strain rate field by PIV for the standard calibration images

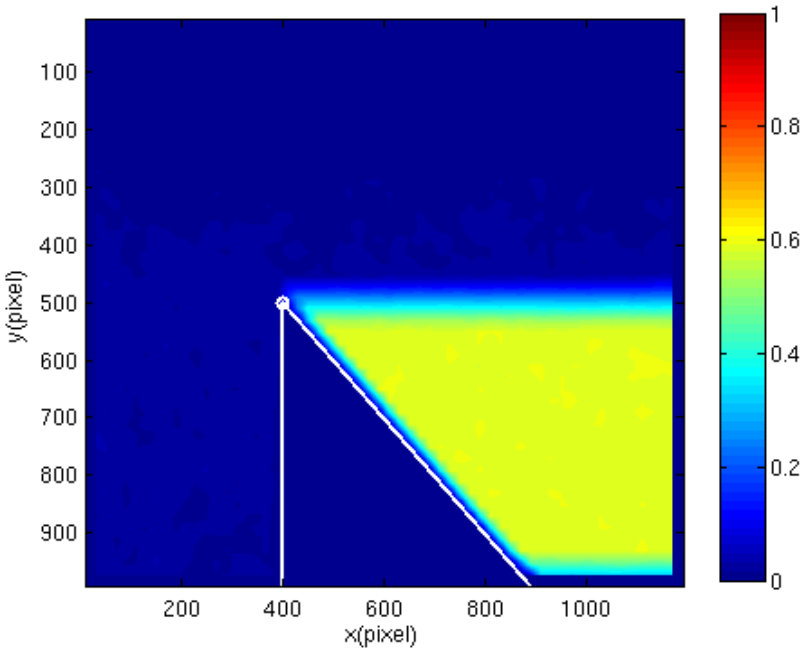


Figure 4.9: Strain field by PIV for the standard calibration images

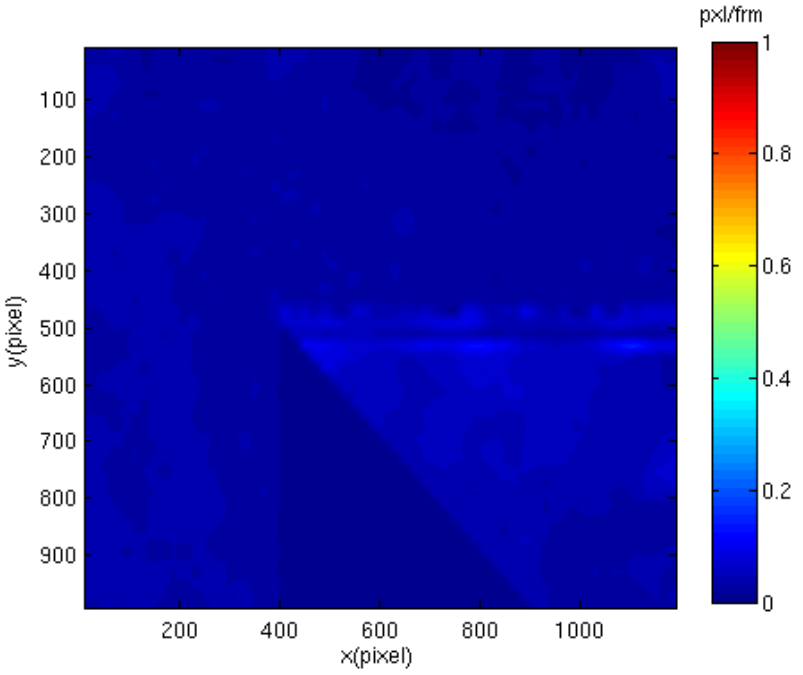


Figure 4.10: Residual velocity field for the standard calibration images

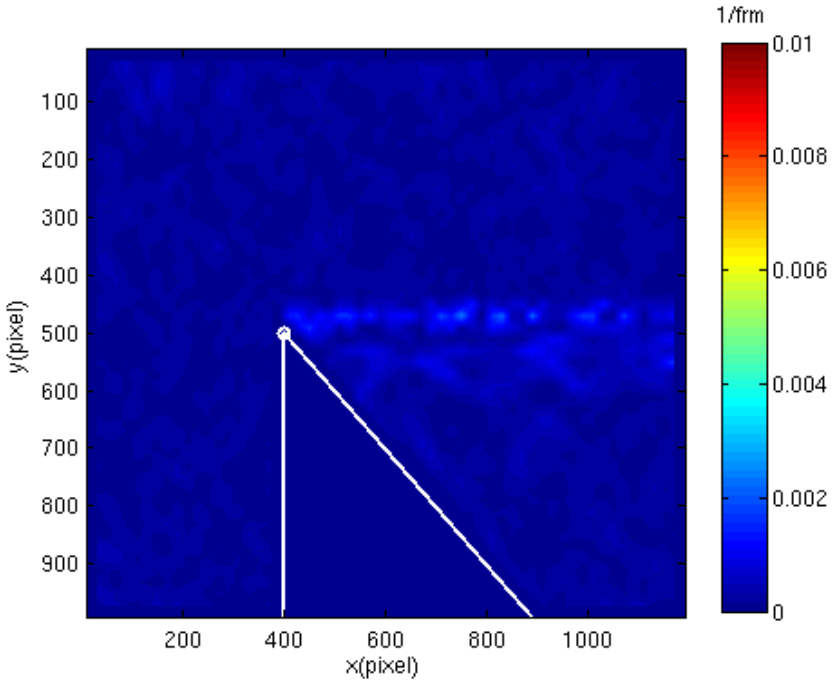


Figure 4.11: Residual strain rate field for the standard calibration images

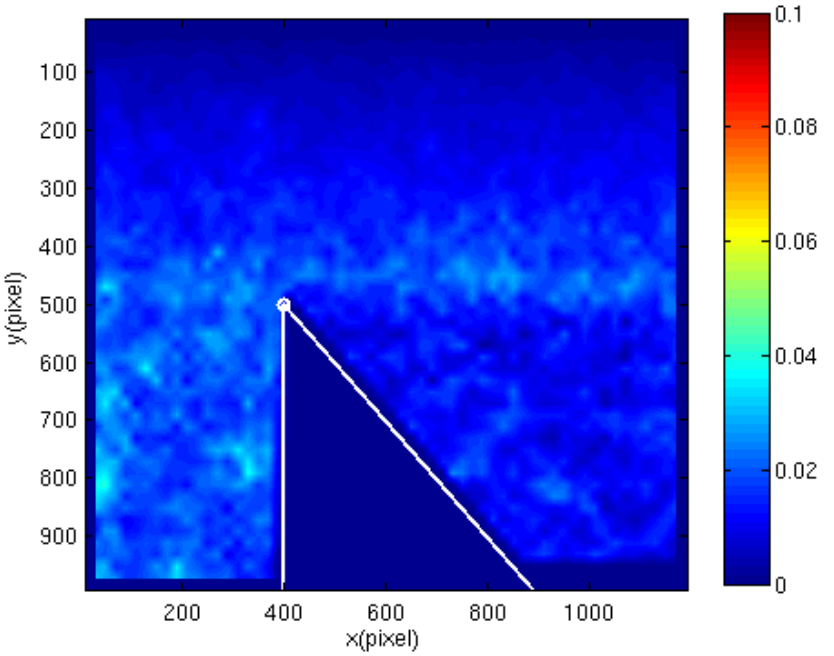


Figure 4.12: Residual strain field for the standard calibration images

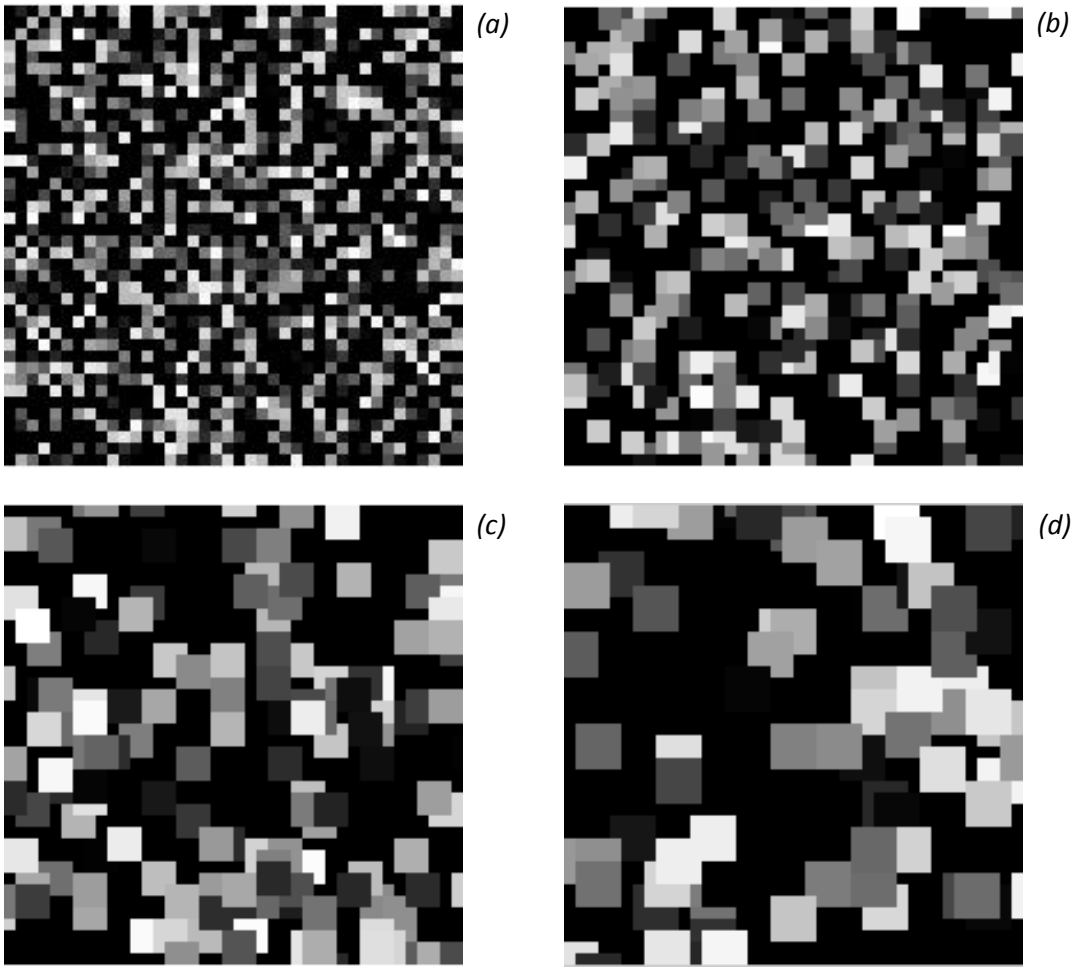


Figure 4.13: Calibration images with different particle sizes: (a) 1 px x 1 px, (b) 2 px x 2 px, (c) 3 px x 3 px, (d) 4 px x 4 px

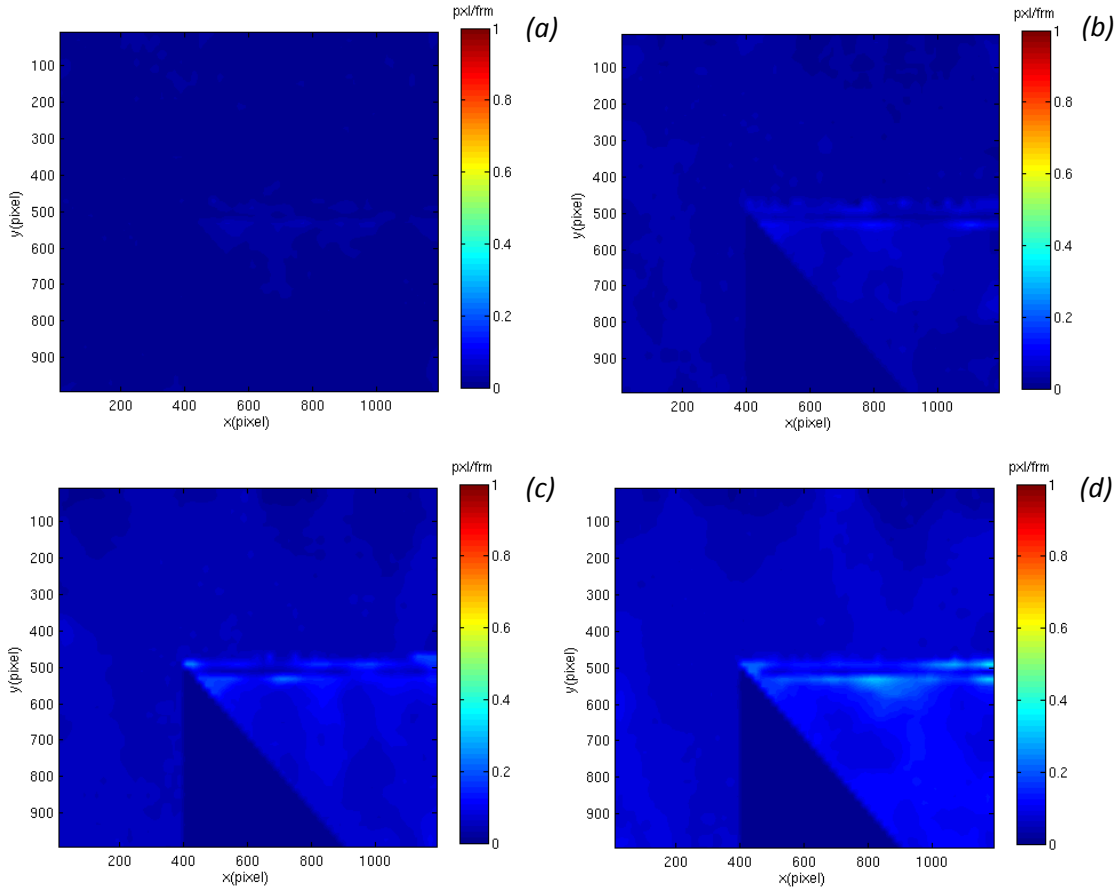


Figure 4.14: Residual velocity fields by PIV for the calibration images with different particle sizes: (a) 1 px x 1 px, (b) 2 px x 2 px, (c) 3 px x 3 px, (d) 4 px x 4 px

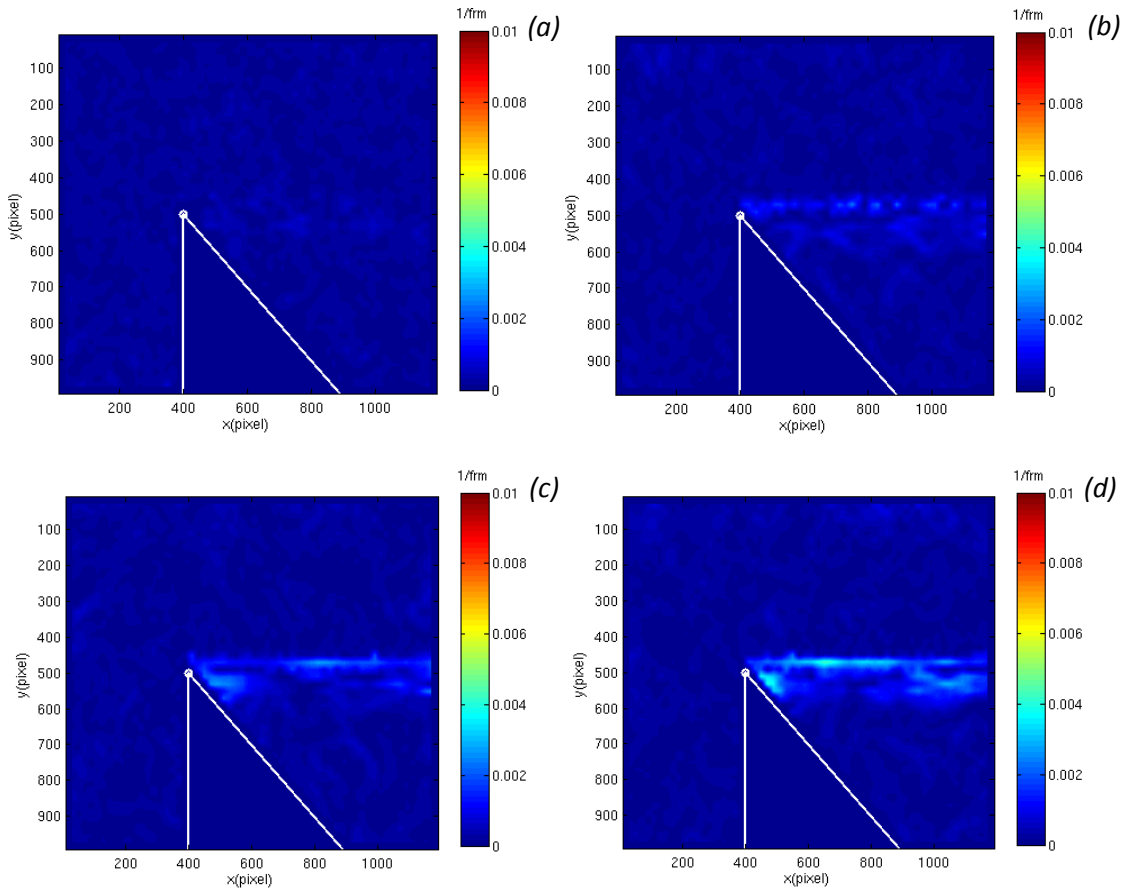


Figure 4.15: Residual strain rate fields by PIV for the calibration images with different particle sizes: (a) 1 px x 1 px, (b) 2 px x 2 px, (c) 3 px x 3 px, (d) 4 px x 4 px

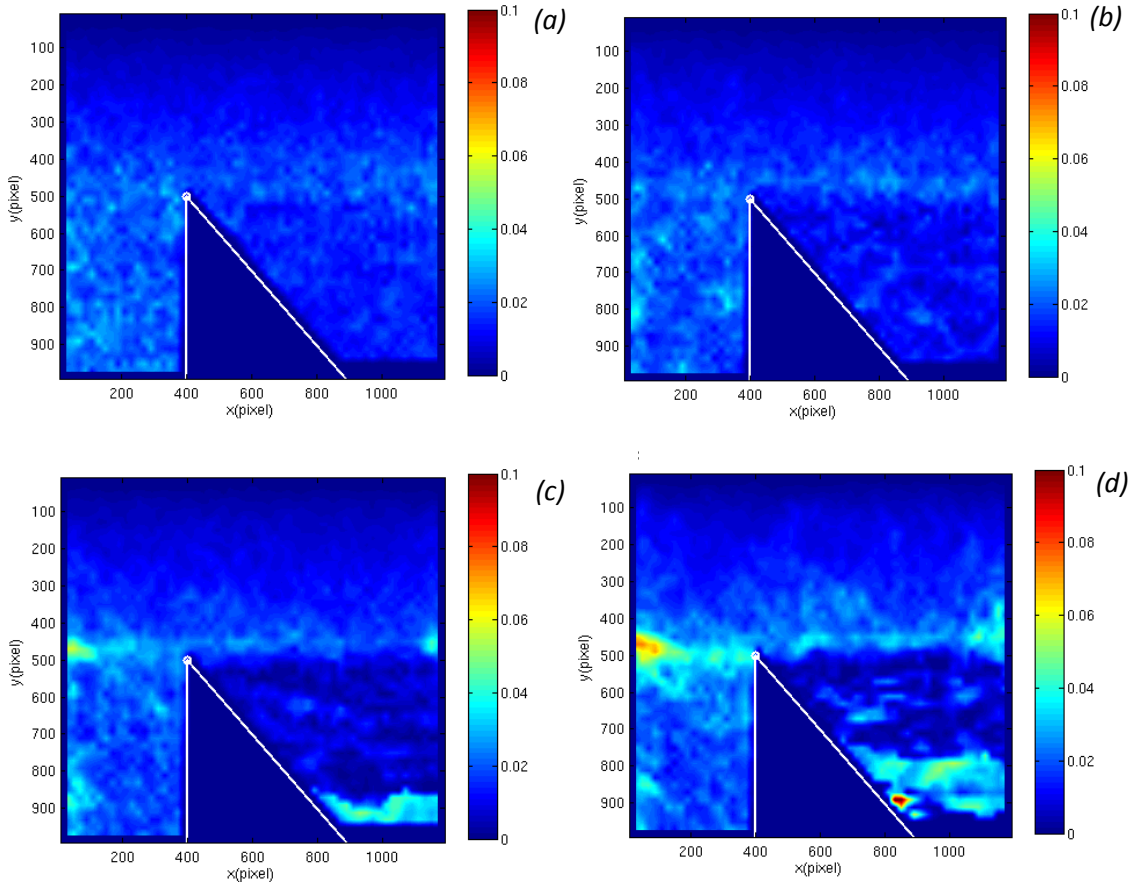


Figure 4.16: Residual strain fields by PIV for the calibration images with different particle sizes: (a) 1 px x 1 px, (b) 2 px x 2 px, (c) 3 px x 3 px, (d) 4 px x 4 px

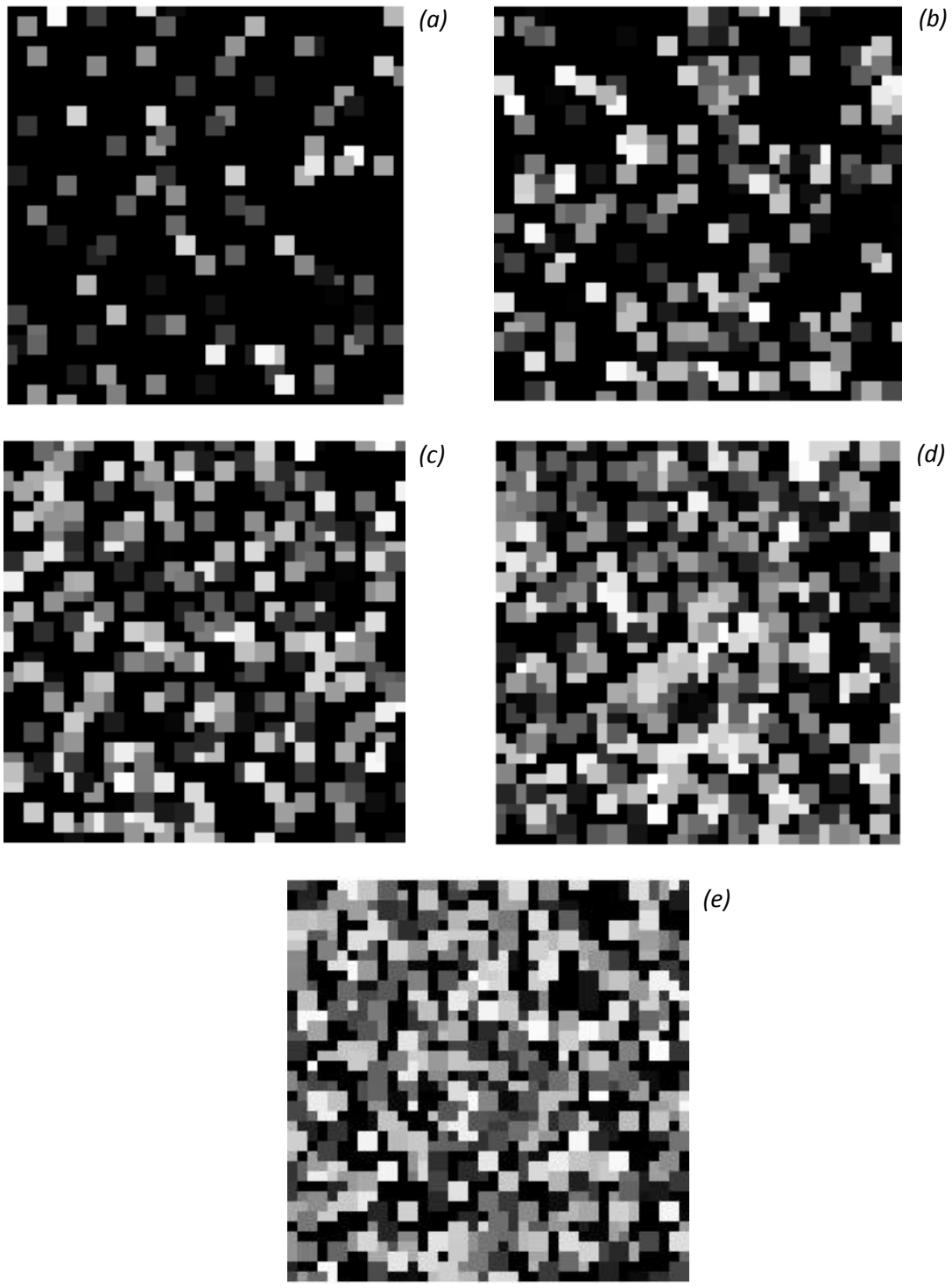


Figure 4.17: Calibration images with different particle densities: (a) 20%, (b) 35%, (c) 50%, (d) 65%, (e) 80%

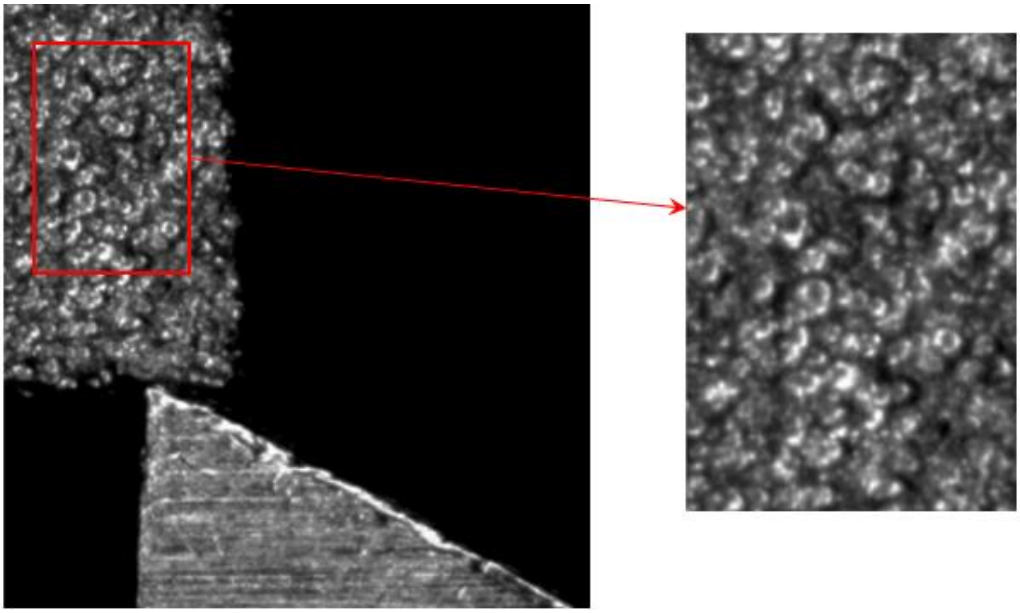


Figure 4.18: Segment used for optimizing the program parameters

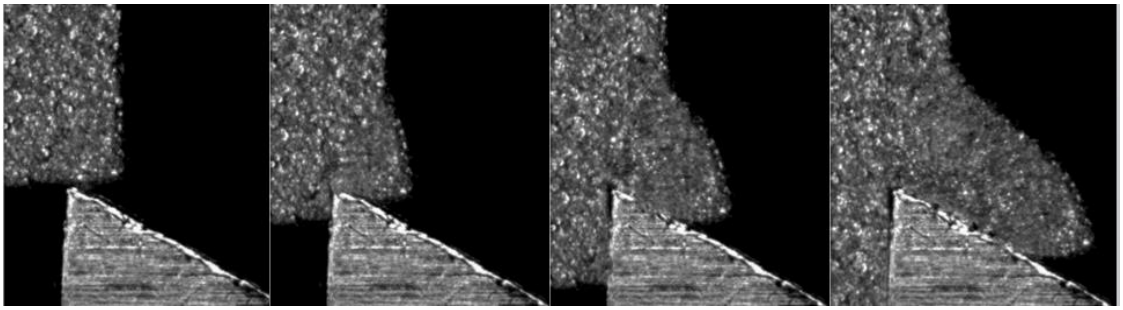


Figure 4.19: An image sequence taken during orthogonal cutting under conditions of ($\alpha = 30°, a_0 = 500\mu m$)

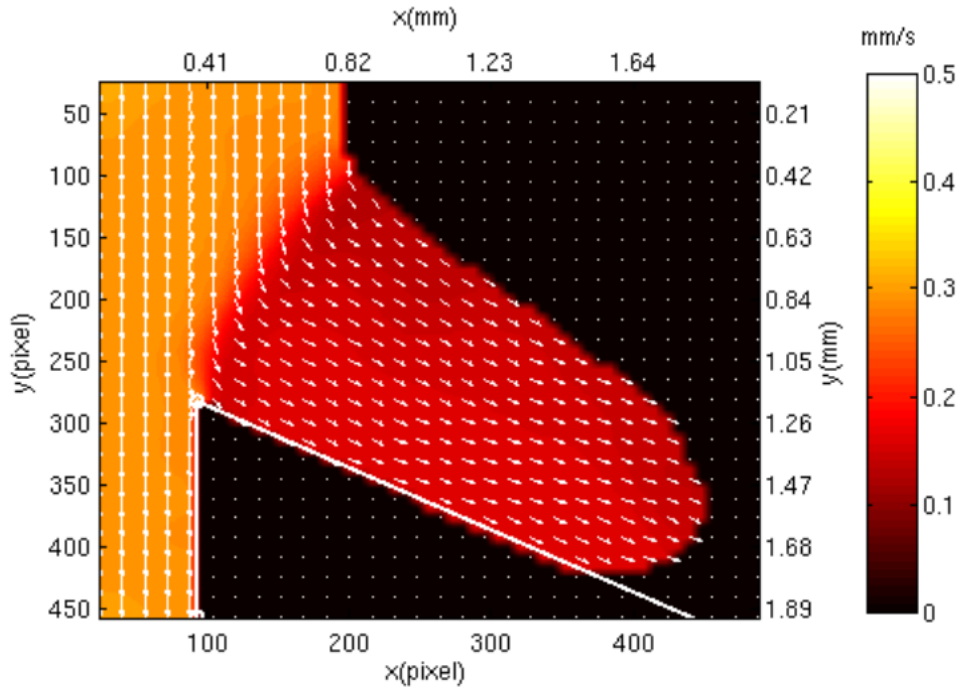


Figure 4.20: A steady velocity field by PIV: $\alpha = 30°, a_0 = 500\mu m$

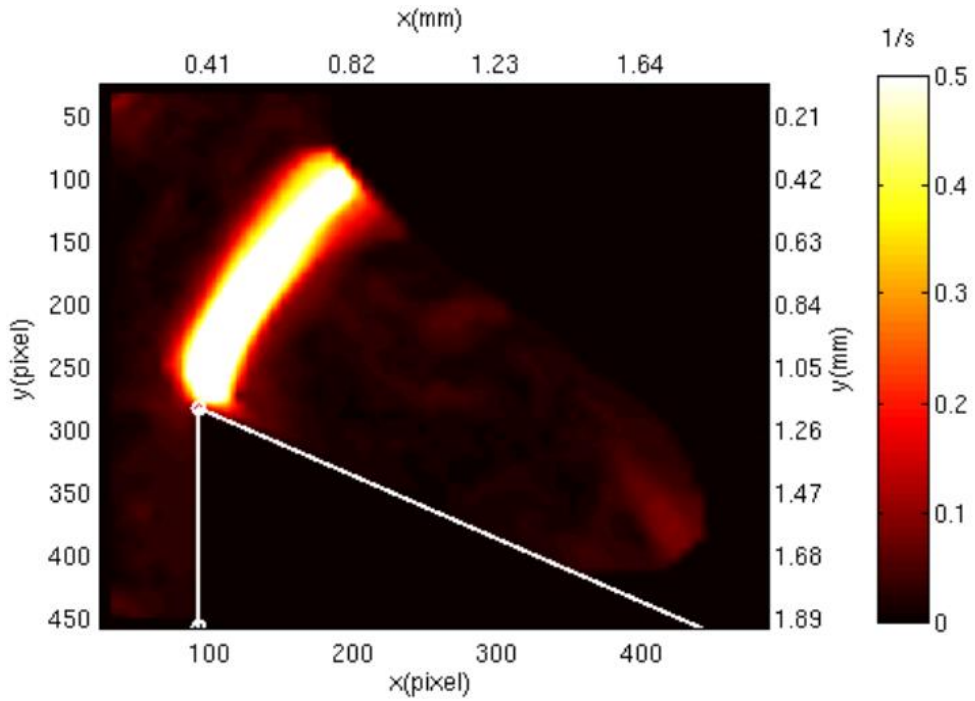


Figure 4.21: A steady strain rate field by PIV: $\alpha = 30°, a_0 = 500\mu m$

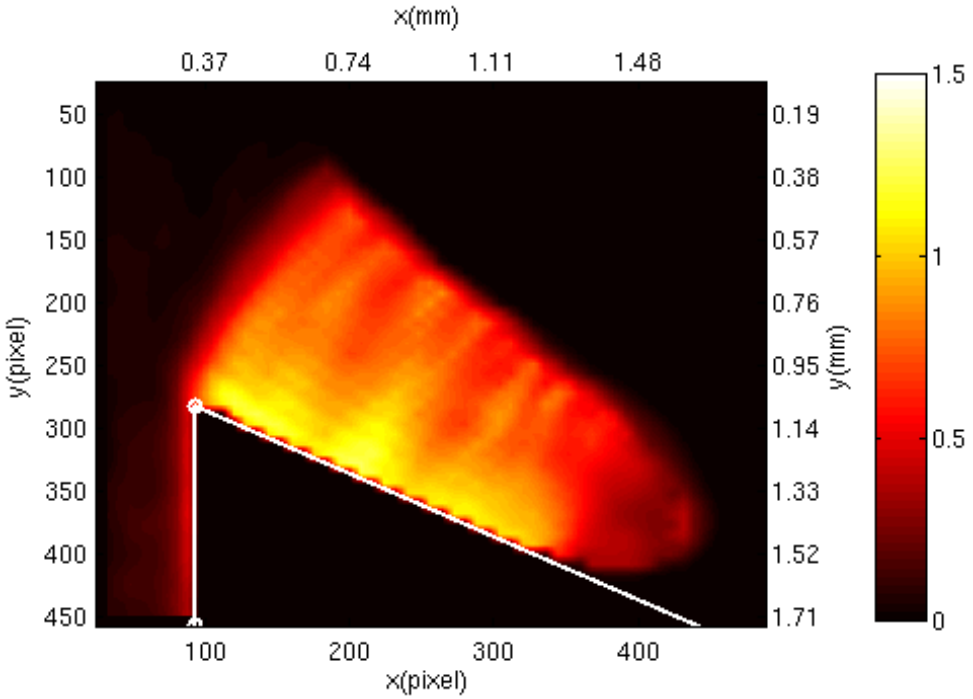


Figure 4.22: Effective strain field by PIV: $\alpha = 30°, a_0 = 500\mu m$

Table 4.1: Comparison between PIV results and expected results for the standard calibration images

