Modeling and Experimental Analysis of Superplastic Forming and Diffusion Bonding

Bryan Ferguson

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington
2020

Reading Committee:
Ramulu Mamidala, Chair
Daniel Sanders
Junlan Wang

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Modeling and Experimental Analysis of Superplastic Forming and Diffusion Bonding

Bryan Ferguson

Chair of Supervisory Committee:

Ramulu Mamidala

Department of Mechanical Engineering

Superplastic forming and diffusion bonding are well-established, but not fully understood, manufacturing processes. They both operate at high temperatures and achieve extremely large failure strains and near flawless welding respectively. Superplastic forming and diffusion bonding have been implemented using a variety of materials and can be used to create complex multi-sheet structures. There has been a significant amount of both experimental and theoretical work done on both processes. However, there has yet to be a definitive and accurate model for diffusion bonding and the effects of more complex sheet formation. This work is aimed at alleviating this knowledge gap. This work specifically deals with two and four sheet bonding of titanium. Several models were created with increasing complexity. Work started with more complex initial conditions applied to conventional bonding models. Then models were created that used conventional physics but with relaxed assumptions about the void shape to see how the voids changed without restrictions. Finally, micrographs of voids were digitized and used as

inputs into an energy-based phase field simulation that allowed each of the alloying elements to diffuse independently, accounting for grain growth, as well as void closure due to strain energy. Superplasticity modeling was done using finite element methods in two and three dimensions to examine the deformation of the four-sheet structure. Research was also done to determine how one can better destructively and non-destructively test diffusion bonded specimens. The results showed that, while superplasticity and diffusion bonding are complex subjects, there is a lot of information to gain through experiments and modeling of the process. This research specifically highlighted the performance of different alloys during SPF/DB, the challenges with structural testing of diffusion bonding, the changes in the surface due to superplasticity and its effects on diffusion bonding, the inadequacies of conventional diffusion bonding modeling, theoretically how the surface could effect diffusion bonding, and how voids diffusively close from a phase field perspective.

Table of Contents

Chapter 1 Introduction

1.1 Background

In the aerospace industry weight savings and cost reduction are important. Reducing the weight of the airplane reduces fuel consumption and associated emissions as well as allowing for increased load carrying capability reducing cost for the customer. Cost to the manufacturer to achieve the weight savings involved is also important and encompasses the purchased material required to make the final part (buy-to-fly ratio), the equipment and labor costs for manufacturing, and other sources of overhead. Therefore, finding manufacturing processes that produce dimensionally accurate parts that fully utilize the material supplied without significant post-processing is highly desirable. A manufacturing process like that reduces the labor, manufacturing costs, and material costs since there is no need for additional trimming of the part to the correct dimensions. For these reasons, the aerospace industry utilizes superplastic forming and diffusion bonding of titanium [1].

1.2 Titanium Alloys

Titanium alloys are often used in the aerospace industry for their strength to weight ratio, good corrosion resistance, and creep resistance. There are a variety of alloys to choose from each with different advantages. However, despite its excellent properties as a metal, titanium is expensive. The same properties that give it corrosion resistance, its affinity to bind to oxygen forming a thick oxide layer, also prevents the source oxide ores from readily disassociating from oxygen with conventional methods. Therefore, titanium is smelted with an expensive batch process using hazardous elements. Once refined from its ore, titanium's strength and toughness along with its

relatively low heat conductivity create problems for nearly every conventional machining technique available further increasing costs. This is why titanium is used in superplastic forming and diffusion bonding which makes efficient use of the material and doesn't require significant machining afterward.

1.3 Superplastic Forming

Superplasticity is a property among fine grained materials that exhibit large strains at certain temperatures and strain rates with no appreciable necking or thickness variation. It also exhibits low variations in material properties over the course of the process. It is most commonly implemented as a sheet metal forming operation. As such it is capable of tight corners and features that would be impossible with conventional sheet forming. The dominant deformation mechanism in superplasticity is grain boundary sliding which is why it is only observed in fine grained materials. It also gives superplasticity one of its most desirable characteristics: negligible residuals stresses and negligible spring back after forming. This allows for near net shape manufacturing and gives the finished part the dimensional tolerance of the forming die used. Many crystalline materials display superplasticity, both ceramics and metals, and among them is titanium. Titanium alloys have a naturally fine crystal structure which allows them to have good forming properties while retaining their excellent material properties that make them viable for aerospace applications [2].

1.4 Diffusion Bonding

Diffusion bonding is a solid state joining process whereby two flat surfaces are joined together. When done properly, it produces a weld over an area that retains the nearly the same properties as

the base materials. The only discernable defect is a series of joined crystals at the bond line that have grown together as a result of the process. Diffusion bonding occurs at a high temperature, similar to superplastic temperatures, and under a pressure that is large enough to help close the voids but not large enough to cause any substantial dislocations (typically 1-3MPa in titanium). Prior to bonding, microscopic ridges due to the surface roughness of the materials form a series of connected areas and voids. The voids are considered the primary defect involved in the process and must be closed in order to obtain a good joint. Voids are closed by a combination of superplastic collapse of the material and diffusional flow. At the right temperature and pressure diffusion bonding can be done with any geometry making it possible to form complex structures with superplasticity and then weld them together with diffusion bonding [3].

1.5 Objectives

Superplasticity and diffusion bonding of titanium has been the topic of extensive research over the years but there are still complexities left to understand. Much of the research done has been on the manufacturing process itself in order to establish the exact temperatures and pressures needed to form different structures. A significant amount of research has been done on the mechanisms of superplasticity and how to effectively model the process [4]. Diffusion bonding has had many experiments conducted as well and some models have been developed. However, these models fail to accurately predict bonding behavior in most cases. There has been especially little research on the topic of dissimilar alloy diffusion bonding. Dissimilar bonding is important to the industry for its ability to create a sheet with two surfaces having different properties. Due to the expense of titanium and the expense of the process itself, trial and error research and development to determine the performance of bonding with new alloys is time consuming and expensive.

Therefore, this research will focus on developing a model that can accurately predict diffusion bonding processing conditions for a variety of dissimilar titanium alloy joints and combine that model with a superplasticity model to predict overall manufacturing behavior. This will help alleviate research and help bring the complex forms of the superplastic forming and diffusion bonding (SPF/DB) process to a level ready for mass manufacturing and widespread implementation. This research will also address some of the issues with testing diffusion bonding.

This thesis is organized into chapters of the major categories that are being discussed. Chapter 2 discusses the necessary background information about superplastic forming, diffusion bonding, titanium, and the numerical methods used in modeling. Chapter 3 explains the research methodology and plan used to go about this research. Chapter 4 covers the work done solely on diffusion bonding. Chapter 5 covers the work done on combined diffusion bonding and superplastic forming. Chapter 6 covers the advanced modeling work done. Chapter 7 discusses the research and draws correlations between different chapters. Chapter 8 is a summary with recommendations for future work.

Chapter 2 Background and Literature Review

2.1 Introduction

The literature review required for this research covers several topics. A review of titanium and its alloys is necessary to understand the effects and classifications of dissimilar alloys. An overview of superplasticity and diffusion bonding and how it pertains to titanium is presented next. Following that are modeling approaches to solving SPF/DB problems including molecular dynamics. Finally, there is a review of the popular structural testing methods for diffusion bonding using ultrasonic non-destructive evaluation and shear testing.

2.2 Titanium and its Alloys

2.2.1 Titanium

The element titanium was first discovered in 1790 and was first reformed into a metal in 1906. However, it wasn't until 1940 that commercial production of the metal began in the United States by the development and implementation of the Kroll process. Titanium has many beneficial properties that make it ideal for aerospace applications. Its high temperature corrosion properties combined with its high strength make it practically necessary for some engine components. It also has many applications outside of the aerospace industry that include bicycles, gold clubs, and medical implants [5].

Economically viable titanium mainly comes in the form of rutile, an oxide with high titanium content. Titanium bonds extremely well to oxygen and as a result it requires significant extractive metallurgy to remove it from oxygen at high temperature in an inert environment. There are two commercially viable processes available at the moment: the Kroll process and the Hunter process. The main difference between the two is the element used to bind the chlorine away from titanium. These processes are done in batches and start with titanium oxide ore. This is converted to titanium tetrachloride via chlorine gas. The titanium tetrachloride is reduced with a metal to produce a salt. The metal is magnesium for the Kroll process and sodium for the Hunter process. The titanium is then purified by distillation and acid leaching to remove the excess $MgCl_2$ or $NaCl_2$ to get pure titanium sponge. The sponge is then crushed and combined with other elements then melted to form the desired titanium alloy. The high operating costs involved in this process makes commercial titanium expensive. However, due to its properties of high strength to weight and long service life in corrosive environments it is still used in the aerospace industry despite the costs.

2.2.2 Alloying Elements

Titanium is a transition metal with a moderate density and median atomic diameter when compared with other elements. It is allotropic, meaning that it has multiple solid-state crystal structures that occur in different temperature ranges. This results in favorable alloying with many different elements. The relevant crystal structures of titanium are classified as alpha (hexagonal close packed, HCP) and beta (body center cubic, BCC). The alloys produced are classified by the main crystal structure that is present at room temperature. In commercially pure titanium, only the alpha phase is present at room temperature and the beta phase is present above the transition temperature (well above normal operating temperatures). However, various alloying elements that are added are able to stabilize the different phases and are capable of maintaining the beta phase at room temperature. Therefore, the alloys are classified into alpha alloys, alpha+beta alloys, and beta alloys corresponding to the ratio of alpha to beta present. Each alloy class has distinctive characteristics and desirable properties. The transition temperature between the two phases is called the beta transus temperature and varies with alloying elements. It is the lowest temperature at which 100% of the metal exists in the BCC beta phase.

Alloying elements are classified into either alpha stabilizers, beta stabilizers, or neutral additions. The alpha stabilizers increase the temperature at which the alpha phase can exist (increasing the beta transus temperature). The beta stabilizers decrease the beta transus temperature. Neutral

additions have no effect on the crystal structure stability. Each alloying element has additional effects on the mechanical properties of the resulting alloy.

Alpha stabilizing	Beta isomorphous	Beta eutectoid	Neutral
Aluminum	Vanadium	Copper	Zirconium
Gallium	Niobium	Silver	Hafnium
Germanium	Tantalum	Gold	Tin
Lanthanum	Molybdenum	Indium	…
Cerium	Rhenium	Lead	…
Oxygen	…	Bismuth	…
Nitrogen	…	Chromium	…
Carbon	…	Tungsten	…
…	…	Manganese	…
…	…	Iron	…
…	…	Cobalt	…
…	…	Nickel	…
…	…	Uranium	…
…	…	Hydrogen	…
…	…	Silicon	…

Figure 2.1 - Classification of various alloying elements [5]

Alpha stabilizers increase the temperature where the alpha phases exist because these elements preferentially dissolve in the hexagonal close packed crystal structure. The predominant alpha stabilizers are aluminum followed by nitrogen, carbon, and oxygen. Aluminum is present in many titanium alloys because the ambient yield strength is greatly increased by the addition of it with relatively little effect on the overall ductility. Figure 2 below displays the yield strength increase with negligible ductility reduction gained by alloying aluminum with titanium. The interstitial alloying elements are able to increase strength by limiting the movement of dislocations at the expense of ductility. They are not used as often in modern alloys with the exception of oxygen. Their influence on alpha-beta alloys is complex because the solubility is more in the alpha phase than the beta phase.

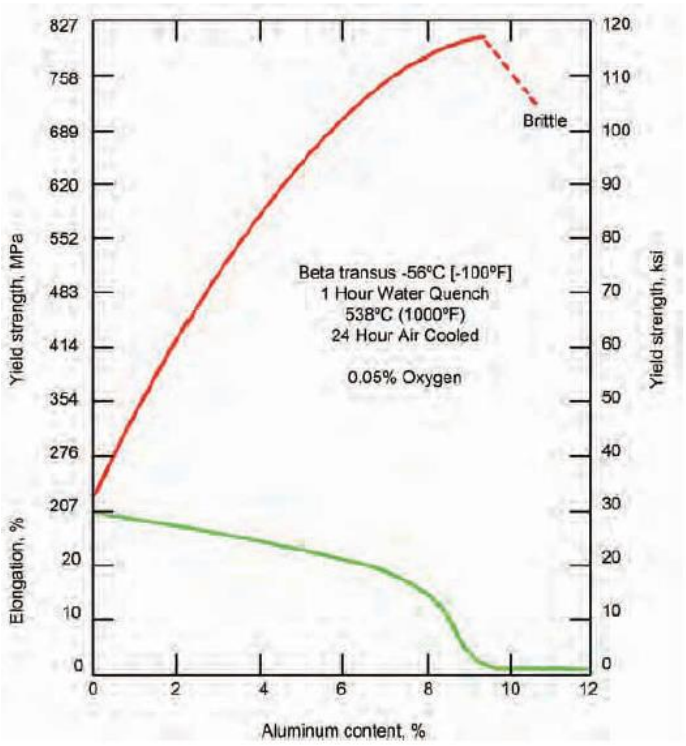


Figure 2.2 – Aluminum content increases the yield strength significantly with negligible decrease in ductility. The green line above shows that ductility reduction with increasing aluminum is small while the red line shows that the yield strength increases significantly. [5]

Beta stabilizers are divided among two groups based on their performance. There are beta isomorphous stabilizers which effectively retain the beta phase at temperatures lower than the beta transus temperature. This is achieved by the decomposition of the beta phase being so slow that for their operational lifetime they can be treated as isomorphous. The most used elements in this group are vanadium, niobium, and molybdenum. The other group of stabilizers is the active eutectoid which are rarely used due to their thermal instability where they rapidly change into other phases. The main exception is silicon which is commonly used in small quantities due to it being effective in enhancing creep resistance.

Neutral additions neither increase nor decrease the beta transus temperature but do enhance the properties of the alloy. Zirconium and tin are commonly used neutral additions. They still have a weak stabilizing effect for beta and alpha respectively but not enough to significantly influence the alloy. Their main advantages are greater strength and creep resistance [5].

2.2.3 Alloy Classifications

Titanium alloys are classified in to alpha, near alpha, alpha+beta, near beta, and beta alloys. This is because each different group has characteristic properties. These properties emerge as a result of the additions of alloying elements that stabilize the different crystal structure systems present.

Alpha and Near Alpha Alloys

The alpha and near alpha alloys have predominantly hexagonal close packed (HCP) crystal structure. Aluminum is the dominant alloying element in alpha alloys. These alloys are not heat treatable since the alpha phase exists regardless of cooling rate. They are typically used in their annealed state and they have low tensile strengths compared with the other groups of titanium alloys. They are typically not as formable as the other alloys because the HCP structure is not as ductile as the body center cubic (BCC) structure. They do have good welding properties because of the alpha phase stability and fair fabrication properties. The main benefit to alpha alloys is their high temperature creep resistance. The other main benefit is their good fracture toughness. These combine to make them excellent in high temperature engine applications where good creep resistance and good fatigue properties are necessary. The near alpha alloys have a small amount of beta phase in them giving them better ductility than the pure alpha alloys. The main alloying elements in alpha alloys are aluminum and tin but are usually combined with a small amount of beta stabilizers for improved mechanical properties.

Alpha + Beta Alloys

The combination of alpha and beta phases represent a middle ground between the properties of alpha and beta alloys. These alloys can be heat treated to improve their mechanical properties after forming. However, because of the beta phase they are not very weldable unless the amount of beta stabilizers are very low. They have good toughness, fatigue crack growth resistance, and fair fabricability. The alpha+beta alloys contain Ti-6Al-4V which is the most common titanium alloy used in the aerospace industry.

Beta Alloys

Beta alloys are the smallest group of alloys and are characterized by having nearly 100% of their grains be BCC in structure. Beta alloys are very high in strength relative to the other titanium alloys and capable of large deformations. They are formable in the annealed condition due to the ductile beta phase. The excellent strength properties are due to a stress based transformation from beta to alpha before fracture and can be heat treated to even higher levels. The heat treatment changes some of the beta phase into alpha and results in comparable strength to some alpha-beta alloys. The density of the beta phase is higher than alpha so the strength to weight ratio is somewhat lessened [5]. Beta alloys are very weldable, have good toughness and fatigue resistance, and have excellent workability relative to other titanium alloys. The downside of beta alloys are that they have poor creep resistance relative to alpha alloys so they are not commonly used in higher temperature situations. [5]

2.3 Superplastic Forming Literature Overview

2.3.1 Superplasticity

Superplasticity is defined as the ability for a polycrystalline material to achieve very high strains before failure where subjected to a small range of temperatures and low deformation rates [6]. It is exhibited in both metals and ceramics or a combination of the two (intermetallics, metal matrix composites). Superplastic forming is typically feasible only under very low strain rates and in materials that are very fine grained. The history of superplasticity dates back to 1912 with Bengough and was first formed in 1964 in the MIT labs under the direction of Backofen, Avery, and Turner. They blew a bubble in ZnAl sheet and noted that this could be used in forming operations [3]. Superplastic forming has now been implemented in a variety of industries with the dominant use being in the aerospace and automotive industries. A prime example of the extreme performance of superplasticity is shown in Figure 2.3 where a CuAl alloy obtained a record 8000% elongation.

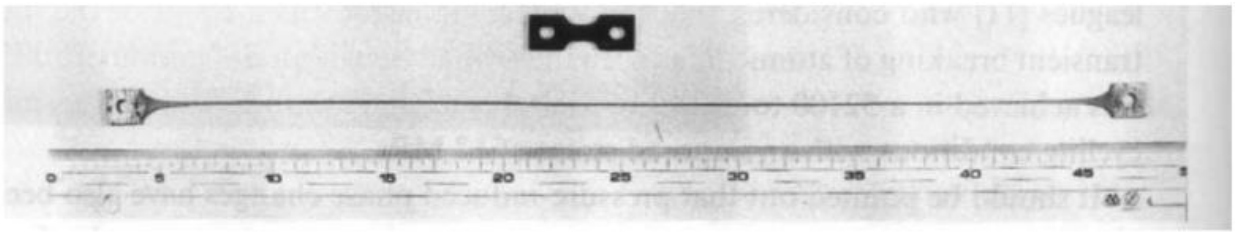


Figure 2.3 - 8000% elongation of CuAl alloy [4]

2.3.2 Mechanisms and Process Parameters of Superplasticity

The exact mechanisms of superplasticity are still not fully understood and have been found to depend on the material system. However, the consensus is that the majority of the strain comes from grain boundary sliding. Superplastic forming (SPF) is heavily dependent on temperature and the typical temperatures are above 50% of the melting point of the material with strain rates being very low and are typically on the order of $1x10^{-3}$/s as shown in Figure 2.4. Titanium alloys have a fine grained equiaxed microstructure that is ideal for superplastic forming as processed from the mill [4]. This gives titanium alloys an advantage over other materials since many materials require special processing to be superplastic. Superplasticity is measured by the standardized ASTM E2440-18 which is an extension test over a given length at a strain rate and temperature. The main defects in superplastic forming are sheet wrinkles and thickness variations.

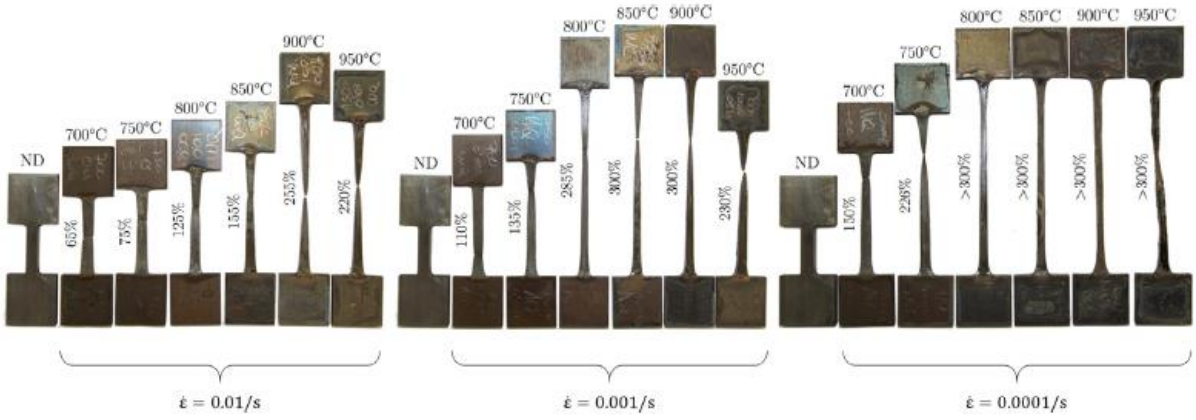


Figure 2.4 - SPF temperature and strain rate dependence [8]

Grain boundary sliding is the dominant mechanism in superplasticity. However, in grain boundary sliding there would be significant voids formed due to differences in grain shape without accommodation mechanisms. The mechanisms that accommodate the grain boundary sliding

depend on the material and the process conditions and are still being investigated [7]. The two main theories are Rachinger sliding or Lifshitz sliding. Rachinger sliding is based on dislocations in the lattice and Lifshitz sliding is based on diffusion of material. As a result of both of these mechanisms there are many microstructural changes that occur during superplasticity. The texture of the microstructure become less organized due to the scattering of grains as they switch and slide between each other as seen in Figure 2.5. The grains grow both due to the natural grain growth at 50-70% of the melting temperature, called static coarsening, as well as due to the diffusion processes inherent in the forming operation, called dynamic coarsening. Even though the material is deforming plastically, there is no work hardening and negligible residual stresses after forming. With improper forming conditions, cavities can form usually at triple points where grains contact and can end up coalescing and causing failure of the part as shown in Figure 2.6.

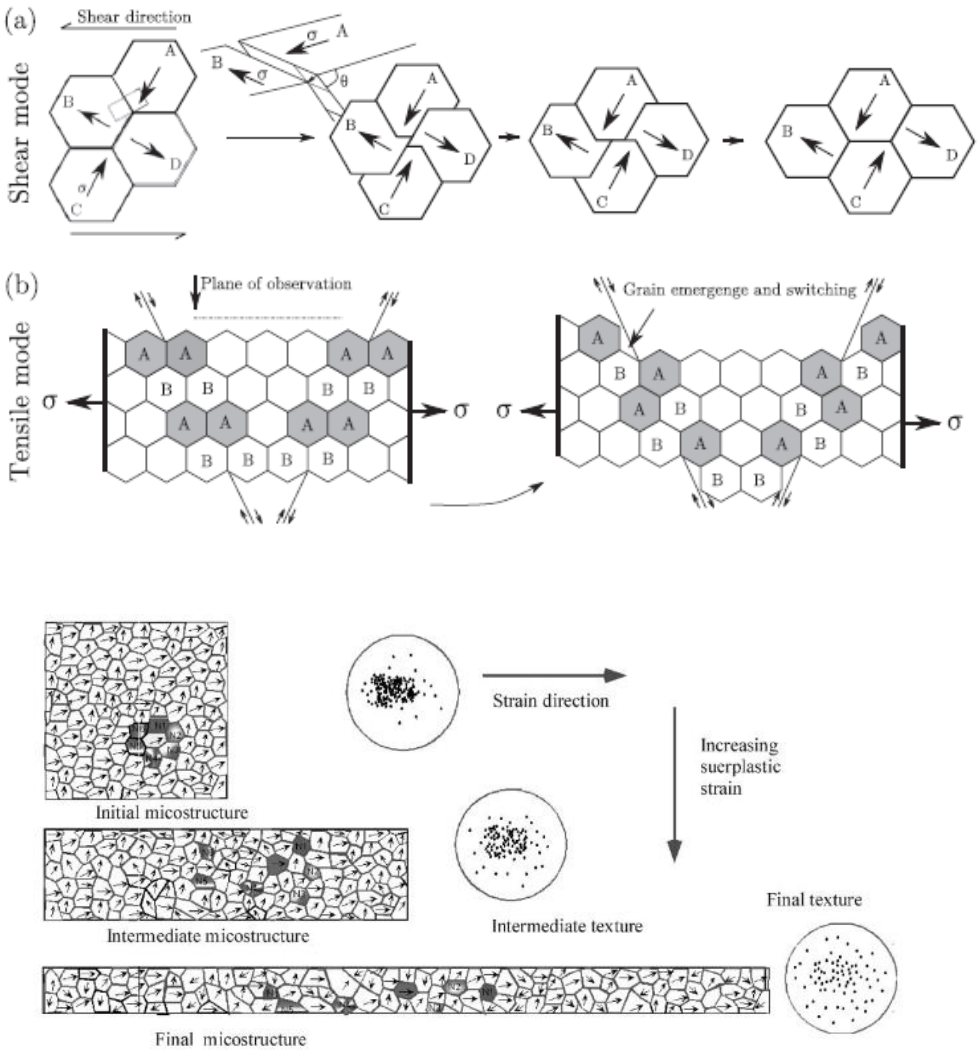


Figure 2.5 – Superplasticity mechanisms and resulting texture and microstructure. Grain switching in superplasticity (top) [7] and reduction in crystal texture during SPF (bottom) [4]

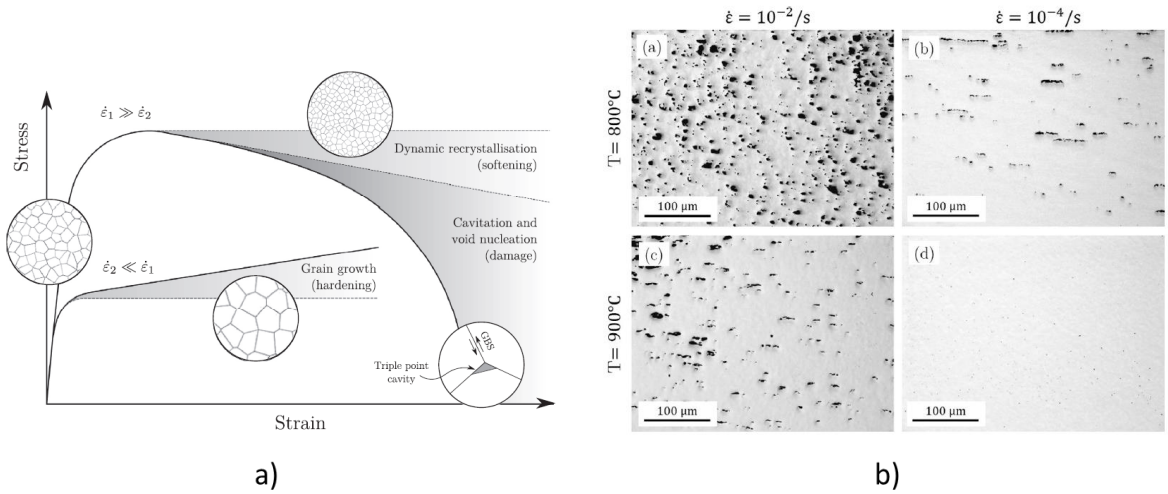


Figure 2.6 - Changes in grain size due to superplasticity (a) and void formation at different temperatures and strain rates in Ti-6Al-4V (Ti64) (b) [8]

Superplasticity also changes the topology of the free surface. Since the grains are migrating between each other there is distortion and grains can migrate from the interior to the exterior of the specimen or workpiece [4]. The exact nature of this effect has never been quantified but the changes in surface roughness are very important for diffusion bonding. As shown in Figure 2.7 the superplastic strain causes grain rotations. At the free surface this results in roughening.

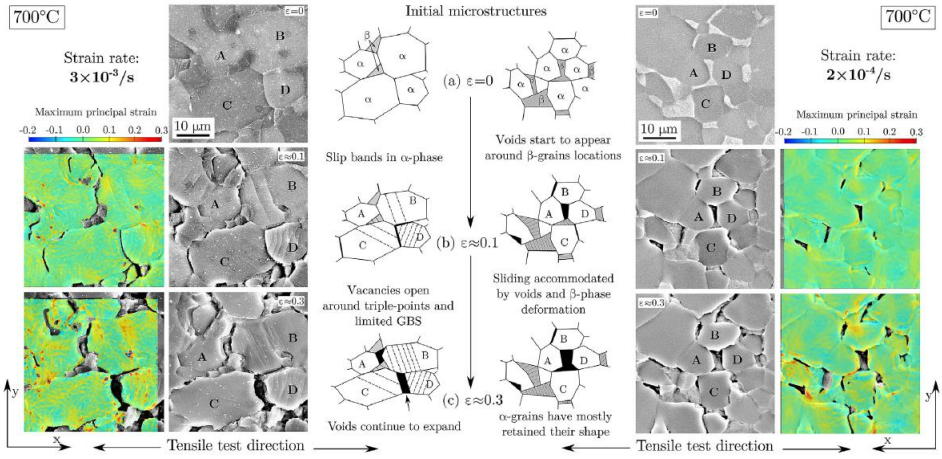


Figure 2.7 - Changes in surface from superplasticity in Ti-6Al-4V. The micrographs are at 700°C and strain rates of $3x10^{-5}$/s on the left and $2x10^{-4}$/s on the right. Each strain rate contains points for the maximum principal strains and SEM micrographs. [7]

2.3.3 Superplastic Forming Models

Modeling of superplastic forming has been done both analytically and more accurately with finite element simulations. The superplastic forming process is expensive and lends itself to using simulations to reduce the number of experiments needed to determine processing parameters. The testing to determine the material behavior of the sheet as well as the frictional behavior of the mold is also expensive. Therefore, the main focus of research in SPF modeling has been to circumvent the amount of testing required by developing constitutive models that can accurately predict forming conditions with fewer inputs. Constitutive models map the stress to strain for a variety of strain rates. Models accurately reflect the stress-strain behavior. This is shown in Figure 2.8.

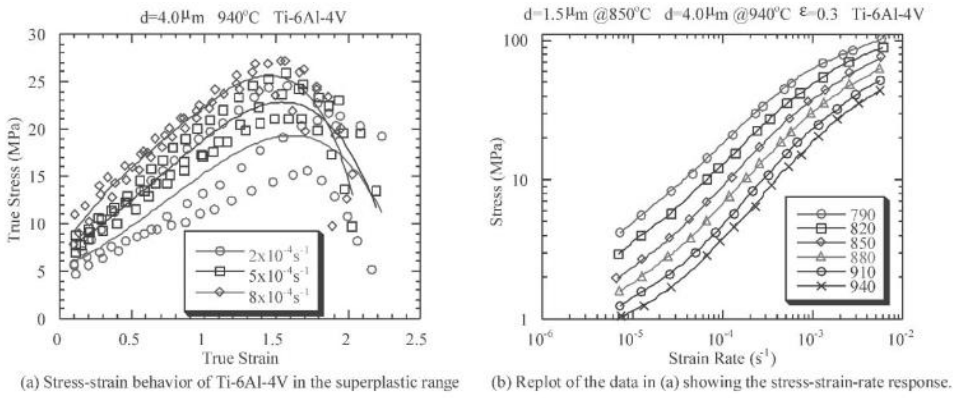


Figure 2.8 - Thermomechanical behavior of Ti-6Al-4V which can be used to build constitutive models [4]

There has been a substantial amount of research on predicting superplasticity. The equation most commonly used for titanium has been the power law equation or the viscoplastic equation. In these equations k is the Boltzmann constant, G is the shear modulus, D is the diffusion constant, σ is the stress, ε is the strain, b is the burgers vector, d is the grain size, and A is a material constant. Other

equations are there that estimate the grain size increases, the void formation, and other characteristic in forming that are commonly seen [7, 8].

$$\dot{\varepsilon} = \frac{ADGb}{kT}\left(\frac{b}{d}\right)^{p}\left(\frac{\sigma}{G}\right)^{n} \quad (power\ law\ creep) \tag{2-1}$$

$$\sigma = A\varepsilon^{n}\dot{\varepsilon}^{m}d^{p} \quad (viscoplastic) \tag{2-2}$$

There have also been atomistic methods used in superplasticity [4]. These operate only over very short time periods and have been used to build models of grain boundary sliding but require a significant amount of extrapolation and so have varying success.

2.3.4 Superplastic Forming and Diffusion Bonding in Practice

The practical applications of superplastic forming and diffusion bonding with respect to titanium extend back to the Rockwell Science Center in *1971*. Testing was performed there to confirm that superplastic titanium sheets could be blown and form internal diffusion bonded supports that create elaborate sandwich structures. Examples of these sandwich structures can be seen in Figure 2.9 which were developed by Rockwell for the B1-B bomber program.

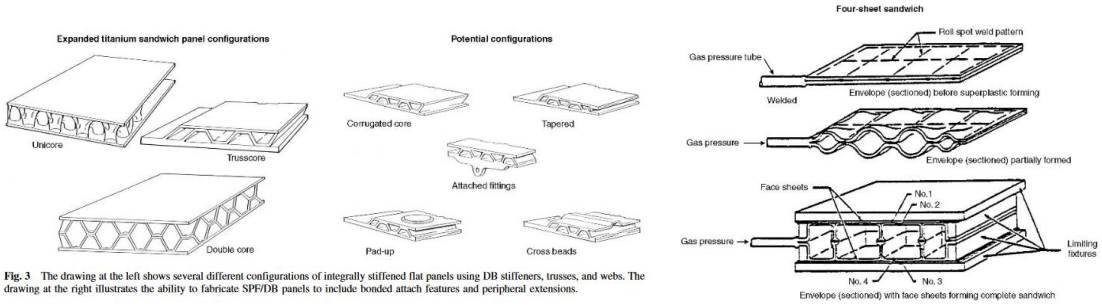


Fig. 3 The drawing at the left shows several different configurations of integrally stiffened flat panels using DB stiffeners, trusses, and webs. The drawing at the right illustrates the ability to fabricate SPF/DB panels to include bonded attach features and peripheral extensions.

Figure 2.9 - Rockwell multi-sheet forming and McDonnell four-sheet truss [3]

The multi-sheet structure used in this research is the McDonnell four-sheet truss pictured in Figure 2.10. This was first developed for the F-15E by the US Air Force which used the technology extensively. These four sheets ended up forming two outer sheets and a welded I-beam core that supports the outer sheets. This was also implemented on the B1-B bomber but has not been implemented in any commercial application since [3].

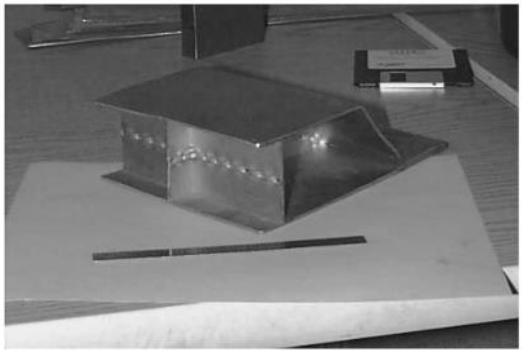


Figure 2.10 - Photo of the four-sheet SPF/DB product [3]

In contrast with the multi-sheet structures, most commercial SPF/DB has been done with two sheets patterned with Yttria powder. Yttria powder is silk screened onto flat sheets of titanium then the sheets are sealed and a vacuum is pulled in between the sheets. This powder prevents bonding and so, when these are bonded at temperature and pressure, they result in a pattern of unbonded area that can be superplastically formed. These bonded multiple sheets are then put into a superplastic press. Argon provides internal and external pressure for forming and the unbonded area forms the stiffeners in the part [9]. Examples of parts formed with two sheet silk screened bonding are shown in Figure 2.11.

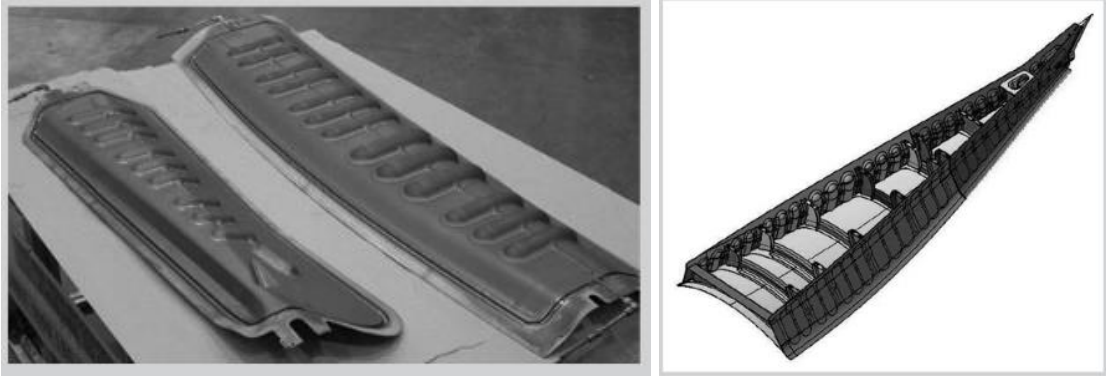


Figure 2.11 - Two sheet SPF/DB at Boeing [9]

2.4 Diffusion Bonding

2.4.1 Introduction

Published research has been available on diffusion bonding since the 1950s [10] and it has been investigated both experimentally and theoretically. Research has primarily been done on determining how to control the voids formed by the microscopic differences between the surfaces when they contact. This background focuses on the theoretical modeling of the diffusion bonding along with the experimental methods used to investigate its structural performance.

2.4.2 Mechanisms and Process Parameters of Diffusion Bonding

Diffusion bonding occurs due to the minimization of the chemical potential energy caused by the void under pressure at a high enough temperature to promote diffusion. The main parameters in determining the diffusion bonding performance are the materials involved and their surface topologies along with the applied temperature, applied pressure, and time period. Each material will bond differently. This is based on material surface topology as well as the grain structure and composition. In dissimilar alloys there can be wide discrepancies in the bonding performance

simply due to the differences in alloys. The decrease in grain size, decrease in surface roughness, the increase in temperature, pressure, and time are all well documented to increase the closure of voids depicted in Figure 2.12 and Figure 2.13.

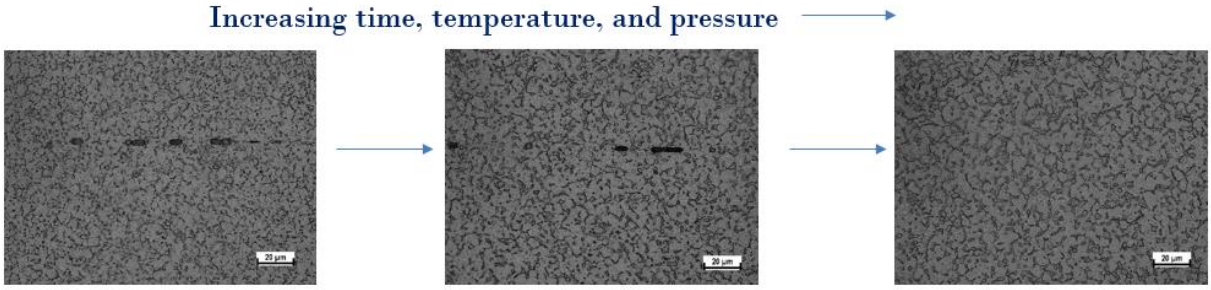


Figure 2.12 - Trends in diffusion bonding processing conditions with respect to the voids leftover after bonding. Increases in the time that the part is allowed to bond, elevations in the temperature used for that time, or increases in the pressure used for that time all result in a higher quality part with fewer voids.

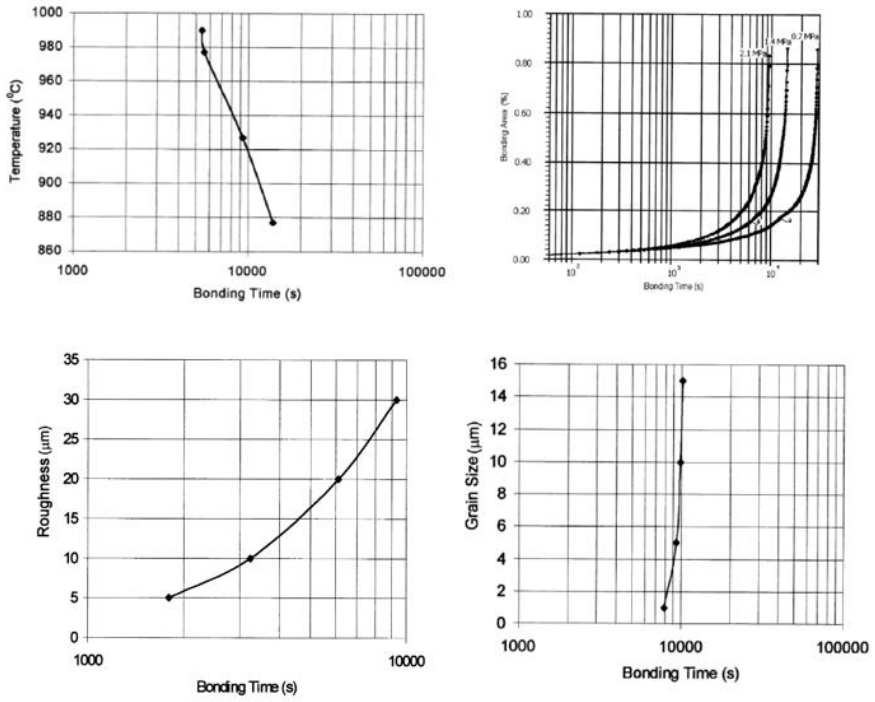


Figure 2.13 – Simulated effects of temperature (top left), pressure (top right), surface roughness (bottom left), and grain size (bottom right) on diffusion bonding of Ti-6Al-4V [11]

2.4.3 Diffusion Bonding Models

Diffusion bonding models were initially published in 1973 with C.H. Hamilton and have been routinely published to the present day with increasingly modern approaches. These models have included a variety of bonding mechanisms but are predominantly analytical and based on a combination of diffusion processes and superplastic or power law creep deformation.

Hamilton [12]

The first attempt at modeling diffusion bonding was done by C.H. Hamilton's in his seminal paper in 1973. The diffusion bonding process in Hamilton's model, shown in Figure 2.14, assumed that the surfaces in contact with each other were triangular, saw tooth shaped, and subject to plane strain. The voids formed were also triangular and were modeled as reducing uniformly keeping the same triangular shape. Reduction in void size was based on superplastic collapse. Superplastic material has very little ability to withstand a stress without constant deformation. Therefore the total strain to collapse was related to the bonding time via the strain rate. The model correctly predicted that the void closure increases with both temperature and pressure and that surface roughness plays an important role.

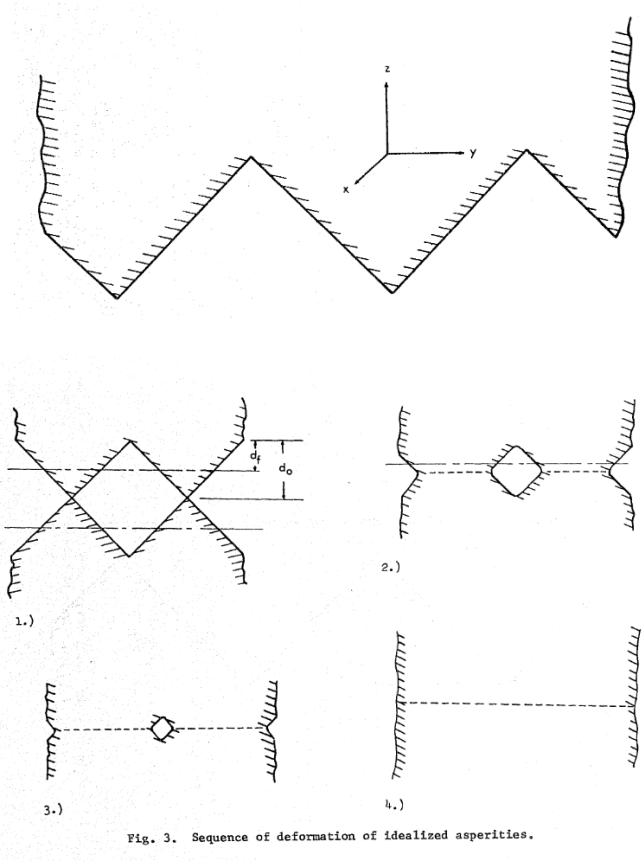


Fig. 3. Sequence of deformation of idealized asperities.

Figure 2.14 – Hamilton's triangular ridge geometry and closure [9]

The strain rate is approximated as the total plastic strain averaged over the bonding time. The stresses used were simply ratios of the normal stress, pressure times the total area over the asperity area. These are then transferred in to plane strain effective stresses and strain rates using the distortion energy (von Mises) equivalence equations. The von Mises distortion energy equivalent stress allows Hamilton to equate the bonding stresses to the material parameters found with tensile experiments. In the equations below, ϵ_z is the total plastic strain to close, d_0 is the initial height, d_f is the final height, t_b is the bonding time, P_b is the bonding pressure, σ is the stress, and A is the bonding area.

$$\epsilon_z = \ln\frac{d_f}{d_0} = \ln\frac{1}{2} \qquad d_f = \frac{d_0}{2} \tag{2.2-3}$$

$$\dot{\epsilon_z} = \frac{\epsilon_z}{t_b} \qquad \sigma = \frac{P_b A}{\frac{A}{2}} = \frac{P_b}{2} \tag{2.2-4}$$

$$\dot{\bar{\epsilon}} = \frac{2}{\sqrt{3}}|\dot{\epsilon}_z| \qquad \bar{\sigma} = \frac{\sqrt{3}}{2}|\sigma_z| \tag{2.2-5}$$

Hamilton presented his results for various bonding specimens as the difference between the predicted bonding pressures and the experimentally derived bonding pressures for a given temperature and time combination. The discrepancy is large with some bonding pressures. For example, 1440F bonded for 60 minutes experimentally provides a bonding pressure 0.016 times the theoretically predicted bonding pressure as shown in Table 2-1. This table shows most importantly the bonding temperature, a set bonding time, the experimental pressure, the predicted bonding pressure for two models, and ratios of the experimental pressure over the predicted pressure. The P_a/P_b shows how effective this model fit the experimental data. This was an important first attempt at diffusion bonding modeling and is still used as a guideline for production.

Table 2-1 - Hamilton's experimental results compared with calculations

Comparison of Analytical and Reported Experimental Data
For Diffusion Bonded Ti-6Al-4V

Bond Temp. (°F)	Bond Time (t_b)	Bond Press., P_a (psi)	Pred. Press., P_b (1) (psi)	Pred. Press., P'_b (2) (psi)	P_a/P_b	P_a/P'_b	Bond Strength σ_b (ksi)	σ_b/σ_{pm}
1350F	45 Min	5,000	9,300	7,500	0.54	0.666	72	0.47
1440	60 Min	100	6,400	5,000	0.016	0.02	28(s)	0.304
1540	60 Min	650	2,900	2,250	0.23	0.28	81(s)	0.89
1400	400 Min	300	3.800	3,000	0.078	0.10	79.4(s)	0.88
1650	96 Min	50	695	600	0.072	0.08	76.3(s)	0.78
1650	230 Min	50	400	350	0.12	0.14	81.2(s)	0.83
1675	960 Min	2,000	145	100	14.0	20.0	143	1.02
1600	960 Min	2,000	290	200	6.9	10.0	132	1.02
1700	60 Min	100	640	500	0.16	0.19	133	0.92
1700	240 Min	100	230	150	0.42	0.67	137	0.948
1700	180 Min	2,000	290	200	6.9	10.0	141	1.007
1700	960 Min	1,000	120	80	8.3	18.2	140	1.0
1750	15 Min	25	1,450	1,000	0.017	0.025	59.6(s)	0.67
1750	960 Min	2,000	60	40	33	50.0	137	0.978

(s) - Block Shear Test

(1) - Plane Strain Condition
(2) - Plane Stress Condition

Garmong, Paton, and Argon [13]

Garmong et al. developed a model based on Hamilton's model but with different bonding mechanisms in 1975. They considered the surface roughness to be bimodal based on observations of profilometer data and partially bonded samples. The bimodal surface roughness consists of two a summation of two sinusoidal curves, one with a large wavelength and amplitude and another with a much smaller wavelength and smaller amplitude. Along with this, Garmong, as shown in Figure 2.15, assumed a multi-stage bonding process where initial bonding occurred by plastic collapse of the large wavelength asperities and then the small voids are reduced by creep flow and diffusion.

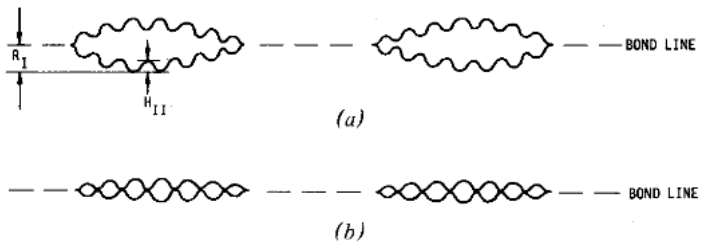


Fig. 5—Schematic development of the bondline shown in Fig. 3. The long-wavelength asperities are flattening during stage I in part (*a*). In part (*b*) the short-wavelength asperities remaining between the bonded regions have just achieved contact, and stage II closure begins.

Figure 2.15 - Garmong's multistage closure mechanisms

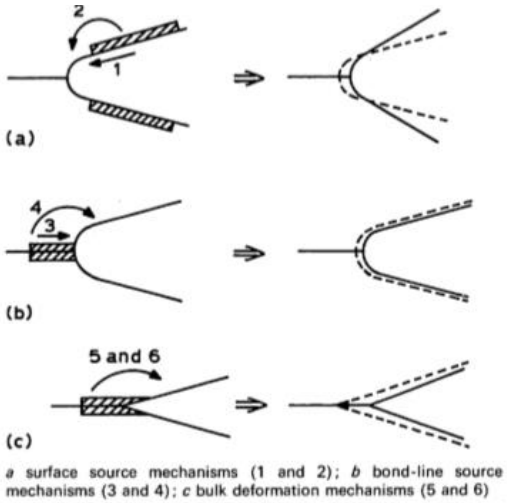
The geometry of their model was similar to Hamilton's. It used long triangular troughs in plane strain that contacted at the tips. However, they developed the model to be a series of slices that collapsed. The final voids were modeled as thick walled spheres that were subjected to creep and diffusion.

Derby and Wallach [14]

In 1982, Derby and Wallach published a diffusion bonding model based on pressure sintering equations. They implemented six different bonding mechanisms and two different geometry models both of which are 2D. Figure 2.16 shows the schematic view of void closure by the six different mass transfer mechanisms. They used triangular ridges to model the initial geometry then assumed that these ridges closed in a predictable way. The geometry is significantly different at later stages in bonding and therefore they used a secondary geometry that is simply circular with uniform curvature. They listed the primary first stage bonding mechanisms as the six mechanisms below which correspond to the numbers in Figure 2.16.

1. surface diffusion from surface sources to the neck

2. volume diffusion from surface sources to the neck
3. diffusion along the bond interface from interfacial sources to the neck
4. volume diffusion from interfacial sources to the neck
5. power law creep deformation of the ridge
6. plastic yielding of the ridge



a surface source mechanisms (1 and 2); *b* bond-line source mechanisms (3 and 4); *c* bulk deformation mechanisms (5 and 6)

1 Schematic view of growing bond showing routes of six mechanisms of mass transfer

Figure 2.16 - Diffusion mechanisms

Mechanisms 1 and 2 are driven by the curvature of the void and cease to play a role in bonding after the curvature becomes uniform. Mechanisms 3 and 4 are driven by the chemical potential gradient along the void which occurs due to the stress gradient and curvature. Mechanisms 5 and 6 are primarily due to the applied pressure as shown schematically in Figure 2.17. As the bonding progresses the voids are assumed to become spherical and the closure mechanisms are adjusted accordingly part of the way through the model. The first two closure mechanisms no longer apply since there is constant curvature. The second two have a reduced effect due to the neck and void surface being the same. Finally there is no longer any plastic yielding since the stresses are all below the yield stress.

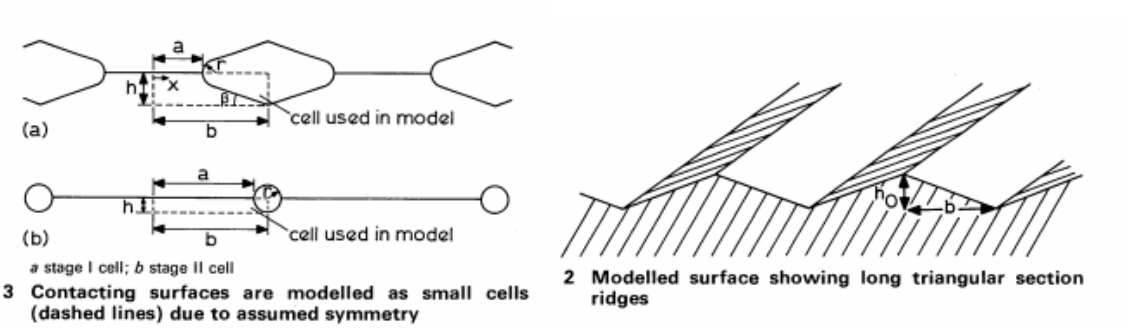


Figure 2.17 - Closure geometry

Diffusion surface source equations are based on pressure sintering models and are due primarily to the chemical potential from curvature according to the Gibbs-Thompson relationship. Therefore the diffusion process for the first and second mechanisms are purely dependent on the material and the radius of curvature. This gives the equation below which is a series of constants to model diffusive mobility and the surface energy from the curvature.

$$\dot{V}_1 = \frac{2\Omega\delta_s D_s}{rkT}\frac{\gamma}{r}\left(1 - \frac{r}{b-a}\right) \tag{2-6}$$

The equation above contains the diffusion rate ($\dot{V}_1$), atomic constant (Ω), diffusion thickness (δ_s), diffusion constant (D_s), stephan boltzmann constant (k), temperature (T), surface energy (γ), void edge radius (r), half length of the cell (b), and the half length minus the void radius (a). The interfacial diffusion mechanisms are also based on pressure sintering equations and are very similar to the surface source equations with the exception of different diffusion constants.

$$\dot{V}_2 = \frac{2\Omega D_V}{kT}\frac{\gamma}{r}\left(1 - \frac{r}{b-a}\right) \tag{2-7}$$

Where $\dot{V}_2$ and D_V are the diffusion rate and the volume diffusion. From interfacial sources to the neck the equations are also similar with the equation being derived from pressure sintering references but with different constants and geometric parameters.

$$\dot{V}_3 = \frac{3\Omega\delta_B D_B}{2kTa}\left(P\frac{b}{a} - \frac{\gamma}{a} - \frac{\gamma}{r}\right) \tag{2-8}$$

$$\dot{V}_4 = \frac{3\Omega r D_V}{kTa}\left(P\frac{b}{a} - \frac{\gamma}{a} - \frac{\gamma}{r}\right) \tag{2-9}$$

Where $\dot{V}_3, \dot{V}_4, \delta_B, D_B$ and P are the diffusion rates, boundary diffusion thickness, boundary diffusion constant, and the bonding pressure. For the power law creep and plastic deformation the authors used an assumed stress field of an area of hydrostatic compression at the center of a thick walled cylinder as shown in Figure 2.18. Derby used the diffusion bonding pressure as the pressure applied at the outer cylinder and a hydrostatic inner region. The plastic deformation is due to local stresses that are much higher than the ones accommodated by the creep equations. These are primarily from contact stresses during initial loading and relax and disappear after the contact area has spread out. When compared with the time scale the other mechanisms operate on, the time taken for the initial plastic yielding is relatively instantaneous. Derby used a slip line model for plastic deformation of a wedge against a plate to find the extent of void height reduction due to plastic collapse. In the equation below, aside from previously defined constants, *S* is the sign of the bonding pressure, A_c is the power law creep pre-exponential constant, n_c is the power law creep exponent, μ is the chemical potential, and *P* is the applied bonding pressure.

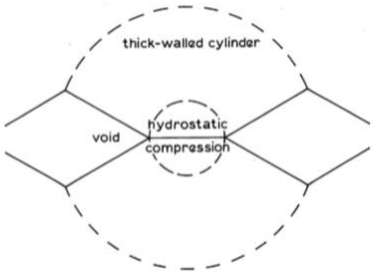


Figure 2.18 – Loading geometry of Derby Model for creep deformation

$$\dot{V}_5 = \frac{-\sqrt{3}\pi r^2 S A_c}{2\left[1-\left(\frac{a}{b}\right)^{\frac{2}{n_c}}\right]^{n_c}}\left(\frac{2\left|\mathrm{P}\frac{\mathrm{b}}{\mathrm{a}}-\frac{\gamma}{a}\right|}{\mu n_c}\right)^{n_c} \tag{2-10}$$

Derby and Wallach compared their equations with experiments for diffusion bonding of pure copper as shown in Figure 2.19. Derby and Wallach found reasonable agreement between the data and the model. The model underestimated the bonding quality for given processing conditions but they did cite that there was minimal reference to the surface conditions for the experimental data. The general trend appears to be appropriate however and the dominant mechanism is listed as power law creep. They give no reason for discrepancies between experimental and theoretical results.

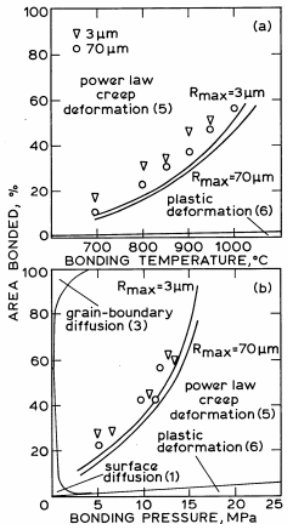


Figure 2.19 - Derby and Wallach experimental theoretical comparison

Hill and Wallach [15]

Hill worked with Wallach to modify Derby and Wallach's model in several ways and published in 1989. Specifically they used a slip-line field analysis to better determine the initial plastic yielding and reduction in void volume due to yielding. They also used more precise geometry in the form of infinitely long ridges in plane strain with elliptical geometry. They improved the creep void closure contribution to be more realistic. By using the elliptical geometry they eliminated the need for a final bonding stage that was present in Derby and Wallach's model since the elliptical geometry can become the cylindrical final geometry. This is less restrictive of the void shape, it eliminates the forced change in shape from triangular to cylindrical, and allows any bonding mechanism to become dominant if they turn out to be. Figure 2.20 shows the Hill and Wallach elliptical geometry.

Hill changed how the initial plastic deformation was modeled from the hydrostatic compression/thick walled cylinder used by Derby to a more robust slip-line plasticity model. The

stress distribution in the equation below is the stress of the contact area between voids. It is maximized at the midpoint between the voids and is used to equate the initial contact area with the bonding pressure. In the equation below, σ is the stress distribution between the edge of the void and the midpoint of the neck, K is the factor in the von Mises yield criterion, y is the distance across the neck between voids, and r_c is the radius of curvature of the major semi-axis.

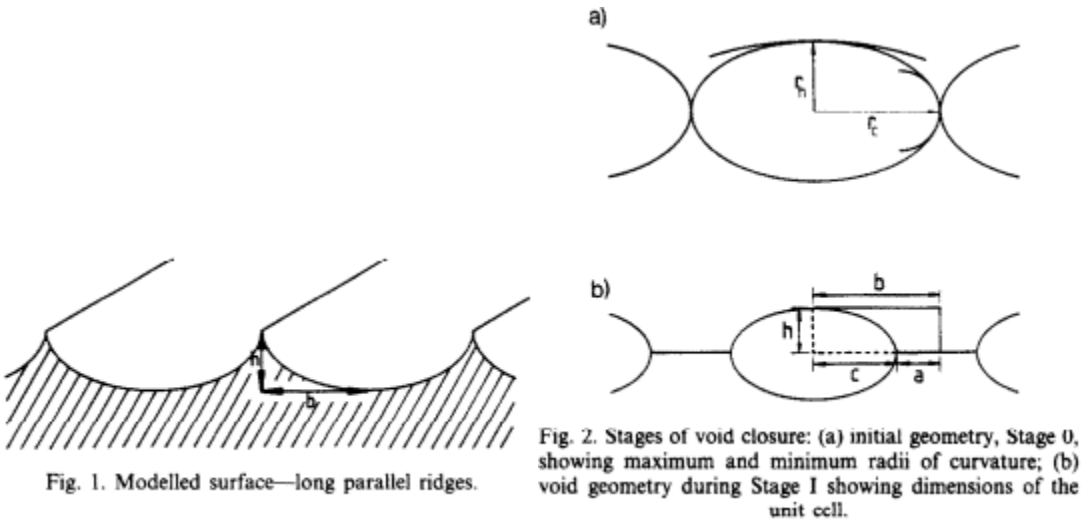


Fig. 1. Modelled surface—long parallel ridges.

Fig. 2. Stages of void closure: (a) initial geometry, Stage 0, showing maximum and minimum radii of curvature; (b) void geometry during Stage I showing dimensions of the unit cell.

Figure 2.20 - Hill elliptical geometry

$$\sigma = 2K\left[1 + \ln\left(\frac{y}{r_c} + 1\right)\right] \tag{2-11}$$

The same surface source and volume source equations were used for diffusion (shown in equations 2.6 through 2.9). However, they were modified for the elliptical geometry. Their power law creep equations were also modified in the way they were implemented by fitting the stress to the elliptical geometry and by only using the final stage creep mechanics. When compared with experimental data for copper, the Hill model over-approximates the bonding percentage. Hill's material constants are listed in Table 2-2 and experimental and theoretical results are shown in Figure 2.21.

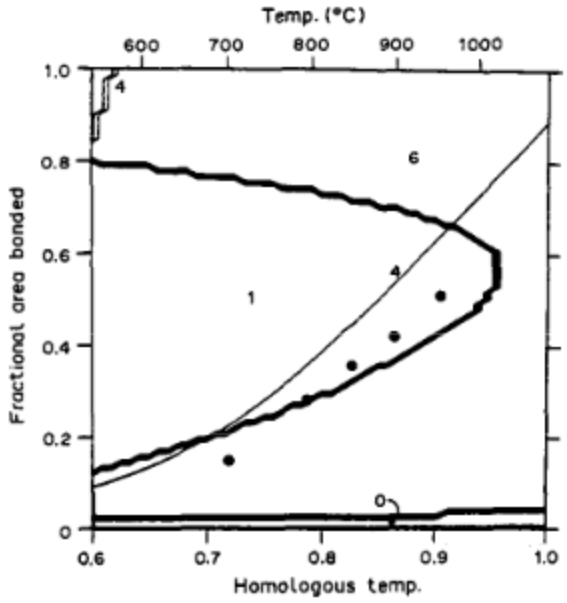


Figure 2.21 - Hill experimental theoretical comparison. The figure shows both fraction of bonding (thin line) and the experimental data (dots). It also shows the dominant bonding mechanism operating at a given temperature and final bonded area fraction. The numbers refer to the mechanisms listed in Figure 2.16.

Table 2-2 - Hill material values

Table 1. Data for model

Parameter	Symbol	Units	Value		
			α-iron	γ-iron	copper
Atomic volume	Ω	nm^3	1.18×10^{-2}	1.21×10^{-2}	1.18×10^{-2}
Burgers vector	b_v	nm	0.248	0.258	0.256
Density	ρ	$kg\,m^{-3}$	7.87×10^{3}	7.69×10^{3}	8.96×10^{3}
Melting point	T_m	K	1810	1810	1356
Shear modulus at 300 K	μ	GPa	64	81	42.1
Temperature coefficient of shear modulus	—	—	4.48×10^{-4}	5.05×10^{-4}	3.97×10^{-4}
Yield stress, normalised in terms of shear modulus	$\frac{\sigma_y}{\mu}$	—	3×10^{-3}	3×10^{-3}	2×10^{-3}
Surface energy	γ_S	$J\,m^{-2}$	1.95	2.17	1.75
Interface energy	γ_B	$J\,m^{-2}$	0.78	0.756	0.625
Diffusion					
Volume pre-exponential	D_{oV}	$m^2\,s^{-1}$	2×10^{-4}	1.8×10^{-5}	2×10^{-5}
Volume activation energy	Q_V	$kJ\,mol^{-1}$	251	270	197
Boundary pre-exponential	D_{oB}	$m^2\,s^{-1}$	1.1×10^{-2}	7.5×10^{-4}	5×10^{-5}
Boundary activation energy	Q_B	$kJ\,mol^{-1}$	174	159	104
Surface pre-exponential	D_{oS}	$m^2\,s^{-1}$	10	0.4	7×10^{-6}
Surface activation energy	Q_S	$kJ\,mol^{-1}$	241	222	79.1
Boundary layer thickness	δ_B	nm	0.1	0.1	0.1
Surface layer thickness	δ_S	nm	0.1	0.1	0.1
Power law creep					
Activation energy	Q_{crp}		as volume diffusion activation energy		
Constant	—	—	7×10^{13}	4.3×10^{5}	7.4×10^{5}
Exponent	m	—	6.9	4.5	4.8

Pilling [16–19]

John Pilling presented two different models as well as a several papers and a textbook on the subject of diffusion bonding [20]. His first model was published in 1984 and has the simplest geometry. His second model was published in 1988 and had the same basic equations but different void geometry. He collaborated on a paper with Salehi in 1992 that discussed the importance of void geometry. His final diffusion bonding paper was on super-alpha 2, a titanium aluminide.

Pilling's models are primarily based on the work of Chen and Argon [21] and operate under the assumption that creep deformation is the primary closure mechanism. Chen and Argon presented an explanation of void formation during superplastic forming. This included equations for superplastic opening of the voids and diffusion from the voids to the surrounding material. Pilling's models are the reverse of the Chen and Argon theory and assume the same mechanism.