		PIV	Expected	Error
u_w (px/fr)	Mean	0.21xE-4	0	
	Var.	0.78xE-4		
v_w (px/fr)	Mean	3.9863	4	0.34%
	Var.	1.48xE-4		
u_d (px/fr)	Mean	1.9576	2	2.12%
	Var.	8.82xE-4		
v_d (px/fr)	Mean	3.9875	4	0.31%
	Var.	0.32xE-4		
u_c (px/fr)	Mean	3.9612	4	0.97%
	Var.	3.09xE-4		
v_c (px/fr)	Mean	3.9894	4	0.27%
	Var.	0.28xE-4		
$\dot{\varepsilon}_w$ (fr^{-1})	Mean	1.86xE-4	0	
	Var.	1.69xE-8		
$\dot{\varepsilon}_d$ (fr^{-1})	Mean	0.02836	0.0289	1.87%
	Var.	34.51xE-8		
$\dot{\varepsilon}_c$ (fr^{-1})	Mean	2.14xE-4	0	
	Var.	2.64xE-8		
$\bar{\varepsilon}_c$	Mean	0.5892	0.578	1.94%
	Var.	16.0xE-6		

Table 4.2: Comparison of the PIV results obtained from the calibration images with different particle sizes

		1x1	2x2	3x3	4x4	Expected
u_w (px/fr)	Mean	0.78xE-4	0.21xE-4	-7.30xE-4	-5.16xE-4	0
	Var.	0.50xE-4	0.78xE-4	1.05xE-4	1.72xE-4	
v_w (px/fr)	Mean	4.0002	3.9863	3.9768	3.9655	4
	Error	0.01%	0.34%	0.58%	0.86%	
	Var.	0.28xE-4	1.48xE-4	2.90xE-4	6.73xE-4	
u_d (px/fr)	Mean	1.9904	1.9576	1.9247	1.8889	2
	Error	0.48%	2.12%	3.77%	5.56%	
	Var.	3.07xE-4	8.82xE-4	19.97xE-4	70.19xE-4	
v_d (px/fr)	Mean	3.9998	3.9875	3.9749	3.9608	4
	Error	0.01%	0.31%	0.63%	0.98%	
	Var.	0.20xE-4	0.32xE-4	0.33xE-4	0.45xE-4	
u_c (px/fr)	Mean	3.9946	3.9612	3.9332	3.9025	4
	Error	0.14%	0.97%	1.67%	2.44%	
	Var.	1.14xE-4	3.09xE-4	6.91xE-4	29.78xE-4	
v_c (px/fr)	Mean	3.9995	3.9894	3.9778	3.9669	4
	Error	0.01%	0.27%	0.56%	0.83%	
	Var.	0.17xE-4	0.28xE-4	0.30xE-4	0.49xE-4	
$\dot{\bar{\varepsilon}}_w$ (fr^{-1})	Mean	1.85xE-4	1.86xE-4	1.91xE-4	2.25xE-4	0
	Var.	0.87xE-8	1.69xE-8	1.69xE-8	2.97xE-8	
$\dot{\bar{\varepsilon}}_d$ (fr^{-1})	Mean	0.02870	0.02836	0.02802	0.02757	0.0289
	Error	0.69%	1.87%	3.04%	4.60%	
	Var.	9.12xE-8	34.51xE-8	56.60xE-8	275.4xE-8	
$\dot{\bar{\varepsilon}}_c$ (fr^{-1})	Mean	1.80xE-4	2.14xE-4	2.54xE-4	3.27xE-4	0
	Var.	1.19xE-8	2.64xE-8	5.63xE-8	37.33xE-8	
$\bar{\varepsilon}_c$	Mean	0.5886	0.5892	0.5896	0.5904	0.578
	Error	1.83%	1.94%	2.01%	2.15%	
	Var.	8.99xE-6	16.0xE-6	121xE-6	377xE-6	

Table 4.3: Comparison of the PIV results obtained from the calibration images with different particle densities

		20%	35%	50%	65%	80%	Expected
u_w (px/fr)	Mean	-0.91xE-4	-5.5xE-4	0.21xE-4	0.02xE-4	4.00xE-4	0
	Var.	1.02xE-4	0.83xE-4	0.78xE-4	0.64xE-4	0.62xE-4	
v_w (px/fr)	Mean	3.9870	3.9852	3.9863	3.9882	3.9899	4
	Error	0.33%	0.37%	0.34%	0.30%	0.25%	
	Var.	1.78xE-4	1.62xE-4	1.48xE-4	1.11xE-4	0.87xE-4	
u_d (px/fr)	Mean	1.9473	1.9498	1.9576	1.9633	1.9678	2
	Error	2.64%	2.51%	2.12%	1.84%	1.61%	
	Var.	32.4xE-4	12.7xE-4	8.82xE-4	6.64xE-4	5.69xE-4	
v_d (px/fr)	Mean	3.9878	3.9857	3.9875	3.9889	3.9906	4
	Error	0.31%	0.36%	0.31%	0.28%	0.24%	
	Var.	0.40xE-4	0.33xE-4	0.32xE-4	0.31xE-4	0.29xE-4	
u_c (px/fr)	Mean	3.9503	3.9566	3.9612	3.9700	3.9724	4
	Error	1.24%	1.09%	0.97%	0.75%	0.69%	
	Var.	10.90xE-4	4.53xE-4	3.09xE-4	2.58xE-4	2.05xE-4	
v_c (px/fr)	Mean	3.9891	3.9869	3.9894	3.9904	3.9920	4
	Error	0.27%	0.33%	0.27%	0.24%	0.20%	
	Var.	0.32xE-4	0.28xE-4	0.28xE-4	0.27xE-4	0.27xE-4	
$\dot{\bar{\varepsilon}}_w$ (fr^{-1})	Mean	1.88xE-4	1.89xE-4	1.86xE-4	1.84xE-4	1.83xE-4	0
	Var.	3.51xE-8	2.07xE-4	1.69xE-8	1.38xE-8	1.23xE-8	
$\dot{\bar{\varepsilon}}_d$ (fr^{-1})	Mean	0.02811	0.02828	0.02836	0.02844	0.02849	0.0289
	Error	2.77%	2.15%	1.87%	1.59%	1.42%	
	Var.	128.6xE-8	49.9xE-8	34.5xE-8	26.4xE-8	22.3xE-8	
$\dot{\bar{\varepsilon}}_c$ (fr^{-1})	Mean	2.67xE-4	2.25xE-4	2.14xE-4	2.07xE-4	2.00xE-4	0
	Var.	12.86xE-8	3.66xE-8	2.64xE-8	2.20xE-8	1.83xE-8	
$\bar{\varepsilon}_c$	Mean	0.5886	0.5902	0.5892	0.5898	0.5889	0.578
	Error	1.66%	2.11%	1.94%	2.04%	1.89%	
	Var.	64.5xE-6	21.2xE-6	16.0xE-6	14.6xE-6	12.5xE-6	

Table 4.4: Comparison of the PIV results obtained from the calibration images with different inter-frame displacements

		$v = 2$	$v = 4$	$v = 6$	$v = 8$
u_w (px/fr)	Mean	0.05xE-4	0.21xE-4	-7.71xE-4	3.60xE-4
	Var.	0.42xE-4	0.78xE-4	1.21xE-4	1.77xE-4
v_w (px/fr)	Mean	1.9925	3.9863	5.9686	7.9571
	Error	0.38%	0.34%	0.52%	0.54%
	Var.	0.27xE-4	1.48xE-4	0.48xE-4	0.73xE-4
u_d (px/fr)	Mean	0.9794	1.9576	2.9286	3.9011
	Error	2.06%	2.12%	2.38%	2.47%
	Var.	4.31xE-4	8.82xE-4	15.49xE-4	64.54xE-4
v_d (px/fr)	Mean	1.9936	3.9875	5.9697	7.9586
	Error	0.32%	0.31%	0.51%	0.52%
	Var.	0.27xE-4	0.32xE-4	0.19xE-4	0.27xE-4
u_c (px/fr)	Mean	1.9809	3.9612	5.9405	7.9051
	Error	0.96%	0.97%	0.99%	1.19%
	Var.	1.39xE-4	3.09xE-4	9.43xE-4	41.5xE-4
v_c (px/fr)	Mean	1.9938	3.9894	5.9730	7.9607
	Error	0.31%	0.27%	0.45%	0.49%
	Var.	0.35xE-4	0.28xE-4	0.26xE-4	50.49xE-4
$\dot{\varepsilon}_w$ (fr^{-1})	Mean	1.75xE-4	1.86xE-4	1.88xE-4	2.02xE-4
	Var.	0.82xE-8	1.69xE-8	1.72xE-8	2.48xE-8
$\dot{\varepsilon}_d$ (fr^{-1})	Mean	0.01413	0.02836	0.04231	0.05582
	(Exp.)	(0.0144)	(0.0289)	(0.0433)	(0.0577)
	Error	1.88%	1.87%	2.29%	3.26%
	Var.	13.73xE-8	34.51xE-8	62.72xE-8	2.2xE-5
$\dot{\varepsilon}_c$ (fr^{-1})	Mean	2.10xE-4	2.14xE-4	2.51xE-4	5.60xE-4
	Var.	1.75xE-8	2.64xE-8	10.93xE-8	484xE-8
$\bar{\varepsilon}_c$	Mean	0.5857	0.5892	0.5896	0.5885
	(Exp.)	(0.578)	(0.578)	(0.578)	(0.578)
	Error	1.33%	1.94%	2.01%	1.82%
	Var.	15.3xE-6	16.0xE-6	130xE-6	311xE-6

Table 4.5: Comparison of the PIV results obtained under different window sizes

		12	16	20	Expected
u (px/fr)	Mean	-21.57xE-4	-73.85xE-4	-64.77xE-4	0
	Var.	1.27xE-4	0.96xE-4	1.01xE-4	
v (px/fr)	Mean	1.4878	1.62	1.7011	2
	Error	25.61%	19.00%	14.95%	
	Var.	68.51xE-4	23.78xE-4	26.53xE-4	
		24	28	32	Expected
u (px/fr)	Mean	-73.68xE-4	-27.53xE-4	-22.84xE-4	0
	Var.	1.06xE-4	0.77xE-4	0.68xE-4	
v (px/fr)	Mean	1.7395	1.7759	1.8095	2
	Error	13.03%	11.21%	9.53%	
	Var.	13.19xE-4	12.82xE-4	14.39xE-4	

Table 4.6: Comparison of the PIV results obtained under different grid sizes

		6	8	10	Expected
u (px/fr)	Mean	-17.44xE-4	-22.44xE-4	-22.84xE-4	0
	Var.	0.70xE-4	0.78xE-4	0.68xE-4	
v (px/fr)	Mean	1.8075	1.8048	1.8095	2
	Error	9.63%	9.76%	9.53%	
	Var.	16.32xE-4	12.81xE-4	14.39xE-4	
$\dot{\varepsilon}$ (fr^{-1})	Mean	10.72xE-4	7.99xE-4	6.42xE-4	0
	Var.	109.26xE-8	58.20xE-8	33.49xE-8	
		12	14	16	Expected
u (px/fr)	Mean	-17.67xE-4	-11.98xE-4	-16.91xE-4	0
	Var.	0.58xE-4	0.47xE-4	0.58xE-4	
v (px/fr)	Mean	1.814	1.820	1.812	2
	Error	9.30%	9.10%	9.40%	
	Var.	8.15xE-4	4.40xE-4	10.03xE-4	
$\dot{\varepsilon}$ (fr^{-1})	Mean	5.20xE-4	3.80xE-4	4.70xE-4	0
	Var.	19.48xE-4	7.62xE-4	12.07xE-8	

Chapter 5: Results

For each orthogonal cutting experiment performed on the uniaxial testing machine (depth of cut ~ 750, 500, 250μm), the deformation in chip formation was studied using PIV, micro-hardness mapping and load measurement. For the experiments carried out on the linear slide (depth of cut ~ 200, 150, 100μm) PIV was not used. The results of these investigations are summarized in this chapter.

5.1 PIV Results

The resulting strain field for cutting of copper under cutting conditions of ($\alpha = 30°, a_0 = 500\mu m$) is shown in Fig. 5.1. The strain profile as a function of distance (X) from the chip end was gathered from these fields and is shown in Figs. 5.1 and 5.2. These strain values were obtained along a contour parallel to the chip rake face and set at a distance of 150 μm away. The distance was taken as ($\mathrm{X}/\cos\alpha$) to facilitate direct comparison with the load measurements. To derive an expression that can be used to describe the relationship between strain and distance, one first must consider the general shape of the data. Fig. 5.2 shows that the relationship between strain and distance exhibits a sigmoidal-type relationship. Three different models were considered as candidates for fitting this behavior; these include the Verhulst (Eq. 5.1), Gompertz (Eq. 5.2) and Bertalanffy (Eq. 5.3) models. The general forms of these relationships are as follows:

$$y = a + \frac{b}{1 + e^{c+dx}} \qquad (\mathrm{Eq.}\ 5.1)$$

$$y = a + be^{ce^{dx}} \qquad (\mathrm{Eq.}\ 5.2)$$

$$y = a + b(1 + ce^{dx})^3 \qquad (\mathrm{Eq.}\ 5.3)$$

The fits according to the different models are shown in Table 5.1, and the fit curves are shown in Fig. 5.3 with the original data. The fit results for the three models were all quite satisfactory ($R^2 > 0.98$), but the correlation coefficient of the Verhulst model (Eq.

5.1) was the highest at R^2 =0.9827. Thus, this general model was used to fit the subsequent PIV data for all conditions. Because the data exhibited a sigmoidal-type relationship, the response function plateaus at increasing values of distance. For the purpose of the present study, the plateau was identified as the length corresponding to 95% of the maximum value.

The effective strain fields by PIV for the remaining experiments exhibited similar characteristics as that of Fig. 5.1 and are provided in Figs. 5.4–5.10. In each of these fields, the initial portion of chip formation exhibits a low strain value and is followed by a transition where the change of strain is most rapid. As the strain distribution and the chip thickness approach stability, the chip formation enters steady state. From these results, it is clear that smaller rake angles result in higher steady strain levels and larger chip thickness ratios. The actual strain measurements for the other experimental conditions are provided in Appendix H. The Verhulst model fits and the actual strain measurements are shown in Figs. 5.11–5.13 and the detailed data are provided in Tables 5.2–5.4. As was the case earlier, high correlation coefficients are evident with R^2>0.98 for all of the experimental conditions. The plateau values for the strain profiles obtained using the PIV method can be compared to the upper bound solution for the deformation strain according to Eq. 2.4. The velocity field information for each condition was used to obtain the velocity ratio (V_0/V_c) which is equivalent to the chip thickness ratio (a_c/a_0). The theoretical strain values were calculated and are listed in Table 5.5 and shown as dotted lines in Figs. 5.11–5.13. From the figures, the actual upper bounds of the PIV strain curves are very close to their corresponding theoretical values with differences smaller than 10%. These observations further enhance the reliability of the PIV results.

The PIV measurements were used to determine the length of the incipient stage by using 95% of the maximum value as the threshold condition. The raw data is provided in Tables 5.2-5.4 and are plotted together on Fig. 5.14. From the results, it is clear that the length of the incipient stage increases with depth of cut and also decreases with increasing rake angle. A clear linear trend is exhibited corresponding to constant rake

angle. The linear fits of the data are provided in Table 5.6 and exhibit high R^2 values greater than 0.99.

5.2 Hardness Mapping Results

The evolution of hardness as a function of distance from the chip end was investigated for the range of experimental conditions. Fig. 5.15 shows an optical image of how these impressions were oriented and the resulting measurements under conditions of ($\alpha = 30°, a_0 = 500\mu m$). As is evident from the figure, the hardness increases with increasing distance from the chip end, followed by saturation of the hardness after some threshold length. Thus, it is clear that an incipient region of hardening, indicative of incipient straining, is present for the machined chips. The other experimental conditions exhibited similar trends in hardness and all hardness measurements are provided in Appendix H. A strain-based understanding of the incipient regime can come from these hardness measurements based on empirical relations that establish the correlation between strain and hardness. A relation between effective strain and Vickers hardness in annealed pure copper was established by Chaudhri through controlled compression tests [47] and is given by:

$$HV = 132\bar{\varepsilon}^{0.1968} \qquad \text{(Eq. 5.4)}$$

While this empirical relation is founded on a fundamentally different loading configuration (compression versus shear), it enables a first-pass estimation of an effective strain from hardness measurement. The strain profile obtained using the hardness measurements and Eq. 5.4 for the experimental condition of ($\alpha = 30°, a_0 = 500\mu m$) is shown in Table 5.7 and Fig. 5.16. It is clear that the effective strains, which increase from 0.21 to 1.1 are quite similar to the PIV-based strain measurements which range from 0.27 to 1.1. The average error between the two data sets is less than 10%. To model the relation between hardness-converted strain and distance from the chip end, the same three statistical models were used (Verhulst, Gompertz, Bertanlanffy). All exhibited R^2 values greater than 0.98 as shown in Table 5.8 and Fig. 5.17. As was the case for the PIV data, the Verhulst model (Eq. 5.1) provided the best

performance R^2 = 0.9882 and was thus used to model the data. The hardness-converted strain curves and model fit results for all the experimental conditions are provided in Figs. 5.18–5.20 and Tables 5.9–5.11. These results showed similar behavior to the ($\alpha = 30°, a_0 = 500\mu m$) case, as all exhibited similar shape and high R^2 values greater than 0.98.

To determine the length of incipient straining in the chip using the hardness-converted strain profiles, an identical threshold (e.g., 95% of the maximum value) to that of the earlier PIV measurements was applied. These results are provided in Tables 5.9–5.11. The incipient stage length increases with depth of cut and decreases with rake angle; identical behavior to that of the PIV case. The incipient stage lengths for all the experimental conditions are summarized in Fig. 5.21 and linear fits were applied to those data corresponding to constant rake angle. The corresponding fits are provided in Table 5.12 and the R^2 values are greater than 0.98.

5.3 Cutting Load Analysis Results

The cutting load was recorded during all experiments. The original load data corresponding to the experimental condition of ($\alpha = 30°, a_0 = 500\mu m$) is shown in Fig. 5.22, together with the rate of change of this curve as a function of the feed of the workpiece. Because the displacement (and velocity) of the chip and workpiece are not equal in the presence of plastic straining, the x-axis needs to be converted to the chip-oriented length scale to be directly compatible with the earlier reviewed hardness and PIV measurements. To accomplish this, one simply must divide the x-axis by the velocity ratio (V_0/V_c) obtained from the PIV velocity fields. This transforms the x-axis to the chip-orientation as is shown in Fig. 5.23.

To improve the fidelity of the curve-fitting procedures, two data clean-up tasks were used. First, the initial stages of contact have a discontinuity in the load curve. This is related to the take-up of compliance and alignment of the tool and work at the initial stages of contact. This portion of the load trace was removed from the data analysis. Second, a three-point moving average was applied to the load curve to enable more

robust statistical fitting. The resulting load-curve is shown in Fig. 5.23 for the experimental condition of ($\alpha = 30°, a_0 = 500\mu m$). From the figure, the instantaneous derivative of the load has a shape that can be fitted using a power function, since the change rate of cutting load approaches zero as the chip enters the steady-state plastic regime. This is given in Eq. 5.5, where a, b and c are constants and x is the distance from the chip end.

$$y = a(x + b)^c \qquad \text{(Eq. 5.5)}$$

From the figure, it is clear that the rate of change of the cutting load decreases rapidly with increasing distance from chip end. To identify the length of the incipient loading of the tool, 95% of the maximum rate of change was used (e.g. ~2.5 N/10 μm) as the lower threshold for the function. The actual load-displacement curves for all experiments are provided in Appendix H. The instantaneous slope of these curves, their fits according to Eq. 5.5 and the corresponding incipient stage length estimates are shown in Figs. 5.24–5.26 and Tables 5.13–5.15. The nature of the loading behavior is clearly similar across the range of conditions, with a decreasing rate of change of load as the distance from chip end increases. Further, the length of the incipient stage increases with depth of cut and decreases with rake angle, as was evident in the hardness and PIV measurements. A summary of these measurements is provided in Fig. 5.27, which shows linear dependence of the incipient stage length as a function of depth of cut for multiple rake angles. The fits corresponding to these linear relationships are provided in Table 5.16 and all conditions have R^2 values greater than 0.98. It should be noted that two five-point moving averages (instead of only one three-point moving average) were performed for the linear stage experiment date due to greater noise in the force traces for those experiments.

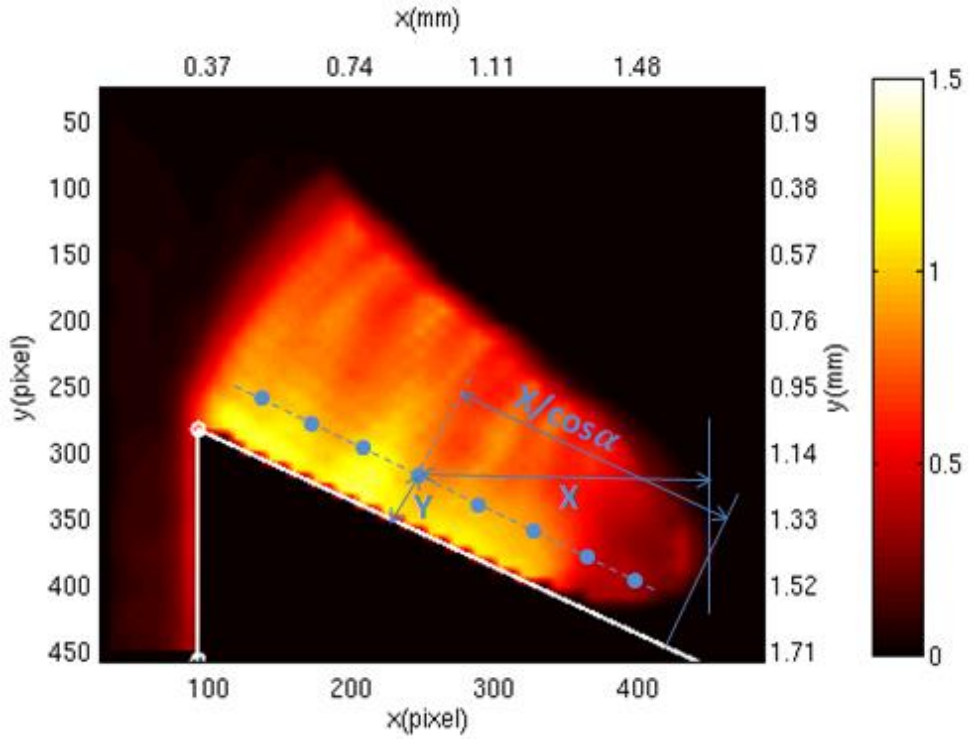


Figure 5.1: Measurement point selection in the effective strain field: $\alpha = 30°, a_0 = 500\mu m$

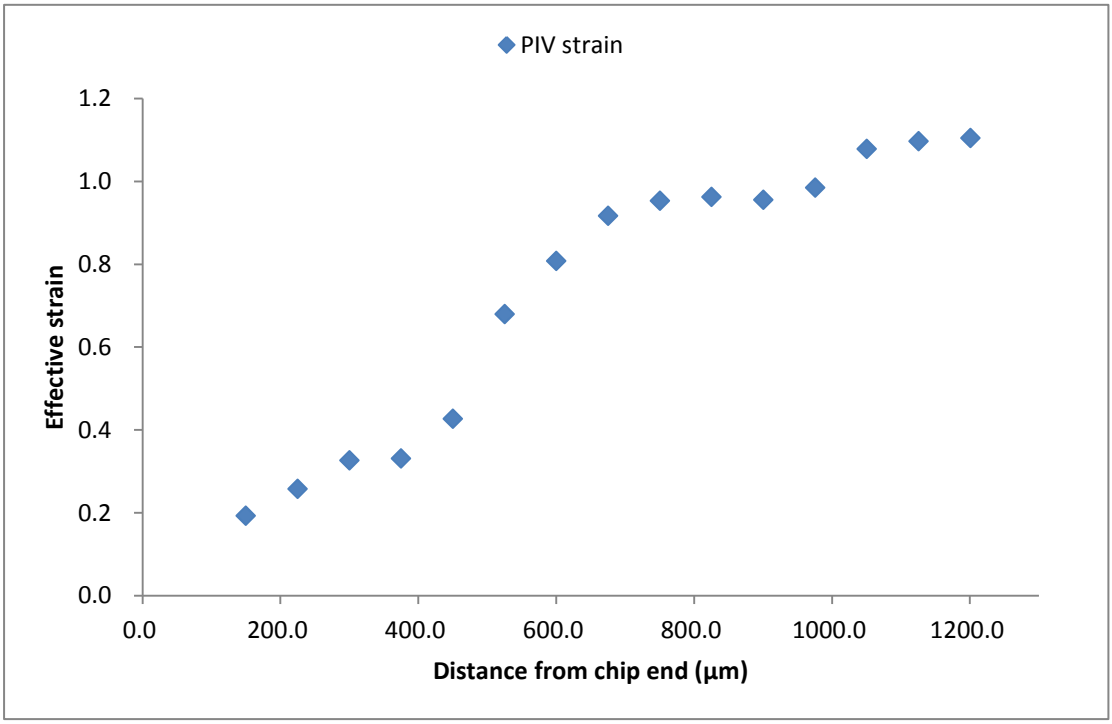


Figure 5.2: Measured strain values from the effective strain field: $\alpha = 30°, a_0 = 500\mu m$

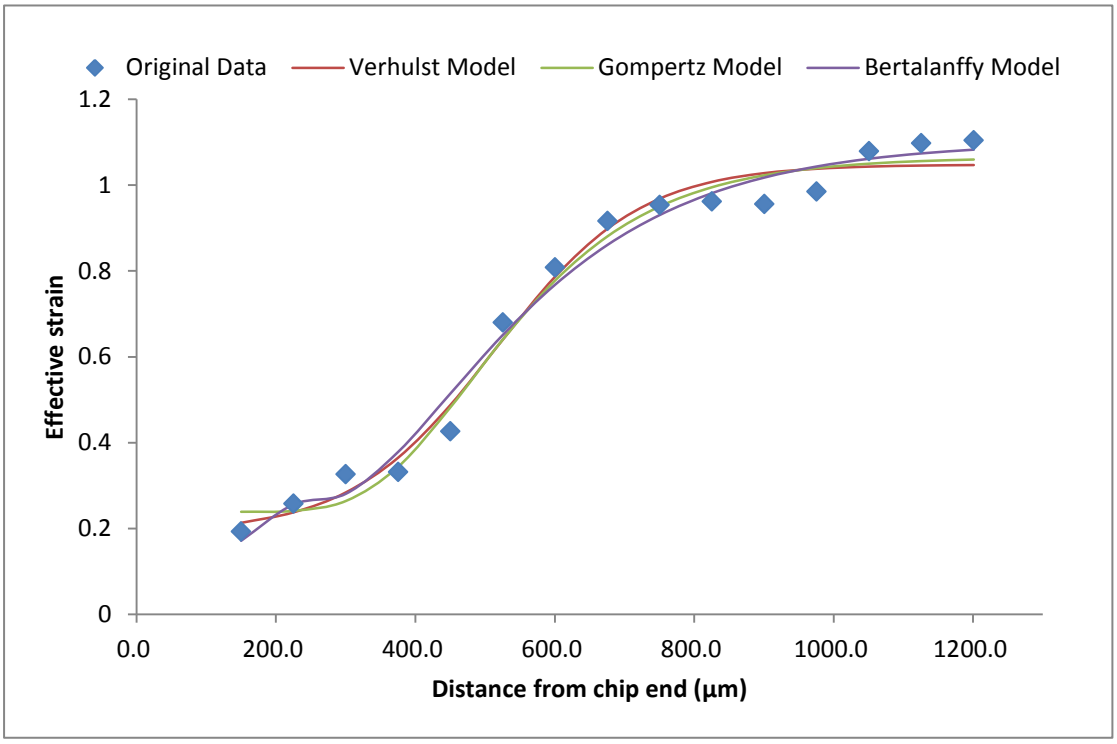


Figure 5.3: Curve fitting for each model based on the strain measurements: $\alpha = 30°, a_0 = 500\mu m$