Pilling's first model uses infinitely long cylinders as the void geometry and models the void closure mechanisms as superplastic collapse and volume diffusion. It is assumed that when infinitely long ridges come in contact with each other there is an initial plastic collapse before the void closure mechanisms occur. In Pilling's model, everything is related to the fractional area of bonding. Therefore the initial plastic collapse, and the initial bonded area fraction A_f, is the ratio of the bonding pressure (σ_b) to the yield strength (σ_y) [16].

$$V = \pi r^2 \ \ with\ unit\ depth \tag{2-12}$$

$$A_f = \frac{\sigma_b}{\sigma_y} \tag{2-13}$$

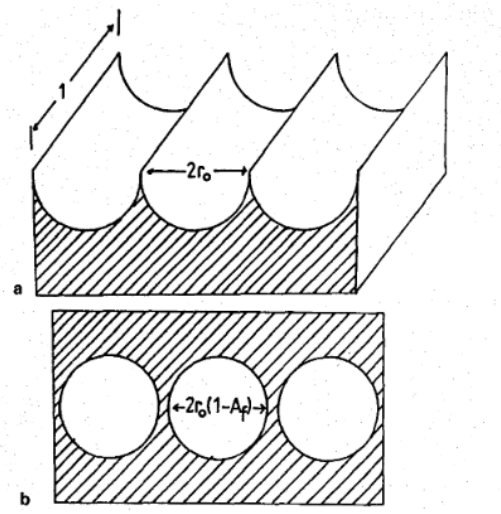


Figure 2.22 - Pilling 1984 geometry. These are prismatic cylinders that close uniformly.

The creep closure was based on Chen and Argon's growth rate of voids. The final equation is modified from spherical geometry in Chen and Argon to cylindrical. Pilling spent time discussing the inadequacies of both Derby and Wallach's model and another creep void growth model by Hancock [22] stating that neither contained the correct diffusion processes.

The final equation used by Pilling is given below. For the material properties Pilling used the rule of mixtures from alpha and beta titanium. In Equations 2.14 and 2.15 the variables $\frac{dV}{dt}, D_B, \delta, \Omega, \sigma, A_f, k, T\ r,$ and $\dot{\varepsilon}$ are the volume transfer rate, boundary diffusion constant, diffusion thickness, atomic volume, bonding pressure, initial bonding fraction, stephan boltzmann constant, temperature, void radius, and creep strain rate.

$$\frac{dV}{dt} = \frac{16 D_B \delta \Omega \sigma_b}{B A_f k T} \left(\frac{1}{4 \ln\left(\frac{B+r}{r}\right) - \left[3 - \left(\frac{r}{r} + B\right)^2\right]\left[1 - \left(\frac{r}{r} + B\right)^2\right]} \right) \tag{2-14}$$

$$B = \left| \frac{D_B \delta \Omega |\sigma_b|}{A_f k T \dot{\varepsilon}} \right|^{\frac{1}{3}} \tag{2-15}$$

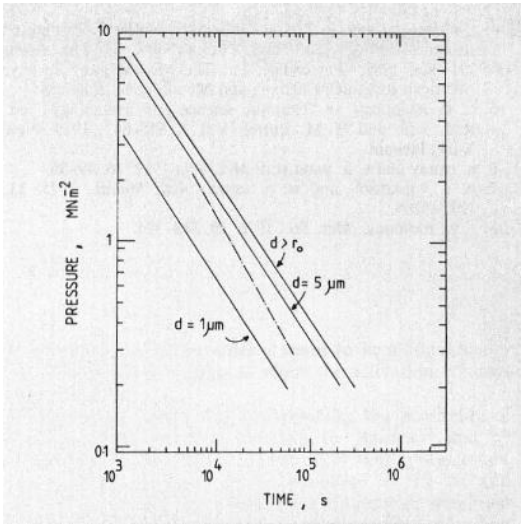


Figure 2.23 - Pilling model experiment and theory comparison

Pilling lists four different reasons for discrepancies between his experiments and theory. First, the average surface roughness used may not represent the voids that are actually produced. Second, the surface roughness parameter does not accurately represent the surface. Third, mass transport is only coming from the bondline and could be modified to come from the grain boundaries as well. Finally, the experimental setup may be incorrect and not provide uniaxial compression. Pilling worked on remedying some of these problems in his next model.

Pilling's second model was published in 1988 and was based on the same three mechanisms: i) plastic collapse of contact points, ii) superplastic flow into the voids; and iii) diffusion into the voids. The void geometry was modified from infinitely long cylinders to upright cylinders that are perpendicular to the plane of bonding. For this model he does not use a re-derived equation

from Chen and Argon but instead derives his own based on the pressure distribution. The pressure distribution is assumed to be isostatic and completely uniform around the voids [17].

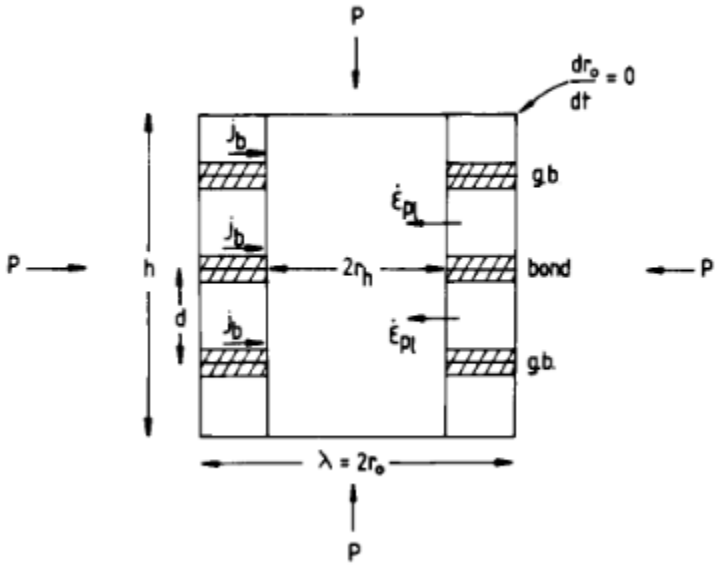


Figure 2.24 – Second Pilling model geometry. It is a cylinder oriented perpendicular to the bonding plane.

The stress equations used for superplastic collapse are based on radial force equilibrium equations for the cylinder sides and have a pressure distribution of the area for the top and bottom with respect to the radial dimensions times the bonding pressure as shown in Figure 2.24.

$$\frac{d\sigma_r}{dr} + \frac{\sigma_r - \sigma_0}{r} = 0 \tag{2-16}$$

$$\sigma_r = \frac{r_h^2 r_o^2 (P - P_i)}{r_o^2 - r_h^2}\frac{1}{r^2} + \frac{P_i r_h^2 - P r_o^2 r_o^2}{r_o^2 - r_h^2} \tag{2-17}$$

$$\sigma_\theta = -\frac{r_h^2 r_o^2 (P - P_i)}{r_o^2 - r_h^2}\frac{1}{r^2} + \frac{P_i r_h^2 - P r_o^2}{r_o^2 - r_h^2} \tag{2-18}$$

$$\sigma_z = -\frac{P r_o^2}{r_0^2 - r_h^2} \tag{2-19}$$

These equations are related to the plastic strain by the Von Mises effective stress. The effective plastic strain rate is related to the stress by a creep equation. The creep equation includes both superplasticity and power law creep where A_{sp} and A_{pl}, n_{sp} and n_{pl}, and Q_{sp} and Q_{pl} are the rate constants, stress dependencies, and temperature dependencies for superplastic flow and power law creep respectively.

$$\dot{\varepsilon} = \frac{A_{sp} e^{\left(-\frac{Q_{sp}}{RT}\right)} \sigma^{n_{sp}}}{T} + \frac{A_{pl} e^{\left(-\frac{Q_{pl}}{RT}\right)} \sigma^{n_{pl}}}{T} \tag{2-20}$$

$$\frac{\dot{\varepsilon}}{\sigma} = \frac{\dot{\varepsilon}_z}{\sigma_z - \frac{1}{2}(\sigma_\theta + \sigma_r)} = \frac{\dot{\varepsilon}_r}{\sigma_r - \frac{1}{2}(\sigma_\theta + \sigma_z)} = \frac{\dot{\varepsilon}_\theta}{\sigma_\theta - \frac{1}{2}(\sigma_z + \sigma_r)} \tag{2-21}$$

Pilling assumes that any axial collapse (in the z direction) increases the void radius but keeps the same r_o by creating a bore of material in the center of the cylinder. Pilling also assumes that the height changes the bore as well by moving material around. Therefore all the superplastic and power law creep flow can be related to an area fraction. By reformulating the equations (and eliminating second order terms) the above equations become the ones below along with a diffusion equation.

$$\left.\frac{df_h}{dt}\right|_{pl} = -2\dot{\varepsilon}_r(1 - f_h) \tag{2-22}$$

$$\frac{df_h}{dt} = -\frac{1}{\pi r_0^2 h}\frac{dV}{dt} \tag{2-23}$$

$$\left.\frac{df_h}{dt}\right|_{d} = -N_i\left(\frac{1}{\pi r_0^2 h_0 f_h}\frac{dV}{dt}\right) \tag{2-24}$$

$$\frac{dV}{dt} = \frac{-2\pi D_{gb}\delta\Omega\sigma_z}{kT}\frac{1 - f_h}{\ln\left(\frac{1}{f_h}\right) - \frac{1 - f_h}{2}} \tag{2-25}$$

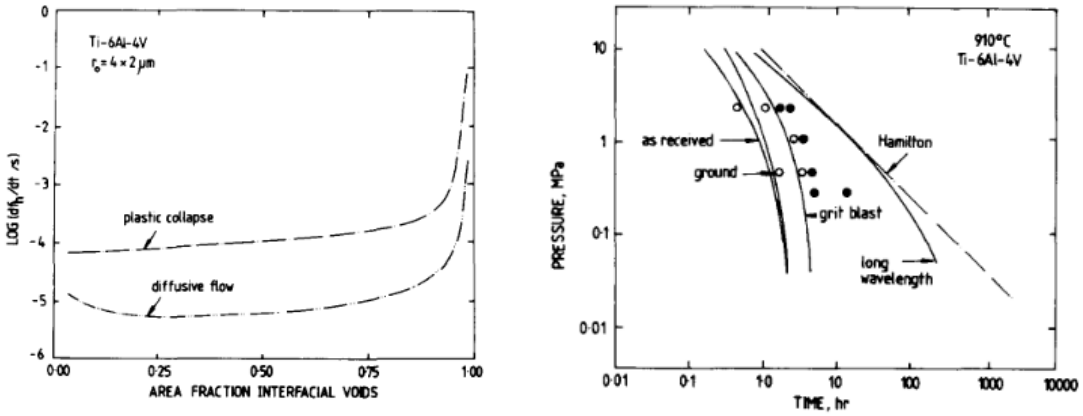


Figure 2.25 - Kinetic of void closure (left) and experimental numerical comparison (right). The kinetics are at P=1 MPa and T=910°C. The bonding time comparison is for surfaces ground with 120 grit where circles are partially bonded, filled circles are fully bonded, and the theoretical trends are labeled.

Pilling states that this isostatic model underestimates the bonding time that he determined experimentally in his first paper. The main reasons stated for this discrepancy is that considering only short wavelength asperities does not adequately represent the surface geometry. This model was meant to determine closure of very small voids and it is probable that the voids formed are much larger and from the longer wavelength asperities. Pilling states that as the voids get larger the bonding trends toward the plane strain Hamilton model.

In 1992 Salehi and Pilling published experimental results on isostatic diffusion bonding of Ti-6Al-4V with an isostatic diffusion bonding model from his previous work for both cylindrical and ellipsoidal geometry [15]. This model was important because it differentiated the effects that the geometry and surface roughness has on diffusion bonding prediction. They used metallographic analysis as well as lap shear testing to determine bond quality. The model was based on axial collapse of the cylinder due to time-dependent plastic flow and radial diffusion. The same equations used in Pilling's second model were used in this model with the exception of changing

the void geometry from a cylinder to an oblate spheroid. They did not publish any of the equations but did publish the results and noted that oblate spheroids reduced the void volume by $\pi/4$ and reduced the average surface roughness by a factor of two. They compared the bonding pressure with experiments for polished and unpolished Ti-6Al-4V surfaces as shown in Figure 2.26.

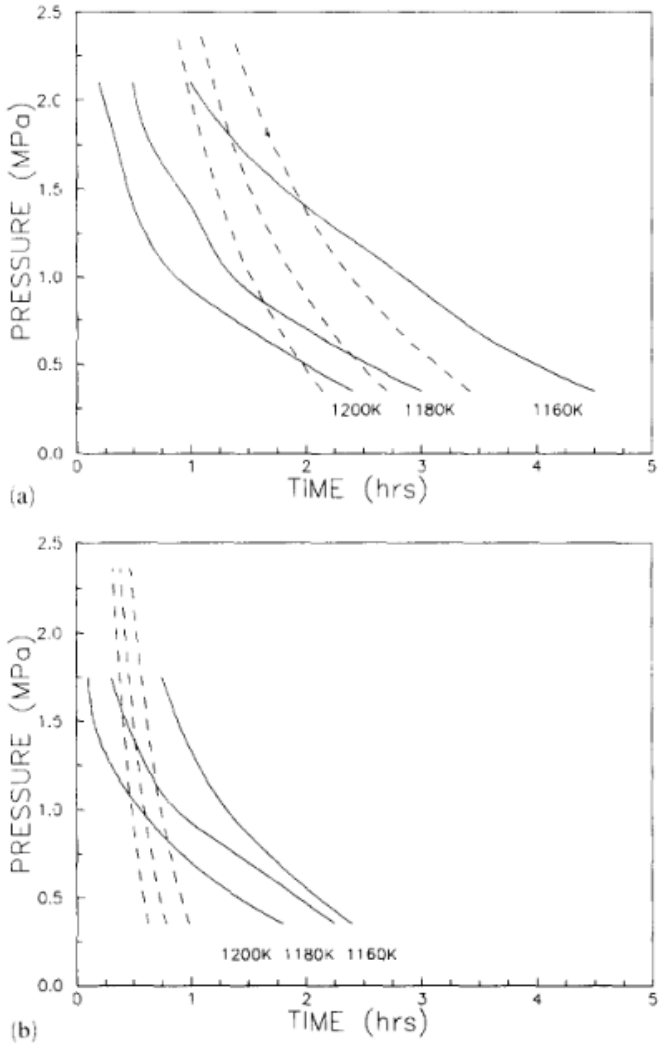


Figure 2.26 – Experimental and theoretical bonding time comparison for as received and pickled (top) and p1200 finished (bottom). Solid lines are experimental while dashed are predicted.

The final paper Pilling wrote on the subject was for the material super-alpha 2. This model used the same model geometry as the Pilling 1988 model but instead utilizing Fick's second law of diffusion.

$$\left.\frac{df}{dt}\right|_{vol} = -\frac{4\Omega D_V}{kT}\left[\frac{f}{\ln\left(\frac{1}{f}\right) - \frac{1-f}{2}}\right]\frac{p}{r_0^2} \tag{2-26}$$

$$\left.\frac{df}{dt}\right|_{gb} = -\frac{2\Omega\delta D_{gb}}{kT}\left[\frac{f}{\ln\left(\frac{1}{f}\right) - \frac{1-f}{2}}\right]\frac{1}{r_0^2}p^{\frac{1+\frac{h}{d}}{h}} \tag{2-27}$$

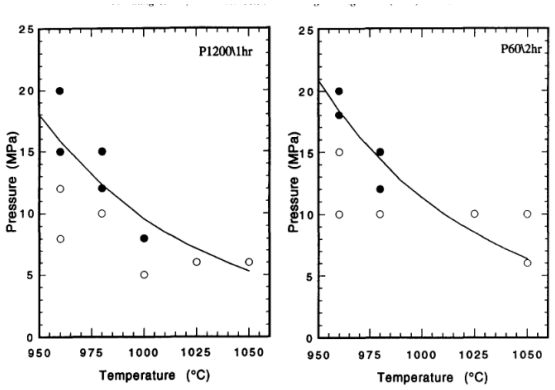


Figure 2.27 - Pilling 1996 experimental theoretical comparison

RuiFang [23]

RuiFang created a dynamic model of void closure that took into account the time dependent effect of bonding. This model used the exact same mechanisms as the Hill model described above. The

effects are initial plastic deformation, surface diffusion from surface sources to void neck, volume diffusion from surface sources to void neck, grain boundary diffusion from surface sources to void neck, volume diffusion from interface sources to void neck, and power law creep, These equations were rewritten for time dependent effects. The main goal of this paper was to predict diffusion bonding processing conditions over relatively small times. Results were that they predicted fast bonding times better than Hill's model as shown in Figure 2.28.

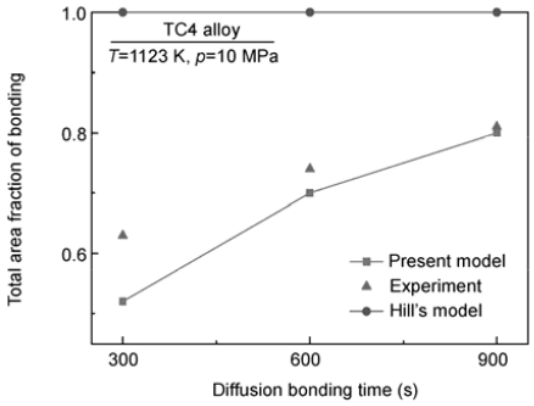


Figure 2.28 - RuiFang's comparison between model and experiments

Li, Tu, and Xuan [24]

Li et al. created a diffusion bonding model that used probabilistic information about the initial voids with Pilling's second model for the equations. Instead of taking one single void size that would relate the closure time, they used a probability distribution function to model the void dataset. Then they used a Monte Carlo simulation to determine the cumulative probability density function. Which, in effect, is the bonding percentage. The time dependent equation below is all from Pilling's deterministic simulation that Li et al. implemented stochastically.

$$t = \int_{f_0}^{f} \frac{-\frac{\sigma_e}{\dot{\varepsilon}}}{\left(3p\sqrt{f} - \frac{2\gamma\sqrt{f}}{r_0} + \frac{6\gamma}{r_0}\right) + \frac{2r_0^2 p}{\ln\left(\frac{1}{f}\right) - \frac{1-f}{2}} \frac{\Omega}{KT}\left[\frac{1+\frac{h}{d}}{h} D_{gb}\delta + 2\Omega D_v\right]} df \qquad (2\text{-}28)$$

They then used the probability density functions below to model the void height and void radius respectively. For this dataset, parameters for the probability density function are μ=0 (normal distribution mean) , σ=1.25R_a (normal distribution standard deviation) and λ=30μm (exponential distribution rate parameter). Li then compared his model with other models used previously in published literature as shown in Figure 2.29.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (2\text{-}29)$$

$$f(x) = \lambda e^{-\lambda x} \ (\lambda > 0) \qquad (2\text{-}30)$$

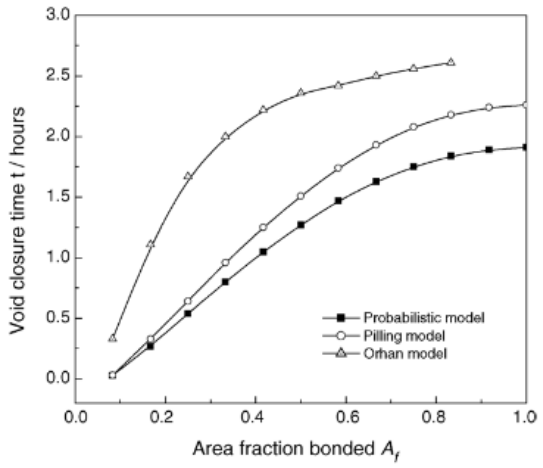


Figure 2.29 - Model performance comparison

Kulkarni, Mamidala, and Sanders [25]

Kulkarni et al. modified Li's stochastic diffusion bonding model by applying it to both the bonding between two identical titanium alloys and bonding between different alloys. Kulkarni estimated the initial geometry of the voids by taking surface roughness profiles and simulating how they could contact together. The surface roughness profiles measured by a contact profilometer. The distribution functions based on those surface roughness profiles were then used to determine what statistical height and width should be used as inputs into the probabilistic equations. Figure 2.30 shows the details. Finally the different material properties of the different alloys were determined using rule of mixtures for the properties of alpha and beta titanium. The results of the analysis were used to determine the effects of temperature, void height, void width, and grain size on bonding time.

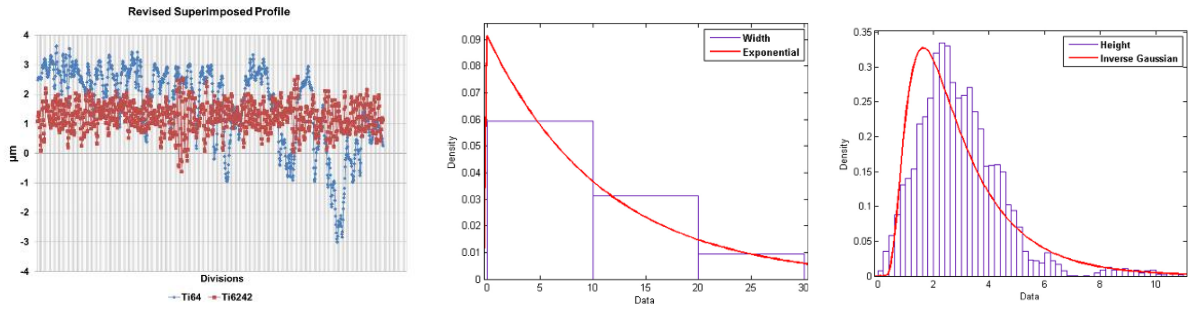


Figure 2.30 - Superimposed surface roughness profiles to determine the geometry of voids formed and their resulting width and height distributions

2.4.4 Experimental Work in Diffusion Bonding

There has been a significant amount of experimental work done on diffusion bonding and it remains the only way to correctly determine processing conditions. Some of the experimental data

is proprietary and has not been published while some has been published in a reduced form (for example, Kulkarni published the temperatures in her thesis on a relative scale). Many of the papers discussing modeling in the previous chapter were validated by experimental work. The diffusion bonding experiments are analyzed with micrographs, structural tests, and/or with non-destructive methods. This section will deal with the micrograph analysis and experimental setups along with structural and non-destructive testing. The bulk of current diffusion bonding research published addresses intermetallics, bonding with an interlayer, and liquid diffusion bonding. This section will only deal with solid state diffusion bonding between conventional titanium alloys.

Experimental work for diffusion bonding typically entails doing a variety of experiments. Bonding data is collected using small parts, called coupons, which are two flat sheets bonded together. This allows one to determine the adequate bonding conditions with a minimal amount of material and labor. Coupons can be constructed using a variety of methods. A typical setup is the bonding press, where multiple coupons can be bonded at the same time by heating them and sandwiching them in between a piston or linear actuator that can provide a calibrated pressure.

The primary data for diffusion bonding comes from micrographs that are taken of the bondline area. The voids are typically quantified as the percentage of bonded length versus to the total length along the bondline. This gives the bond percentage. Structural testing of diffusion bonds have been attempted many times with varying success.

2.4.5 Structural Testing of Diffusion Bonding

Thin sheet geometry and bond strength are some difficulties with structural testing of diffusion bonding. Many tests accurately asses the bonding strength with low quality bonds such as the peel and shear tests. Figure 2.31 shows a typical lap shear test coupon while Figure 2.32 shows a

variety of shear testing setups. However, when the bond strength increases, specimen geometry becomes extremely important. Many designs have been developed with many different specimen configurations with varying success. The most common and prominent structural test was first performed by Pilling in 1984 where he took two flat sheets bonded together with sections of the specimen machined out so there is only a limited area that is bonded and overlaps between the sheets (as shown in Figure 2.31). The specimen is pulled in tension and creates a shear stress at the overlap area. This type of shear specimen was refined by Salehi and has been used in a variety of forms.

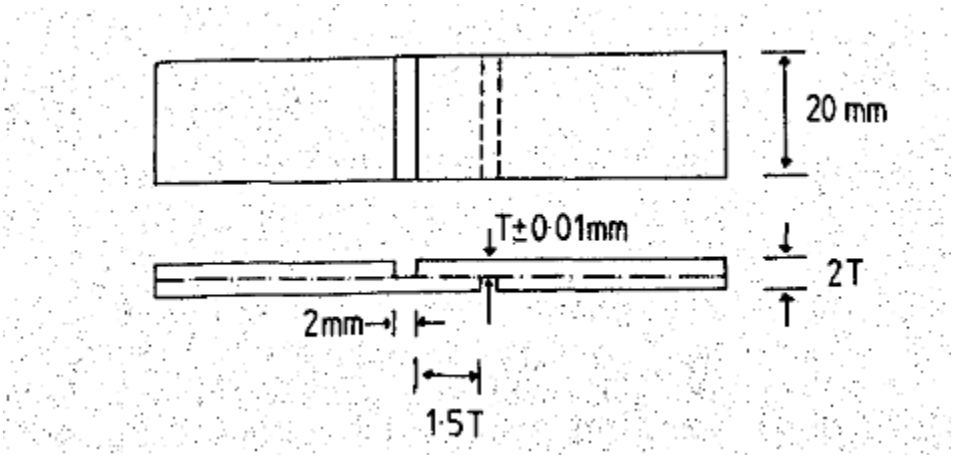


Figure 2.31 - Pilling lap shear [16]

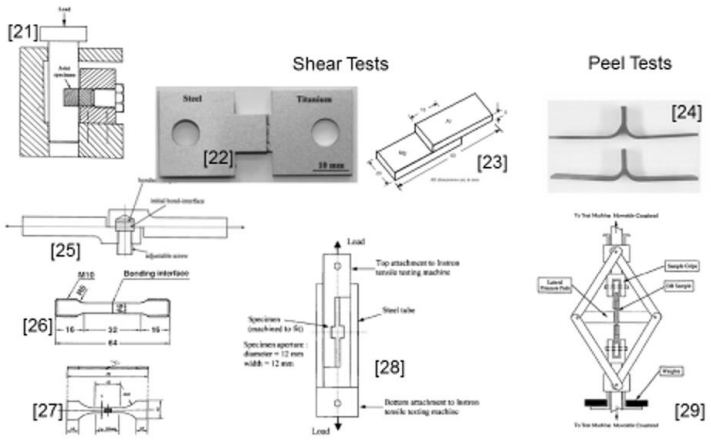


Figure 2.32 - Diffusion bonding test specimens

The common deficiency among all these test is that they will test well for poor bonding but not for good bonding. When the bonding interface approaches the strength of the base material then the specimen must fracture along the bondline based purely on its geometry and not because the bondline is weakened significantly. Poor specimen geometry results in behavior like that shown in Figure 2.33.

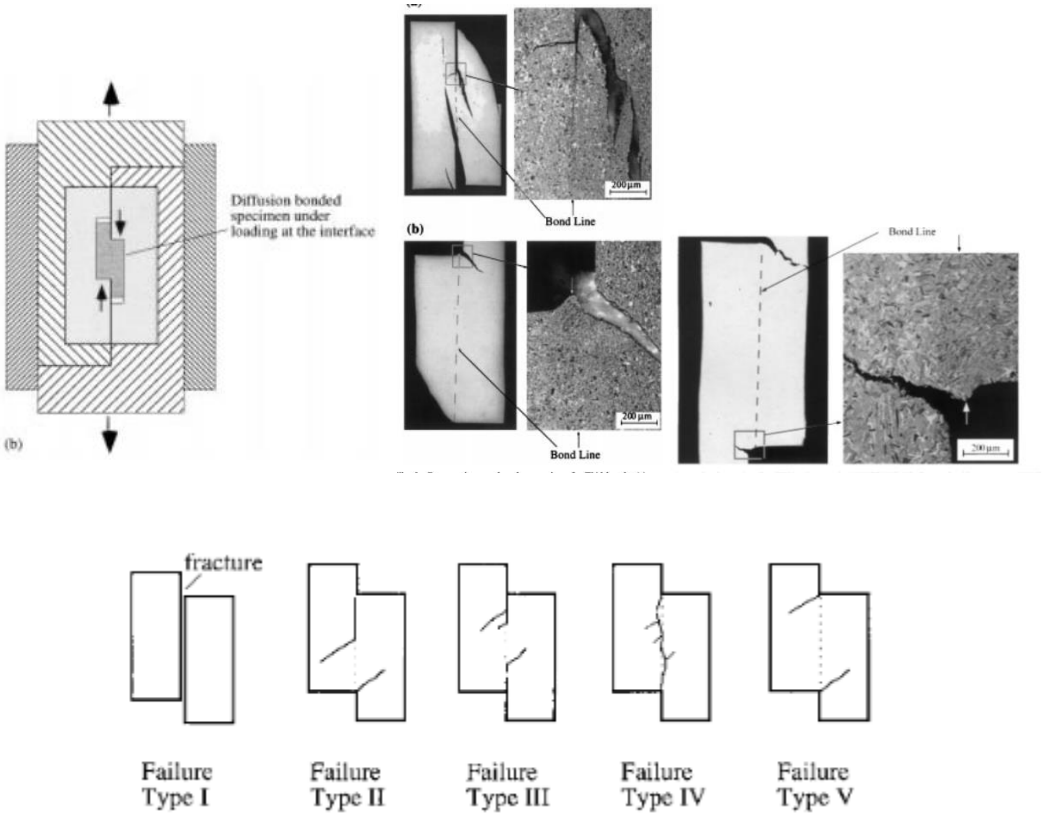


Figure 2.33 - Shear testing with poor specimen geometry [26]

If the specimen geometry is adequate then one can see the results in the fracture surface even for good bonding. The shear surface will be smeared.

2.4.6 Non-Destructive Testing

Non-destructive testing (NDT) of diffusion bonding is an attractive way to circumvent the common analysis methods of taking micrographs and determining the extent of the voids. Structural testing cannot be performed on parts that are to be sold and micrograph analysis only looks at a small surface and is time consuming and expensive. NDT can be used in the field and on parts without damaging them. A variety of non-destructive tests have been used with diffusion bonding with varying results. This is because NDT of diffusion bonding has many common, inherent problems.

Most of these are due to the fact that as the voids shrink in size it becomes increasingly difficult to detect them reliably. Many methods increase in accuracy as the bonding becomes poorer and the voids become larger.

Derby et al. [27] looked at non-destructive testing and provided a summary of the different testing methods in 1983. Radiographic NDT is where radiation is passed through the part and the transmission is picked up on film on the opposite side of the part. Derby states that radiography is too coarse to detect the voids unless they are unusually large. Ultrasonic NDT utilizes an ultrasonic probe that pulses shear or plane elastic waves in the sample and measures the reflected signal. Conventional ultrasonic NDT is of no use for diffusion bonding. This is because at typical ultrasonic frequencies, elastic waves have a wavelength of around 100μm to 200μm which is significantly larger than the voids present in most diffusion bonds. Therefore there is negligible reflection from the voids at these frequencies. This has been confirmed experimentally. There is a technology called acoustic microscopy which uses gigahertz frequencies of ultrasound and a computer controlled stage to map out the reflection intensity in a 2D pattern. However, since the voids are on the order of the grain size, acoustic microscopy has significant random reflections from the grain boundaries themselves as shown in Figure 2.34. This can be alleviated by going to lower frequencies but with the complete loss of ability to detect small voids.

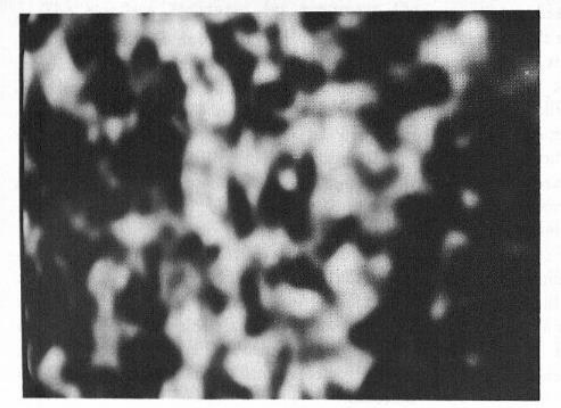


Figure 2.34 - Image of a sample at high frequency with the variation in contrast due to scattering off grain boundaries [27]

Derby looked at another technique using electrical resistance. A section of uniform dimensions was cut and electrical contacts were positioned on either side of the sample. Electrical current was passed through the sample and the drop in voltage is measured. This voltage drop is calibrated with the drop across an unbonded sample. The voltage drop, δV_b, theoretically follows some simple equations with voids in a uniform sample and is shown below. Figure 2.35 shows a plot of this equation that relates the instrument sensitivity with the bonded fraction. While it is easy to detect voids in poorly bonded samples, as with the other technologies, it gets increasingly difficult to detect the voids as they become smaller. This method has been researched further with potentially viable results [28, 29].

$$R = \frac{\rho l}{A_1} \;\; ; \;\; \delta V_b = -\frac{2\rho I h(a^2-1)}{A_1^2(a^2+1)\sin\frac{\pi}{2}\frac{A_2}{A_1}}\delta A_2 \tag{2-31}$$

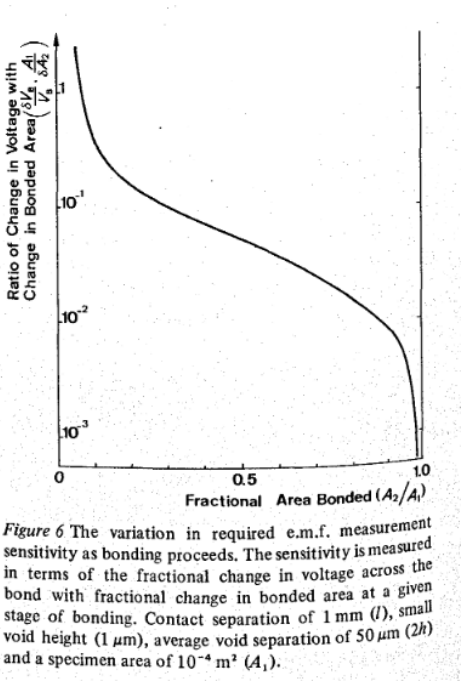


Figure 6 The variation in required e.m.f. measurement sensitivity as bonding proceeds. The sensitivity is measured in terms of the fractional change in voltage across the bond with fractional change in bonded area at a given stage of bonding. Contact separation of 1 mm (l), small void height (1 μm), average void separation of 50 μm ($2h$) and a specimen area of 10^{-4} m^2 (A_1).

Figure 2.35 - Relationship between resistive instrument sensitivity and bonded fraction [27]

Ultrasonic NDT has been revisited throughout the years for diffusion bonding. Like Derby determined, there still isn't a viable way to obtain a reflection off the void if the void is sufficiently small. However, there are some more complex methods to determine bonding information and the most prominent of these is the interfacial stiffness method. This method uses the effect that the different elastic properties at the interface have on the dominant mode shapes of the elastic wave as well as the diffraction of the wave. This method is a promising way of determining the diffusion bonding percentage through ultrasonic methods [30–32]. Figure 2.36 shows that promising results can be obtained by analyzing the ultrasonic waves such as with the phase shift method.

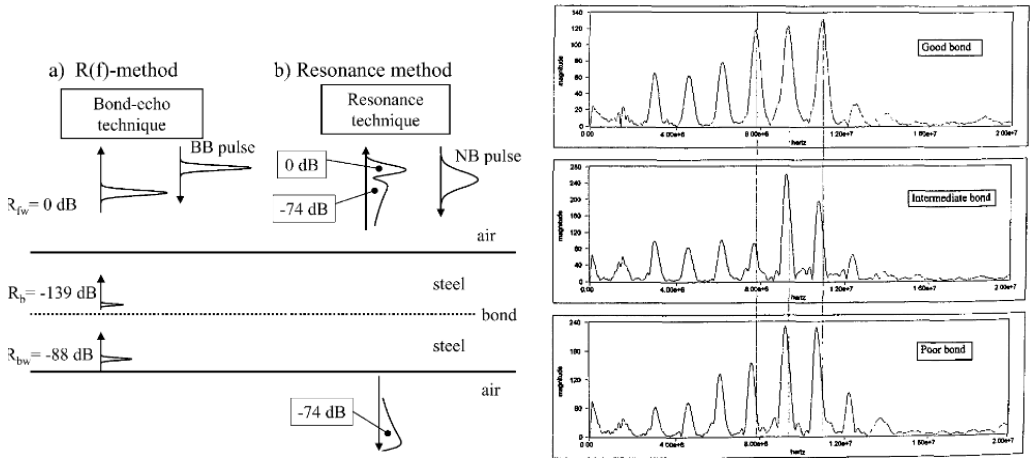


Figure 2.36 - Difference between phase shift bond echo method and resonance method (left) [30] along with an example of the phase shift method on diffusion bonds (right) [31]

The basis for the equations in the phase shift method are presented below. It is simulated as two plates connected by springs. It assumes normal incidence along the z direction and has the boundary conditions of equating the displacements and stresses at the spring interface between the two plates as shown in Figure 2.37.

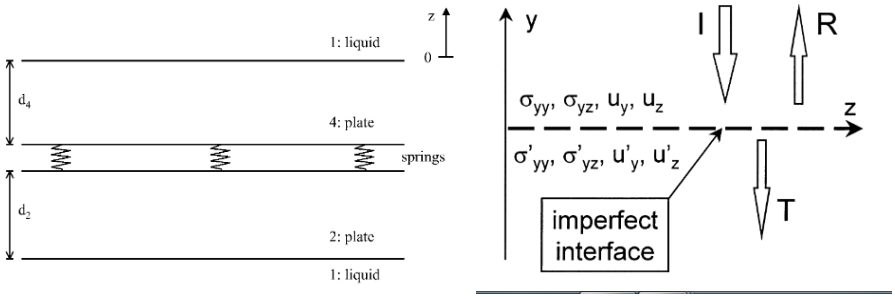


Figure 2.37 - Phase shift layouts [30, 32]

In these equations, ω is the circular frequency, k_1 is the wavenumber, d_1 and d_2 are the thicknesses of the plates, R is the reflection coefficient, Z_i is the impedance, and T is the transmission coefficienent. The incident wave is:

$$u_i = e^{-i\omega t}e^{-ik_1 z} \tag{2-32}$$

The reflected wave is:

$$u_R = Re^{-i\omega t}e^{-ik_1 z} \tag{2-33}$$

Out the lower liquid boundary the transmitted wave is [30]:

$$u_T = Te^{-i\omega t}e^{-ik_1(z+d_2+d_4)} \tag{2-34}$$

After a transfer matrix formalism the relfection and transmission coefficients are [32]:

$$R(\omega) = \frac{Z_2 - Z_1 + i\left(\frac{\omega}{K_n}\right)Z_1 Z_2}{Z_2 + Z_1 - i\left(\frac{\omega}{K_n}\right)Z_1 Z_2} \quad ; \quad T(\omega) = \frac{2Z_2}{Z_2 + Z_1 - i\left(\frac{\omega}{K_n}\right)Z_1 Z_2} \tag{2-35}$$

With identical materials this gives:

$$R_I = \frac{i\omega}{1 - \frac{i\omega}{\Omega}} \; ; \; T_I = \frac{1}{1 - \frac{i\omega}{\Omega}} \quad ; \quad \Omega = \frac{2K_n}{Z_1} \tag{2-36}$$

By taking the Fourier Transform you can get a phase shift related to the interfacial stiffness. For large grained materials where the grain size is significantly larger than the void size, this method works very well. However, with fine grained materials, some additional techniques can be used to detect bond quality. One group found that using the real part of a symmetrically reflected signal allow for separation of good quality bonds and voided bonds as shown in Figure 2.38 [33].

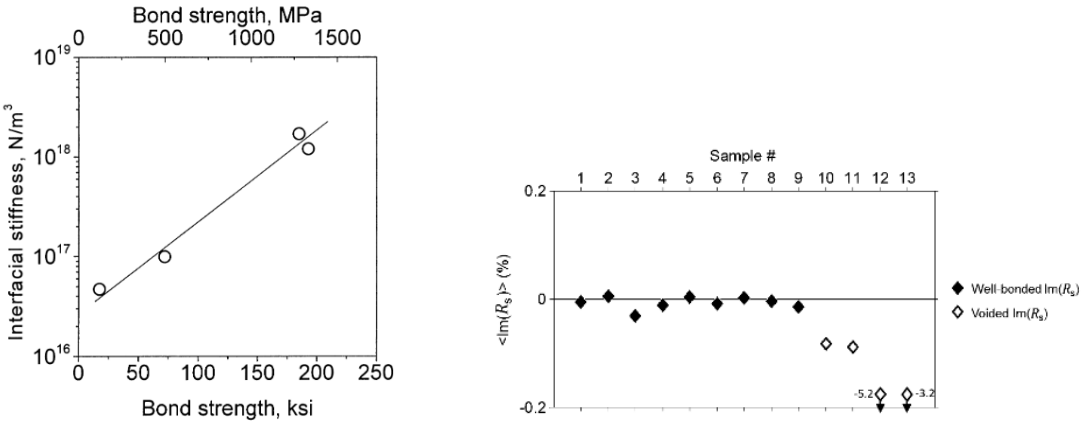


Figure 2.38 - Results of the phase shift interfacial stiffness method with large grains (left) [32] and small grains (right) [33]

2.5 Atomistic and Mesoscale Solutions to Diffusion Bonding

Atomistic methods are an attractive way to solve bonding problems because they provide detailed information about the basic mechanisms of bonding and the strength of the bond. Many bonding problems have been solved with molecular dynamics calculations with reasonable success. Among them are thin film deposition [34], friction welding [35], and explosive welding [36]. Molecular dynamics typically solves problems on the scale of nanometers and picoseconds. This poses a significant problem for diffusion bonding which occurs on the scale of micrometers and hours. Below is a summary of the research on diffusion bonding that has included molecular dynamics and some potential mesoscale methods that have not been applied to diffusion bonding but might be used to solve the problem.

2.5.1 Published Diffusion Bonding Modeling with Molecular Dynamics

There has only been a handful of papers published on diffusion bonding with molecular dynamics which will be discussed below. They all deal with the problem of bonding at the nanoscale for short times. There have been other papers on diffusion processes that used molecular dynamics but this section only deals with papers that are specifically geared toward diffusion bonding to illustrate the possibilities and deficiencies of the method. The first and second papers on the subject were written by Chen and used the embedded atom method (EAM) potential. So far, only intermetallic bonding has been calculated with MD. The first paper by Chew [31] was on the CuAg bonding at 1150 K for 50, 100, and 150 MPa. Bonding time was 200 ps and was unloaded and cooled down after bonding. For this bonding time they were unable to see any diffusion until the stress reached 100 MPa. The results of this are shown in Figure 2.39. They concluded that stress plays an important role in bonding, that the thickness of the interfacial region depends on stress, that the interfacial region was amorphous under high stress, and that cooling would transform the amorphous region into an ordered one if the cooling rate was significantly low.

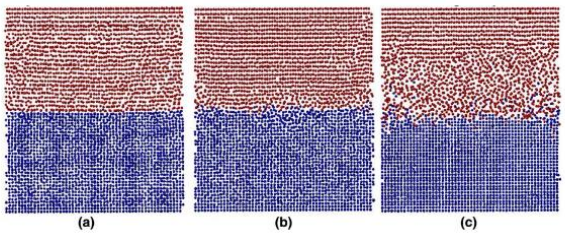


Figure 2.39 - Simulations for 200ps at 1150K with Cu-Ag bonding at a) 50MPa, b) 100MPa, and c) 150MPa

The same group did an investigation of diffusion bonding for copper and aluminum [34]. The methods were the same, however, the simulation included a spectra of different conditions as

shown in Figure 2.40. Firstly, they used flat surfaces bonded at different temperatures for 600ps for the same pressure of 20MPa. Next they used a variety of surface profiles for different temperatures under the same pressure. They concluded that temperature significantly affects bonding. They also concluded that, with the rough surfaces, the bonding can split into the stages of plastic collapse before heating, collapse after heating, and full contact.

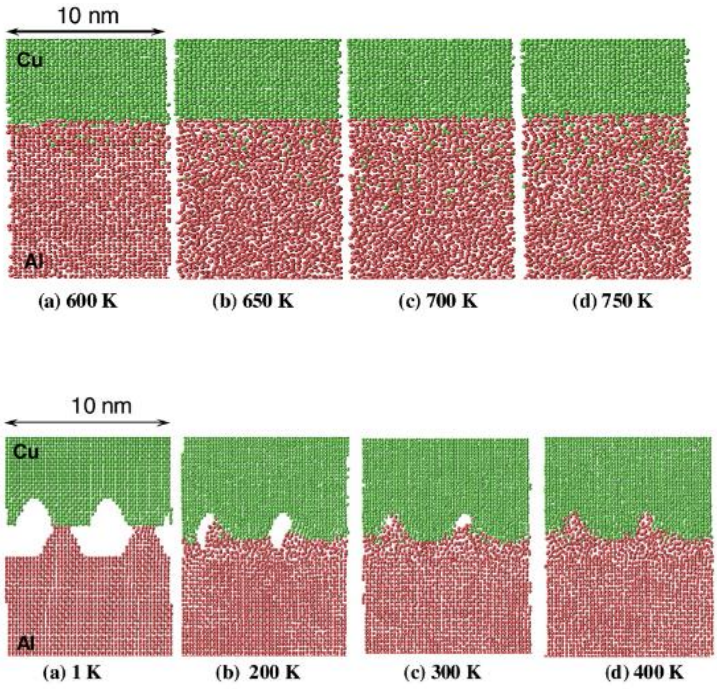


Figure 2.40 - Flat (above) and rough (below) surfaces bonded at 20MPa for 600ps [28]

Li, et al. [32] dealt with the verification of diffusion coefficients in aluminum copper bonding. The embedded atom method was used with Nose-Hoover thermostat and constant NPT (atoms N, pressure P, and temperature T) for a low temperature for 20 ps. Then they rapidly heated the system up to temperature and held it for 6 ns. The diffusion of atoms was determined and was used to calculate the diffusion coefficient and activation energy. They concluded that the diffusion

coefficient was similar to measured diffusion coefficients for thin interfaces and that diffusion increased with temperature.

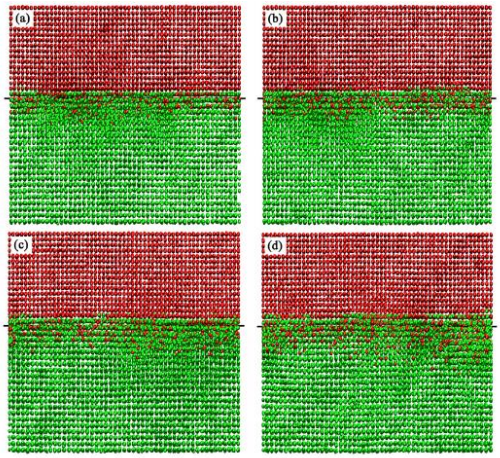


Figure 2.41 - Atmospheric pressure for 6ns with temperatures of a) 650K, b) 700K, c) 750K, and d) 800K [32]

Li, et al. [37] published an additional conference paper on diffusion bonding with MD using the same parameters as above with the exception of using a three dimensional unit cell rather than two dimensional. They used different surfaces with different pressures and temperatures. Simulations were performed for 650ps. Their conclusions were that bonding can be divided into three zones, rapid growth, stagnation, and slow growth. The voids were concluded to be filled by aluminum and not the copper. They also concluded that pressure has little impact on bonding. Results are shown in Figure 2.42.

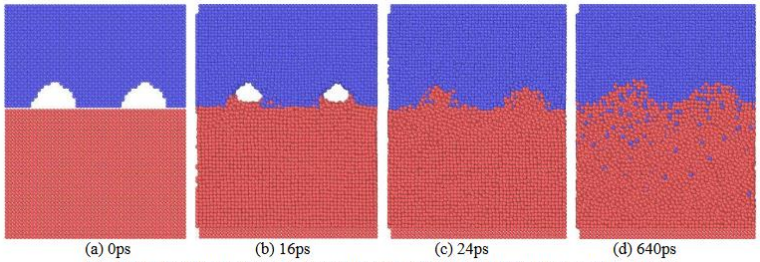


Fig. 5 Deformation figures of a cross-section for case (2) at different time (700k, 20Mpa)

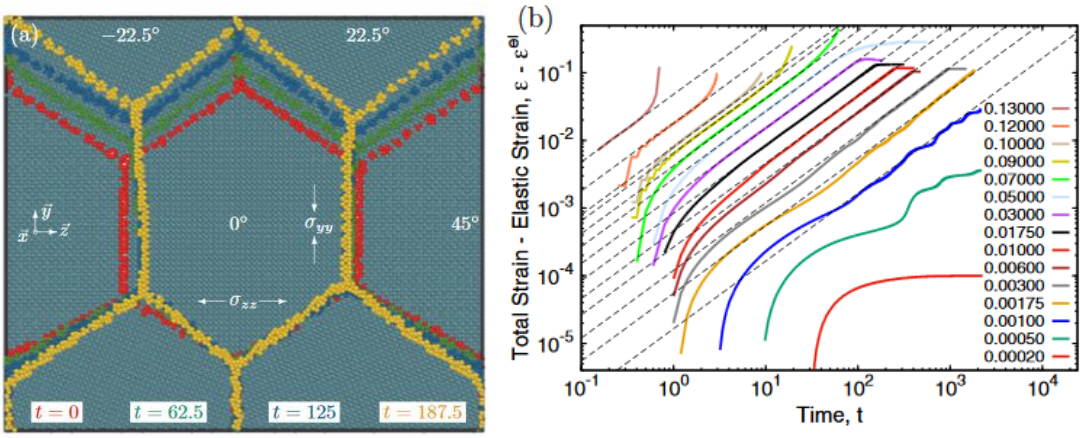
Figure 2.42 - Two dimensional view of 3D cell with surface roughness

2.5.2 Possible Alternative Methods Researched

The molecular dynamics modeling in Section 2.5.1 has been done at the length scale of nanometers and the time scale of at most a few nanoseconds and are observing good mixing of atoms at the bonding interface [37]. These papers confirm that if surfaces contact during diffusion bonding then the part will have sufficient time to form a good weld before the heat and pressure are removed. However, they do not provide any insight into the dominant mechanisms in void closure or prediction of the bonding parameters. Initial voids formed are going to be on the scale of micrometers and a significant amount of material will have to be transferred to the void before bonding occurs. There are some other methods that could be used to address this issue of scale. Several possibilities are the phase field methods, dislocation dynamics, and phase field crystal methods. None of these have been applied to superplasticity or solid state diffusion bonding and could shed new light to the specific mechanisms in the process especially with multiple alloys.

Phase field crystal methods are a relatively new method of using phase field equations with molecular dynamics. Specifically it is a phase field implementation in the molecular dynamics discrete element framework. These equations use an atomistic formulation but with an interatomic potential from the phase field method allowing the simulation to operate on time scales well

beyond the characteristic vacancy diffusion time. Therefore this method can shed light on the long term diffusion processes. Recently this method has been used to model Nabarro-Herring creep with good accuracy in nanocrystals [38]. An example of the results that can be obtained by phase field crystal methods are shown in Figure 2.43. This figure shows the elongation of the crystal in the stressed direction and the simulated creep response for different sized nanocrystals.



a) b)

Figure 2.43 - Creep in nanocrystals using phase field crystal method showing how the crystals elongate in the direction of the stress (a) and how the creep response is for different sized nanocrystals (b) [38]

Another method that could be useful is a special condition of discrete dislocation dynamics (DDD) [39] which has also been used to model diffusion creep as shown in Figure 2.44. Discrete dislocation dynamics is a Newtonian physics method that operates at atomistic length scales but under the assumption that the basic energy of a dislocation in a crystal is the same at any point and therefore only the energy of its movement needs to be calculated. Dislocation climb and diffusion are likely to be the effects that dominate the diffusion bonding process as well as superplasticity.

Dislocation dynamics uses the Peach-Koehler force for dislocation motion and the gradient of chemical potential for vacancy diffusion.

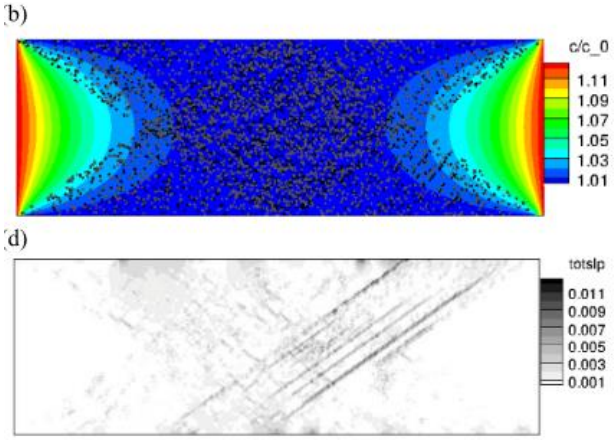


Figure 2.44 - Concentration of dislocation in a tensile creep simulation (top) and the resulting slip contours at t=4000s [39]

2.6 Phase Field Methods

Phase field methods are attractive because they are a set of diffusion oriented differential equations that have never been used to model diffusion bonding. However, they have covered most of the physics involved. There are two main types of phase field formulations. The first is based on a microscopic theory where the material properties used in the phase field equations are based on theoretical work researches have done energies and mobilities are in a solid diffusive framework. This models diffusion well and has been applied to solid phase transformations, magnetic properties, and the influence of elastic strains on the microstructure. This has also been related to dislocation dynamics, crack propagation, and solid state sintering [40–42]. The second main type

of phase field equations are based on phenomenological and stochastic equations. This is mainly applied to solidification and growth of grains during the liquid to solid transition.

One commonality between all phase field calculations is the diffuse interface. All variables are continuous within the simulation dimensions and therefore are continuous across the interface boundaries. This means that there is no need to track the boundaries or restrict the simulation with boundary conditions at each interface. Unlike with the sharp interface methods where one would have to apply boundary conditions at the grain boundary interface, contact algorithms for structural transfer, and diffusive transfer between the two materials. It would also mean that the resolution of the simulation must be very fine in order to take into account the sharp transition between boundaries. With diffuse interfaces the locations of the boundaries in the simulation are smooth transitions. This can be used to solve problems very similar to diffusion bonding such as the sintering of a void within a medium with conserved advection shown in Figure 2.45. The second formulation has been applied to transient liquid diffusion bonding. This is a process that is very related to this research but with different physics at the bonding interface. Transient liquid diffusion bonding is where two base materials are bonded with an interlayer that has a melting point below the diffusion bonding temperature. The interlayer forms an alloy with the base materials and demonstrates sharp grain growth to form the bond. The first set of equations are most promising for this research and are discussed below.

Simulations have been done for many of the phenomena that dominate solid state diffusion bonding including the changes of the shape of voids in a stress field, multiple materials interfacing together, microstrains due to mixed microstructures, changes in microstructure due to transport of species, grain growth due to diffusion, and elastoplastic flow [43].

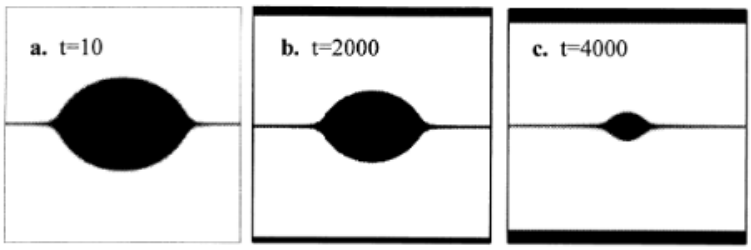


Figure 4. Simulated pore shape evolution and sample shrinkage during sintering in the presence of rigid-body motion.

Figure 2.45 – Phase field pressure sintering with rigid body motion of grains [40]

The basis for the simulations are the Cahn-Hilliard non-linear diffusion equation [44] which describes phase separation and the time dependent Allen-Cahn equation [45] which describes order information. Both equations relate the temporal evolution of variables with respect to a bulk energy functional with the primary difference between the two equations being that the Cahn-Hilliard equation deals with conserved variables while the Allen-Cahn equation deals with non-conserved variables. With the right formulation of coefficients and the correct functional, these equations can be used to describe many phenomena. The Cahn-Hilliard equation (Equation 2.35) minimizes the free energy of the whole system at the rate of a given mobility for a set of conserved variables usually called phase fields. The Allen-Cahn equation (Equation 2.36) has the same mechanism but for non-conserved variables called order parameters. The phase fields describe the conserved parts of the system such as the mass. The order parameters describe characteristics of sections of the system such as the phase. Using this as an example, one would have phase variables that conserve the mass after initialization but whether the mass is alpha phase or beta phase does not need to be conserved and is free to change as the simulation heats or cools. The energy functional, F, for this equation consists of the bulk free energy, interfacial energy, elastic strain energy, and energy due to electrostatic or magnetic terms. In the equations below (2.35-2.37) c_i is

a conserved variable, M_i is the associated mobility, η_j is a non-conserved variable, and L_j is the non-conserved mobility.

$$\frac{\partial c_i}{\partial t} = \nabla \cdot M_i \nabla \frac{\delta F}{\delta c_i} \quad (Cahn - Hilliard) \tag{2-37}$$

$$\frac{\partial \eta_j}{\partial t} = -L_j \frac{\delta F}{\delta \eta_j} \quad (Allen - Cahn) \tag{2-38}$$

$$F = F_{bulk} + F_{int} + F_{el} + F_{mag} \tag{2-39}$$

There are formulations for each of these terms according to Cahn-Hillard. The difficulty in phase field analysis is the formulation of the free energy functional. This will depend on the simulation required. The free energy functional equation is based on the bulk chemical free energy, a function of the internal energy of the phase, along with any interface energies and elastic strain energy that is additionally included. The Gibbs free energy is commonly used to determine the bulk free energy which is commonly included in many thermodynamic databases and can be computed through the CALPHAD (CALculation of PHAse Diagrams) method. The elastic strain energy can be determined by mechanical simulations using the elevated temperature elastic moduli of the different phases. The interface energies will be the energies included from the different boundaries in the system, mainly the boundaries along the grains and along the surface of the voids. Typically the free energies are mapped to polynomial functions. Grain boundary energy, or interfacial energy, is projected onto the free energy surface. Note that the projection is still spatially dependent despite being a diffuse interface since it is still a function of the order parameters, η. The projection (Equation 2.38) for the interface energy, f_{int}, contains the constants of gradient energy, κ, and potential well height, ω, which can be combined to select the interface energy and the interface width.

$$f_{int} = \frac{\kappa}{2}\nabla\eta^2 + \omega g(\eta) \qquad (2.38)[46]$$

Phase field methods are difficult to apply to large scale problems since the diffuse interface still has very small features. The energy of the interface is directly related to the interface dimensions. In a traditional microstructure the minimum feature is the grain boundary which is on the order of a few nanometers. This makes it very difficult to simulate a box tens of microns wide while still maintain good mesh quality at the relatively small interface. There are several ways of treating this problem but the best way is to decouple the interface width from the interface energy. This allows the mesh to be coarse and at a reasonable scale to the rest of the model while still obtaining the same energy that would be present. This method was pioneered by Kim-Kim-Suzuki and is called the KKS method [47]. Other methods are the grand potential formulation [48] and other formulations [49]. While KKS does allow for larger length scales, it's not without issues. The interface energy is no longer at a concentrated location and without anisotropic diffusion constants, there can be errors involved in diffusive transfer out of the energetic interface area that are not actually part of the real interface. Another problem is now that the interface is much larger than reality, other quantities that are computed over that interface are also over a larger area. This applies to diffusion where the interface typically has higher diffusion constants. It also applies to the elastic strain energy where over the interface there is some switching based on the interface. There are several methods to blend the different interfaces together [50].

There are relatively few models for titanium in phase field. Specifically there is predominantly one group that has worked on titanium phase field. The group of Y. Wang from Ohio State University has been working on variant selection during thermomechanical processing of titanium. Variant selection is the process by which, when titanium or other metals cool and produce another

phase, the phase precipitates as a function of the orientation of the initial phase[46, 51–53]. The alpha beta titanium model used in these papers use three different elements but only two phases. The primary purpose is to investigate how shear stresses change the selections of the precipitate phases. To perform a diffusion bonding simulation or a superplastic simulation with voids one needs a more complex phase field model than has been published for titanium. Some results of Shi, et al.'s model are shown in the Figure 2.46 where one can create a phase field model that changes the variant selection mechanisms based on an initial polycrystalline beta orientation resulting in a change in crystal texture as time evolves.

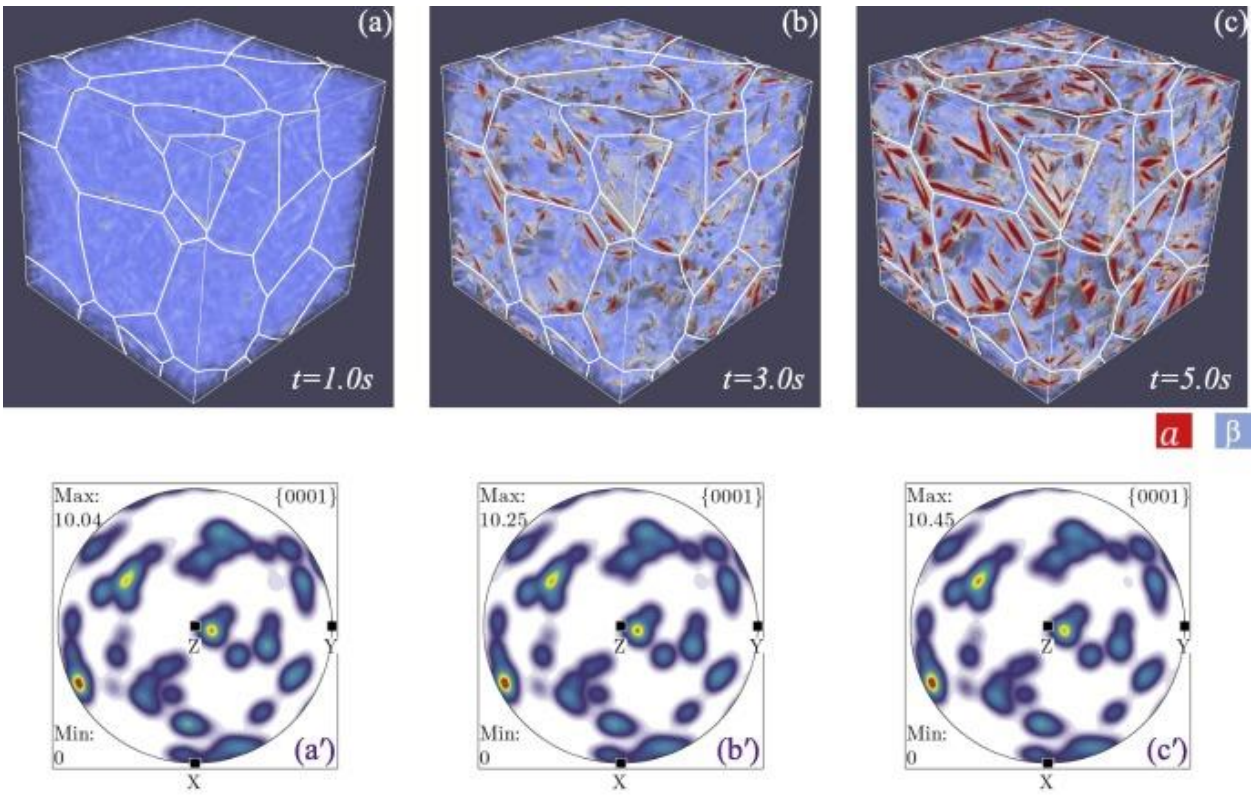


Figure 2.46 – Shi, et al. Variant selection in alpha+beta titanium during cooling under the influence of stress. The simulation takes into account elastic strain energy. [46]

While alpha+beta titanium is mostly neglected with phase field, there are other systems that are similar in nature if not in exact composition. The physics of diffusion bonding specifically are very similar to that of pressure sintering [40, 41, 54, 55]. Pressure sintering is often modeled in a similar manner to diffusion bonding with bulk creep of the material, diffusion along the surface of the material, and diffusion along the grain boundaries. The major differences are that, with diffusion bonding, the void is constrained within the material while, with sintering, the voids are unconstrained. The sintering particles are also unconstrained.

Since one of the major differences between previous models and one that can handle diffusion bonding is the existence and closure of voids, one must examine how voids have been treated within the phase field framework. Traditionally, a void is treated simply as the absence of the conserved phase variable [41]. In other words, a conserved variable with a value of zero can be used to model the voids. This can be seen in the pressure sintering models as well as some of the nuclear models.

2.7 Molecular Dynamics of Grain Boundary Sliding

Molecular dynamics is a discrete element method of solving computational problems. Each particle has an interaction potential energy with other particles that determines the forces based on distance between particles. It uses ab initio computations to determine the energy potentials between particles. With respect to superplasticity, there has been some research into nanocrystalline superplasticity [56–58]. However, because of the small scale, most of the research has been in the form of grain boundary sliding simulations. Chandra, et al. investigated it with respect to superplastic deformation [4]. Many more simulations have been done on grain boundary sliding using essentially the same setup to investigate how vacancies effect sliding , how

orientation effects sliding , and other effects [59–63]. The essential setup for grain boundary sliding with molecular dynamics is shown in Figure 2.47. The setup uses one half of its volume as one phase and the other half a different phase. The end of one side is fixed while the other side rotates. Periodicity is usually employed along the moving direction.

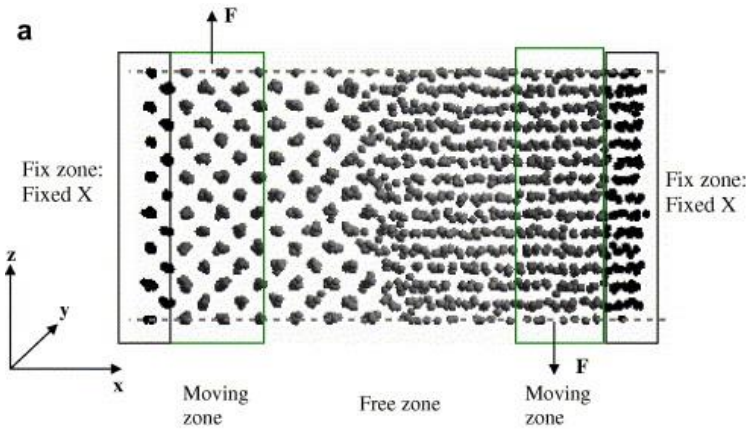


Figure 2.47 - Grain boundary sliding setup [63]

2.8 Summary

The research provided has had several consistencies and inconsistencies. The superplastic forming research so far has been quite robust in its determining constitutive models that will approximate superplasticity with finite elements. However, there is still research left to be done on the exact mechanisms involved and the use of more complex methods to confirm them. Diffusion bonding calculations all use very similar diffusion and creep or superplasticity equations and the main thrust of research in several articles has simple been to apply the same equations to different void geometry. These equations depend greatly on the specific material constants used in simulations and often cites the lack of detailed information about the material and surface as a main reason for discrepancies. Molecular dynamics approaches to diffusion bonding have been over time and

length scales that are too small to do a full bonding simulation. Phase field methods are an ideal way to handle this problem and haven't been performed before. Structural destructive testing has yet to have an established reliable method for testing diffusion bonded samples. Finally, non-destructive testing also does not have an established reliable method for testing diffusion bonded samples. The research done in this thesis will attempt to solve these problems.

Chapter 3 Research Goals and Objectives

While aspects of diffusion bonding and superplastic forming have been researched since its inception, several key components of the process are left unanswered. Primarily, the modeling has not been successful in accurately determining how different alloys will bond together. This issue leaves determining the diffusion bonding processing conditions solidly in the realm of trial and error experiments based on rough guesses. The expense required to do the tests mandates that the results are a closely guarded secret and limits the implementation of superplastic forming and diffusion bonding. Secondly, there has been very little research on the exact effects of superplasticity on diffusion bonding. It has never been necessary to characterize this phenomena since the four-sheet process and multi-sheet processes are so infrequently used. The goals of this dissertation are broken down into three different sections: characterizing the effects of bonding at different temperatures with dissimilar and similar alloys along with characterizing the effects of superplastic forming on diffusion bonding, developing diffusion bonding and superplastic models, and finally developing additional testing techniques and modeling that would assist implementation of the superplastic forming and diffusion bonding process.

3.1 Characterization of dissimilar and similar diffusion bonding process, four-sheet process, and raw materials (pre-SPF)

The purpose of this section is to investigate how diffusion bonding operates for similar and dissimilar diffusion bonding, four-sheet bonding, and characterizing the raw materials involved. This will be done with the following methods:

- Bonding performance: The percentage of bonding in the samples is the primary metric that will be used for the bonding process and has been computed for the feasible samples since it is highly correlated with the structural capabilities of the bond.
- Effects of composition differences in dissimilar alloy bonding: With dissimilar diffusion bonding, the differing alloying elements will migrate at these temperatures forming a combination alloy at the bonding interface. How these elements migrate is investigated through the use of scanning electron microscopy along with energy dispersive xray spectroscopy.
- Surface roughness changes with SPF: The surface changes during superplasticity which can substantially affect bonding. The potential effects of surfaces has been looked at.
- Characterization of the raw materials and process conditions: As discussed in the literature, bonding is based on the time, temperature, pressure, and the specifics of the material and surface conditions. Therefore the process conditions of all samples are determined and documented. Samples of raw materials used in this research are characterized for both surface morphology and initial grain structure.

3.2 Modeling of diffusion bonding and superplasticity

The primary task and goal of this research is to improve diffusion bonding models so that the bonding quality based on processing conditions can be predicted. The inputs are the parameters controlled by a press (pressure schedule and temperature), the die geometry and sheet welding, a simulation of the forming process to determine stresses and contact time, along with information collected about the materials involved (surface morphology, microstructure, and material behavior). The outputs are how the bonded percentage changes for given processing conditions. The four-sheet process will be modeled for superplasticity with conventional finite element methods in both two and three dimensions. An attempt has been made to illustrate some of the more complex aspects of superplasticity such as how grain boundary sliding operates.

3.3 Efforts to increase implementation of SPF/DB

With better bonding models and an understanding of how superplasticity influences diffusion bonding, there could be accelerated research and development of SPF/DB structures. However, it is important to be able to confirm the structural effects and be able to test produced parts.

- Structural testing: Efforts have been made to develop a structural test that can accurately determine the shear strength of the resultant diffusion bond.
- Non-destructive testing: Ultrasonic tests have been attempted and simulations were created to explain the results.

Chapter 4 Two Sheet Diffusion Bonding: Analysis of Raw Materials and Experiments

4.1 Introduction

In this chapter, the raw material analysis, the experimental approach, and the theoretical calculations are presented. What follows is an explanation of the materials, an explanation of the experimental diffusion bonding setup, and the results of the bonding setup. Work has been done to analyze the raw materials by scanning electron microscopy and energy dispersive xray spectroscopy (SEM/EDS) and surface profilometry along with work to characterize the microstructure and bonding performance of the two sheet DB samples. In addition, the structural and non-destructive test work done on the two sheet DB samples.

4.2 Material Systems and Composition

Several titanium alloys were considered. For the two sheet diffusion bonding samples there were both alpha, alpha+beta, and beta alloys. Four-sheet samples were based on alpha-beta alloys. The base alloys used in this research are shown in Figure 4.1 and Table 4-1. This research used several experimental alloys that were developed recently. Ti54MX, T6242SX, and Ti BETA 21SX were all proprietary experimental alloys which had undisclosed compositions but were slight modifications from the base alloys listed in the table. Of these alloys, Ti54M, Ti64 and ATI425 were all alpha beta alloys. Ti1100, Ti834, and Ti6242 were all alpha or near alpha alloys. Ti BETA 21S was the only true beta alloy. Raw sheet material of different thicknesses were obtained for surface roughness measurements as well as for some EDS analysis. The material properties of the alloys used are shown in Table 4-2 and the list of raw materials are shown in Table 4-3.

Table 4-1 – Base alloys compositions

Alloy	Al	V	Fe	Mo	Sn	Zr	Si	Nb	O	C	Ti
Ti54M	5	4	0.4	0.6							90
ATI425	4	2.5	1.5						0.25		91.75
Ti64	6	4									90
Ti1100	6			0.4	2.7	4	0.45				86.45
Ti834	5.8			0.5	4	3.5	0.35	0.7		0.06	85.09
Ti6242	6			2	2	4	0.08				85.92
Ti21S	3			15			0.2	3			78.8

Table 4-2 – Published material properties for a selection of alloys

Base Alloy	Melting Point (C)	Vickers Hardness	Ultimate Strength (MPa)	Yield Strength 0.2% (MPa)	Elongation (%)	Elastic Modulus (GPa)	Poisson's Ratio
Beta-21S			915	880	15	83	
Ti54M			972	889	16.5		
Ti64SG	1660	349	950	880	14	113.8	0.342
Ti6242	1705	333	1010	990	3	120	0.32
Ti1100	1637		1131	1069	12	107	
Ti834			1050	930	11	120	
ATI 425	1600		1100	1050	14	143	

Table 4-3 - 2D surface roughness samples

Surface Roughness Sheets	Heat lot	Thickness	Thickness
ATI425	HT10RHWC	0.040"	1.016mm
Ti54MX	HT 410225	0.040"	1.016mm
RTI64FG	HT8F36895-03-00	0.040"	1.016mm
Ti54M	HT15976	0.032"	0.813mm
Ti54M	HT15976	0.025"	0.635mm
Ti6242	HT21953	0.040"	1.016mm
Ti6242	HT22305	0.032"	0.813mm
Ti6242	HT19650	0.025"	0.635mm

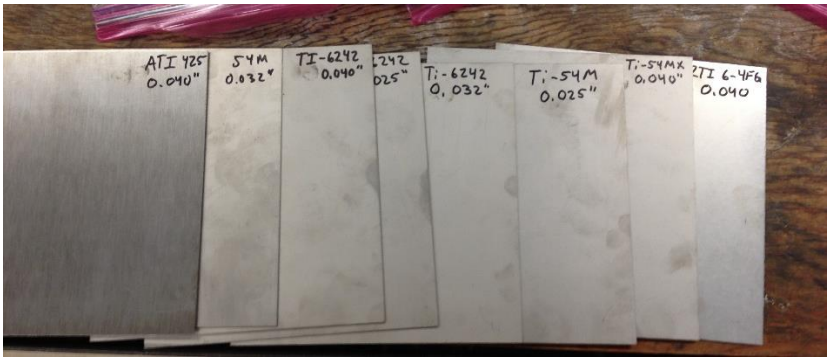


Figure 4.1 - Raw material for surface analysis

After concerns of contamination, raw material was obtained as a triangular snipping of Ti54m from a sheet that was ready to be used as a 4-sheet test panel. The sheet had been out of the chemical etch for six days and had not been cleaned but had been stacked with other sheets separated by paper. There were significant residual oils so the sample was cleaned with acetone and isopropyl alcohol then brought directly to the SEM.

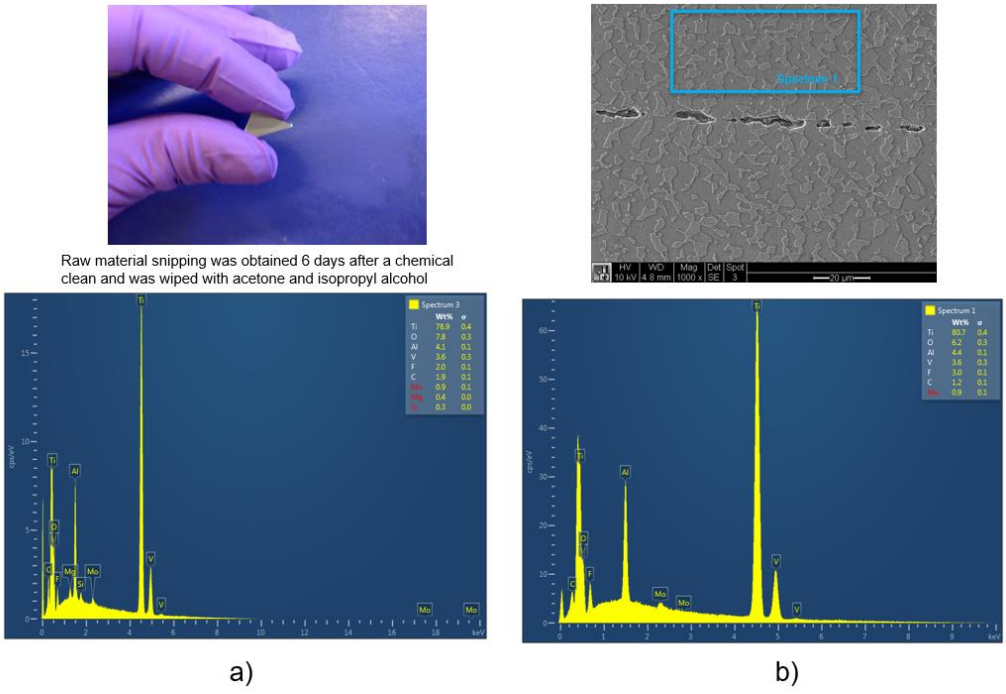


Figure 4.2 - EDS of raw material (a) and post etched material (b) (Ti54m)

Since the differences in surface profiles determine the initial voids it is an important characterization method to determine the initial bonding when combining two different materials. Therefore the surface roughness was characterized for the different raw materials in multiple ways. Each raw material sample was characterized with a Mahr Surface Profilometer system with a 2.4µm spherical diamond stylus. Typical surface roughness profiles recorded for Ti54M sheet material is shown in Figure 4.3 and the EDS profile compositions are shown in Figure 4.2.

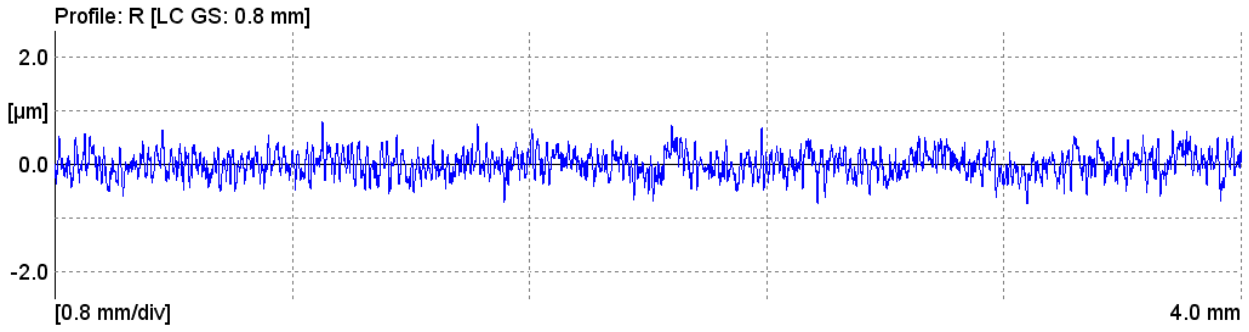


Figure 4.3 – 2D surface roughness profile example – Ti-6Al-4V 0.040" taken in the longitudinal direction

4.3 Diffusion Bonding Process

The Boeing Company prepared all diffusion bonding coupons. Coupons were produced by taking two sheets of titanium that have been chemically milled to remove the oxidation layer with a combination of hydrofluoric acid, nitric acid, and sulfuric acid. The treated sheets were then resistance welded on a custom designed roll seam welder. The weld pattern was performed in such a way that the edges were sealed off with the exception of a single opening. Plasma welding was used to attach a gas vent on that opening. This was to draw a vacuum so that diffusion bonding could occur without contamination from the atmosphere. Once a vacuum was completely achieved and while it was still applied, roll seam resistance welding across the entire the coupon was

performed to hold the vacuum within the coupon. Figure 4.4 shows the typical specimen. The vacuum tube was cut and the specimens were arranged into sets of twelve. The pressure necessary for diffusion bonding was achieved by two sheets of plasma welded Ti-6Al-4V with a gas pressure tube attached. The Ti-6Al-4V expands superplastically, conforming to the geometry of the coupons, and provides isostatic pressure to them. The coupons were put into a heated press for the desired temperature and pressure to create a diffusion bonded specimen. The coupons were machined into both a microstructural and a 3-point bending specimen using a waterjet cutting machine.

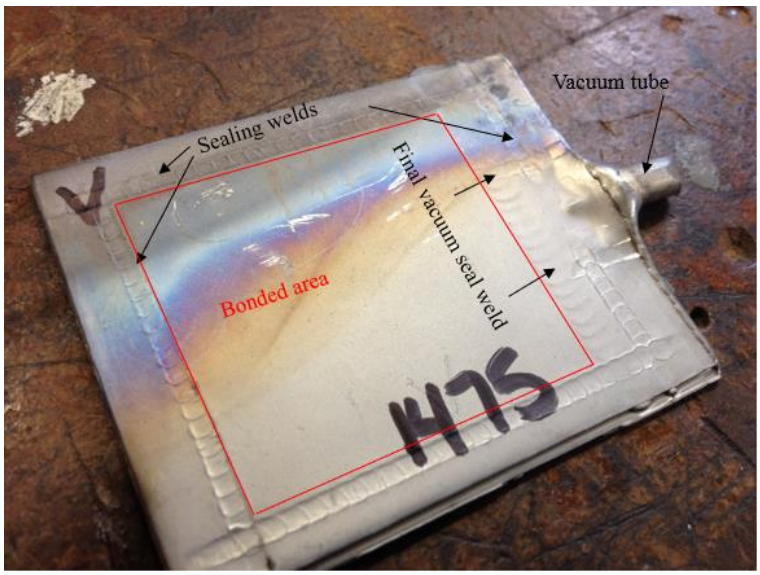


Figure 4.4 – Standard diffusion bonding coupon

The first set of materials used consisted of three fine grained titanium alloys: RTI Ti-6Al-4V fine grain, VSMPO Ti-6Al-4V fine grain, and TIMET 54M. Coupons were produced for each alloy bonded with itself as well as the other alloys for three different temperatures. The set was duplicated with a different preparation step of sanding with P120 grit sandpaper before chemical milling. This sanding step is part of the Rockwell protocol on diffusion bonding from the 1970s.

The result was a total of 36 coupons which were then delivered to University of Washington. The next set of 45 coupons delivered were made from six different titanium alloys: Beta-21SX, Ti-54MX, Ti-6242SX, Ti-1100, Ti-834, and VSMPO Ti-6Al-4V fine grain. These six materials were combined at the temperatures of 1425F, 1475F, and 1525F for most samples and 1650F, 1700F, and 1750F for the rest with a variance of +-5F. The reason for the difference between temperature sets was based on the alloys used. Beta-21SX, Ti-6242, and Ti-834 all bond at a higher temperature than the other alloys and any bonding including only these alloys was performed at the elevated temperature set. Due to the limited amount of material available to Boeing at the time, the second set of bonding coupons were smaller in dimensions. All samples were bonded at a pressure of 300 psi with a linear ramp up of 22 minutes and a holding time of 3 hours. All the processing conditions and material combinations are located in Table 4-5 after the description of the analysis.

4.4 Dissimilar and Similar Titanium Alloys Diffusion Bonding Quality Analysis

The results were assessed for their bonding quality from the data collected for each specimen. The goal during the experiment design was to determine how well bonding occurred and obtain a spread of void closure over the desired temperature range. The results ranged from significant voids that delaminated even before structural testing occurred with visible voids to perfect bonding where no voids were visible under high magnification and the specimen behaved as a single material. This was as expected because of the wide range in temperature and material combinations. However, because of the untested nature of some of the alloys and the new, there were some surprises and some of the materials performed significantly worse than expected.

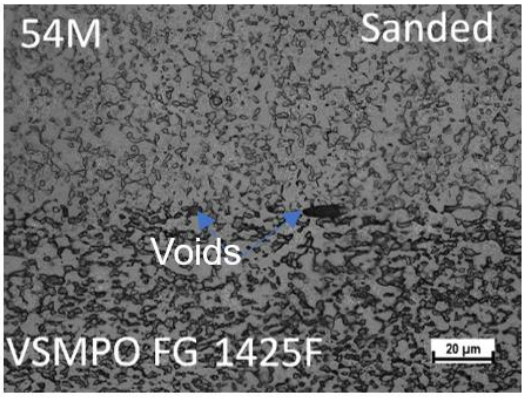


Figure 4.5 – Example microstructural test results

The first set of coupons consisted of the well-established materials of Ti-54M and two fine grain Ti-64 materials. At the lower temperature of 1425F, some significant voids were present in most of the samples. Of the twelve samples in that temperature group, only five showed bonding without voids. The rest of the samples had voids but in some cases not to the extent that would justify a drastic increase in processing temperature. For the 1475F group of samples, the bonding was vastly improved over the 1425F group. Only two of twelve samples had voids present in this group. For the 1525F group, only one sample had voids present. This pattern of low temperature with the most voids and less voids as temperatures get higher was expected and part of the test setup since it was desired to get a spread over the optimum temperature. The pattern can be seen in Figure 4.6 where the increase in temperature decreases the voids for the same material. This was represented in the structural data in the form of premature cracking. Both the 1425F and 1475F groups had seven samples that cracked prematurely though the 1475F group some less ductile cracks. The most prominent structural observation is the fact that the 1525F group had significantly decreased ductility when compared with the other groups. This only occurred in a couple samples in the second set of 45 coupons and was most probably the leftover oxide layer on

the outside of the specimens creating a sharp crack that then broke the specimen prematurely. The effect of sanding was determined to be negligible and therefore was not included in future tests.

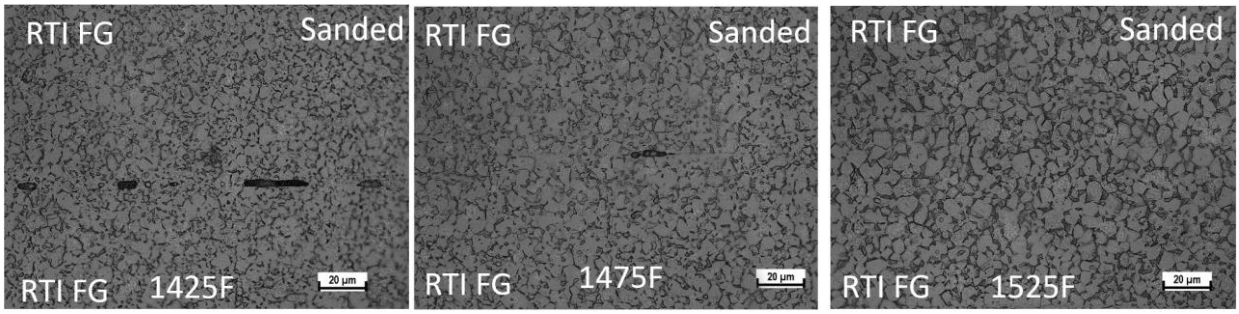


Figure 4.6 – Trend in bonding performance with respect to temperature in RTI 64 FG sanded.

The second set of coupons had a much larger variance in bond quality. This is because many of these alloys are newly created and their bonding performance is unknown. The ones that are not new are also uncommon diffusion bonding alloys. Of the materials used, only VSMPO fine grain Ti-64 is a well-established diffusion bonding material. In many cases the samples had significantly more voids than expected. The lowest performer was the samples of Beta-21SX bonded to itself. These had extremely poor bonding to the point where the voids were visible without magnification. All sample micrographs including the Beta-21SX micrographs can be seen in the Appendix B. Ti834 samples had voids visible at all temperatures for nearly all samples. However, it did bond well at high temperatures to the low temperature alpha/beta alloys (VSMPO 64 fine grain and Ti54MX). This was a general trend among the high temperature forming alloys. They benefited by combining them with the lower temperature alloys. This is fortunate due to the high temp alloys bonded with the low temp alpha/beta alloys being desired by Boeing. Ti1100 had at least some voids at all temperature and material combinations with the exception of when combined with Ti6242SX, which were void free at 1700F and 1750F. Ti1100 bonded best with Ti54MX at 1750F of the lower temperature alloys. Ti6242SX outperformed the other alpha alloys and bonded well

at higher temperatures both with itself and with Ti54MX. Structural behavior by temperature was as expected. Low temperature bonding samples had significantly more premature fractures and delaminations than the middle temperature which had more than the high temperature samples. In only one case there was significant microstructural change at the bonding interface. Ti6242SX-Ti6242SX at 1700F formed an extensive alpha beta mix about 50um in thickness along the bondline. Overall, the quality of bonding for the second set of coupons was lower than the first set. A summary of the results of both testing groups are in Table 4-5. The samples in the table are labeled with the extent of voids. This ranges from no voids to few voids to moderate voids to extensive voids and is explained in Table 4-4. Additionally these are labeled for whether or not the sample delaminated during structural testing.

Table 4-4 - Bonding quality explanation

Bonding	**Approximate Bonded %**
Few Voids	90-100%
Moderate Voids	70-90%
Extensive Voids	<70%
No Voids	0%

Table 4-5 - Sample list and results

Material 1	Material 2	Pressure	Time	Temp	Ultrasonic NDT	Bonding
RTI FG	Ti-54M	300psi	22min ramp 3 hour hold	1425F	No voids detected	No voids
				1475F	No voids detected	No voids
				1525F	No voids detected	Moderate voids
VSMPO FG	RTI FG	300psi	22min ramp 3 hour hold	1425F	No voids detected	Few voids
				1475F	No voids detected	No voids
				1525F	No voids detected	No voids
VSMPO FG	VSMPO FG	300psi	22min ramp 3 hour hold	1425F	Not tested	Moderate voids
				1475F	Not tested	No voids
				1525F	Not tested	Few voids
VSMPO FG - Sanded	VSMPO FG - Sanded	300psi	22min ramp 3 hour hold	1425F	Not tested	Moderate voids
				1475F	Not tested	No voids
				1525F	Not tested	Few voids
54M - Sanded	54M - Sanded	300psi	22min ramp 3 hour hold	1425F	Not tested	No voids
				1475F	Not tested	No voids
				1525F	Not tested	Moderate voids
VSMPO FG - Sanded	54M - Sanded	300psi	22min ramp 3 hour hold	1425F	Not tested	Few voids
				1475F	Not tested	No voids
				1525F	Not tested	Few voids
RTI FG - Sanded	RTI FG - Sanded	300psi	22min ramp 3 hour hold	1425F	Not tested	Moderate voids
				1475F	Not tested	Few voids
				1525F	Not tested	Moderate voids
54M	VSMPO FG	300psi	22min ramp 3 hour hold	1425F	No voids detected	No voids
				1475F	No voids detected	No voids
				1525F	No voids detected	No voids
RTI FG - Sanded	VSMPO FG - Sanded	300psi		1425F	Not tested	Extensive voids

			22min ramp 3 hour hold	1475F	Not tested	Few voids
				1525F	Not tested	No voids
RTI FG	RTI FG	300psi	22min ramp 3 hour hold	1425F	Not tested	Moderate voids
				1475F	Not tested	No voids
				1525F	Not tested	No voids
RTI FG - Sanded	54M - Sanded	300psi	22min ramp 3 hour hold	1425F	Not tested	No voids
				1475F	Not tested	No voids
				1525F	Not tested	No voids
54M	54M	300psi	22min ramp 3 hour hold	1425F	No voids detected	Few voids
				1475F	No voids detected	No voids
				1525F	No voids detected	No voids
Ti-834	Ti-834	300psi	22min ramp 3 hour hold	1650F	No voids detected	Extensive voids and delamination
				1700F	No voids detected	Extensive voids and delamination
				1750F	No voids detected	Moderate voids
Ti-834	Beta 21SX	300psi	22min ramp 3 hour hold	1650F	Inconclusive	Extensive voids and delamination
				1700F	Inconclusive	Extensive voids
				1750F	Inconclusive	Extensive voids and delamination
Ti-834	Ti-54MX	300psi	22min ramp 3 hour hold	1425F	No voids detected	Extensive voids and delamination
				1475F	No voids detected	Extensive voids and delamination
				1525F	Not tested	Few voids
Ti-834	VSMPO	300psi	22min ramp 3 hour hold	1425F	No voids detected	Extensive voids and delamination
				1475F	No voids detected	Extensive voids and delamination
				1525F	No voids detected	Few voids
Ti-1100	Ti-1100	300psi	22min ramp 3 hour hold	1650F	No voids detected	Extensive voids and delamination
				1700F	No voids detected	Extensive voids
				1750F	No voids detected	Moderate voids
Ti-1100	Ti-54MX	300psi	22min ramp 3 hour hold	1425F	No voids detected	Extensive voids and delamination
				1475F	No voids detected	Extensive voids and delamination
				1525F	No voids detected	Few voids

Ti-1100	VSMPO	300psi	22min ramp 3 hour hold	1425F	No voids detected	Extensive voids and delamination
				1475F	No voids detected	Extensive voids
				1525F	No voids detected	Moderate voids
Ti-6242SX	Ti-1100	300psi	22min ramp 3 hour hold	1650F	No voids detected	Extensive voids and delamination
				1700F	No voids detected	No voids
				1750F	No voids detected	No voids
Ti-6242SX	Ti-6242SX	300psi	22min ramp 3 hour hold	1650F	No voids detected	Extensive voids
				1700F	No voids detected	Moderate voids
				1750F	No voids detected	No voids
Ti-6242SX	Ti-54MX	300psi	22min ramp 3 hour hold	1425F	No voids detected	Extensive voids and delamination
				1475F	No voids detected	Moderate voids
				1525F	No voids detected	No voids
VSMPO	Ti-6242SX	300psi	22min ramp 3 hour hold	1425F	No voids detected	Extensive voids and delamination
				1475F	No voids detected	Extensive voids
				1525F	No voids detected	Extensive voids
Beta 21SX	Beta 21SX	300psi	22min ramp 3 hour hold	1650F	Inconclusive	Extensive voids and delamination
				1700F	Inconclusive	Extensive voids
				1750F	Inconclusive	Extensive voids
Beta 21SX	VSMPO	300psi	22min ramp 3 hour hold	1425F	No voids detected	Few voids
				1475F	Inconclusive	Extensive voids
				1525F	No voids detected	Extensive voids
Ti-54MX	Ti-54MX	300psi	22min ramp 3 hour hold	1425F	No voids detected	Moderate voids
				1475F	No voids detected	Few voids
				1525F	No voids detected	No voids
Ti-54MX	VSMPO	300psi	22min ramp 3 hour hold	1425F	No voids detected	Moderate voids and delamination
				1475F	No voids detected	Few voids
				1525F	No voids detected	No voids

Along with the temperature, pressure, and time the material has a significant effect on the efficiency of the bonding process. This can be seen by analyzing the performance of the different

bonding specimens. Dissimilarity also has an effect on the bonding. To analyze these effects the table of void performance was reduced into several charts.

4.4.1 Spectroscopic Analysis of Dissimilar Diffusion Bonding

Electron dispersive xray spectroscopy (EDS) analysis was used to determine the extent of migration of elements between dissimilar titanium alloys when they were bonded. The EDS line scans were taken at 10kV perpendicular to the bondline and located on a bonded area as shown in Figure 4.7 left. There is commonly a transition zone where the elements migrate and a different alloy is created. This diffusion area can be seen on the line scan plots as a relative increase or decrease in the response from an element as shown in Figure 4.7 right (between 18 and 22µm). Table 4-6 lists all the EDS samples taken while Figure 4.7 shows an example of the data obtained. All the line scans are in Appendix D.

Table 4-6 - List of EDS samples. Of the TIM3 official run set the samples highlighted in blue were analyzed with EDS.

Sample	Temp (F)	Material 1	Material 2	Sample	Temp (F)	Material 1	Material 2	Sample	Temp (F)	Material 1	Material 2
1	1650	Ti-834	Ti-834	16	1700	Ti-834	Ti-834	31	1750	Ti-834	Ti-834
2	1650	Ti-834	Beta-21SX	17	1700	Ti-834	Beta-21SX	32	1750	Beta-21SX	Ti-834
3	1425	Ti-834	Ti-54MX	18	1475	Ti-834	Ti-54MX	33	1525	Ti-834	Ti-54MX
4	1425	Ti-834	VSMPO	19	1475	Ti-834	VSMPO	34	1525	Ti-834	VSMPO
5	1650	Ti-1100	Ti-1100	20	1700	Ti-1100	Ti-1100	35	1750	Ti-1100	Ti-1100
6	1425	Ti-1100	Ti-54MX	21	1475	Ti-1100	Ti-54MX	36	1525	Ti-1100	Ti-54MX
7	1425	Ti-1100	VSMPO	22	1475	Ti-1100	VSMPO	37	1525	Ti-1100	VSMPO
8	1650	Ti-6242SX	Ti-1100	23	1700	Ti-6242SX	Ti-1100	38	1750	Ti-6242SX	Ti-1100
9	1650	Ti-6242SX	Ti-6242SX	24	1700	Ti-6242SX	Ti-6242SX	39	1750	Ti-6242SX	Ti-6242SX
10	1425	Ti-6242SX	Ti-54MX	25	1475	Ti-6242SX	Ti-54MX	40	1525	Ti-6242SX	Ti-54MX
11	1425	VSMPO	Ti-6242SX	26	1475	Ti-6242SX	VSMPO	41	1525	Ti-6242SX	VSMPO
12	1650	Beta-21SX	Beta-21SX	27	1700	Beta-21SX	Beta-21SX	42	1750	Beta-21SX	Beta-21SX
13	1425	Beta-21SX	VSMPO	28	1475	Beta-21SX	VSMPO	43	1525	Beta-21SX	VSMPO
14	1425	Ti-54MX	Ti-54MX	29	1475	Ti-54MX	Ti-54MX	44	1525	Ti-54MX	Ti-54MX
15	1425	Ti-54MX	VSMPO	30	1475	VSMPO	Ti-54MX	45	1525	Ti-54MX	VSMPO

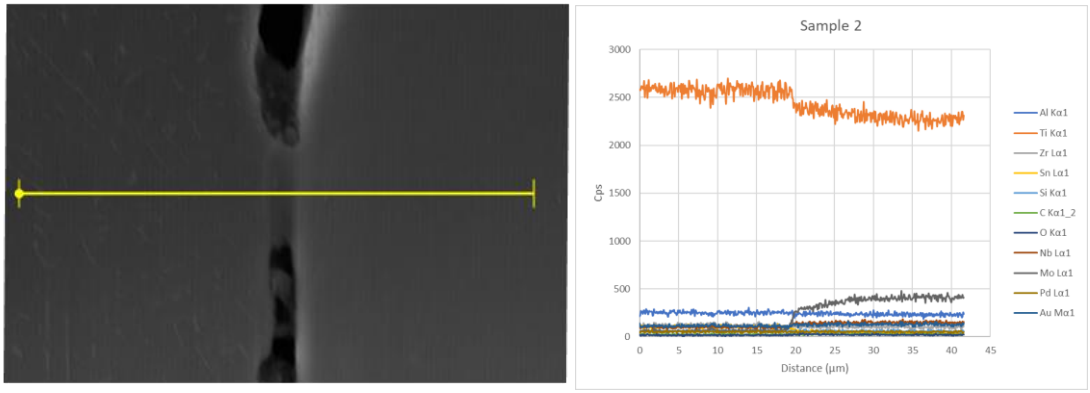


Figure 4.7 - Sample 2 – Ti-834 to Beta-21SX 1650F linescan location and response

4.5 Structural and Non-Destructive Analysis

4.5.1 Bending Test Setup and Modeling

For the two sheet dissimilar and similar diffusion bonding coupons, the first round of 36 structural samples were deformed in an Instron tensile testing machine using a three point bending fixture. The fixture consists of three Inconel dowels with a support span of 32mm. The first set of testing followed the ASTM D790 standard. Each of these specimens were deformed until breaking or until an extension of 22mm. During the test it was found that the span was too large in some cases for such a ductile material. Therefore, three point bending of the next 45 samples were conducted following the ASTM E290 ductility standard (as shown in Figure 4.8). This standard is a forming test and meant to determine the extent of ductility capable by a material. It provides a better layout for bending ductile materials which deform far past the standard three point bending test described in ASTM D790. These three point bending tests were also conducted with an Instron machine with a change in the fixture, the difference between the two standards being a reduced span (parameter "C" in Figure 4.8). Most of the samples were tested at a rate of extension of 2mm/minute with a 10kN load cell. The purpose of this test is to give a qualitative assessment of

diffusion bond quality by loading the bondline with in-plane shear. Due to the bending load, the top half of the specimen experiences a compression load while the bottom half of the specimen experiences tension. The difference induces a shear stress that is maximized at the neutral axis of the specimen. This is also location of the bondline if the sheet thicknesses are the same. If the bonding is poor, then then the specimen will split along the bondline because of the shear stress. This test was done because the microstructure only provides bonding information on one plane at a very small location. There may be some variances in bonding quality. Therefore doing a qualitative structural test over the whole bondline is advantageous because it provides information over a much larger area and is more representative of microscopic performance.

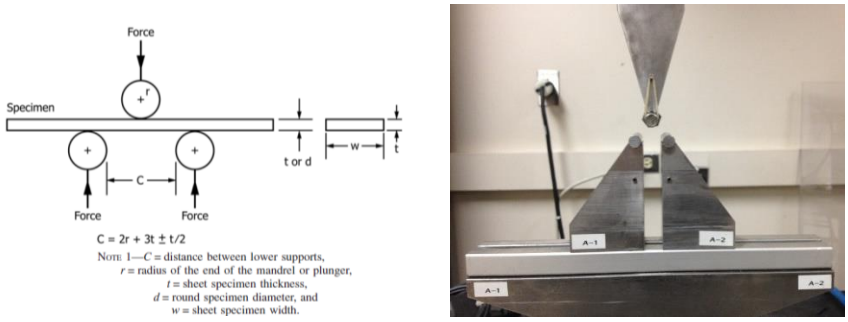


Figure 4.8 - E290 Testing Layout (left) and actual testing setup (right)

The maximum flexural stress in the beam is $\sigma_f = \frac{3PL}{2bd^2}$ where P is the force applied, L is the support span, b (or w) is the specimen width, and d (or t) is the specimen depth. This is the stress that would occur during low deformation of an isotropic material and refers to the stress at the bottom-center of the beam. The modulus of elasticity is calculated by $E = \frac{L^3 m}{4bd^3}$ where m is the slope of the tangent to the initial straight line section of the stress strain curve. These calculations along with the linear displacement (extension) from the Instron machine gives flexural data (shown

in Figure 4.9). The results of these tests were used to get a qualitative understanding of the overall bonding.

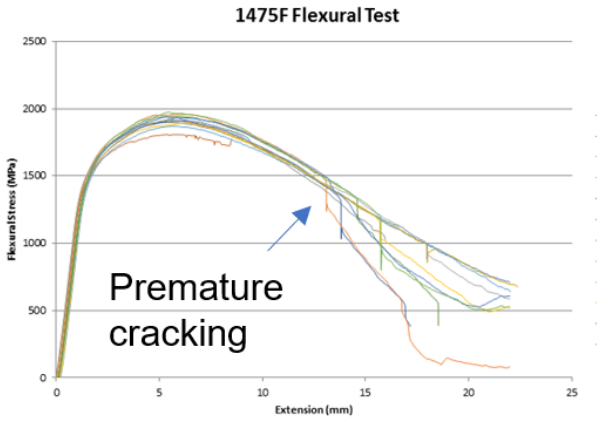


Figure 4.9 – Example structural test results from the group of 1475F samples in 3 point bend testing that shows the rough behavioral trend of titanium and what is meant by premature cracking (left).

Due to the high levels of plasticity during the bending test, the Euler beam bending derived equation for the flexural stress equation stated above is no longer accurate. Nor would the values of in-plane shear stresses be accurate when basic beam theory is applied. The best solution for this is to use finite element simulations to determine the actual stresses experienced during bending and calculate the maximum stress that the bondline was subjected to. This can be used as a lower bound for the stresses that the bondline is capable of handling. An ABAQUS explicit simulation was developed to test this theory. It uses an elastic-perfectly plastic material model with independent values for the two sheets and cohesive zoning in between. It replicates the behavior seen in specimens that didn't fracture. This simulation can be seen in Figure 4.10.

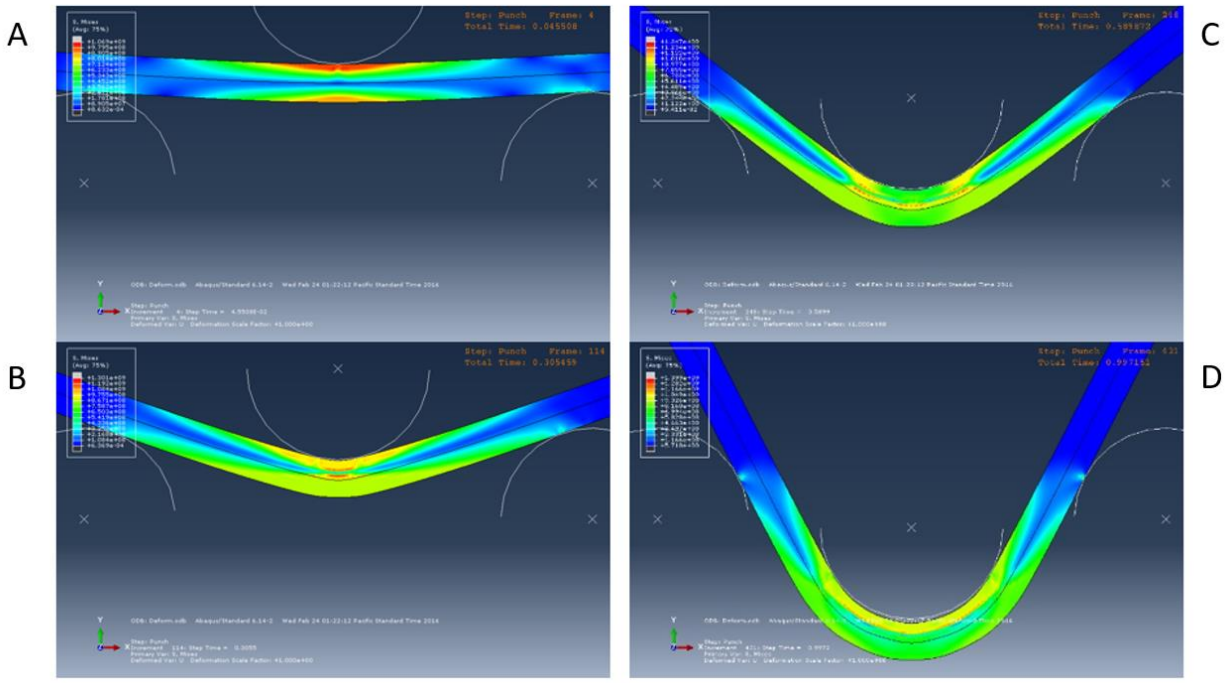


Figure 4.10 - FEM simulation of bending specimen. The simulation confirms the basic behavior of the specimen. Time progresses from figures A through D during the test. As the specimen deforms the contact points change which replicates experimental observations. Contact on the two outer supports migrates slightly while the plunging dowel changes contact points continuously. This can be seen at the end of the test in the bottom image.

The fracture behavior of the specimens were determined to be of four different types. The four types are pictured in Figure 4.11. Delamination, shown in Figure 4.11A, where the fracture occurs on the bondline and separates the two different sheets, occurred in the specimens with enough voids to shear completely through the specimen. Single sheet fracture with delamination, shown in Figure 4.11B, occurred in specimens where the bond was strong enough to allow for fracture at the specimen edge but not strong enough to allow for the crack to propagate through the sample. Some specimens completely fractured, shown in Figure 4.11C, with no evidence of any bondline and a crack propagating all the way through the material, and therefore had a good bond quality. IV) On the other hand some specimens didn't fracture at all, shown in Figure 4.11d, but simply deformed which also is evidence of good bond quality.

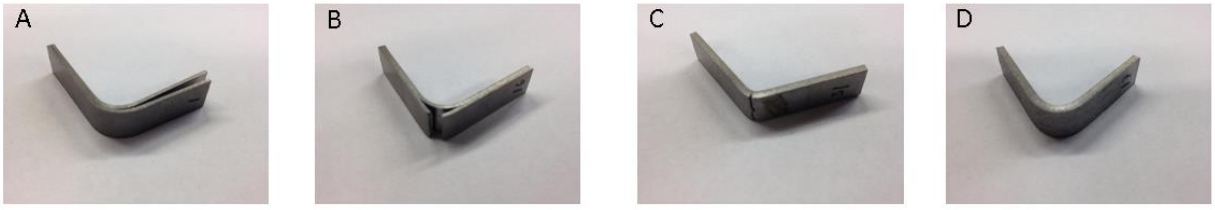


Figure 4.11 – The four main specimen types: A – delamination, B- single sheet fracture with delamination, C – complete fracture, and D – no fracture

The fracture behavior was further analyzed by looking at the different types through a microscope. The fracture initiation sites can be seen to come from the corner edge of the material at the center of the bending radius. This can be seen in the top left of Figure 4.12 and Figure 4.13. In the delaminations one can see fine vertical ridges in some of the specimens. It is unclear exactly what these are. They are too large to be the voids from diffusion bonding. Therefore they are most probably from the rolling process at the mill. Depending on the material, significant 45 degree shear zones were seen at the edges in the fracture surface (can be seen in Figure 4.14 and Figure 4.15). Though all materials have plane stress zones at the edges. Full delamination occurs only when diffusion bonding is very weak and propagates from the center out to a single edge (Figure 4.16).

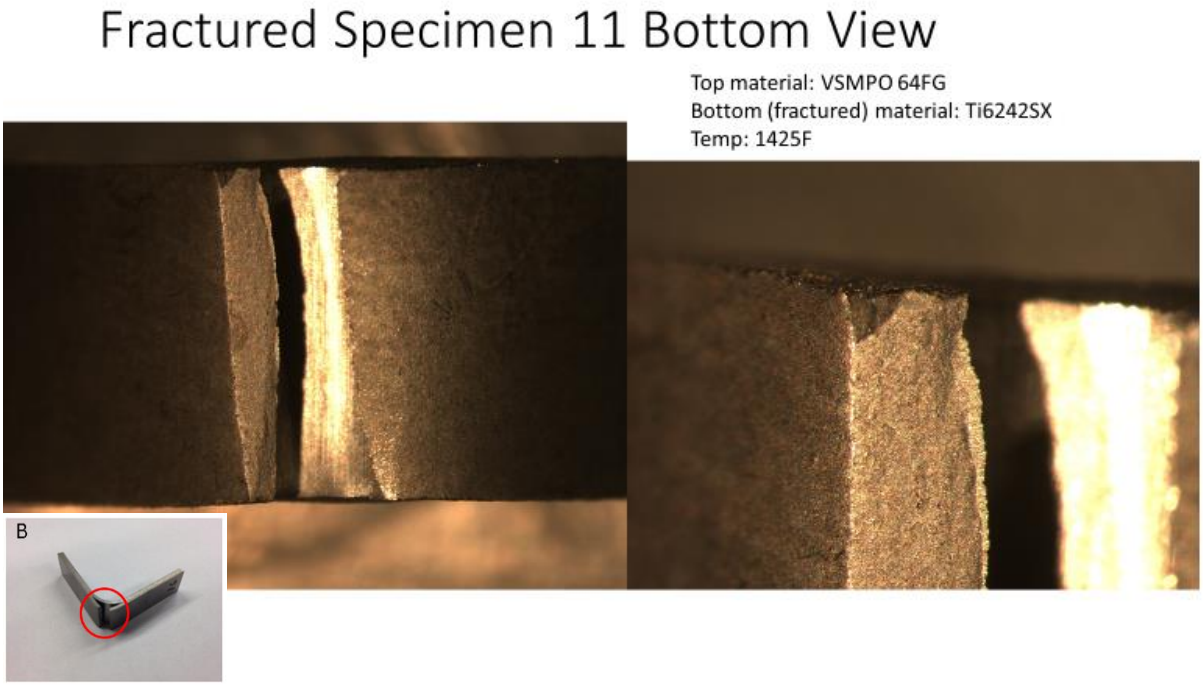


Figure 4.12 – Fracture behavior of a specimen with single sheet fracture and delamination. Fracture initiation occurs at the center of the fracture surface.

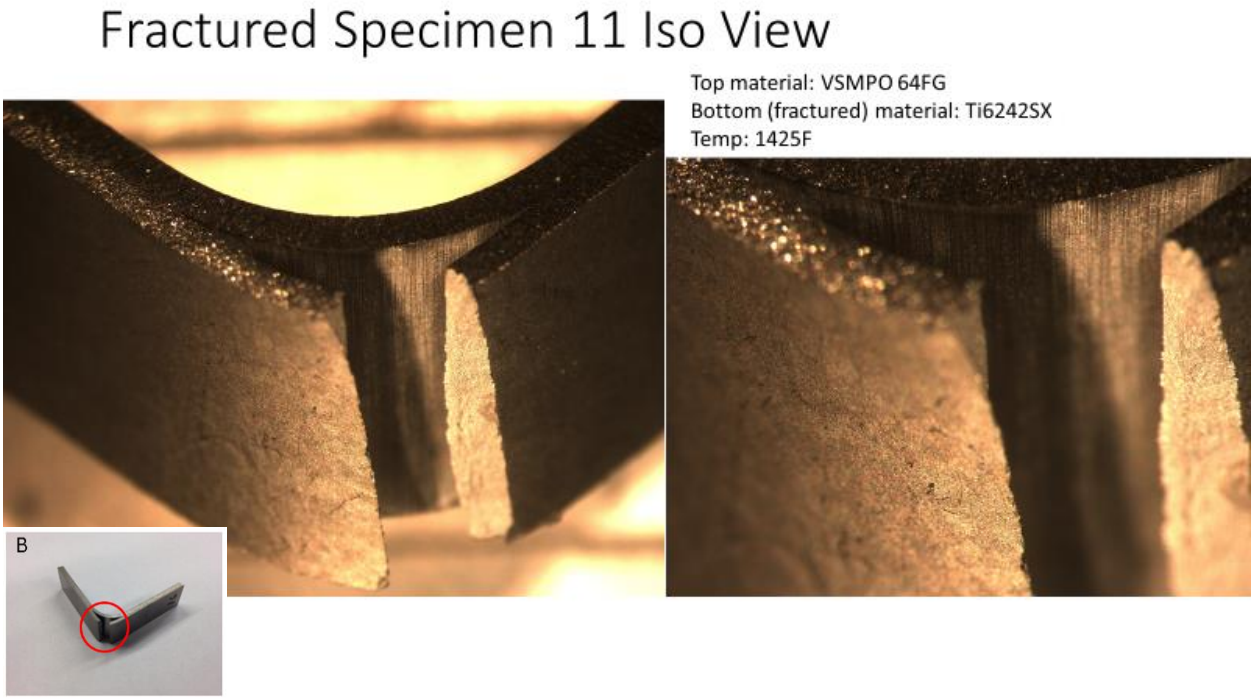


Figure 4.13 – Fracture behavior of the same specimen as Figure 4.12 but in isometric view.

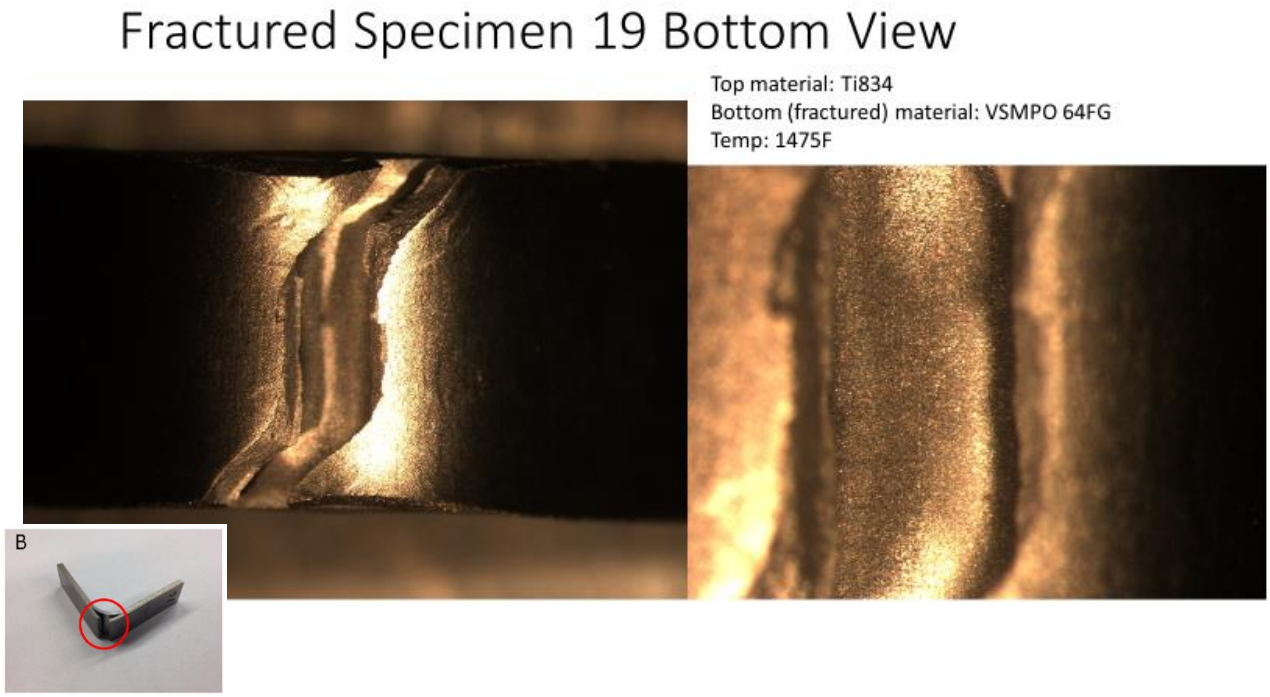


Figure 4.14 – Fracture specimen with single sheet fracture. This however has significant shear lip zones.

Figure 4.15 – This complete fracture specimen has cracking going all the way through the material and significant shear lips at the edges.

Figure 4.16 – Specimen with full delamination. This only occurs in specimens with very weak bonding.

4.5.2 Shear Testing

Structural testing has been explored in a variety of ways for diffusion bonding both with specifically created specimens as well as specimens obtained from the industrial process. Testing consisted of tensile or shear specimens. The specimens were created for lap shear, tensile, and peel testing.

Lap shear testing is a common way to perform structural testing on diffusion bonded specimens. It has been performed on both similar and dissimilar diffusion bonding with or without an interlayer for a variety of materials. Since ductile materials fail in shear, an ideal way to fracture the bondline is to use the stress concentrations of the voids to induce shear fracture. Diffusion bonding is primarily an aerospace technology and the lap shear test of diffusion bonded material

has been applied to the additional aerospace alloys of magnesium [64, 65], aluminum [66, 67], and multiple types of titanium [16]. Testing also included laminated layers of dissimilar materials such as magnesium to aluminum [68], titanium to steel with a copper interlayer [69, 70], titanium to aluminum [71], and metal matrix composites [72]. The structurally weak bond between these different alloys gives accurate results with the lap shear test. Structural testing on diffusion bonded specimens of aerospace titanium was first performed by Pilling et al [16]. In his paper, he determined the structural properties of thin titanium sheets using the lap shear method. The titanium was milled out on the top surface and milled out on an adjacent part of the bottom surface up to the center bondline with an overlap area between the two sheets which can be seen in Figure 4.17. This allowed the part to pulled in tension and produce a shear failure in specimen at the bondline only if an offset moment isn't induced. The tests were presented as either good bonding or poor bonding. There is no fractography of the specimens which could show the shear surface at the bondline.

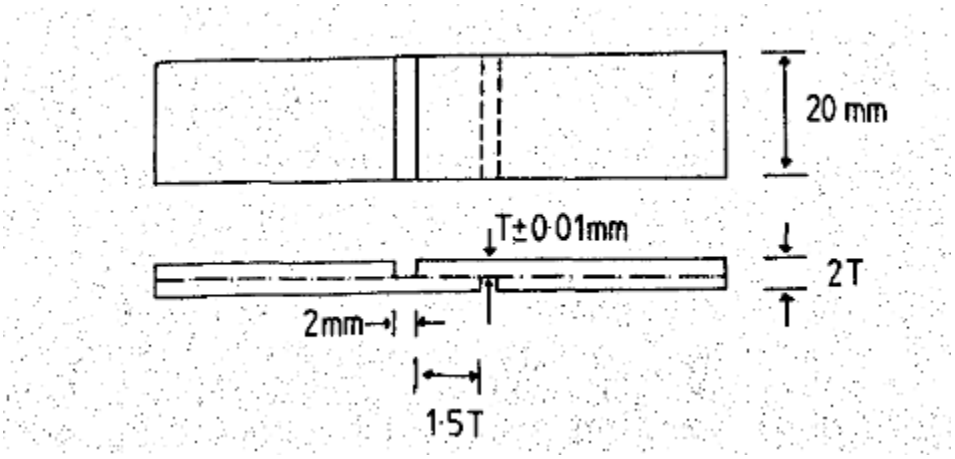


Figure 4.17 - Pilling lap shear [16]

Alternatives to the lap shear test are tensile tests and peel tests. Several research groups have explored these types of bonding tests. Tensile diffusion bonding specimens have been prepared by bonding the two halves of a tensile specimen. The two halves are machined then pressed together under heat to give a diffusion bond that is perpendicular to the loading direction. This form of bond is uncommon in application but provides a convenient way to determine the bonding load. Tensile tests have been used to test mainly titanium [73, 74] along with titanium aluminide [75]. Peel tests use the fact that many of the applications of diffusion bonding are with relatively thin sheets and therefore the peel strength would be close to a failure mode in application. Specimens are created with an unbonded section at one end that provides locations for tensile grips. The unbonded sections are bent 90 degrees and pulled in a load frame. The diffusion bond naturally creates a crack at the edge of the bonded section and when the sample is pulled the crack propagates. The peel test produces ductile tensile failure in the bonded regions and no effect in the unbonded regions. There are some authors that voice concerns with the ability of the peel test to produce accurate results due to the tests heavy reliance on test setup [73]. Diagrams of both tests can be seen in Figure 4.18.

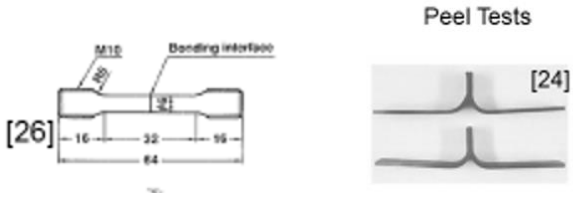


Figure 4.18 – Tensile specimen (left) and peel specimen (right)

The structural testing of diffusion bonding has had some significant difficulties. The fundamental problem of diffusion bonding structural testing is the fact that the bond is near base material properties. When done well, the diffusion bond will be effectively indistinguishable from the

surrounding material with the exception of a few small but very important voids. Due to the high plasticity involved in fracturing titanium, there can be distortion of the bondline. In high quality bonds, often the fracture will occur in the base materials [73]. A secondary problem is that diffusion bonding is a high temperature bond that is material dependent and usually created with a large section of the base material. Diffusion bonding cannot be tested like adhesives where the adhesive is laminated onto significantly stronger and stiffer material. Therefore, one is left with attempting to test a bond that can be nearly as strong as the base material and is surrounded by a large volume of base material on either side. This situation is easily tested under poor processing conditions or with dissimilar material bonds where the bond is significantly weaker than the surrounding material and the bond fails regardless of the loading situation. Stronger bonds are more difficult to test. Many tests have significant difficulties when it comes to near-solid bonding. There is a significant lack of fractographic results on near full property bonds to assist in developing a test.

Experimental Methods

This research has the goal of determining what parameters are necessary to structurally test for a thin sheet diffusion bonded specimen of variable geometry. This means obtaining a shear surface at the bondline, a target plane that is less than $5\mu m$ in thickness judging from the height of voids in titanium micrographs (see Figure 4.19). In a poor diffusion bond, this will occur over a wide range of specimen geometries. However, as the bond quality approaches a fully bonded specimen, the crack may not propagate in shear along the bondline resulting in a load that isn't representative of the shear surface. To do this a combination of finite element simulations and experiments were performed. The simulations explored the fracture behavior of the shear specimen by varying the overlap length, the milling depth, and the fillet radii. The experimental work was performed to

determine the effectiveness of the test. The theory behind these calculations and solid material experiments was that in order for the test to be valid for bonded samples, the specimen must fracture at full material strength at a specific location and in the correct fracture mode. If this is possible, and the specimen fails at the bondline whether it is there or not, then the test will valid whether there are a significant amount of voids or not. Therefore the simulations were performed without any degradation in the bondline area.

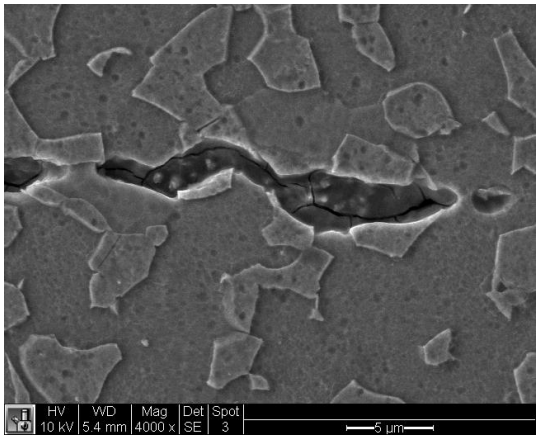


Figure 4.19 - A typical void showing how thin the bonding plane is

Simulation

The parametric finite element simulation was designed to determine the effects of different geometry on the macroscopic failure of the shear specimen. The specimen will be prepared by machining one side, flipping the sample, then machining the other side. There can therefore be some differences in the edge heights and overlap. The specimen can be surprisingly sensitive to offset moments. Therefore these effects are also included in the parametric study. The effect of the thickness on each of the bonded sheets is also determined. The optimum overlap area is arguably the most important parameter and was determined. If the overlap is too large then the fracture will

not occur anywhere near the bondline but will occur in the connecting beam area. The general layout for the simulations is presented in Figure 4.20. There is two large untouched areas on either side of the area of interest. There are two machined areas, one upper and one lower, that remove a single sheet worth of material and form the overlap. The other important parameters are the fillet at the overlap and the overlap itself. The simulation was performed in ABAQUS explicit. The material model used in this was the Johnson-Cook viscoplasticity method. The Johnson-Cook model parameters are for Ti-6Al-4V [76] and the model is run at a rate such that the strain rate dependence is low.

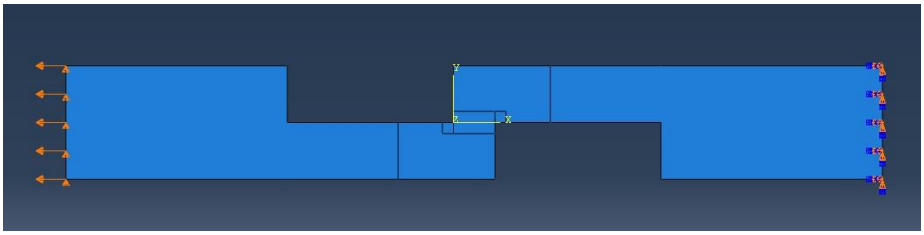


Figure 4.20 - Simulation layout

The parametric study tested the material in Table 4-7 and covers the primary geometric parameters of the overlap, thickness, over/under machined, and connecting beam length. Studying these parameters should give the main features of the specimen geometry so that a shear surface at the bondline can be obtained.

Experiments

The experimental work consisted of shear specimens of both solid material as well as diffusion bonded material and micrographic verification of the diffusion bonded results. The exact manufacturers of the materials and processing conditions are proprietary and unnecessary for this work since the results will be compared with the percentage bonding. Presented below in Table

4-7 is a list of the different specimens used along with the machining characteristics and overlap. Specimen fabrication was improved over time but some of the less accurate specimens are included in this document for the purpose of illustrating how important accurate specimen fabrication is. These are delineated by the labels machining types one and two for past and present machining. In machining type one, the specimens were waterjet cut out of solid material and then machined using a HAAS 3-axis CNC Mill with a two flute 1/8" end mill. Machining defects always occur especially when machining this small of a specimen. Here, over-machining is when too much material is removed and the corner stress concentration is past the bondline. Under-machining is when too little material is removed. Ideally machined is when the corner is within ±0.002" of the bondline vertically. The end mill used always produced a sufficiently sharp corner and the overlap is specified for each sample. Four solid samples of bulk Ti64 standard grain were used to test the procedure and some of the failure mechanisms calculated by FEM. Three diffusion bonded samples were used to test the effect of overlap, over-machining, and the machining to the ideal geometry. The shear strength of the samples that fracture at the bondline were compared with micrographs and theoretical values.

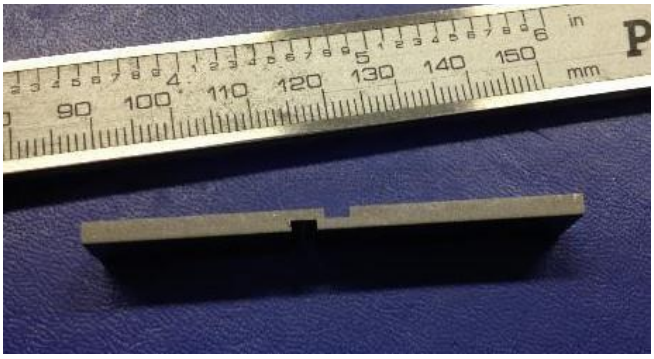


Figure 4.21 - Example machined specimen showing the upper and lower milled sections

The simulations provided insight into the machining characteristics needed to produce a fracture in pure shear along the bondline. The results of the simulation are categorized by their fracture behavior which is presented in Figure 4.22. They are described as fracturing along the bondline, the desired behavior, along the connecting beam, or in a mixed mode fracture of the connecting beam. The results of the different tests are summarized in Table 4-8. Failure always occurs from the beam side of the stress concentration at the corner fillet. The specimen will fail at the bondline if the overlap is small enough. By these calculations the largest overlap without fracturing the connecting beam is 75% of the beam thickness (for the diffusion bonding specimens this works out to a maximum overlap of $762\mu m$. This is only for similar beam thicknesses since as one beam gets smaller, even though one picks an overlap based on the smallest beam, the overlap will need to be reduced or it will fail along the small beam. The beam length has very little to do with the failure location. The corner fillet needs to be small but is still very achievable with conventional machining. The fillet required is less than R=0.025" with a beam width of 0.04". Over-machining and under-machining are a significant difficulty since in order to machine the specimen since it must be removed from the fixture, flipped, then remeasured and machined again to produce the second beam. The other consideration is that the exact bond location may deviate, especially in more complex samples, yet will still need to machined perfectly. Meaning if the sheets were not uniform and the bondline varies over distance this will have to be accounted for. With over-machining there is a thick bondline section and the stress concentrations are not at the bondline but are above and below it. Therefore over-machining allows for plasticity throughout the bondline area and gives a predominant shear fracture but one that is located at the stress concentration and not at the bondline. The over-machined case also gives a higher probability of connecting beam fracture. Under-machining also has stress concentrations that aren't located at the bondline but

since the beams are too thick the fracture is across the bondline at an angle giving both shear and tensile fracture and inaccurate results. The different fracture behaviors can be seen in Figure 4.22. Based on the machining of the specimen, one can get fracture that occurs unlocalized from the bondline.