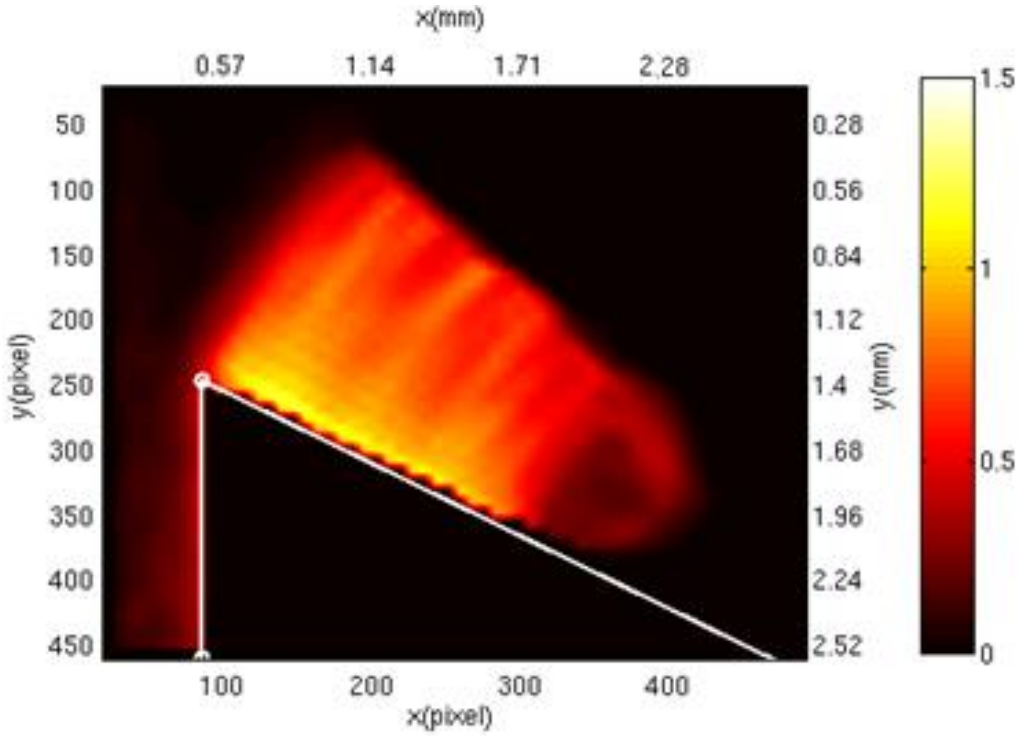


Figure 5.4: Effective PIV strain field: $\alpha = 30°, a_0 = 750\mu m$

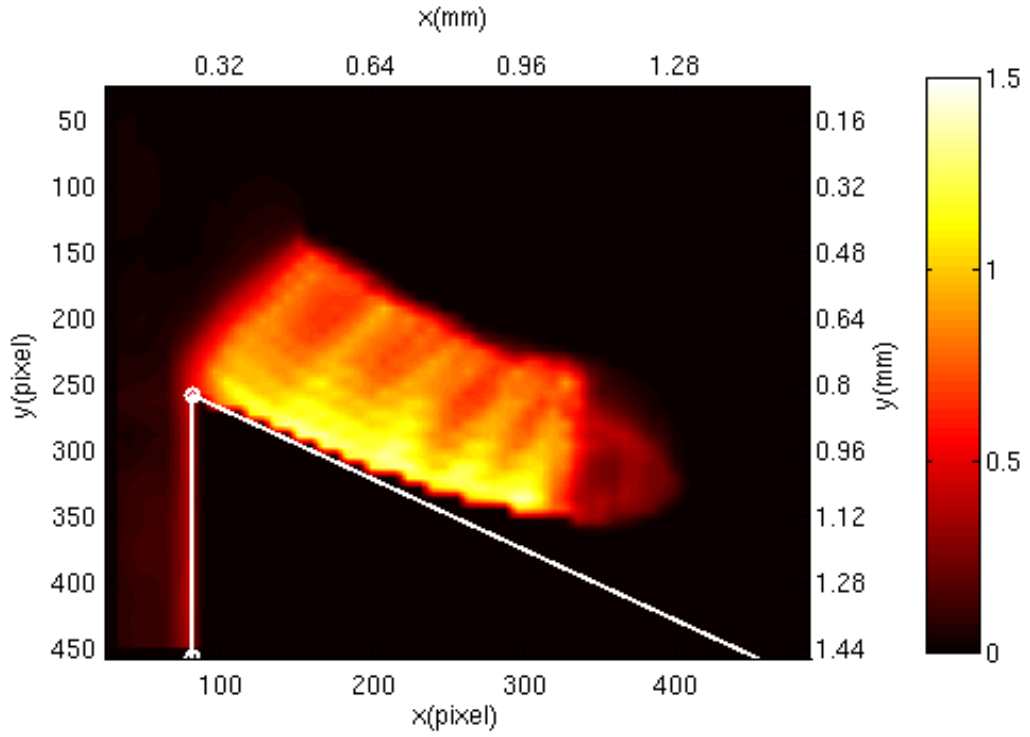


Figure 5.5: Effective PIV strain field: $\alpha = 30°, a_0 = 250\mu m$

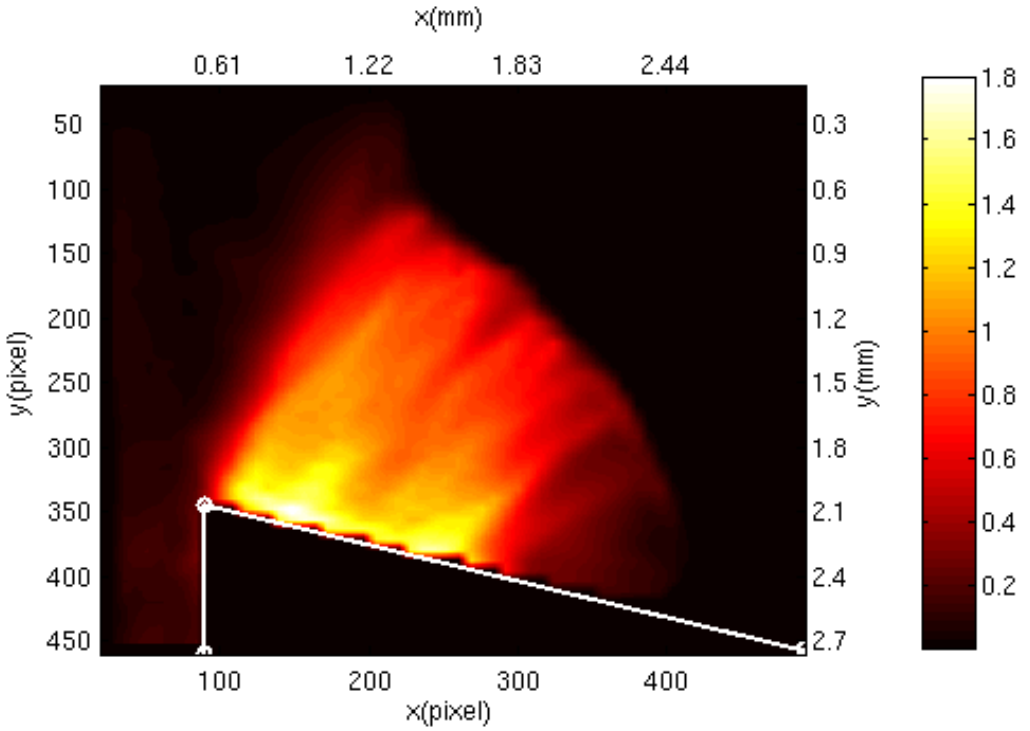


Figure 5.6: Effective PIV strain field: $\alpha = 15°, a_0 = 750\mu m$

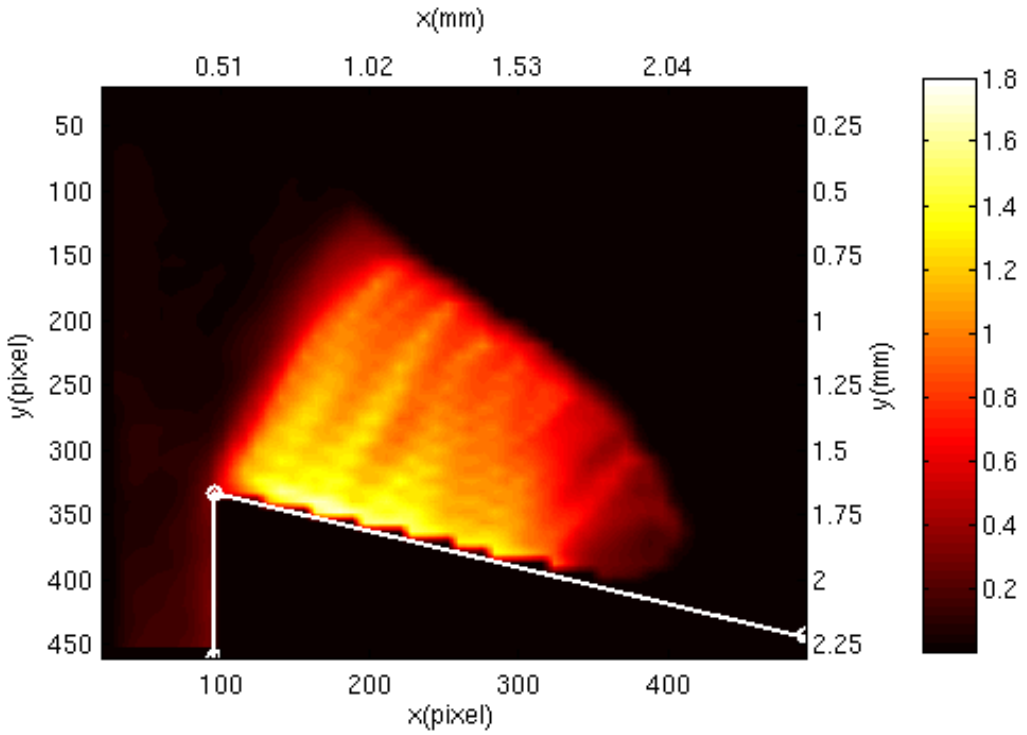


Figure 5.7: Effective PIV strain field: $\alpha = 15°, a_0 = 500\mu m$

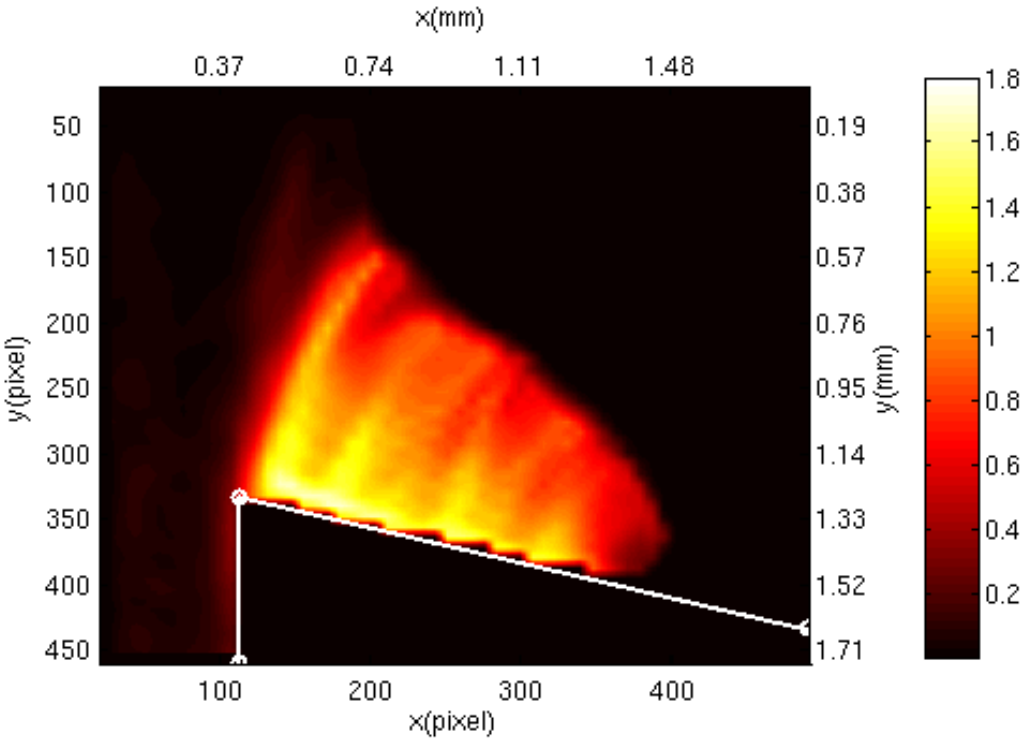


Figure 5.8: Effective PIV strain field: $\alpha = 15°, a_0 = 250\mu m$

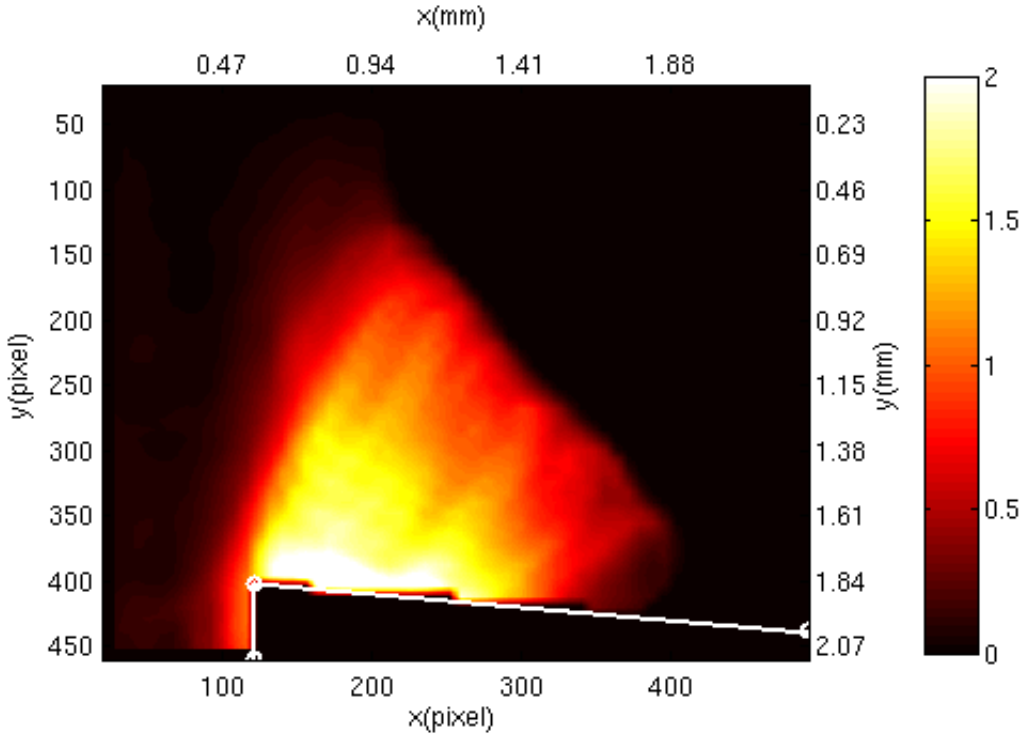


Figure 5.9: Effective PIV strain field: $\alpha = 5°, a_0 = 500\mu m$

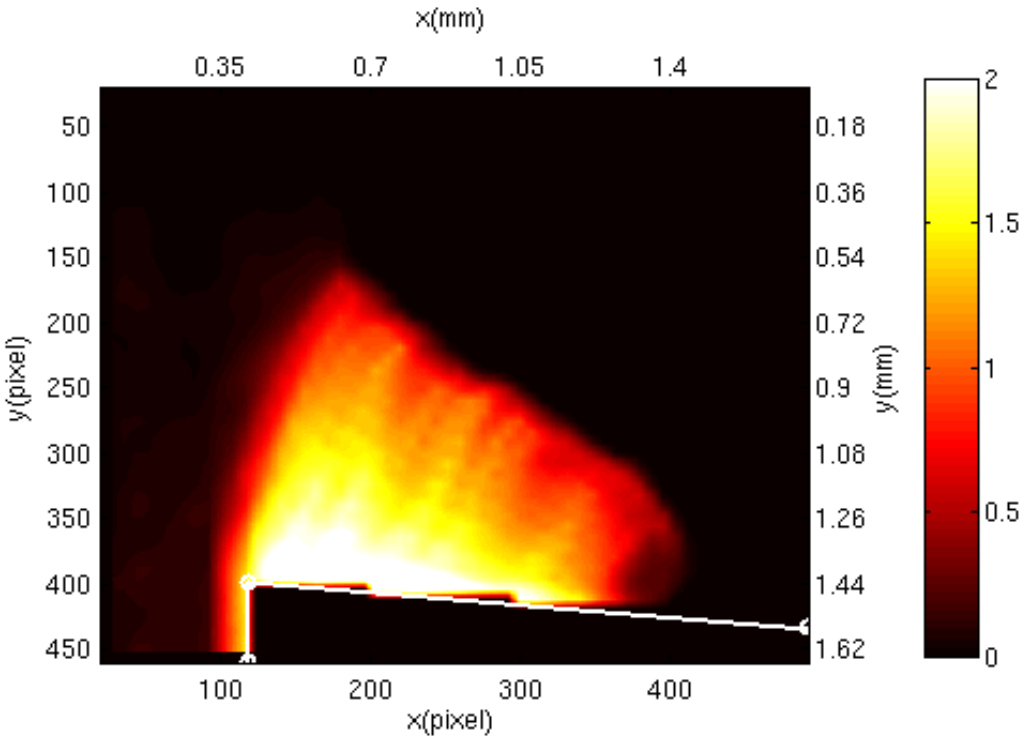


Figure 5.10: Effective PIV strain field: $\alpha = 5°, a_0 = 250\mu m$

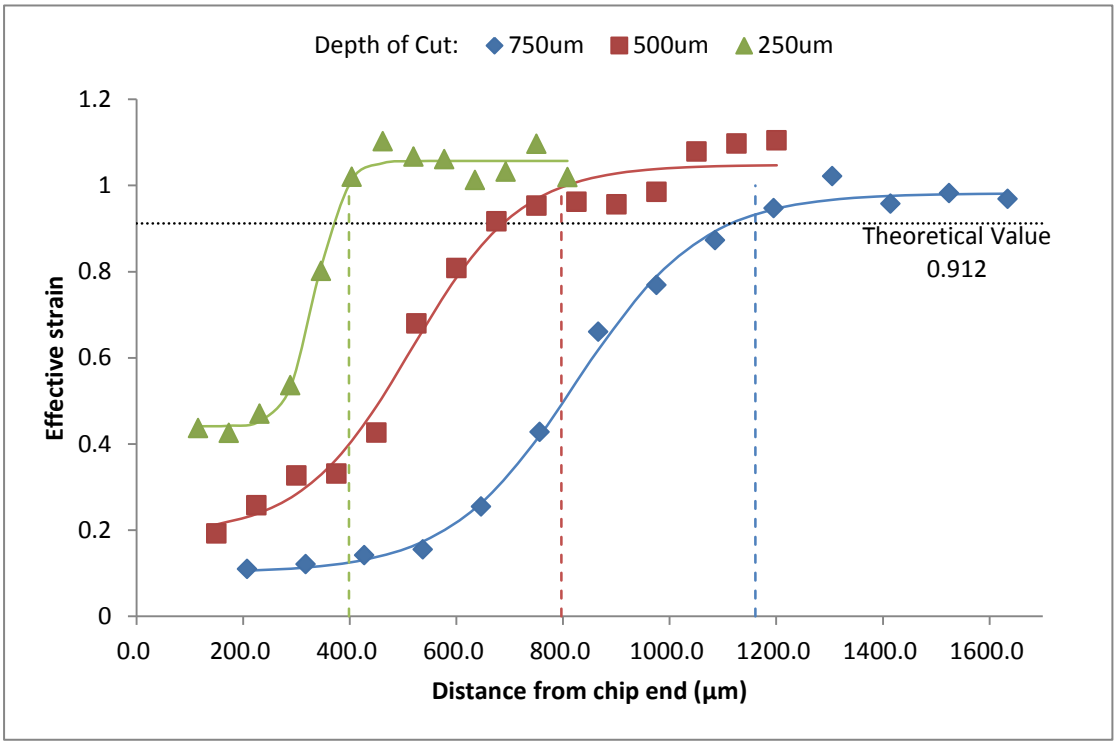


Figure 5.11: Fitting and analysis results of PIV strain curves: $\alpha = 30°$

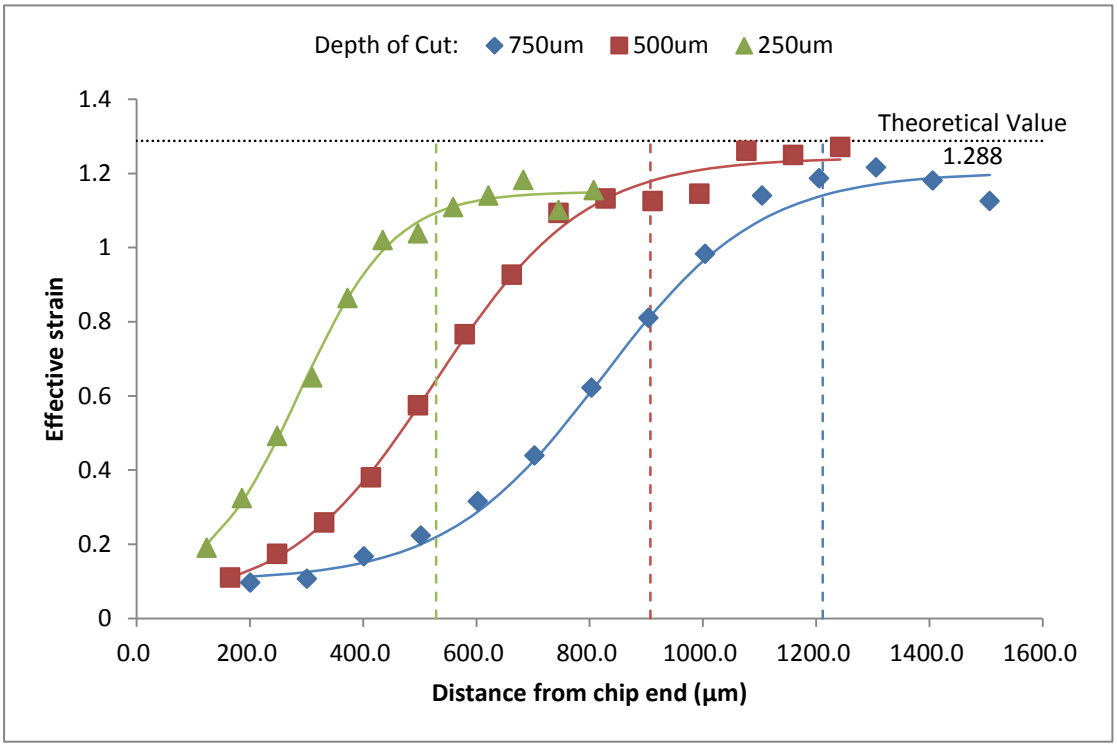


Figure 5.12: Fitting and analysis results of PIV strain curves: $\alpha = 15°$

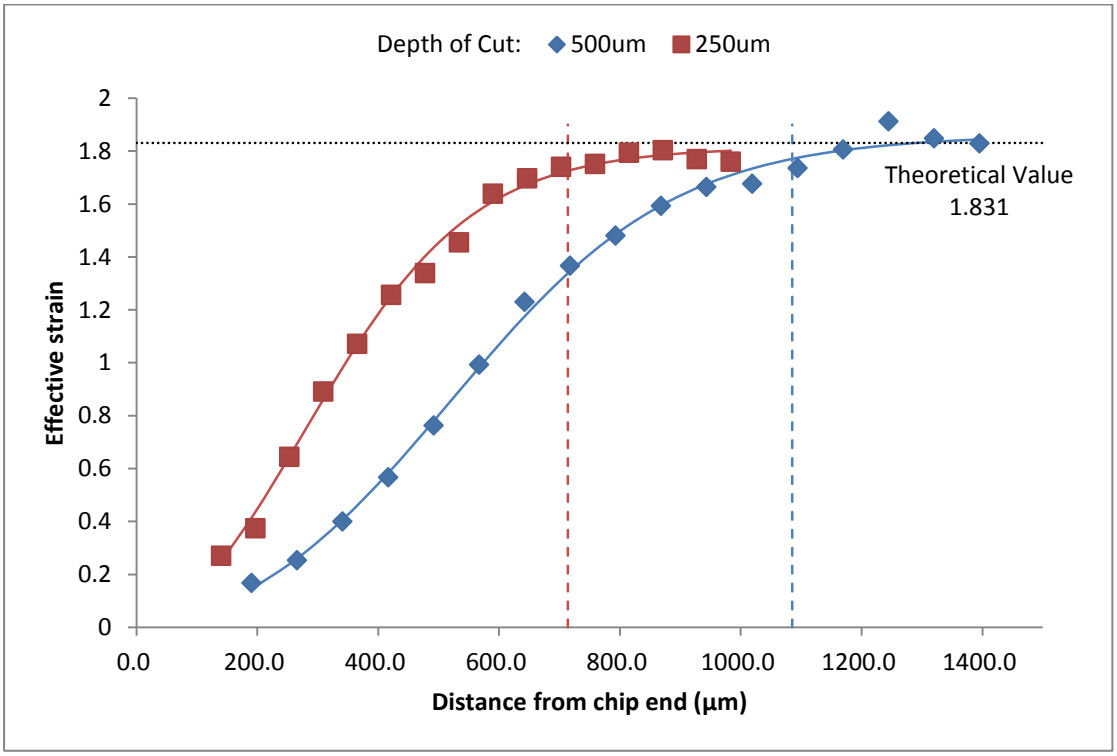


Figure 5.13: Fitting and analysis results of PIV strain curves: $\alpha = 5°$

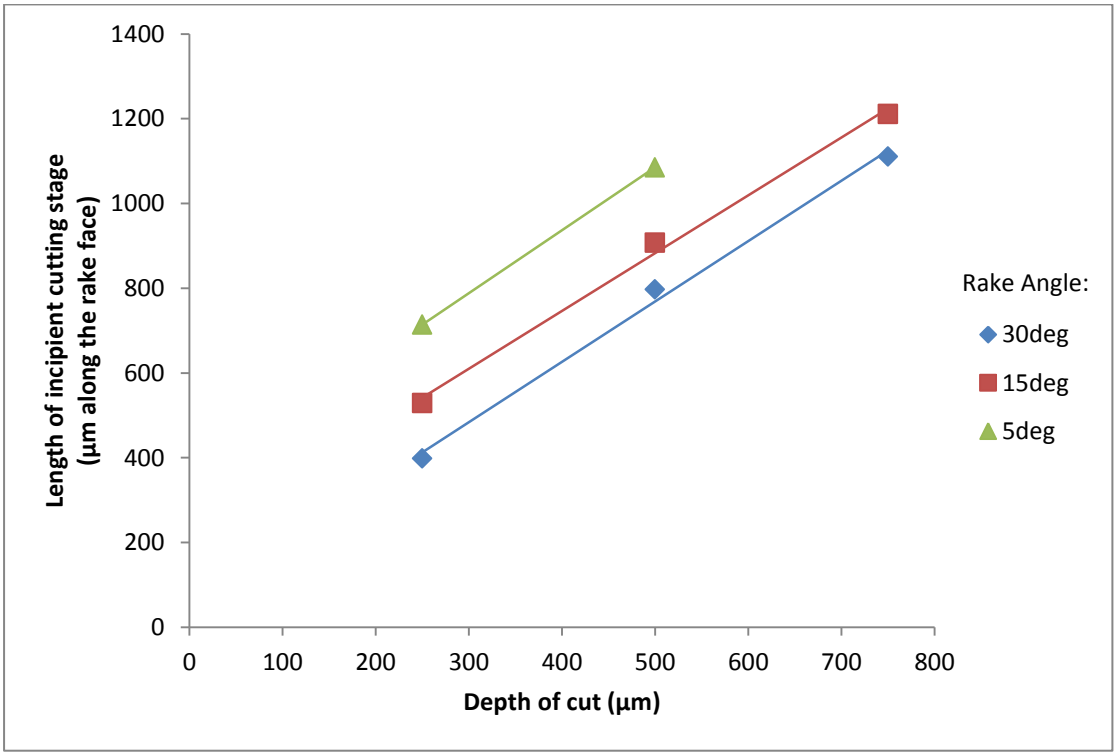


Figure 5.14: PIV results summary

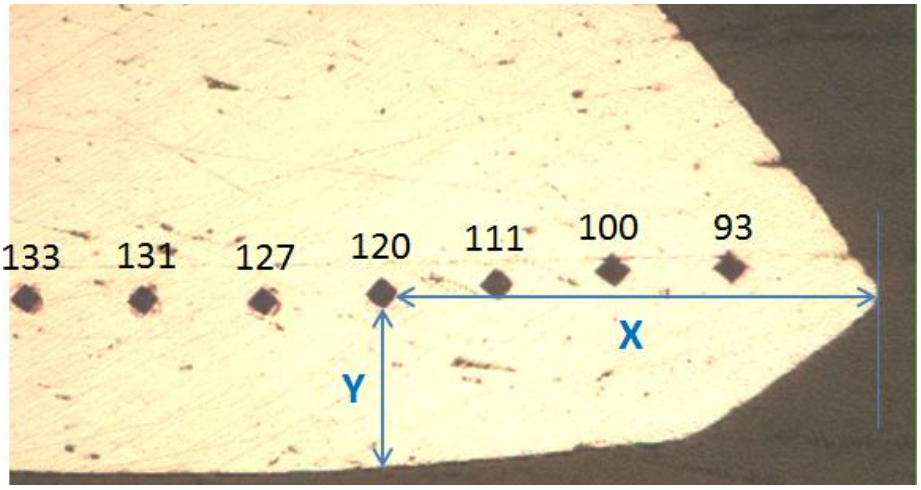


Figure 5.15: Hardness measurements on the chip: $\alpha = 30°, a_0 = 500\mu m$

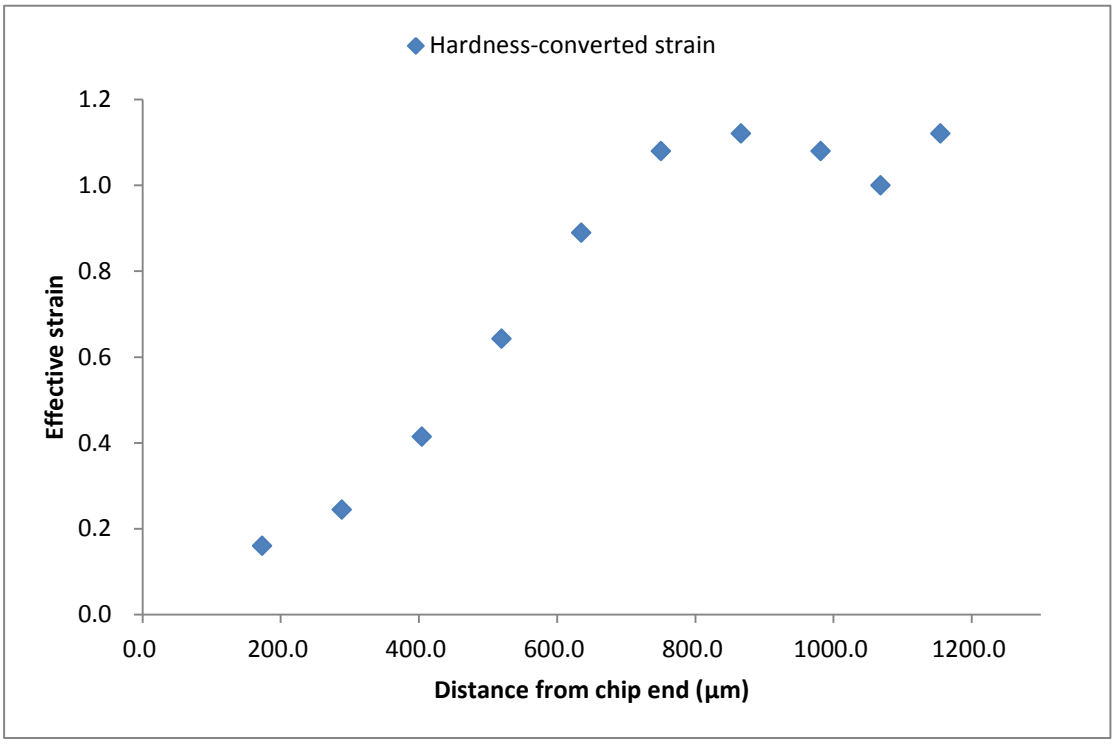


Figure 5.16: Converted strain values from the hardness field: $\alpha = 30°, a_0 = 500\mu m$

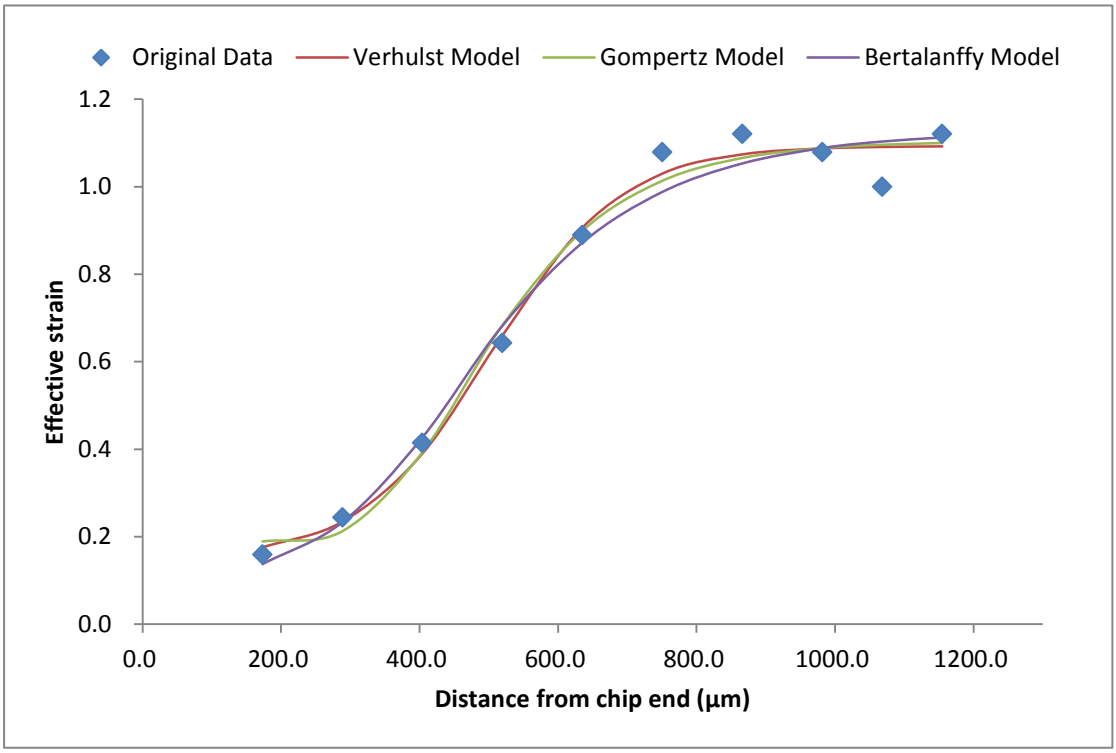


Figure 5.17: Fitting curve for each model based on the converted hardness measurements: $\alpha = 30°, a_0 = 500\mu m$

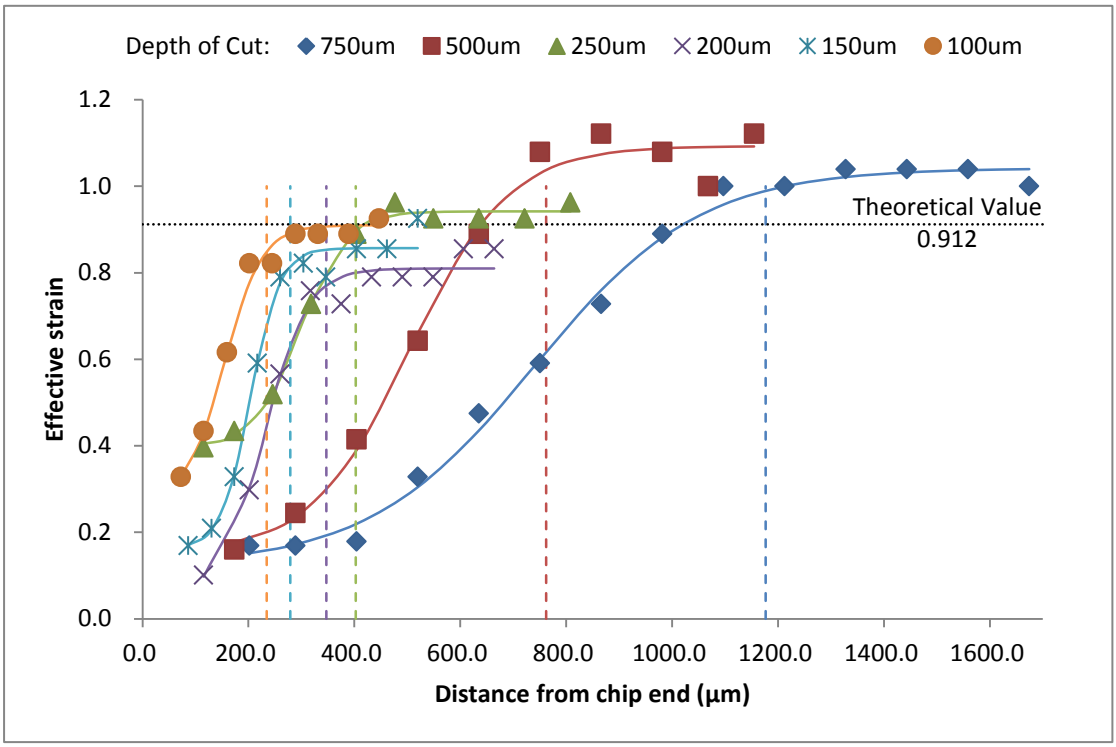


Figure 5.18: Fitting and analysis results of hardness-converted strain curves: $\alpha = 30°$