Table 4-7 – Machining conditions for samples where T is the single sheet thickness

Material	Machining Defect	Overlap
Solid Ti64 SG	Over-Machined	0.75T
Solid Ti64 SG	Under-Machined	0.75T
Solid Ti64 SG	Over-Machined	0.7T
Solid Ti64 SG	Ideal	0.75T
Ti64 FG	Ideal	1.5T
Ti64 FG	Ideal	1T
Diffusion Bonded Ti64 FG	Ideal	0.75T
Diffusion Bonded Ti64 FG	Ideal	0.75T

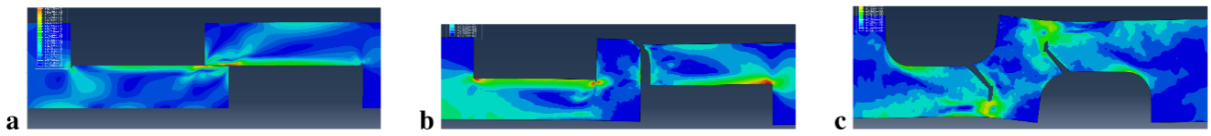


Figure 4.22 - The different fracture behavior observed in the simulations: (a) is the desired failure at the bondline, (b) is failure at the connecting beam due to too large of an overlap, and (c) is complex failure due to the large fillet

The experiments both confirmed some of the fracture behavior that was described in the simulations as well as providing insight into the accuracy of the shear test. The fracture behavior is demonstrated for the cases of overmachining, under-machining, accurate machining, and too large of an overlap. The simulation for the case of over-machining confirms the finite element simulation. The shear stresses are distributed over the high bond area. However, unlike the simulation, the bond does not fail in pure shear but demonstrates a moderate amount of tensile deformation. The behavior for the under-machining case is as described in the simulation. Some

small differences between the simulation and experiments arise from the plane stress condition at the edges as shown in Figure 4.23a. When the material fractures purely in shear, such as in Figure 4.23.b there is no discernable change in the surface due to plane stress. However, in the case when the overlap was too large, such as in Figure 4.23.a, the fracture includes shear lips at the edges where plane stress occurs.

Table 4-8 - Machining failure results

Test Type	Overlap	Other Dimension	Failure
Overlap	0.5T		Bondline
Overlap	0.75T		Bondline
Overlap	1T		Connection
Overlap	1.25T		Connection
Different Thickness	$0.75T_1$	$T_1 = 0.625T_2$	Connection
Different Thickness	$0.75T_1$	$T_1 = 0.8T_2$	Bondline
Corner Fillet	0.75T	$R = 0.0125$”	Bondline
Corner Fillet	0.75T	$R = 0.025$”	Bondline
Corner Fillet	0.75T	$R = 0.05$”	Complex
Corner Fillet	0.75T	$R = 0.075$”	Complex
Over-Machining	0.75T	$T_1 = T_2 = 0.95T$	Bondline area
Under-Machining	0.75T	$T_1 = T_2 = 1.05T$	Bondline area

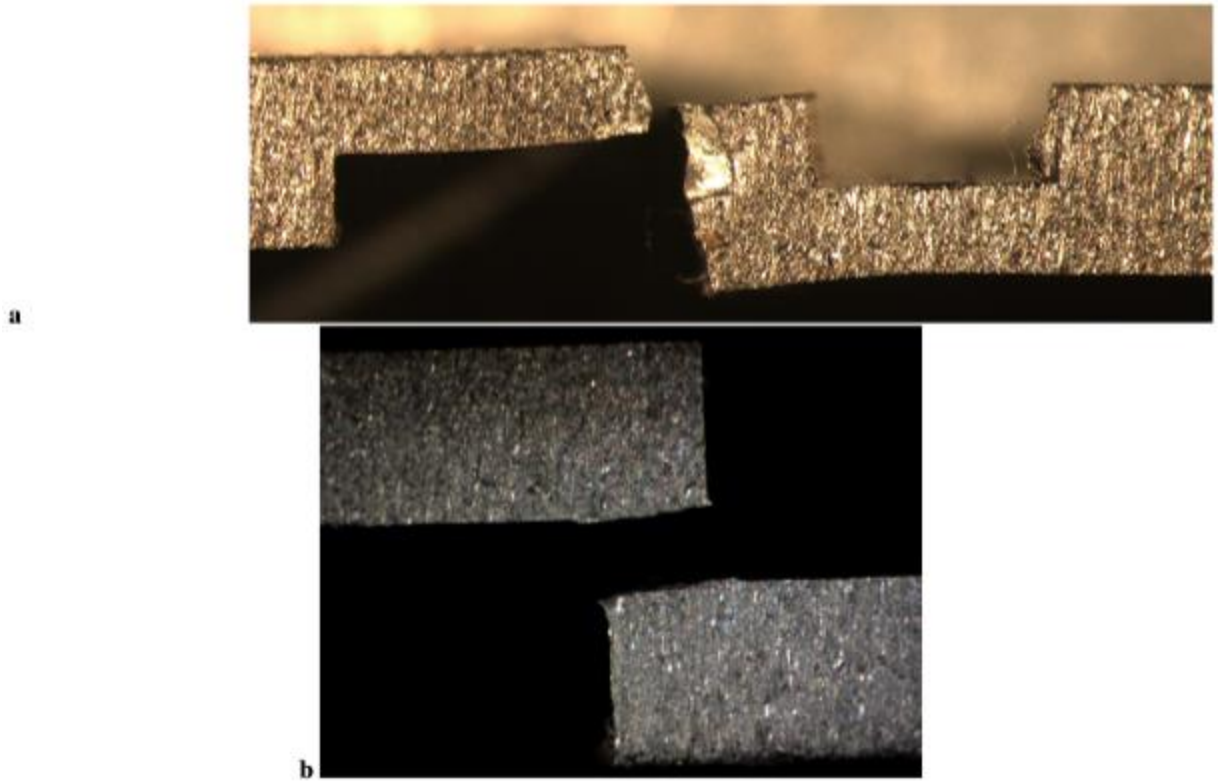


Figure 4.23 - Fracture behavior of different specimens: (a) The large overlap caused failure along the beam with a shear lip at the plane stress edge while (b) the appropriate overlap allows for predominantly shear failure

Micrographic data was taken from the two diffusion bonding specimens and analyzed for the percentage of bonded material. Diffusion bonding strength is thought to be proportional to the amount of bonded material. This therefore provides the basis for comparison between the relative shear strength of the test specimens and the information from the micrographs. The simulated bondline calculated determined that the first specimen had a bonded percentage of 72% and the second specimen had a bonded percentage of 64% as shown in Figure 4.24.

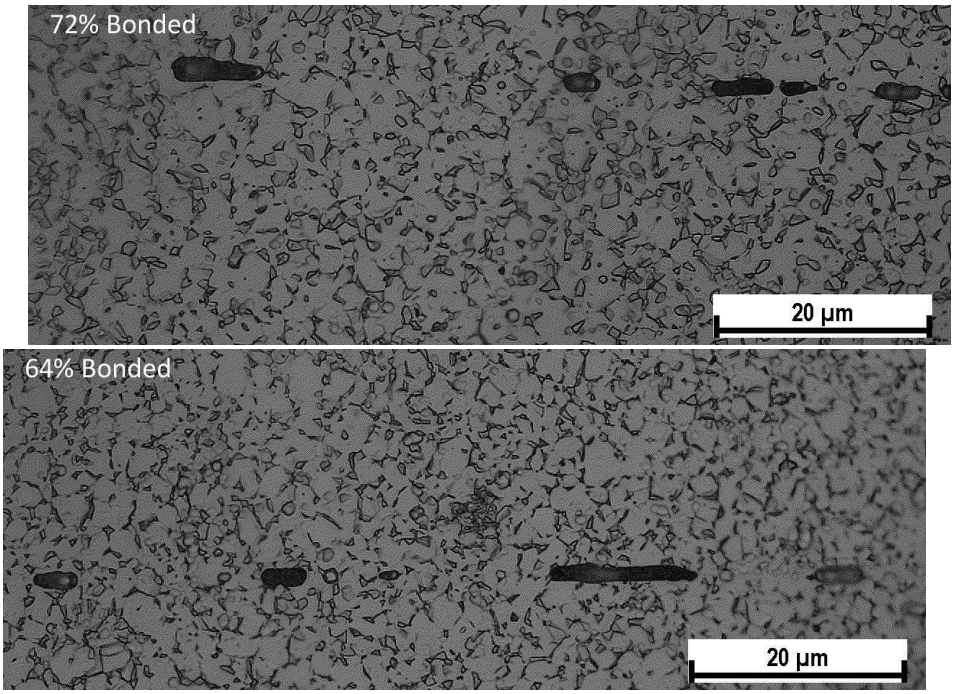


Figure 4.24 - Bonding behavior of the different samples: (top) First diffusion bonded sample of Ti64 FG with a bonded area of 72 % (bottom) Second sample with an area of 64%

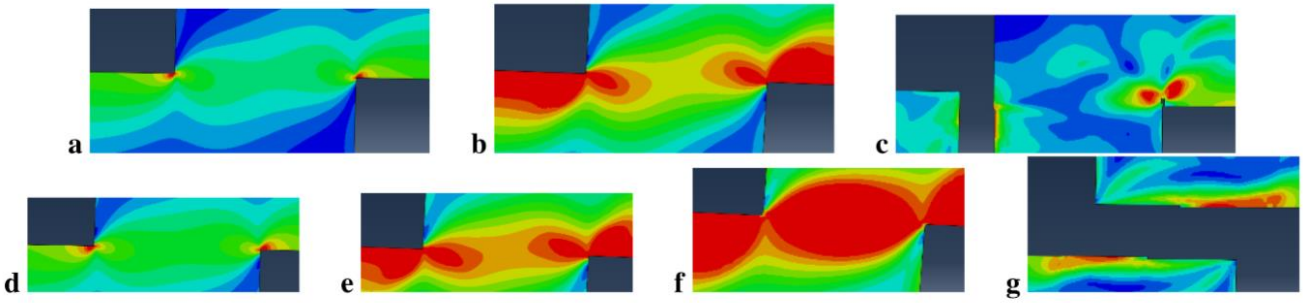


Figure 4.25 - The dominant von Mises stress area change with overlap progressing in time from left to right. (a-c) is the case of a large overlap and beam failure while (d-g) is the case of small overlap and bondline failure

Discussion

Difficulties in developing a structural test for high quality diffusion bonding in thin sheets arises as a result of the high quality of the bond, the strength of the base material, and the small size of the thin sheet titanium. However, with the results from the simulation, one can build an appropriate test to determine the diffusion bonding characteristics. The simulations provided not only a prediction of the geometry needed to create a shear fracture at the bondline but it also showed the reasons for these geometric constraints. The differences in geometry change the way the stress evolves throughout the test. For example, the stress evolution during the overlap test is different depending on the failure condition as shown in Figure 4.25. In both cases the stress initiates at the stress concentrations at the corners of the overlap (Figure 4.25.a and Figure 4.25.d). The stress builds and becomes a dominant bending stress at the interior side of each beam in both cases (Figure 4.25.b and Figure 4.25.e). In the connecting beam failure mode, the bending stress remains dominant at the beam until fracture occurs (Figure 4.25.c). In the bondline failure mode, the beams bend and the dominant stress eventually becomes pure shear at the bondline (Figure 4.25.f). Fracture therefore occurs at the bondline instead of in the connecting beam (Figure 4.25.g). This suggests that there is a significant limitation on the overlap length.

The validity of the test can be determined by examining the fractography of the shear surface. The difference between shear failure and tensile failure is easy to see on the fracture surface. If the machining is not done properly, the fracture surface will not be dominantly shear. The shear areas show up as ductile shear lips in the direction of loading. The tensile failure areas show up as darker and dimpled. If, like Figure 4.26, the specimen is under-machined then the shear must transition from shear failure to tensile failure then back to shear to accommodate the difference in vertical height. This can be seen with the typical ductile shear failure surface at both ends and dimpled

fracture in the middle instead of pure shearing. One can see the height difference across the overlap section in this figure by the loss of focus for the far shear surface.

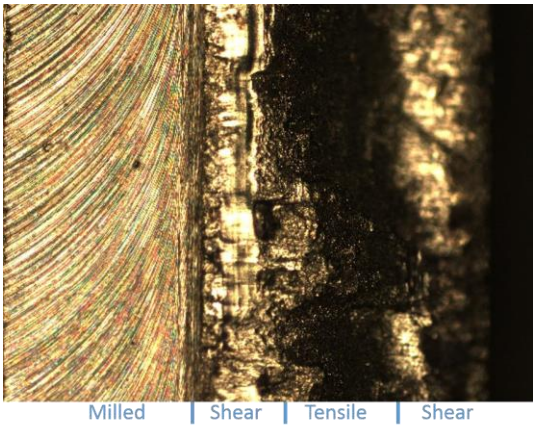


Figure 4.26 - An under-machined sample that shows the tensile failure in the center. The cracks initiate from the stress concentrations on either side of the bondline area and propagate in shear for a certain distance after which the specimen pulls apart in tension because the crack tips are not at the same vertical height in the specimen.

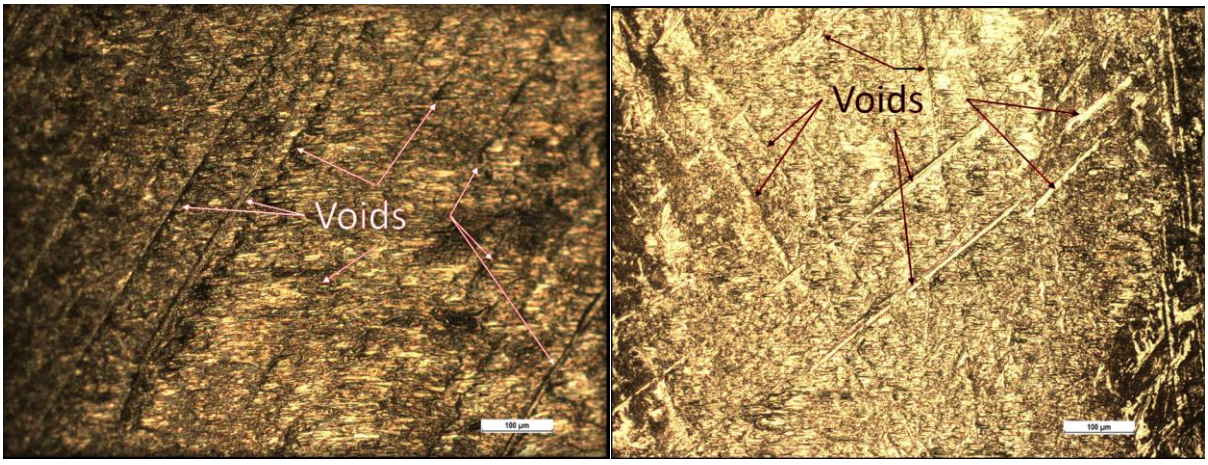


Figure 4.27 - Bonding behavior of the different samples: (left) First shear sample corresponding to the 72% bonded micrograph (right) Second sample corresponding to the 64% micrograph. The scale for both samples is 100 µm.

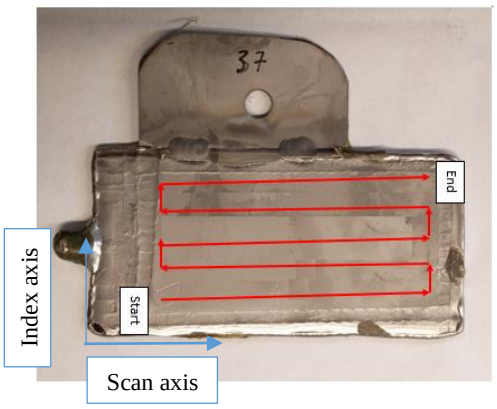
As a result of properly machined specimens, the diffusion bonded specimens fracture directly at the bondline. The dominant shear smearing of the fracture surfaces shows that the stresses read from the load frame correctly represent the shear capability of the bond. The fracture surface also shows the void locations. These can be seen in the form of elongated irregular lines that are the width of a common void but oriented in directions that aren't horizontal, which is the shear direction. Both these samples were sanded before diffusion bonding with 120 grit sandpaper. However, they were sanded in different ways with one sample using a belt sander while the other sample was sanded by hand. The voids in Figure 4.27 top show that the belt sanded specimen has voids that are elongated in a single direction oriented at about 45 degrees from the shear direction. This orientation is due to the fact that the sample was taken at about that angle. In the second specimen the voids are at random orientations and due to the random orbital sanding. While it would be difficult to quantify the bonded percentage from the fracture surface due to the significant shear smearing, it does allow one to validate that the shear test was performed appropriately and that the surface tested is the voided bondline.

Conclusions

A shear test was developed for diffusion bonded specimens that has the capability of testing high quality specimens with high percentages of bonded material. To develop the test, finite element simulations were performed to determine the basic geometric constraints that are required. Experiments were performed for both solid material and diffusion bonded material in order to confirm the simulations. Fracture surfaces were imaged to show that the tests were conducted properly. Future work could include a statistical analysis of many diffusion bonded specimens with this test.

4.5.3 Conventional Ultrasonic Non-Destructive Testing

Ultrasonic tests were conducted on most samples to test for large voids and delaminations prior to machining the test coupons. Ultrasonic testing bounces a high frequency stress wave through the sample and computes the delay in receiving the reflected wave and the magnitude of the reflection. By calibrating with the wave speed through the specific material, one can see if there is a void in the sample by determining if there is a reflection at the distance that corresponds to the interface between the two bonded materials. The testing was done using a 10 MHz single element probe on a ~ 58 mm (scan axis) X 18.5 mm (index axis) area within the visible weld lines along (shown in Figure 4.28). Most samples were tested using an automated raster scan with a scan resolution of 0.25 mm along scan axis and 0.25 mm along index axis. This scan was made possible by a hand programmed, computer controlled, 2-axis stage. This allows encoding of the exact position of any detected voids. A few samples had to be tested manually because they weren't flat enough for the stage and are not accurate with respect to positioning along the scan axis, but are accurate in terms of inspecting the presence of defects. Of the samples tested there were no defects found but the 10MHz wave is too large to detect anything but fully delaminated sections.



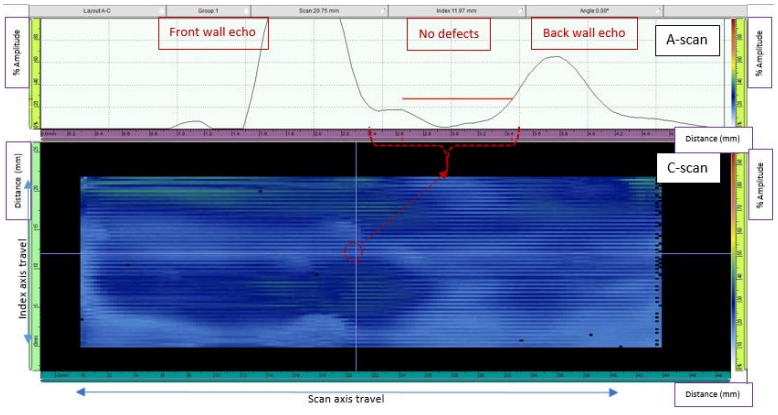


Figure 4.28 - Ultrasonic scan direction and sample results

4.5.4 Ultrasonic Analysis

Two sheet specimens were analyzed for their bonding performance ultrasonically in the experimental section of this dissertation. However, there were no voids detected despite the fact that there were many voids present in the microstructural images. The difficulty of detecting diffusion bonds is well established in the literature and is due to wave mechanics which is discussed in Section 2.4.6 of this dissertation. When the wavelength of the incoming wave is large compared with the defect size then the wave will essentially see an average of the material it's exposed to. There is a change in the waveform due to the diffusion bond but no reflection from it. This can be seen from analytical calculations from conventional wave mechanics given in graduate textbooks. Given a cylindrical void in a uniform medial of infinite length and radius a there is an analytical solution to how the wave will reflect and diffract off the void [77]. This gives the final equation of scattered power off of the obstacle as:

$$P_s = \frac{2I_0}{\pi\gamma}\int_0^{2\pi} |\psi_s(\theta)|^2 d\theta \tag{4-1}$$

Where I_0 is the intensity of the acoustic wave, P_s is the scattered power, ψ is the wavefuntion. The equation shows that when the wavelength is about equal to the void size a then the most scattering power is seen. When the wavelength is much smaller than the void size then the void will create a shadow zone in front of it. However, when the wavelength is much larger than the void size then the scattered power goes to zero. This code was implemented to confirm and can be seen in Appendix F.

The simulation passes a longitudinal ultrasonic wave through a meshed microstructure containing voids and then reads out the displacement at one end. The initial conditions for the simulation are a two dimensional approximation of real microstructures. These meshed microstructures are approximated from optical micrographs. The approximation scheme uses a variety of thresholds on an optical microscopic image. The image is discretized and seed points are placed on each grain. A threshold is taken to isolate the grain boundaries and the seed points are allowed to expand until they reach the threshold. Then another threshold is taken to isolate the voids. The grains are allowed to fill until the reach the voids. The result is an approximated microstructure. The stiffness of the mesh is broken down into voids, alpha phase, and beta phase with stiffness properties based on the bulk stiffnesses except for the voids which have a negligible stiffness. The microstructural data does not include the crystal texture but each grain is rotated to a random orientation. See Figure 4.29 for details on generating the mesh. Using the finite element software MOOSE [78], which is developed by Idaho National Labs. Using this software one can determine what the exact effects of the void are on the ultrasonic wave. This is done by performing a dynamic simulation with a moving lower boundary, a stiff upper boundary, and edges locked in the opposite direction. A micrograph is inputted into MOOSE by the procedure described above. See Figure 4.30 for the simulation layout. The equation used to solve the ultrasonic system is the wave equation using the Hilber-Hughes-Taylor time integration scheme [79]. This is a well established kinetic equation and is provided below. In this equation, M is the mass, K is the stiffness, x_0 is the displacement at time 0, and α is a parameter for numerical damping. The stiffness is numerically damped by including some of the previous timesteps in order to smooth out the results. In this case $\alpha = 1$. A unit pressure square wave is applied to the free end of the system that produces a triangular wave as the input wave.

$$M\ddot{x}_0 + K\big((1+\alpha)x_0 - \alpha x_{-1}\big) = 0 \tag{4-2}$$

$$\frac{2}{3} \leq \alpha \leq 1 \tag{4-3}$$

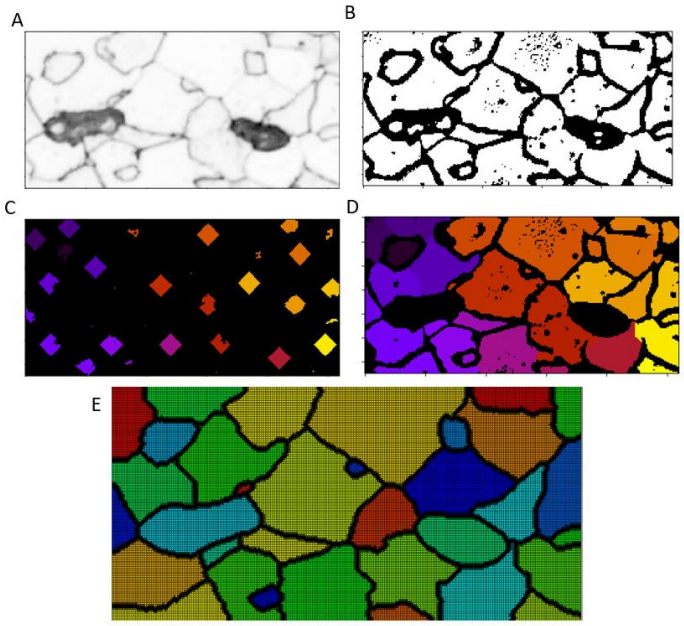


Figure 4.29 - Approximate geometry into MOOSE Framework. The optical microscopic image used for geometric information (A) is thresholded (B) which is used to restrict growth of seed points (C) until the reach the threshold (D). The threshold is removed except at the voids and the

resultant meshed geometry is used for the initial conditions (E).

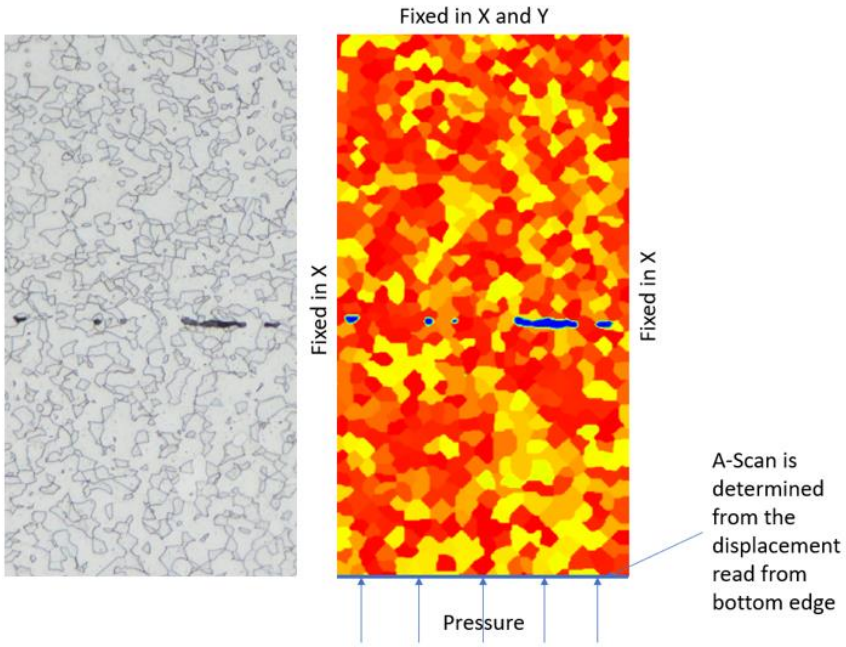


Figure 4.30 - Initial microstructural image (left) and the model geometry in the MOOSE FEM program (right). The voids are shown in blue.

To determine the effects of the voids on the wave the reference micrograph from Figure 4.30 has been modified to create a spread of bonding percentages. The reference micrograph had a bonding percentage of 64%. The far right void was then duplicated and positioned along the bondline to create the bonding percentages of 52.8% down to 11.6%. The largest void in the reference micrograph was removed for percentages 75.3% up to 100%. The same void that was used to build the lower bonding percentages was added or removed as necessary to complete the spread of bonding percentages. The advantage of building a custom bondline over taking a set of micrographs that have the appropriate bonding percentages is that the rest of the microstructure remains the same. This allows one to more effectively isolate the effects of the voids on the stress wave. The bondline of the various bonding percentages can be see in Figure 4.31.

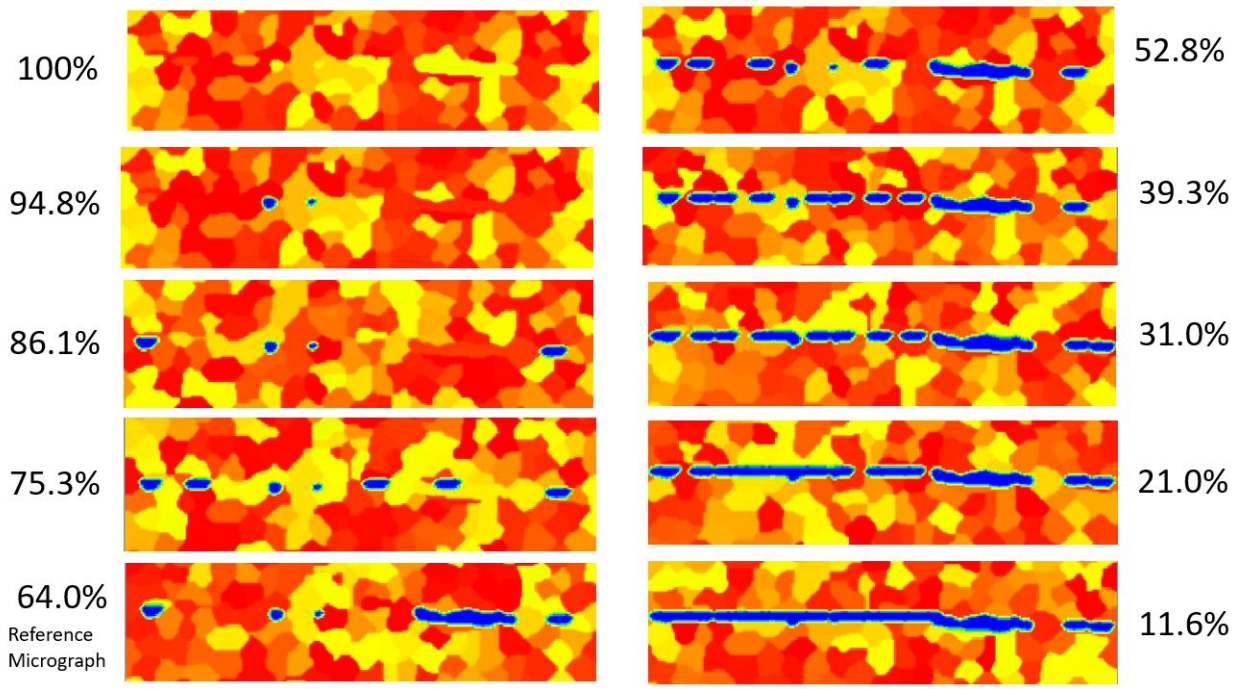


Figure 4.31 - The different void configurations corresponding to the bonded percentages of each simulation. The voids in blue have negligible stiffness.

Results

The results show that the ultrasonic wave passes through the voids reducing the deformation in the areas behind the void relative to the wave. In effect the void casts a shadow and the displacement must be transferred by the bonded material. The displacement field has little noticeable change with respect to the microstructure. However, the acceleration field shows that there is an effect from the grain boundaries and grain misorientations that changes the way the wave propagates. These effects can be seen in Figure 4.32.

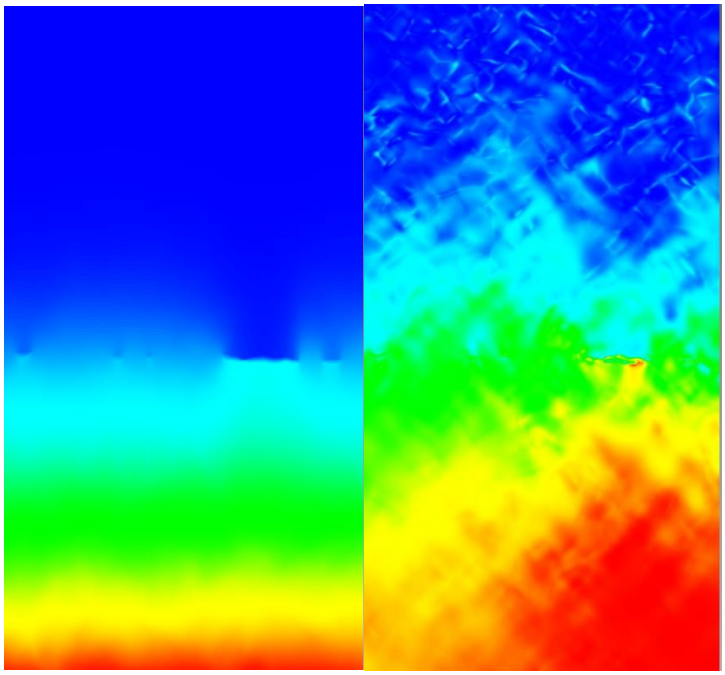


Figure 4.32 - Ultrasonic wave propagation with voids. The displacement field (left) and the acceleration field (right).

The 10MHz wavelength is large enough to not get substantial reflections off the majority of these small voids according to the experimental data. Therefore, before any analysis is done, this model is confirmed by using a 1GHz wave to see if the wave is propagating correctly and to see if it gets reflections off the voids. The results are pictured in Figure 4.33. It shows that you do get reflections off the voids, the program is operating correctly, and that one could use acoustic microscopy to detect voids in this sample but only if the sample thickness was small. However, when the limits of the plot are fixed, one can see that there is substantial loss in intensity of the wave. There is no viscous damping in this model. Therefore the losses are due to scattering off the microstructure as the wave passes through the different crystals. This explains the changes in intensity that are to be found in Derby's research [27].

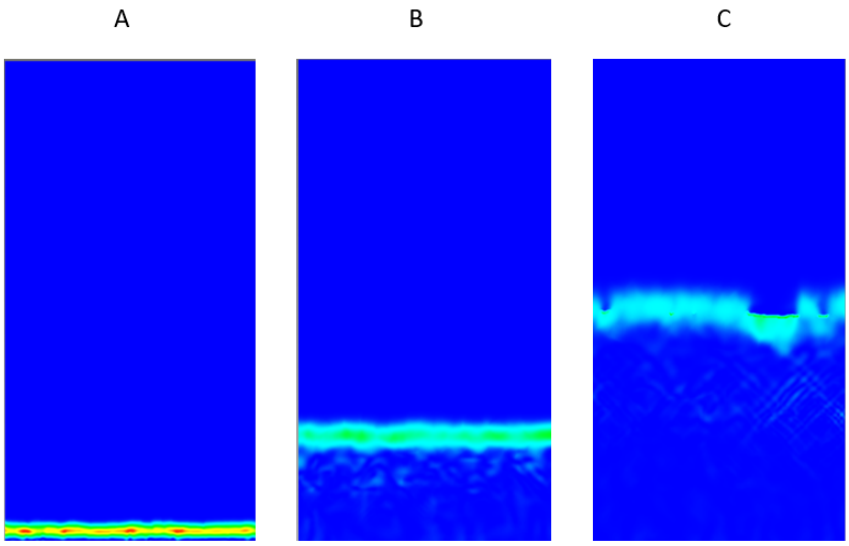


Figure 4.33 – 1GHz high frequency wave propagation. The displacement is shown here with all images having the same scale range. In A) the displacement is concentrated as the input wave. In B) the input wave has reduced in intensity because of scattering off the randomized grain interfaces. By the time it reaches the voids C) it has lost a substantial amount of its coherency though it does produce a reflection.

The displacement of the lower boundary of the simulation volume is used to determine how the wave is affected by the voids. Each of the bonding percentages have the same mesh density, 0.25μm quads, and the same input wave. The variation in the response of the different reflected waves which can be seen in Figure 4.34. These are separated into three different sections. The input wave is the base wave that is pushed using a square pressure wave. The reflection off the voids can be seen next (if any). Finally, the reflections caused by the wave that bounces off the back wall.

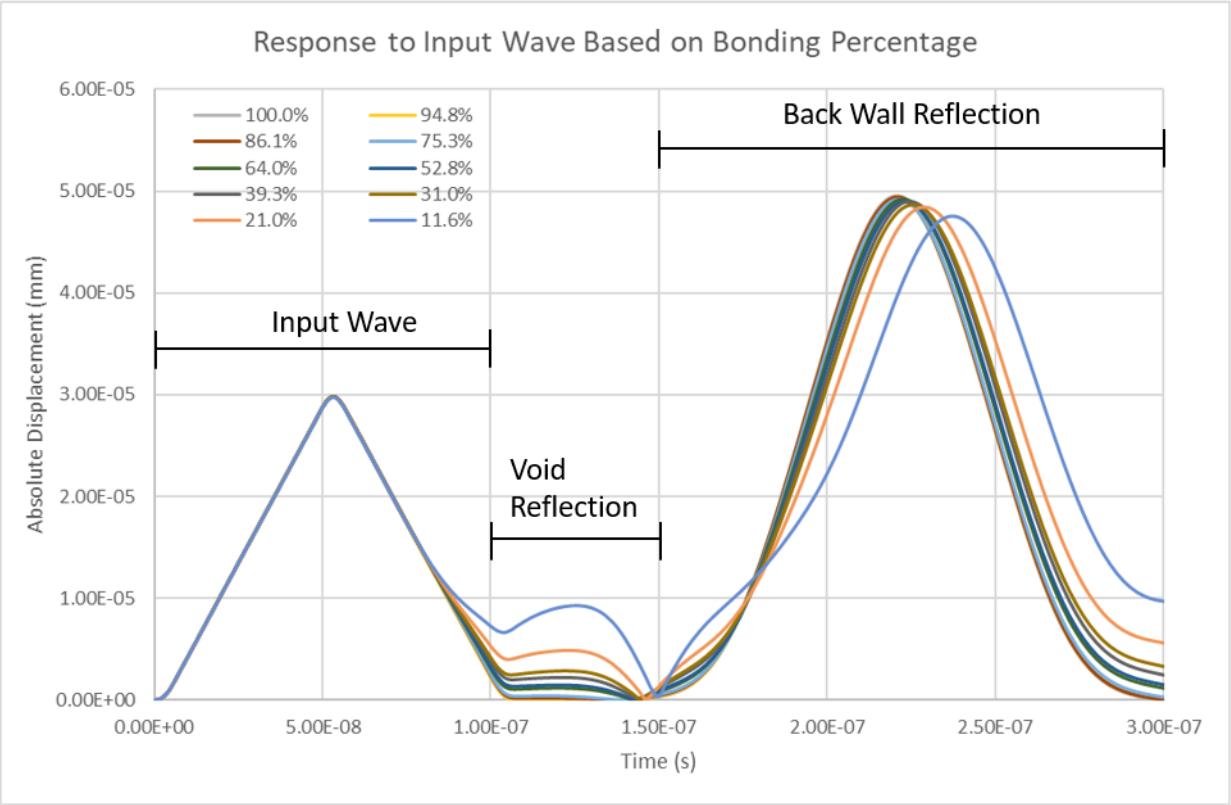


Figure 4.34 - Effects of bonding percentage on reflected wave. The voids do cause a small reflection but it is undetectable at higher bonding percentages.

Discussion

The voids cause a change in the reflected waveform as well as some scattering off the voids especially in lower bonding percentages. This can be seen in Figure 4.34. The triangular section at the beginning of the wave is the input displacement caused by a square pressure wave. The small hump to the right of the triangular section is the reflection off the voids. Note that while it is fairly high at the lower bonding percentages, it is very small at large percentages. This simulation does not take into account the ultrasonic couplant. In reality the square wave would interact with the top surface of the sample, then interact with the voids, then the back surface. This means that the small reflection caused by the voids shown below still has to pass through the

titanium/couplant interface to make it to the sensor and therefore is further reduced in intensity. This is most likely why there is no noticeable reflection off the voids in the experimental data. Detection limits for void reflection are most likely around the 21% and lower bonding percentages. Good bonding that companies are looking more for is more on the order of 80-100% but there is no detectable reflection from the voids in these parts. These waves effectively seem indistinguishable from the 100% bonded case.

Unless the processing conditions are extremely poor the most valuable tests with ultrasound are not looking for large debonded areas or low bonding percentages. These would be ruled out during development of the product. Good bonding that companies are looking more for is more on the order of 80-100% but there is no detectable reflection from the voids in these parts. These waves effectively seem indistinguishable from the 100% bonded case. When done properly there is going to be a spread over relatively high bonding percentages. Non-destructive testing would be looking to obtain details on how large this spread is and if there are many major areas that have lower bonding than others. It is therefore improbable that there is going to be any reflections off the voids in a properly made part. Looking at the high bonding percentages form 75.3%-100% there is no noticeable change in the wave except for a small frequency shift in the back wall reflection. The most valuable information that can be gathered, therefore is not from reflection off the voids but from the changes in the waveform due to the voids. With the spectral decomposition one can calculate the spectral power and see that there is a clear correlation of the waveform. See Figure 4.35 for details.

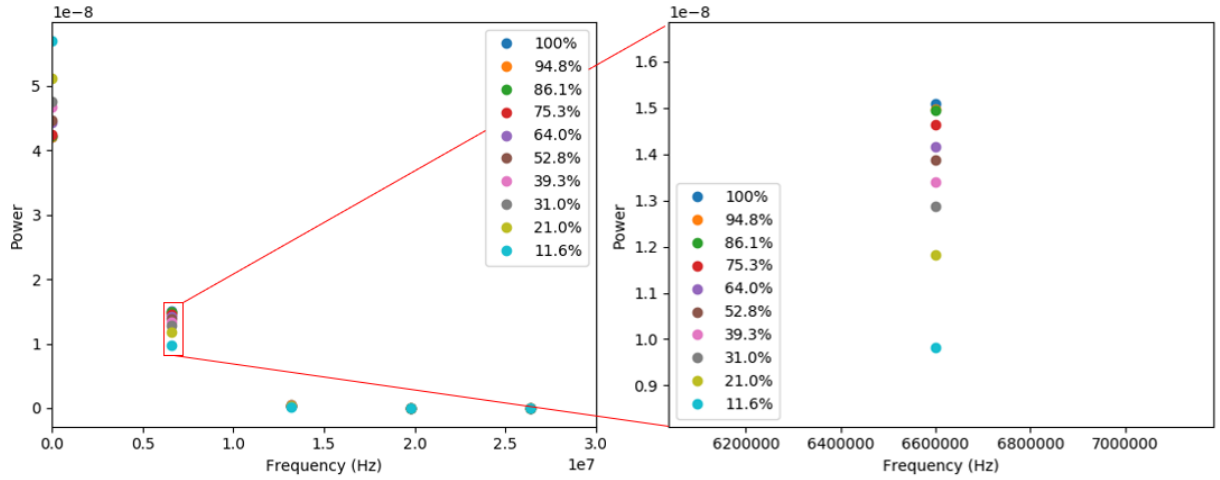


Figure 4.35 - Power spectra. There is a clear correlation between the power and the bonding percentage.

Even though the power spectra of the simulations were well correlated, it is not just to look at the bonding percentage as the primary metric but taking a look at the voids themselves. This is a probable source of error in past research. The bonding percentage only describes part of the relationship between the incoming ultrasonic wave and the bondline. The interaction between the wave and a few larger voids vs interacting with many smaller voids will result in a vastly different reflection despite having the same bonding percentage. Depending on the surface roughness and matching of the two mating surfaces the voids formed can be a variety of lengths. To show the differences in the ultrasonic performance an additional simulation was performed at a bonding percentage of 65.1% and compared with the base micrograph of 64.0%. The additional simulation has voids that are very small relative to the voids in the base micrograph. Figure 4.36 displays the base 64% reference elasticity in the C_{1111} (vertical) direction, the displacement magnitudes at the same timestamp, and the waveform response. The displacement magnitudes show a visible

difference in how the wave passes by the voids with the larger voids of the 64% case causing an increase in displacement before the void and a decrease after it. Logically, the smaller voids do not cause nearly the disturbance the large ones do. This is additionally shown in the waveform where one can see a reflection off the voids and a larger shift in the back wall reflection in the 64% case while the 65.1% case appears closer to a solid sample. Therefore, any experimental calibration using this model would have to take into account the statistics of the void sizes in order to be accurate.

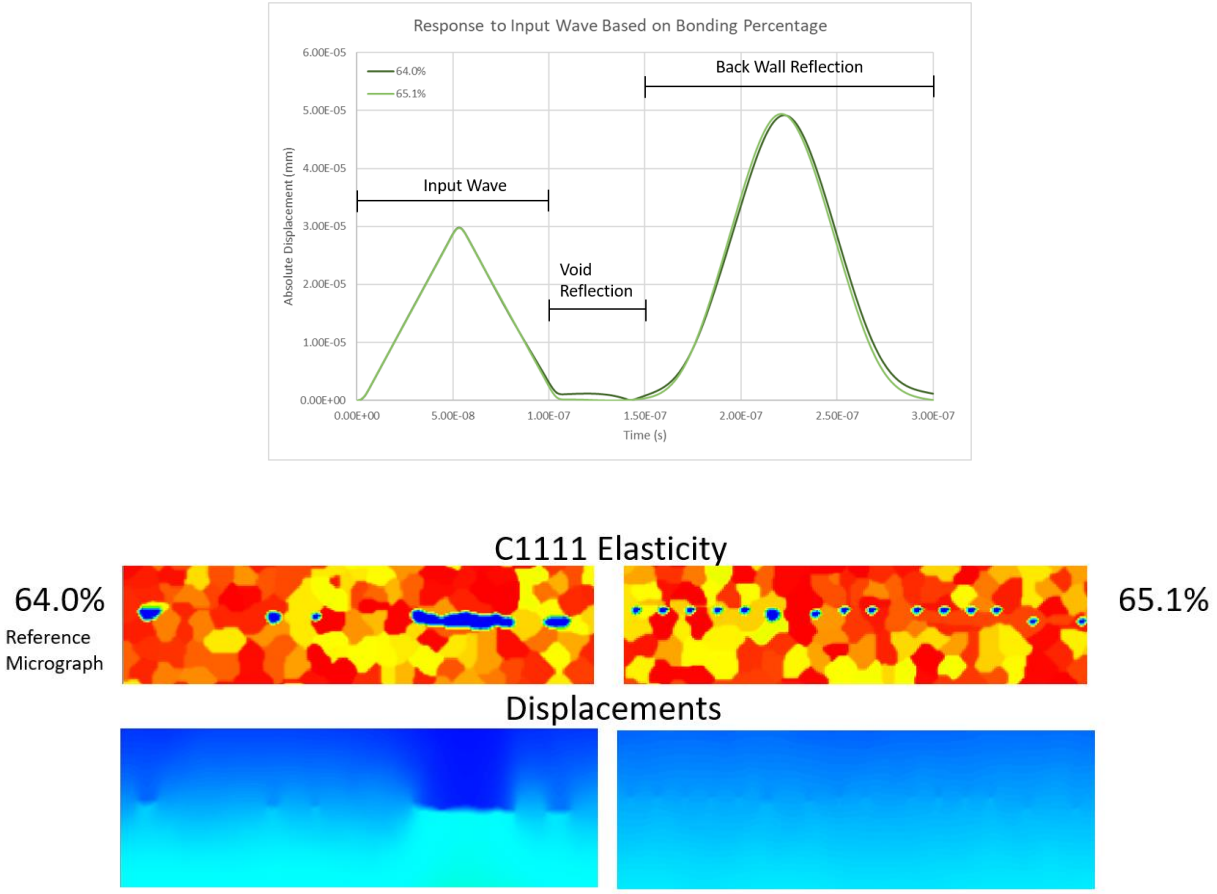


Figure 4.36 - An additional simulation to show that there is an influence from the size of the voids along with the overall bonding percentage. The smaller voids produce no reflections from the incoming wave and they also have less stretching of the back wall reflection (and therefore change the power analysis)

To determine the effects of voids experimentally one would look for a reflection at the void distance first. If that is seen then the bonding is extremely poor and can be calculated. Since there is a frequency correlation in the power spectra with the bonding percentage, the power of the waveform can be used to determine the how voided the sample is by comparing the frequency of the different reflection waves relative to the reference wave without any voids in the sample. This is similar to what others have described with interfacial stiffness methods and phase shift methods. One would have to take into account not just the bonding percentage but obtain data on the void size and distribution as well in order to be accurate. This could be built and calibrated experimentally for each individual material combination. Such a bonding model would be very valuable to companies when inspecting parts that need to determine the exact nature of the bonding over a large area at high bonding percentages.

Conclusions

In this work, a total of 81 different samples were diffusion bonded and analyzed using conventional ultrasound. Micrographs were produced and showed that there were substantial voids in most cases. Despite poor bonding none of these samples produced a significant reflection off the voids while looking for simple reflection off the voids. To determine exactly what these effects would be, the microstructure has been imported and meshed in a finite element program. Then a high frequency longitudinal stress wave is used to determine the reflections off the voids and to determine the changes in waveform due to the voids. It was determined that there would only be a reflection with very low bonding percentages and that at higher percentages spectral analysis would have to be used. This is because there is a very clear change in the wave due to the presence of the voids. It was also determined that the void size has a large effect on the properties of the reflected wave. This model will be very useful in the future, as experiments are done to produce

a material and void specific calibrated model that can be used in practical applications. Additionally, the model can be used to confirm the utility of different ultrasonic techniques.

Chapter 5 Four-Sheet SPF/DB

5.1 Introduction

Superplasticity is often combined with diffusion bonding to create parts out of multiple sheets that contain complex superplastic structures all welded together with diffusion bonding. This dissertation deals with the Rockwell four-sheet process. As the name suggests, this process uses four titanium sheets that are welded together with a pattern to form a part. Two external "face" sheets form out to an external die and form the edges of the part, Figure 5.1a shows the overall forming process. Two internal "core" sheets are roll seam resistance welded intermittently in a rectangular pattern forming 1 inch by 2 inch cells with gas pathways in between the cells, Figure 5.1d shows what the core sheets look like during partial forming without the face sheets. These cells have gas pressure and expand to form I-beam style supports for the external sheets (completed pattern is shown in Figure 5.1b). This creates a finished part, in this case in the form of a test panel, that is both stiff and strong (shown in Figure 5.1c). The important features of this process are the quality of the diffusion bonding and the accuracy of the internal sheet superplastic forming. The superplastic performance is measured by the distribution of the thickness of the formed internal structures while the diffusion bonding performance is measured by the percentage of bonded area at the bondline. The characterization utilizes a variety of image processing techniques on samples mounted from received sectioned panels. The following document is an explanation

of those techniques as well as a summary of the data collected along with modeling efforts done to explain the results.

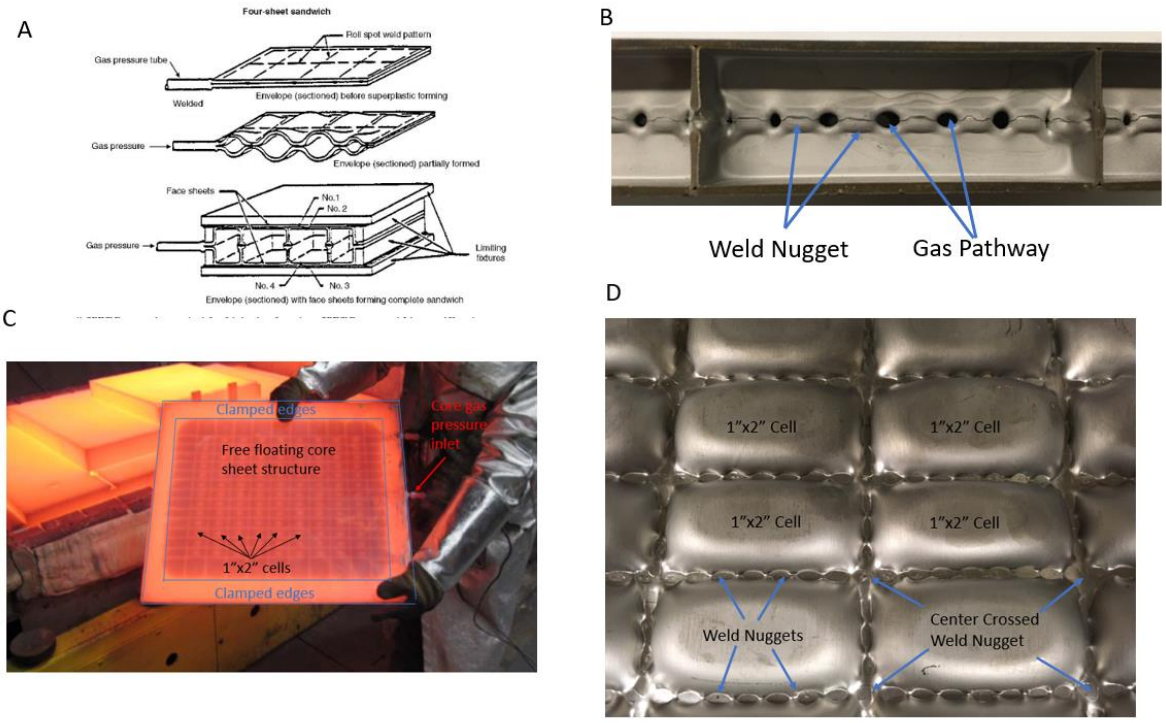


Figure 5.1 – Description of the geometry of the four-sheet system. A) describes how the sheets are formed [3]. B) Shows how the final product looks and the difference between the gas pathways and weld nuggets. C) shows what the panel looks like out of the press. D) the core sheets partially formed with the weld nugget pattern but with the face sheets removed.

5.2 Materials and Methods

Thirty five independent panels of the four-sheet process were analyzed. The panels themselves cover a variety of materials and dimensions but do not vary in the forming/diffusion bonding temperature, pressure, or time (with the exception of a temperature study consisting of 4 samples). The primary processing parameters for bond quality are fixed at 1475F, 300psi, and 3 hours.

Listed in Table 5-1 are the samples obtained from Boeing in the order received. The following section describes each sample group individually along with a summary of the final bonding results. The materials closest to the ones used for these panels are listed in Table 4-2.

Table 5-1 – Sample list of panels obtained

Designation	Core Material	Face Material	Core Thickness (in)	Face Thickness (in)	Cell Size (in)	Analysis Group
6	TIMET 54M	TIMET 54M	0.040"	0.040"	1"x2"	Group 1
8	VSMPO 64 FG	VSMPO 64 FG	0.040"	0.040"	1.275"x2.55"	Group 1
11	VSMPO 64 FG	TIMET 64 SG	0.040"	0.040"	1.25"x2.5"	Group 1
1x	RTI 64 FG	TI 6242 & Beta 21s	0.040"	0.040"	1"x2"	Group 1
F22-Edge A	ATI 425	ATI 425	0.040"	0.040"	1"x2"	Group 2
F22-Edge C	ATI 425	ATI 425	0.040"	0.040"	1"x2"	Group 2
F6-Edge A	RTI 64 FG	RTI 64 FG	0.040"	0.040"	1"x2"	Group 2
F6-Edge C	RTI 64 FG	RTI 64 FG	0.040"	0.040"	1"x2"	Group 2
Tim3F1-Edge A	TIMET 54MX	TIMET 54MX	0.040"	0.040"	1"x2"	Group 2
Tim3F1-Edge C	TIMET 54MX	TIMET 54MX	0.040"	0.040"	1"x2"	Group 2
F23-Edge A	TIMET 54M	TIMET 54M	0.040"	0.040"	1"x2"	Group 2
F23-Edge C	TIMET 54M	TIMET 54M	0.040"	0.040"	1"x2"	Group 2
F6	RTI 64 FG	RTI 64 FG	0.040"	0.040"	1"x2"	Group 3
F7	RTI 64 FG	RTI 64 FG	0.040"	0.040"	1"x2"	Group 3
F8	RTI 64 FG	RTI 64 FG	0.040"	0.040"	1"x2"	Group 3
F9	RTI 64 FG	Ti 6242	0.040"	0.032"	1"x2"	Group 3
F14	TIMET 54M	Ti 6242	0.032"	0.032"	1"x2"	Group 3
F15	TIMET 54M	Ti 6242	0.025"	0.025"	1"x2"	Group 3
F22	ATI 425	ATI 425	0.040"	0.040"	1"x2"	Group 3
F23	TIMET 54M	TIMET 54M	0.040"	0.040"	1"x2"	Group 3
F1	ATI 425	ATI 425	0.032"	0.032"	1"x2"	Group 4
F3	ATI 425	ATI 425	0.022"	0.032"	1"x2"	Group 4
F11	TIMET 54M	TIMET 54M	0.032"	0.032"	1"x2"	Group 4
F12	TIMET 54M	TIMET 54M	0.032"	0.040"	1"x2"	Group 4
F13	TIMET 54M	TIMET 54M	0.025"	0.032"	1"x2"	Group 4
F24	TIMET 54M	TIMET 54M	0.032"	0.032"	1"x2"	Group 5
F26	TIMET 54M	TIMET 54M	0.032"	0.032"	1"x2"	Group 5
F28	RTI 64 FG	RTI THOR	0.040"	0.040"	1"x2"	Group 5
F29	RTI 64 FG	RTI THOR	0.040"	0.040"	1"x2"	Group 5
F30	VSMPO 64 FG	VSMPO 64 FG	0.040"	0.040"	1"x2"	Group 5
F31	TIMET 54M	TIMET 54M	0.025”	0.032”	1"x2"	Group 5
Temp1	RTI 64 FG	TIMET 64 SG	0.040"	0.040"	1"x2"	Group 5
Temp2	RTI 64 FG	TIMET 64 SG	0.040"	0.040"	1"x2"	Group 5
Temp3	RTI 64 FG	TIMET 64 SG	0.040"	0.040"	1"x2"	Group 5
Temp4	RTI 64 FG	TIMET 64 SG	0.040"	0.040"	1"x2"	Group 5

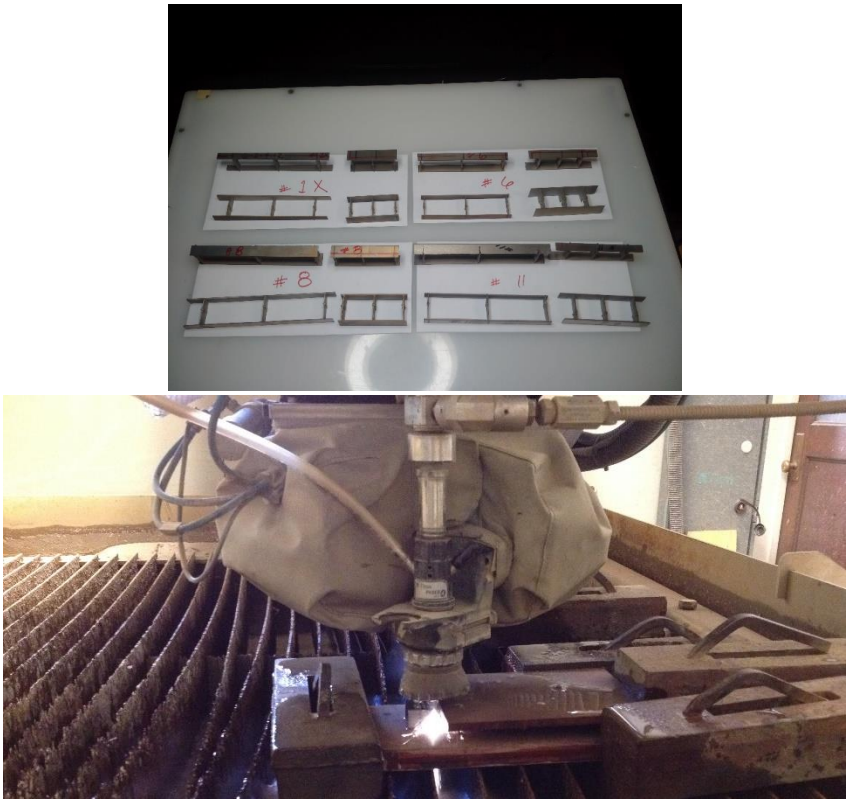


Figure 5.2 - Panels are received as pieces then are trimmed with the waterjet and labeled

The samples received were in the form of trimmed sections from flat panels. The trimmed sections received from Boeing are typically after waterjet, laser, or EDM cutting from the testing company Westmoreland. The samples are cut from the panel by machining a small section with a waterjet cutter (as shown in Figure 5.2). From each panel, usually four samples are mounted. Usually two of the samples will be from the weld nugget and two will be from the spacing in between weld nuggets as shown in Figure 5.3a. These four are evenly distributed with two from the long part of the cell and the other two from the short part of the cell as shown in Figure 5.3b.

The sample is prepared first by mounting the cross section in phenolic resin. Then a series of polishing steps are performed to obtain the desired polish before metallographic etching. Specifically the sample is ground by hand with 240, 320, 400, 600, 800, and 1200 grit silicon

carbide paper. Then the sample is polished with 6µm and 3µm water based diamond suspension on polishing cloth. Finally the samples are polished with 0.05µm aluminum oxide suspension and hydrogen peroxide. The samples are ultrasonically cleaned for 1 minute in clean deionized water between each step from grinding with 1200 onward. After the final polish, the samples are ultrasonically cleaned for 30 minutes. They are then scrubbed vigorously under running deionized water. The samples are dried and etched for 10-20 seconds with Kroll's reagent.

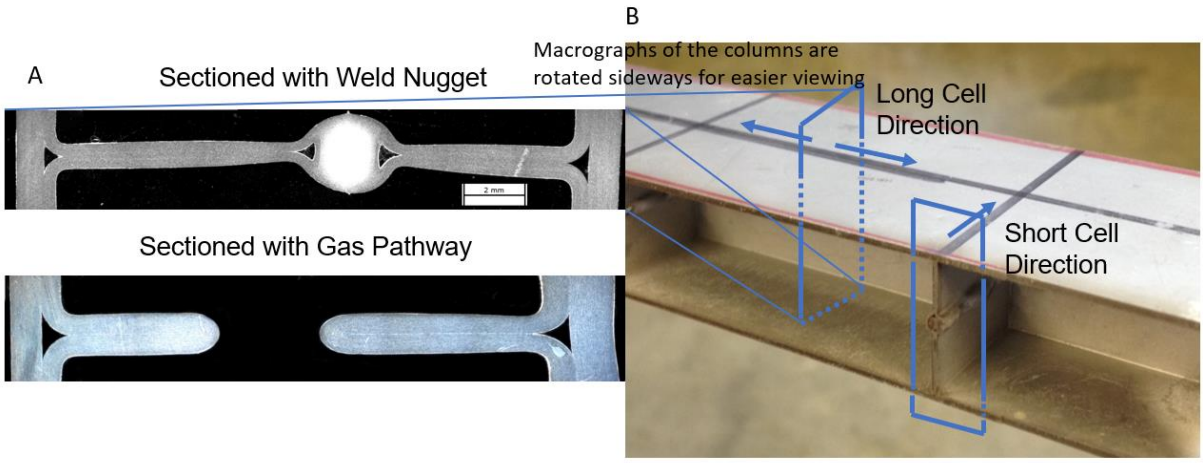


Figure 5.3 – Sectioning of the webs from the panels from an analysis. Each analysis consisted of four webs sections. One each from the weld nugget and gas pathway (a) along both the short cell and long cell directions (b). The column macrographs are rotated sideways for easier viewing.

The bonding performance is characterized by the percentage of solid material along the bondline, called bonded percentage. To determine this, the samples are imaged with a Nikon or Olympus optical microscope at 500X magnification. This gives a micrograph image of around 200 micrometers in width. Each micrograph typically contains several voids but not the complete distribution. Therefore, several micrographs are taken per sample, each at the important locations. One micrograph is taken between the face sheet and the core sheet from both the top and bottom of the sample. Additional micrographs are taken between the core sheets from both the top and

bottom of the sample. The bonding ratio for the total of four micrographs per sample are computed using an image processing algorithm and associated program that detects the void based on the contrast between the voids, which are typically black, and the grain structure which is typically light colored with fine lines at the grain boundaries. Example micrographs with layout are shown in Figure 5.4.

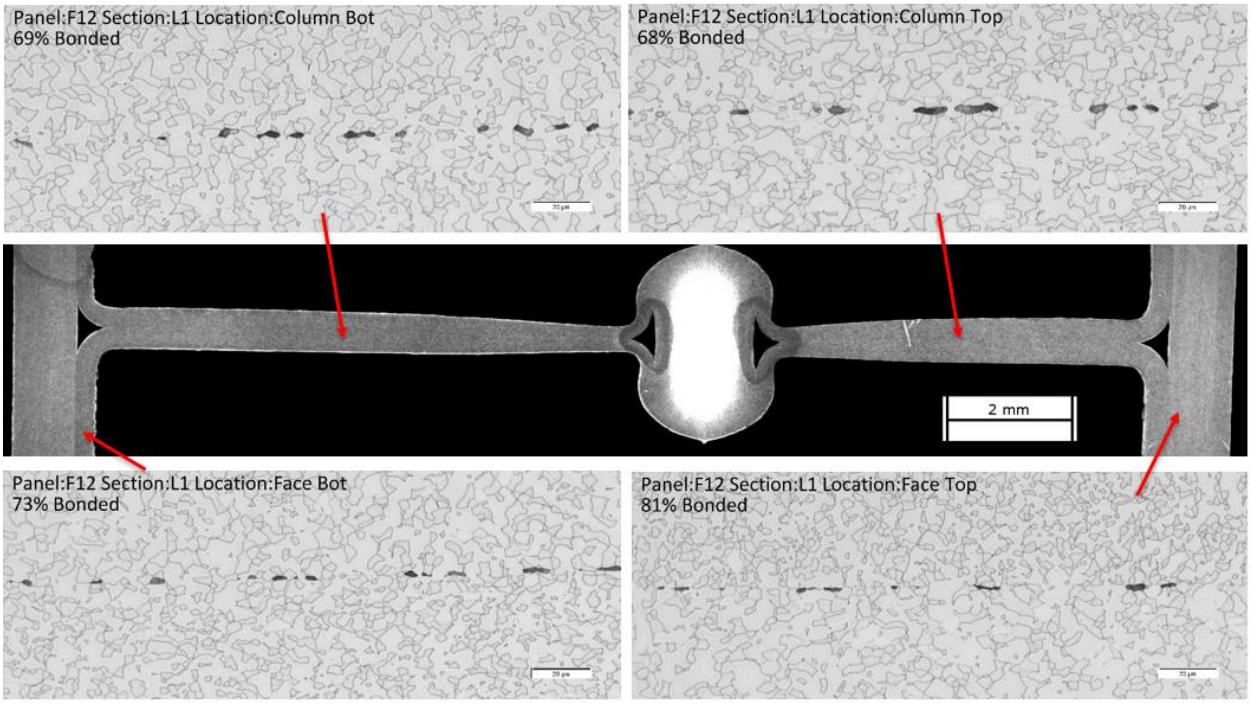


Figure 5.4 – An example of the diffusion bonding analysis for a single web. Four micrographs are taken, one from each major section of the web.

Void Detection Program

The void detection program is listed in Appendix F. The algorithm uses a series of blurs and thresholds to determine where the idealized bondline is and where the voids are along it. The program starts with a heavy blur and a significant threshold. This gives a binary image where the darkest spots, which are inevitably the deepest voids, are detected. Then the locations of the pixels

are averaged out in the y direction to get a location of the center of the voids. These averaged out y values are curve fit with a simple linear function to produce a theoretical bondline. Once the bondline is obtained then the image is once again blurred but by a much smaller value. The purpose for this blur is smooth out the grain boundaries. A normal threshold would detect the grain boundaries since they are one of the darkest things in the image. However, they are also very thin. A median blur takes the average value of all the pixels around the center pixel within a certain distance. This blurs out the sharp black lines of the grain boundaries creating a smooth grey image where they were. Once this is performed, the darkest values on the image within a vertical pixel range of the bondline (plus or minus 15 pixels) is mapped onto the bondline as its value. Then the bondline is thresholded for the value which best detects the voids and neglects the other features. Each image is checked and adjustments are made to the thresholding intensities so that the detection program detects every void. From sample images, the detection is usually within $\pm$1% of the measured values and at most $\pm$3%. The program works very well for clearly defined voids but it has difficulty especially with voids that are as thin as the grain boundaries (because they've been blurred out), beta grains that are small and dark and near the bondline, and voids that are shallow enough to reflect the light from the microscope creating a bright void. However, each image is checked over by hand and the main image processing parameters are saved.