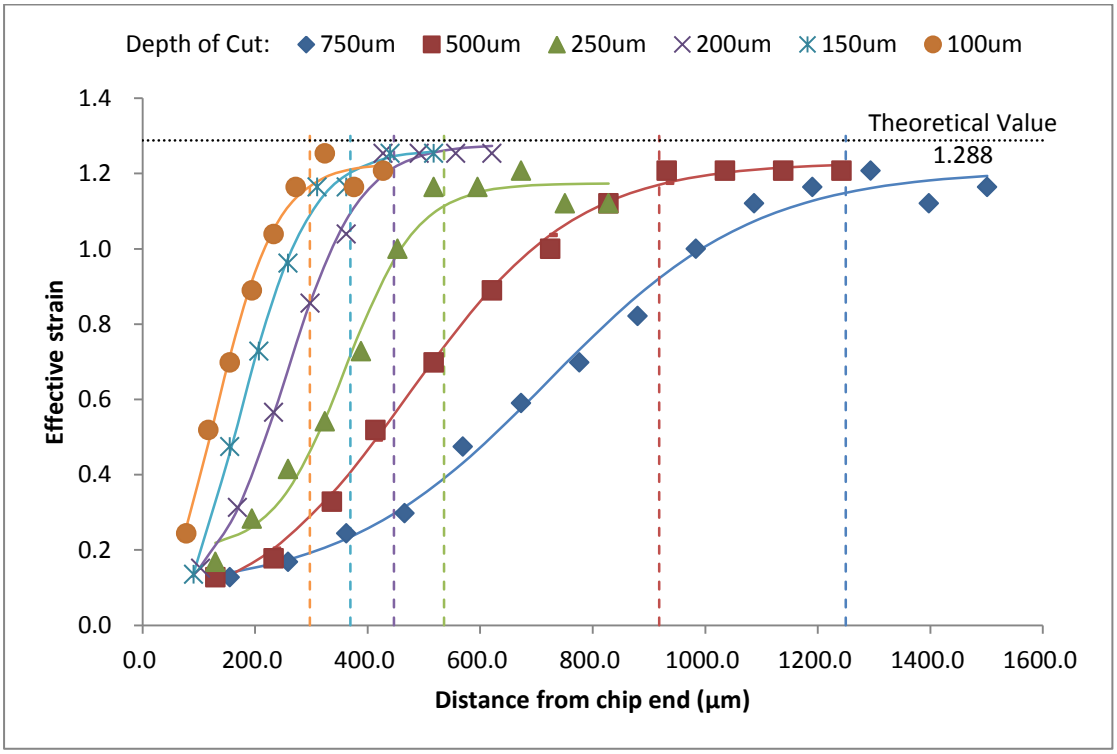


Figure 5.19: Fitting and analysis results of hardness-converted strain curves: $\alpha = 15°$

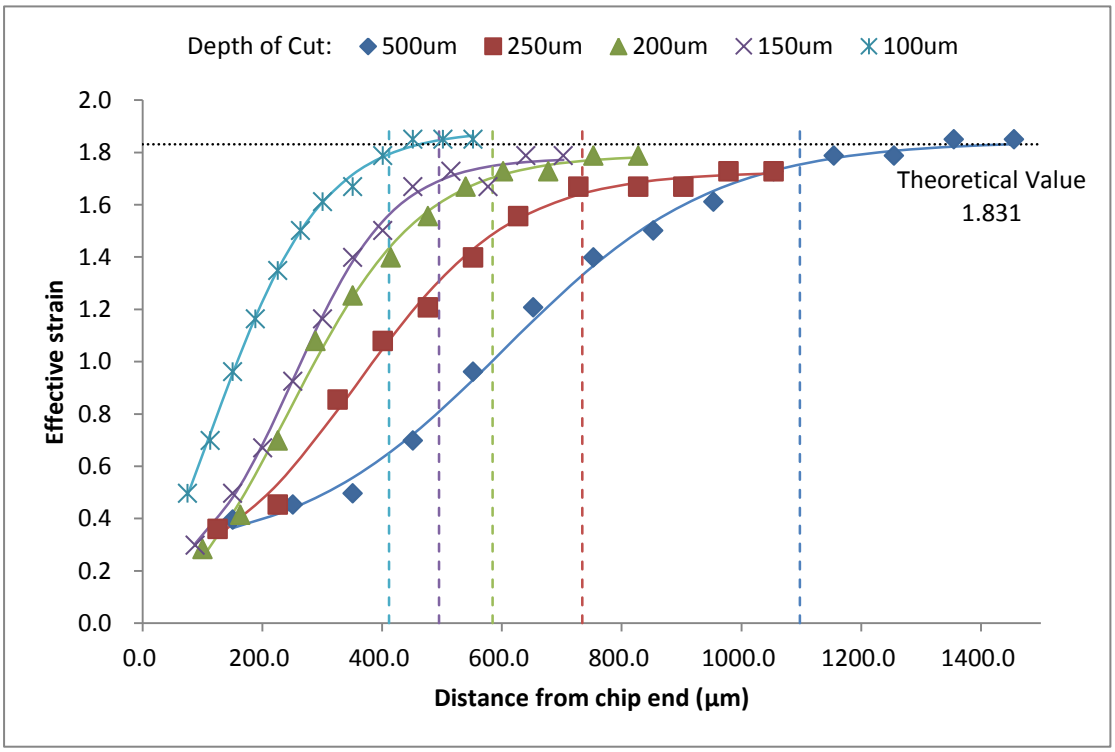


Figure 5.20: Fitting and analysis results of hardness-converted strain curves: $\alpha = 5°$

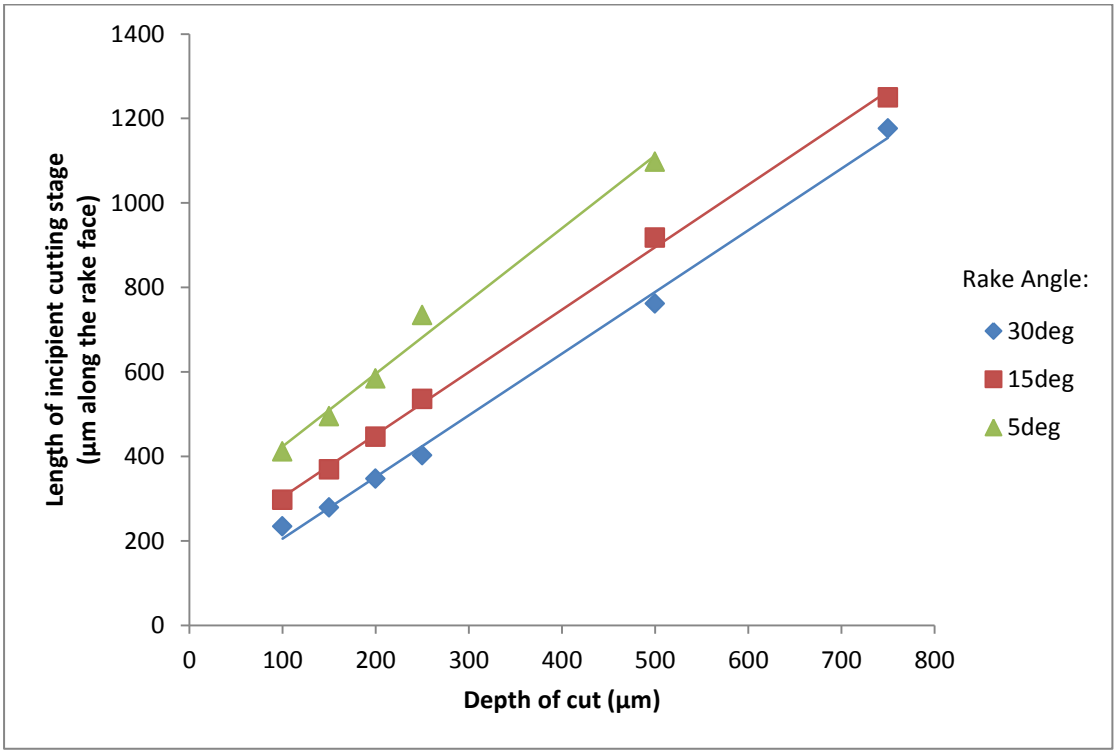


Figure 5.21: Results summary for hardness tests

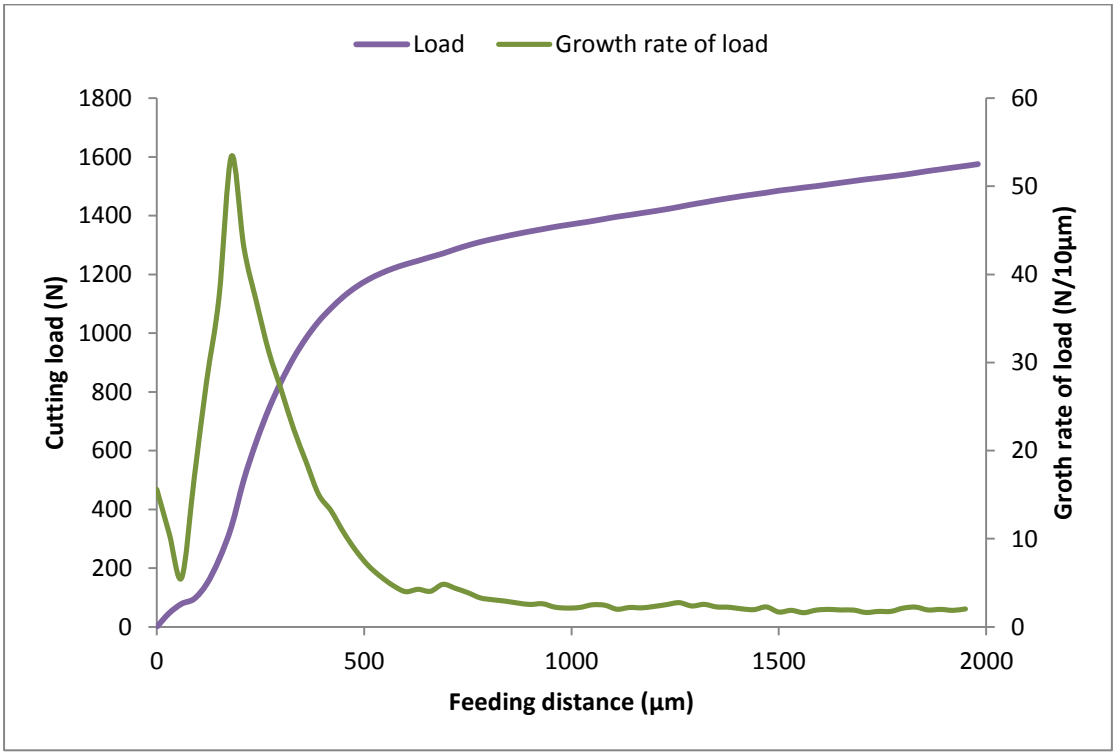


Figure 5.22: Original cutting load curve and its corresponding growth rate curve: $\alpha = 30°, a_0 = 500\mu m$

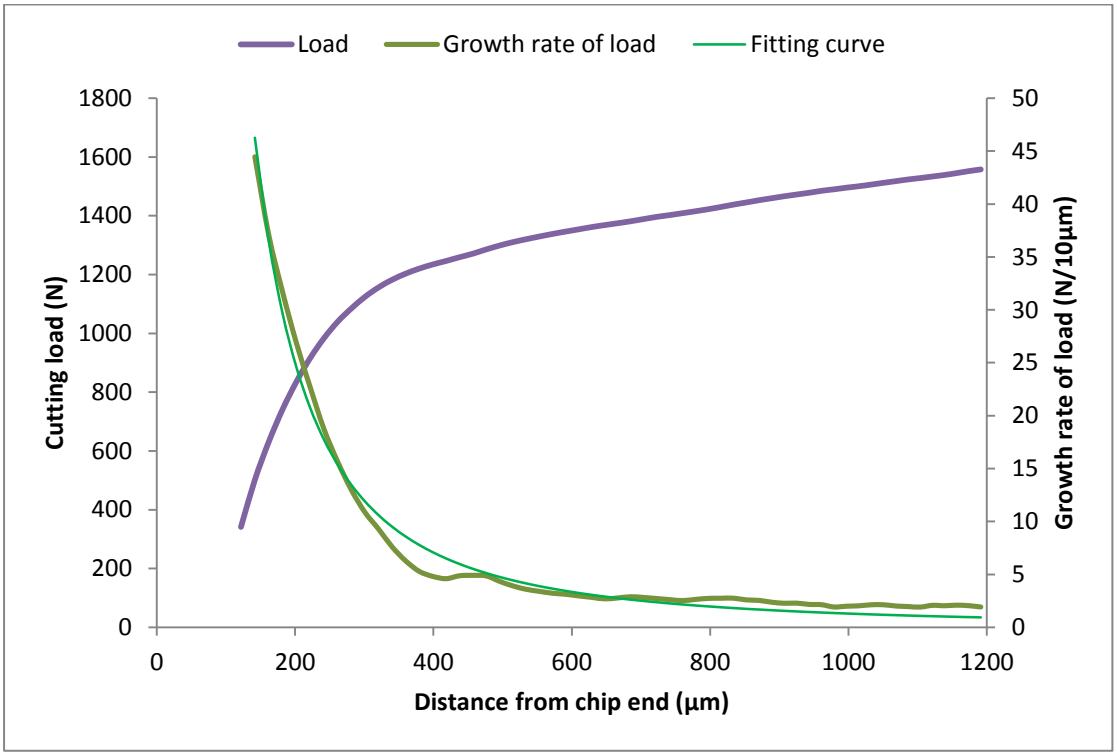


Figure 5.23: Treated cutting load curve and its corresponding growth rate curve: $\alpha = 30°, a_0 = 500\mu m$

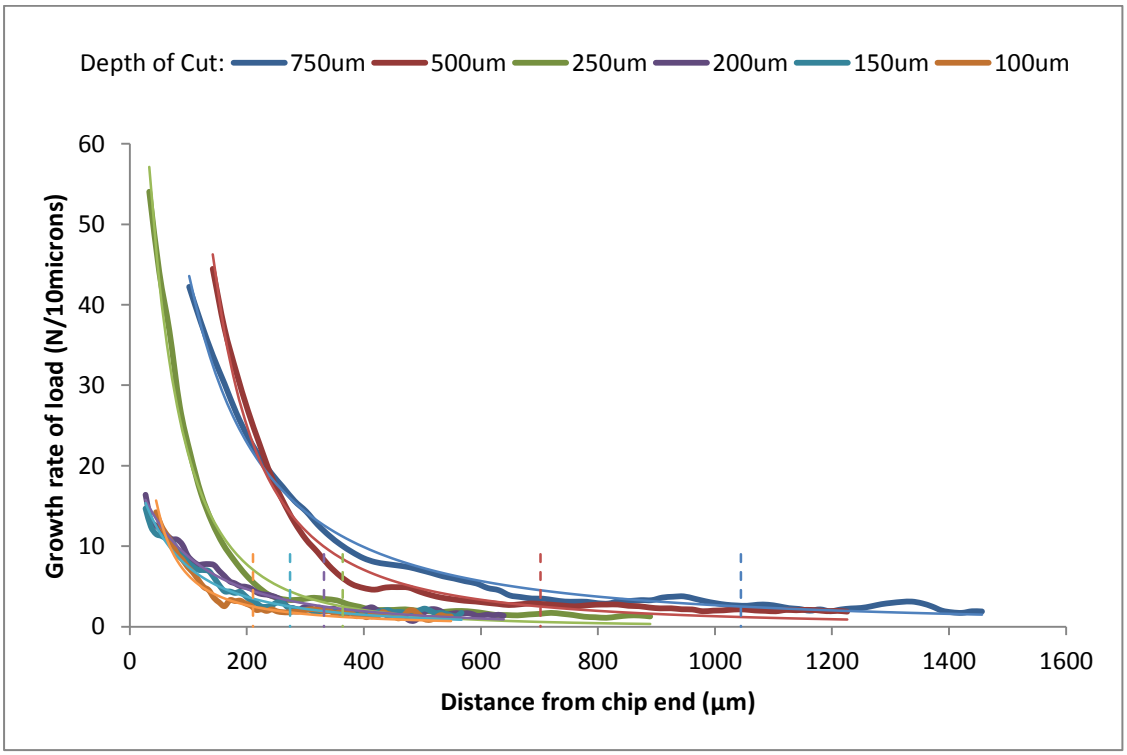


Figure 5.24: Fitting and analysis results of cutting load growth rate curves: $\alpha = 30°$

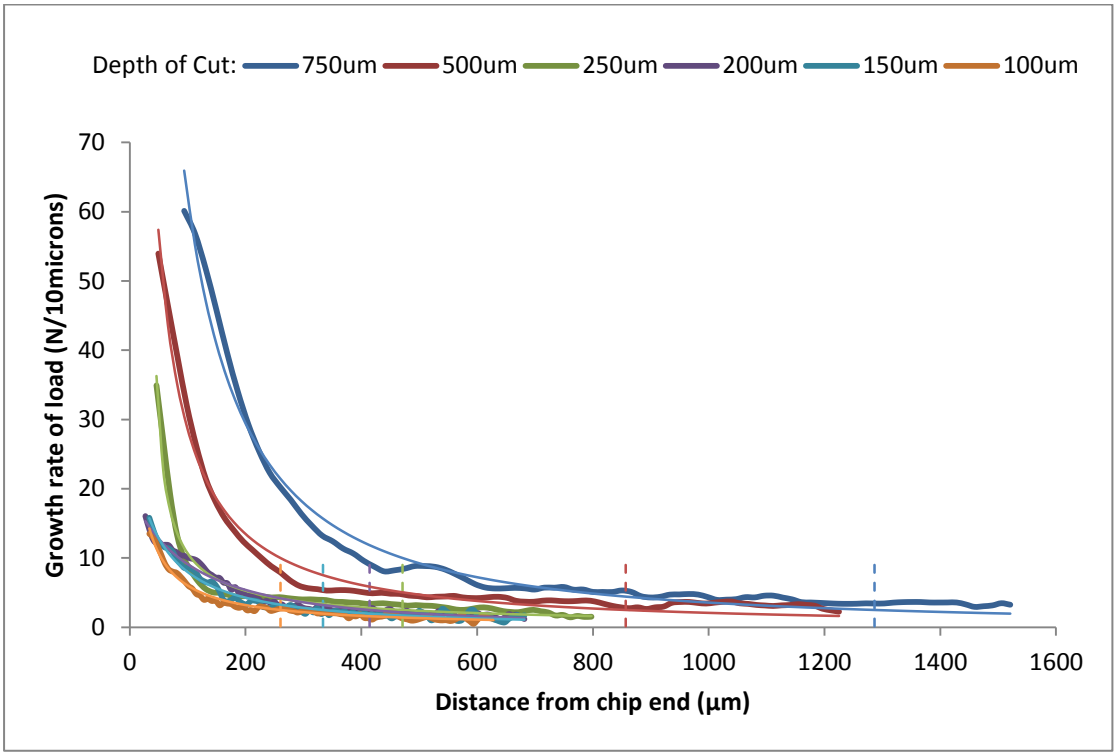


Figure 5.25: Fitting and analysis results of cutting load growth rate curves: $\alpha = 15°$

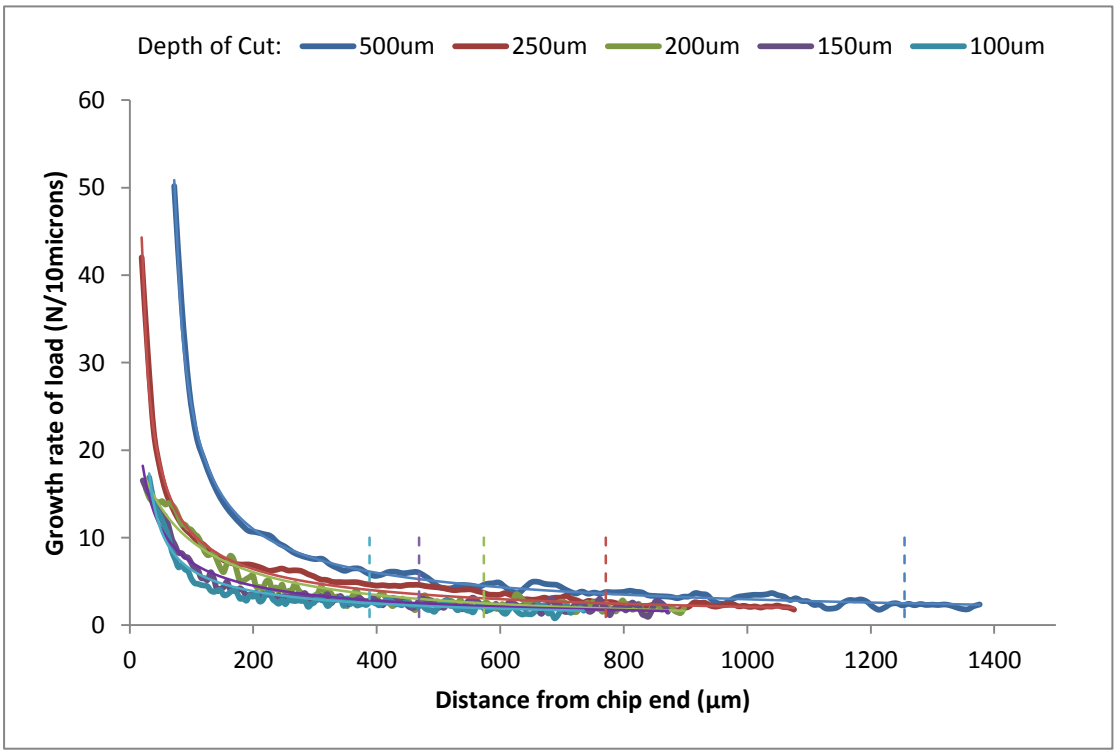


Figure 5.26: Fitting and analysis results of cutting load growth rate curves: $\alpha = 5°$

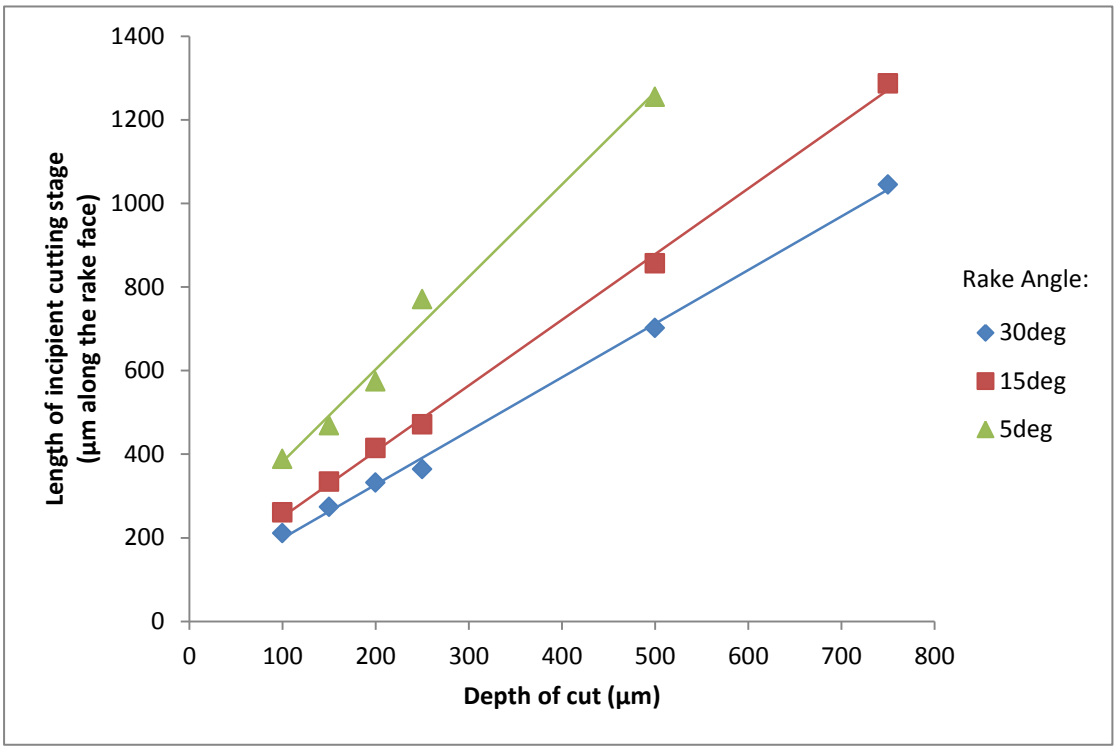


Figure 5.27: Results summary for cutting load analyses

Table 5.1: Fitting result for each model based on the strain measurements: $\alpha = 30°, a_0 = 500\mu m$

Distance from chip end (μm)	PIV strain	Verhulst model fitting	Gompertz model fitting	Bertalanffy model fitting
150.1	0.1925	0.2132	0.2390	0.1715
225.2	0.2575	0.2377	0.2413	0.2560
300.2	0.3265	0.2841	0.2642	0.2809
375.3	0.3313	0.3650	0.3438	0.3785
450.3	0.4265	0.4879	0.4826	0.5147
525.4	0.6799	0.6400	0.6403	0.6510
600.4	0.8082	0.7869	0.7780	0.7682
675.5	0.9167	0.8984	0.8807	0.8608
750.6	0.9534	0.9686	0.9505	0.9305
825.6	0.9622	1.0077	0.9952	0.9814
900.7	0.9559	1.0281	1.0230	1.0178
975.7	0.9847	1.0383	1.0399	1.0435
1050.8	1.0787	1.0433	1.0501	1.0615
1125.8	1.0973	1.0457	1.0561	1.0740
1200.9	1.1044	1.0469	1.0597	1.0827
Verhulst model: $y = 0.189 + \frac{0.859}{1+e^{4.996-0.0097x}}, R^2 = 0.9827$				
Gompertz model: $y = 0.239 + 0.826e^{-28.558e^{-0.007x}}, R^2 = 0.9823$				
Bertalanffy model: $y = 0.256 + 0.846(1 - 3.101e^{-0.005x})^3, R^2 = 0.9826$				

Table 5.2: Fitting and analysis results of PIV strain curves: $\alpha = 30°$

$\alpha = 30°$				
Depth of cut (μm)	Curve-fitting	R^2	THR	Length of incipient stage (μm)
750	$y = 0.101 + \frac{0.881}{1 + e^{6.922-0.0084x}}$	0.9964	95%	1160.8
500	$y = 0.189 + \frac{0.859}{1 + e^{4.996-0.0097x}}$	0.9827	95%	796.9
250	$y = 0.441 + \frac{0.616}{1 + e^{12.65-0.0377x}}$	0.9902	95%	398.5

Table 5.3: Fitting and analysis results of PIV strain curves: $\alpha = 15°$

$\alpha = 15°$				
Depth of cut (μm)	Curve-fitting	R^2	THR	Length of incipient stage (μm)
750	$y = 0.1 + \frac{1.104}{1 + e^{5.923 - 0.0072x}}$	0.9939	95%	1211.4
500	$y = 0.036 + \frac{1.207}{1 + e^{4.009 - 0.0076x}}$	0.9951	95%	907.2
250	$y = 0.066 + \frac{1.085}{1 + e^{3.453 - 0.012x}}$	0.9949	95%	528.9

Table 5.4: Fitting and analysis results of PIV strain curves: $\alpha = 5°$

$\alpha = 5°$				
Depth of cut (μm)	Curve-fitting	R^2	THR	Length of incipient stage (μm)
500	$y = -0.143 + \frac{2.005}{1 + e^{2.816 - 0.0054x}}$	0.9969	95%	1085.2
250	$y = -0.333 + \frac{2.148}{1 + e^{2.001 - 0.0072x}}$	0.9956	95%	713.9

Table 5.5: Theoretical steady strain values for different rake angles

Rake angle (α)	Chip thickness ratio	Theoretical strain
30°	1.82	0.912
15°	2.22	1.288
5°	3	1.831

Table 5.6: Linear fitting of PIV results for different rake angles

Rake angle (α)	Linear-fitting	R^2
30°	$y = 1.4246x + 56.433$	0.9953
15°	$y = 1.365x + 200$	0.9961
5°	$y = 1.4852x + 342.6$	1

Table 5.7: Comparison between converted hardness measurements and strain measurements: $\alpha = 30°, a_0 = 500\mu m$

Distance from chip end (μm)	Original hardness test result (HV)	Hardness-converted strain	PIV strain	Error (%)
150.1	--	--	0.1925	--
225.2	95.9	0.2076	0.2575	19.38
300.2	101.3	0.2611	0.3265	20.03
375.3	108.7	0.3720	0.3313	10.94
450.3	115.4	0.5058	0.4265	15.68
525.4	121.5	0.6551	0.6799	3.65
600.4	126.8	0.8156	0.8082	0.91
675.5	130.8	0.9561	0.9167	4.12
750.6	134	1.0794	0.9534	11.67
825.6	134.7	1.1064	0.9622	13.03
900.7	134.7	1.1085	0.9559	13.77
975.7	134.1	1.0815	0.9847	8.95
1050.8	132.4	1.0159	1.0787	5.82
1125.8	134	1.0807	1.0973	1.51
1200.9	--	--	1.1044	--
Average Error				9.95

** The 2nd and 3rd columns are obtained from the raw hardness test result using interpolation.*

Table 5.8: Fitting result for each model based on the converted hardness measurements: $\alpha = 30°, a_0 = 500\mu m$

Distance from chip end (μm)	Hardness-converted strain	Verhulst model fitting	Gompertz model fitting	Bertalanffy model fitting
173.2	0.1597	0.1756	0.1891	0.1374
288.7	0.2440	0.2345	0.2127	0.2332
404.1	0.4146	0.3900	0.3930	0.4269
519.6	0.6427	0.6598	0.6828	0.6821
635.1	0.8897	0.9063	0.8991	0.8720
750.6	1.0794	1.0299	1.0134	0.9880
866.0	1.1210	1.0737	1.0656	1.0531
981.5	1.0794	1.0873	1.0880	1.0882
1068.1	1.0000	1.0907	1.0958	1.1032
1154.7	1.1210	1.0921	1.0998	1.1124
Verhulst model: $y = 0.149 + \frac{0.944}{1+e^{5.395-0.0107x}}, R^2 = 0.9882$				
Gompertz model: $y = 0.189 + 0.915e^{-33.714e^{-0.0077x}}, R^2 = 0.9838$				
Bertalanffy model: $y = 0.219 + 0.908(1 - 3.886e^{-0.0057x})^3, R^2 = 0.9800$				

Table 5.9: Fitting and analysis results for hardness-converted strain curves: $\alpha = 30°$

$\alpha = 30°$				
Depth of cut (μm)	Curve-fitting	R^2	THR	Length of incipient stage (μm)
750	$y = 0.123 + \frac{0.919}{1 + e^{4.729 - 0.0064x}}$	0.9947	95%	1176.6
500	$y = 0.149 + \frac{0.944}{1 + e^{5.395 - 0.0107x}}$	0.9882	95%	762.1
250	$y = 0.396 + \frac{0.546}{1 + e^{6.846 - 0.0229x}}$	0.9954	95%	402.6
200	$y = 0.056 + \frac{0.754}{1 + e^{5.698 - 0.0247x}}$	0.9836	95%	347.1
150	$y = 0.16 + \frac{0.697}{1 + e^{7.374 - 0.0362x}}$	0.9878	95%	279
100	$y = 0.27 + \frac{0.64}{1 + e^{4.356 - 0.0282x}}$	0.9904	95%	234.5

Table 5.10: Fitting and analysis results for hardness-converted strain curves: $\alpha = 15°$

$\alpha = 15°$				
Depth of cut (μm)	Curve-fitting	R^2	THR	Length of incipient stage (μm)
750	$y = 0.089 + \frac{1.121}{1 + e^{3.896 - 0.0054x}}$	0.9921	95%	1249.7
500	$y = 0.001 + \frac{1.231}{1 + e^{3.174 - 0.0067x}}$	0.9976	95%	918.2
250	$y = 0.193 + \frac{0.981}{1 + e^{5.629 - 0.0156x}}$	0.9861	95%	535.8
200	$y = 0.044 + \frac{1.234}{1 + e^{3.868 - 0.0152x}}$	0.9964	95%	446.8
150	$y = -0.2 + \frac{1.465}{1 + e^{2.61 - 0.0155x}}$	0.9967	95%	369.4
100	$y = -0.138 + \frac{1.366}{1 + e^{2.288 - 0.018x}}$	0.9899	95%	297.4