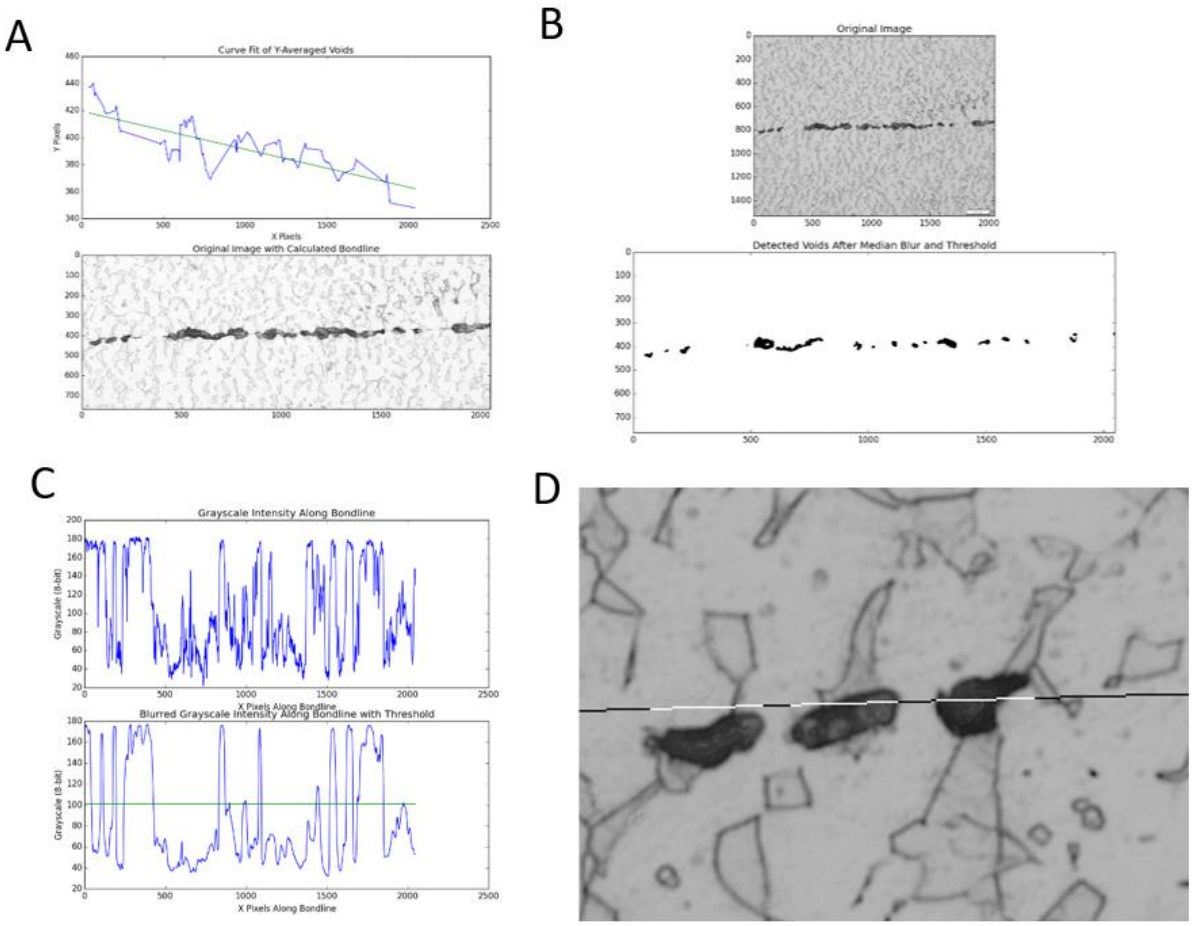


Figure 5.5 - The process starts isolated voids (a), the bondline is a curve fit of the void averaged y values (b), a denoising operation and a threshold to determine which pixels are voids and which aren't (c), and the result with detected voids as white and bonded areas as black along the bondline (d)

5.3 Superplastic Characterization

The superplastic performance is characterized by the distribution of the thickness of the column, selective measurements across a long cross section, and statistics on the gas pathways. Perfect superplasticity would be a completely consistent thickness throughout the sample. This is never achieved in practice so the thickness is measured in a couple ways. The distribution along the columns are measured with image analysis using macroscopic images of the cross sectioned samples that were used to take the micrographs. The macroscopic images are a composite of small

images taken from the microscope of the MicroVu, a microscope used to compute the edges of machined surfaces. A series of seven images are taken at 36X magnification every 2.54mm. This covers the whole length of the sample and gives overlapping images that can be used in a stitching program to combine them together. Photoshop has sophisticated stitching algorithm and uses the pattern of the Harris corner detection on each individual image to align the images together. The combination of using the stage, a level mount, and uniform stitching in Photoshop give a good image that represents the actual geometry with the desired tolerances. The phenolic mounting material has a significant amount of fine white particles in it that show up in the macrograph. This is important to remove, not just because of aesthetics, but because the edge detection program that is run on the image receives false positives from the particles in the mount especially for the particles that are adjacent to the actual sample. Therefore before calculating the thickness the background is manually erased in Photoshop as shown in Figure 5.6.

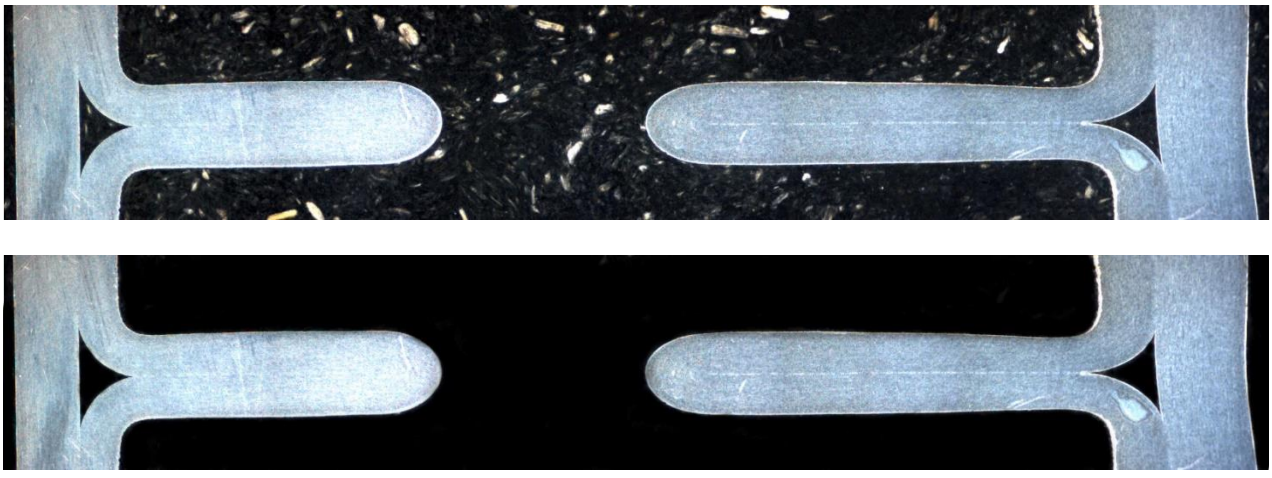


Figure 5.6 - Original macrograph with a typical phenolic mounting material (top) compared with the digitally erased mount (bottom). These columns are 0.75” horizontal.

With a black background and a lighter colored sample, the edges are detected with a simple derivative of the threshold of the image. However, for images with more noise and debris, a more

sophisticated edge detection algorithm is necessary. Canny edge detection is used to produce a binary image of only the major edges. Canny edge detection is a well established image processing algorithm and is built into OpenCV. The way it works is by first taking a Gaussian blur to get rid of the noise in the image. Second, it computes both the magnitude and direction of the gradient of the image. Next is discarding any pixels that are not a local maximum. This results in detecting only thin edges. Finally, a selection step that selects pixels as an edge if they are above a gradient value, discards them if they are below a gradient value, and for pixels between the two values they are selected as an edge only if they are next to an above gradient value pixel. The rest of the program to compute the thickness determines the most upper top pixel that is on the edge and the lowest bottom edge. The difference between the two is the thickness of the column of the four-sheet SPF/DB.

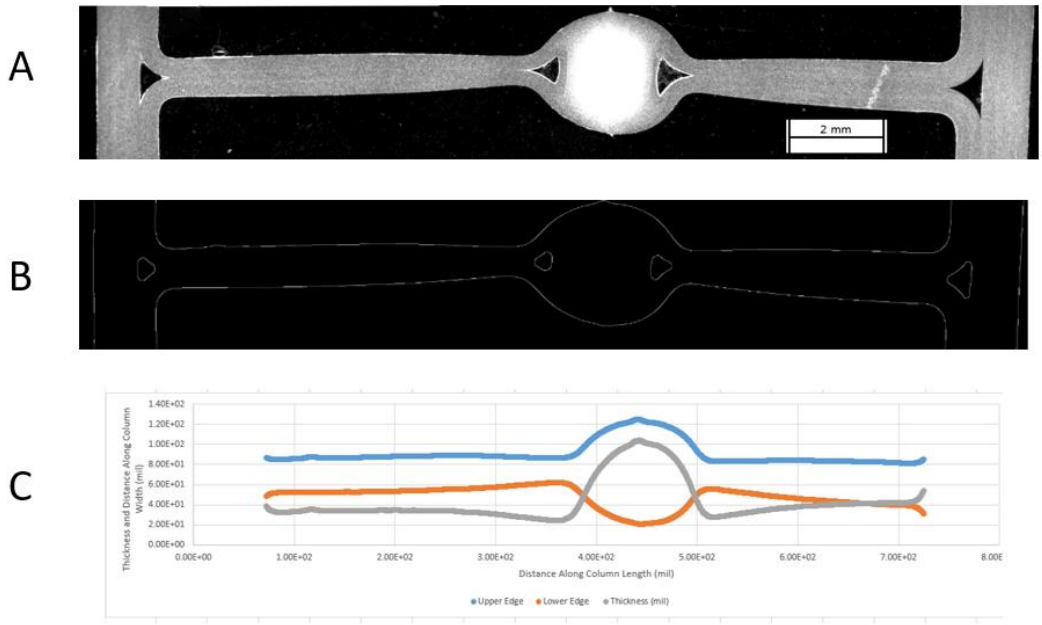


Figure 5.7 - Thickness distribution calculations. (A) The original image is (B) operated on by Canny edge detection then the edges are computed and (C) the thickness distribution is calculated

The final analysis for the four-sheet SPF/DB process is for each panel is large scale measurement of the thickness distribution of a single section of each panel. The typical profile of the thickness distribution is shown in Figure 5.7.

Figure 5.8 - Sample panel measurements with the thickness values and line locations of measure distances

5.4 Materials and Samples

The complexity of the analysis of each group of samples received increased over time. Below are notes corresponding to each analysis group. For the list of panels that are associated with each group see Table 5-1 above.

Group 1

This was the first attempt at imaging the panels, creating micrographs for the panels, and using the bondline calculator and the results were imperfect. There was a significant debris on the micrographs which was a source of confusion for Group 1. It was eventually found to be debris from the final polishing step. There were also significant defects on the micrographs due to the mounting material. This caused disturbances in both programs that calculated the bondline ratio

as well as the thickness distribution. The mounting material was made from a quick cure acrylic produced by Allied that was dyed black with an oil based ink. This gave a dark enough mount to image with the microscope but the mounting material contains some voids that showed in several of the samples. In the micrographs, the calculated bondline is plotted along with the detected voids. The macro images included an attached image based plot of the thickness that was at the exact same scale as the image.

Group 2

The second group of panels consisted of edge trimmed pieces from four different panels. Two of the samples were mounted in a black, glass filled epoxy that was developed for a mounting press. The rest were mounted in phenolic in the same mounting press. The phenolic mounts were digitally erased for clarity while the epoxy mounts were adequate without any manipulation. This was repeated for center sections of the panels for F6, F22, and F23. The micrographs also contained debris and some were out of focus due to slightly angled mounts.

Group 3

The third set of samples were panels F6, F7, F8, F9, F14, F15, F22, and F23. Some of these panels are repeats from the last sample set but are taken from material closer to the center of the panel as opposed to the edge trimmings. These were analyzed using the procedure previously indicated. All panels were all mounted in phenolic in the mounting press with the mounting material being digitally erased from the image. The micrographs no longer have any debris. It was discovered during the last panel group testing that the debris were particles from the final polishing step that could not be lifted by the ultrasonic bath. Therefore the vigorous scrubbing step was performed which successfully removed most of the debris. The micrographs are imaged with an Olympus

microscope with automated image processing that takes images at multiple depths to remove the variation in focus. The micrographs from the Olympus microscope are slightly larger than the micrographs from the Nikon.

Group 4

This panel group consisted of panels F1, F3, F11, F12, and F13. The panels were analyzed with the procedure described in the methods section (Section 5.2) and imaged on the Olympus microscope. Panel F3 had significant distortion due to a pack leak during superplastic forming. This was the only major visual difference between the samples. Some of the materials and process conditions are still unknown.

Group 5

Panel Group 5 includes the last of the panels which were completed with all of the bondline information and most of the superplastic information. They used the procedure described in Section 5.2. The superplastic analysis was deemed unnecessary for these panels since Boeing had essentially perfected their forming and were no longer concerned about it. The purpose of F26 and F28 was to determine the effects of presanding on the finished panels. This group includes an additional temperature study that was to determine the effects of temperature on these panels and what parameters were needed to achieve good bonding. The results of the panels were that pre-sanding has little to no effects on the bonding performance, something that was already studied with two sheet diffusion bonding analysis of Chapter 4. The temperature study determined that, while there are some optimal temperatures that can be reached to complete bonding, they are too expensive for regular use due to creep of the superplastic die. The additional panels were for other

projects and have no conclusions since they were isolated cases or UW was not given sufficient information to make any conclusions.

Bonding Performance

Table 5-2 below is the aggregate of all the micrographs taken and their bonded ratios. In this table, L represents the long cell direction, S represents the short cell direction, CB represents the core sheet bottom, CT represents the core sheet top, FB represents the face sheet bottom, FT represents the face sheet top, and 1 & 2 are there since for most panels two columns were take in each direction.

Table 5-2 – Bonded ratios panel micrographs. Cells that are empty don‘t have data for those locations.

Panel	L1 CB	L1 CT	L1 FB	L1 FT	L2 CB	L2 CT	L2 FB	L2 FT	S1 CB	S1 CT	S1 FB	S1 FT	S2 CB	S2 CT	S2 FB	S2 FT
1X	0.84	0.86	0.67	0.72	0.85	0.7	0.54	0.63	0.67	0.72	0.76	0.52	0.49	0.75	0.49	0.6
6	0.59	0.55	0.42	0.63	0.8	0.68	0.62	0.74	0.79	0.69	0.8	0.37	0.56	0.55	0.72	0.47
8	0.47	0.25	0.61	0.64	0.67	0.49	0.72	0.5	0.56	0.56	0.5	0.69	0.24	0.21	0.24	0.26
11	0.78	0.68	0.77	0.6	0.83	0.76	0.72	0.66	1	0.76	0.7	0.65	0.78	0.65	0.7	0.67
F22A	0.56	0.72	0.8	0.9												
F22C	0.53	0.71	0.89	0.97												
F6A	0.57	0.43	0.83	0.21												
F6C	0.77	0.69	0.75	0.58												
TIM3A	0.41	0.6	0.17	0.66												
TIM3C	0.64	0.74	0.94	0.84												
F23A	0.68	0.57	0.92	0.88												
F23C	0.62	0.6	0.72	0.73												
F6	0.83	0.62	0.86	0.78	0.81	0.7	0.8	0.83	0.47	0.79	0.75	0.58	0.64	0.43	0.55	0.66
F7													0			
F8	0.87	0.67	0.92	0.9	0.75	0.82	0.96	0.95	0.64	0.76	0.94	0.89	0.61	0.66	0.87	0.81
F9	0.88	0.85	0.53	0.78	0.8	0.55	0.47	0.56	0.8	0.78	0.55	0.61	0.45	0.66	0.44	0
F14	0.75	0.7	0.6	0.57	0.82	0.9	0.56	0.53	0.85	0.8	0.55	0.44	0.87	0.75	0.62	0.38
F15	0.45	0.43	0	0.24	0.43	0.72	0	0.59	0.64	0.53	0.24	0.35	0.45	0.61	0.46	0.46
F22	0.44	0.79	0.76	0.84	0.81	0.7	0.7	0.62	0.77	0.56	0.84	0.71	0.69	0.82	0.84	0.78

F23	0.67	0.5	0.69	0.73	0.78	0.76	0.79	0.68	0.8	0.84	0.81	0.88	0.7	0.84	0.89	0.85
F1	0.66	0.7	0.86	0.65	0.83	0.85	0.9	0.88	0.78	0.6	0.93	0.73	0.63	0.66	0.86	0.89
F3	0.13	0.12	0.21	0.04	0.24	0	0	0.01	0.18	0.02	0.29	0.06	0.09	0	0.23	0.16
F11	0.62	0.52	0.74	0.75	0.63	0.76	0.75	0.62	0.41	0.45	0.57	0.7	0.52	0.51	0.57	0.7
F12	0.69	0.68	0.73	0.81	0.63	0.86	0.84	0.75	0.63	0.6	0.73	0.72	0.77	0.64	0.86	0.82
F13	0.8	0.76	0.87	0.92	0.77	0.84	0.91	0.91	0.76	0.79	0.84	0.83	0.79	0.8	0.84	0.84
F24	0.6	0.56	0.83	0.83	0.64	0.47	0.78	0.69	0.53	0.41	0.78	0.68	0.53	0.66	0.65	0.77
F26	0.77	0.68	0.76	0.81	0.75	0.84	0.88	0.81	0.39	0.56	0.57	0.69	0.6	0.53	0.7	0.89
F28	0.59	0.49	0.53	0.29					0.7	0.68	0.48	0.27				
F29	0.77	0.67	0.58	0.54					0.78	0.79	0.6	0.3				
F30	0.88	0.77	0.97	0.96					0.61	0.72	0.94	0.96				
F31									1	1	0.99	1				
Temp1									0.63	0.41	0.78	0.5				
Temp2									0.83	0.87	0.87	0.83				
Temp3									0.82	0.87	0.8	0.91				
Temp4									1	0.95	0.95	0.97				

5.5 Four-sheet Forming Model

Introduction

A forming model is used to investigate the core sheets. These are a large area, free floating, superplastic structure that needs to form a complex and predictable pattern of interior vertical supports for two exterior sheets of titanium. The variability of this structure is much larger and less predictable than typical superplastic parts due to the stress-history dependence of superplasticity magnifying small variations in geometry to create measurable discrepancies in the finished part.

Superplastic forming is highly sensitive to material properties and processing conditions, specifically the temperature and strain rate applied. As a result, it can be difficult to determine the correct processing conditions to form a part. However, once the necessary process parameters

have been selected, it is a repeatable and accurate forming operation [80]. There can be some variation in thicknesses of formed sheets but these are mainly dependent on unavoidable variations in material properties and processing conditions such as slight changes in the microstructure and temperature. This is usually negligible. Single sheet and even the simple multi sheet designs have small variances between parts. While the geometric constraints on the face sheets are at both the edges and the surfaces of the die, the core sheet constraints are only at the edges where all four sheets are clamped. These inner sheets are therefore able to move freely with forces being balanced between the cells and can vary drastically more than a typical superplastically formed part. Due to the stress history dependence of superplasticity, slight variations in initial geometry can lead to large changes in formed geometry. The variability of this free floating core is the subject of this model, both to quantify it and to provide insight into how to cope with the variability in production setting.

The mechanical behavior of superplasticity is modeled in the form of constitutive equations governing the plastic behavior of the material. Several different forms of constitutive equations have been developed. These are developed from empirical forms and internal state variable forms. Empirically derived equations map the strain rate to the stress with constants derived from experiments. These have been developed for both single and multi phase materials. Examples of empirically derived equations use the extended power law creep equation [20]. Described in Equation 5.1, the constants K, along with the rate constants n and m, are all mapped to physical tests. Grain growth has been included in several models that add increased accuracy to the model [81]. Applications of crystal plasticity implemented in a self-consistent viscoplastic form have also been developed [82]. These equations contain the internal state variables as the dislocation density and the grain size. With these constitutive equations, finite element simulations of

superplasticity have been implemented extensively to model the complex geometry used in forming operations. There have been many fits of experimental data using both the hyperbolic sine viscoplastic models [83] and power law creep models. Extensive studies have been done on both of these models for titanium. Hyperbolic sine constitutive equations provide a wider range of strain rates for highly variable parts while the power law model does a sufficient job for a smaller range of strain rates [8]. These have had good correlation between formed parts and the production geometry provided that the material model can accurately simulate the range of processing conditions [84]. The implementation of all of these models is similar in that the stresses are assumed to be a function of the deformation tensor and the elastic stiffness tensor since the stresses are assumed to be below the yield point of the material.

$$\sigma = K\varepsilon^{n}\dot{\varepsilon}^{m} \tag{5.1}$$

Experimental Methods

To determine how controllable the core sheet formation is and where and how the variations in geometry of a produced part can occur, a finite element simulation was performed as well as extensive measurements on the panel geometry. The goal of the simulation was to provide an idealized case where the panel geometry would have minimal variations. The results of the simulation are compared with the geometry from production parts. While many panel parts were produced, only four panels were used in this analysis. Each of these panels, labeled F6-F9, were produced with the same material, Ti-6Al-4V fine grain by the same manufacturer, and at the same proprietary temperature and pressure schedule. By using the same material and process conditions one can demonstrate the variability inherent in the process. Data was collected on the major examples of variability in the core sheets: I) the column location and geometry along with II) the

geometry of the two different gas pathways. Thickness measurements were taken from mounted micrographs in detail and from cross sections of the cell geometry. Measurements of the two primary defects, the side gas connection holes as well as the upper gas pathways, were also taken. Variations in overall geometry were additionally measured.

The cores sheet formation simulation uses the implicit formulation of ABAQUS in combination with the power law creep model. This model only simulates the deformation of the superplastic core material. The deformation of the face sheets is small and mainly at the edges of the panel and is not a source of very significant thickness variations. The deformation of the weld nuggets is extremely small and entirely elastic. Therefore, only the upper half of a small unit cell of the core sheet superplastic material was simulated. The weld nuggets are modeled as half-ellipsoidal cutouts from the core sheet material. The edge weld nuggets and the center crossed weld nugget are shown in Figure 5.9B and are held as encastre. The interior weld nuggets are not allowed to deform in any way through the use of a kinematic coupling of the surface nodes to a reference point but are allowed to translate along their weld line toward or away from the center crossed weld nugget. This is an assumption, since the real weld nuggets are able to deform elastically. However, this deformation is small and can be neglected. The face sheet is modeled as a flat, analytically rigid contact surface with minimal friction. Periodic boundary conditions are applied to the edge nodes in the form of kinematic equations that tie the displacement of each edge node with the node on the opposite side of the mesh. It is assumed that there is symmetry on the z-axis at the weld nuggets (meaning the top half of each cell is identical to the bottom half), there is perfect gas distribution and pressure is applied according to a schedule equally everywhere, the material is perfectly isotropic in nature, and the welds are perfectly placed.

The material model for this simulation uses strain hardening based on the power law plasticity model. Material data was used from a tensile specimen elongated using the ASTM E2448-06 standard at a proprietary temperature and a representative strain rate. This material model is used instead of the time hardening model typically used because it has the advantage of being more consistent during large strains. The difference between strain and time hardening being that the superplastic response changes with deformation (strain) or time exposed to stress (time). The equation for the strain hardening material model is provided below (Equation 5.2). It relates the equivalent creep strain rate and equivalent creep strain to the deviatoric equivalent stress with some material constants.

$$\dot{\varepsilon}^{cr} = \left(A\, q^{-n}\left[(m+1)\bar{\varepsilon}^{cr}\right]^{m}\right)^{\frac{1}{m+1}} \tag{5.2}$$

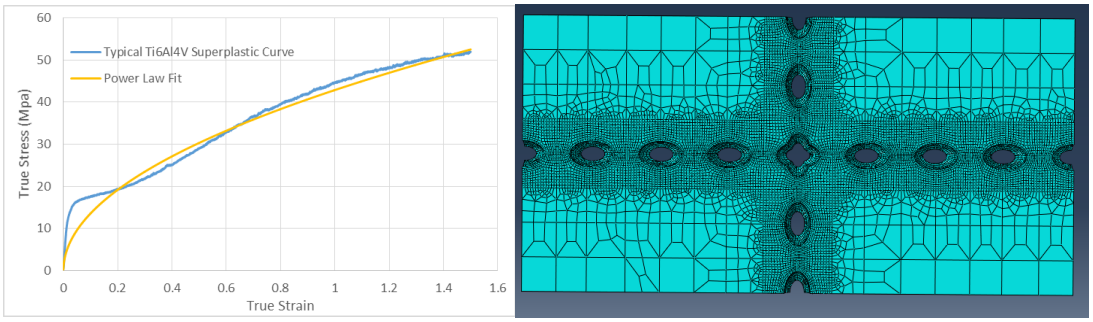


Figure 5.9 - Superplastic curve with power law fit (a) and finite element mesh (b)

To determine the effects of different errors on the major dimension of the final product, a series of two dimensional simulations were conducted. The material model is the time hardening model described for the 3D simulation. The mesh and boundary conditions of the model is different from the 3D model. There are no periodic boundary conditions. The boundary conditions are such that the edges form up to vertical rigid walls that represent the centerline between cells. The model is

built as a set of three cells each with varying dimensions. The cells are delineated by rigid weld nuggets that are able to translate and rotate but not deform. The cells form up to a horizontal boundary of customizable shape. This allows for various tests to be conducted. The tests conducted with this setup are to determine the trends but not specifics of the variations in cell dimensions, the curvature of the top and bottom of the part, and the effects of pressure distributions on the formed geometry. See Figure 5.10 for details.

Figure 5.10 – Diagram of the 2D forming model.

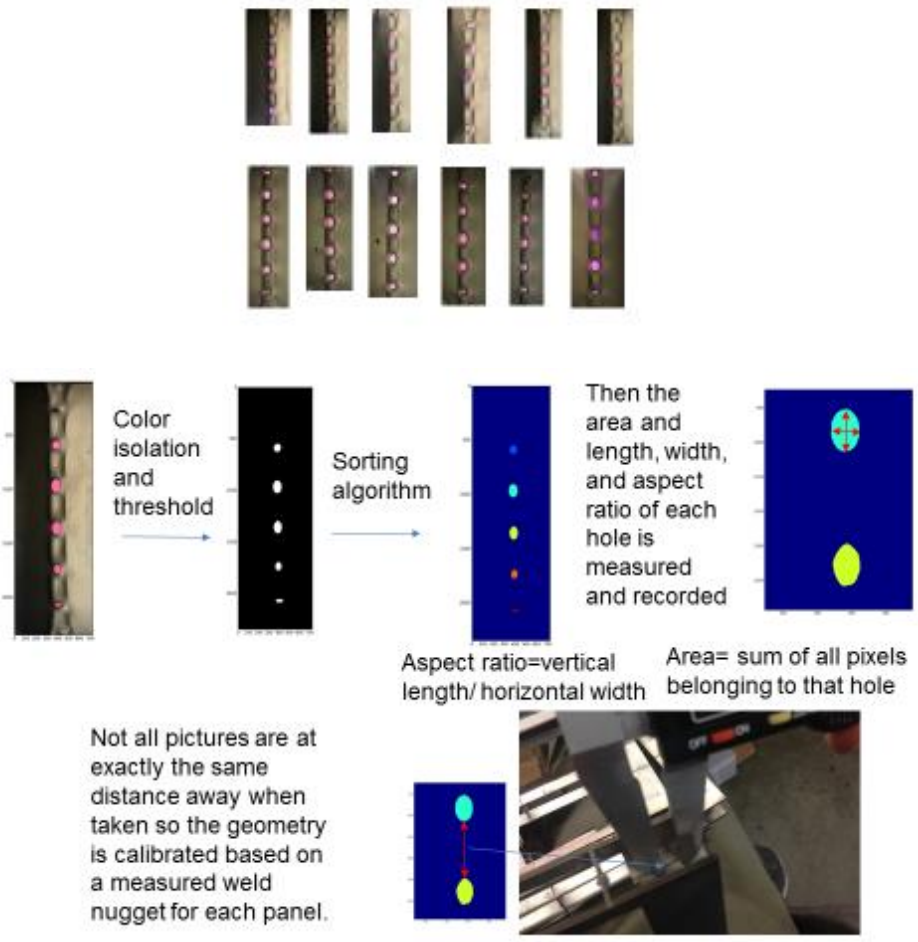


Figure 5.11 - Chroma key dataset (top) and an explanation of the algorithm (bottom).

Results

The results of the finite element simulation are highly symmetric and very uniform. The simulation shows very little of the variations that are present in the finished parts. There are some trends that are exhibited in the experimental results. These trends are detailed below as they are compared with the thickness distributions and the gas pathway statistics. Figure 5.12 shows the fully formed simulation. In the subsequent figures showing finite element results this section of the simulation has been mirrored across the nodes and replicated along the other axes.

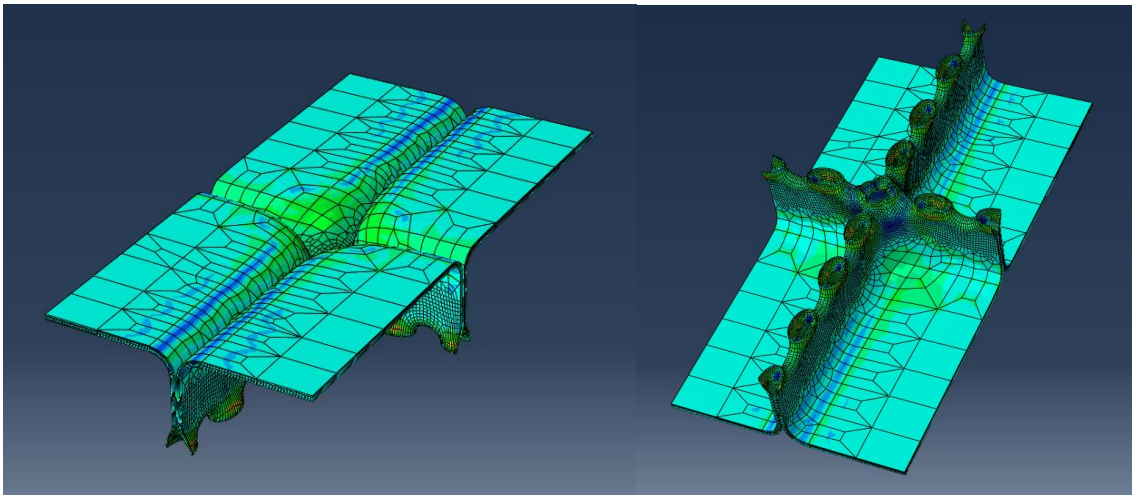


Figure 5.12 - Finite element simulation showing the top (left) contacting the face sheets and the internal weld nuggets (right)

Thinning of the columns is a major problem in this SPF/DB structure where a weakened column can provide a significant failure point. The thickness distributions developed from the edge detection of the mounted samples provide insight into the variance that can develop in the finished parts. The thicknesses were consolidated into two separated groups. There is a significant difference between the thickness that occur at the weld nuggets and the thicknesses that occur at the gas connections and therefore the thickness distributions are analyzed separately. The thickness distributions are best summarized by the plots in Figure 5.13 which show all the thicknesses measured. The extent of the variability can be seen visually. The thickness distributions were also computed for the finite element simulation for comparison. Thicknesses were taken at the midpoint at the center of the weld column and the center of the path column. Twenty thickness measurements were taken at the center column in both the weld nugget area and gas pathway with a standard deviation of 1.2%. This shows that with perfect geometry and conditions there shouldn't be any variation in the column thicknesses. However, as seen in Figure

5.13, there is a significant amount of variation in the column geometry. The analysis showed that there is not always an even stress distribution or perfect weld nugget placement. Length mismatch was also computed between the upper and lower portions of the columns. Ideally the length of both the top and bottom of a column should be equal placing the weld nugget or gas path directly in the center between the top and bottom of the part. This gives the minimum loading to weld nugget area which is beneficial due to the relatively large number of stress concentrations located there. Unequal lengths in the column also leads to unequal thicknesses and can lead to improper processing conditions at thinned sections in the column. The column length mismatch averaged 7.8% of the gas path columns and 9.8% of the weld nugget columns.

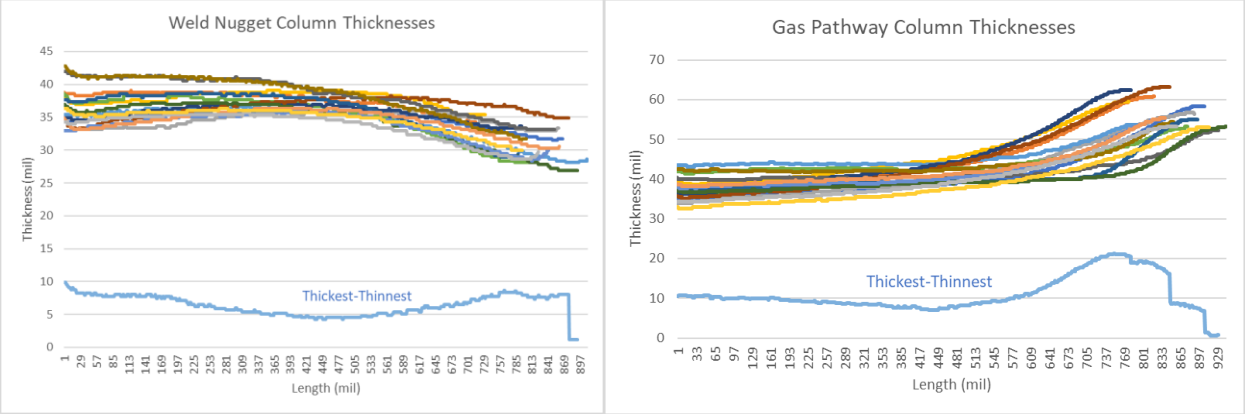
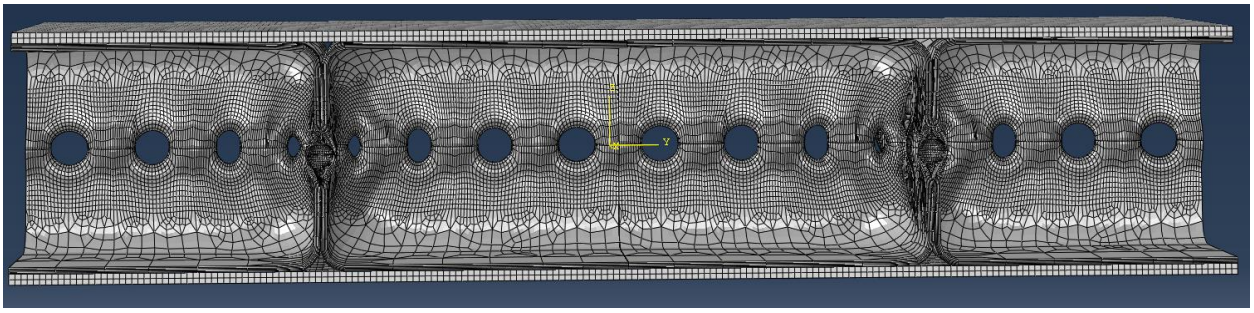


Figure 5.13 - Thickness plots of both the weld nugget columns and the gas pathway columns. There is a large difference between the thickest column vs the thinnest column. These columns support the face sheets and provide the dominant failure points for the finished part. Getting a column that is undersized can be detrimental to the structural capabilities of the part.

The gas pathways represent a major defect that is inherent to the core sheet formation. Gas must be able to transfer between the cells in some way but the regular defects can be problematic. They also can be highly inconsistent in size and shape depending on the local loading during forming.

This makes them a hard feature to predict. Structurally the gas pathway and weld nugget locations are not as important is the upper gas path and thickness variations in the columns near the face sheets since the stresses are minimized at the center of the sandwich during bending. The weld nugget placement is computer controlled and highly accurate in both distance between welds and weld quality.

The finite element simulation predicts some key features in the gas pathway geometry. The trend of large pathways in the center of the cell and horizontally shrunken pathways nearer to the ends of the cell are a result of the material transitioning toward the ends to form the perpendicular columns. The weld nuggets translate and as a result distort and sometimes close off the gas pathways at the ends of the cell. The gas pathway dimensions from image analysis have an average of the 2.055mm^2 with a deviation of 1mm^2. The finite element simulation had gas pathways that were significantly larger with an average area of 5.24mm^2 and a similar relative deviation (2.33mm^2).



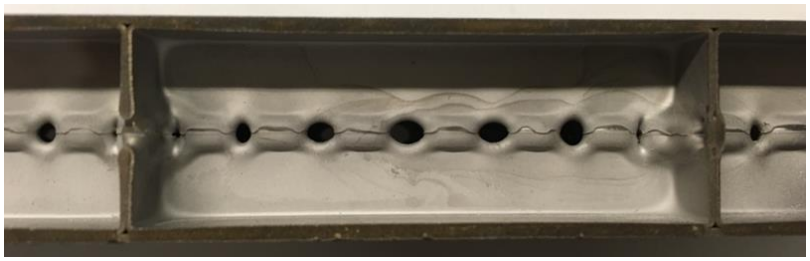


Figure 5.14 - FEM results of the gas pathway geometry (top) and real geometry (bottom). The center of the cell has the largest gas pathways which become more distorted as they reach the edges of the cell. However, perfect gas path geometry is uncommon.

Cell wandering, the process of the column locations not lining up with the initial weld nugget placement, is also an issue in these SPF/DB structures. The variations in the cell size are a natural effect of the stress history dependence of superplasticity. Ideally the cell widths should have a value of 50.8mm and in the finite element simulation they do. Despite excellent weld placement there is still a variability in the cell widths. The cells wander with a variation of 1.7% their average length as shown in Figure 5.15.

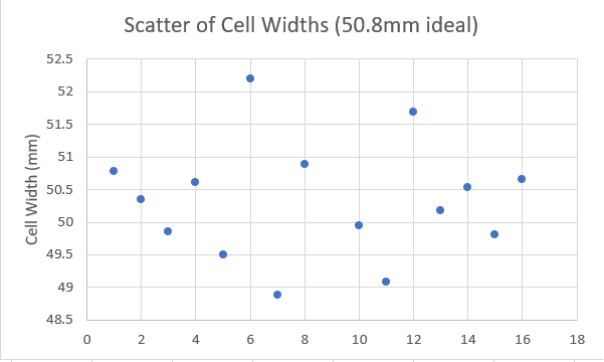


Figure 5.15 - Cell width scatter

The effects of upper gas pathway dimensions are arguably the most important characteristic of all the geometric variations in the core sheet. The location of the upper gas pathways relative to the rest of the geometry are shown in Figure 5.16 and the trend in pathway size is shown in Figure 5.17. This is because the triangular pathway is directly at diffusion bonds in both the column and the core-face sheet interface. It is also a major stress path in both compression and bending. Especially in bending, where the stresses are maximized at the upper and lower surface of the part,

failure always occurs at the largest upper gas pathway. The size of the upper gas path in the finite element simulation only varied at the crossed weld nugget area. This is because of the periodic boundary conditions in the simulation that give a perfectly uniform upper gas path. In the production panel however, they are not perfectly uniform. The trend was to have the smallest pathways nearer to the gas inlet that get larger as a function of distance from the gas inlet. This is because the material must deform plastically to transmit gasses and the gas inlet is on one side giving the area closest to the gas inlet more time to form than the material far away from the inlet. This has the possible effect of having the failure point be a function of the gas inlet location depending on how the part is loaded.

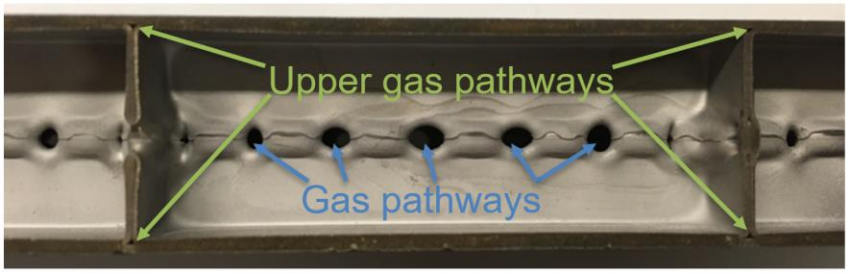


Figure 5.16 – Gas pathway and upper gas pathway locations

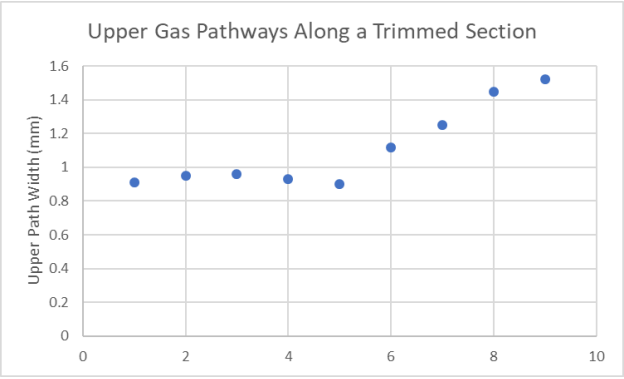


Figure 5.17 - Upper gas pathway trend

The results of the 2D finite element simulations are presented in the discussion. The total change in geometry is computed by using the displacement of the weld nugget locations. This provides both a measure of the changes cell geometry but also of the changes in load carrying capability. The results below should be interpreted as the general effects of the displacements and not the actual displacements since they are in 2D not in 3D. The results are that the weld nugget location is affected by the different errors in geometry in different ways. The effects of an offset in weld pattern are to exaggerate that error in the position of the weld nugget. This effect increases with increasing error in weld nugget placement. The effects of curvature in the part dimensions are to induce and increase the error of the weld nuggets as well. This means that there will be translation in the weld nuggets just simply due to the part curvature without any error in placement. If there is error, then it will be either exaggerated or in a few cases corrected. The change in error depends on the size of the cells relative to adjacent cells and the curvature of the die that the part is forming up to.

Discussion

The purpose of this research was to highlight the unique challenges in producing a consistent part when using an edge supported internal superplastic structure, quantify the extent of the variability in such a structure, and provide insight into possible ways to mitigate problems that could arise. Superplastic forming is a stress history dependent process as is all plasticity. A load causes a change in dimensions of a material which can in turn create larger loads. The difficulties arise in the four-sheet process due to the fact that the internal structures are load balanced and not geometrically constrained. Imperfect geometry causes changes in loading and changes in forming over time. To illustrate this mechanism, an analysis of cell wander is performed using a two dimensional simulation. The simulation has four weld nugget areas encompassing three cells

along the smaller cross section of the rectangular cell (25.4mm between cells). The center cell has displaced weld nuggets relative to the outer cells. Both a flat panel and a panel with an expanded center section were analyzed for both normal precise weld nugget placement and an error in weld nugget placement. The flat panel, with perfect geometry, has perfectly balanced loads on the weld nugget and no increase in error. When the weld nuggets are moved closer together in the center than they should be the error ends up being increased. This is because the outer cells end up forming quicker since they have a larger surface area exposed to the forming pressure and therefore a larger stress. Initially the tension in the larger cells creates a slight tension in the small center cell that expands it decreasing the error. Soon however, the outer cells balloon out first and contact the face sheet first and this creates a horizontal compression of the center cell since it is smaller and exhibits a smaller horizontal force on the weld nuggets. The weld nuggets relocate to compensate for this. By the time the columns are half formed everything has balanced out leaving the center cell even smaller than the weld nugget placement. The 1.27mm (5%) error in weld nugget placement becomes 2.62mm (10.3%) after forming. For the angled panel, the center cell is taller than the outer cells and therefore always horizontally expands compressing the outer cells. The offset weld nuggets are 1.27mm (5%) further apart than desired and this ends up expanding the cell by 4.37mm (17.2%). The simulation layout and error as a function of forming time is shown in Figure 5.18.

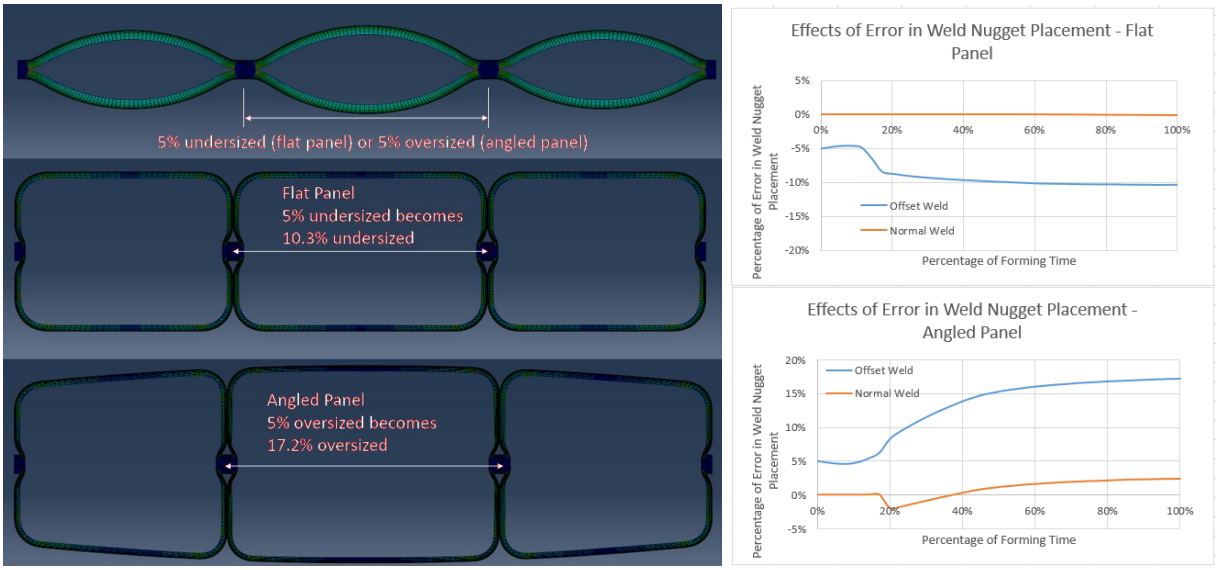


Figure 5.18 - Three short (25.4 mm) cell formation in 2D with an undersized/oversized center cell (top left). The flat panel (mid left) causes an increase in weld error while the angled panel (bottom left) has a tendency to expand the center increasing the error if the welds are oversized but decreasing if undersized. Tracking distance between the center weld nuggets produces the error figure (right) which shows that the predominant source of error occurs near the beginning of the forming cycle where the partially formed cells are significantly different in size.

This cell wandering, as well as other geometry defects, is exacerbated in any case where there is uneven loading in the cells. Small defects in material properties, temperature gradients, providing gas pressure from one side of the part, and even gravity can all have effects on the final formed geometry. As shown above, while flat panels simply increase error, panels with features such as the angle panel are going to have different behavior. Flat panels are not the only shape that the four-sheet process is meant for. It can be molded into a variety of complex shapes for aerospace structures.

A complex die surface can also increase the variation in cell geometry by changing the loading conditions in each cell. Migration in the cells of a production part full of complex contours are going to be significantly more variable than a flat panel. There are also several techniques that can mitigate some of the variation issues. The weight savings by removing fasteners and making a monolithic part can be used increase the thickness and robustness of the SPF/DB structure. Precision placement of the core sheet welds is an excellent additional technique to prevent changes in geometry. Weld placement can be used to compensate for the complex curvatures that could cause cell wandering and thinning.

Summary

Superplasticity when combined with diffusion bonding is an elegant way to produce high strength metal sandwich parts. This research has examined a SPF/DB four-sheet design that has a nearly unsupported internal reinforcement structure. The lack of constraints and the extent of the forming strain can cause large variability in the formed geometry. The internal sheet forming was simulated with the finite element method and compared with the variation in geometry to show that there is significantly larger variability in the produced part. This was explained as the effect the large strains combined with superplasticity, a stress history dependent process, which serves to amplify small variations in initial geometry. The effects of this variability can be mitigated by more precise initial geometry, better control over material properties, and recognizing that certain areas of a part are going to have significantly more variation than others. These techniques can be applied to any large plasticity cases where there is loosely constrained forming.

6 Modeling of Diffusion Bonding and Superplasticity

6.1 Introduction

A primary goal for this dissertation is to model the SPF/DB processes and investigate some of the physics behind the processes using simulations. Three different diffusion bonding models and work on a superplastic model are presented in this chapter. The first bonding model was an expansion on a conventional DB model. It's a surface overlap model that uses Pilling's equations with a stochastic input for the void geometry. This model resulted in poor fit with experimental data. An expansion of the model used ABAQUS to calculate the initial geometry in an elastic-plastic contact model. A second model was produced to expand the complexity of the input surface. The Surface Tracking Model uses conventional bonding physics and equations but was implemented with a discretized surface. This allows the surface to develop naturally without any restrictions on the void. This resulted in a decent fit with experimental data but only with an increase in the creep exponent. Finally, a phase field diffusion bonding model is presented. This model uses the correct material constants based on diffusion studies. It uses phase energies from CALPHAD calculation as well as 2D geometry derived from micrographs. The phase field equations are well established for solving these types of problems and result in a model that accurately computes the remaining time to bond. Several more interesting effects on bonding can be computed by looking at the influence of stress on bonding and the influence of crystal texture on bonding. In addition to the diffusion bonding models, the beginnings of a superplastic model is presented that uses phase field and grain boundary sliding.

6.2 Surface Roughness Overlap with Pilling Model

The most important characteristic of diffusion bonding models is their accuracy and ability to predict bonding process parameters. While most of the model discussed in the literature make some comparison with experimental data, the models present their results in different ways and are for different materials and processing conditions making a direct comparison between models challenging. The models dealing with titanium bonding, the subject of this study, compare with experimental results without micrographic correlation. They use results from structural shear tests and compare the shear strength in binary form (bonded or not bonded for a processing condition for above or below 95% shear strength) to the calculated bonding time. However, there is no set standard for structural testing the shear strength in diffusion bonding. Several papers that have discussed different methods have pointed to the discrepancies in the testing and lack of ability to distinguish between higher bonding percentages [73]. It is therefore debatable that the results were accurate. More importantly, these models are not modeling the shear strength or structural characteristics of the void but instead just modeling void closure. The figure below presents a summary of all the different models that are compared with Pilling's experimental data. Included in the figure are the results from Li's stochastic model [24].

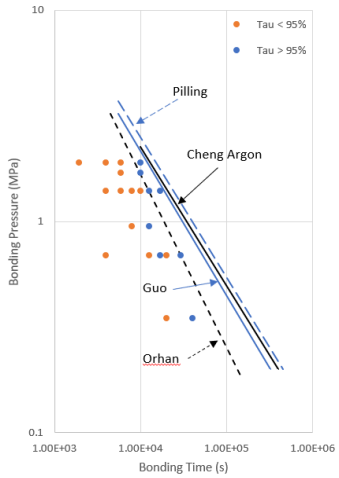


Figure 6.1 - Bonding models compared with experimental results

Experimental Methods

The model presented uses a stochastic formulation similar to Kulkarni, et al.[25] for dissimilar diffusion bonding with the inclusion of different initial conditions. Surface roughness profiles are measured then interacted with each other to determine the initial geometries of the voids followed by developing a statistical distribution based on them. This idea of overlapping the surfaces to determine the potential voids and then running statistics on them was pioneered by Kulkarni. The model in this paper provides a much more robust way of determining the potential voids. The distribution of voids are applied to the equations derived by Pilling in a stochastic manner in the form of a Monte Carlo simulation. The surface roughness dataset is comprised of six different surface roughness profiles for each of the three materials for a total of 18 profiles. The materials involved are TI-54M, Ti64 standard grain, and Ti-6242S. The profiles are taken in random locations with three in the longitudinal direction and three in the transverse direction for each material that was prepared as it would be for diffusion bonding. The profiles were generated using

a Mahr XR20 MIT GD25 contact surface profilometer. The raw data (Figure 6.2) for each profile was aligned horizontally by subtracting a single linear fit of the data from the profile not the series of linear fits commonly used in surface roughness measurements.

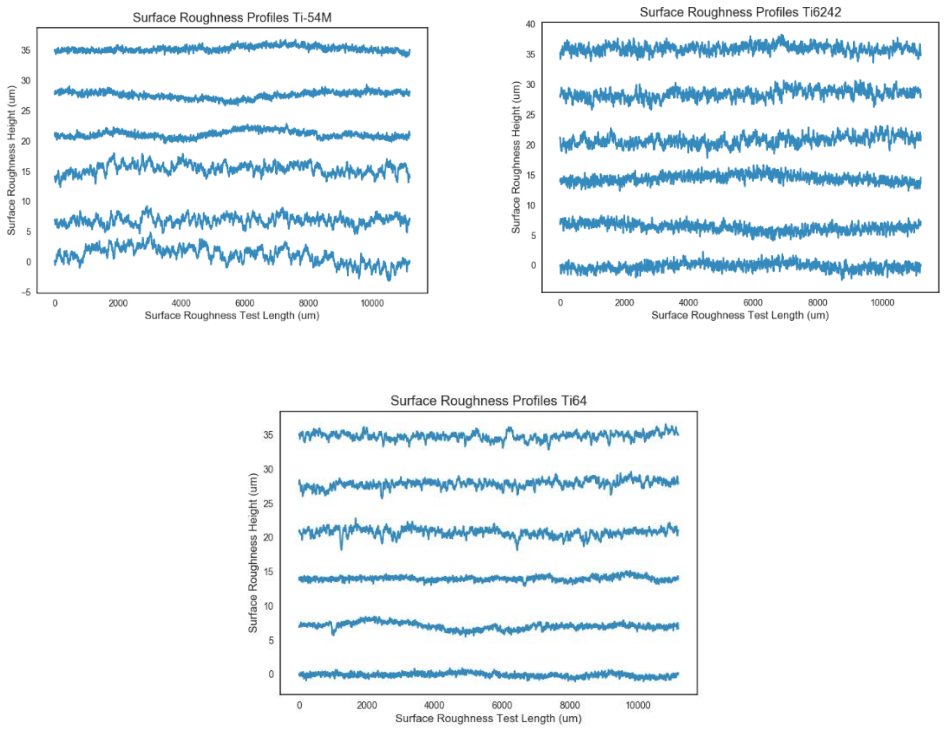


Figure 6.2 - Surface roughness profiles from Ti54m (top left), Ti6242S (top right), and Ti64 (bottom)

The initial conditions for the diffusion bonding model are based on the interaction of the two surfaces. The peaks of the surfaces contact and observe very high stresses causing them to go through an initial plastic deformation and flatten out. This occurs until the flow stress reaches the superplastic region. In the current model, the reduction in void height is assumed to be a simple overlap of the surface roughness profiles and that the initial plasticity of the peaks results in

negligible closure of the voids. This initial collapse provides very little closure to begin with but serves to define the initial void geometry which has a large effect on the overall bonding time.

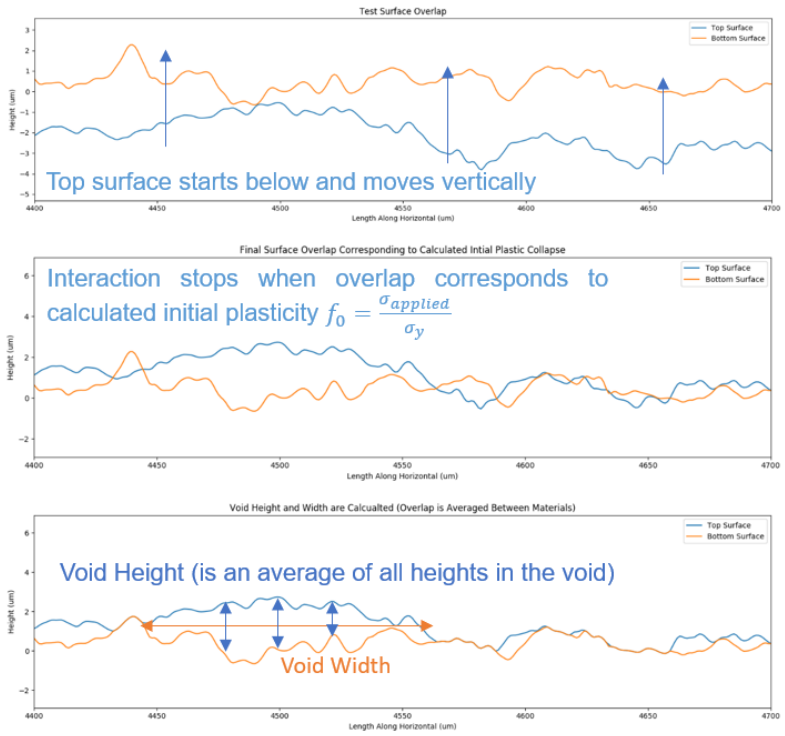


Figure 6.3 – The surface profiles are overlapped to the point where they correspond. A calculated void is the average thickness over the specified width.

The surface profiles are moved into contact with each other until the overlap is at the same amount as specified in the Pilling model (the applied pressure over the yield strength of the material). For each of the interactions between materials, one surface is translated relative to each. This is because there is no way of knowing exactly how the surface profiles will be mated in a production setting. This translation is repeated ten times in equal increments. Therefore, a total of 360 surface profile interactions per material combination are used to develop statistics on the void height and width.

Figure 6.3 describes the algorithm procedure which entails flipping one of the profiles, translating it horizontally a given amount, overlapping it with the other profile until the contact stress would equal the yield strength, then determining the void height and width. The width of the void is the inside distance between each contact point. The height of the void is the average of the overlap

height between the contact points. These are used for the basis for Pilling's geometry in the stochastic model.

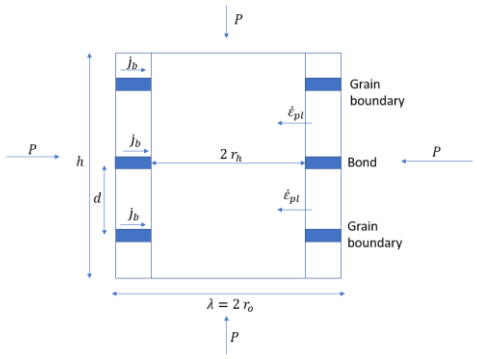


Figure 6.4 – The Pilling void geometry is a revolved cylinder that has the axis of revolution along the bond plane normal vector.

The computed theoretical voids are tabulated and a distribution is fitted from the data. Many different statistical distributions were tried and the exponential distribution provided the best fit for both the height and the width of the voids. The void width had some significant outliers in the distribution where the surfaces randomly aligned to create an extremely large void. Void widths larger than the distribution fit maximum of 200µm occurred around 7.5% of the time and were assumed to be negligible on the overall bonding performance. These large voids are most likely voids on the boundary of the surface profiles and would not occur in practice. An example of the distributions is presented in Figure 6.5.

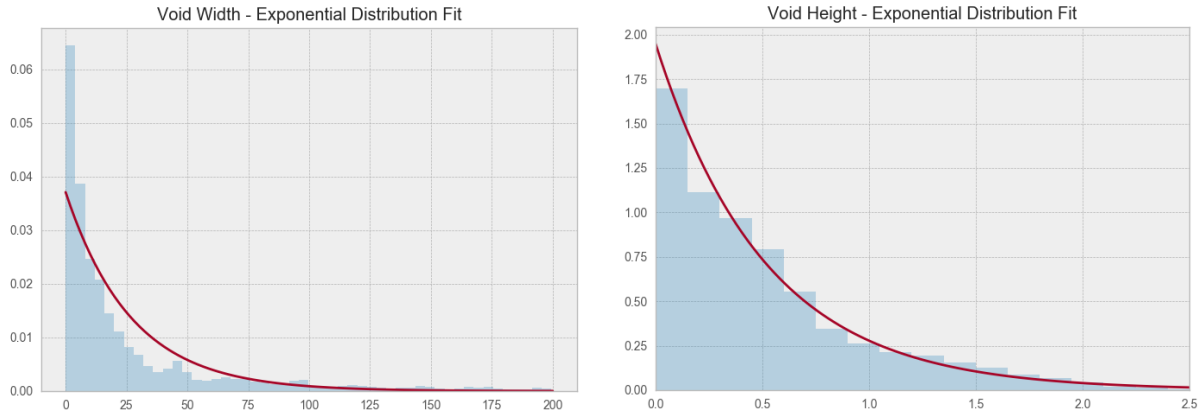


Figure 6.5 - Example histogram of voids and distributions (Ti54m/Ti54m at LT-LP)

The initial void geometry distributions are used as an input for a stochastic Pilling Model (Equation 6.7). The primary equations were developed by Pilling and are for a specific void height and width. Using the probability distributions highlighted above, a random height and random width are selected. Then Pilling's equations are applied and the time required to completely bond is determined. The random selection of heights and widths and calculation of bonding is repeated 1000 times. A cumulative distribution is developed from the data to determine the percentage of randomly selected voids that would be closed after a given time for a set of processing conditions.

$$f_0 = \frac{p}{\sigma_y} \tag{6-1}$$

$$\left[\frac{df}{dt}\right]_{vol} = -\frac{4\Omega D_v}{kT}\left[\frac{f}{ln\left(\frac{1}{f}\right)-\frac{1-f}{2}}\right]\frac{1}{r_o^2}p \tag{6-2}$$

$$\left[\frac{df}{dt}\right]_{gb} = -\frac{2\Omega D_{gb}\delta}{kT}\left[\frac{1}{ln\left(\frac{1}{f}\right)-\frac{1-f}{2}}\right]\frac{1}{r_o^2}p\left(\frac{1+\frac{h}{d}}{h}\right) \tag{6-3}$$

$$\dot{\varepsilon} = \frac{A\, exp^{\left(-\frac{Q_{sp}}{RT}\right)}\, \sigma^n}{T} \tag{6-4}$$

$$\left[\frac{df}{dt}\right]_{plasticity} = \frac{-\dot{\varepsilon}}{\sigma_e}\left(3p\sqrt{f} - \frac{2\gamma\sqrt{f}}{r_0} + \frac{6\gamma}{r_0}\right) \tag{6-5}$$

$$t = \int_{f_0}^{f} \frac{-\sigma_e/\dot{\varepsilon}}{\left(-\frac{\dot{\varepsilon}}{\sigma_e}3p\sqrt{f} - \frac{2\gamma\sqrt{f}}{r_0} + \frac{6\gamma}{r_0}\right) + \frac{4\Omega D_v}{kT}\left[\frac{f}{\ln\left(\frac{1}{f}\right) - \frac{1-f}{2}}\right]\frac{1}{r_0^2}p + \frac{2\Omega D_{gb}\delta}{kT}\left[\frac{1}{\ln\left(\frac{1}{f}\right) - \frac{1-f}{2}}\right]\frac{1}{r_0^2}p\left(\frac{1+\frac{h}{d}}{h}\right)} df \tag{6-6}$$

$$f(x;\mu,\lambda) = \left[\frac{\lambda}{2\pi x^3}\right] exp\, \frac{-\lambda(x-\mu)^2}{2\mu^2 x} \quad (\text{for } x > 0, \mu > 0 \text{ and } \lambda > 0) \tag{6-7}$$

For the equations above, the material constants are defined in Table 6-1 and the geometric variables are defined in Figure 6.4. The initial bonding fraction, f_0, is determined by the applied bonding pressure, p, divided by the high temperature yield strength, σ_y, of the material (Equation 6.1). The volume diffusion equation, $\left[\frac{df}{dt}\right]_{vol}$, and the grain boundary diffusion equation, $\left[\frac{df}{dt}\right]_{gb}$, are both derived from the stress dependent chemical potential due to the applied pressure (Equations 6.2 and 6.3). The plastic deformation from creep, $\left[\frac{df}{dt}\right]_{plasticity}$, is modeled as deformation from an isostatic pressure with the creep equation which relates the strain rate, $\dot{\varepsilon}$, to the bonding stress, σ (Equations 6.4 and 6.5). The total change is bonded fraction per time is integrated to determine the bonding time t. In a numerical implementation, the bonded fraction is incremented from initial to complete closure while the height is kept uniform by applying the equations above. For each increment in bonded fraction the change in time is computed. The bonding time for each random void is a summation of time increments. The material properties are based on the rule of mixtures using the alpha/beta ratio for each material. A flowchart of the simulation is provided in Figure 6.

Table 6-1 – Titanium material properties and variables for the Pilling equations [17]

Properties	α	β
Atomic Volume "Ω" (m^3)	1.76×10^{-29}	1.81×10^{-29}
Burgers vector "b" (m)	2.9×10^{-10}	2.9×10^{-10}
Surface energy "γ" ($J\ m^2$)	1.7	1.7
Grain boundary diffusivity "$D_{gb}\,\delta$"	$3.6 \times 10^{-16} \times \exp(-97000/(R \times T))$	$5.4 \times 10^{-17} \times \exp(-153000/(R \times T))$
Grain boundary width "δ" (m)	5.9×10^{-10}	5.72×10^{-10}
Volume boundary diffusivity "D_v"	$8.6 \times 10^{-10} \times \exp(-150000/(R \times T))$	$1.9 \times 10^{-7} \times \exp(-153000/(R \times T))$
Shear modulus "G" (MPa)	$4.3 \times 10^4 \times (1-((T-300)/T_m) \times 1.2)$	$2.05 \times 10^4 \times (1-((T-300)/T_m) \times 0.5)$
Creep constant "A"	1.2×10^{-9}	1.2×10^{-9}
Strain rate creep constant "A_c"	$A \times (D_{gb}\,\delta)_\alpha \times G_\alpha/(k \times T)$	$A \times (D_{gb}\,\delta)_\beta \times G_\beta/(k \times T)$
Gas constant "R" (J/mol K)	8.314	
Boltzmann's constant "k" (J/K)	1.38×10^{-23}	
Melting temperature "T_m" (K)	1973	
Bonding temperature "T" (K)	1046-1102	
Bonding pressure "P" (MPa)	1-3	

The simulation is compared with experimental data from TIMET [85] and the exact temperatures and pressure are not disclosed due to proprietary reasons. The pressures are labeled as low pressure (LP), intermediate pressure (IP), and high pressure (HP) while the temperatures are labeled as low temperature (LT), intermediate temperature (IT), and high temperature (HT). The grains sizes of the various materials are 2.4µm for Ti54M, 5.9µm for Ti64, and 4.1µm for Ti6242S [85]. The experimental results are summarized in Table 6-2 and are estimated from plots in the reference. The material properties for each of the calculations are show in Table 6-1. The alpha to beta ratio depends heavily on the heat treatment of the alloys and is unknown for this data. The ratio is assumed to be 0.49 alpha for Ti64 [86], 0.42 alpha for Ti54M [87], and 0.33 alpha for Ti6242 [88]. For every material/temperature/pressure combination the differences in alpha/beta ratio generates a unique set of voids to be analyzed (for a total of 54 unique sets).

Table 6-2 – Experimental results [85]

Materials	LT-LP	LT-IP	LT-HP	IT-LP	IT-IP	IT-HP	HT-LP	HT-IP	HT-HP
54m/54m	85%	95%	98%	100%	100%	100%	100%	100%	100%
64/64	51%	52%	53%	54%	81%	98%	91%	98%	99%
6242/6242	5%	8%	16%	37%	39%	49%	50%	59%	83%
54m/64	82%	95%	98%	100%	100%	100%	100%	100%	100%
54m/6242	44%	75%	81%	90%	95%	97%	98%	99%	100%
64/6242	17%	38%	47%	51%	60%	70%	70%	77%	97%

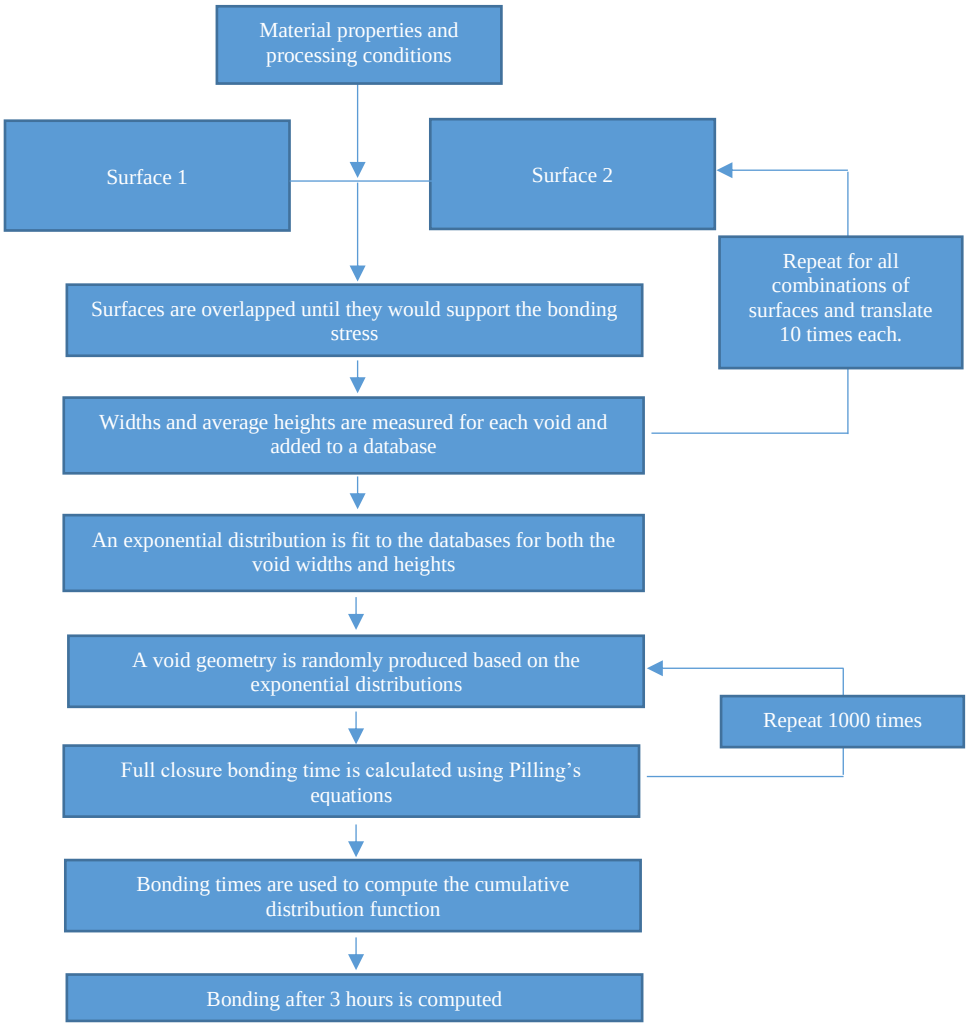


Figure 6.6 - Program flowchart performed for each material combination and processing condition.

As discussed above, none of the models provided a perfectly accurate fit to the experimental data and many of them cite specifically the lack of accurate surface geometry as a reason for this discrepancy. With the robust way of statistically incorporating the surface geometry presented in this paper the surface effects should be well represented. However, as with the other bonding models listed in the background, there are still some differences between the predicted bonding times and the experimental ones. The results of the simulation with respect to the experimental results are presented in Figure 6.7. The high temperature – high pressure group was the most accurate of the processing conditions. However, some of the lower temperature and lower pressure groups had significant inaccuracies. Figure 6.8 shows the average percentage discrepancy between experiments and calculations with respect to the temperature and pressure. The trend is fairly linearly toward high pressure high temperature accuracy and low pressure low temperature inaccuracy.

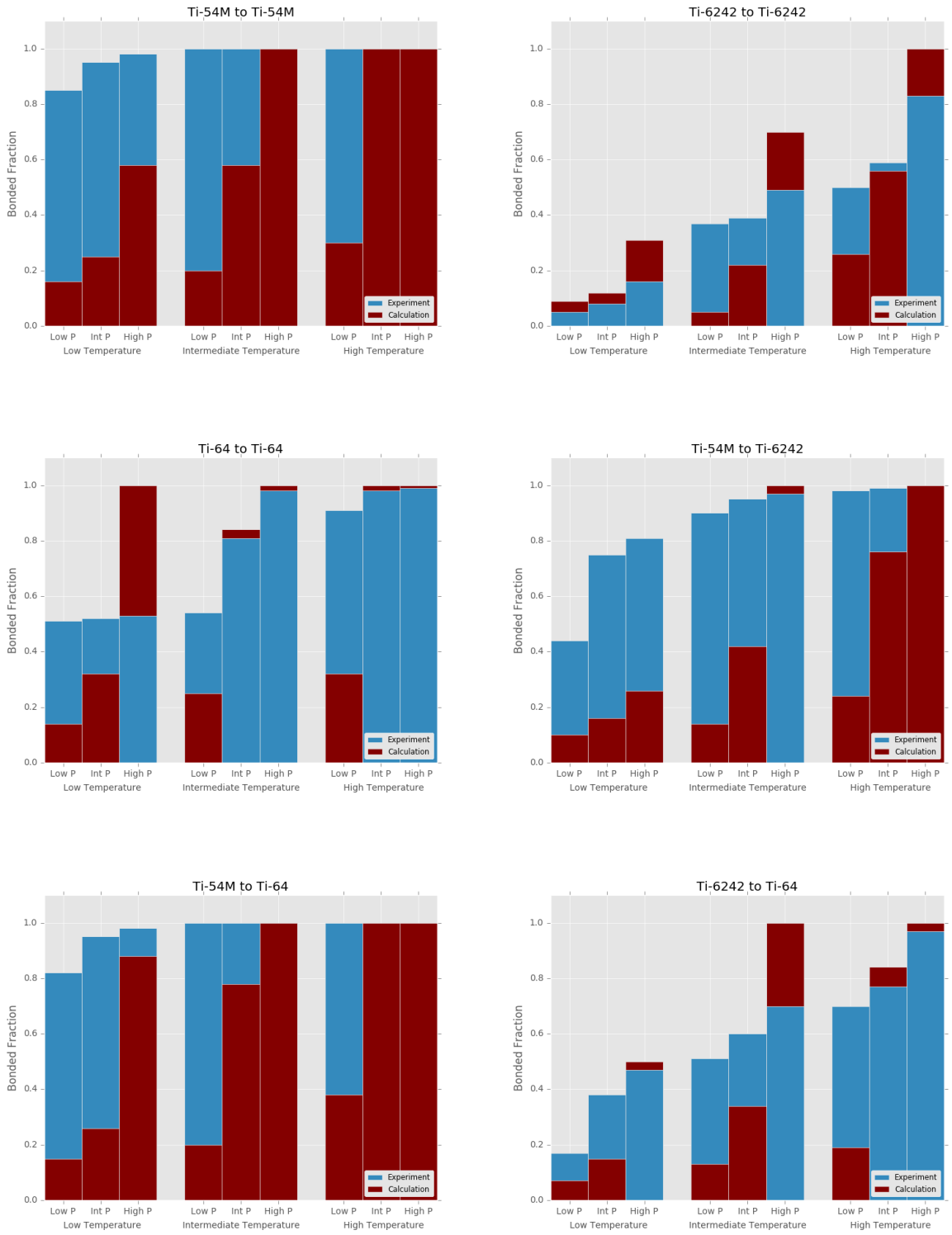


Figure 6.7 - Results from numerical analysis in comparison with experiments

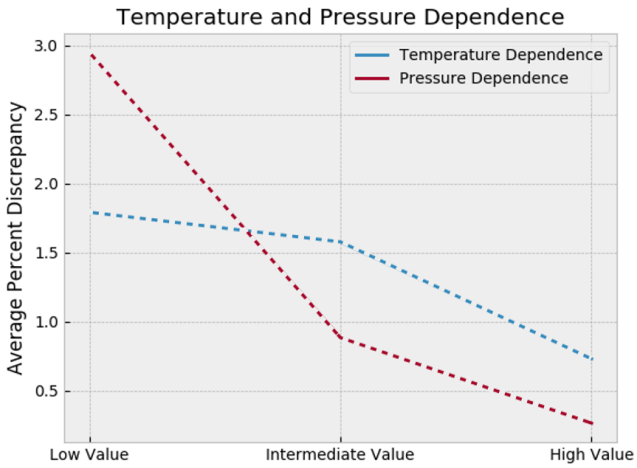
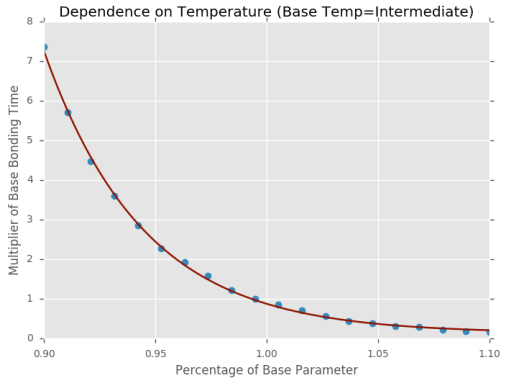
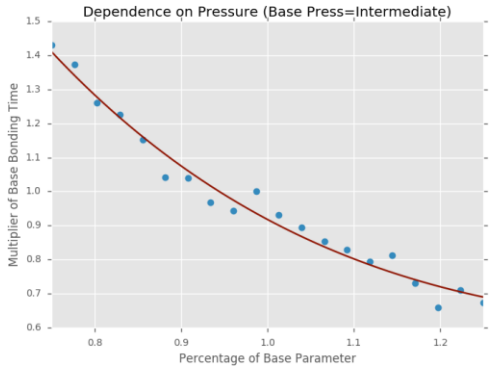


Figure 6.8 – Average discrepancy between experimental data and calculated results by temperature and pressure. The high temperature and high pressure had the most accurate results.

While the theoretical aspects of a model determine its effectiveness, accurate material properties are also of paramount importance and it is sometimes difficult to separate phenomenological issues with the theoretical model from discrepancies in the material properties. Specifically in this case where the diffusion constants are based on the work of Ashby [89], which is a compilation of previous diffusion and deformation research from the 1960s-1980s, and meant for commercially pure titanium and not the alloys in this research. These material properties, cited in previous titanium models, separate the phases into alpha and beta and have diffusion mobilities based on those two phases. Modern methods use the CALPHAD-type coupling of the thermodynamic and kinetic properties of a material to compute the diffusional mobilities making the diffusion of each alloying element a function of the Gibbs free energy and element concentrations [90, 91]. Implementation of this would require more complex modeling of the different energies associated with the bonding process and is beyond the scope of this current model. Therefore, some of the material properties used here and in other diffusion bonding research are most likely incorrect for

these specific alloys despite some accurate results at high temperature and pressure. To investigate how each material property is affecting the results of this simulation, a study is conducted. The effects of each parameter are calculated by using the base material properties in Table 6.1 along with the intermediate temperature and pressure and surface roughness constants for Ti-54M to Ti-54M at those processing conditions. By keeping everything else in the model constant and varying a single parameter, the relative change in complete bonding time is computed. For example, the temperature dependence figure below (Figure 6.9) shows that if you reduce the temperature to 90% of the intermediate temperature, with all else being constant, the time required to obtain a fully bonded sample will increase to seven times what it would be at the intermediate temperature. The relative effects of each influential material property and processing condition is presented in similar fashion.





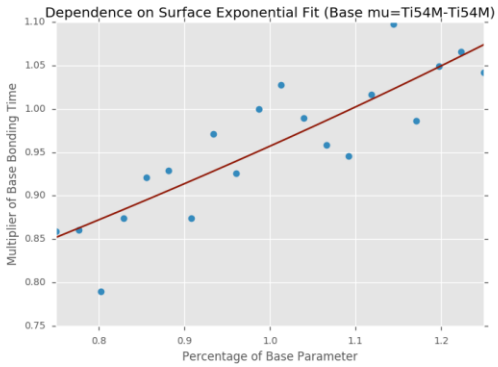


Figure 6.9 - Dependencies on temperature, pressure, and surface parameter.

The primary processing conditions all have a strong effect on the bonding performance in the model and are in line with expectations. Most significant is the temperature which softens the material for better closure during initial contact, increases the different diffusion constants, and increases the creep strain as the temperature increases providing a lower bonding time. A higher pressure provides more stress for power law creep and diffusion which reduces bonding time. As the surface parameter reduces, representing a distribution of smaller voids, the bonding time decreases while a larger surface parameter takes longer to close due to the larger voids. The effect of the surface is less extensive than the effects of temperature and pressure which is to be expected.

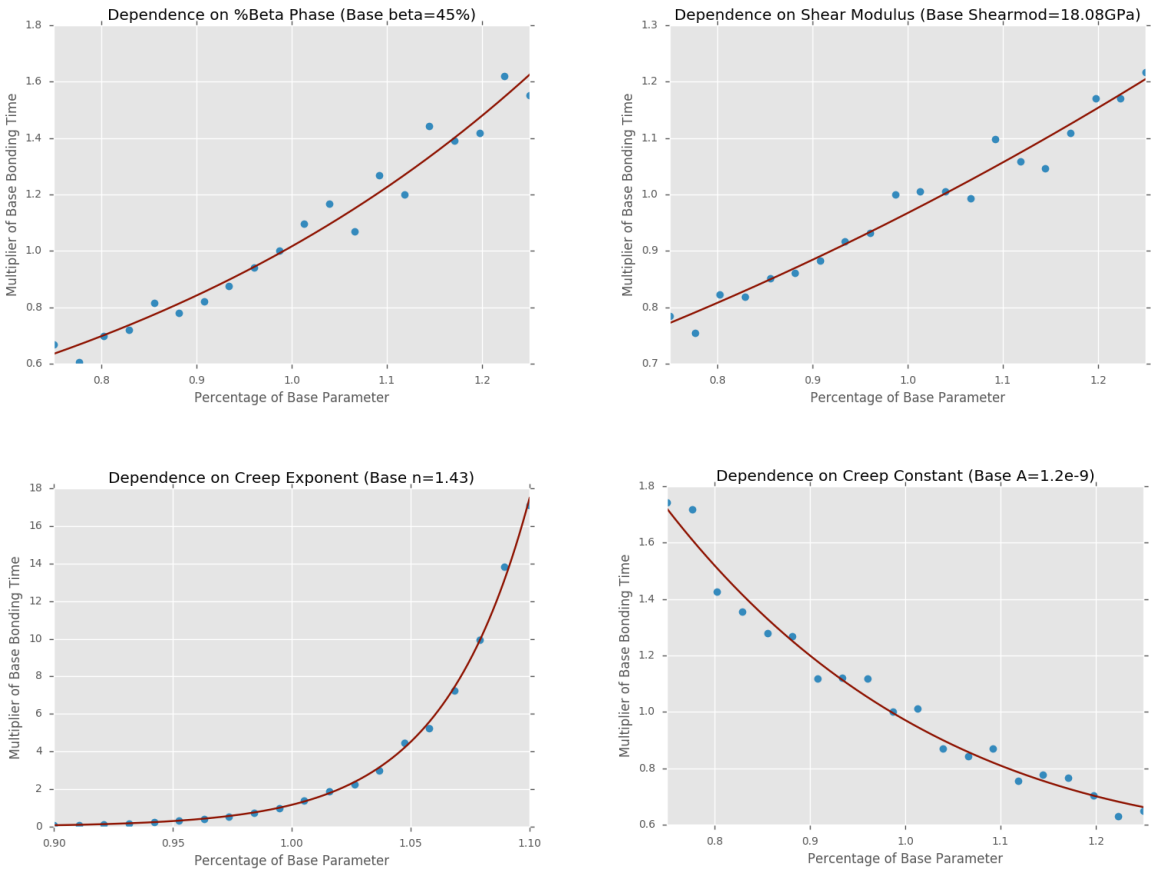


Figure 6.10 - Dependence on alpha/beta ratio, the shear modulus, and the creep material constants.

The material constants for the beta phase are less conducive to bonding than the alpha phase. The amount of alpha and beta will change depending on processing and heat treatment. Since the exact heat treatments were unknown for these materials, and the beta % was unknown, the assumed alpha/beta ratio could be a source of inaccuracy. For those papers that have investigated the rates of the different bonding mechanisms, creep has been the dominant bonding mechanism [11, 15, 17, 92]. Pilling's model is no exception and the bonding time is significantly affected by the creep material constants as shown in Figure 6.10. There is some debate as to whether the mechanism of plastic collapse is due to power law creep (as in Derby and Hill's pressure sintering models) or if it is due to superplastic collapse (Hamilton and Pilling's models). While the equations are similar, the creep exponent is three times larger for power law creep versus superplastic deformation. There are many additional constants that were checked but have no strong influence on the bonding time. Most notably is the grain size, which had no discernable influence, but is additionally incorporated into this version of Pilling's model and Hill's model.

A variation of parameters could be performed to determine exactly which combination of modified material properties allows this model to fit the experimental data accurately. However, this is unphysical and does not demonstrate the effectiveness of the model which is the point of this research. However, since creep is the dominant closure mechanism, the creep exponent is the most influential material property, the exact value of the exponent is unknown for these alloys, and there is some debate on the exponent value anyway it is sufficient to modify this parameter alone to demonstrate how sensitive the results are to this material property. By lowering the creep exponent slightly more toward diffusive creep, Coble creep and Nabarro-Herring creep, the accuracy of the model significantly increases especially in the lower temperature range. The changes in accuracy of the model simply by a 0.025, .05, and .075 reduction in creep exponent

corresponds to an decrease in the average discrepancy between experiments and simulation from 28.32% to 23.0%, 19.5%, and 19.2% respectively. This demonstrates how sensitive these material properties can be and how measuring them on the specific alloys modeled would be necessary for an accurate diffusion bonding model.

Summary

The stochastic Pilling model was computed with significantly more robust initial conditions computed through interactions of surface profiles. A database of potential voids was produced by overlapping several different surface roughness profiles in multiple ways to the overlap specified by the initial conditions of the diffusion bonding model. Pilling's model was then used to determine the statistical bonding percentage for the bonding time. The model predicted the high temperature-high pressure group very well but was less accurate for other cases. An investigation into the effects of different material constants was performed and comments were made on specific material properties that could contribute to inaccuracies. Further research could include simulations of different theoretical models in a similar fashion, implementation into three dimensions, and creation of an energy-based simulation with more modern material properties.

Two dimensional finite element contact simulations

This same concept has been used with finite element simulations with differences between contact profiles driving the void distribution. This was actually performed as a proof of concept for the overlap model described in

Figure 6.3. If the finite element simulation described here and the surface overlap simulation had no significant difference after the bonding simulation then it was safe to use the significantly less complex and far more efficient overlap method. The surface profilometer data of a length of 1mm was imported as a point cloud into ABAQUS and then meshed at one node per micrometer. Two

surfaces were oriented facing each other, then they were lowered into contact. The simulation was stopped when the reaction pressure equaled the bonding pressure (computed by taking the sum reaction forces on the top face and diving by the face area). This was done for both similar and dissimilar materials. Simulations were performed at intermediate temperature and pressure. The final surface profile was compared with the algorithm that simply overlaps the surfaces without any deformation for the exact same profile and processing conditions. The difference between the surfaces was that using the finite element method the initial bonding fraction was 8.84% bonded while using the overlap method it was 10.4% bonded. 1.2% additional bonding is small and therefore the initial plasticity can be estimated using the overlap of the surfaces.

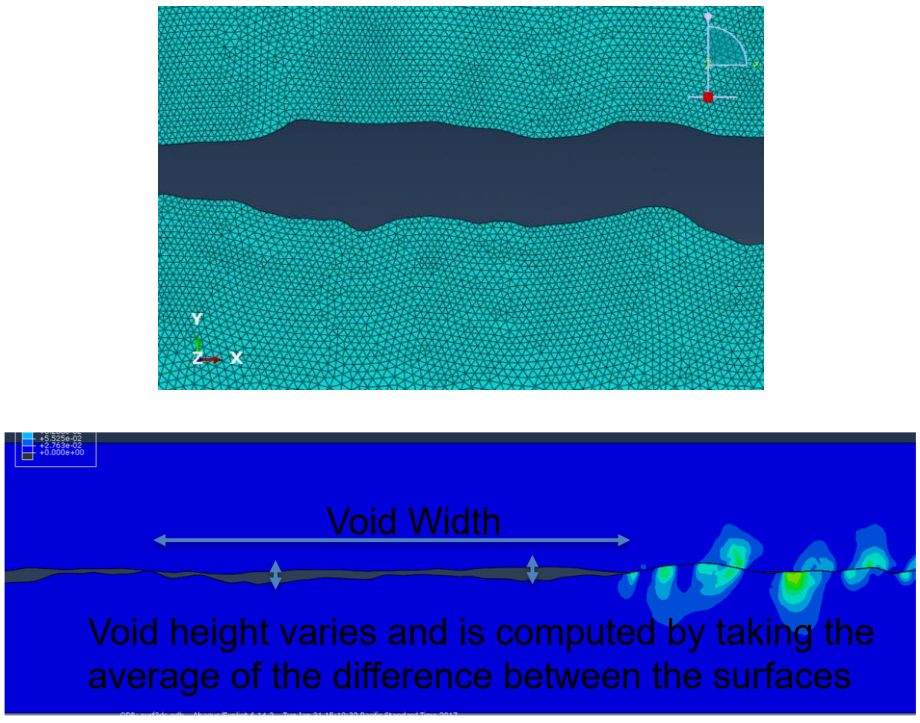


Figure 6.11 – Mesh quality (top) and void dimension calculation (bottom). The mesh used in this simulation is 1µm linear tets

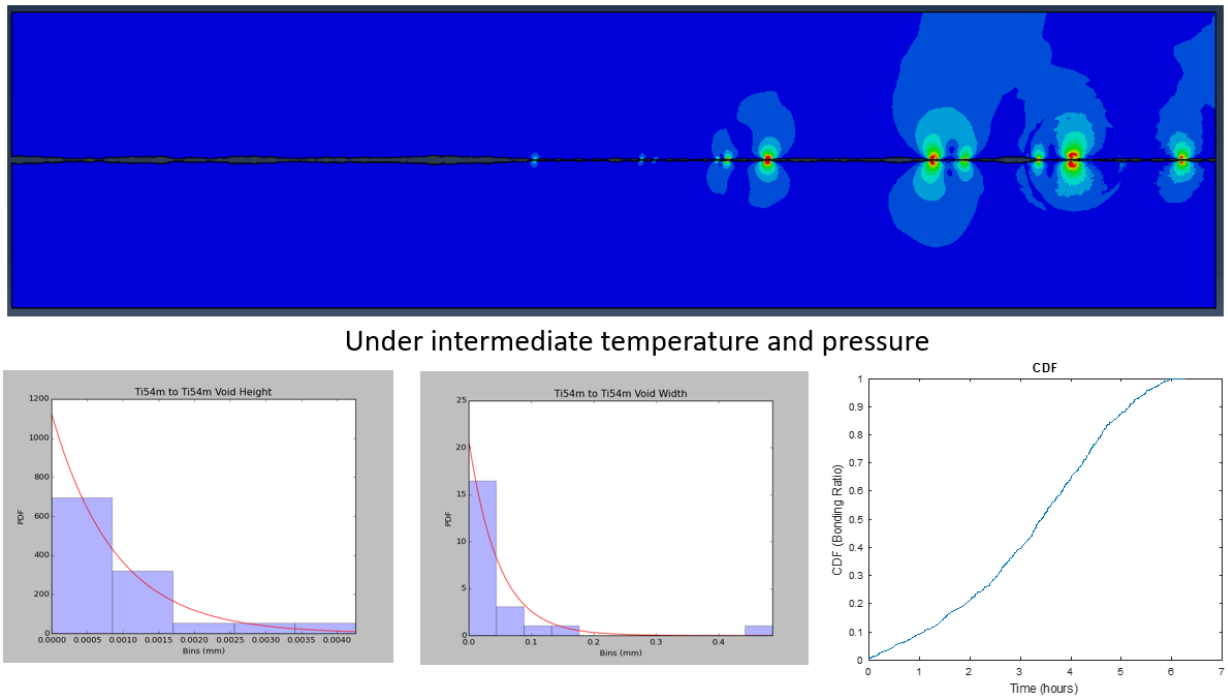


Figure 6.12 - Example of detected contact point locations for a material combination and the detected voids as well as a run of the model for medium temperature and pressure

6.3 Surface Tracking Model

Introduction

Diffusion bonding is a solid state welding process that has seen wide use in the aerospace and electronic manufacturing industries. The process takes two chemically clean, relatively oxide free, surfaces and welds them together under pressure at a temperature around 0.5 to 0.8 times the melting temperature. As the surfaces come together and interact, the micro asperities in each surface form small voids which need to be closed in order to have a structurally sound bond. These voids close due to both diffusion processes and bulk deformation of the material. What is left is a bond that can have nearly indistinguishable mechanical properties from the bulk material. It is well established that the primary processing conditions are the bonding temperature, the bonding

time, the applied bonding pressure, the material properties, and the surface topography of each surface [3]. These process parameters are well understood except for the surface topography. Despite being universally accepted as an important parameter there has been almost no research into the shapes of initially formed voids, the interaction between voids during bonding, or how specific changes in the surface geometry could affect bonding time. To help understand these effects we have developed a model. This model discretizes two surfaces, brings them into contact to form voids, and applies the bonding equations to see how the meshed voids change during the bonding process.

Background

Diffusion bonding takes two interacting parts and goes through a variety of processes and effects in order to achieve reduction and closure of the voids formed. The process starts with an initial contact where the surfaces are brought together. This causes plastic deformation to occur at the surface tips. These contacting points deform until the elastic stresses reach the bonding pressure and form voids in between the points. The voids form a rough line, called the bondline, that has intermittent bonded and voided material. Once the forces are balanced, a variety of diffusive mechanisms close the voids. Typically bonding is modeled with diffusion occurring along the void surface, diffusion occurring from the grain boundary that is created at the bondline, and power law creep of the material surrounding the void. Diffusion also occurs in the bulk of the material at a slow rate. Additionally, at very high temperatures, a small amount of gaseous diffusion occurs due to vaporizing the surface of the void and redepositing that material in a location with lower energy but this has been shown to be negligible [15]. Figure 6.13 shows an example of the voids generated by this process.

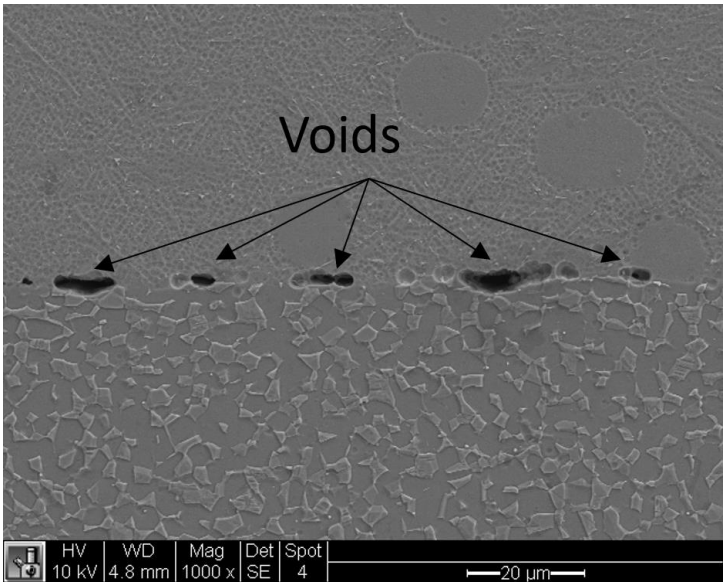


Figure 6.13 Example voids in diffusion bonded titanium alloys

Research attempting to model the process used theory derived from pressure sintering models or reverse formulations of models calculating voids formed during high temperature plastic deformation. They typically include formulations for initial plastic collapse and the diffusive mechanisms of surface diffusion coupled with volume diffusion along the surface of the void, boundary diffusion coupled with volume diffusion along the bondline between the voids, and the complex processes away from the voids are assumed to act as power law creep. Other inclusions have been vapor diffusion within the void and grain boundary diffusion from boundaries along the void surface. There are some differences in how each research treated these basic bonding mechanics but this is not the topic covered here. We are looking at how each model treated the void geometry, what assumptions were made to get the void geometry from the surface topography, and what conclusions were made on how the surface could affect the bonding results. Some of these void geometries can be seen in Figure 6.14. The first attempts at creating a model that could predict diffusion bonding processing conditions began in the 1970s. Hamilton published

the first model for titanium using uniform prismatic pyramidal shaped voids that were independent of any surface characteristics [12]. Garmong used geometry similar to Hamilton's pyramidal model for an initial bonding stage that transitioned into an array of spherical microvoids for final closure. Pyramids were based on the large wavelength and height of the surfaces while the microvoids were based on the superimposed short wavelength and height. He concluded that the large geometry of the initial stage had no effect on bonding time but the short wavelength details that formed the microvoids did [13]. Derby used Hamilton's pyramid void shape that transitioned to a cylindrical void during bonding. The pyramids are also non-uniform and based on the average roughness and wavelength of the measured surface. He concluded that bonding is a function of roughness and wavelength but only up to a point after which there was no influence [14, 93, 94]. Hill used elliptical voids with the minor and major ellipse axis based on the root mean squared surface roughness or average surface roughness, depending on machining type, and the average wavelength respectively. He did not show any trends for surface effects but noted that his model was nearly independent of wavelength but highly dependent on roughness [15]. Pilling, et al. created several models specifically for Ti-6Al-4V using both prismatic cylinders on the diffusion bonding plane as voids [16] and cylinders oriented perpendicular to the diffusion bonding plane as voids [17]. Cylinder geometry was based on the surface roughness and wavelength (in the latter model). His models had a significant dependence on the roughness parameters with larger voids leading to longer bonding times [18]. Orhan uses a sine wave for the void geometry and found that increasing surface roughness increased the bonding time [11]. Ma created a bonding model using an ellipse and used the surface roughness parameter Rz for the void height and root mean squared wavelength for the width [23]. His model was implemented by Li who found that there was negligible influence on the surface roughness during bonding [95]. Complete models using

lenticular voids were created by Guo [92] and by Takahashi [96] with void widths and heights being based on roughness and wavelength but the dependencies were not discussed. Stochastic models created based on the Pilling 1988 model by Li [24] and Kulkarni [25]. Kulkarni treated the surface uniquely and interacted two surface profiles to develop a statistical distribution of void widths and heights for input into the model. A strong trend for the void width and a weak trend for the void height was found. Overall, most of these models have covered different void geometry and implemented the bonding mechanics in different ways. This causes the conclusions about the effects of surface topography on bonding performance to range from no effect at all to a strong trend. There is no consensus on how the surface effects bonding but, aside from Hamilton, it is universally included as important and in most models a rougher surface leads to a longer bonding time.

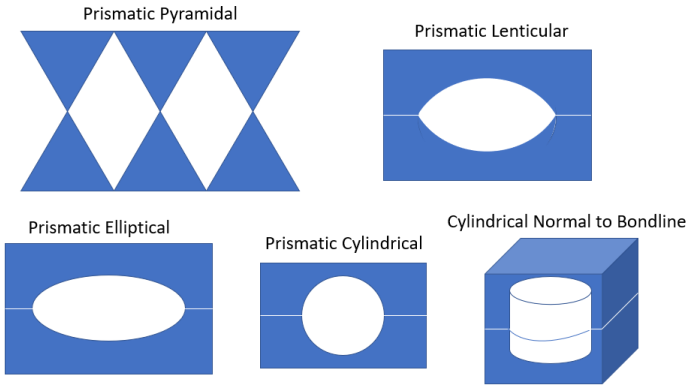


Figure 6.14 Examples of different void geometries.

The surface effects on diffusion bonding have been examined experimentally by correlating surface roughness parameters with bonding performance. This has primarily been conducted by

grinding surfaces down to different surface roughnesses and analyzing bonded material through a micrographic look at the voids or by doing structural testing. The first person to look at this problem was Islam [19] who compared as received/pickled surfaces and rougher P60 ground surfaces of titanium with results from a lap shear test. He observed that the voids on the ground material were much larger and more numerous after a given bonding time than the smoother received surfaces. The rougher surfaces with more voids correlated with lower shear strengths while the smoother surfaces were stronger with fewer voids. Similar tests have been performed in a variety of ways with a variety of materials but usually yielding the same general trend. Wang [97] bonded cylinders of copper with different surfaces then tested them in tension with the same trend. Shao [98] bonded titanium and nickel together with ground and polished surfaces then performed shear testing with the same trend. Similar results were found by Guo [99] for tungsten carbide cobalt bonded to steel. Li [95] used titanium surfaces ground with four different grits spanning two different pressures with the same trend. Zhang [100] had the same conclusion using steel at two different polishing grits. There have been a few exceptions to this trend. Zuruzi [101] determined that between grinding with two grits, P180 and P1800 on 6061 Al, the rougher P180 grit yielded better bonding both through micrographic analysis and structural experiments. Somekawa [102] compared a variety of grit blasted magnesium specimens with shear tests. He determined that up to a certain point the rougher surfaces created better bonds though this is partly due to the interaction of a significant oxide layer on those surfaces. Overall the conclusions have been that the surface topography has a significant effect on bonding performance and in most cases smoother surfaces have led to shorter bonding times.

So far, the extent of theoretical and experimental knowledge about the effects of the surface topography on diffusion bonding has been that it is likely that a rougher surface leads to larger

voids which take longer to close. However, there has been no detailed examination of the surface effects on bonding. The current model is proposed to determine what those effects might be. As stated above, the model takes two surfaces and brings them into contact. Then the model applies the diffusion bonding mechanisms of surface diffusion, boundary diffusion, and power law creep to those surfaces. The results are confirmed by a comparison with a selection of similar and dissimilar bonded titanium specimens. Then a variety of tests are performed to determine the possible effects of surface geometry on bonding performance. Tests are conducted to determine the effect of surface roughness, wavelength, void-void interaction using a variety of void shapes, as well as the distribution of contact points. What follows is a description of the model theory and numerical implementation, an explanation of the experiments run both physically and numerically, and a look at the results and conclusions.

Theoretical Model

In the literature so far, the simulations for diffusion bonding have been analytical models that describe the bonding mechanisms based on an assumed void shape. The current model investigates the effects of surface geometry by tracking the deformation of discretized voids without forcing them to have a specific shape. While this type of simulation has not been specifically studied for diffusion bonding, and to the authors knowledge has never been studied using surface profiles, it is theoretically similar to micromechanic damage simulations that model void growth and coalescence in creep deformation. These methods have been implemented in both finite differencing as well as finite element forms and all use nodes that are placed along the void surface and along the primary grain boundary connecting voids together. Then creep and diffusion equations are used to describe how the position of the nodes update. Similar mechanics that have been studied are void generation and microcracking due to creep and diffusion [103]; high

temperature crack healing in metals [104, 105]; cavity growth controlled by surface diffusion [106, 107]; and grain boundary growth in combination with power law creep using a finite element framework for two [108] and three dimensional voids [109, 110]. The primary basis for this research is the work by Takahashi for void shrinkage [111, 112].

Initial Contact

When surfaces are brought together, the differences between them form complex voids. The shapes of these voids are based on the topography of the surfaces and produce a range of void shapes and contact points. These voids are too complex to be handled by the basic shapes described in Figure 6.14 since they have peaks and valleys at random intervals. An example of this is shown in Figure 6.15, where two surface profiles are contacted together and a detail of the surface geometry is shown. Initial plasticity occurs as the tips of each surface contact the opposite surface and, due to the extremely high stresses, the tips deform until the elastic stresses in the material are capable of handling the bonding pressure. In this model it is assumed that the initial fraction of bonding, f_0, after the initial contact is the applied pressure, P, divided by the yield stress, σ_y.

$$f_0 = \frac{P}{\sigma_y} \tag{6-8}$$

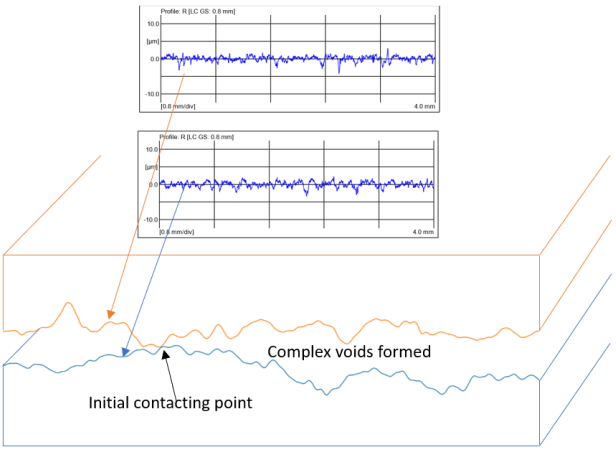


Figure 6.15 Initial contact from surface profiles

Surface Diffusion

The equations for diffusion are applied to the surface profiles in different ways depending on the type of diffusion. Diffusion equations are formulated by Fick's second law using the chemical potential due to the surface curvature along the void surface and the stress along the bonded region. Fick's diffusion law is given by Equation 6.9, where J is the atomic flux, D is the diffusion constant, k is the Boltzmann constant, T is the temperature, Ω is the atomic volume, ∇ is the gradient, and Δμ is the chemical potential.

$$J = -\frac{D}{kT\Omega}\,\nabla\Delta\mu \tag{6-9}$$

Along the void surface, the chemical potential is based on the Gibbs-Thompson relation which describes the variation in potential based on the curvature of a free surface. The variation in strain energy is neglected on the void surface along with normal stress terms since these terms provide little potential energy difference and are very small compared with the curvature. The two types

of diffusion that are applied to the surface curvature potential are surface diffusion and volume diffusion. Equation 6.10 gives the chemical potential of the flux equation (Equation 6.9) denoted $\Delta\mu_s$ for surface diffusion. It is a function of the atomic volume, Ω, the surface energy γ_s, and the surface curvature κ. The adjacent volume diffusion is assumed to have the same chemical potential but a different diffusion constant in Equation 6.9. The minus sign in the equation is to account for the sign of the curvature and enforces energy minimization.

$$\Delta\mu_s = -\Omega\gamma_s\kappa \tag{6-10}$$

Boundary Diffusion

Boundary diffusion, which is diffusion from the grain boundary formed at the bondline into the void, uses the same flux equation as surface diffusion but with a chemical potential that comes from the work expended to remove a layer of atoms from the grain boundary. The chemical potential defined along the grain boundary, $\Delta\mu_b$, a combination of the atomic volume and the average stress along the bonded area, is given in Equation 6.11. The bondline grain boundary intersects the adjacent voids at points that have typically very high curvature called the void tips. The difference between the void tip chemical potential (Equation 6.12), $\Delta\mu_{tip}$, and the grain boundary chemical potential (Equation 6.11) provides the potential gradient for the boundary diffusion equation.

$$\Delta\mu_b = -\sigma\Omega \tag{6-11}$$

$$\Delta\mu_{tip} = -\Omega\gamma_s\kappa_{tip} \tag{6-12}$$

Where the grain boundary intersects with the void tip there is a balance between the mass transfer from the void tip and the mass transfer away due to surface diffusion. The material from the void tip must transition from flowing through the boundary to diffusing along the surface. In the case of titanium there is significantly higher surface diffusion than boundary diffusion (Ds>>Db). Therefore, the atom transfer into the void from the grain boundary is rate limited by boundary diffusion and not by surface diffusion removing it from the tip. Therefore, the tip geometry can remain constant. We can also assume that the mass flux at the center point of the void is zero and that the process is steady state. The resulting integration and force balance gives flux along the boundary and its respective displacement [111]. In the equations below, β is the steady state atom flux, L is the half length of a void, X is the half length of an adjacent boundary, $\frac{du_r}{dt}$ is the change in height of the boundary material.

$$\nabla \cdot J_b = \beta \qquad \beta = -3\frac{D_b}{kT}\frac{1}{X^2}\left(\frac{L}{X}\sigma - \kappa_{tip}\gamma_s\right) \qquad \frac{du_r}{dt} = -\frac{\delta_b}{2}\Omega\beta \qquad \kappa_{tip} = \frac{L}{X\gamma_s}\sigma - \frac{Xf}{3}\left(\frac{d\kappa}{ds}\right)_{tip} \tag{6-13}$$

Power Law Creep

The bonding pressure that is applied to the exterior of the material is supported by the bonded contacting areas at the interface. The variety of diffuse properties are assumed to act as power law creep. The power law creep equation (Equation 6.14) contains the local stress, σ, the high temperature shear modulus, G, creep exponent, n, and a creep constant A_c.

$$\dot{\varepsilon} = A_c \left|\frac{\sigma}{G}\right|^n \tag{6-14}$$

The simulation occurs in two different steps, an initial contact step and a closure step, and is formulated is such a way that it is completely independent of the input surfaces. The contact step discretizes the surfaces and brings them together at the correct amount to provide the initial conditions for the simulation. The closure step applies the diffusion equations to the surfaces at each timestep and keeps running until the bonding time is reached. The voids are assumed to be prismatic, infinitely long, and the equations are formulated in plane strain allowing for a 2D representation. The void closure process is modeled sequentially from surface diffusion to boundary diffusion to power law creep. The upper and lower surfaces are tracked individually and can be set with different material properties. Before each timestep, the surfaces are characterized for the positions of voided areas and bonded areas. An algorithm sorts through the entire bonding surface and logs the beginning and end of each void and bonded section. This allows new bonds and new voids to form as a result of the previous timestep. Neumann boundary conditions are set at the edges of the surfaces with the differential set to zero, $\frac{\partial y(edge)}{\partial x} = 0$, so that the edges are allowed to change in height but not allowed to have significant curvature that would affect the results. The stress state of the system is uniaxial pointed inward toward the bondline. Diffusion and creep are applied to the voided sections but not the bonded material which is taken as unaffected by the closure process. After deformation calculations are completed for each time step, the top and bottom surfaces are re-meshed to prevent errors that would occur due to extremely large mesh deformation. A flowchart of the process is located below in Figure 6.16.

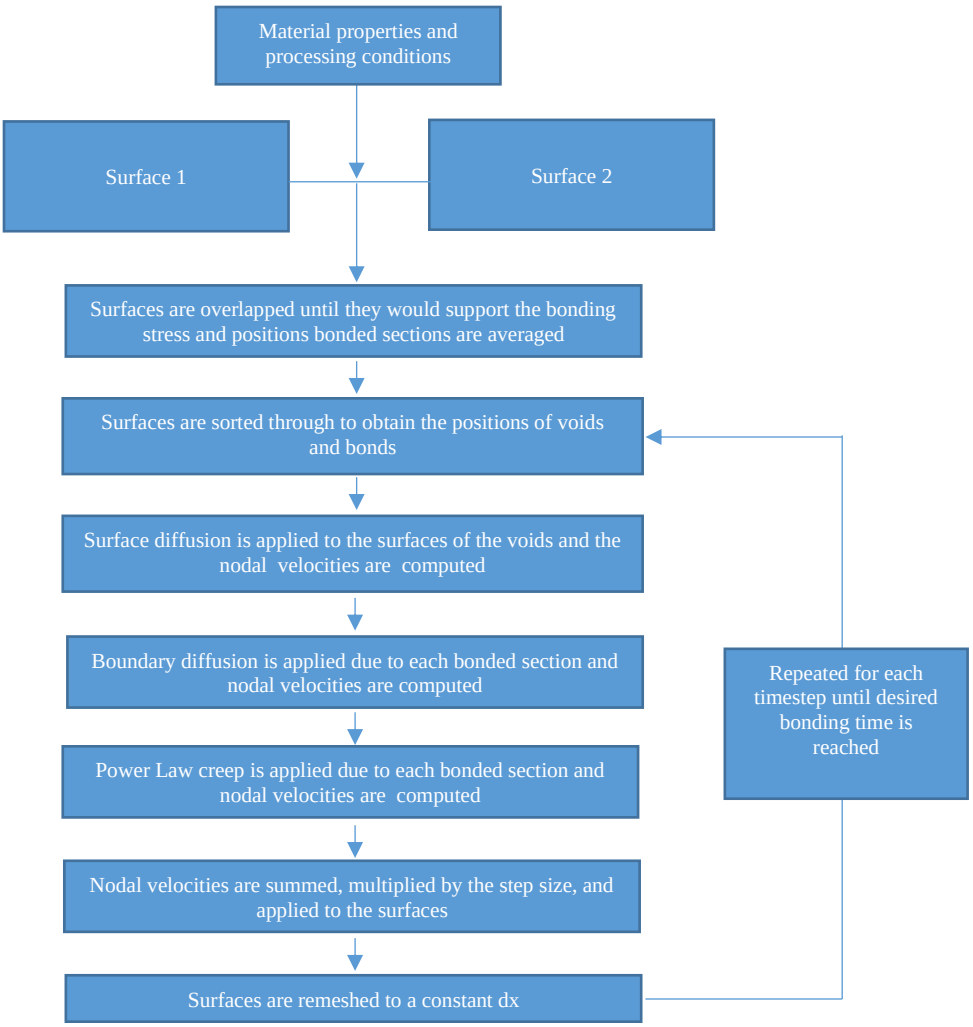


Figure 6.16 Flowchart of the implementation of the bonding model

Initial Contact

The initial contact step takes two different surface profiles and overlaps them until the length of overlap corresponds to the initial plastic deformation represented in Equation 6-8. The overlapped surfaces are represented as a set of two lines with equal values in some sections and deviating values in others. At the equal valued points the material is considered bonded and at the deviating points it is considered to be a void. Figure 6.17 below shows a typical example of what the surfaces

look after the initial contact step. The bonds and voids are labeled and a coarse mesh is shown. An actual input is much larger with many more voids and a denser mesh.

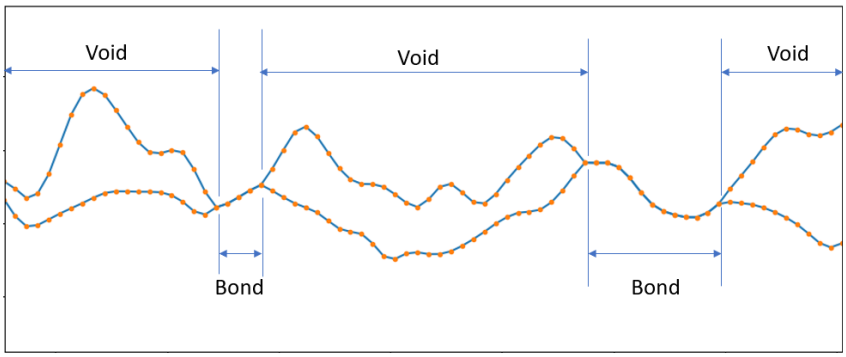


Figure 6.17 A magnified section of a full simulation showing the initial conditions after overlap with a coarse mesh for visibility. The two surfaces are brought together to the amount represented in Equation 6-8 then the overlap is averaged to form the plot above

Surface Diffusion

Surface diffusion and volume diffusion act on the surface of the void reducing the variations in the surface curvature. Numerically, the curvature is calculated using the differences in adjacent points to obtain a derivative (Equation 6.15). Surface diffusion acts on the surface thickness described in the material properties while the volume diffusion is assumed to act on a layer two times that thick. The curvature, κ, is listed in Equation 6.15.

$$\kappa = \frac{\frac{d^2y}{dx^2}}{\left(1+\left(\frac{dy}{dx}\right)^2\right)^{\frac{3}{2}}} \tag{6-15}$$

The changes in coordinates are calculated by determining the relative flux of material and computing the difference in positions. The velocity of the node perpendicular to the void surface,

$\frac{dv_n}{dt}$ (Equation 6.17), is determined by the change of the atomic flux along the surface J_n, (Equation 6.16). This can be seen in Figure 6.18a where the flux, J_n is transferred along the surface due to both types of diffusion. The velocities are always normal to the surface in varying magnitudes depending on the amount of transfer shown in Figure 6.18b. These velocity vectors have the highest magnitude at the highest curvature and point in a direction that acts to smooth out the surface. The positions are updated by taking the angle between the x axis and the void surface, α, and finding the new x (Equation 6.18) and y (Equation 6.19) positions with the velocity and time step.

$$J_n = \frac{\frac{D\gamma_s}{kT}(\kappa_{n+1} - \kappa_{n-1})}{\sqrt{dy^2 + dx^2}} \tag{6-16}$$

$$\frac{dv_n}{dt} = -\delta\Omega \frac{J_{n+1} - J_{n-1}}{\sqrt{dy^2 + dx^2}} \tag{6-17}$$

$$x_n(t + \Delta t) = x_n(t) - \frac{dv_n}{dt}\Delta t \cos(\alpha) \tag{6-18}$$

$$y_n(t + \Delta t) = y_n(t) - \frac{dv_n}{dt}\Delta t \sin(\alpha) \tag{6-19}$$

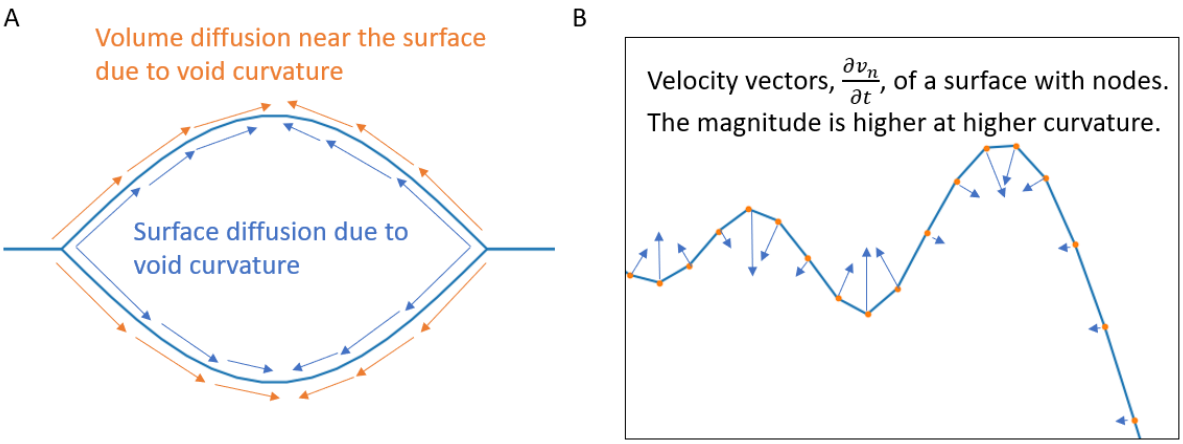


Figure 6.18 Basic effects of surface diffusion (left) and the velocity vectors on a real surface (right)

Boundary Diffusion

The boundary diffusion step provides both an expansion of the bondline and a decrease in the void height. The physical process has a small amount of material that is diffused from the boundary at the bonded area that relocates into the void as shown in Figure 6.19. The amount of material transferred is based on the boundary equations 6.18 and 6.19. The calculation adds material to the void by changing the void positions to account for the material transfer. Since titanium has significantly larger surface diffusion than boundary diffusion the material removed from the bonded area is assumed to rearrange itself on the adjacent voids uniformly without any buildup of material around the void tips. There is an additional effect caused by removing material from the bonded regions. Since the bonds are the vertical supports for the voids, when you remove material from the supports, the entire void translates toward the bondline. This is called rigid collapse in the literature and is far more influential than the amount of material actually transferred into the

void from diffusion. The voids and bonded sections are of non-uniform shape and therefore have different amounts of material flow and different rigid collapse contributions on either side of the void. Therefore, they are treated independently with each bonded section contributing half of its material to each adjacent void.

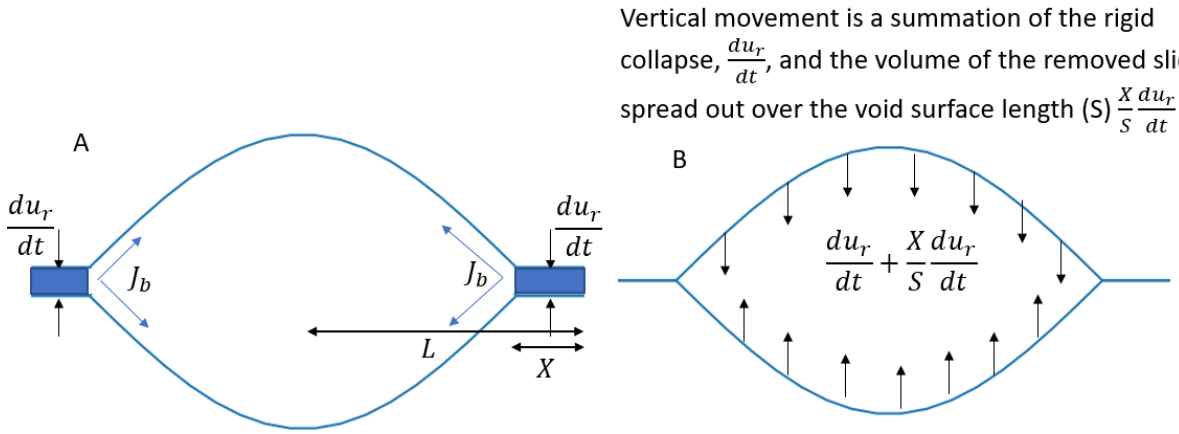


Figure 6.19 The effects of boundary diffusion are to rigidly collapse the voids and transfer material into the voids (left). The material transferred is assumed to spread over the surface of the void (right)

Power Law Creep

The bonding pressure applied on the surface of the titanium sheets is relatively far away from the voids and with the high number of contact points, the bonding area acts as a large, flat plane with distributed random supporting points. Any bending and elastic deformation is neglected and stresses are maximized at the contact points. It is assumed that the stresses are uniform and purely a function of the distance away from the contact points and that they are uniaxial compression directed inward toward the bondline. The stresses decrease from the contact points (i.e. the bonded

areas) until they reach the top of the voids where they equal the applied bonding pressure. The stressed material deforms vertically at the rate governed by power law creep, Equation 6.14, and horizontally due to the constant volume assumption.

Implementing this equation requires slicing the solid material above the bonds and applying power law creep to each slice then summing the deformation of the slices. However, there are some adjustments for random geometry. Not all parts of a void are evenly increasing in height, some material is hanging and would not be stressed; furthermore, not all parts of a void are subjected to the stresses caused by a single bonded section so the effects of multiple bonded areas needs to be accounted for. Additionally, with fixed horizontal spacing the vertical component of the mesh is non-uniform resulting in the nodes on one side of a superplastic section being at different locations than the other side meaning that one must determine the span of the stressed slices by interpolation of the mesh. For this implementation, each bonded section is detected as described above, then a list of nodes is created that are ever increasing vertically away from a bond until it reaches the maximum points on either side of the bond. Each key point has its own slice but only a fraction of the effects are taken into account. The horizontal span of a slice for each key point is calculated by interpolating on voids located opposite their locations. The vertical height of each slice is represented by the average vertical distance between two adjacent key points. The contribution of each bonded section to the stress at a key point slice is summed. Vertical deformation rate of each key point is calculated by power law creep and half the horizontal deformation rate is applied. The vertical deformation is a series operation with each slice changing its own position and the position of every slice above it. To obtain the total vertical deformation of each key point, a cumulative sum is performed on the left and right halves of the void. The halves are partitioned by the point on the void with the maximum height (which gets an average of the deformation of the two halves).

This way each point on the void gets the contribution of all the material acting on it and the material that was hanging and unstressed is relocated to the proper position. Horizontal deformation is not in series and is only applied locally. The slicing done at key points can be seen for two different voids in Figure 6.20. Each individual slice changes the position of a node.

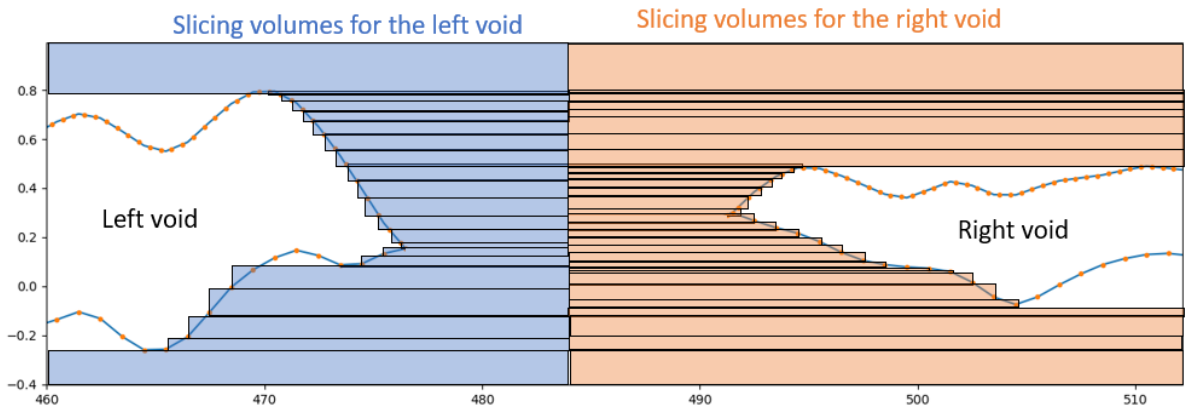


Figure 6.20 – Geometry of slices for two different voids. Power law creep is applied to each slice deforming the slice vertically and horizontally.

The calculation can be described by the following equations for each slice. Each node has a slice of material that will deform and pass the change in position due to that deformation to the node. Each slice needs the stresses associated with it (σ_n), and the width (w_n), the height of the slice (h_n). For any given node, the width between that node and the opposite side is an interpolation to determine the x position of the nodes on the opposite side that have y positions near the node in question (Equation 6.20). σ_n is the stress experienced by that slice which is a summation of the local bonding pressure, $\frac{P}{w_{bi}}$, multiplied by the current width (Equation 6.21). The vertical height of a slice is the y position of the current node subtracting the previous node position (Equation 6.22). The change in slice height over time is the power law creep equation and is equated to the

vertical velocity of the slice (Equation 6.23). The horizontal change is the constant volume assumption (Equation 6.24).

$$w_n = interp\left(x_{n\ opposite}\right) - x_n \tag{6-20}$$

$$\sigma_n = \sum_{i=0}^{\#\ of\ bonds} \frac{P}{w_{bi}} w_n \tag{6-21}$$

$$h_n = y_n - y_{n-1} \tag{6-22}$$

$$\frac{dy_n}{dt} = h_n A_c \left|\frac{\sigma_n}{G}\right|^n \tag{6-23}$$

$$\frac{dx_n}{dt} = \frac{dy_n}{dt}\frac{h_n}{2w_n} \tag{6-24}$$

Material Properties

The material properties used in the simulation were applied by the rule of mixtures for each material's alpha beta ratio. The alpha-beta percentages were determined through micrographic analysis. For each material combination a representative 1mm length of surface profile was selected. The material properties used for alpha and beta are listed Table 6-3.

The material properties are a primary factor that affects bonding performance but, depending on the system, not every material property has a large influence. The surface diffusivity has very little influence on the overall bonding. Its influence is to adjust the curvature of the void and aid the other bonding processes but does very little closure on its own. The grain boundary diffusivity has more influence and can be seen in the figure below but is still very small. Since power law

Table 6-3 – Alpha and beta material properties.

Properties	α	β
Atomic Volume "Ω" (m^3)	1.76×10^{-29}	1.81×10^{-29}
Burgers vector "b" (m)	2.9×10^{-10}	2.9×10^{-10}
Surface energy "γ" (J m^2)	1.7	1.7
Grain boundary diffusivity "$D_{gb}\,\delta$" (m^3/s)	$3.6 \times 10^{-16} \times \exp(-97000/(R \times T))$	$5.4 \times 10^{-17} \times \exp(-153000/(R \times T))$
Grain boundary width "δ" (m)	5.9×10^{-10}	5.72×10^{-10}
Volume boundary diffusivity "D_v" (m^2/s)	$8.6 \times 10^{-10} \times \exp(-150000/(R \times T))$	$1.9 \times 10^{-7} \times \exp(-153000/(R \times T))$
Shear modulus "G" (MPa)	$4.3 \times 10^4 \times (1-((T-300)/T_m) \times 1.2)$	$2.05 \times 10^4 \times (1-((T-300)/T_m) \times 0.5)$
Surface diffusivity "D_{0s}" (m^2/s)	$5 \times 10^{-6} \times \exp(89000/(R \times T))$	$5 \times 10^{-6} \times \exp(89000/(R \times T))$
Surface diffusion width "δ_s" (m)	1×10^{-10}	1×10^{-10}
Creep constant "A"	1.2×10^{-9}	1.2×10^{-9}
Strain rate creep constant "A_c"	$A \times (D_{gb}\,\delta)_\alpha \times G_\alpha/(k \times T)$	$A \times (D_{gb}\,\delta)_\beta \times G_\beta/(k \times T)$
Creep Exponent "m"	4.3 or 3.8	4.3 or 3.8
Gas constant "R" (J/mol K)	8.314	
Boltzmann's constant "k" (J/K)	1.38×10^{-23}	
Melting temperature "T_m" (K)	1973	

creep is the dominant closure mechanism its material properties have the most influence. The shear modulus has a large effect since it influences both the initial overlap as well as creep. The creep constant as well changes the power law creep performance. However, by far the most important material property is the creep exponent. It also happens to also be the most debatable property. Several papers that use titanium have referenced different exponents for power law creep with these same titanium material properties specifically 4.3 for creep and 1.43 for superplastic flow [11, 17, 18]. Changing the creep exponent this much would result in drastically different bonding performance. Since there is a debate, a modified creep exponent of 3.8 is used which results in a much better fit of the experimental data.

Experimental Procedure

Physical Experiments

Experiments were performed to determine the surface roughness profiles for the initial conditions of the simulation along with an analysis of diffusion bonding specimens to determine the accuracy of results. This data was from the two sheet diffusion bonding section of this dissertation.