Table 5.11: Fitting and analysis results for hardness-converted strain curves: $\alpha = 5°$

$\alpha = 5°$				
Depth of cut (μm)	Curve-fitting	R^2	THR	Length of incipient stage (μm)
500	$y = 0.252 + \frac{1.592}{1 + e^{3.444-0.0057x}}$	0.9965	95%	1097.6
250	$y = 0.097 + \frac{1.63}{1 + e^{2.749-0.0077x}}$	0.9942	95%	734.3
200	$y = -0.17 + \frac{1.961}{1 + e^{2.201-0.009x}}$	0.9958	95%	584.5
150	$y = 0.101 + \frac{1.678}{1 + e^{3.036-0.012x}}$	0.9968	95%	495.1
100	$y = -0.477 + \frac{2.363}{1 + e^{1.166-0.0106x}}$	0.9984	95%	411.6

Table 5.12: Linear fitting of hardness test results for different rake angles

Rake angle (α)	Linear-fitting	R^2
30°	$y = 1.4606x + 58.966$	0.9962
15°	$y = 1.4796x + 155.36$	0.9986
5°	$y = 1.7233x + 251.04$	0.9882

Table 5.13: Fitting and analysis results for cutting load growth rate curves: $\alpha = 30°$

$\alpha = 30°$				
Depth of cut (μm)	Curve-fitting	R^2	THR	Length of incipient stage (μm)
750	$y = 3.577 \times 10^5 (x + 109.8)^{-1.6836}$	0.9923	2.5	1044.4
500	$y = 5.521 \times 10^5 (x + 7.815)^{-1.8744}$	0.9860	2.5	701.9
250	$y = 5.964 \times 10^7 (x + 122.2)^{-2.7459}$	0.9914	2.5	363.9
200	$y = 4.698 \times 10^6 (x + 219.8)^{-2.2883}$	0.9793	2.5	331.9
150	$y = 4.086 \times 10^5 (x + 139)^{-1.993}$	0.9725	2.5	273.9
100	$y = 5.059 \times 10^3 (x + 16.33)^{-1.4034}$	0.9750	2.5	210.6

Table 5.14: Fitting and analysis results for cutting load growth rate curves: $\alpha = 15°$

$\alpha = 15°$				
Depth of cut (μm)	Curve-fitting	R^2	THR	Length of incipient stage (μm)
750	$y = 1.029 \times 10^5 (x + 51.77)^{-1.4759}$	0.9825	2.5	1286.5
500	$y = 8.645 \times 10^3 (x + 15.1)^{-1.2035}$	0.9789	2.5	856.7
250	$y = 284.37(x - 32.04)^{-0.778}$	0.9641	2.5	471.2
200	$y = 1.528 \times 10^5 (x + 181.2)^{-1.7248}$	0.9754	2.5	414.3
150	$y = 6.196 \times 10^3 (x + 63.29)^{-1.306}$	0.9732	2.5	333.8
100	$y = 746.32(x + 15.84)^{-1.0138}$	0.9811	2.5	260.4

Table 5.15: Fitting and analysis results for cutting load growth rate curves: $\alpha = 5°$

$\alpha = 5°$				
Depth of cut (μm)	Curve-fitting	R^2	THR	Length of incipient stage (μm)
500	$y = 353.04(x - 55.97)^{-0.6983}$	0.9948	2.5	1254.7
250	$y = 231.88(x - 7.958)^{-0.6825}$	0.9912	2.5	770.9
200	$y = 2.103 \times 10^3 (x + 78.65)^{-1.0393}$	0.9504	2.5	573.4
150	$y = 202.71(x + 8.389)^{-0.7127}$	0.9507	2.5	468.5
100	$y = 104.26(x - 14.23)^{-0.6297}$	0.9574	2.5	388.3

Table 5.16: Linear fitting of the cutting load analysis results for different rake angles

Rake angle (α)	Linear-fitting	R^2
30°	$y = 1.2832x + 70.726$	0.9976
15°	$y = 1.5699x + 93.603$	0.9986
5°	$y = 2.211x + 160.52$	0.9897

Chapter 6: Discussion

For each experimental condition using the uniaxial testing machine, the incipient stage of chip formation was characterized using PIV, micro-hardness mapping and load measurements. Fig. 6.1 shows the measurements using all three methods for the experimental condition corresponding to ($\alpha = 30°, a_0 = 500\mu m$). It is clear that each method provides similar results in terms of the estimated incipient stage length. The average of these measurements is 753.6 μm with a standard deviation of 39.2 μm as shown in Table 6.1. For the experiments conducted on the linear slide, PIV was not used and the incipient stage length was evaluated from the micro-hardness mapping and load measurements. Fig. 6.2 shows the measurements using these methods for the experimental condition corresponding to ($\alpha = 30°, a_0 = 250\mu m$). Similarity again exists for both estimates of the incipient stage length. The average of these measurements is 339.5 μm with a standard deviation of 7.6 μm as shown in Table 6.2. The average incipient stage lengths for the remaining experimental conditions investigated in the present study are summarized in Fig. 6.3 and Table 6.3. Error bars (y-axis direction) are included showing the overall range of the data for each experimental condition. Indeed, the small deviations present in the ranges for these values are indicative of good degree of consistency across the experimental results and measurement methods. Table 6.4 provides the linear fits of the average incipient stage lengths; all have R^2 values greater than 0.99.

The incipient stage lengths in the present study can be compared with those found in some other works such as FEM simulations. The strain field in FEM of copper machining from Ref. [5] is shown in Fig. 6.4. From the figure, the length of incipient deformation is about 620μm for conditions of ($\alpha = 0°, a_0 = 250\mu m, V_0 = 10mm/s$). From the present study, conditions of ($\alpha = 5°, a_0 = 250\mu m$) resulted in an incipient stage length of 739.7μm. Another example could be found in Ref. [50] which provided a measure of deformation in distortional energy in the deformed chip. Fig. 6.5 shows this energy distribution for experimental conditions of ($\alpha = 10°, a_0 = 500\mu m$), where

the incipient stage is about 820μm in length. This can be compared with the results of the present study for the ($\alpha = 15°, a_0 = 500\mu m$) experimental condition, which resulted in an incipient length of 894μm. While these comparisons indicate that the lengths of incipient cutting stage obtained in the present study are somewhat consistent with that of simulations, perhaps the most telling evidence of the reliability of the results is the degree of agreement across multiple measurement methods (e.g., PIV, hardness, load).

The relationships between the incipient stage length and the controllable process parameters can be elucidated from the summary data. First, the slopes of the linear fits describing the relationship between incipient stage length and depth of cut are positive. That is, the length of incipient cutting stage increases with increasing depth of cut. This is expected as the incipient cutting stage is directly related to the time (or length traversed) for the primary deformation zone to grow to a steady size. Recall that the steady primary deformation zone in orthogonal cutting is a shear plane extending from the immediate vicinity of the cutting edge to the free surface [18]. Given that the shear angle (ϕ) of this plane is a constant for a given rake angle, the length of the shear plane ($a_0/sin\phi$) increases with the increase linearly with depth of cut. As the shear plane length increases, the length of the incipient stage also should increase. Thus, a larger depth of cut (at constant rake angle) will lead to a longer time for the formation of a steady-state shear plane. This is consistent with the present results which show a linear dependence of incipient stage length on depth of cut. It should also be noted that because the slopes of the fitting lines are larger than unity, the length of incipient cutting stage is larger than the depth of cut.

Second, the relative offsets between the individual curves in Fig. 6.1 indicate that the length of incipient cutting stage increases with decreasing rake angle. This can be rationalized by considering the effect of shear angle on the incipient cutting stage. According to Eq. 2.2 and the chip thickness ratios listed in Table 6.4, the shear angle for each rake angle is calculated and shown in Table 6.5. These values range from 33° to 19° as rake angle decreases from 30° to 5°, respectively. As the shear angle

decreases with rake angle, and the length of the shear plane is given by $a_0/sin\phi$, smaller rake angles increase the length of the shear plane. As larger shear plane lengths correspond to longer incipient stages, the decrease in rake angle is expected to increase incipient stage length. It also can be seen that the relative slope of the linear fits increases with decreasing rake angle, indicating a higher more sensitivity to depth of cut for smaller rake angles. This can be rationalized by considering the sensitivity of the length of the shear plane to given changes in depth of cut for changing shear angles. If a variation in depth of cut of Δa_0 occurs, the resulting variation on the length of shear plane is given by $\Delta a_0/sin\phi$. For constant Δa_0, this quantity is larger at smaller values for $sin\phi$ which occur at smaller rake angles. Thus, the length of the incipient stage is expected to be more sensitive to depth of cut at smaller rake angles. The shear plane length for each set of cutting conditions is listed in Table 6.6, and the shear plane length versus incipient stage length plot is shown in Fig. 6.6. The fitting result exhibits a good proportional relationship (R^2 greater than 0.98) between them, which enhances the persuasiveness of the analysis here.

A final observation is that the intercepts of the three trend lines are all non-zero. While this may not be sensible, it may be explained by the relative importance of the cutting edge geometry at small depths of cut. In this size regime, the cutting edge appears to have a more rounded geometry. This leads to non-linear response at small depths of cut and is often referred to as the “size-effect” in machining [53,54]. Moreover, because this effect is more significant under smaller tool rake angle [53], it can also be used to explain why the intercept of fitting line increases with the decrease of rake angle. Fig. 6.7 shows the cutting edge of the cutting die and the cutter used in planer machining, respectively; both of them were ground before being used in the experiments. According to Ref. [55], it is usual for a sharp edge obtained by grinding of high-speed steel to have nose radius smaller than 5 μm. Under this estimation, the height of the round-corner-affected zone (a_n in Fig. 6.7(c)) ranges from 5.4 to 7.5 μm for different rake angles, which is only about 5-8% of the 100 μm depth of cut. Thus,

the “size effect” was hardly to be observed for the depths of cut tested in the present study.

While the overall range of experimental results in Table 6.3 has low variance, several potential sources of error in the present measurements are possible. First, no repeat experiments were performed due to the difficulty in executing the actual experimental setup. Replication of the experimental conditions may reduce the error in the final results. Second, error in setting the actual depths of cut in the uniaxial testing machine is possible (depths of cut ~ 250, 500, 750 μm). For example, when preparing the workpieces, the dimension error may reach as much as 25 μm. In Fig. 6.3, the error bars in x-axis direction indicate the potential errors caused by this dimensional uncertainty, although they will not influence the resulting conclusions significantly (less than 10%). Third, the PIV algorithm utilized in this research has an inherent error. According to the error analysis of the PIV algorithm described in Section 4.2, the error of the strain field can be 10% at most, which results in the inaccuracy of the PIV strain curves. Lastly, the raw cutting load data has noise that had to be removed by data smoothing methods prior to data analysis. This could lead to the loss of important information at the beginning of the cutting process and the error in fitting the cutting load curves.

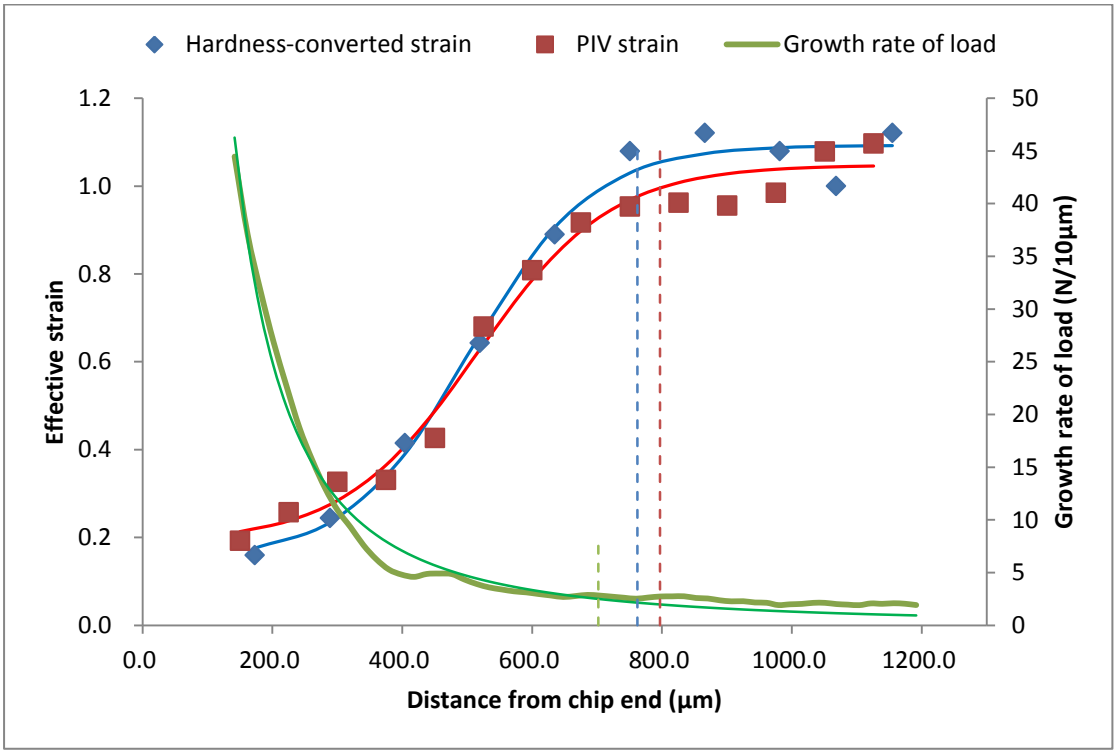


Figure 6.1: Comprehensive results of curves analysis: $\alpha = 30°, a_0 = 500\mu m$

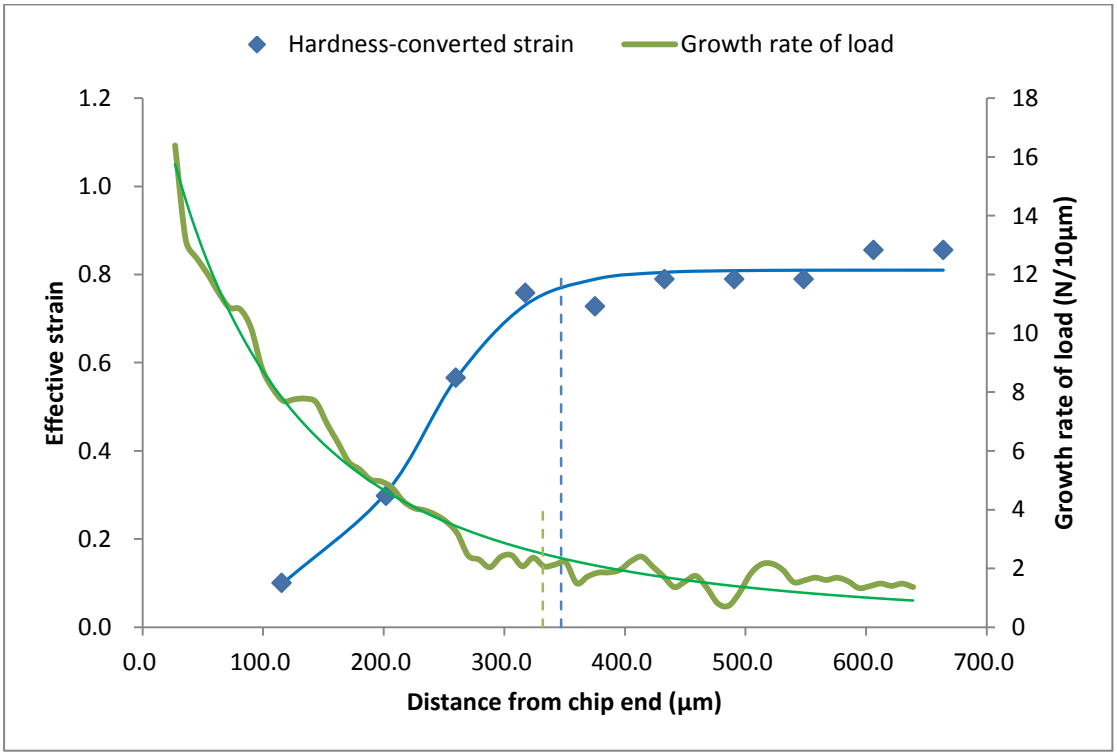


Figure 6.2: Comprehensive results of curves analysis: $\alpha = 30°, a_0 = 200\mu m$

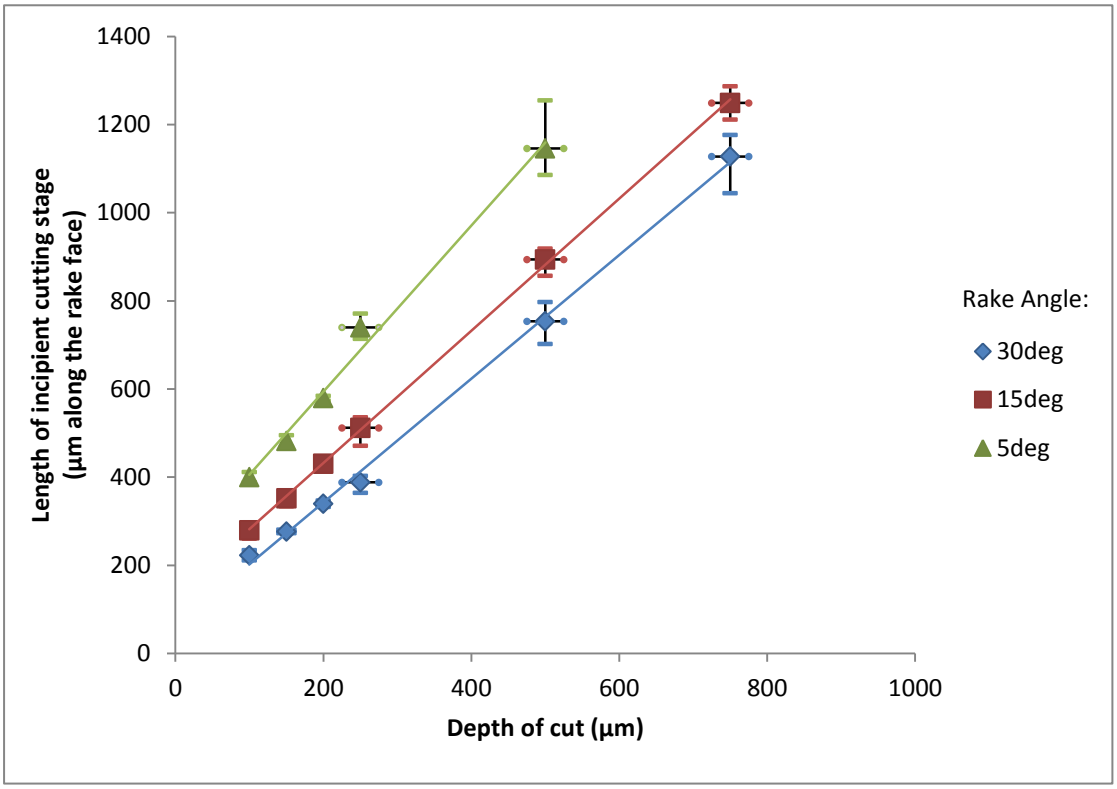


Figure 6.3: Summary and linear fittings of final results

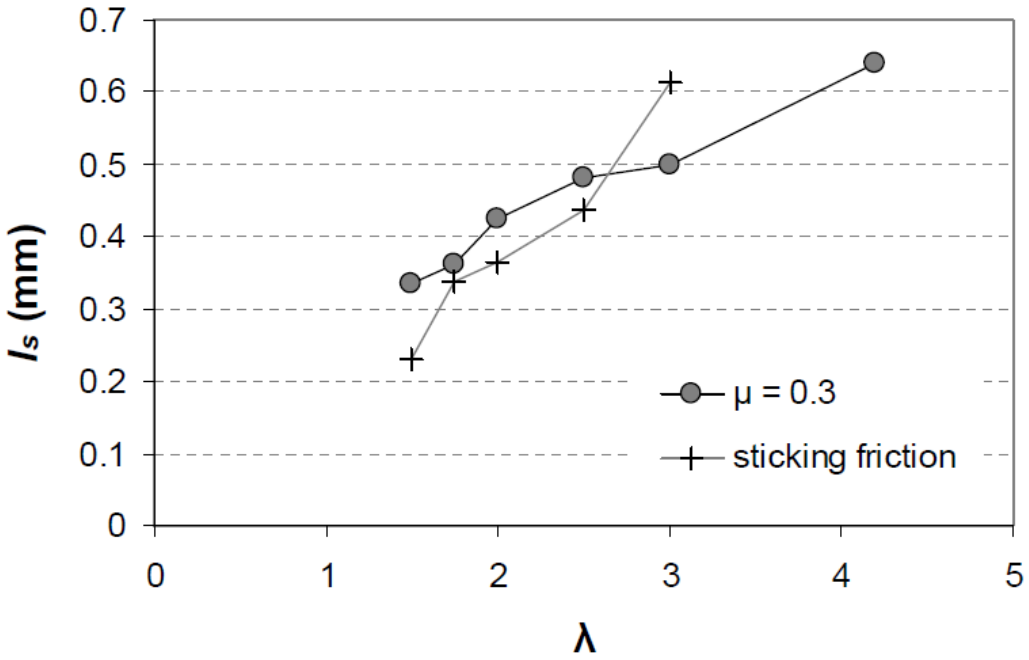


Figure 6.4: Dependence of the length of incipient cutting stage on chip thickness ratio (copper, $\alpha = 0°, a_0 = 250\mu m, V_0 = 10mm/s$) [5]

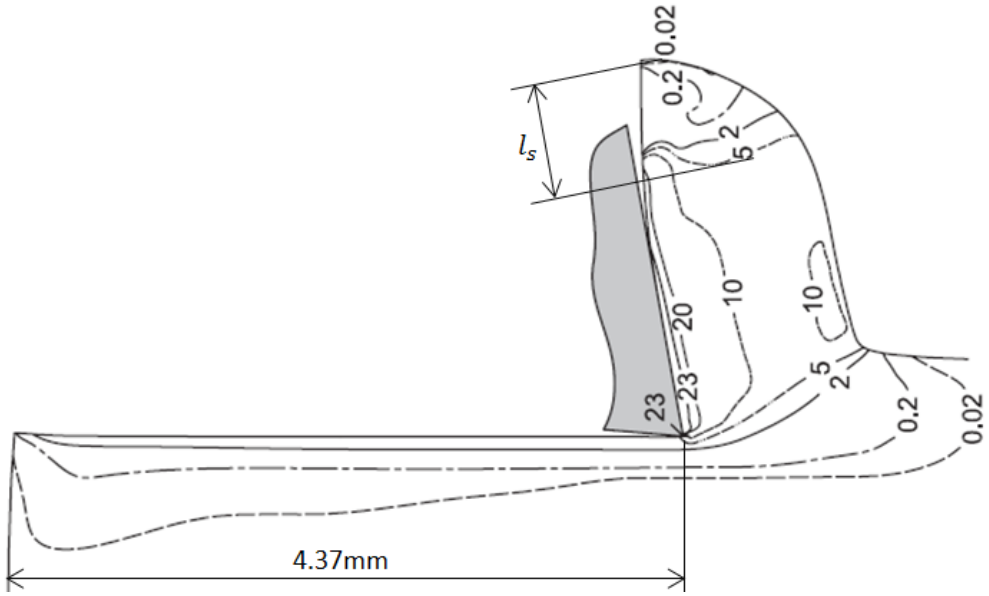


Figure 6.5: Estimation of the length of incipient cutting stage based on the distortional energy distribution (copper, $\alpha = 10°, a_0 = 500\mu m, V_0 = 0.83mm/s$) [50]

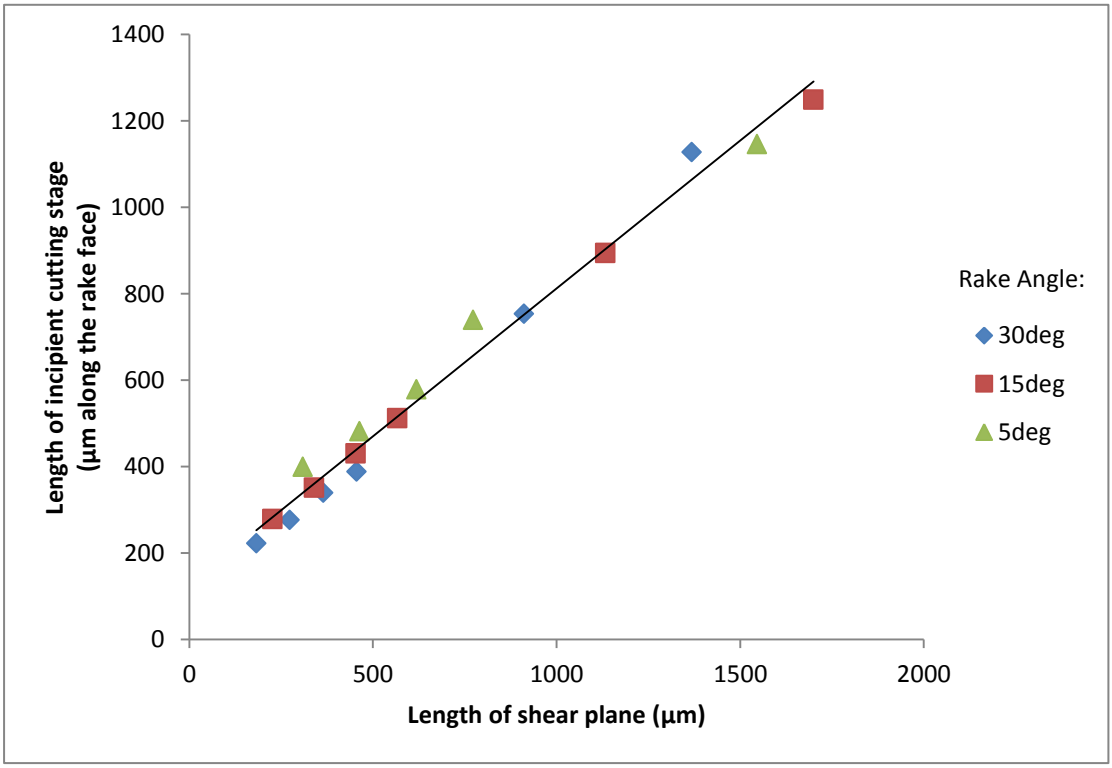


Figure 6.6: Relationship between shear plane length and incipient cutting stage length

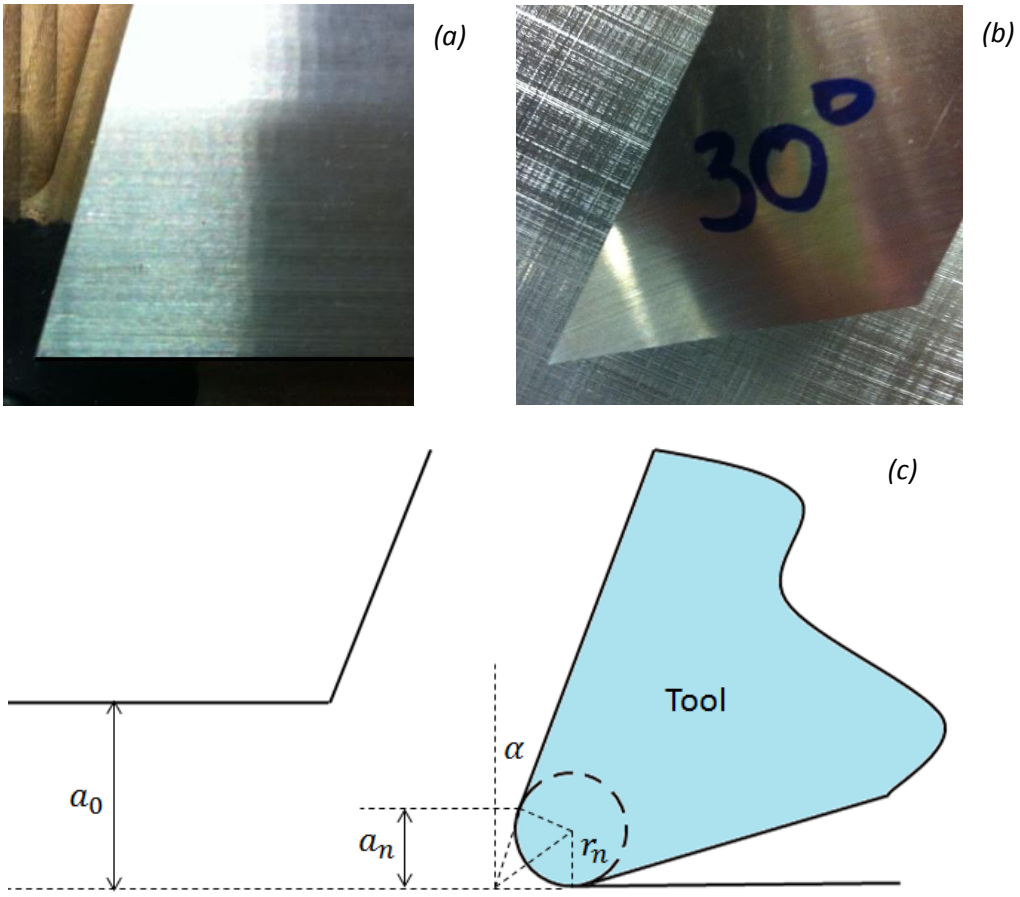


Figure 6.7: Cutting edge and the effect of nose radius: (a) cutting edge of the 15° cutting die, (b) cutting edge of the 30° cutter used in the planer machining, (c) evaluation of the effect of nose radius

Table 6.1: Final result of curves analysis: $\alpha = 30°, a_0 = 500\mu m$

$\alpha = 30°, a_0 = 500\mu m$			
Method	Length of incipient stage (μm)	Final result – Mean (μm)	Standard deviation (μm)
PIV	796.9		
Hardness Test	762.1	753.6	39.2
Cutting Load Analysis	701.9		

Table 6.2: Final result of curves analysis: $\alpha = 30°, a_0 = 200\mu m$

$\alpha = 30°, a_0 = 250\mu m$			
Method	Length of incipient stage (μm)	Final result – Mean (μm)	Standard deviation (μm)
Hardness Test	347.1		
Cutting Load Analysis	331.9	339.5	7.6

Table 6.3: Summary of the final lengths of incipient cutting stage

	Rake angle (α)	30°		15°		5°	
Depth of cut (a_0)	Length of incipient stage (μm)		Mean		Mean		Mean
750μm	PIV	1160.8	**1127.3**	1211.4	**1249.2**	--	
	Hardness Test	1176.6		1249.7			
	Cutting Load	1044.4		1286.5			
500μm	PIV	796.9	**753.6**	907.2	**894**	1085.2	**1145.8**
	Hardness Test	762.1		918.2		1097.6	
	Cutting Load	701.9		856.7		1254.7	
250μm	PIV	398.5	**388.3**	528.9	**512**	713.9	**739.7**
	Hardness Test	402.6		535.8		734.3	
	Cutting Load	363.9		471.2		770.9	
200μm	Hardness Test	347.1	**339.5**	446.8	**430.6**	584.5	**579**
	Cutting Load	331.9		414.3		573.4	
150μm	Hardness Test	279	**276.5**	369.4	**351.6**	495.1	**481.8**
	Cutting Load	273.9		333.8		468.5	
100μm	Hardness Test	234.5	**222.6**	297.4	**278.9**	411.6	**400**
	Cutting Load	210.6		260.4		388.3	

Table 6.4: Linear fittings of the final results

Rake angle (α)	Linear Fitting	R^2
$30°$	$y = 1.4054x + 61.206$	0.9979
$15°$	$y = 1.5007x + 131.65$	0.9996
$5°$	$y = 1.8844x + 217.01$	0.9902

Table 6.5: Theoretical shear angles for different tool rake angles

Rake angle (α)	Chip thickness ratio	Theoretical shear angle
$30°$	1.82	33.27°
$15°$	2.22	26.20°
$5°$	3	18.88°

Table 6.6: Length of shear plane for different cutting conditions

Rake angle	$30°$		$15°$		$5°$	
Depth of cut (μm)	Shear plane length (μm)	Incipient stage length (μm)	Shear plane length (μm)	Incipient stage length (μm)	Shear plane length (μm)	Incipient stage length (μm)
750	1367.8	1127.3	1699.5	1249.2	--	
500	911.8	753.6	1133	894	1545.9	1145.8
250	455.9	388.3	566.5	512	772.9	739.7
200	364.7	339.5	453.2	430.6	618.4	579
150	273.6	276.5	339.9	351.6	463.8	481.8
100	182.4	222.6	226.6	278.9	309.2	400
Linear fitting: $y = 0.684x + 127.98,\ R^2 = 0.9846$						

Chapter 7: Conclusions and Future Work

The present study investigated the influences of tool rake angle and depth of cut on the incipient stage length of metal cutting. Six depths of cut ranging from 100 μm to 750 μm and three rake angles of 5°, 15°, 30° were selected to combine into different cutting conditions. Under each of them, orthogonal cutting experiment was carried out on a specially-fabricated cutting die or a servo-controlled linear slide stage, and a variety of methods were applied to measure the incipient regime length, including in situ methods (e.g., PIV, load analysis) and a post-mortem method (hardness mapping). Annealed copper was employed as the cutting material, and low cutting speed of 0.3 mm/s was used in all the cutting experiments to avoid the thermal effects.