Numerical Experiments

Along with replicating the experimental results, tests were conducted to determine the potential effects of surface roughness, surface wavelength, the interaction between voids, and the contact area geometry. Surface roughness tests and surface wavelength tests were performed on a consistent set of profilometer data that has been variably stretched in a given direction. For the surface roughness this means taking the surface profile and stretching it vertically away from the centerline. The distance between each data point on the surface profile and the centerline is multiplied by a factor, A, that is located within the Ra parameter (Equation 6.25). This stretches the profile away from the centerline. Changing the average wavelength of a surface is done by simply changing the assumed spacing in between points and reinterpolating to obtain a standard dx. This gives a horizontally stretched surface with the same overlap ratio and geometry. By performing the roughness and wavelength tests in this way, the rest of the surface characteristics remain constant and the only thing that changes in the simulation are those two parameters. This prevents some of the other potential effects of unique surfaces that could skew the simulation results. Void-void interaction is tested by creating synthetic surfaces of voids of different shapes next to each other. It is tested with tall voids next to small voids and long wavelength voids next to short wavelength voids. These tests are all done with the same spacing and initial overlap. The effects of the distribution of contacting points is also tested. The distribution of contacting points is examined by taking voids of the same shape and translating them around. All of these tests were performed multiple times to get a distribution and the basic geometry can be seen in Figure 6.21.

$$R_a = \frac{1}{n}\sum_{i=1}^{n} Ay_i \quad (6\text{-}25)$$

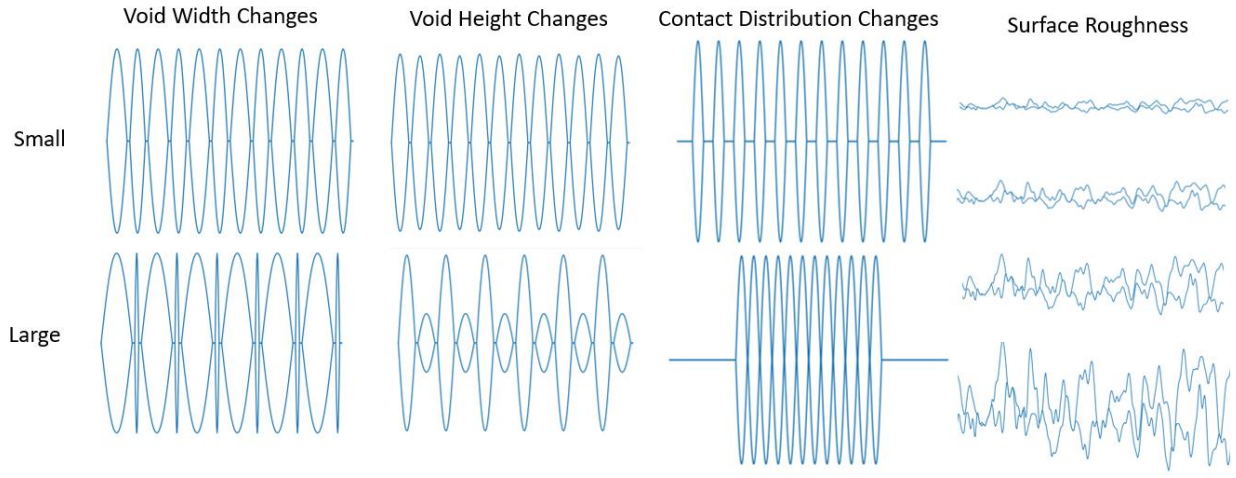


Figure 6.21 - Numerical Experiments for void-void interaction, contact, and surface roughness

Results and Discussion

The results of the physical experiments are presented below in Table 6-4 along with the simulations on those surface profiles. In Figure 6.22 bonding can be seen in the plot below at the initial condition, the final condition, as well as an experimental micrograph for comparison. The trends in bonding were well predicted. As temperature increases then so does the bonding percentage. The theoretical bonding percentage is well below the experimentally determined bonding percentages. While it is common for there to be some discrepancy since all of the other titanium based diffusion bonding models have not been close to their experimental data, this is not ideal. For reasons discussed in the material properties section, the creep exponent has been adjusted to account for the difference between the two and fits the experimental data significantly better. On average, the difference between the theoretical bonded % and the experimental data is 56.5%. The

average difference between the creep adjusted theoretical bonded %, with a creep exponent of 3.8 instead of 4.3 and the experimental data is 3.4%.

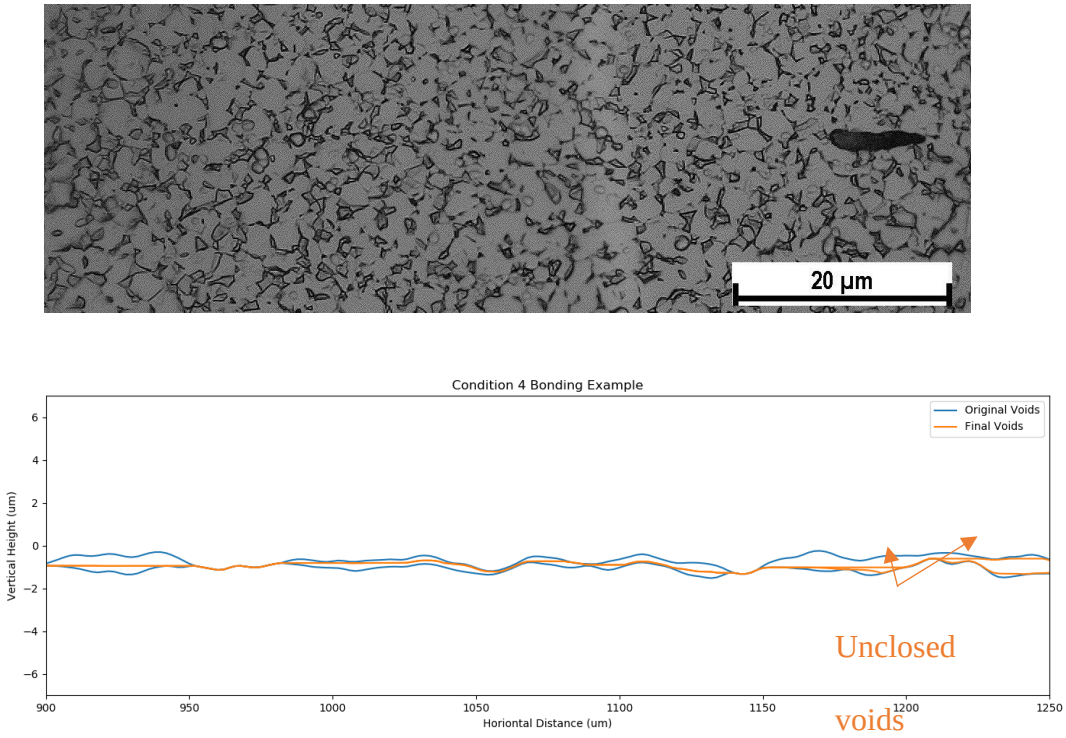


Figure 6.22 – Experimental micrograph (top) and simulation results (bottom).

Table 6-4 - Results comparison

Condition	Top Material	Bottom Material	Temperature	Experimental Bonded %	Theoretical Bonded %	Creep Adjusted Theoretical Bonded %
1	Ti64 FG	Ti64 FG	0.53Tm	64%	31%	70%
2	Ti64 FG	Ti64 FG	0.544Tm	92%	36%	89%
3	Ti64 FG	Ti64 FG	0.559Tm	100%	41%	97%
4	Ti54M	Ti54M	0.53Tm	96%	32%	92%
5	Ti54M	Ti54M	0.544Tm	100%	38%	100%
6	Ti54M	Ti54M	0.559Tm	100%	41%	100%

7	Ti54M	Ti64 FG	0.53Tm	95%	33%	88%
8	Ti54M	Ti64 FG	0.544Tm	100%	40%	95%
9	Ti54M	Ti64 FG	0.559Tm	100%	46%	97%

Above a certain point, the surface roughness and wavelength have a negligible effect on the bonding performance in this model. This is shown Figure 6.23 where the bonding percentage of scaled surfaces increases only at very small dimensions. While this may seem counterintuitive, especially since those are the primary metrics used in the literature to quantify a surface, this is not the first model to conclude this. The reason is the power law creep mechanism assumed in this model scales with those dimensions, both horizontally and vertically. Specifically the stresses, which are assumed to scale from top of the tallest void down to the bondline with the distribution in between as a function of the widths between voids. So scaling to a larger void, for example, also scales the dimensions over which the creep mechanism is operating and results in equally larger material transferred to the void. This can be seen in the Equations 6.23-6.25. Increasing the surface roughness has the effect of scaling h_n which is a corresponding scaling of the position change $\frac{dy_n}{dt}$. Analytically, the creep equation describes the strain rate (Equation 6.14) and is not specifically a function of dimensions. This means a 50% vertical strain is going to take the same amount of time regardless of whether the voids are 1μm tall or 1mm tall. This was shown experimentally by Garmong [13] where surfaces were cut to have triangular ridges and pressed against a flat plate. He determined that at larger sizes the bulk power law creep mechanism is the only mechanism that applies and that bonding times were equal. So in situations where creep is the dominant mechanism the bonding time will be constant no matter the surface topography. The

other bonding processes don't scale in this way and at some point the voids are small enough that the boundary diffusion and surface diffusion close the voids at an accelerated rate.

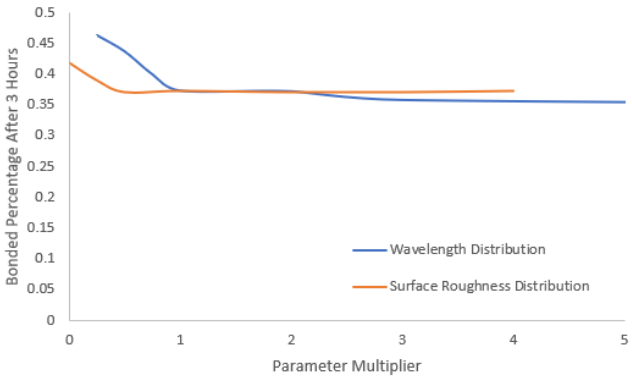


Figure 6.23 - Closure of roughness and wavelength

When the shapes of different voids interact the bonding rates change significantly. Despite having identical volume, material properties, initial overlap percentage, and all other parameters, slight changes in geometry can have a significant effect on the bonding time. The results of these simulations are very consistent, with each simulation being repeated multiple times producing identical bonding times. However, the results of the simulations are challenging to interpret. There appears to be a dramatic change in the bonding time from slight perturbations in the both the heights and void widths when moving away from identical voids (represented by the 0 on the multiplier in Figure 6.24 below). After a short peak in the void height times, the trend is generally downward toward shorter bonding times. This means that as they bond, having some small voids and large voids results in better bonding than having only voids of an identical geometry especially when the difference is dramatic. The void widths are a function of the wavelengths and are periodic. Aside from this periodicity there seems to be very little effects on the bonding time.

Finally, when looking at the distribution of contacting points, the closer the voids are together the shorter the bonding time. This can be seen in the contacting points plot below.

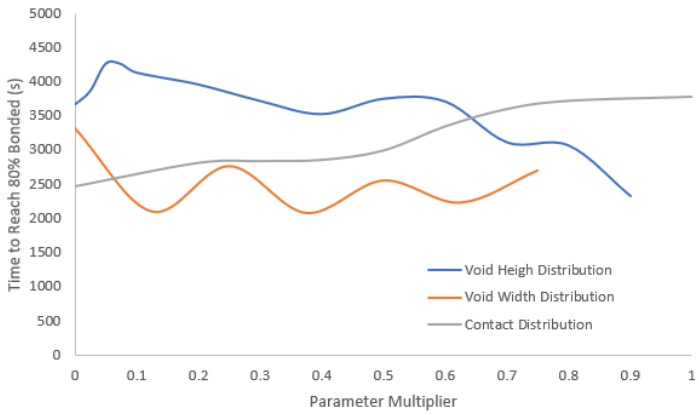


Figure 6.24 - Plot of the effects of periodic variances in void height, void width, and contact point distribution. The results are representative curve fits with enough datapoints to fit the key features of the curve.

Each of the bonding simulations had widely varied sets of voids due to the unique initial surfaces profiles and the interaction of the bonding mechanics. The dominant bonding mechanism for these processing conditions is by far the power law creep collapse. The surface diffusion and boundary diffusion are minor in comparison. This is consistent with other models in the literature. This is primarily due to the surface diffusion and boundary diffusion constants being very small and the thickness of the volume diffusion area is equally small. The summation of the volume of material transferred into each void by each bonding mechanism is logged by calculating the difference between the volumes for each void for each time step. The cumulative sum of the changes in volume divided by the total original volume is presented in Figure 6.25.

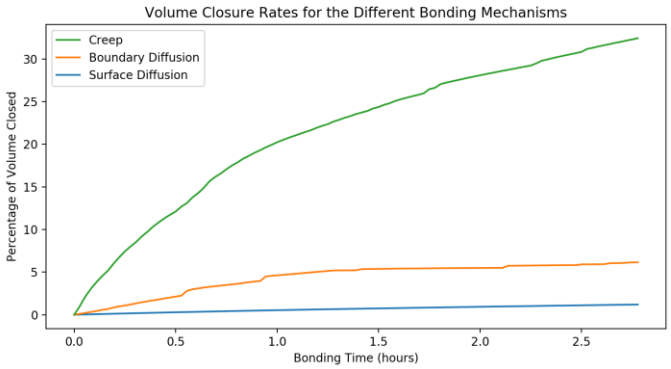


Figure 6.25 - Ratio of volume closure for the different bonding mechanisms for condition 1.

The closure of voids is shown to be highly dependent on the surface. The general size of the voids from a surface is not necessarily a good representation of how long the surface will take to bond, the results are therefore highly varied with some voids bonding faster than others. This simulation provides unique insight into the bonding mechanisms and how they interact with each other since void-void interaction effects have yet to be studied in diffusion bonding literature. This is because voids close to each other modify their bonding behavior. Void interactions and closure are highly dependent on the adjacent bondline, the void and bondline position vertically, and the adjacent void shape. The results of the simulations on surface geometry lead to several conclusions and practical implications. Firstly, it is best to have a surface that mates extremely well and forms voids that are in the diffusive regime. Void sizes above that are subject to power law creep and are going to close at a fixed rate. Practically, this means that surface treatments such as sanding to remove the oxide layer before bonding will have little effect on the bonding time provided that the voids formed are above a certain size. Fine polishing to produce voids in the diffusive regime would also be beneficial. The patterning of voids is important as well but would be very difficult to control. There are some conclusions that can be made however. Producing a surface with some small voids at the expense of producing large voids would still be beneficial. Along with this, for

a fixed bonded percentage, the more concentrated the voids are the quicker bonding will be. While this research does not specifically test these theories, surfaces could be produced that test these theories. An example could be stamping or creating a surface that would provide concentrated initial contact points.

Looking at the power law creep mechanism by its effects on individual voids there is a large variation single void closure time depending on the void shape and the extent of bonded material next to the voids. Voids with similar volume can have vastly different bonding performance. An example of this is demonstrated in the figure below from the condition 9 simulation. The two voids in question have nearly identical volumes with the taller void on the right being 1.9% larger than the flatter void on the left. They share the bonded material in between them and have nearly identical bonded lengths on either side of them. However, they have vastly different closure. The flatter void decreased by 48.2% during the three hour simulation while the taller void decreased by 79.5%. This is due to the difference in shapes of the voids along with the locations of the bonded material and is specifically a function of the power law creep mechanism. The flat void on the left experiences a large compression at its edge but only at its edge up to its maximum point, which is toward the right edge, since it is very flat. This results in a small compression on the voids right edge. The tall void is subjected to the full stress from center bond as well as stresses coming from other contacted areas outside of the figure. It also is a sharp void with relatively little volume when compared with its height and only one local maximum. A combination of these mechanisms causes the difference in void closure. Once the void is small however, the stresses are not so concentrated and the tall void ends up roughly the same height as the flat void. This phenomena is part of the reason for the heavy dependence on the surface roughness described in the literature [11, 18].

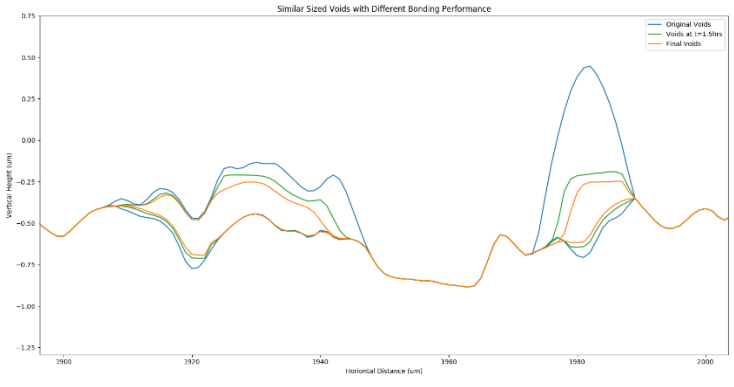


Figure 6.26 - Example of the effects of void shape and bond location to the void closure. The voids have equal volumes but significantly different closure rates due to the shape of the voids and power law creep.

The closure rate not only depends on the height of the voids and the length of the bonded material next to it but the original shape of the void itself. As voids close the create contact points and bonded areas within the void. These bonded areas become stressed and begin pressing on the other surrounding voids. Therefore, irregular voids can bond well despite a large volume by created sub-voids during bonding. As stated above, the closure is highly dependent on the situation the void is in and can be quite complex (shown in Figure *6.27*). As an example, in the figure below, the left void has a volume of 15.45 um^2 while the void on the right has a volume of 0.28 um^2 making the large void 54.6 times the volume of the small void. The void on the left ends up creating nine different unique voids during its closure process. The void on the left takes 560 seconds to close while the small void on right takes 500 seconds to close. The void that is 54.6 times larger only took 1.12 times longer to close. It is extremely difficult therefore to determine how long it will take to close a void simply by looking at the volume of the void or the length of the void. One must take into account the specific location, shape, and its evolution through the bonding process.

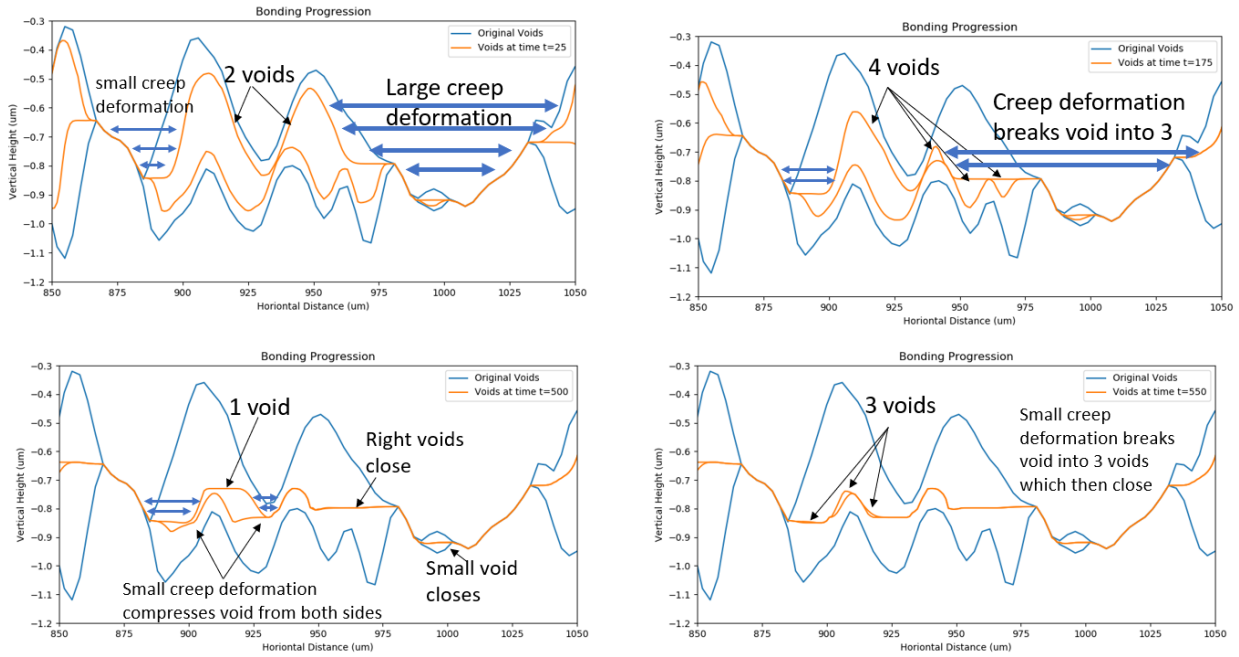


Figure 6.27 - Void closure process. Irregular shaped voids form bonds as they close which then interact with the sub-voids created. This ramps up the closure process. Closure is highly dependent on how the creep interacts with the shape of the void.

Conclusions

The following conclusions can be drawn from this research:

- It has been shown that one can simulate diffusion bonding by starting with surface profiles of the base material, interacting them together, and applying the traditional bonding mechanisms of creep, surface diffusion, and boundary diffusion dynamically in a two-dimensional setting.
- Experimental diffusion bonding samples were prepared and the model was verified with that data. Numerical experiments were performed to determine the effects of different types of surfaces and voids on bonding performance.

- The results showed that the closure of most surfaces is dominated by power law creep. However, when the surfaces are fine enough and the voids are small enough the surfaces are dominated by the other diffusion mechanisms and not power law creep. Voids that are small enough to be diffusive dominated exhibit better bonding performance.
- It was also shown that it is more beneficial to have the voids concentrated into one area rather than evenly spread out over the surface.

Further research on this subject could entail looking into statistically quantifying bonding dependence on the surface profile average parameters, the relationship with void geometry and bonding time, as well as implementing this method for different materials.

6.4 Phase Field Model

Solid state diffusion bonding is a high temperature welding process commonly used in the aerospace and electronics industries. The process takes two chemically clean, usually metallic surfaces bonds them together at high temperature and elevated pressures. The chemical cleaning process removes any unwanted surface contaminants and oxides through the use of acid milling. The temperatures are usually 50-70% of the melting temperature and the pressures are high but not high enough to cause any dislocation accumulation in the material. The process has the benefit of producing near flawless welds. The primary defect in diffusion bonding are voids that are created by the difference between the microscopic irregularities of the mating surfaces. In the aerospace industry, diffusion bonding is commonly implemented with superplastic forming and used to create monolithic strong parts that save weight and cost. The subject of this research is Ti-6Al-4V, a frequently used titanium diffusion bonding alloy. The consensus among published literature is that voids close from a variety of diffusive processes and plastic deformation.

However, the process has only been modeled analytically in a way that decouples the physics from each other, assumes a void geometry and neglects the geometry of the surrounding microstructure. The models make many assumptions on how void closure progresses and how diffusion is applied. There has been no research into the effects of microstructure on bonding and the effects of crystal texture on bonding. The model in this research is proposed with the intent of investigating some of these effects.

Background

The diffusion bonding process is a high temperature energy minimization process by diffusion combined with plasticity. Traditionally the governing equations of diffusion bonding are derived from pressure sintering models [14, 15] or a reverse formulation of void formation in high temperature plasticity [16, 17, 113]. The bonding processes in these models is assumed to start with high stress plastic deformation at the tips of the micro-irregularities in the surfaces. The tips deform as they come in contact with each other under the bonding pressure. Deformation continues until the stress is low enough to no longer cause plastic deformation. The models then assume that the governing equations incorporate diffusion equations along the surface of the voids and at the grain boundaries. The bulk of the material not directly near the void is assumed to deform in the form of power law creep. The models assumed a void shape and analytically derive the listed closure mechanisms. More modern models still use these basic principles and have been derived for different void shapes [23] and for stochastic methods [24, 25]. The physical features of the microstructure in these models are as follows. The void shape is assumed to be a uniform function, commonly cylindrical or elliptical but sometimes lenticular. The diffusion along the surface of the void is a significant mass transfer mechanism and changes the shape of the void. It is assumed that the grain boundaries especially the grain boundary that is formed at the line where the two

sheets contact act as mass sources for the void. This mass transfer is thought to be a function of stress. The primary driving factor in bonding in these models is the power law creep mechanism which is a function of stress. By decoupling the physics from each other and assuming void shape and closure there are substantial assumptions

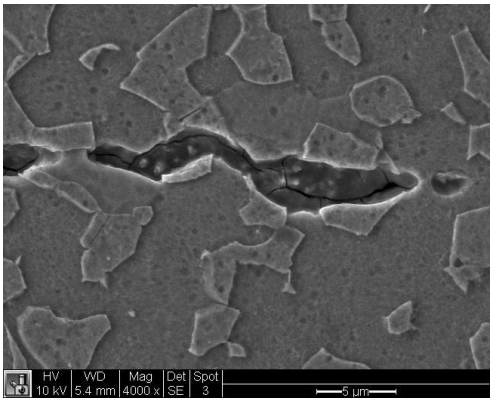


Figure 6.28 - Void location and microstructure in Ti-6Al-4V

As an energy minimization process, with significant similarities to powder sintering, diffusion bonding can be simulated with phase field methods. Phase field in combination with elastic strain energy can account for the coupling of the different physics involved and shed light on some of the more complex stress effects during void closure. While solid state diffusion bonding has never been covered with the phase field method to the authors knowledge, pressure sintering has been widely covered with this method [40, 41, 54, 55, 114]. Sintering is primarily applied as diffusion along the surface of a particle, diffusion along grain boundaries, bulk diffusion within a particle, rigid body motion of particles, and grain growth. Solid state diffusion bonding has similar kinetics and include all the diffusion effects listed above [15]. The primary difference lies in the treatment of bulk plastic deformation in the form of creep, the use of multiple phases in the polycrystalline framework, the inclusion of elastic strain energy, and the relatively complex initial microstructure.

It should be noted that there are several solidification phase field models that cover transient liquid diffusion bonding [115–118]. This is a fundamentally different process that requires melted material in between the two bonding sheets. The phase field models used in transient liquid diffusion bonding are based on melt solidification.

Ti-6Al-4V is a complex, binary phase alloy with significantly different diffusion parameters in each phase. The phase field method has been applied to titanium alloys. A multiphase, multi parameter model of the alloy has been developed [119]. This model uses scaling of the mobility and free energies to simulate bulk diffusion at large length and time scales. The model was used to investigate melt solidification and grain growth in titanium alloys [120]. The model was expanded to incorporate elastic strain energy and investigate the effects of precipitate equilibrium shapes [121]. This was used to simulate variant selection during cooling under elastic stresses [51, 52, 122]. Titanium alloys have not been investigated with the presence of voids by anyone to date.

Void treatment in phase field models have been done for a variety of situations. Voids in solid phases have been described for their mobility [123]. They have been modeled for their formation and mobility during dislocation climb [124]. Many phase field systems have been used to determine the creation and annihilation of voids during irradiation of nuclear materials [78].

Theoretical Model

The model presented in this research is a polycrystalline multi-phase field model to investigate the effects of void closure during diffusion bonding under the influence of elastic stresses. To do this, the phase field model is combined with a texture analysis to determine those effects on bonding performance.

Initial Conditions

The initial conditions for the simulation are a two dimensional approximation of experimentally determined microstructures. The microstructure is broken down into voids, an alpha-phase, and a beta-phase. The microstructural data does not include the crystal texture (the distribution of crystallographic orientations). The approximation scheme uses a variety of thresholds on an optical microscopic image. The image is discretized and seed points are placed on each grain. A threshold is taken to isolate the grain boundaries and the seed points are allowed to expand until they reach the threshold. Then another threshold is taken to isolate the voids. The grains are allowed to fill until the reach the voids. The result is an approximated microstructure. This microstructure is written as an EBSD file and imported into the finite element program. The mesh in the FEM program used is a square mesh with the dimensions that are the same as the microstructural image resolution. The initial conditions are the lack of material within the void and Ti-6Al-4V outside the void.

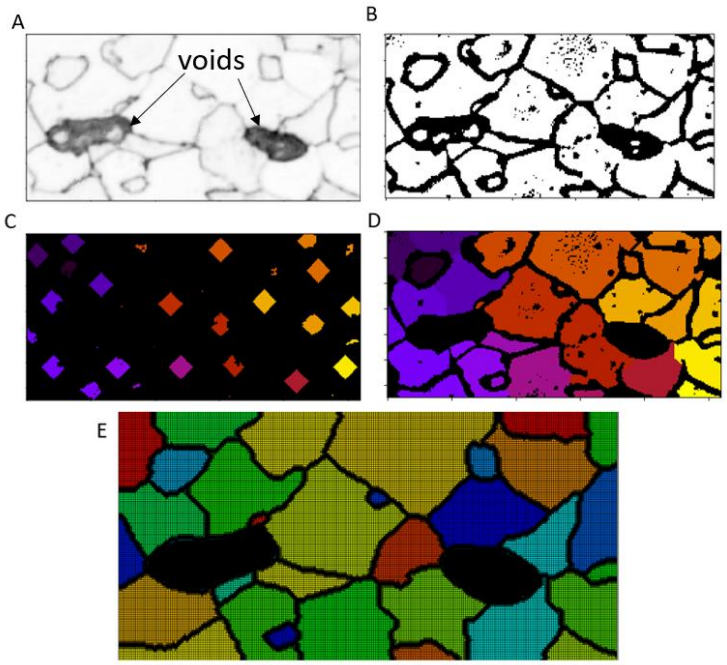


Figure 6.29 - Approximate geometry into MOOSE Framework. The optical microscopic image used for geometric information (A) is thresholded (B) which is used to restrict growth of seed points (C) until the reach the threshold (D). The threshold is removed except at the voids and the resultant meshed geometry is used for the initial conditions (E).

During solidification, titanium exhibits the burgers orientation relationship where the precipitating alpha phase is a function of the beta phase [125]. There are six {110} planes and two ⟨111⟩ directions on each of these planes giving a total of 12 different alpha variants based on the original beta orientations. While there are twelve orientations available, they are not equally occurring and depending on the conditions some will be more prevalent than others [122]. In Ti-4Al-4V, the variant selection has been extensively studied [53, 126] and shown to have some preference in thermomechanically processed sheets for different orientations [127]. The alpha phase has been shown to be qualitatively calculated from the beta phase orientations using 3 to 5 random orientations [128]. In this model, in the future, it will be assumed that the orientations of the beta

phase grains are randomized and the alpha phase grains have a texture that is a randomly calculated variant of the nearest beta phase grain.

Assumptions

There are several assumptions that go into this model when compared with historical models of diffusion bonding. I) The initial plastic deformation that happens when surfaces first contact each other is assumed to already have taken place. At these length scales modeling that deformation would require the inclusion of a crystal plasticity and rigid body motion which is beyond the scope of this current research. II) Power law creep is assumed to be accounted for by the diffusive processes modeled by the phase field equations. Deformation maps of titanium show that while there is some creep involved, it must be only at the tips of the voids where the stress concentration could be large enough to cause dislocations. III) The deformation is predominantly in the form of diffusive flow which is well accounted for by phase field [129]. The mobility used in this study is for the bulk phase. IV) It is assumed to be enhanced along the grain boundaries and surface of the void by an amount that correlates with the relative difference in referenced diffusion for pure titanium. Rigid body motion is usually included in both the phase field sintering models as well as the traditional diffusion boding models. In sintering, the grains come in contact under load and still have several degrees of freedom to consolidate. Phase field equations without rigid body motion would equate to having the grains diffuse across empty space to contact each other which is unphysical. In diffusion bonding, the voids are enclosed in solid material connected to a stiff and, relative to the void size, very large sheet. In the initial stages of bonding where the sheets come together and plastically deform the contacting points between the surfaces there is some rigid body motion. However, after the loads on the sheets are balanced assuming any rigid body motion

above the void would cause massive shearing stresses in the surrounding material. In this model, material can diffuse in from the boundaries to account for the material used to close the void.

Free Energy Formulation

The chemical free energy of a non-uniform system is formulated as a function of the order parameters [44]. G_m^p is the local molar free energy of each phase where p is the phases of α, β and the void phase v. c_i is the concentration of each of the elements where i represents the elements titanium, aluminum, and vanadium. The equilibrium free energies are a function of temperature, T, and the conserved concentrations C_i. η_j are the phase variables.

$$F = \frac{1}{V_m}\int_V \left[G_m^p(T, c_i, \eta_j) + \frac{\kappa_c}{2} \sum_{i=Ti,Al,V} |\nabla c_i|^2 + \frac{\kappa_j}{2} \sum_{j=1}^{N} \nabla \eta_j^2 + f_{el} \right] dV \tag{6-26}$$

In this simulation there are nine phase variables. One for the void phase and four for each of the alpha and beta phases. The double well potential height is ω_j while the double well potential function is $g(\eta) = \eta^2(1-\eta)^2$. The switching function that provides a blend of the phases is $h(\eta) = \eta^3(5\eta^2 - 15\eta + 10)$. κ_c is the gradient energy coefficient for the conserved variables while κ_j is the gradient energy coefficient for the order parameters.

$$G_m(T, c_i, \eta_j) = \sum_{p=\alpha,\beta,v} \sum_{j=1}^{N} h(\eta_j) G_m^p(T, c_i^p) + \sum_{j=1}^{N} \omega_j g(\eta) \tag{6-27}$$

The energy of the grain boundaries and surfaces are typically integrated over the physical dimensions of the phase field variables. This results in the situation where the one needs nanometer resolution to obtain the correct grain boundary energy and having such a dense mesh significantly

increases computation time. Kim-Kim-Suzuki (KKS) developed a method to resolve this issue which is implemented in this model. KKS decouples the interface dimensions and the interface energy with the physical dimensions of the model. This relaxes the constrain on the mesh size and allows for quantitative energy computation over larger dimensions. The grain boundary thickness and energy are determined by mapping those parameters to the double well potential and the gradient energy coefficients. The interface energy is γ, without the presence of elastic strain energy, and the boundary thickness is 2λ. These can be computed from the terms κ and ω.

$$\gamma = \frac{\sqrt{\kappa\omega}}{3\sqrt{2}} \tag{6-28}$$

$$2\lambda = (2.2)\sqrt{2}\sqrt{\frac{\kappa}{\omega}} \tag{6-29}$$

Strain Energy Formulation

In the presence of elastic deformation, the strain energy contributes to the interface energy between phases. While there are several different ways of computing the strain energy within the KKS framework. Examples include the Voight-Taylor scheme [130], the Steinbach-Apel scheme [131]), and the one used in this formulation which is the scheme used by Khachaturyan [132]. This provides an interpolation of the misfit strain energy as well as the stiffness tensors between phases. Khachaturyan interpolation has been shown to be more accurate than the other interpolation schemes [50]. Between phases, the elastic free energy becomes [133]

$$f_{el} = \frac{1}{2}C_{ijkl}\epsilon_{ij}^{el}\epsilon_{kl}^{el} = \frac{1}{2}C_{ijkl}\left[\epsilon_{ij} - h(\eta)\epsilon_{ij}^{*}\right]\left[\epsilon_{kl} - h(\eta)\epsilon_{kl}^{*}\right] \tag{6-30}$$

The elastic strain tensor ϵ_{ij} is modified by the interpolated misfit strain tensor ϵ_{ij}^{*} which is interpolated by the same switching function $h(\eta)$ used in the base model formulation. The stiffness tensor is additionally interpolated between the different phases. The elastic tensors in this simulation are a result of a base stiffness that is rotated to the grain orientation. The base stiffness is different for alpha and for beta. An example stiffness and von mises stress can be seen in Figure 6.30.

$$C_{ijkl} = \sum h(\eta) C_{ijkl}^{\eta} \tag{6-31}$$

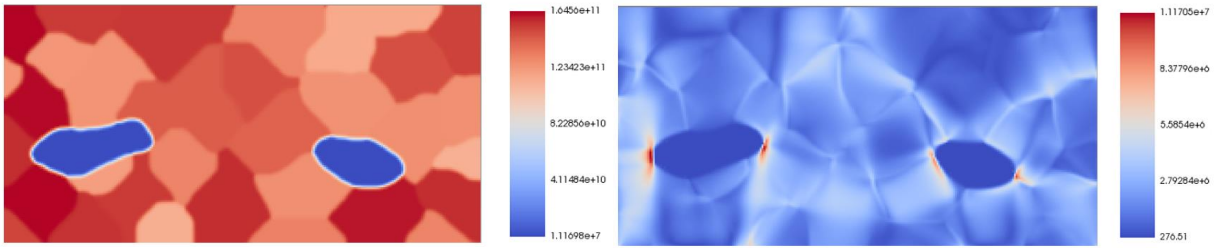


Figure 6.30 - The C_{1111} stiffness tensor (left) and the von mises stress (right). The voids are stress concentrations that result in relatively large stress, and therefore strain energy, at either edge of each void. The mesh size is 0.1 µm and the units are in pascals.

Kinetic Equations

The evolution of the conserved variables and order parameters are governed by the Cahn-Hilliard [134] and time dependent Ginzburg-Landau equations respectively. A multi-phase, multi-parameter model is used in order to describe the system. The mobility of each individual element is computed and effects the chemical potential of that element. The phase variables are constrained by Lagrange multipliers to enforce them to sum to 1 ($\sum_{j=1}^{n} \eta_j = 1$).

$$\frac{\partial c_i}{\partial t} = \nabla \cdot \left(\sum_{i=1}^{N} M_i \nabla \frac{\delta F}{\delta c_i} \right) \tag{6-32}$$

$$\frac{\partial \eta_j}{\partial t} = -L_j \sum_{i \neq k} \left[\frac{\delta F}{\delta \eta_i} - \frac{\delta F}{\delta \eta_k} \right] \tag{6-33}$$

The Kim-Kim-Suzuki model [47, 135] is implemented by constraining the chemical potential at each point to be a equal between the different sub phases. In this case, the subphases are the different elements in Ti-Al-V. These conditions are that each of the subvariables contribute to the bulk concentration by a function of the switching functions. In addition, there is a pointwise equality of the chemical potential of each of the subphases. The result is that there is no need to discretize the grain boundary because the energy is captured by the decoupling of the boundary energy from the physical dimensions.

$$c_i = h^{\alpha} c_i^{\alpha} + h^{\beta} c_i^{\beta} + h^{v} c_i^{v} \tag{6-34}$$

$$\frac{\delta F}{\delta c_i^{\alpha}} = \frac{\delta F}{\delta c_i^{\beta}} = \frac{\delta F}{\delta c_i^{v}} \tag{6-35}$$

Model Parameters

The individual element mobility M_i is a function of the phase variables and the local concentrations. In this model the diffusion parameters are computed for each timestep based on the local concentrations of the elements. The composition dependence of these elements are in the form of a Redlich-Kister polynomial [136]. The mobilities for each element are computed from diffusion studies for the alpha phase [137] and the beta phase [90]. The void phase mobility is a fraction of the alpha phase.

$$M_i^p = \frac{M_0}{RT} e^{-\frac{\Delta Q_i^*}{RT}}, \text{ Where } p = \alpha, \beta, v \tag{6-36}$$

$$\Delta Q_i^* = \sum_j c_j Q_i^j + \sum_r \sum_{j>r} c_r c_j \sum_k A_i^{rj} (c_r - c_j)^k \tag{6-37}$$

Where i, j, k are the primary, secondary, and tertiary elements

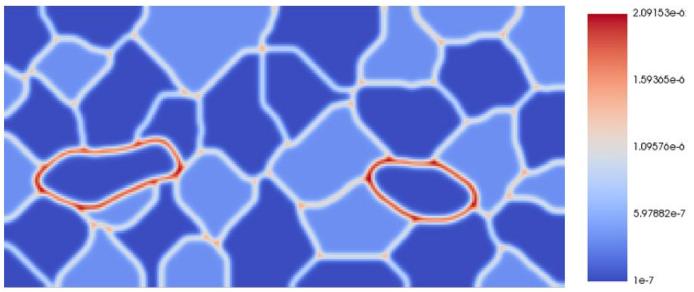


Figure 6.31 - Map of the mobility of titanium in $\frac{\mu m^2 mole}{J\ s}$. The mobility is enhanced along the grain boundaries and along the surface of the void. The mobility is lowest within the void and within the alpha phase.

The grain boundary energy is assumed to be $\gamma_{gb} = 0.5 \frac{J}{m^2}$ [119] and the values for κ and ω are selected accordingly to provide a grain boundary width 5 times the smallest mesh size. The surface energy is assumed to be $\gamma_s = 1 \frac{J}{m^2}$ and the variables are computed in the same way. The atomic volume is $V_m = 10^{-5} \frac{m^3}{mol}$. The effects of misfit strain are assumed to be included in grain boundary energy. The equilibrium Gibbs free energies of the different phases (G^α and G^β) are computed using the CALPHAD method from a ternary thermodynamic database [138] available and are functions of the temperature of the compositions of the respective phases. The free energy of the void is selected to be a function of the reduced presence of the sub elements but have the same chemical potential as the alpha phase. The void phase free energy is selected as a function of the

void phase subcomponents [123]. In the equation below k_B is the Stephan Boltzmann constant. The mobility and free energies are normalized.

$$G^v = k_B T \left[\left(1 - \sum c_i^v\right) \ln\left(1 - \sum c_i^v\right) + \sum c_i^v \ln\left(\sum c_i^v\right) \right] \tag{6-38}$$

Grain Growth Calibration

A wide range in the interface mobility has been shown in the literature [51, 119]. In order to be sure that this model replicates experimental reality a method of calibrating it by using grain growth data is demonstrated here. An arbitrary microstructure was generated using Voronoi tessellation and random seed points that correlate with the grain density within Ti-6Al-4V. The grains are subjected to the KKS model in order to determine the interface mobility. The base grain size was an average of $d = 6.1\ \mu m$. Grain growth follows a power law function as shown in Equation 6.39 [8]. The equation has d as the grain size, d, and a and b as temperature dependent material properties. The parameters for the grain growth model are taken from [139] and are $a = 0.128 \times 10^{-16}$ and $b = 5.0$.

$$\frac{\partial d}{\partial t} = a d^{-b} \tag{6-39}$$

The experimental results from the paper and the calculation match at a growth of 83.2% over the course of 410 minutes. A simulation time of 100 minutes is used and a calibration area of $25\mu m \times 25\mu m$ which should result in an increase in diameter of a grain to $8.96\mu m$ and a reduction of the initial 25 grains to 11.5 grains. The interface mobility, L, is selected to fit the grain growth model.

The model is implemented using Idaho National Lab's MOOSE software. It is a versatile opensource multiphysics program that can be customized to fit a variety of requirements [78]. The model takes advantage of preinstalled adaptive meshing algorithms to provide a moving dense mesh along the grain boundaries, grain tracking reduction of order parameters, and several solution methods. The simulation is solved using preconditioned Jacobian Free Newton Krylov methods with a boomerang HYPRE preconditioner developed by Lawrence Livermore National Lab and implemented in PETSC [140].

Results and Discussion

The model computed for diffusion bonding showed that under this microstructure the closure occurred as shown in Figure 6.32. The microstructure evolved by reduction of the void and by grain growth. The change in void size is a primary function on the reduction of the surface energy around the void and the diffusivity of the various elements. The grain growth is a function of the grain boundary energy and the diffusivity. The diffusion is enhanced along the grain boundaries and along the surface of the void. All the elements, but especially titanium, have a much lower diffusivity in the alpha phase than in the beta phase. The mobility of the alpha phase is roughly 40,000 times lower than the beta phase in titanium. Therefore the elements have trouble diffusing through the alpha phase. This is why, in Figure 6.32 one can see the grain boundaries in the alpha phase but not in the beta phase. The figure also shows that aluminum is an alpha stabilizer and is concentrated in the alpha phase. While vanadium stabilizes beta and is concentrated in the beta phase. This is a well known phenomena and discussed in Chapter 2 of this dissertation. When the voids are removed the grains combine to form a double grain where the void used to be. This simulation was done at a temperature of 1025K and 2MPa pressure. For this temperature, time to complete bonding is 40 minutes. In Figure 6.32 each column represents a percentage of the

simulation time increasing from left to right. The initial conditions on the far left have Ti-64 everywhere except the voids. This diffuses in the next column (T=30%) to partially close each void all while the alpha and beta stabilizers diffuse into their respective phases. In the next column (T=60%), a single void is left since the other has been closed. Finally, the simulation finishes with both voids closed.

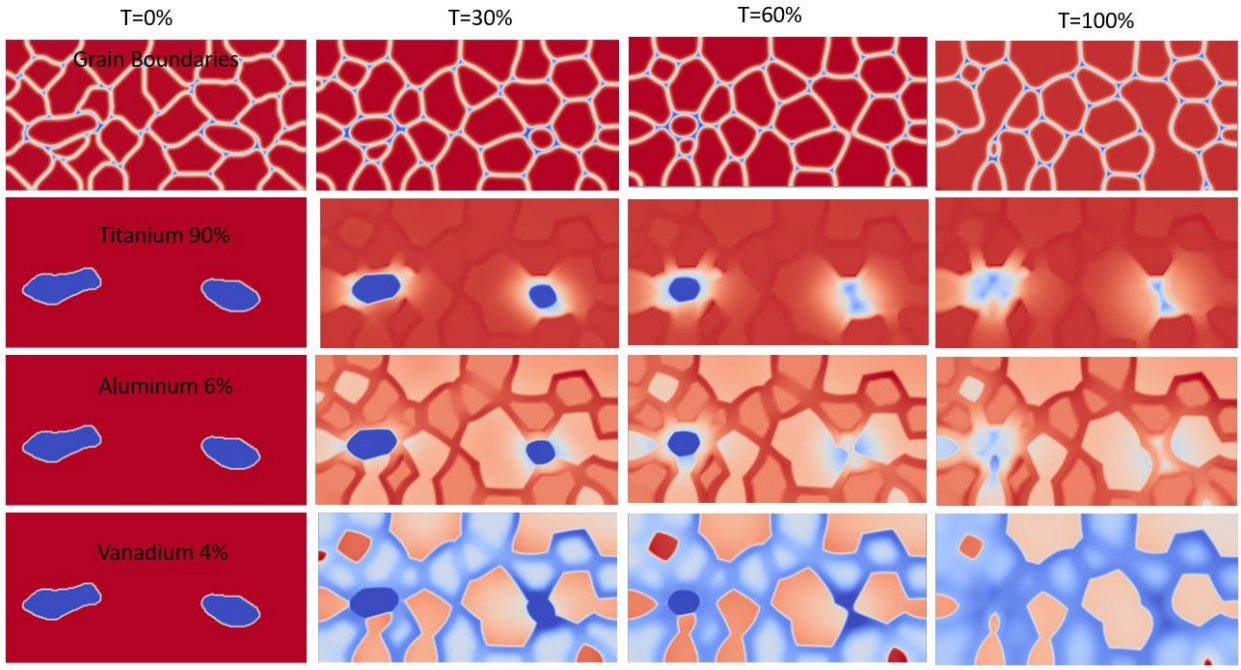


Figure 6.32 - Bonding of Ti-6Al-4V. The columns are the bonding time as a percentage of the simulation run (0%, 30%, 60%, and 100%). The rows represent a map of the grain boundaries (top), and titanium, aluminum, and vanadium descending. The scale of each is 0 to the initial element percentage. Red is the upper bound and blue is the lower bound.

The presence of alpha and beta directly adjacent to the void have a significant effect on the void progression. This results in a change in the void shape due to the presence of that material. The diffusion that would normally occur surrounding the void is significantly reduced thereby restricting void shape. However, since there is some contact with the beta phase and there is a significant amount of grain boundary diffusion and surface diffusion the material is transferred

from the beta phase to the alpha phase. In fact, the closure simulation will not function properly without enhanced diffusion along the void surface since the majority of shape change in the void results from this diffusion. The effects of alpha and beta around the void can be seen by adding or removing surface diffusion. Figure 6.33 shows the difference between having surface/grain boundary diffusion and not having it. Without surface diffusion the only diffusion into the void comes from the beta phase. The alpha phase is relatively locked. Diffusion and void closure will occur but not in the same timeframe. Therefore, the majority of diffusion into a void comes from having beta present at the void to provide a diffusive source which then moves along the void surface. This means that there could be significant changes to the bonding performance if there is a large alpha concentration around the voids or if there is significant contamination at the void surface.

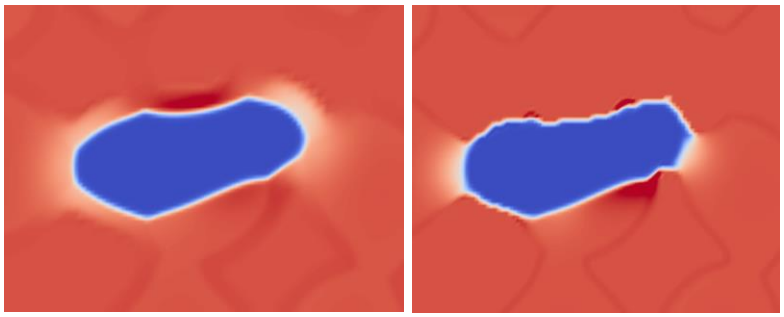


Figure 6.33 - Difference between having surface and boundary diffusion (left) and not having surface or boundary diffusion (right). Since the mobility is so much lower in the alpha phase void closure requires surface diffusion in order to relocate the material transferred into the void from the beta phase. This is sufficient when there is enough beta present. Without diffusion along the surface of the void, diffusion only occurs from the beta phase and the void will not close.

Conclusions

A diffusion bonding model has been developed from the energy based multiphase phase field method. This method incorporates the physics of diffusion bonding much more accurately than

conventional models. Future work would be to confirm it with more experimental data. It can also be used to determine some of the more complex effect in diffusion bonding such as an analysis of different processing conditions and an analysis of the effects of crystal texture.

6.5 Superplastic Modeling

Titanium is a well-established superplastic material. It is frequently used in the aerospace industry to produce parts. However, the mechanisms of superplasticity are debatable. Superplastic forming is done at high temperatures and low strain rates and, when done properly, results in a part that has effectively no void or residual stresses. Therefore, the mechanisms have been traditionally based on investigations into the microstructures after the forming process has been completed or deductions from changes in the bulk behavior of the material making it difficult to determine the underlying mechanisms. There have been very few simulation investigations into these mechanisms. The consensus is that superplasticity is primarily due to grain boundary sliding. However, it cannot be grain boundary sliding alone due to the fact that grain boundary sliding produces a significant number of voids in the material. The accommodation mechanisms that close these voids are thought to be Lifshitz sliding (diffusion based), Rachinger sliding (dislocation based), or a combination of both [7]. The purpose of this research is to determine the underlying accommodation mechanism in superplasticity but performing a coupled phase field diffusion, molecular dynamics grain boundary sliding, and plastic deformation simulation.

There has been a wealth of research on the mechanisms of superplasticity in metals. Experiments have been carried out to determine the high temperature fracture behavior, void growth behavior, and creep mechanics. Grain boundary sliding occurs at high temperatures at certain strain rates. In the diffusion accommodated Lifshitz sliding the sliding is explained as a natural effect of high

temperature diffusion in response to an external stress. On the other hand, the dislocation accommodated Rachinger sliding assumes that the movement is based on grain displacement without significant changes in the grains overall shape. Lifshitz sliding can occur at larger grain sizes while Rachinger sliding is mainly applicable to small grain sizes. However, there is some debate in T-64 which mode is operating and when and the effect at the triple points [141].

Experimentally there have been many papers published on both of these accommodation mechanisms. There have been two different ways to determine the effects experimentally [142]. The mechanisms are separable by looking at the bulk deformation since they both give different deformation rates . In situ surface observations have also been done where the material superplastically deformed under a microscope in order to determine which mechanisms is operating when [7]. The flaw with both of these methods is that there isn't an understanding of the mechanisms operating within the material and all information must be inferred from either the free surface or the bulk. The free surface will deform differently since it is unconstrained than the bulk material. However, there is no experimental as of yet to examine the internal structure during superplasticity.

Grain boundary sliding simulations have been used to gain information about the material behavior. Constitutive equations that map the bulk deformation with the mechanisms have been used [129, 142]. This can be useful when looking at the deformation and determining which mode is most likely operating.

Theoretical Model

Formulation

The model presented in this research is a polycrystalline multi-phase field model to investigate the effects of void closure due to diffusion, grain boundary sliding, and some simplistic plastic deformation. The goal of the simulation is to determine how much of the deformation of the material is due to each of these mechanisms. In order to do this, a 3 phase, 3 element, 8 order parameter phase field model is presented based on Ti-Al-V. Grain boundary sliding is estimated using molecular dynamics simulations. The molecular dynamics sliding information is incorporated into an elastic plastic simulation where the deformation behavior is modified at the nodes along the grain boundaries. The simulations alternate between each other by passing information to one another. The phase field simulation diffuses the phase variables which change the location of elastic and plastic constants used in the mechanics simulation. The mechanics simulation provides a deformation that is then passed into the phase field model via a conserved advection kernel along with strain energy that added to the bulk free energy.

Initial Conditions

The initial conditions for the simulation is a polycrystalline Voronoi tessellation map of a crystal structure than has been diffused for a certain amount of time. The effective grain size corresponds to the typical 2-3 um that is referenced for fine grain titanium. Since there will be a large amount of deformation in this simulation the boundary conditions are of upmost importance. They must be able to move large amounts while continuing to retain the grain structure of the material. Therefore, a moving front grain is used on all sides that has the elastic behavior of alpha titanium but has no diffusion attached to it and moves with the material to accommodate the deformation.

Free Energy Formulation

The chemical free energy of a non-uniform system is formulated as a function of the order parameters [44].

$$F_{chem} = \frac{1}{V_m}\int_V \left[G_m(T, c_i, \eta_j) + \frac{\kappa_j}{2}\sum_{j=1}^{N} \nabla\eta_j^2 + f_{el} \right] dV \tag{6-40}$$

G_m^p is the local molar free energy of each phase where p is the phases of α, β and the void phase v. c_i is the concentration of each of the elements where i represents the elements titanium, aluminum, and vanadium. The equilibrium free energies are a function of temperature, T, and the conserved concentrations C_i. ϕ_j are the phase variables. In this simulation there are 9 phase variables. One for the void phase and four for each of the alpha and beta phases. The double well potential height is ω_j while the double well potential function is $g(\eta) = \eta^2(1-\eta)^2$. The switching function that provides a blend of the phases is $h(\eta) = \eta^3(5\eta^2 - 15\eta + 10)$. κ_c is the gradient energy coefficient for the conserved variables while κ_j is the gradient energy coefficient for the order parameters.

$$G_m(T, c_i, \eta_j) = \sum_{p=\alpha,\beta,v}\sum_{j=1}^{N} h(\eta_j) G_m^p(T, c_i^p) + \sum_{j=1}^{N} \omega_j g(\eta) \tag{6-41}$$

The energy of the grain boundaries and surfaces are typically integrated over the physical dimensions of the phase field variables. This results in the situation where the one needs nanometer resolution to obtain the correct grain boundary energy and having such a dense mesh significantly increases computation time. Kim-Kim-Suzuki (KKS) developed a method to resolve this issue which is implemented in this model. KKS decouples the interface dimensions and the interface energy with the physical dimensions of the model. This relaxes the constrain on the mesh size and

allows for quantitative energy computation over larger dimensions. The grain boundary thickness and energy are determined by mapping those parameters to the double well potential and the gradient energy coefficients. The most important parameter in performing accurate quantitative simulations in the KKS framework is using the correct grain boundary and surface energies. The interface energy is γ, without the presence of elastic strain energy, and the boundary thickness is 2λ. These can be computed from the terms κ and ω.

$$\gamma = \frac{\sqrt{\kappa\omega}}{3\sqrt{2}} \tag{6-42}$$

$$2\lambda = 2.2\sqrt{2}\sqrt{\frac{\kappa}{\omega}} \tag{6-43}$$

Strain Energy Formulation and Grain Boundary Sliding Mapping

Sliding produces both plastic deformation in the bulk of the material as well as elastic. The plastic deformation stores the energy in the form of dislocations while the elastic is immediately recoverable in the form of strain energy. In this simulation the plastic deformation energy is neglected while the elastic strain energy contributes to the bulk energy used in the phase field equations. While there are several different ways of computing the strain energy within the KKS framework but they predominantly rely on estimating the strain energy between two different phase variables. The elastic free energy becomes [133]

$$f_{el} = \frac{1}{2} C_{ijkl} \epsilon_{ij}^{el} \epsilon_{kl}^{el} \tag{6-44}$$

The elastic strain tensor ϵ_{ij} is coupled with the stiffness tensor to compute the energy. The stiffness tensor is additionally interpolated between the different phases.

$$C_{ijkl} = \sum h(\eta) C^{\eta}_{ijkl} \tag{6-45}$$

During the interpolation however, the stiffness at the midpoint node between the two different phases is replaced by the temporal grain boundary sliding deformation. This term is the linear fit of the grain boundary sliding rate. In other words, it is the grain boundary sliding determined by the molecular dynamics simulation is incorporated into the equation above. The details of this calculation are shown in Figure 6.34.

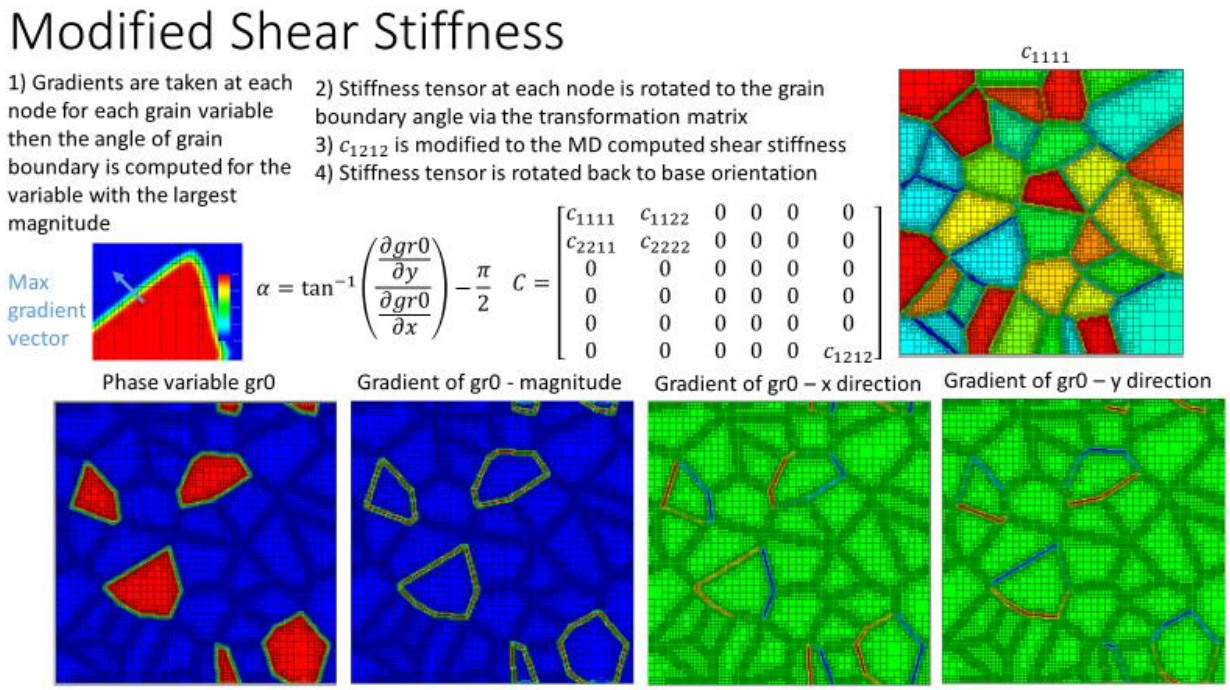


Figure 6.34 – Modification of the shear stiffness

Kinetic Equations

The evolution of the variables are the time dependent Ginzburg-Landau equations. A multi-parameter model is used in order to describe the system. The phase variables are constrained by Lagrange multipliers to enforce them to sum to 1 ($\sum_{j=1}^{\eta} \eta_j = 1$).

$$\frac{\partial \eta_j}{\partial t} = -L_j \sum_{j \neq k} \left[\frac{\delta F}{\delta \eta_j} - \frac{\delta F}{\delta \eta_k} \right] \tag{6-46}$$

For molecular dynamics, the method used is the embedded atom method. This uses the pairwise potential between atoms mapped from first principle quantum mechanics calculations. The

$$E_i = F_\alpha \left(\sum_{i \neq j} \rho_\beta (r_{ij}) \right) + \frac{1}{2} \sum_{i \neq j} \phi_{\alpha\beta}(r_{ij}) \tag{6-47}$$

In this equation, r_{ij} is the distance between atoms i and j, $\phi_{\alpha\beta}$ is the pair-wise potential, ρ_β is the electron charge contribution, and F_α is an embedding funciton.

Implementation

The model is implemented in Idaho National Lab's MOOSE software with the same solution mechanisms as the phase field simulation described in this dissertation.

Initial Results

While the model is not fully implemented yet, some of the results of each individual model can be shown. The molecular dynamics model produces results in the form of the stress relative to a sliding velocity. This is a function of the materials and the grain orientation along with temperature and sliding velocity. A plot of the stress results is in Figure 6.35.

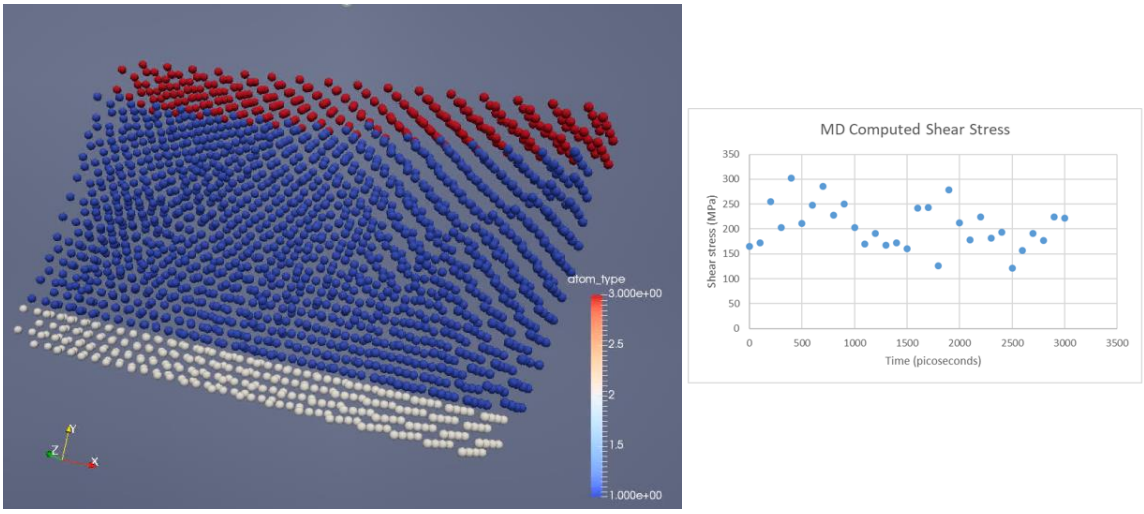


Figure 6.35 – Molecular dynamics mapping of the edge shear stress

The grain sliding calculation shows the coupling of Rachinger and Lifshitz sliding. The effects of each are based on the relative amount of diffusion occur versus grain slip. At the moment, there is no void formation or implementation of the multi-order parameter model. However, one can see the movement of the grain. The grains flux out of the simulation area due to the movement of Rachinger sliding. A plot of a uniaxial tension test with both sliding types is in Figure 6.36.

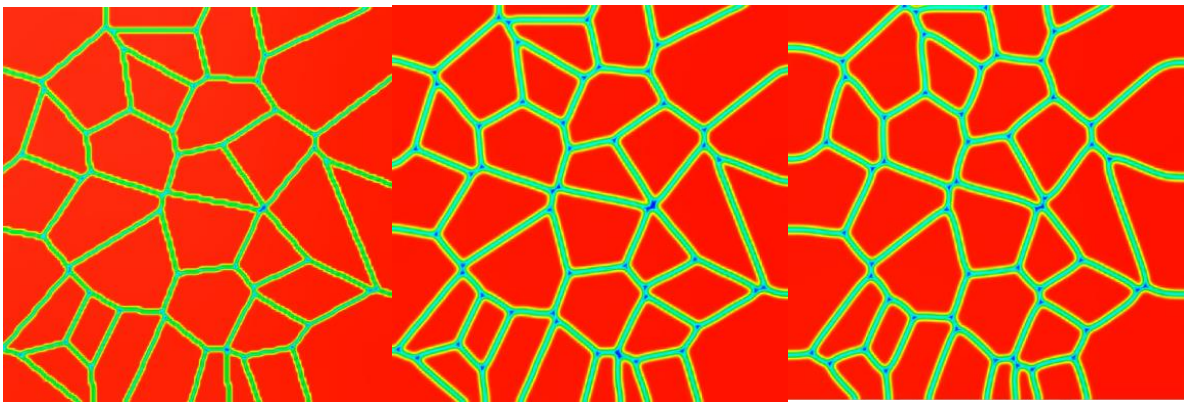


Figure 6.36 – Evolution of the grain boundaries under both sliding mechanisms. The edge grains are the easiest to see the movement.

7 Discussion

7.1 Introduction

Advancements in diffusion bonding and superplasticity have been relatively slow since the establishment of the base technology. The evolution of theoretical models has not had significantly new modeling methods applied to the process and experiments have remained largely unchanged and inadequate at explaining the complex processes going on. The theoretical models have been incapable of replicating experiments accurately and the experiments have elucidated only the primary process parameters for individual materials. This is primarily due to limitations of the processes. For diffusion bonding, surfaces on this scale are inherently irregular and the voids formed from them are random and close surrounded by effectively random microstructures. This makes it incredibly difficult to determine how voids closed exactly by looking at them after bonding has completed. There is no current technology to examine the void surfaces as they close, and experiments are based on post-process examinations. Superplasticity has a similar problem and while there have been some insitu microscopy, one can only look at the free surface and all data is under in the influence of edge effects. The primary difficulty with both of these processes is the high degree of variability combined with the inability to examine the process on the scale that is occurring as it is proceeding. So far, one can merely infer from what we know of the physics involved and down select the best time temperature and pressure/strain rate through experiments.

The objective of this chapter is to tie the different sections together, discuss how they combine to fit into the current literature, and to present some additional insights. The goal of this research was to characterize superplastic and diffusion bonding samples and use them to understand the process better through experiments and theoretical models. The experimental alloys used in the

two sheet diffusion bonding, along with the structural testing involved in the two sheet process, was new for those materials. There are also no details published on the effects of dissimilar diffusion bonding and alloy element migration provided by the EDX linescans and how this could affect the bonding process. Structural and non-destructive testing have been explored but unsuccessfully in the past and the work presented here helps to understand the process. The four-sheet forming process has very little literature discussing it and the data collected from it is essentially undocumented publicly. Nor have the effects of superplasticity on diffusion bonding been documented. While there have been many models created for diffusion bonding for a variety of materials, the analytical equation based models created in this work have been done to show the effect of surface geometry on the process. Something that despite being a process parameter has been relatively poorly understood. The phase field model was created to update the physics to a more modern formulation and was new for this topic. The details of each model and experiments have been discussed in the respective sections. The relative contributions of sliding mechanisms in superplasticity have been a contentious topic and modeling this can help better understand superplasticity. All of this work contributes to the current published knowledge.

7.2 Superplasticity and Diffusion Bonding

There are some insights that one can make about the processes that could be investigated further.