The main contributions of this research are as follows. First, a new PIV algorithm was developed in Matlab for computing the velocity, strain rate and strain fields for material flows. The algorithm was calibrated in order to find the optimal experimental environment, and was shown to be capable of accurately predicting the deformation behaviors in chip formation. Second, the basic principle and detailed procedure of the three methods for measuring the incipient cutting stage were provided in the thesis. Their analysis results exhibited good consistency and all of them were proved to be reliable tools for studying the incipient straining processes. Third, the final results indicated that the incipient stage length increases with increasing depth of cut or decreasing tool rake angle. For a specific rake angle, there is a good linear relationship between the length of incipient cutting stage and the depth of cut; the accurate relational expressions were generated. These experimental results are helpful for developing a better understanding of the deformation behaviors in unsteady material removal processes. Furthermore, the relationships obtained will aid in designing and understanding processes wherein the chip formation can occur at small length scales, such as vibration-assisted machining processes, surface

texturing methods, machining of brittle material systems, and the use of novel chip breaking tools.

In the future, some other meaningful works are desired to be carried out as the extension of this research. With more sophisticated laboratory equipment, smaller depths of cut are desired to be tested so that the “size effect” on the length of incipient cutting stage can be carefully investigated. The PIV algorithm has potential to achieve better measurement accuracy, by adding improved function modules such as shifting sub-windows [52] into the current Matlab program and upgrading the camera lens with higher resolution or a better way to introduce visual markers onto the workpiece surface. Also, some other material systems should be studied in the future, to establish the robustness of the incipient machining results from this study.

Reference

[1] Merchant, M. E., 1945, “Mechanics of the Metal Cutting Process. I. Orthogonal Cutting and a Type 2 Chip,” Journal of Applied Physics, Vol. 16, pp. 267-275

[2] Thompson, P. J., Ogden, H., and Butterworth, N. A., 1969, “An Apparent Strain Analysis of Orthogonal Cutting,” Int. J. Mach. Tool Des. Res., Vol. 9, pp. 97-116

[3] Tyan, T., and Yang, W. H., 1992, “Analysis of Orthogonal Metal Cutting Processes,” International Journal for Numerical Methods in Engineering, Vol. 34, pp. 365-389

[4] Shaw, M., 1984, *Metal Cutting Principles*, Oxford University Press, New York, NY

[5] Sevier, M., Yang, H. T. Y., Moscoso, W., and Chandrasekar, S., 2008, “Analysis of Severe Plastic Deformation by Large Strain Extrusion Machining,” Metallurgical and Materials Transactions A, Vol. 39(11), pp. 2645-2655

[6] Sevier, M., Lee, S., Shankar, M. R., Yang, H. T. Y., Chandrasekar, S., and Compton, W. D., 2006, “Deformation Mechanics Associated with Formation of Ultra-Fine Grained Chips in Machining,” Materials Science Forum, Vols. 503-504, pp. 379-384

[7] Shih, A. J., 1996, “Finite Element Analysis of the Rake Angle Effects in Orthogonal Metal Cutting,” Int. J. Mech. Sci., Vol. 38(1), pp. 1-17

[8] Zhang, X., Wu, S., Wang, H., and Liu, C. R., 2011, “Predicting the Effects of Cutting Parameters and Tool Geometry on Hard Turning Process Using Finite Element Method,” Journal of Manufacturing Science and Engineering, Vol. 133, pp. (041010)1-13

[9] Stevenson, M. G., and Oxley, P L. B., 1970, “An Experimental Investigation of the Influence of Strain-Rate and Temperature on the Flow Stress Properties of a Low Carbon Steel Using a Machining Test,” Proceedings of the Institution of Mechanical Engineers, Vol. 185, pp. 741-754

[10] Okushima, K., and Hitomi, K., 1972, “Transitional Phenomenon in Metal Cutting,” International Journal of Production Research, Vol. 10(3), pp. 234-246

[11] Oguri, M., Fujii, H., Yamaguchi, K., and Kato, S., 1976, “On Transient Cutting Mechanics at the Initial Stage of Peripheral Milling Process,” Bulletin of the JSME, Vol. 19(127), pp. 61-70

[12] Mhalselar, S. D., Shrikantha, M. G., Rao, S., and Gangadharan, K. V., 2009, “Determination of Transient and Steady State Cutting in Face Milling Operation Using Recurrence Quantification Analysis,” ARPN Journal of Engineering and Applied Sciences, Vol. 4(10), pp. 36-46

[13] Guo, Y. B., and Yen, D. W., 2004, “A FEM Study on Mechanisms of Discontinuous Chip Formation in Hard Machining,” Journal of Materials Processing Technology, Vol. 155-156, pp. 1350-1356

[14] Zhang, D., and Wang, L., 1998, “Investigation of Chip in Vibration Drilling,” International Journal of Machine Tools and Manufacture, Vol. 38(3), pp.165-176

[15] Palanisamy, S., McDonald, S. D., and Dargusch M. S., 2009, “Effects of Coolant Pressure on Chip Formation while Turning Ti6Al4V Alloy,” International Journal of Machine Tools and Manufacture, Vol. 49, pp. 739-743

[16] Mann, J. B., Guo, Y., Saldana, C., Compton, W. D., and Chandrasekar, S., 2011, “Enhancing Material Removal Processes Using Modulation-Assisted Machining,” Tribology International, Vol. 44, pp. 1225-1235

[17] Mann, J. B., Saldana, C., Moscoso, W., Compton, W. D., and Chandrasekar, S., 2009, “Effects of Controlled Modulation on Interface Tribology and Deformation in Machining,” Tribology Letters, Vol. 35, pp. 221-227

[18] Weinmann, K. J., and Von Turkovich, B. F., 1971, “Mechanics of Tool-Workpiece Engagement and Incipient Deformation in Machining of 70/30 Brass,” Journal of Engineering for Industry, Vol. 93(4), pp. 1079-1089

[19] Ko, D., Ko, S., and Kim, B., 2002, “Rigid-Thermoviscoplastic Finite Element Simulation of Non-Steady-State Orthogonal Cutting,” Journal of Materials Processing Technology, Vol. 130-131, pp. 345-350

[20] Klamecki, B. E., 1973, “Incipient Chip Formation in Metal Cutting – A Three-Dimension Finite Element Analysis,” Ph. D. dissertation, University of Illinois

[21] Lo, S., 1999, “An Analysis of Cutting under Different Rake Angles Using the Finite Element Method,” Journal of Materials Processing Technology, Vol. 105, pp. 143-151

[22] Saldana, C., 2006, “Nanostructured Particulate by Modulation-Assisted Machining,” Master thesis, Purdue University

[23] Calistes, R. A., 2006, “Characteristics of Surface Layer Produced in Plane Strain Machining of Copper,” Master thesis, Purdue Univerisity

[24] Hong, Z., 2000, “Plastic Deformation and Chip Formation Mechanisms during Machining of Copper, Aluminum and an Aluminum Matrix Composite,” Ph.D. dissertation, University of Windsor

[25] Brown, T. L., Saldana, C., Murthy, T. G., Mann, J. B., Guo, Y., Allard, L. F., King, A. H., Compton, W. D., Trumble, K. P., and Chandrasekar, S., 2009, “A Study of the Interactive Effects of Strain, Strain Rate and Temperature in Severe Plastic Deformation of Copper,” Acta Materialia, Vol. 57, pp. 5491-5500

[26] Guo, Y., Saldana, C., Compton, W. D., and Chandrasekar, S., 2011, “Controlling Deformation and Microstructure on Machined Surfaces,” Acta Metarialia, Vol. 59, pp. 4538-4547

[27] Guo, Y., Efe, M., Moscoso, W., Sagapuram, D., Trumble, K. P., and Chandrasekar, S., 2012, “Deformation Field in Large-Strain Extrusion Machining and Implications for Deformation Processing,” Scripta Materialia, Vol. 66, pp. 235-238

[28] Oxley, P. L. B., and Hastings, W. F., 1977, “Predicting the Strain Rate in the Zone of Intense Shear in Which the Chip is Formed in Machining from the Dynamic Flow Stress Properties of the Work Material and the Cutting Conditions,” Proc. R. Soc. Lond. A, Vol. 356, pp. 395-410

[29] Tay, A. O., Stevenson, M. G., and De Vahl Davis, G., 1974, “Using the Finite Element Method to Determine Temperature Distributions in Orthogonal Machining,” Proceedings of the Institution of Mechanical Engineers, Vol.188, pp. 627-638

[30] Komvopoulos, K., and Erpenbeck, S. A., 1991, “Finite Element Modeling of Orthogonal Metal Cutting,” Journal of Engineering for Industry, Vol. 113, pp. 253-267

[31] Kilicaslan, C., 2009, “Modeling and Simulation of Metal Cutting by Finite Element Method,” Master thesis, Izmir Institute of Technology

[32] Priyadarshini, A., Pal, S. K., and Samantaray, A. K., 2011, “Finite Element Modeling of Chip Formation in Orthogonal Machining,” IJMMME, Vol. 1(4), pp. 19-45

[33] Mamalis, A. G., Horvath, M., Branis, A. S., and Manolakos, D. E., 2001, “Finite Element Simulation of Chip Formation in Orthogonal Metal Cutting,” Journal of Materials Processing Technology, Vol. 110, pp. 19-27

[34] Shih. A. J., 1996, “Finite Element Analysis of Orthogonal Metal Cutting Mechanics,” Int. J. Mach. Tools Manufact., Vol. 36(2), pp. 255-273

[35] Bil, H., Kilic, S. E., and Tekkaya, A. E., 2004, “A Comparison of Orthogonal Cutting Data from Experiments with Three Different Finite Element Models,” International Journal of Machine Tools and Manufacture, Vol. 44, pp. 933-944

[36] Potdar, Y. K., and Zehnder, A. T., 2003, “Measurements and Simulations of Temperature and Deformation Fields in Transient Metal Cutting,” Journal of Manufacturing Science and Engineering, Vol. 125, pp. 645-655

[37] Potdar, Y. K., and Zehnder, A. T., 2004, “Temperature and Deformation Measurements in Transient Metal Cutting,” Society for Experimental Mechanics, Vol. 44(1), pp. 1-9

[38] Lee, S., 2006, “Direct Measurements of Severe Plastic Deformation in Machining and Equal Channel Angular Pressing,” Ph. D. dissertation, Purdue University

[39] Lee, S., Hwang, J., Shankar, M. R., Chandrasekar, S., and Compton, W. D., 2006, “Large Strain Deformation Field in Machining,” Metallurgical and Materials Transactions, Vol. 37A, pp. 1633-1643

[40] Pan, B., Qian, K., Xie, H., and Asundi, A., 2009, “Two-Dimensional Digital Image Correlation for In-plane Displacement and Strain Measurement: A Review,” Meas. Sci. Technol., Vol. 20(6):062001, 17pp

[41] Westerweel, J., 2008, “On Velocity Gradients in PIV Interrogation,” Exp Fluids, Vol. 44, pp. 831-842

[42] Gnanamanickam, E. P., Lee, S., Sullivan, J. P., and Chandrasekar, S., 2009, “Direct Measurement of Large-Strain Deformation Fields by Particle Tracking,” Meas. Sci. Technol., Vol. 20(9):095710, 12pp

[43] Weinmann, K. J., 1969, “Incipient Plastic Deformation in Orthogonal Cutting of 70/30 Brass,” Ph. D. dissertation, University of Illinois

[44] Choi, Y., Park, J. H., Kim, B. M., Choi, J. C., and Min, B. H., 2000, “Estimation of Relation between Effective Strain and Hardness by Rigid-Plastic FEM,” Metals and Materials, Vol. 6(2), pp. 111-116

[45] Sonmez, F. O., and Demir, A., 2007, “Analytical Relations between Hardness and Strain for Cold Formed Parts,” Journal of Material Processing Technology, Vol. 186, pp. 163-173

[46] Chaudhri, M. M., 1996, “Subsurface Plastic Strain Distribution around Spherical Indentations in Metals,” Philosophical Magazine A, Vol. 74(5), pp.1213-1224

[47] Chaudhri, M. M., 1998, “Subsurface Strain Distribution around Vickers Hardness Indentations in Annealed Polycrystalline Copper,” Acta Materialia, Vol. 46(9), pp. 3047-3056

[48] Weinmann, K. J., 1977, “On Wedge Indentation and Its Relationship with Incipient Plastic Deformation in Metal Cutting,” Journal of Engineering for Industry, Vol. 99(3), pp. 702-707

[49] Ben Salem, S., Bayraktar, E., Boujelbene, M., and Katundi, D., 2012, “Effect of Cutting Parameters on Chip Formation in Orthogonal Cutting,” Journal of Achievements in Materials and Manufacturing Engineering, Vol. 50(1), pp. 7-17

[50] Rosa, P. A. R., Kolednik, O., Martins, P. A. F., and Atkins, A. G., 2007, “The Transient Beginning to Machining and the Transition to Steady-State Cutting,” International Journal of Machine Tools & Manufacture, Vol. 47, pp. 1904-1915

[51] Adrian, R. J., and Westerweel, J., 2011, *Particle Image Velocimetry*, Cambridge University Press, New York, NY

[52] Sveen, J. K., and Cowen, E. A., 2004, “Quantitative Image Techniques and Their Application to Wavy Flows,” WSPC, 49pp

[53] Josh, S. S., and Melkote, S. N., 2004, “An Explanation for the Size-Effect in Machining Using Strain Gradient Plasticity,” Journal of Manufacturing Science and Engineering, Vol. 126, pp. 679-684

[54] Arsecularatne, J. A., 1997, “On Tool-Chip Interface Stress Distributions, Ploughing Force and Size Effectt in Machining,” International Journal of Machine Tools & Manufacture, Vol. 37(7), pp. 885-899

[55] Rodriguez, C. J. C., 2009, *Cutting Edge Preparation of Precision Cutting tools by Applying Micro-Abrasive Jet Machining and Brushing*, Kassel University Press, Kassel, Germany

Appendix A: Matlab Code for Velocity Field Detection in PIV

```
%-----------------------------------------------------------------------
%VELOCITY FIELD DETECTING OF DEFORMATION PROCESS THROUGH PIV TECHNIQUE
%Fei Du 06/2012
%-----------------------------------------------------------------------

%-----------------------------------------------------------------------
%PART 1: main framework of processing
%-----------------------------------------------------------------------
function piv()
clear all   % clear all variables

%input analysis parameters
frameNum=200;           %number of frames to process (at least 2)
gridsize=8;             %sampling size
validNum=3;             %number of data validation times
mask_r=8;               %used for generating updated mask
dataFileName='data.mat';%name of the velocity data file
iws=[32,3];             %window size and iteration setup
regionx=[20 492];       %set the region of raw image to process
regiony=[20 460];
ImgSizePxl=512;         %image size (unit: pixel)
ImgSizeReal=2.4;        %image size (unit: mm)
DOC=0.5;                %depth of cut (unit: mm)
DOCPxl=floor(ImgSizePxl*DOC/ImgSizeReal);%calculate the DOC in pixel

%generate grid coordinates
gridx=regionx(1):gridsize:regionx(2);
gridy=regiony(1):gridsize:regiony(2);
%number of grids
m=length(gridy);
n=length(gridx);
%allocate memory for velocity matrix
xdis=zeros(m,n,frameNum-1);
ydis=zeros(m,n,frameNum-1);
%used to identify the masked grid-points
flagpoint=zeros(m,n,frameNum-1);

%select the starting image file
[filename, pathname] = uigetfile('*.TIF','Choose starting image');
[a name ext]=fileparts(filename);
c=isstrprop(name,'digit');
```

```
digitLen=sum(c); %number of digits
fileLen=length(name);

fileBase=name(1:fileLen-digitLen);
startNum=str2double(name(fileLen-digitLen+1:fileLen));
format=strcat('%0',int2str(digitLen),'d');

%load image, convert to number array
[x1,map]=imread(fullfile(pathname,filename));
%load the prepared orginal mask (cover the cutter)
orgmask=imread(fullfile(pathname,'mask.bmp'));
%vectors near the sharp edge are not valided to aviod misreplacements
sharppoint=zeros(m,n);
%show the mask for the first image (not actually used)
[updmask,gridpointin]=geneupdmask(orgmask,x1,mask_r,gridx,gridy,DOCPx
l);
imshow(updmask)
%convert uint8 numbers to double precision numbers
x1d=double(x1);

%process the images pair by pair
for i=1:frameNum-1
    %record the beginning time
    t0=cputime;
    %find the next image
    index=startNum+i;
    [fileIndex errmsg]=sprintf(format,index);
    if ~isempty(errmsg)
        disp(errmsg);
    end;
    filename=strcat(fileBase,fileIndex,ext); %file name of the next image
    disp(['processing image: ',filename]);

    %load the next image into number array
    [x2,map]=imread(fullfile(pathname,filename));
    fx2=filename;
    %update mask (actually used, shared by the image pair)
    [updmask,gridpointin]=geneupdmask(orgmask,x2,mask_r,gridx,gridy,D
    OCPxl);
    x2d=double(x2);

    %image processing to get displacement field (cross-correlation)
    [dx,dy]=image_data_process(x1d,x2d,gridx,gridy,iws,gridpointin,sh
    arppoint);
```

```
    x1d=x2d;

    %data validation (Normalised Median Test)
    for rec=1:validNum
        dx=data_validation1(dx,0.1,sharppoint); %thres can be adjusted
        dy=data_validation1(dy,0.1,sharppoint);
    end

    %save the displacement field for this image
    xdis(:,:,i)=dx;
    ydis(:,:,i)=dy;
    flagpoint(:,:,i)=gridpointin;

    %record the ending time, used to predict the remaining time
    t1=cputime;
    disp(['minutes remaining: ',num2str((t1-t0)*(frameNum-i-1)/60.0)]);
end

save(dataFileName,'gridsize','iws','fx2','pathname','gridx','gridy','
xdis','ydis','orgmask');
save('flagpoint.mat','flagpoint');

clear all;

%-----------------------------------------------------------------------
%PART 2: IMAGE DATA PROCESSING (PIV ANALYSIS)
%-----------------------------------------------------------------------
%CONVENTION: imheight~m~y~i; imwidth~n~x~j
function
[dx,dy]=image_data_process(x1d,x2d,gridx,gridy,iws,gridpointin,sharpp
oint)
%get grid number along x and y direction
m=length(gridy);
n=length(gridx);
[x,y]=meshgrid(gridx,gridy);
dx=zeros(m,n);
dy=zeros(m,n);

%get number of levels of grid refinement
[refineNum,dumb]=size(iws);

for k=1:refineNum
    d=power(2,refineNum-k);
    indexm=1:d:m;
```

```
    indexn=1:d:n;
    %this function implements the cross correlation and peak searching
    [dxtmp,dytmp]=get_vfield(x1d,x2d,gridx(indexn),gridy(indexm),...
    dx(indexm,indexn),dy(indexm,indexn),iws(k,1),gridpointin,sharppoi
    nt);
    i=1;
    while(i<iws(k,2))
      [dxtmp,dytmp]=get_vfield(x1d,x2d,gridx(indexn),gridy(indexm),...
           dxtmp,dytmp,iws(k,1),gridpointin,sharppoint);
        i=i+1;
    end
    if k<refineNum
        dx=interp2(x(indexm,indexn),y(indexm,indexn),dxtmp,x,y);
        dy=interp2(x(indexm,indexn),y(indexm,indexn),dytmp,x,y);
    else
        dx=dxtmp;
        dy=dytmp;
    end
end


function
[dx1,dy1]=get_vfield(x1d,x2d,gridx,gridy,dx,dy,winsize,gridpointin,sh
arppoint)

halfwinsize=winsize/2;
n=length(gridx);
m=length(gridy);
[imm,imn]=size(x1d);
imdx=zeros(imm,imn);
imdy=zeros(imm,imn);

[x,y]=meshgrid(gridx,gridy);
[imx imy]=meshgrid(gridx(1):gridx(n),gridy(1):gridy(m));
[winx,winy]=meshgrid(-halfwinsize:halfwinsize-1,-halfwinsize:halfwins
ize-1);

%interpolate the velocities on the grid-points to each pixel
imdx(gridy(1):gridy(m),gridx(1):gridx(n))=interp2(x,y,dx,imx,imy);
imdy(gridy(1):gridy(m),gridx(1):gridx(n))=interp2(x,y,dy,imx,imy);

for i=2:m-1
    for j=2:n-1
        if gridpointin(i,j)==1  %means inside the masked region
```

```
    dx(i,j)=0;
    dy(i,j)=0;
else
    %edge of the window
    x0=x(i,j)-halfwinsize;
    x1=x(i,j)+halfwinsize-1;
    y0=y(i,j)-halfwinsize;
    y1=y(i,j)+halfwinsize-1;
    %locate a specified window (can move if multiple iterations)
    iwx=x(i,j)+winx+imdx(y0:y1,x0:x1);
    iwy=y(i,j)+winy+imdy(y0:y1,x0:x1);
    %center of the window after movement
    cx=floor(x(i,j)+dx(i,j));
    cy=floor(y(i,j)+dy(i,j));
    %increase the window size, higher velocity limit and accuracy
    gx0=cx-1.5*halfwinsize;
    gx1=cx+1.5*halfwinsize;
    gy0=cy-1.5*halfwinsize;
    gy1=cy+1.5*halfwinsize;
    [iwx0,iwy0]=meshgrid(gx0:gx1,gy0:gy1);

    %find the corresponding window in image 2
    x2dd=interp2(iwx0,iwy0,x2d(gy0:gy1,gx0:gx1),iwx,iwy);

    %calculate the fft correlation
    g1=fft2(x1d(y0:y1,x0:x1));
    g2=fft2(x2dd);
    g1=conj(g1);
    phi=ifft2(g1.*g2);
    phi=fftshift(phi);

    %locate the peak of correlation value
    [aa,bb]=size(phi);
    %return the location of maximum
    [mpeakint,npeakint]=find(max(max(phi))==phi);
    if numel(mpeakint)==1 && numel(npeakint)==1
        mpeakint=mpeakint(1);%in case there are multiple peaks
        npeakint=npeakint(1);
        %in case the peak is on the edge of window
        if mpeakint==1||mpeakint==aa||npeakint==1||npeakint==bb
        dx(i,j)=dx(i,j)+npeakint-halfwinsize-1;
        dy(i,j)=dy(i,j)+mpeakint-halfwinsize-1;
        else
        %locate subpixel location of the peak by gaussian function
```

```
            mpeaksub=(log(phi(mpeakint+1,npeakint))-log(phi(mpeakint...
                -1,npeakint)))/(-2*log(phi(mpeakint-1,npeakint))+...
                4*log(phi(mpeakint,npeakint))-2*log(phi(mpeakint+...
                1,npeakint)))+mpeakint-halfwinsize-1;
            npeaksub=(log(phi(mpeakint,npeakint+1))-log(phi(mpeakint,...
                npeakint-1)))/(-2*log(phi(mpeakint,npeakint-1))+...
                4*log(phi(mpeakint,npeakint))-2*log(phi(mpeakint,...
                npeakint+1)))+npeakint-halfwinsize-1;
                %final vector obtained in this iteration
                dx(i,j)=dx(i,j)+npeaksub;
                dy(i,j)=dy(i,j)+mpeaksub;
                end
             else
                %multiple peaks means incorrectness, give up this iteration
                dx(i,j)=dx(i,j);
                dy(i,j)=dy(i,j);
             end
        end
    end
end

%data validation is needed before next iteration
dx(2:m-1,2:n-1)=data_validation1(dx(2:m-1,2:n-1),0.1,sharppoint(2:m-1
,2:n-1));
dy(2:m-1,2:n-1)=data_validation1(dy(2:m-1,2:n-1),0.1,sharppoint(2:m-1
,2:n-1));
%interpolate edge points
dx(2:m-1,1)=2*dx(2:m-1,2)-dx(2:m-1,3);
dy(2:m-1,1)=2*dy(2:m-1,2)-dy(2:m-1,3);
dx(2:m-1,n)=2*dx(2:m-1,n-1)-dx(2:m-1,n-2);
dy(2:m-1,n)=2*dy(2:m-1,n-1)-dy(2:m-1,n-2);
dx(1,:)=2*dx(2,:)-dx(3,:);
dy(1,:)=2*dy(2,:)-dy(3,:);
dx(m,:)=2*dx(m-1,:)-dx(m-2,:);
dy(m,:)=2*dy(m-1,:)-dy(m-2,:);
dx1=dx;
dy1=dy;
```

Appendix B: Matlab Code for Background Masking in PIV

```
%-------------------------------------------------------------------
%GENERATE UPDATED MASKS FOR IDENTIFYING THE BACKGROUNG REGION IN IMAGES.
%Fei Du 06/2012
%-------------------------------------------------------------------

function
[updmask,gridpointin]=geneupdmask(mask,ima,mask_r,gridx,gridy,DOC)

[my,mx]=find(mask==255);  %find the region of cutter
minmx=min(mx);
minmy=min(my);            %find the cutting edge
edge=zeros(1,size(ima,1));%the boundary btw workpiece and background

BW=im2bw(ima,0.02);%convert gray-scale to binary, threshold can be varied
f=fspecial('disk',mask_r);
BW=imfilter(BW,f,'replicate','same');%1st blurring
BW(mask==255)=0;
BW(:,1:minmx)=1;
f=fspecial('gaussian',mask_r,round(mask_r/2));
BW=imfilter(BW,f,'replicate','same');%2nd blurring

[temp,edge(1)]=min(BW(1,minmx+DOC-30:400));
edge(1)=edge(1)+minmx+DOC-21;%find the boundary of the 1st row
BW(1,edge(1):size(ima,2))=0; %remove all the noise on the background-side
BW(1,1:edge(1)-1)=1;
%find the boundary row by row
for i=2:size(ima,1)
    if i<minmy-30
        [temp,edge(i)]=min(BW(i,edge(i-1):edge(i-1)+30));
        edge(i)=edge(i)+edge(i-1)-1;
        BW(i,edge(i):size(ima,2))=0;
        BW(i,1:edge(i)-1)=1;
    elseif i>=minmy-30 && i<minmy
        [temp,edge(i)]=min(BW(i,edge(i-1)-5:edge(i-1)+30));
        edge(i)=edge(i)+edge(i-1)-6;
        BW(i,edge(i):size(ima,2))=0;
        BW(i,1:edge(i)-1)=1;
    else
        [temp,edge(i)]=min(BW(i,edge(i-1)-10:edge(i-1)+20));
        edge(i)=edge(i)+edge(i-1)-11;
        BW(i,edge(i):size(ima,2))=0;
```

```
    end
end

%black is the workpiece, white is background and cutter
updmask=1-BW;

%record the grid-points inside the masked region
gridpointin=zeros(length(gridy),length(gridx));
for i=1:length(gridy)
    for j=1:length(gridx)
        if BW(gridy(i),gridx(j))==0
            gridpointin(i,j)=1;
        end
    end
end
```

Appendix C: Matlab Code for Data Validation in PIV

```
%-----------------------------------------------------------------------
%DATA VALIDATION AND REPLACEMENT OF VELOCITY FIELD (NORMALIZED MEDIAN
TEST)
%Fei Du 06/2012
%-----------------------------------------------------------------------

function Y=data_validation(X,thresh,exception)

[m,n]=size(X);
validX=zeros(m,n);  %for saving the normalized medians

%validate inner points (5x5)
for i=3:m-2
    for j=3:n-2
        if exception(i,j)==1
        %no test, replaced by the validated value nearby
        X(i,j)=X(i-1,j+1);
        else
        [X(i,j),validX(i,j)]=median_test(X(i-2:i+2,j-2:j+2),9,thresh);
        end
    end
end

%points on the second outer layer (3x3)
for j=2:n-1
    [X(2,j),validX(2,j)]=median_test(X(1:3,j-1:j+1),0,thresh);
    [X(m-1,j),validX(m-1,j)]=median_test(X(m-2:m,j-1:j+1),0,thresh);
end
for i=3:m-2
    [X(i,2),validX(i,2)]=median_test(X(i-1:i+1,1:3),0,thresh);
    [X(i,n-1),validX(i,n-1)]=median_test(X(i-1:i+1,n-2:n),0,thresh);
end

for j=2:n-1
    %upper boundary (3x3)
    [X(1,j),validX(1,j)]=median_test(X(1:3,j-1:j+1),2,thresh);
    %lower boundary (3x3)
    [X(m,j),validX(m,j)]=median_test(X(m-2:m,j-1:j+1),1,thresh);
end

for i=2:m-1
```

```
    %left boundary (3x3)
    [X(i,1),validX(i,1)]=median_test(X(i-1:i+1,1:3),3,thresh);
    %right boundary (3x3)
    [X(i,n),validX(i,n)]=median_test(X(i-1:i+1,n-2:n),4,thresh);
end

%left upper corner
[X(1,1),validX(1,1)]=median_test(X(1:3,1:3),5,thresh);
%right upper corner
[X(1,n),validX(1,n)]=median_test(X(1:3,n-2:n),6,thresh);
%right lower corner
[X(m,n),validX(m,n)]=median_test(X(m-2:m,n-2:n),7,thresh);
%left lower corner
[X(m,1),validX(m,1)]=median_test(X(m-2:m,1:3),8,thresh);
%output the validated values
Y=X;


function [a,valida]=median_test(U,n,thresh)
Un=0;%neighboring points around the center
Uc=0;%center point
switch n
    case 0;%second outer layer (3x3)
        Un=[U(1,1:3),U(2,1),U(2,3),U(3,1:3)];
        Uc=U(2,2);
    case 1;%lower boundary (3x3)
        Un=[U(1,1:3),U(2,1:3),U(3,1),U(3,3)];
        Uc=U(3,2);
    case 2;%upper boundary (3x3)
        Un=[U(1,1),U(1,3),U(2,1:3),U(3,1:3)];
        Uc=U(1,2);
    case 3;%left boundary (3x3)
        Un=[U(1,1:3),U(2,2:3),U(3,1:3)];
        Uc=U(2,1);
    case 4;%right boundary (3x3)
        Un=[U(1,1:3),U(2,1:2),U(3,1:3)];
        Uc=U(2,3);
    case 5;%left upper corner (3x3)
        Un=[U(1,2:3),U(2,1:3),U(3,1:3)];
        Uc=U(1,1);
    case 6;%right upper corner (3x3)
        Un=[U(1,1:2),U(2,1:3),U(3,1:3)];
        Uc=U(1,3);
    case 7;%right lower corner (3x3)
```

```
        Un=[U(1,1:3),U(2,1:3),U(3,1:2)];
        Uc=U(3,3);
    case 8;%left lower corner (3x3)
        Un=[U(1,1:3),U(2,1:3),U(3,2:3)];
        Uc=U(3,1);
    case 9;%inner points (5x5)
        Un=[U(1,1:5),U(2,1:5),U(3,1:2),U(3,4:5),U(4,1:5),U(5,1:5)];
        Uc=U(3,3);
    otherwise
        disp('error');
end

%if pass the test, no need to change the value
%if fail to pass the test, replace the value by local median
Um=median(Un); %local median
if Uc==0
    a=0; %indicate that it is in the masked region, set to zero
    valida=0;
elseif Um==0  %indicate that it is half-masked around the chip end
    a=0; %set to zero to avoid the potential outliers
    valida=0;
else
    updU8=Un(find(Un));  %removed the values in the background
    updUm=median(updU8); %updated local median
    updrm=median(abs(updU8-updUm));   %updated local residual medain
    valida=abs(updUm-Uc)/(updrm+0.1); %normalized median
    if valida<thresh
        a=Uc;       %pass the test
    else a=updUm;  %fail to pass the test, replaced by local median
    end
end
```

Appendix D: Matlab Code for Velocity Field Display in PIV

```
%-----------------------------------------------------------------------
%PRESENTATION OF VELOCITY FIELD AND/OR SAVE THE PRESENTATION AS AVI MOVIE
%COLOR TO INDICATE MAGNITUDE; ARROW TO INDICATE DIRECTION
%Fei Du 06/2012
%-----------------------------------------------------------------------

function showv

FrameRate=30;    %unit is frames/sec
SizeInPxl=512;   %image size (unit: pixel)
SizeInReal=2.4;  %image size (unit: mm)
load('data.mat') %load the velocity data file saved in the main code
aviname='noavi'; %name of the avi movie ('noavi' means no movie)
avispeed=4;      %speed of movie (unit: frames per sec)
ArrowDen=2;      %density of arrows
vmax=0.4;        %set the upper bound for colorbar (magnitude)

Real_Pxl=SizeInReal/SizeInPxl;%mm/pixel
scale=Real_Pxl*FrameRate;     %mm*frame/pixel*sec
xdisR=xdis*scale;           %mm/s=(pixel/frame)*(mm*frame/pixel*sec)
ydisR=ydis*scale;
[m,n,frames]=size(xdis);
[x,y]=meshgrid(gridx,gridy);
[imx,imy]=meshgrid(gridx(1):gridx(n),gridy(1):gridy(m));

%one arrow represents multiple vectors when grid size is small
[x1,y1]=meshgrid(gridx(1:ArrowDen:n),gridy(1:ArrowDen:m));
v=zeros(m,n);
imv=zeros(gridy(m)-gridy(1)+1,gridx(n)-gridx(1)+1);

%show the image
set(0,'defaultfigurecolor','w');
IMA=imread(fullfile(pathname,fx2));
IMA1=IMA(gridy(1):gridy(m),gridx(1):gridx(n));
imshow(IMA1,'XData',[gridx(1) gridx(n)],'YData',[gridy(1) gridy(m)]);

%specify aviname as 'noavi' to skip making avi movie
if strcmp(aviname,'noavi'); bSaveImage=0;
else bSaveImage=1; end

if bSaveImage
```

```
    hfig=figure;
    havi=avifile(aviname,'fps',avispeed,'compression','None');
else
    figure(2);
end

%show the velocity fields one by one
for i=1:frames
    v=sqrt(xdisR(:,:,i).^2+ydisR(:,:,i).^2); %magnitude
    imv=interp2(x,y,v,imx,imy);

    imagesc(gridx(1),gridy(1),imv);
    colormap(hot)
    hold on

    quiver(x1,y1,xdis(1:ArrowDen:m,1:ArrowDen:n,i),ydis(1:ArrowDen:m,
    1:ArrowDen:n,i),0.8,'w');
    caxis([0,vmax]);
    colorbar;
    g=gridsize;
    w=iws(1);
    ch=colorbar;
    xlabel(ch,'mm/s');
    xh=get(ch,'XLabel');
    set(xh,'Position',get(xh,'Position')+[0 1.1*vmax 0]);
    title([fx2,'-copper500','-GS',int2str(g),'-WS',int2str(w),'-V0.3'
    ,'-alpha30-',int2str(i),'/',int2str(frames)]);
    xlabel('x(mm)');
    ylabel('y(mm)');

    %mark out the cutter
    [my,mx]=find(orgmask==255);
    minmx=min(mx);
    minmy=min(my);
    vertex.idx(2)=minmx;vertex.idy(2)=minmy;
    vertex.idx(1)=minmx;vertex.idy(1)=gridy(m);
    vertex.idx(3)=gridx(n);vertex.idy(3)=min(my(mx==gridx(n)));
    h1=plot([vertex.idx],[vertex.idy],'r-o') ;
    set(h1,'LineWidth',2);
    set(h1,'color','w');

    %the second set of coordinates (in pixel)
    h1=gca;
```

```
    set(h1,'YAxisLocation','right','XAxisLocation','top');
    set(h1,'XTickLabel',roundn(100*Real_Pxl,-2):roundn(100*Real_Pxl,-
    2):roundn(410*Real_Pxl,-2));
    set(h1,'YTickLabel',roundn(50*Real_Pxl,-2):roundn(50*Real_Pxl,-2)
    :roundn(460*Real_Pxl,-2));
    h2=axes('Position',get(h1,'Position'));
    set(h2,'Color','none');
    xlabel('x(pixel)');
    ylabel('y(pixel)');
    set(h2,'XTick',get(h1,'XTick'),'YTick',get(h1,'YTick'));
    set(h2,'XTickLabel',100:100:400);
    set(h2,'YTickLabel',50:50:450,'YDir','reverse');
    set(h2,'XLim',get(h1,'XLim'),'Layer','top');
    set(h2,'YLim',get(h1,'YLim'),'Layer','top');

    hold off
    pause(0.5); %how long time each field displays
    drawnow;
    if bSaveImage
        F=getframe(hfig);
        havi=addframe(havi,F);
    end
    if i<frames
    delete(h1);
    delete(h2);
    end
end

if bSaveImage
    close(hfig);
    havi=close(havi);
end
clear all;
```

Appendix E: Matlab Code for Strain Rate Field Calculation in PIV

```
%-----------------------------------------------------------------------
%STRAIN RATE CALCULATION FROM VELOCITY USING CENTRAL DIFFERENCE METHOD
%Fei Du 06/2012
%-----------------------------------------------------------------------

function vf2srf

FrameRate=30;    %unit is frames/sec
SizeInPxl=512;   %image size (unit: pixel)
SizeInReal=2.4;  %image size (unit: mm)
load('data.mat') %load the velocity data file saved in the main code
aviname='noavi'; %name of the avi movie ('noavi' means no movie)
avispeed=4;      %speed of movie (unit: frames per sec)
srmax=0.5;       %set the upper bound for colorbar

Real_Pxl=SizeInReal/SizeInPxl;%mm/pixel
scale=Real_Pxl*FrameRate;     %mm*frame/pixel*sec
xdisR=xdis*scale;            %mm/s=(pixel/frame)*(mm*frame/pixel*sec)
ydisR=ydis*scale;
[m,n,frames]=size(xdis);
gridSize=gridx(2)-gridx(1);
[x,y]=meshgrid(gridx(2:n-1),gridy(2:m-1));
[imx,imy]=meshgrid(gridx(1):gridx(n),gridy(1):gridy(m));
%allocate memory for strain rate matrix
epsilon=zeros(m-2,n-2,frames);
imepsilon=zeros(gridy(m)-gridy(1)+1,gridx(n)-gridx(1)+1,frames);
%Theoretically, there should be no more deformation imposed after a point
%having passed the shear plane (and 2nd deformation zone), but the noise
%strain rate will be still accumulated when computing the strain. In order
%to avoid such an error, only the strain rate values in an interval will
% be taken into account in the stran calculation algorithm.
interval=zeros(1,frames);
%find the cutter region and cutting edge
[my,mx]=find(orgmask==255);
minmx=min(mx);
minmy=min(my);

if strcmp(aviname,'noavi'); bSaveImage=0;
else bSaveImage=1; end

if bSaveImage
```

```
    scrsz = get(0,'ScreenSize');
    hfig=figure;
    havi=avifile(aviname,'fps',avispeed,'compression','None');
end

%calculate the strain rate using central difference method
du1_dx1=zeros(m-2,n-2);
du2_dx1=zeros(m-2,n-2);
du1_dx2=zeros(m-2,n-2);
du2_dx2=zeros(m-2,n-2);
%show the strain rate fields one by one
for i=1:frames
    %in real scale (unit: /sec)
    du1_dx1=(xdisR(2:m-1,3:n,i)-xdisR(2:m-1,1:n-2,i))/(2*gridSize*Rea
    l_Pxl);
    du1_dx2=(xdisR(3:m,2:n-1,i)-xdisR(1:m-2,2:n-1,i))/(2*gridSize*Rea
    l_Pxl);
    du2_dx1=(ydisR(2:m-1,3:n,i)-ydisR(2:m-1,1:n-2,i))/(2*gridSize*Rea
    l_Pxl);
    du2_dx2=(ydisR(3:m,2:n-1,i)-ydisR(1:m-2,2:n-1,i))/(2*gridSize*Rea
    l_Pxl);
    e1=du1_dx1;
    e2=du2_dx2;
    e3=0.5*(du1_dx2+du2_dx1); %shear strain rate

    %convert to effective strain rate
    epsilon(:,:,i)=sqrt((2/9)*((e1-e2).^2+e1.^2+e2.^2)+(4/3)*e3.^2);

    %Some strain rate values (grids) require validation, 'flg' is used
    %to record the type of each grid. 0-need no validation; 1-bad value,
    %adjoin the boundary,have to be replaced; 2-suspected value, next to
    %bad values, normalized median test; 3-exceptional value, in the shear
    %band but near to free surface,no validation; 4-next to exceptional
     %values.
    flg=zeros(m-2,n-2);
    %used to save the original values before median tests
    epsorg=zeros(m-2,n-2);

    %outstanding noise value, set to zero directly
    for j=1:m-2
        for k=1:n-2
            if epsilon(j,k,i)~=0
                if xdis(j+1,k+1,i)==0&&ydis(j+1,k+1,i)==0
                epsilon(j,k,i)=0;
```

```
            end
        end
    end
end
% mark the bad values (adjoin the boundary, inside the workpiece)
for j=1:m-2
    for k=1:n-2
        if epsilon(j,k,i)~=0
            if
(xdis(j+1,k+2,i)==0&&ydis(j+1,k+2,i)==0)||(xdis(j+2,k+1,i)==0&&ydis(j
+2,k+1,i)==0)||...
(xdis(j,k+1,i)==0&&ydis(j,k+1,i)==0)||(xdis(j+2,k+2,i)==0&&ydis(j+2,k
+2,i)==0)||...(xdis(j,k+2,i)==0&&ydis(j,k+2,i)==0)
            flg(j,k)=1;
            epsilon(j,k,i)=nan; %set to NAN temporarily
            end
        end
    end
end
% mark the suspected values (not bad, adjoin the bad values)
for j=1:m-2
    for k=1:n-2
        if epsilon(j,k,i)~=0 && flg(j,k)==0
            if
flg(max(1,j-1),k)==1||flg(max(1,j-1),min(n-2,k+1))==1||flg(j,min(n-2,
k+1))==1||...
flg(min(m-2,j+1),k)==1||flg(min(m-2,j+1),min(n-2,k+1))==1||flg(j,min(
n-2,k+2))==1||...
                flg(min(m-2,j+2),k)==1
               %mark the exceptional values (in shear band, near surface)
                if j>5 && j<m/2 && k>5 && k<n/2
                    if
(epsilon(j+1,k,i)>0.4||epsilon(j+1,k-1,i)>0.4)&&...
(epsilon(j+2,k-1,i)>0.4||epsilon(j+2,k-2,i)>0.4)
                        flg(j,k)=3;
                    else
                        flg(j,k)=2;
                        %save original value
                        epsorg(j,k,i)=epsilon(j,k,i);
                        epsilon(j,k,i)=nan;   %set to NAN temporarily
                    end
                else
                    flg(j,k)=2;
                    epsorg(j,k,i)=epsilon(j,k,i);
```

```
                epsilon(j,k,i)=nan;
            end
        end
    end
end
% mark the suspected values next to Type 3 values
for j=3:m-4
    for k=3:n-4
        if flg(j,k)==2
            if flg(j-2,k-2)==3||flg(j-1,k-2)==3||flg(j,k-2)==3||...
                flg(j-2,k-1)==3||flg(j-1,k-1)==3||flg(j,k-1)==3
                flg(j,k)=4;
            end
        end
    end
end

% deal with the obvious noises inside the undeformed workpiece
for j=4:m-2
    for k=1:n/2
        if gridx(k+3)<minmx&&epsilon(j,k,i)>0.04
    epsilon(j,k,i)=min(mean(mean(epsilon(j-3:j-1,k:k+1,i))),0.04);
        end
    end
end
% deal with Type 2 values: normalized median test (ignoring bad values)
for j=1:m-2
    for k=1:n-2
        if flg(j,k)==2
           epsilon(j,k,i)=sr_validation(j,k,epsilon(:,:,i),epsorg);
        end
    end
end
% deal with Type 4 values: no test, replaced by the value nearby inside
shear plane
for k=1:n-2
    for j=1:m-2
        if flg(j,k)==4
            epsilon(j,k,i)=epsilon(j+1,k-1,i);
            if isnan(epsilon(j,k,i))
                epsilon(j,k,i)=epsorg(j,k,i);
            end
        end
```

```
    end
end
% deal with Type 1 values: replaced by the mean of neighboring values
(ignoring bad values but using the validated Type 2 values)
for j=1:m-2
    for k=1:n-2
        if flg(j,k)==1
           epsilon(j,k,i)=sr_validation(j,k,epsilon(:,:,i),epsorg);
        end
    end
end

%Validated strain rate field
imepsilon(:,:,i)=interp2(x,y,epsilon(:,:,i),imx,imy);
%Considering 0.3 as the edge of shear plane
[shearY,shearX]=find(epsilon(:,:,i)>0.3); %find the shear plane
if numel(shearY)~=0
    %interval determination: max(right end of primary DZ, right end
    of 2nd DZ)
    interval(i)=max(floor((minmx-gridx(1))/gridSize)+15,max(shearX
    )+3);
else
    %for the first few images, DZ hasn't yet completely formed
    interval(i)=n-2;
end

%show the field
imagesc(gridx(1),gridy(1),imepsilon(:,:,i));
colormap(hot)
hold on
caxis([0,srmax]);
colorbar;
ch=colorbar;
xlabel(ch,'1/s');
xh=get(ch,'XLabel');
set(xh,'Position',get(xh,'Position')+[0 1.1*srmax 0]);
g=gridsize;
w=iws(1);
title([fx2,'-copper500','-GS',int2str(g),'-WS',int2str(w),'-V0.3'
,'-alpha30-',int2str(i),'/',int2str(frames)]);
xlabel('x(mm)');
ylabel('y(mm)');

%mark out the cutter
```

```
    vertex.idx(2)=minmx;vertex.idy(2)=minmy;
    vertex.idx(1)=minmx;vertex.idy(1)=gridy(m);
    vertex.idx(3)=gridx(n);vertex.idy(3)=min(my(mx==gridx(n)));
    h1=plot([vertex.idx],[vertex.idy],'r-o') ;
    set(h1,'LineWidth',2);
    set(h1,'color','w');

    %the second set of coordinates (in pixel)
    h1=gca;
    set(h1,'YAxisLocation','right','XAxisLocation','top');
    set(h1,'XTickLabel',roundn(100*Real_Pxl,-2):roundn(100*Real_Pxl,-
    2):roundn(410*Real_Pxl,-2));
    set(h1,'YTickLabel',roundn(50*Real_Pxl,-2):roundn(50*Real_Pxl,-2)
    :roundn(460*Real_Pxl,-2));
    h2=axes('Position',get(h1,'Position'));
    set(h2,'Color','none');
    xlabel('x(pixel)');
    ylabel('y(pixel)');
    set(h2,'XTick',get(h1,'XTick'),'YTick',get(h1,'YTick'));
    set(h2,'XTickLabel',100:100:400);
    set(h2,'YTickLabel',50:50:450,'YDir','reverse');
    set(h2,'XLim',get(h1,'XLim'),'Layer','top');
    set(h2,'YLim',get(h1,'YLim'),'Layer','top');

    hold off
    pause(0.5);  %how long time each field displays
    if bSaveImage
        F=getframe(hfig);
        havi=addframe(havi,F);
    end
    if i<frames
    delete(h1);
    delete(h2);
    end
end

save('srfieldRS.mat','epsilon','flg','interval');

if bSaveImage
    close(hfig);
    havi=close(havi);
end
```

```
%normalized median test (or direct replacement)
function a=sr_validation(j,k,E,Eorg)
[M,N]=size(E); %M=m-2,N=n-2
Ec=Eorg(j,k);  %center value, to be tested
%3x3 region used as neigbors
%inner points
if j>1 && j<M && k>1 && k<N
    E8=[E(j-1,k-1:k+1),E(j,k-1),E(j,k+1),E(j+1,k-1:k+1)];
%upper edge
elseif j==1 && k>1 && k<N
    E8=[E(1,k-1),E(1,k+1),E(2,k-1:k+1)];
%lower edge
elseif j==M && k>1 && k<N
    E8=[E(M-1,k-1:k+1),E(M,k-1),E(M,k+1)];
%left edge
elseif j>1 && j<M && k==1
    E8=[E(j-1,1:2),E(j,2),E(j+1,1:2)];
%right edge
elseif j>1 && j<M && k==N
    E8=[E(j-1,N-1:N),E(j,N-1),E(j+1,N-1:N)];
%left upper corner
elseif j==1 && k==1
    E8=[E(1,2),E(2,1:2)];
%right upper corner
elseif j==1 && k==N
    E8=[E(1,N),E(2,N-1:N)];
%left lower corner
elseif j==M && k==1
    E8=[E(M-1,1:2),E(M,2)];
%right lower corner
elseif j==M && k==N
    E8=[E(M-1,N-1:N),E(M,N-1)];
end

Em=nanmedian(E8); %local median ignoring bad values
if Ec~=0          %indicate this is a Type 2 value, perform the test
    rm=nanmedian(abs(E8-Em));  %local residual median ignoring bad values
    if isnan(Em)
        a=0;
    elseif abs(Em-Ec)/(rm+0.1)<0.5 %threshold is 0.5 in this test
        a=Ec;       %pass the test
    else
        a=Em;       %fail and replace
    end
```

```
elseif isnan(Em)    %indicate this is a Type 1 (bad) value, replace directly
    a=0;
else a=Em;
end
```

Appendix F: Matlab Code for Strain Field Calculation in PIV

```
%-----------------------------------------------------------------------
%STRAIN CALCULATION BASED ON THE POINT TRACKING AND STRAIN RATE HISTORY
%Fei Du 08/2012
%-----------------------------------------------------------------------

function Sfield

FrameRate=30;          %unit is frames/sec
SizeInPxl=512;         %image size (unit: pixel)
SizeInReal=2.4;        %image size (unit: mm)
load('data.mat')       %load the velocity data file
load('srfieldRS.mat') %load the strain rate data file
smax=1.5;              %set the upper bound for colorbar

scale=1/FrameRate;              %sec/frame
Real_Pxl=SizeInReal/SizeInPxl; %mm/pixel
[m,n,frames]=size(xdis);
[x,y]=meshgrid(gridx(2:n-1),gridy(2:m-1));
[imx,imy]=meshgrid(gridx(1):gridx(n),gridy(1):gridy(m));
gridsize=gridx(2)-gridx(1);
Strain=zeros(m-2,n-2);          %allocate memory for strain matrix
imStrain=zeros(gridy(m)-gridx(1)+1,gridx(n)-gridx(1)+1);

%If we have N frames of velocity fields, we need to backtrack N-1 steps.
%This step is from the last image (Nth) to the previous one (N-1th)
for j=2:m-1
    for k=2:n-1
        px(j,k,1)=gridx(k); %location of this point in the last image (px)
        py(j,k,1)=gridy(j);
        gridix(j,k,1)=k;   %corresponding id number of the grid
        gridiy(j,k,1)=j;

        %velocity vector and incremental strain for this grid-point at this
         step
        vx(j,k,1)=xdis(gridiy(j,k,1),gridix(j,k,1),frames);
        vy(j,k,1)=ydis(gridiy(j,k,1),gridix(j,k,1),frames);
        sr(j,k,1)=epsilon(gridiy(j,k,1)-1,gridix(j,k,1)-1,frames);
        %imposed strain at this step (unit: /frame)
        deltaS(j-1,k-1,1)=sr(j,k,1)*scale;
    end
end
```

```
%Backtrack the steps one by one
for i=2:frames
    for j=2:m-1
        for k=2:n-1
            %where is this point in the (N-1th) image
            px(j,k,i)=px(j,k,i-1)-vx(j,k,i-1);
            py(j,k,i)=py(j,k,i-1)-vy(j,k,i-1);
            %which grid does this point belong to in this image
            if
(px(j,k,i)-gridx(1))/gridsize-floor((px(j,k,i)-gridx(1))/gridsize)<0.
5
            grididx(j,k,i)=floor((px(j,k,i)-gridx(1))/gridsize)+1;
            else
            grididx(j,k,i)=floor((px(j,k,i)-gridx(1))/gridsize)+2;
            end
            if
(py(j,k,i)-gridy(1))/gridsize-floor((py(j,k,i)-gridy(1))/gridsize)<0.
5
            grididy(j,k,i)=floor((py(j,k,i)-gridy(1))/gridsize)+1;
            else
            grididy(j,k,i)=floor((py(j,k,i)-gridy(1))/gridsize)+2;
            end

            %consider the point has left the field of view from this step
            if
grididx(j,k,i)<2||grididx(j,k,i)>n-1||grididy(j,k,i)<2||grididy(j,k,i
)>m-1
               vx(j,k,i)=0;
               vy(j,k,i)=0;
               deltaS(j-1,k-1,i)=0;
            %consider the point far away from deformation zone in this step
            elseif grididx(j,k,i)>interval(i)
              vx(j,k,i)=xdis(grididy(j,k,i),grididx(j,k,i),frames-i+1);
              vy(j,k,i)=ydis(grididy(j,k,i),grididx(j,k,i),frames-i+1);
                deltaS(j-1,k-1,i)=0;
            %obtain the velocity of this point at this step, calculate the
             imposed strain
            else
              vx(j,k,i)=xdis(grididy(j,k,i),grididx(j,k,i),frames-i+1);
              vy(j,k,i)=ydis(grididy(j,k,i),grididx(j,k,i),frames-i+1);
     sr(j,k,i)=epsilon(grididy(j,k,i)-1,grididx(j,k,i)-1,frames-i+1);
               deltaS(j-1,k-1,i)=sr(j,k,i)*scale;
            end
```

```
        end
    end
end

%Eventually, accumulate the imposed strain for each effective step.
for j=1:m-2
    for k=1:n-2
        for i=1:frames
            Strain(j,k)=Strain(j,k)+deltaS(j,k,i);
        end
    end
end

%show the field
imStrain=interp2(x,y,Strain,imx,imy);
imagesc(gridx(1),gridy(1),imStrain);
colormap(hot)
hold on
caxis([0,smax]);
colorbar;
g=gridsize;
w=iws(1);
title([fx2,'-copper500','-GS',int2str(g),'-WS',int2str(w),'-V0.3','-a
lpha30-','strain field']);
xlabel('x(mm)');
ylabel('y(mm)');

%mark out the cutter
[my,mx]=find(orgmask==255);
minmx=min(mx);
minmy=min(my);
vertex.idx(2)=minmx;vertex.idy(2)=minmy;
vertex.idx(1)=minmx;vertex.idy(1)=gridy(m);
vertex.idx(3)=gridx(n);vertex.idy(3)=min(my(mx==gridx(n)));
h1=plot([vertex.idx],[vertex.idy],'r-o') ;
set(h1,'LineWidth',2);
set(h1,'color','w');

%the second set of coordinates (in pixel)
h1=gca;
set(h1,'YAxisLocation','right','XAxisLocation','top');
set(h1,'XTickLabel',roundn(100*Real_Pxl,-2):roundn(100*Real_Pxl,-2):r
oundn(410*Real_Pxl,-2));
```

```
set(h1,'YTickLabel',roundn(50*Real_Pxl,-2):roundn(50*Real_Pxl,-2):rou
ndn(460*Real_Pxl,-2));
h2=axes('Position',get(h1,'Position'));
set(h2,'Color','none');
xlabel('x(pixel)');
ylabel('y(pixel)');
set(h2,'XTick',get(h1,'XTick'),'YTick',get(h1,'YTick'));
set(h2,'XTickLabel',100:100:400);
set(h2,'YTickLabel',50:50:450,'YDir','reverse');
set(h2,'XLim',get(h1,'XLim'),'Layer','top');
set(h2,'YLim',get(h1,'YLim'),'Layer','top');
hold off

save('SfieldRS.mat','Strain');
```

Appendix G: Expected Deformation Fields for Artificial Images

Window size = 20 px, Width of deformation zone = 80 px

1) $v_w = v_d = v_c = 2, u_d = 1, u_c = 2$

Using central difference method in the deformation zone:

$$d\varepsilon_{xx} = \frac{\partial u}{\partial x} = 0, \;\; d\varepsilon_{yy} = \frac{\partial v}{\partial y} = 0, \;\; d\gamma = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \frac{1}{2 \times 20} + 0 = \frac{1}{40}$$

Expected inter-frame effective strain increment (strain rate) in the deformation zone:

$$d\bar{\varepsilon} = \sqrt{\frac{4}{9}\left(\frac{1}{2}\left[(d\varepsilon_{xx} - d\varepsilon_{yy})^2 + {d\varepsilon_{xx}}^2 + {d\varepsilon_{yy}}^2\right] + \frac{3}{4}d\gamma^2\right)} = \sqrt{\frac{4}{9}\left(\frac{3}{4} \times \left(\frac{1}{40}\right)^2\right)} = \mathbf{0.01444}$$

Number of frames to pass the deformation zone:

$$\frac{80}{2} = 40$$

Effective strain in the chip:

$$40 \times 0.01443 = \mathbf{0.578}$$

2) $v_w = v_d = v_c = 4, u_d = 2, u_c = 4$

Using central difference method in the deformation zone:

$$d\varepsilon_{xx} = \frac{\partial u}{\partial x} = 0, \;\; d\varepsilon_{yy} = \frac{\partial v}{\partial y} = 0, \;\; d\gamma = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \frac{2}{2 \times 20} + 0 = \frac{1}{20}$$

Expected inter-frame effective strain increment (strain rate) in the deformation zone:

$$d\bar{\varepsilon} = \sqrt{\frac{4}{9}\left(\frac{1}{2}\left[(d\varepsilon_{xx} - d\varepsilon_{yy})^2 + {d\varepsilon_{xx}}^2 + {d\varepsilon_{yy}}^2\right] + \frac{3}{4}d\gamma^2\right)} = \sqrt{\frac{4}{9}\left(\frac{3}{4} \times \left(\frac{1}{20}\right)^2\right)} = \mathbf{0.02887}$$

Number of frames to pass the deformation zone:

$$\frac{80}{4} = 20$$

Effective strain in the chip:

$$20 \times 0.02887 = \mathbf{0.578}$$

3) $v_w = v_d = v_c = 6, u_d = 3, u_c = 6$

Using central difference method in the deformation zone:

$$d\varepsilon_{xx} = \frac{\partial u}{\partial x} = 0, \;\; d\varepsilon_{yy} = \frac{\partial v}{\partial y} = 0, \;\; d\gamma = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \frac{3}{2 \times 20} + 0 = \frac{3}{40}$$

Expected inter-frame effective strain increment (strain rate) in the deformation zone:

$$d\bar{\varepsilon} = \sqrt{\frac{4}{9}\left(\frac{1}{2}\left[(d\varepsilon_{xx} - d\varepsilon_{yy})^2 + {d\varepsilon_{xx}}^2 + {d\varepsilon_{yy}}^2\right] + \frac{3}{4}d\gamma^2\right)} = \sqrt{\frac{4}{9}\left(\frac{3}{4} \times \left(\frac{3}{40}\right)^2\right)} = \mathbf{0.04331}$$

Mean number of frames to pass the deformation zone:

$$\frac{80}{6} = 13.33$$

Effective strain in the chip:

$$13.33 \times 0.04331 = \mathbf{0.578}$$

4) $v_w = v_d = v_c = 8, u_d = 4, u_c = 8$

Using central difference method in the deformation zone:

$$d\varepsilon_{xx} = \frac{\partial u}{\partial x} = 0, \;\; d\varepsilon_{yy} = \frac{\partial v}{\partial y} = 0, \;\; d\gamma = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \frac{4}{2 \times 20} + 0 = \frac{1}{10}$$

Expected inter-frame effective strain increment (strain rate) in the deformation zone:

$$d\bar{\varepsilon} = \sqrt{\frac{4}{9}\left(\frac{1}{2}\left[(d\varepsilon_{xx} - d\varepsilon_{yy})^2 + d\varepsilon_{xx}{}^2 + d\varepsilon_{yy}{}^2\right] + \frac{3}{4}d\gamma^2\right)} = \sqrt{\frac{4}{9}\left(\frac{3}{4} \times \left(\frac{1}{10}\right)^2\right)} = \mathbf{0.05774}$$

Number of frames to pass the deformation zone:

$$\frac{80}{8} = 10$$

Effective strain in the chip:

$$10 \times 0.05774 = \mathbf{0.578}$$

Appendix H: Raw Experimental Data

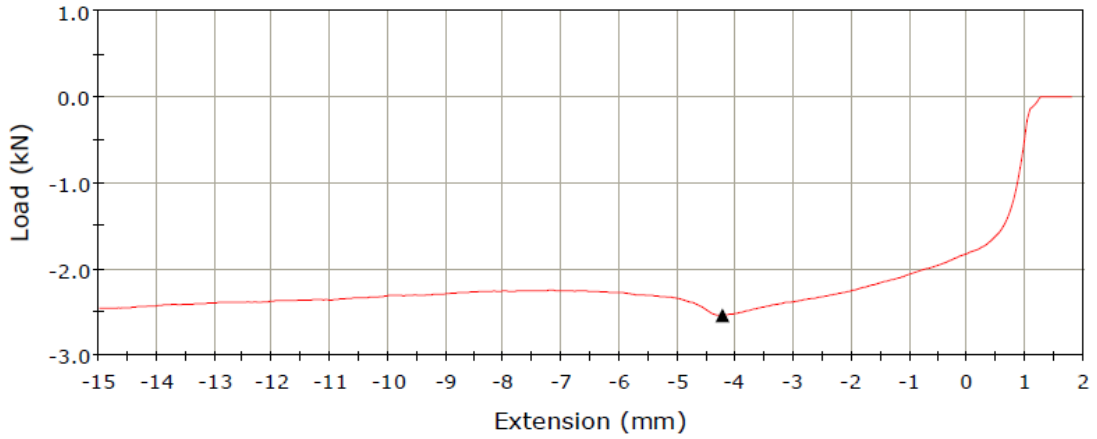


Figure H.1: Raw cutting load curve: $\alpha = 30°, a_0 = 750\mu m$

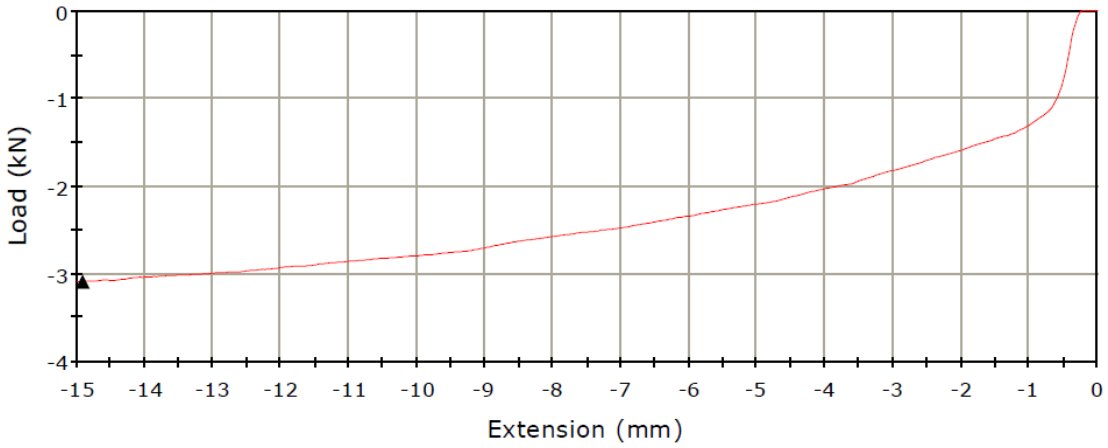


Figure H.2: Raw cutting load curve: $\alpha = 30°, a_0 = 500\mu m$

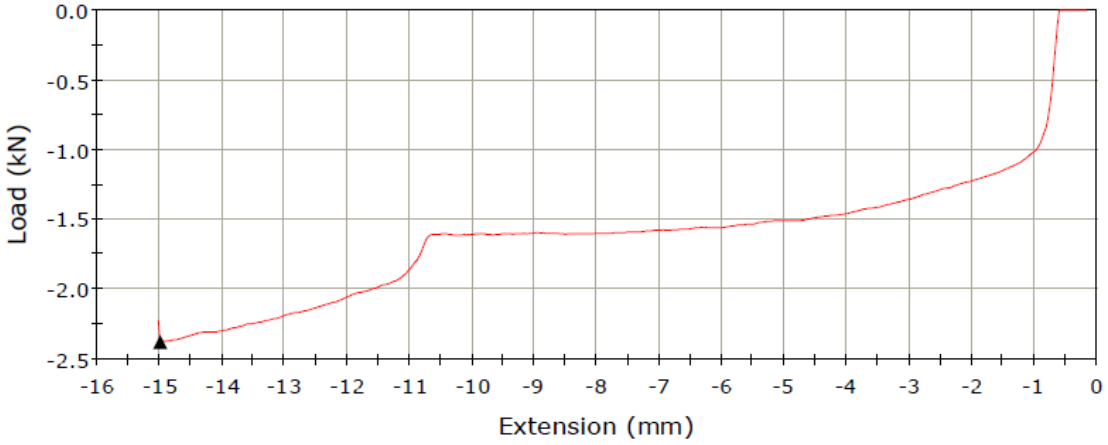


Figure H.3: Raw cutting load curve: $\alpha = 30°, a_0 = 250\mu m$

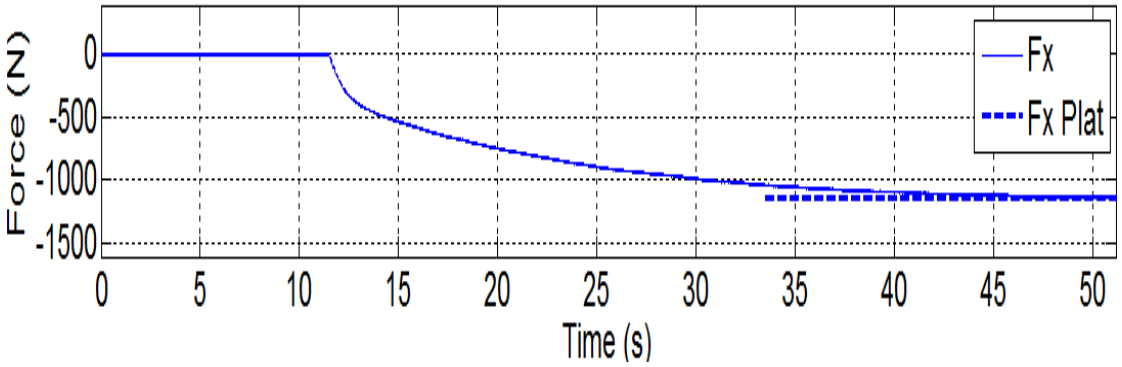


Figure H.4: Raw cutting load curve: $\alpha = 30°, a_0 = 200\mu m$

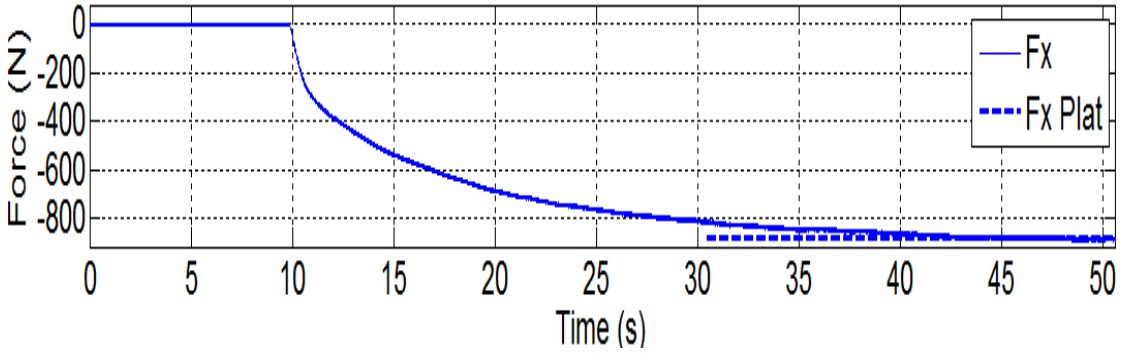


Figure H.5: Raw cutting load curve: $\alpha = 30°, a_0 = 150\mu m$

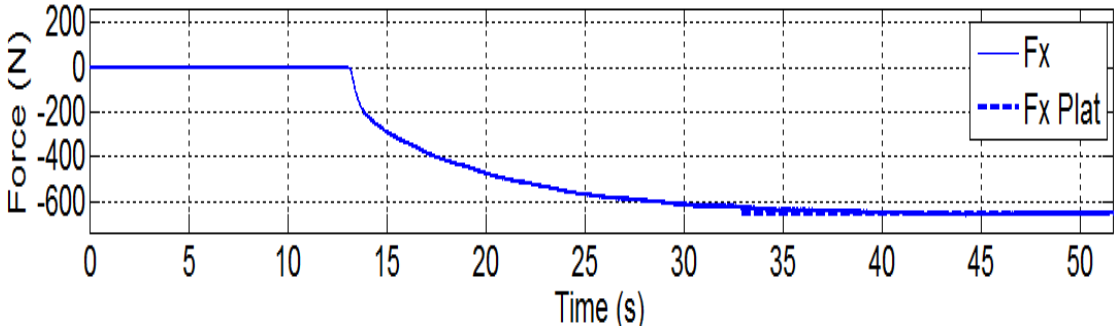


Figure H.6: Raw cutting load curve: $\alpha = 30°, a_0 = 100\mu m$

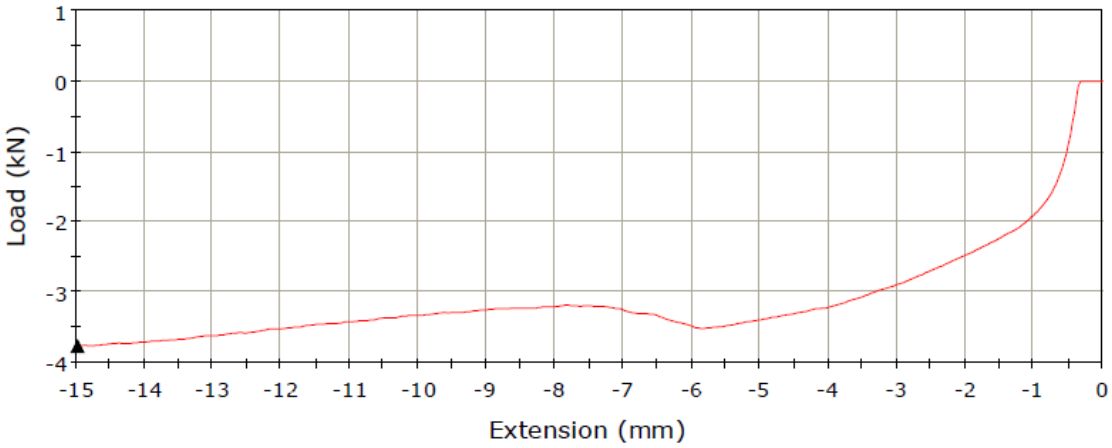


Figure H.7: Raw cutting load curve: $\alpha = 15°, a_0 = 750\mu m$

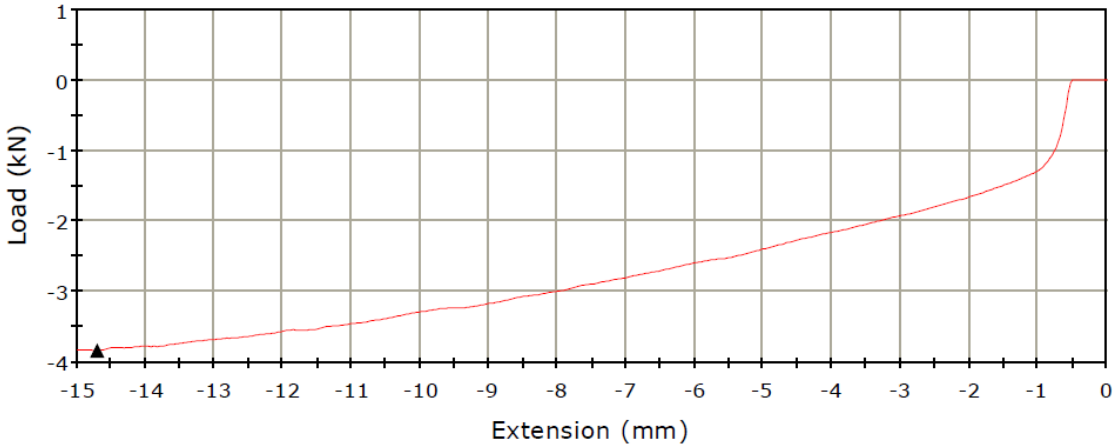


Figure H.8: Raw cutting load curve: $\alpha = 15°, a_0 = 500\mu m$

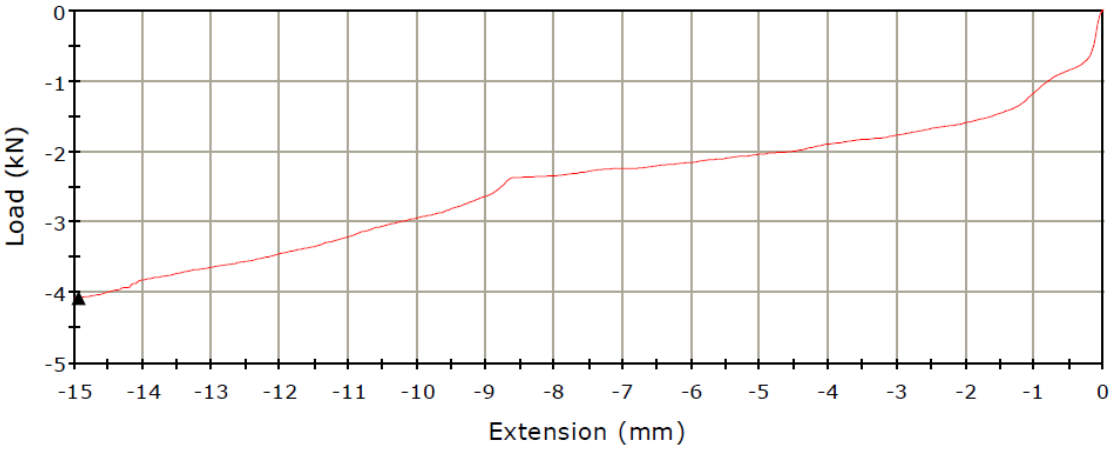


Figure H.9: Raw cutting load curve: $\alpha = 15°, a_0 = 250\mu m$

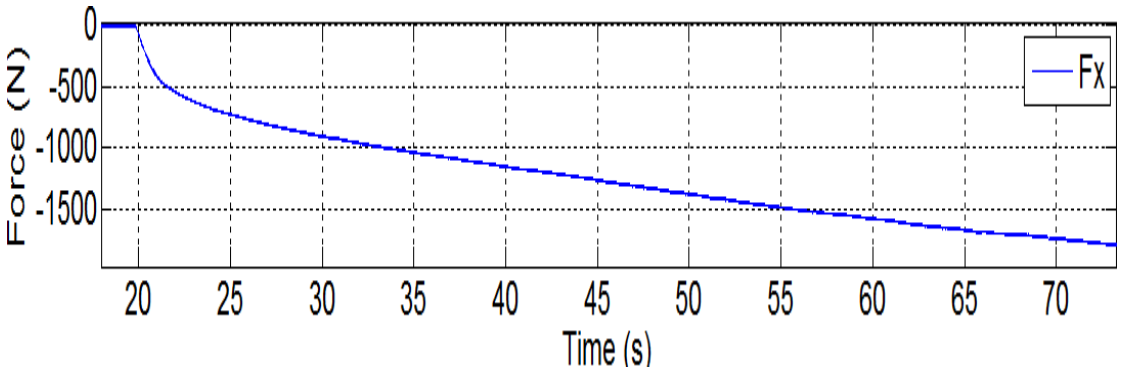


Figure H.10: Raw cutting load curve: $\alpha = 15°, a_0 = 200\mu m$

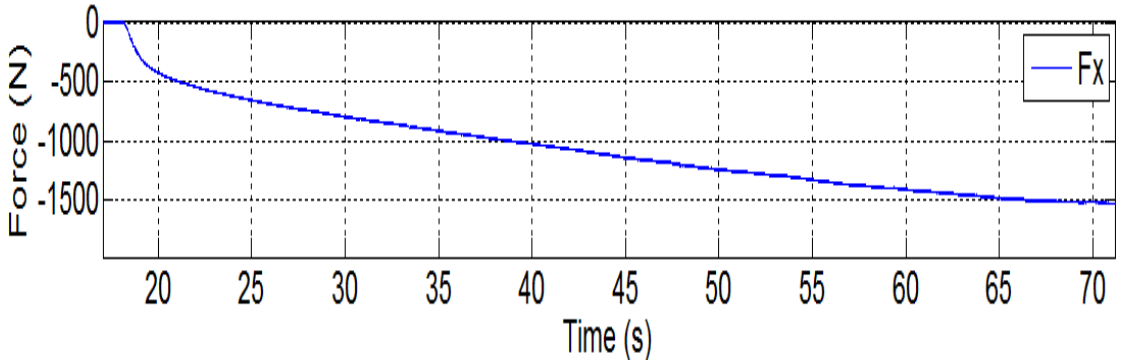


Figure H.11: Raw cutting load curve: $\alpha = 15°, a_0 = 150\mu m$

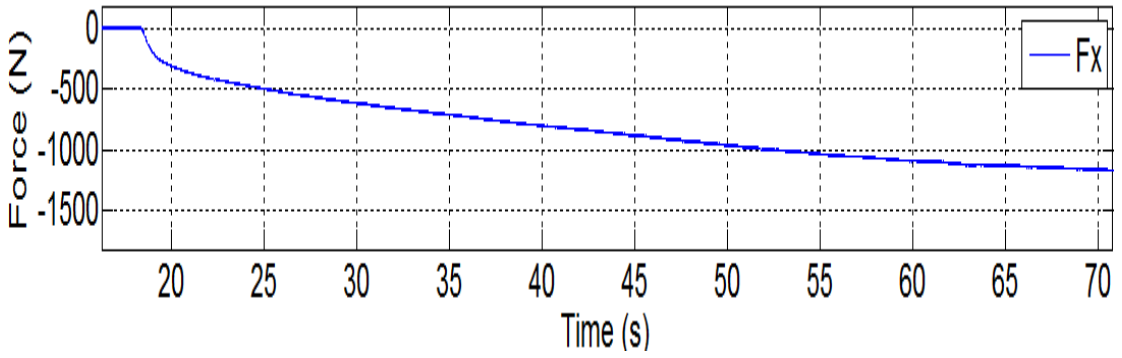


Figure H.12: Raw cutting load curve: $\alpha = 15°, a_0 = 100\mu m$

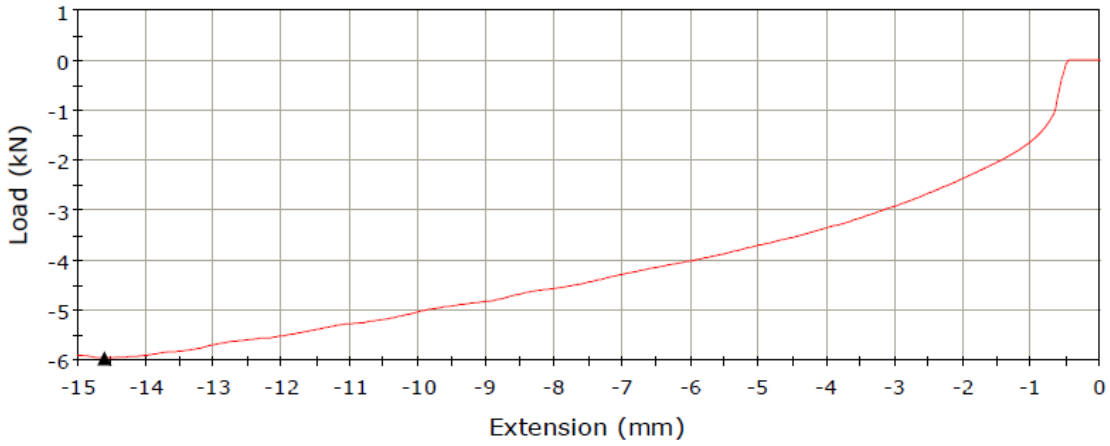


Figure H.13: Raw cutting load curve: $\alpha = 5°, a_0 = 500\mu m$

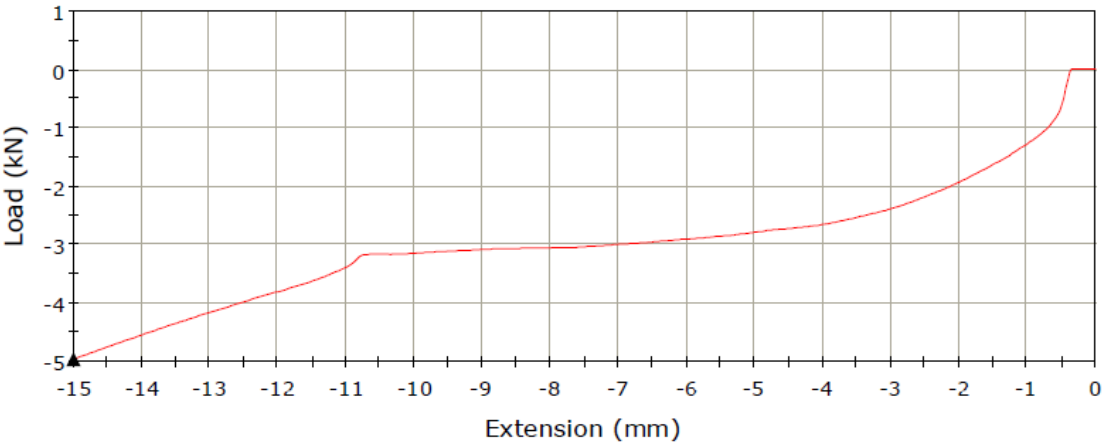


Figure H.14: Raw cutting load curve: $\alpha = 5°, a_0 = 250\mu m$

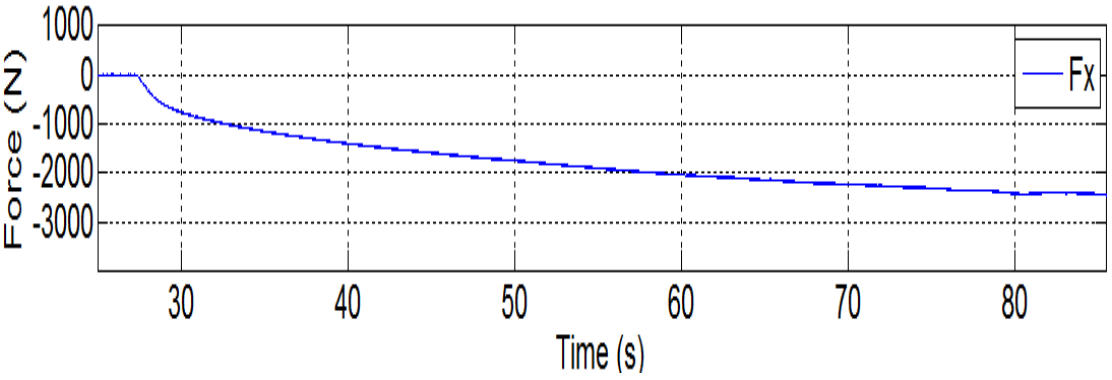


Figure H.15: Raw cutting load curve: $\alpha = 5°, a_0 = 200\mu m$

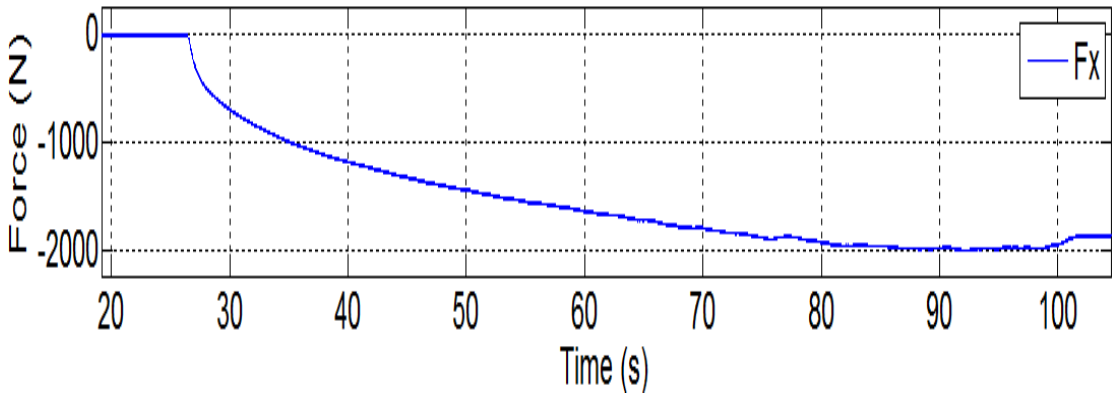


Figure H.16: Raw cutting load curve: $\alpha = 5°, a_0 = 150\mu m$

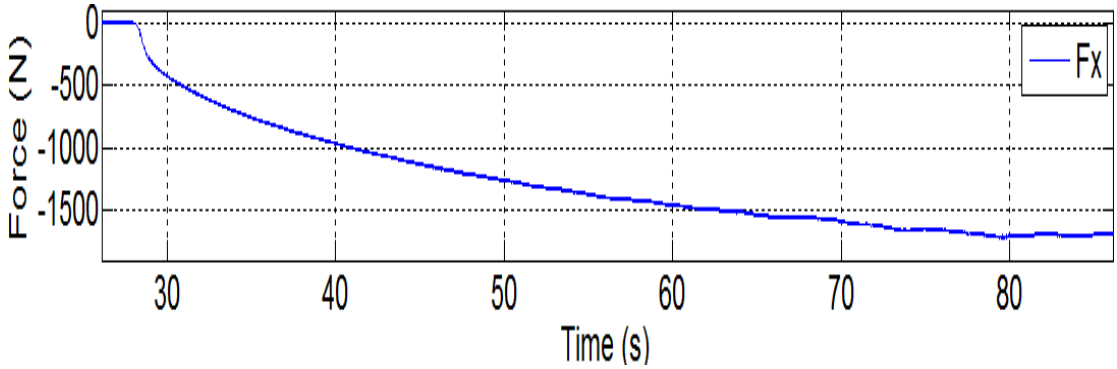


Figure H.17: Raw cutting load curve: $\alpha = 5°, a_0 = 100\mu m$

Table H.1: Raw data of PIV result and hardness test result: $\alpha = 30°, a_0 = 750\mu m$

Distance from chip end (μm)	Original hardness test result (HV)	Hardness-converted strain	Fitting result	Distance from chip end (μm)	PIV strain	Fitting result
202.1	93	0.1687	0.1538	207.8	0.1096	0.1060
288.7	93	0.1687	0.1745	317.5	0.1206	0.1133
404.1	94	0.1781	0.2219	427.2	0.1417	0.1314
519.6	106	0.3280	0.3034	536.9	0.1550	0.1735
635.1	114	0.4748	0.4267	646.6	0.2548	0.2630
750.6	118	0.5657	0.5820	756.3	0.4280	0.4196
866.0	122	0.6985	0.7378	866.0	0.6604	0.6185
981.5	127	0.8553	0.8624	975.7	0.7688	0.7895
1097.0	132	1.0000	0.9451	1085.4	0.8728	0.8938
1212.4	132	1.0000	0.9934	1195.1	0.9471	0.9447
1327.9	133	1.0391	1.0196	1304.8	1.0217	0.9667
1443.4	133	1.0391	1.0332	1414.5	0.9576	0.9759
1558.8	133	1.0391	1.0401	1524.2	0.9823	0.9795
1674.3	132	1.0000	1.0436	1633.9	0.9690	0.9810

Table H.2: Raw data of PIV result and hardness test result: $\alpha = 30°, a_0 = 250\mu m$

Distance from chip end (μm)	Original hardness test result (HV)	Hardness-converted strain	Fitting result	Distance from chip end (μm)	PIV strain	Fitting result
115.5	110	0.3960	0.4040	115.5	0.4366	0.4412
173.2	112	0.4339	0.4249	173.2	0.4258	0.4423
246.0	116	0.5186	0.5202	230.9	0.4702	0.4527
318.7	124	0.7278	0.7282	288.7	0.5365	0.5305
404.1	129	0.8897	0.8963	346.4	0.8012	0.8102
476.9	131	0.9621	0.9327	404.1	1.0200	1.0135
549.6	130	0.9254	0.9402	461.9	1.1026	1.0517
635.1	130	0.9254	0.9417	519.6	1.0669	1.0564
721.7	130	0.9254	0.9420	577.4	1.0605	1.0569
808.3	131	0.9621	0.9420	635.1	1.0125	1.0570
				692.8	1.0319	1.0570
				750.6	1.0965	1.0570
				808.3	1.0193	1.0570

Table H.3: Raw data of hardness test result: $\alpha = 30°, a_0 = 200, 150 \,\&\, 100\mu m$

Distance from chip end (μm)	Original hardness test result (HV)	Hardness-converted strain	Fitting result
	Depth of cut = 200		
115.5	84	0.1006	0.0973
202.1	104	0.2978	0.3044
259.8	117	0.5657	0.5622
317.5	125	0.7582	0.7306
375.3	124	0.7278	0.7892
433.0	126	0.7895	0.8049
490.7	126	0.7895	0.8088
548.5	126	0.7895	0.8097
606.2	128	0.8553	0.8099
664.0	128	0.8553	0.8100
	Depth of cut = 150		
86.6	93	0.1687	0.1699
130.5	97	0.2090	0.2060
173.2	106	0.3280	0.3335
217.1	119	0.5905	0.5913
259.8	126	0.7895	0.7762
303.7	127	0.8218	0.8388
346.4	126	0.7895	0.8530
404.1	128	0.8553	0.8565
461.9	128	0.8553	0.8569
519.6	130	0.9254	0.8570
	Depth of cut = 100		
72.7	106	0.3280	0.3282
115.5	112	0.4339	0.4303
159.3	119	0.6161	0.6128
202.1	127	0.8218	0.7781
244.8	127	0.8218	0.8639
288.7	129	0.8897	0.8959
331.4	129	0.8897	0.9057
389.1	129	0.8897	0.9092
446.9	131	0.9254	0.9098

** In ($\alpha = 30°, a_0 = 100\mu m$) experiment, all the hardness measurements are taken $100\mu m$ away from the bottom edge of the chip instead of $150\mu m$, because the chip is too narrow in the beginning part.*

Table H.4: Raw data of PIV result and hardness test result: $\alpha = 15°, a_0 = 750\mu m$

Distance from chip end (μm)	Original hardness test result (HV)	Hardness-converted strain	Fitting result	Distance from chip end (μm)	PIV strain	Fitting result
155.3	88	0.1274	0.1394	200.8	0.0955	0.1125
258.8	93	0.1687	0.1744	301.3	0.1064	0.1256
362.4	100	0.2440	0.2303	401.7	0.1669	0.1517
465.9	104	0.2978	0.3151	502.1	0.223	0.2019
569.4	114	0.4748	0.4327	602.5	0.3155	0.2920
672.9	119	0.5905	0.5781	703.0	0.4389	0.4351
776.5	123	0.6985	0.7340	803.4	0.622	0.6236
880.0	127	0.8218	0.8776	903.8	0.8104	0.8190
983.5	132	1.0000	0.9925	1004.2	0.9827	0.9771
1087.1	135	1.1210	1.0745	1104.7	1.14	1.0813
1190.6	136	1.1638	1.1284	1205.1	1.1863	1.1411
1294.1	137	1.2079	1.1619	1305.5	1.2155	1.1727
1397.7	135	1.1210	1.1820	1405.9	1.1802	1.1887
1501.2	136	1.1638	1.1938	1506.4	1.1248	1.1965

Table H.5: Raw data of PIV result and hardness test result: $\alpha = 15°, a_0 = 500\mu m$

Distance from chip end (μm)	Original hardness test result (HV)	Hardness-converted strain	Fitting result	Distance from chip end (μm)	PIV strain	Fitting result
129.4	88	0.1274	0.1100	165.6	0.1101	0.1089
232.9	94	0.1781	0.2021	248.5	0.1737	0.1662
336.5	106	0.3280	0.3469	331.3	0.2584	0.2596
414.1	116	0.5186	0.4889	414.1	0.3802	0.3977
517.7	123	0.6985	0.6989	496.9	0.5743	0.5743
621.2	129	0.8897	0.8906	579.8	0.7658	0.7630
724.7	132	1.0000	1.0327	662.6	0.9265	0.9294
828.2	135	1.1210	1.1224	745.4	1.0931	1.0532
931.8	137	1.2079	1.1736	828.2	1.1318	1.1341
1035.3	137	1.2079	1.2010	911.1	1.1255	1.1826
1138.8	137	1.2079	1.2153	993.9	1.1449	1.2101
1242.4	137	1.2079	1.2226	1076.7	1.2604	1.2253
				1159.5	1.2497	1.2335
				1242.4	1.2707	1.2379

Table H.6: Raw data of PIV result and hardness test result: $\alpha = 15°, a_0 = 250\mu m$

Distance from chip end (μm)	Original hardness test result (HV)	Hardness-converted strain	Fitting result	Distance from chip end (μm)	PIV strain	Fitting result
129.4	93	0.1687	0.2190	124.2	0.1898	0.1994
194.6	103	0.2835	0.2619	186.4	0.3242	0.3131
258.8	111	0.4146	0.3607	248.5	0.4916	0.4815
324.0	117	0.5418	0.5500	310.6	0.6498	0.6805
388.2	124	0.7278	0.7911	372.7	0.8638	0.8615
453.5	132	1.0000	0.9901	434.8	1.0201	0.9911
517.7	136	1.1638	1.0976	496.9	1.0381	1.0686
595.3	136	1.1638	1.1500	559.1	1.1084	1.1102
672.9	137	1.2079	1.1668	621.2	1.1397	1.1312
750.6	135	1.1210	1.1718	683.3	1.1824	1.1415
828.2	135	1.1210	1.1734	745.4	1.1007	1.1465
				807.5	1.1544	1.1488

Table H.7: Raw data of hardness test results: $\alpha = 15°, a_0 = 200, 150 \,\&\, 100\mu m$

Distance from chip end (μm)	Original hardness test result (HV)	Hardness-converted strain	Fitting result
	Depth of cut = 200		
103.5	91	0.1511	0.1567
168.8	105	0.3126	0.3064
232.9	118	0.5657	0.5585
298.2	128	0.8553	0.8558
362.4	133	1.0391	1.0753
427.6	138	1.2534	1.1940
491.8	138	1.2534	1.2448
557.0	138	1.2534	1.2655
621.2	138	1.2534	1.2732
	Depth of cut = 150		
91.1	89	0.1349	0.1386
155.3	114	0.4748	0.4559
207.1	124	0.7278	0.7425
258.8	131	0.9621	0.9729
310.6	136	1.1638	1.1175
362.4	136	1.1638	1.1949
440.0	138	1.2534	1.2431
517.7	138	1.2534	1.2583
	Depth of cut = 100		
77.6	100	0.2440	0.2588
117.0	116	0.5186	0.4816
155.3	123	0.6985	0.7129
194.6	129	0.8897	0.9139
232.9	133	1.0391	1.0498
272.3	136	1.1638	1.1339
324.0	138	1.2534	1.1893
375.8	136	1.1638	1.2124
427.6	137	1.2079	1.2218

** In ($\alpha = 15°, a_0 = 100\mu m$) experiment, all the hardness measurements are taken $100\mu m$ away from the bottom edge of the chip instead of $150\mu m$, because the chip is too narrow in the beginning part.*

Table H.8: Raw data of PIV result and hardness test result: $\alpha = 5°, a_0 = 500\mu m$

Distance from chip end (μm)	Original hardness test result (HV)	Hardness-converted strain	Fitting result	Distance from chip end (μm)	PIV strain	Fitting result
150.6	110	0.3960	0.3632	190.7	0.1675	0.1439
251.0	113	0.4540	0.4386	266.0	0.2531	0.2584
351.3	115	0.4963	0.5547	341.3	0.3994	0.4041
451.7	123	0.6985	0.7191	416.6	0.5673	0.5790
552.1	131	0.9621	0.9260	491.9	0.7623	0.7745
652.5	137	1.2079	1.1513	567.1	0.9927	0.9768
752.9	141	1.3982	1.3609	642.4	1.2301	1.1698
853.2	143	1.5019	1.5293	717.7	1.3664	1.3403
953.6	145	1.6117	1.6493	793.0	1.4807	1.4810
1054.0	147	1.7279	1.7277	868.3	1.5930	1.5907
1154.4	148	1.7885	1.7761	943.6	1.6641	1.6725
1254.8	148	1.7885	1.8049	1018.9	1.6767	1.7315
1355.1	149	1.8507	1.8216	1094.1	1.7356	1.7731
1455.5	149	1.8507	1.8313	1169.4	1.8055	1.8018
				1244.7	1.9125	1.8214
				1320.0	1.8488	1.8348
				1395.3	1.8294	1.8437

Table H.9: Raw data of PIV result and hardness test result: $\alpha = 5°, a_0 = 250\mu m$

Distance from chip end (μm)	Original hardness test result (HV)	Hardness-converted strain	Fitting result	Distance from chip end (μm)	PIV strain	Fitting result
125.5	108	0.3607	0.3309	196.7	0.3745	0.4332
225.9	113	0.4540	0.5301	253.0	0.6439	0.6411
326.2	128	0.8553	0.8121	309.2	0.8911	0.8569
401.5	134	1.0794	1.0457	365.4	1.0720	1.0636
476.8	137	1.2079	1.2587	421.6	1.2561	1.2470
552.1	141	1.3982	1.4262	477.8	1.3389	1.3990
627.4	144	1.5560	1.5433	534.0	1.4546	1.5179
727.8	146	1.6690	1.6365	590.2	1.6384	1.6069
828.1	146	1.6690	1.6838	646.4	1.6969	1.6714
903.4	146	1.6690	1.7025	702.7	1.7405	1.7168
978.7	147	1.7279	1.7131	758.9	1.7516	1.7484
1054.0	147	1.7279	1.7192	815.1	1.7940	1.7700
				871.3	1.8033	1.7847
				927.5	1.7700	1.7947
				983.7	1.7606	1.8014

Table H.10: Raw data of hardness test results: $\alpha = 5°, a_0 = 200, 150 \,\&\, 100 \mu m$

Distance from chip end (μm)	Original hardness test result (HV)	Hardness-converted strain	Fitting result
	Depth of cut = 200		
100.4	103	0.2835	0.2496
163.6	111	0.4146	0.4660
225.9	123	0.6985	0.7244
289.1	134	1.0794	0.9998
351.3	138	1.2534	1.2437
414.6	141	1.3982	1.4379
476.8	144	1.5560	1.5721
540.0	146	1.6690	1.6605
602.3	147	1.7279	1.7141
677.6	147	1.7279	1.7511
752.9	148	1.7885	1.7705
828.1	148	1.7885	1.7805
	Depth of cut = 150		
88.3	104	0.2978	0.3046
150.6	115	0.4963	0.4788
200.8	122	0.6701	0.6818
251.0	130	0.9254	0.9249
301.1	136	1.1638	1.1705
351.3	141	1.3982	1.3797
401.5	143	1.5019	1.5335
451.7	146	1.6690	1.6348
514.9	147	1.7279	1.7080
577.2	146	1.6690	1.7445
640.4	148	1.7885	1.7626
702.7	148	1.7885	1.7712
	Depth of cut = 100		
75.3	115	0.4963	0.4878
113.4	123	0.6985	0.7233
150.6	131	0.9621	0.9513
188.7	136	1.1638	1.1669
225.9	140	1.3485	1.3470
264.0	143	1.5019	1.4962
301.1	145	1.6117	1.6078
351.3	146	1.6690	1.7139
401.5	148	1.7885	1.7816
451.7	149	1.8507	1.8234
501.9	149	1.8507	1.8487
552.1	149	1.8507	1.8639