7.2.1 Effects of Dissimilarity in Bonding

The two sheet diffusion bonding samples had many combinations of dissimilar alloys. While these alloys are all predominantly titanium, they differ in the alloying elements and have slightly different bonding properties. An investigation was done into whether or not dissimilarity has an affect on bonding performance. According to the EDS analysis in Chapter 4 there is a significant

amount of diffusion of the alloying elements when there are two different alloys present. It is possible that this difference allows for enhanced bonding of the two alloys. By looking the relative bonding performance of each of the different alloys one can determine if there is a significant effect from dissimilarity. This is done here by assigning a point value to the amount of bonding and summing up the similar and dissimilar alloy samples. Assigning bonding points of "No voids"=0, "Few voids"=1, "Moderate voids"=3, "Extreme voids"=5, and "Extreme voids and delamination"=6. First, to check to see if the method makes sense, the well established temperature correlation is computed. Summing all the bonding points that correspond to a temperature group, shown in Figure 7.1a, gives the most voids at the low temperatures, less voids at moderate temperatures, and the least at high temperatures. Therefore there is some value to this analysis. With respect to dissimilarity, the average bonding points per sample is computed (since there are more dissimilar than similar samples). The dissimilar samples had only a slightly more voids than the similar as seen in Figure 7.1b. Therefore there may not be as strong of an effect of dissimilarity but more of an independent effect of the individual materials. Additionally, the surface tracking model discussed in Chapter 6 showed that theoretically the material will not fill in voids formed by the opposite sheet. While there is some extra diffusion going on with dissimilar alloys it is still a predominantly up to each sheet to fill the voids formed on its side of the bondline. This is a possible reason why the effects of dissimilarity among alloys of the same material could be negligible.

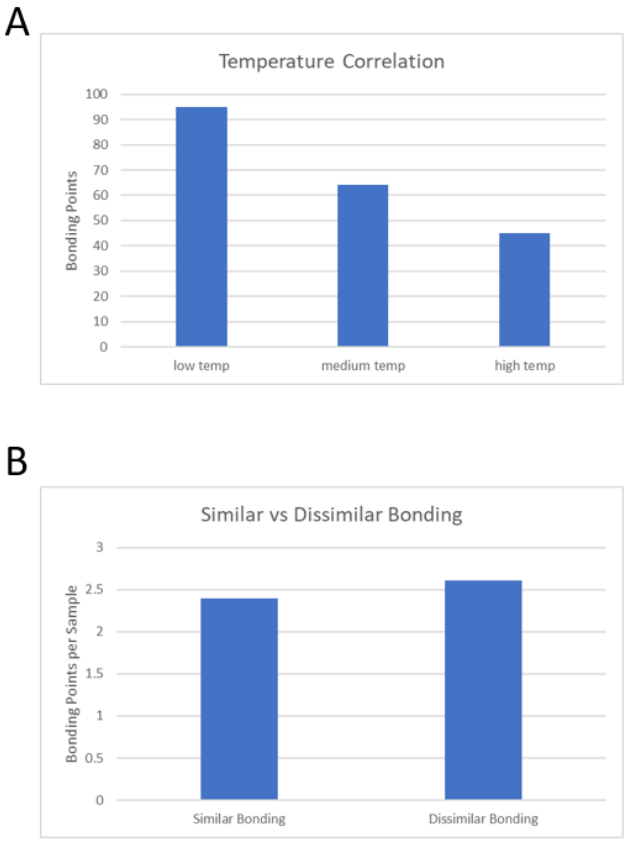
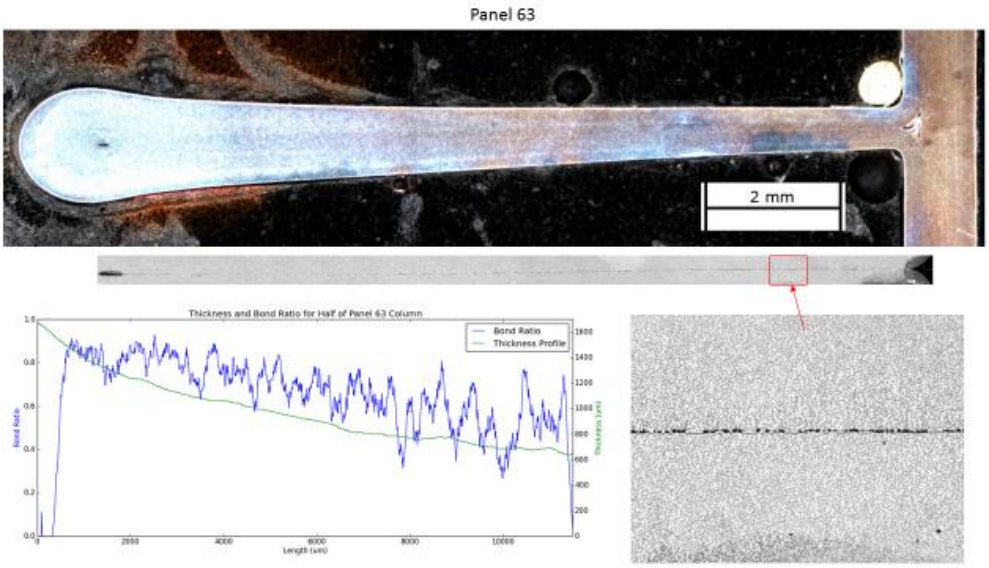


Figure 7.1 – The correlation of voids with temperature (a) and the correlation with dissimilarity (b).

7.2.2 Bonding Quality vs Thickness

There is a possible correlation between bonding quality and thickness of the sample which was investigated. Two panel columns were selected for this analysis and micrographs were taken all along one half of the column section at 200x. These micrographs were composited with photoshop and then the same bondline program used to detect voids in the other images was used to determine the bond ratio trend along the panel. The bond ratio is defined as the percentage of bonded material if one took a standard micrograph (500x magnification, around 180um) at that specific location. The thickness distribution was already calculated from the program. The micrograph location and

the thickness were correlated by using the rathole dimensions as the beginning and end of each. While the trend of thickness vs bonding was not entirely conclusive, it is important to note that depending on where you take the micrograph you could get a bonded percentage of anywhere between around 30% bonded to 90% bonded on panel 63 and around 35% bonded to 98% bonded on panel 62. While it is unfeasible to measure this distribution for each sample prepared it is useful to note that there is both a major trend in the bond ratio as well as some variance. This is a more dramatic trend than a diffusion bonding coupon would be without any superplasticity. However, it does show that in the center of the specimen one can get a fairly decent representation of the bonding quality.



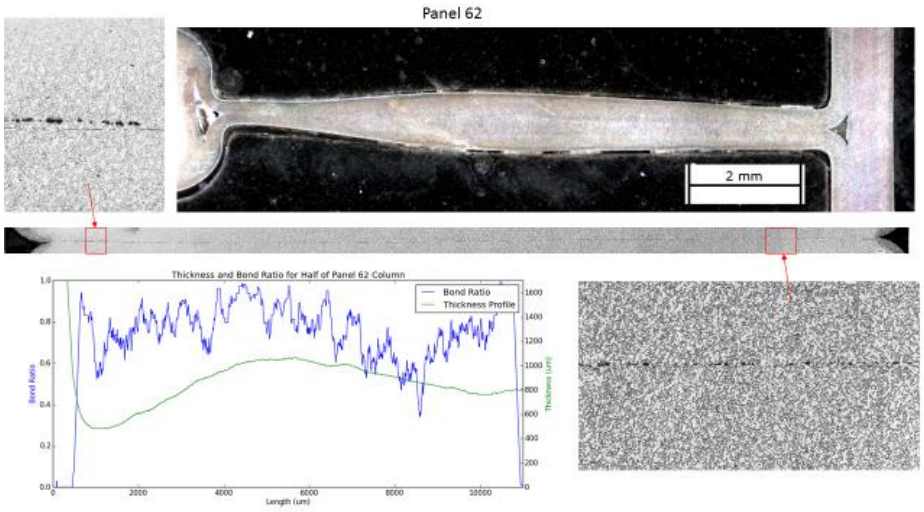


Figure 7.2 - Superplastic forming vs diffusion bonding trends

7.2.3 Void Size Distribution

One of the benefits of using a program to calculate the void sizes is that one can run statistics on the void length to see if there is any correlation between the surface roughness and the voids. Void sizes are one of the more important factors in the bonding. If voids are similarly sized and well distributed, then it really isn't an issue structurally. However, if voids are concentrated in a certain area then they will be significantly more of a problem. Running statistics on the calculated bonding can give insight into the number of outliers and the maximum size of voids in the process. This is a lead up to determining more information on the shape of the voids.

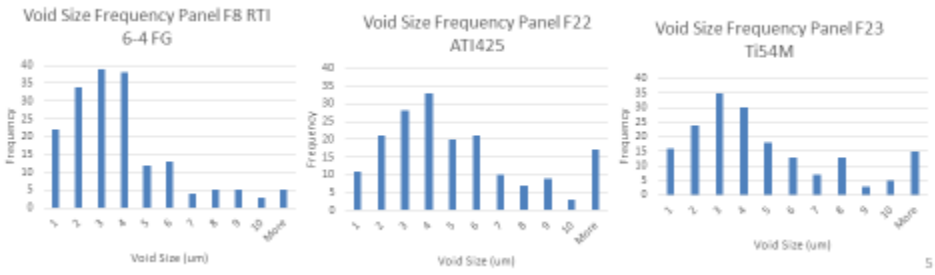


Figure 7.3 - Void width distributions for a variety of materials

7.2.3 Gas Pathway Statistics

The gas pathways are one of the primary defects in the panel and can be a significant source of structural weakness if they are not formed correctly. They also can vary significantly along the length of the cell since superplasticity is a somewhat chaotic process there can be isolated very large pathways and ones that never opened at all. Since they represent a free edge in the forming they are also a good indicator of how uniform the forming is along the panel. To gain some estimation of how the pathways are shaped and they have been detected on some select panels by again using image processing. This was done by chroma key, where you take a uniformly colored background and isolate that color digitally. Then the unique pathways are isolated with an openCV connected components algorithm. This takes a binary image and sorts through all the pixels and then does a check if they are a value of 1 and if they are next to another pixel with a value of 1. Then it gives that pixel a group value of the same value as the adjacent pixel. This way the entire image is grouped by connections. With each pathway marked individually, one can measure the length and width and aspect ratio using the number of pixels long an axis and a scale (from the measured weld nugget).

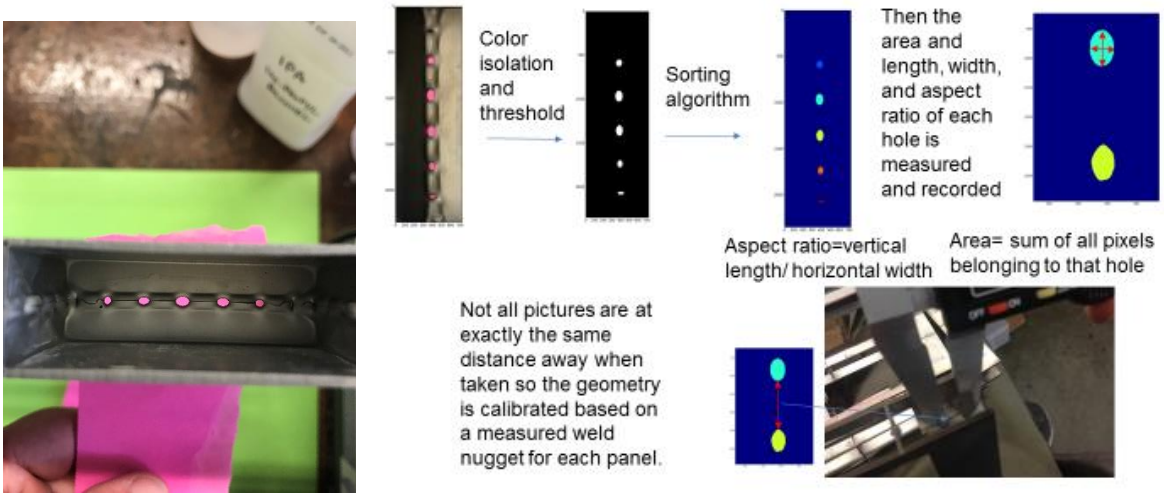


Figure 7.4 - Gas pathway detection imaging (left) and detection program explanation

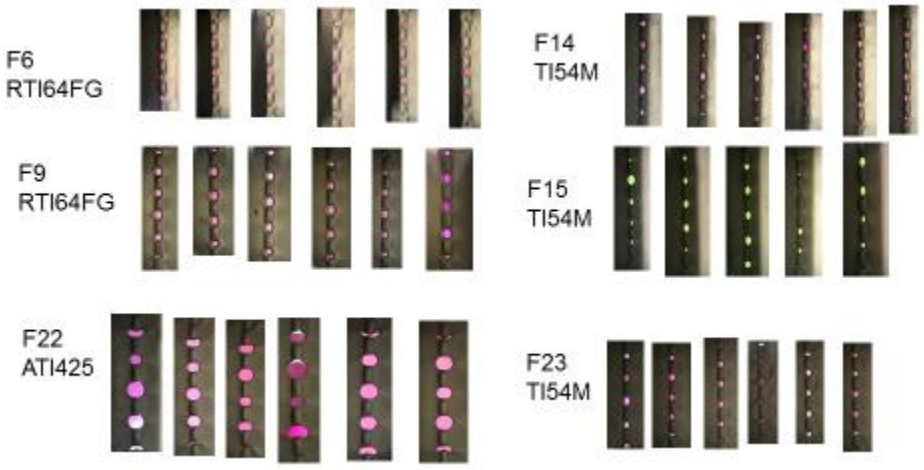


Figure 7.5 – Dataset

The results of the analysis were that Ti54m had the smallest pathways and were the most elongated while ATI425 had largest pathways by a significant margin. This data will be used as inputs and comparisons for the superplasticity models.

Table 7-1 - Gas pathway statistical results

Panel	Core Sheet Material	Area Average (mm^2)	Area Standard Deviation (mm^2)	Aspect Ratio Average	Aspect Ratio Standard Deviation
F6	RTI64FG	2.10	1.08	1.23	0.33
F9	RTI64FG	2.01	0.92	1.20	0.28
F14	TI54M	1.35	1.16	1.58	0.42
F15	TI54M	1.35	1.16	1.58	0.42
F22	ATI425	6.00	3.95	0.67	0.22
F23	TI54M	1.54	0.88	1.07	0.50

7.2.4 Influence of Superplasticity on Diffusion Bonding

Diffusion bonding seems to perform significantly worse under the influence of superplasticity. This is very likely due to the rotation of grains on the free surface, as seen in Figure 2.7, causing a rougher surface combined with the fact that diffusion bonding has a strong reduction in bonding performance based on the surface roughness. A secondary effect is that the bonding performance is somewhat dependent on grain size as well. Superplasticity also increase the grain size with increase in strain. The combination of the two effects are what results in Figure 7.6, where the two idential initial surfaces are subjected to the different strains involved in the four-sheet forming process and result in vastly different bonding. The bottom surface has a larger superplastic strain

and has a much rougher profile. While the top surface had a very low superplastic strain and a profile much closer to the initial profile before forming.

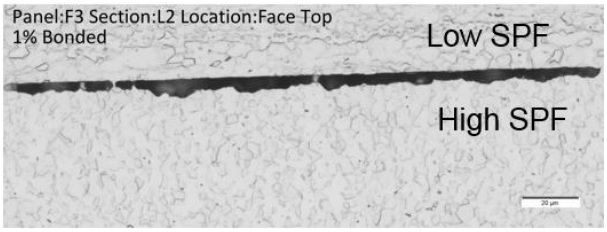


Figure 7.6 - Changes in surface roughness due to superplasticity for the same material

One can see this in a roughening of the material as a result of superplastic deformation. A rudimentary study on the roughening behavior of superplasticity can been seen in Figure 7.7 for the profile location and Table 7-2 for the changes. The location of the profile was taken because that section would have been exposed to argon during forming and exposed to the full superplastic strain. A significant increase in surface roughness was observed.

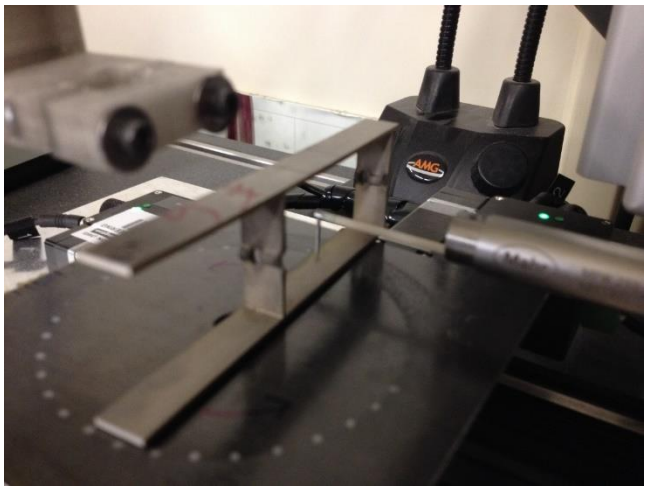


Figure 7.7 - Surface profile location when testing the roughening during superplasticity.

Table 7-2 - Changes in roughness parameters due to superplasticity

Material	PreSPFDB Ra	PreSPFDB Rz	PostSPFDB Ra	PostSPFDB Rz
54m T	0.619506	2.876042	1.003	11.003
VS FG T	0.475488	4.044442	0.915	6.853
VS FG T 2	0.475488	4.044442	1.783	16.703
RTI FG T	0.28702	1.978406	0.873	7.957

To determine the necessary steps to counter these effects Boeing produced several panels in the form of a temperature study. The purpose of the study was to determine the increase in temperature necessary to achieve good bonding despite the effects of superplasticity on bonding. The panels were made from RTI 6-4 FG core sheets and TIMET 6-4 SG face sheets. The pressure was set at 550 psi. The temperatures were selected at 1475F, 1500F, 1525F, and 1550F. The results of the study showed that there was a significant increase in energy required to bond panels after superplastic forming. Near perfect bonding was achieved at 1550F. Therefore temperature plays an important role in good bonding.

Table 7-3 - Temperature study results. The study determined that by raising the temperature up to 1550F one can achieve nearly complete bonding.

Panel	S1 BC	S1 TC	S1 BF	S1 TF
1475F	63%	41%	78%	50%
1500F	83%	87%	87%	83%
1525F	82%	87%	80%	91%
1550F	100%	95%	95%	97%

7.2.5 Analysis and Composition of Voids

In an effort to understand more about the four-sheet process and void closure, several samples were analyzed for their composition around the voids. These were from a four-sheet sample called

F24 which was composed of 54M-54M. Several voids were taken for analysis and were examined in a way similar to Figure 7.8. It was to be expected that there would be some contaminates. Evidence of particulates, such as aluminum oxide from sanding or carbon dust were in several voids. However, the majority were clean and particle contaminant free. The interesting thing that was observed was that there was frequent evidence of large amounts of a titanium, fluorine, and oxygen compound near the surface of the void. It is unknown how this could affect the bonding performance but would benefit from further research.

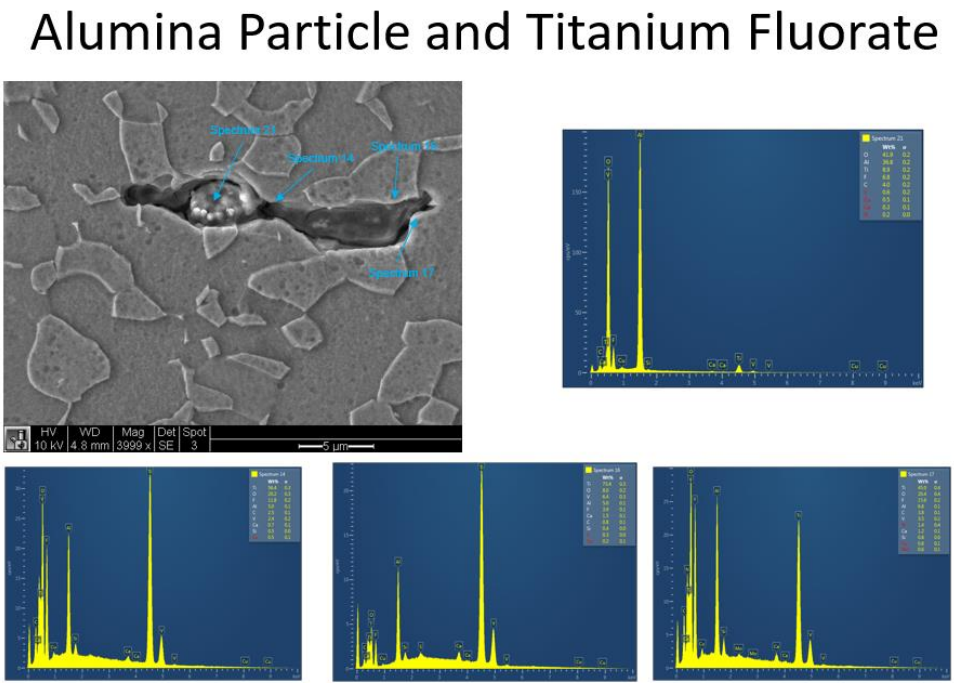


Figure 7.8 - Looking at the void there can be particle contaminants and chemical contaminants. In the void above, there is a residual aluminum oxide particle and the fluoride in the water and etchant have created a compound

8 Summary, Conclusions, and Recommendations for Future Work

Superplastic forming and diffusion bonding are complex topics and have been researched for many years. They are both highly dependent on the processing conditions as well as the properties of the materials used. In the past researchers have examined these processes through a variety of both experimental and numerical methods. This work expanded on the current available scientific knowledge by examining two sheet diffusion bonding as well as the four-sheet SPF/DB process through a combination of experiments and simulations. This work had the following contributions and conclusions:

- This research characterized two sheet diffusion bonding samples and demonstrated the performance of many different unexplored materials through the micrographic characterization of bonding, structural tests, and EDS diffusion studies. Structural testing was performed on bonding samples and new tests were developed through the use of finite element simulations. These simulations showed the difficulties of developing a bonding test and how to do so effectively.
- Four-sheet SPF/DB samples were analyzed and characterized for their superplastic and bonding performance through the use of computer vision processing. This research provided a standardized and expedited way to compute the bondline and characterize the bondline percentage, allowing for a robust examination of many different materials and process conditions. Simulations were performed to model the core sheet formation process and to investigate the source of variability of the internal sheet geometry. This included determining why cell wandering occurs and how to mitigate it.

- Diffusion bonding models were created to expand on current methods and to investigate the deficiencies in current methods. This included updating previous models for better geometric inputs. A new surface tracking method that removes the constraints on the void geometry was developed. This showed that different combinations of voids performed differently and that there is a way to optimize the void geometry for better bonding performance.
- Advanced bonding and superplastic models were created using phase field methods and molecular dynamics. These unique models were used to show that the type of material around a void is a dominant factor in the bonding performance. The superplastic model showed the interaction between the two different sliding modes and how they contribute to the overall sliding mechanism.
- Additionally, non-destructive evaluation was performed on the bonding samples. It was determined that conventional ultrasound was ineffective at capturing the voids. Simulations were used to explain that while there is no reflection off the voids, since the wavelength is too large, there is a slight change in waveform due to the void presence.

While SPF/DB is a complex process, through the work that has been done in this PhD it is better understood. There is still a long way to go in order to understand superplasticity and diffusion bonding fully and the author has several recommendations for future work. Additional modeling work would be to expand on the phase field model to include three dimensional EBSD initial conditions for a 3D simulation. The initial EBSD conditions would ideally be from bonded samples that were only heated for a brief period of time, capturing the beginning of the bonding process after the initial plastic collapse. Future experimental work could be done on quantifying

the exact nature of the effects of superplastic strain on the free surface. This would allow one to more accurately predict the bonding after superplastic forming. Additional experiments should be done to reinforce the ultrasonic modeling and the shear test. For diffusion bonding a rigorous statistical data collection could be performed to catalog the voids and the effects of different microstructural features. All of these would add to the work presented in this dissertation and contribute to the current knowledge on superplastic forming and diffusion bonding.

References

1. Filice L, Gagliardi F, Lazzaro S, Rocco C (2010) FE simulation and experimental considerations on TI alloy superplastic forming for aerospace applications. Int J Mater Form 3:41–46

2. Langdon TG (2009) Seventy-five years of superplasticity: historic developments and new opportunities. J Mater Sci 44:5998–6010. https://doi.org/10.1007/s10853-009-3780-5

3. Sanders DG, Ramulu M (2004) Examination of superplastic forming combined with diffusion bonding for titanium: Perspective from experience. J Mater Eng Perform 13:744–752. https://doi.org/10.1361/10599490421574

4. Chandra N (2002) Constitutive behavior of superplastic materials. Int J Non-Linear Mech 37:461–484

5. Froes FH (2015) Titanium: physical metallurgy, processing, and applications. ASM International

6. Barnes AJ (2007) Superplastic Forming 40 Years and Still Growing. J Mater Eng Perform 16:440–454. https://doi.org/10.1007/s11665-007-9076-5

7. Alabort E, Kontis P, Barba D, et al (2016) On the mechanisms of superplasticity in Ti–6Al–4V. Acta Mater 105:449–463. https://doi.org/10.1016/j.actamat.2015.12.003

8. Alabort E, Putman D, Reed RC (2015) Superplasticity in Ti–6Al–4V: Characterisation, modelling and applications. Acta Mater 95:428–442. https://doi.org/10.1016/j.actamat.2015.04.056

9. Hefti LD (2010) Fine-grain titanium 6AI-4V for superplastic forming and diffusion bonding of aerospace products. JOM 62:42–45. https://doi.org/10.1007/s11837-010-0076-9

10. Wulff J, Cline JE, Shaler AJ, et al (1952) Titanium Carbide-Nickel Cermets: Processing and Joing. MASSACHUSETTS INST OF TECH CAMBRIDGE

11. Orhan N, Aksoy M, Eroglu M (1999) A new model for diffusion bonding and its application to duplex alloys. Mater Sci Eng A 271:458–468

12. Hamilton CH (1973) Pressure requirements for diffusion bonding titanium. Titan Sci Technol 625–648

13. Garmong G, Paton NE, Argon AS (1975) Attainment of full interfacial contact during diffusion bonding. Metall Trans A 6:1269–1279

14. Derby B, Wallach ER (1982) Theoretical model for diffusion bonding. Met Sci 16:49–56

15. Hill A, Wallach E (1989) Modelling solid-state diffusion bonding. Acta Metall 37:2425–2437

16. Pilling J, Livesey DW, Hawkyard JB, Ridley N (1984) Solid state bonding in superplastic Ti-6Al-4V. Met Sci 18:117–122

17. Pilling J (1988) The kinetics of isostatic diffusion bonding in superplastic materials. Mater Sci Eng 100:137–144

18. Salehi MT, Pilling J, Ridley N, Hamilton DL (1992) Isostatic diffusion bonding of superplastic Ti-6Al-4V. Mater Sci Eng A 150:1–6

19. Islam MF, Pilling J, Ridley N (1997) Effect of surface finish and sheet thickness on isostatic diffusion bonding of superplastic Ti-6Al-4V. Mater Sci Technol 13:1045–1050. https://doi.org/10.1179/mst.1997.13.12.1045

20. Pilling J (1989) Superplasticity in crystalline solids. Inst Met 175–178

21. Chen I-W, Argon AS (1981) Diffusive growth of grain-boundary cavities. Acta Metall 29:1759–1768. https://doi.org/10.1016/0001-6160(81)90009-2

22. Hancock JW (1976) Creep cavitation without a vacancy flux. Met Sci 10:319–325

23. Ma R, Li M, Li H, Yu W (2012) Modeling of void closure in diffusion bonding process based on dynamic conditions. Sci China Technol Sci 55:2420–2431. https://doi.org/10.1007/s11431-012-4927-1

24. Li S-X, Tu S-T, Xuan F-Z (2005) A probabilistic model for prediction of bonding time in diffusion bonding. Mater Sci Eng A 407:250–255. https://doi.org/10.1016/j.msea.2005.07.003

25. Kulkarni N, Ramulu M, Sanders DG (2016) Modeling of Diffusion Bonding Time in Dissimilar Titanium Alloys: Preliminary Results. J Manuf Sci Eng 138:121010

26. Çm G, Clemens H, Gerling R, Kocak M (1999) Diffusion bonding of γ-TiAl sheets. Intermetallics 7:1025–1031

27. Derby B, Briggs GAD, Wallach ER (1983) Non-destructive testing and acoustic microscopy of diffusion bonds. J Mater Sci 18:2345–2353

28. Xuan FZ, Zhang B, Tu ST (2007) Interfacial Resistance Method for Quality Evaluation of Diffusion Bonded Joints. Key Eng Mater 353–358:1944–1947. https://doi.org/10.4028/www.scientific.net/KEM.353-358.1944

29. Li Y, Xuan F, Li S, Tu S (2011) Quality Evaluation of Diffusion Bonded Joints by Electrical Resistance Measuring and Microscopic Fatigue Testing. Chin J Mech Eng 24:1

30. Windels F, Leroy O (2002) Air-coupled ultrasonic testing of diffusion bonds. Ultrasonics 40:171–176. https://doi.org/10.1016/S0041-624X(02)00132-4

31. Rose JL, Zhu W, Zaidi M (1998) Ultrasonic NDT of titanium diffusion bonding with guided waves. Mater Eval 56(4):535–539

32. Lavrentyev AI, Beals JT (2000) Ultrasonic measurement of the diffusion bond strength. Ultrasonics 38:513–516

33. Milne K, Cawley P, Nagy PB, et al (2011) Ultrasonic Non-destructive Evaluation of Titanium Diffusion Bonds. J Nondestruct Eval 30:225–236. https://doi.org/10.1007/s10921-011-0111-y

34. Chen S, Ke F, Zhou M, Bai Y (2007) Atomistic investigation of the effects of temperature and surface roughness on diffusion bonding between Cu and Al. Acta Mater 55:3169–3175. https://doi.org/10.1016/j.actamat.2006.12.040

35. Jiao Z, Song C, Lin T, He P (2011) Molecular dynamics simulation of the effect of surface roughness and pore on linear friction welding between Ni and Al. Comput Mater Sci 50:3385–3389

36. Chen SY, Wu ZW, Liu KX, et al (2013) Atomic diffusion behavior in Cu-Al explosive welding process. J Appl Phys 113:044901. https://doi.org/10.1063/1.4775788

37. Li X, Chu W, Ma T, Wang Q (2016) Molecular Dynamics Simulation on Diffusion Welding Between Cu and Al Under Different Pressures and Roughnesses. In: ASME 2016 Heat Transfer Summer Conference collocated with the ASME 2016 Fluids Engineering Division Summer Meeting and the ASME 2016 14th International Conference on Nanochannels, Microchannels, and Minichannels. American Society of Mechanical Engineers, p V002T13A004–V002T13A004

38. Berry J, Rottler J, Sinclair CW, Provatas N (2015) Atomistic study of diffusion-mediated plasticity and creep using phase field crystal methods. Phys Rev B 92:. https://doi.org/10.1103/PhysRevB.92.134103

39. Keralavarma SM, Benzerga AA (2015) High-temperature discrete dislocation plasticity. J Mech Phys Solids 82:1–22. https://doi.org/10.1016/j.jmps.2015.05.003

40. Kazaryan A, Wang Y, Patton BR (1999) Generalized phase field approach for computer simulation of sintering: incorporation of rigid-body motion. Scr Mater 41:487–492. https://doi.org/10.1016/S1359-6462(99)00179-7

41. Wang YU (2006) Computer modeling and simulation of solid-state sintering: A phase field approach. Acta Mater 54:953–961. https://doi.org/10.1016/j.actamat.2005.10.032

42. Jing XN, Zhao JH, Subhash G, Gao X-L (2005) Anisotropic grain growth with pore drag under applied loads. Mater Sci Eng A 412:271–278. https://doi.org/10.1016/j.msea.2005.08.220

43. Moelans N, Blanpain B, Wollants P (2008) An introduction to phase-field modeling of microstructure evolution. Calphad 32:268–294. https://doi.org/10.1016/j.calphad.2007.11.003

44. Cahn JW, Hilliard JE (1958) Free Energy of a Nonuniform System. I. Interfacial Free Energy. J Chem Phys 28:258–267. https://doi.org/10.1063/1.1744102

45. Larché F, Cahn JW (1973) A linear theory of thermochemical equilibrium of solids under stress. Acta Metall 21:1051–1063. https://doi.org/10.1016/0001-6160(73)90021-7

46. Shi R, Dixit V, Fraser HL, Wang Y (2014) Variant selection of grain boundary α by special prior β grain boundaries in titanium alloys. Acta Mater 75:156–166. https://doi.org/10.1016/j.actamat.2014.05.003

47. Kim SG, Kim WT (2005) Phase field modeling of dendritic growth with high anisotropy. J Cryst Growth 275:e355–e360. https://doi.org/10.1016/j.jcrysgro.2004.11.057

48. Aagesen LK, Gao Y, Schwen D, Ahmed K (2018) Grand-potential-based phase-field model for multiple phases, grains, and chemical components. Phys Rev E 98:. https://doi.org/10.1103/PhysRevE.98.023309

49. Moelans N (2011) A quantitative and thermodynamically consistent phase-field interpolation function for multi-phase systems. Acta Mater 59:1077–1086. https://doi.org/10.1016/j.actamat.2010.10.038

50. Aagesen LK, Schwen D, Ahmed K, Tonks MR (2017) Quantifying elastic energy effects on interfacial energy in the Kim-Kim-Suzuki phase-field model with different interpolation schemes. Comput Mater Sci 140:10–21. https://doi.org/10.1016/j.commatsci.2017.08.005

51. Shi R, Wang Y (2013) Variant selection during α precipitation in Ti–6Al–4V under the influence of local stress – A simulation study. Acta Mater 61:6006–6024. https://doi.org/10.1016/j.actamat.2013.06.042

52. Qiu D, Zhao P, Shi R, et al (2016) Effect of autocatalysis on variant selection of α precipitates during phase transformation in Ti-6Al-4V alloy. Comput Mater Sci 124:282–289. https://doi.org/10.1016/j.commatsci.2016.07.032

53. Gey N, Humbert M, Philippe MJ, Combres Y (1996) Investigation of the α- and β- texture evolution of hot rolled Ti-64 products. Mater Sci Eng A 219:80–88. https://doi.org/10.1016/S0921-5093(96)10388-9

54. Hötzer J, Seiz M, Kellner M, et al (2019) Phase-field simulation of solid state sintering. Acta Mater 164:184–195. https://doi.org/10.1016/j.actamat.2018.10.021

55. Chockalingam K, Kouznetsova VG, van der Sluis O, Geers MGD (2016) 2D Phase field modeling of sintering of silver nanoparticles. Comput Methods Appl Mech Eng 312:492–508. https://doi.org/10.1016/j.cma.2016.07.002

56. Yamakov V, Wolf D, Phillpot SR, et al (2004) Deformation-mechanism map for nanocrystalline metals by molecular-dynamics simulation. Nat Mater 3:43–47. https://doi.org/10.1038/nmat1035

57. Yamakov V, Wolf D, Phillpot SR, Gleiter H (2002) Grain-boundary diffusion creep in nanocrystalline palladium by molecular-dynamics simulation. Acta Mater 50:61–73. https://doi.org/10.1016/S1359-6454(01)00329-9

58. Zhang L, Shibuta Y, Huang X, et al (2019) Grain boundary induced deformation mechanisms in nanocrystalline Al by molecular dynamics simulation: From interatomic potential perspective. Comput Mater Sci 156:421–433. https://doi.org/10.1016/j.commatsci.2018.10.021

59. Cheng K, Tieu K, Lu C, et al (2014) Molecular dynamics simulation of the grain boundary sliding behaviour for Al Σ5 (210). Comput Mater Sci 81:52–57. https://doi.org/10.1016/j.commatsci.2013.05.005

60. Petisme MVG, Gren MA, Wahnström G (2015) Molecular dynamics simulation of WC/WC grain boundary sliding resistance in WC–Co cemented carbides at high temperature. Int J Refract Met Hard Mater 49:75–80. https://doi.org/10.1016/j.ijrmhm.2014.07.037

61. Pérez-Prado MT, González-Doncel G, Ruano OA, McNelley TR (2001) Texture analysis of the transition from slip to grain boundary sliding in a discontinuously recrystallized superplastic aluminum alloy. Acta Mater 49:2259–2268. https://doi.org/10.1016/S1359-6454(01)00128-8

62. Du N, Qi Y, Krajewski PE, Bower AF (2010) Aluminum Σ3 grain boundary sliding enhanced by vacancy diffusion. Acta Mater 58:4245–4252. https://doi.org/10.1016/j.actamat.2010.04.016

63. Qi Y, Krajewski PE (2007) Molecular dynamics simulations of grain boundary sliding: The effect of stress and boundary misorientation. Acta Mater 55:1555–1563. https://doi.org/10.1016/j.actamat.2006.10.016

64. Somekawa H, Hosokawa H, Watanabe H, Higashi K (2003) Diffusion bonding in superplastic magnesium alloys. Mater Sci Eng A 339:328–333

65. Somekawa H, Watanabe H, Mukai T, Higashi K (2003) Low temperature diffusion bonding in a superplastic AZ31 magnesium alloy. Scr Mater 48:1249–1254. https://doi.org/10.1016/S1359-6462(03)00054-X

66. Huang Y, Ridley N, Humphreys FJ, Cui J-Z (1999) Diffusion bonding of superplastic 7075 aluminium alloy. Mater Sci Eng A 266:295–302

67. Britto ASF, Raj RE, Mabel MC (2017) Prediction of shear and tensile strength of the diffusion bonded AA5083 and AA7075 aluminium alloy using ANN. Mater Sci Eng A 692:1–8

68. Mahendran G, Balasubramanian V, Senthilvelan T (2009) Developing diffusion bonding windows for joining AZ31B magnesium–AA2024 aluminium alloys. Mater Des 30:1240–1244. https://doi.org/10.1016/j.matdes.2008.06.015

69. Elrefaey A, Tillmann W (2009) Solid state diffusion bonding of titanium to steel using a copper base alloy as interlayer. J Mater Process Technol 209:2746–2752. https://doi.org/10.1016/j.jmatprotec.2008.06.014

70. Kurt B, Orhan N, Evin E, Çalik A (2007) Diffusion bonding between Ti–6Al–4V alloy and ferritic stainless steel. Mater Lett 61:1747–1750. https://doi.org/10.1016/j.matlet.2006.07.123

71. Rajakumar S, Balasubramanian V (2016) Diffusion bonding of titanium and AA 7075 aluminum alloy dissimilar joints—process modeling and optimization using desirability approach. Int J Adv Manuf Technol 86:1095–1112

72. Zhang X-P, Ye L, Mai Y-W, et al (1999) Investigation on diffusion bonding characteristics of SiC particulate reinforced aluminium metal matrix composites (Al/SiC p-MMC). Compos Part Appl Sci Manuf 30:1415–1421

73. Calvo FA, De Salazar JG, Urena A, et al (1992) Diffusion bonding of Ti-6Al-4V alloy at low temperature: metallurgical aspects. J Mater Sci 27:391–398

74. Glatz W, Clemens H (1997) Diffusion bonding of intermetallic Ti-47Al-2Cr-0.2 Si sheet material and mechanical properties of joints at room temperature and elevated temperatures. Intermetallics 5:415–423

75. Nakao Y, Shinozaki K, Hamada M (1991) Diffusion bonding of intermetallic compound TiAl. ISIJ Int 31:1260–1266

76. Lesuer D (2000) Experimental investigation of material models for Ti-6Al-4V and 2024-T3. Avaliable Online Httpse-Rep-Ext Llnl Govpdf236167 Pdf Accessed 29 Dec 2015

77. Graff KF (2012) Wave motion in elastic solids. Courier Corporation

78. Gaston DR, Permann CJ, Peterson JW, et al (2015) Physics-based multiscale coupling for full core nuclear reactor simulation. Ann Nucl Energy 84:45–54. https://doi.org/10.1016/j.anucene.2014.09.060

79. Hilber HM, Hughes TJ, Taylor RL (1977) Improved numerical dissipation for time integration algorithms in structural dynamics. Earthq Eng Struct Dyn 5:283–292

80. Hales SJ, Wagner JA (1991) Superplastic forming of Al-Li alloys for lightweight, low-cost structures

81. Giuliano G (2008) Constitutive equation for superplastic Ti–6Al–4V alloy. Mater Des 29:1330–1333. https://doi.org/10.1016/j.matdes.2007.07.001

82. Luo J, Li M, Li X, Shi Y (2010) Constitutive model for high temperature deformation of titanium alloys using internal state variables. Mech Mater 42:157–165. https://doi.org/10.1016/j.mechmat.2009.10.004

83. Khaleel MA, Zbib HM, Nyberg EA (2001) Constitutive modeling of deformation and damage in superplastic materials. Int J Plast 17:277–296. https://doi.org/10.1016/S0749-6419(00)00036-X

84. Carrino L, Giuliano G, Palmieri C (2003) On the optimisation of superplastic forming processes by the finite-element method. J Mater Process Technol 143–144:373–377. https://doi.org/10.1016/S0924-0136(03)00423-0

85. Gudipati PP, Kosaka Y (2016) Diffusion Bonding of Similar and Dissimilar Titanium Alloys. Wiley Online Library, pp 1631–1636

86. Jadhav S, Powar A, Patil S, et al (2017) Effect of volume fraction of alpha and transformed beta on the high cycle fatigue properties of bimodal Ti6Al4V alloy. IOP Conf Ser Mater Sci Eng 201:012035. https://doi.org/10.1088/1757-899X/201/1/012035

87. Kosaka Y, GUDIPATI P (2014) Method for the manufacture of alpha-beta ti-al-v-mo-fe alloy sheets

88. Kassner ME, Kosaka Y, Hall JS (1999) Low-cycle dwell-time fatigue in Ti-6242. Metall Mater Trans A 30:2383–2389

89. Sargent PM, Ashby MF (1982) Deformation maps for titanium and zirconium. Scr Metall 16:1415–1422

90. Huang L, Cui Y, Chang H, et al (2010) Assessment of Atomic Mobilities for bcc Phase of Ti-Al-V System. J Phase Equilibria Diffus 31:135–143. https://doi.org/10.1007/s11669-009-9641-8

91. Liu Y, Ge Y, Yu D, et al (2009) Assessment of the diffusional mobilities in bcc Ti–V alloys. J Alloys Compd 470:176–182. https://doi.org/10.1016/j.jallcom.2008.02.111

92. Guo ZX, Ridley N (1987) Modelling of diffusion bonding of metals. Mater Sci Technol 3:945–953

93. Derby B, Wallach ER (1984) Diffusion bonding: development of theoretical model. Met Sci 18:427–431

94. Derby B, Wallach ER (1984) Diffusion bonds in copper. J Mater Sci 19:3140–3148

95. Li H, Li MQ, Kang PJ (2016) Void shrinking process and mechanisms of the diffusion bonded Ti–6Al–4V alloy with different surface roughness. Appl Phys A 122:. https://doi.org/10.1007/s00339-015-9546-9

96. Takahashi Y, Inoue K (1992) Recent void shrinkage models and their applicability to diffusion bonding. Mater Sci Technol 8:953–964

97. Wang A, Ohashi O, Ueno K (2006) Effect of Surface Asperity on Diffusion Bonding. Mater Trans 47:179–184. https://doi.org/10.2320/matertrans.47.179

98. Shao X, Guo X, Han Y, et al (2015) Characterization of the diffusion bonding behavior of pure Ti and Ni with different surface roughness during hot pressing. Mater Des 1980-2015 65:1001–1010. https://doi.org/10.1016/j.matdes.2014.09.071

99. Guo Y, Wang Y, Gao B, et al (2016) Rapid diffusion bonding of WC-Co cemented carbide to 40Cr steel with Ni interlayer: Effect of surface roughness and interlayer thickness. Ceram Int 42:16729–16737. https://doi.org/10.1016/j.ceramint.2016.07.145

100. Zhang C, Li H, Li M (2017) Role of surface finish on interface grain boundary migration in vacuum diffusion bonding. Vacuum 137:49–55. https://doi.org/10.1016/j.vacuum.2016.12.021

101. Zuruzi AS, Li H, Dong G (1999) Effects of surface roughness on the diffusion bonding of Al alloy 6061 in air. Mater Sci Eng A 270:244–248. https://doi.org/10.1016/S0921-5093(99)00188-4

102. Somekawa H, Higashi K (2003) The Optimal Surface Roughness Condition on Diffusion Bonding. Mater Trans 44:1640–1643. https://doi.org/10.2320/matertrans.44.1640

103. Huang P, Li Z, Sun J (2002) Shrinkage and splitting of microcracks under pressure simulated by the finite-element method. Metall Mater Trans A 33:1117–1124. https://doi.org/10.1007/s11661-002-0213-3

104. Huang PZ, Sun J, Li ZH, Gao H (2003) Morphological healing evolution of intragranular penny-shaped microcracks by surface diffusion: Part I. Simulation. Metall Mater Trans A 34:277–285. https://doi.org/10.1007/s11661-003-0329-0

105. Huang P, Sun J (2004) A numerical analysis of intergranular penny-shaped microcrack shrinkage controlled by coupled surface and interface diffusion. Metall Mater Trans A 35:1301–1309. https://doi.org/10.1007/s11661-004-0304-4

106. Pharr GM, Nix WD (1979) A numerical study of cavity growth controlled by surface diffusion. Acta Metall 27:1615–1631. https://doi.org/10.1016/0001-6160(79)90044-0

107. Chuang T-J, Kagawa KI, Rice JR, Sills LB (1979) Overview no. 2: Non-equilibrium models for diffusive cavitation of grain interfaces. Acta Metall 27:265–284. https://doi.org/10.1016/0001-6160(79)90021-X

108. Needleman A, Rice JR (1980) Plastic creep flow effects in the diffusive cavitation of grain boundaries. Acta Metall 28:1315–1332. https://doi.org/10.1016/0001-6160(80)90001-2

109. Martinez L, Nix WD (1982) A numerical study of cavity growth controlled by coupled surface and grain boundary diffusion. Metall Trans A 13:427–437

110. Wang H, Li Z (2004) The three-dimensional analysis for diffusive shrinkage of a grain-boundary void in stressed solid. J Mater Sci 39:3425–3432. https://doi.org/10.1023/B:JMSC.0000026945.89767.25

111. Takahashi Y, Takahashi K, Nishiguchi K (1991) A numerical analysis of void shrinkage processes controlled by coupled surface and interface diffusion. Acta Metall Mater 39:3199–3216

112. Takahashi Y, Ueno F, Nishiguchi K (1988) A numerical analysis of the void-shrinkage process controlled by surface-diffusion. Acta Metall 36:3007–3018

113. Pilling J (1989) Superplasticity in crystalline solids. Inst Met 175–178

114. Biswas S, Schwen D, Singh J, Tomar V (2016) A study of the evolution of microstructure and consolidation kinetics during sintering using a phase field modeling based approach. Extreme Mech Lett 7:78–89. https://doi.org/10.1016/j.eml.2016.02.017

115. Campbell CE, Boettinger WJ (2000) Transient liquid-phase bonding in the Ni-Al-B system. Metall Mater Trans A 31:2835–2847. https://doi.org/10.1007/BF02830355

116. Jabbareh MA, Assadi H (2009) Modelling of microstructure evolution in transient-liquid-phase diffusion bonding under temperature gradient. Scr Mater 60:780–782. https://doi.org/10.1016/j.scriptamat.2009.01.013

117. Natsume Y, Ohsasa K, Narita T (2003) Phase-field Simulation of Transient Liquid Phase Bonding Process of Ni Using Ni-P Binary Filler Metal. Mater Trans 44:819–823. https://doi.org/10.2320/matertrans.44.819

118. Park MS, Gibbons SL, Arróyave R (2012) Phase-field simulations of intermetallic compound growth in Cu/Sn/Cu sandwich structure under transient liquid phase bonding conditions. Acta Mater 60:6278–6287. https://doi.org/10.1016/j.actamat.2012.07.063

119. Chen Q, Ma N, Wu K, Wang Y (2004) Quantitative phase field modeling of diffusion-controlled precipitate growth and dissolution in Ti–Al–V. Scr Mater 50:471–476. https://doi.org/10.1016/j.scriptamat.2003.10.032

120. Wang Y-Z, Ma N, Chen Q, et al (2005) Predicting phase equilibrium, phase transformation, and microstructure evolution in titanium alloys. JOM 57:32–39. https://doi.org/10.1007/s11837-005-0112-3

121. Shi R, Ma N, Wang Y (2012) Predicting equilibrium shape of precipitates as function of coherency state. Acta Mater 60:4172–4184. https://doi.org/10.1016/j.actamat.2012.04.019

122. Qiu D, Shi R, Zhang D, et al (2015) Variant selection by dislocations during α precipitation in α/β titanium alloys. Acta Mater 88:218–231. https://doi.org/10.1016/j.actamat.2014.12.044

123. Hu SY, Henager Jr. CH (2010) Phase-field simulation of void migration in a temperature gradient. Acta Mater 58:3230–3237. https://doi.org/10.1016/j.actamat.2010.01.043

124. Rovelli I, Dudarev SL, Sutton AP (2017) Non-local model for diffusion-mediated dislocation climb and cavity growth. J Mech Phys Solids 103:121–141. https://doi.org/10.1016/j.jmps.2017.03.008

125. Burgers WG (1934) On the process of transition of the cubic-body-centered modification into the hexagonal-close-packed modification of zirconium. Physica 1:561–586. https://doi.org/10.1016/S0031-8914(34)80244-3

126. Furuhara T, Maki T (2001) Variant selection in heterogeneous nucleation on defects in diffusional phase transformation and precipitation. Mater Sci Eng A 312:145–154. https://doi.org/10.1016/S0921-5093(00)01904-3

127. Gey N, Humbert M, Philippe MJ, Combres Y (1997) Modeling the transformation texture of Ti-64 sheets after rolling in the β-field. Mater Sci Eng A 230:68–74. https://doi.org/10.1016/S0921-5093(97)80111-6

128. Semiatin SL, Kinsel KT, Pilchak AL, Sargent GA (2013) Effect of Process Variables on Transformation-Texture Development in Ti-6Al-4V Sheet Following Beta Heat Treatment. Metall Mater Trans A 44:3852–3865. https://doi.org/10.1007/s11661-013-1735-6

129. Frost HJ, Ashby MF (1982) Deformation mechanism maps: the plasticity and creep of metals and ceramics. Pergamon press

130. Hill R (1963) Elastic properties of reinforced solids: Some theoretical principles. J Mech Phys Solids 11:357–372. https://doi.org/10.1016/0022-5096(63)90036-X

131. Steinbach I, Apel M (2006) Multi phase field model for solid state transformation with elastic strain. Phys Nonlinear Phenom 217:153–160. https://doi.org/10.1016/j.physd.2006.04.001

132. Khachaturyan AG (2013) Theory of structural transformations in solids. Courier Corporation

133. Durga A, Wollants P, Moelans N (2015) A quantitative phase-field model for two-phase elastically inhomogeneous systems. Comput Mater Sci 99:81–95. https://doi.org/10.1016/j.commatsci.2014.11.057

134. Cahn JW (1961) On spinodal decomposition. Acta Metall 9:795–801. https://doi.org/10.1016/0001-6160(61)90182-1

135. Kim SG, Kim WT, Suzuki T (1999) Phase-field model for binary alloys. Phys Rev E 60:7186–7197. https://doi.org/10.1103/PhysRevE.60.7186

136. Redlich O, Kister AT (1948) Algebraic representation of thermodynamic properties and the classification of solutions. Ind Eng Chem 40:345–348

137. Lindwall G, Moon K-W, Chen Z, et al (2018) Diffusion in the Ti-Al-V System. J Phase Equilibria Diffus 39:731–746. https://doi.org/10.1007/s11669-018-0673-9

138. Lu X, Gui N, Qiu A, et al (2014) Thermodynamic Modeling of the Al-Ti-V Ternary System. Metall Mater Trans A 45:4155–4164. https://doi.org/10.1007/s11661-014-2317-y

139. Zhou M, Dunne FPE (1996) Mechanisms-based constitutive equations for the superplastic behaviour of a titanium alloy. J Strain Anal Eng Des 31:187–196

140. Balay S, Buschelman K, Eijkhout V, et al (2004) PETSc users manual. Technical Report ANL-95/11-Revision 2.1. 5, Argonne National Laboratory

141. Langdon TG (2006) Grain boundary sliding revisited: Developments in sliding over four decades. J Mater Sci 41:597–609. https://doi.org/10.1007/s10853-006-6476-0

142. Langdon TG (1994) A unified approach to grain boundary sliding in creep and superplasticity. Acta Metall Mater 42:2437–2443. https://doi.org/10.1016/0956-7151(94)90322-0

Appendix

A Raw Material Analysis

Surface Roughness

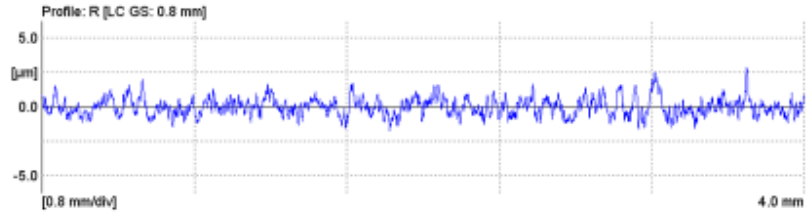
Ti 6242 0.025”

Longitudinal

Ra (um)	Rq (um)	Rz (um)
0.544	0.680	3.866
0.593	0.743	4.029
0.511	0.653	3.848

Transverse

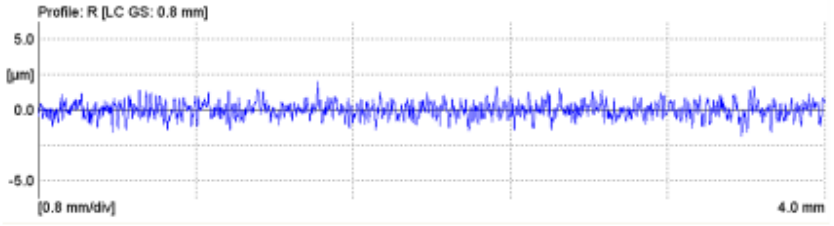
Ra (um)	Rq (um)	Rz (um)
0.512	0.646	3.589
0.543	0.686	4.060
0.534	0.672	3.867



Ti 6242 0.032”

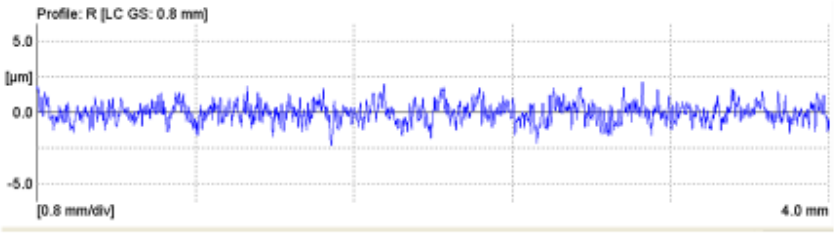
Longitudinal

Ra (um)	Rq (um)	Rz (um)
0.417	0.525	3.156
0.428	0.533	3.024
0.407	0.511	3.044



Transverse

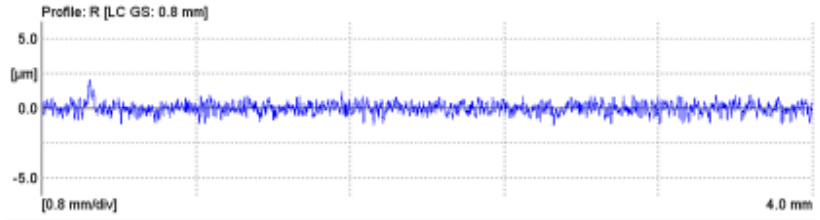
Ra (um)	Rq (um)	Rz (um)
0.517	0.656	3.709
0.514	0.647	3.771
0.495	0.616	3.463



Ti 6242 0.040”

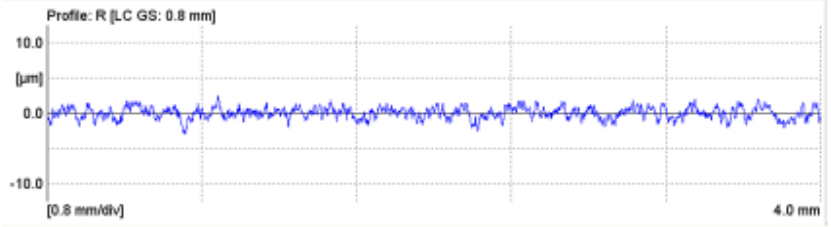
Longitudinal

Ra (um)	Rq (um)	Rz (um)
0.310	0.392	2.367
0.309	0.389	2.194
0.348	0.440	2.774



Transverse

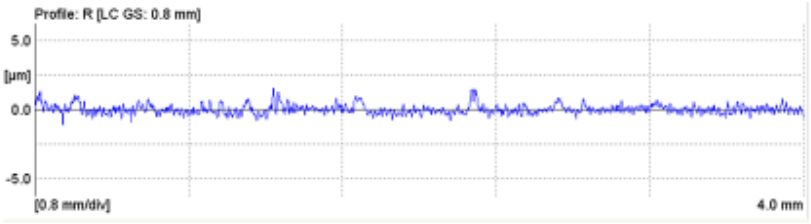
Ra (um)	Rq (um)	Rz (um)
0.672	0.826	4.138
0.749	0.927	4.736
0.650	0.814	4.410



ATI 425 0.040”

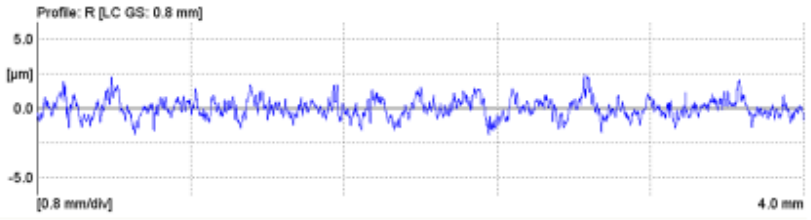
Longitudinal

Ra (um)	Rq (um)	Rz (um)
0.249	0.334	2.011
0.219	0.297	2.186
0.251	0.335	2.179



Transverse

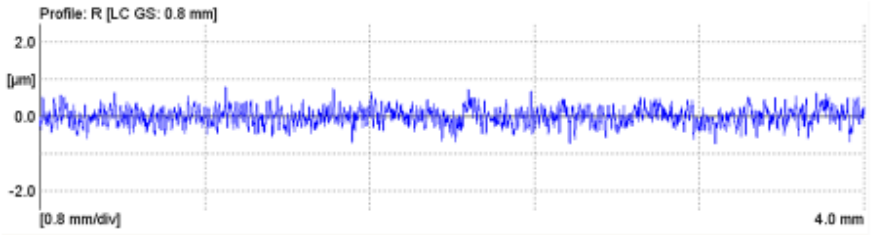
Ra (um)	Rq (um)	Rz (um)
0.533	0.675	3.676
0.489	0.661	3.592
0.499	0.639	3.331



RTI 64 FG 0.040”

Longitudinal

Ra (um)	Rq (um)	Rz (um)
0.185	0.231	1.349
0.196	0.247	1.394
0.185	0.236	1.432



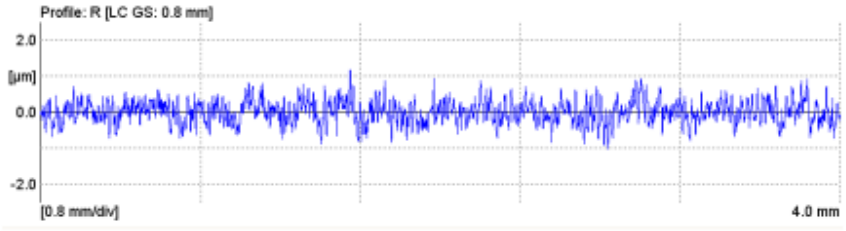
Transverse

Ra (um)	Rq (um)	Rz (um)
0.366	0.476	2.799
0.340	0.441	2.598
0.358	0.462	2.549

Ti 54M 0.025”

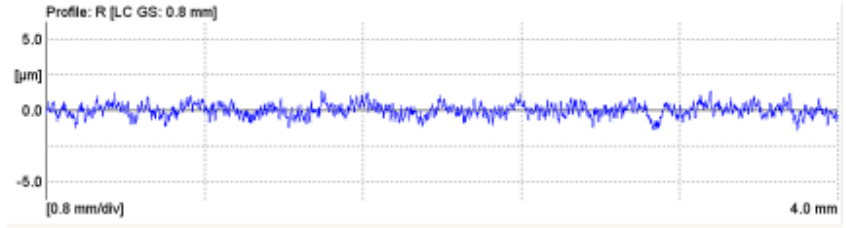
Longitudinal

Ra (um)	Rq (um)	Rz (um)
0.245	0.307	1.780
0.242	0.312	1.922
0.264	0.332	2.026



Transverse

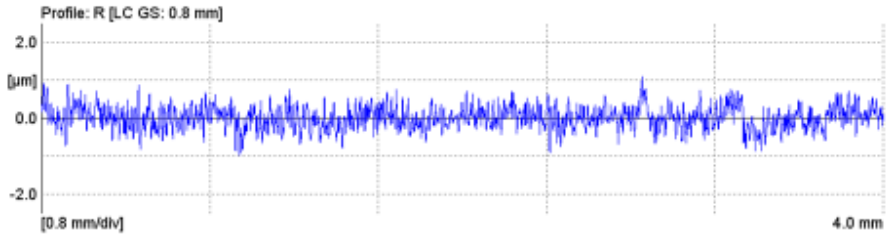
Ra (um)	Rq (um)	Rz (um)
0.344	0.432	2.514
0.395	0.500	2.816
0.362	0.451	2.582



Ti 54M 0.032”

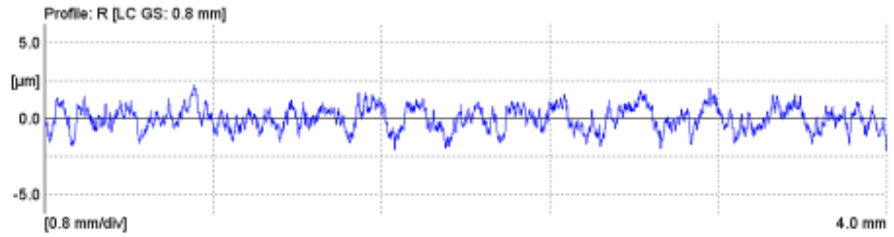
Longitudinal

Ra (um)	Rq (um)	Rz (um)
0.234	0.294	1.696
0.219	0.275	1.612
0.244	0.305	1.769



Transverse

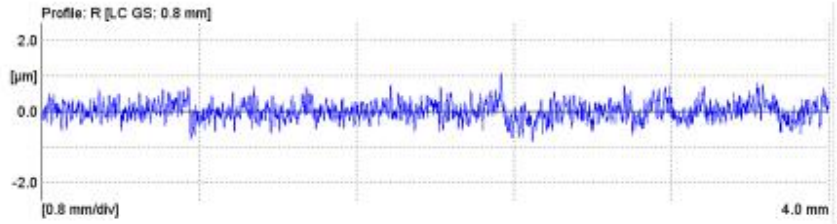
Ra (um)	Rq (um)	Rz (um)
0.625	0.757	3.682
0.567	0.707	3.717
0.508	0.629	3.431



Ti 54MX 0.040”

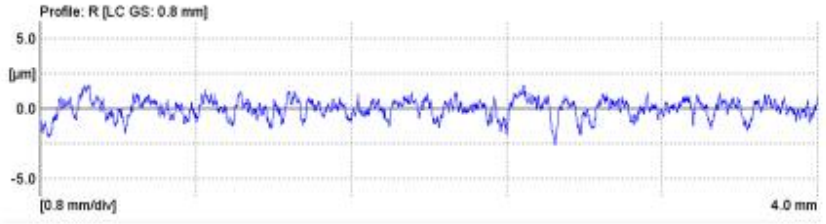
Longitudinal

Ra (um)	Rq (um)	Rz (um)
0.201	0.253	1.520
0.227	0.286	1.679
0.201	0.252	1.428



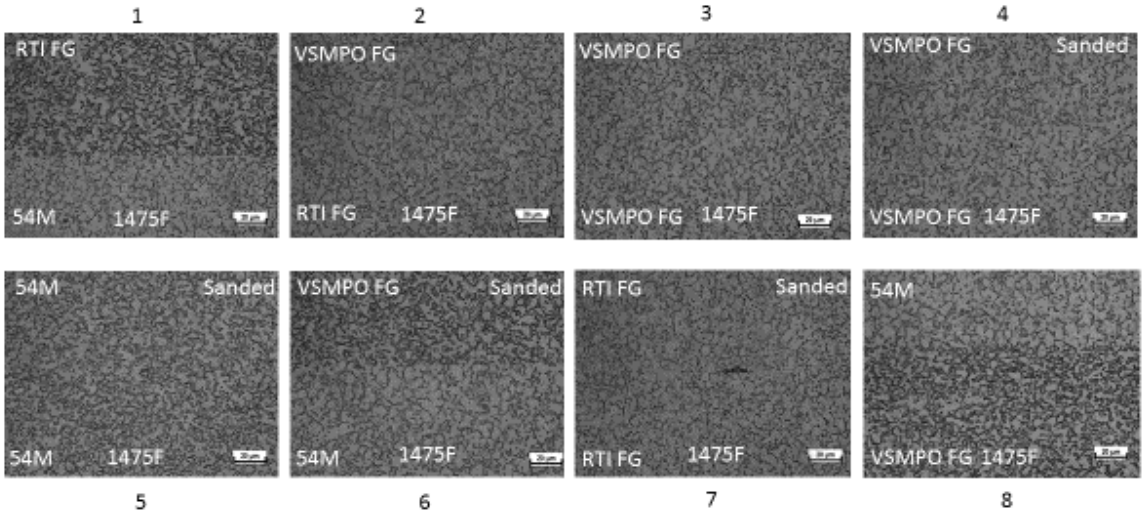
Transverse

Ra (um)	Rq (um)	Rz (um)
0.519	0.654	3.323
0.472	0.590	3.076
0.444	0.559	3.023

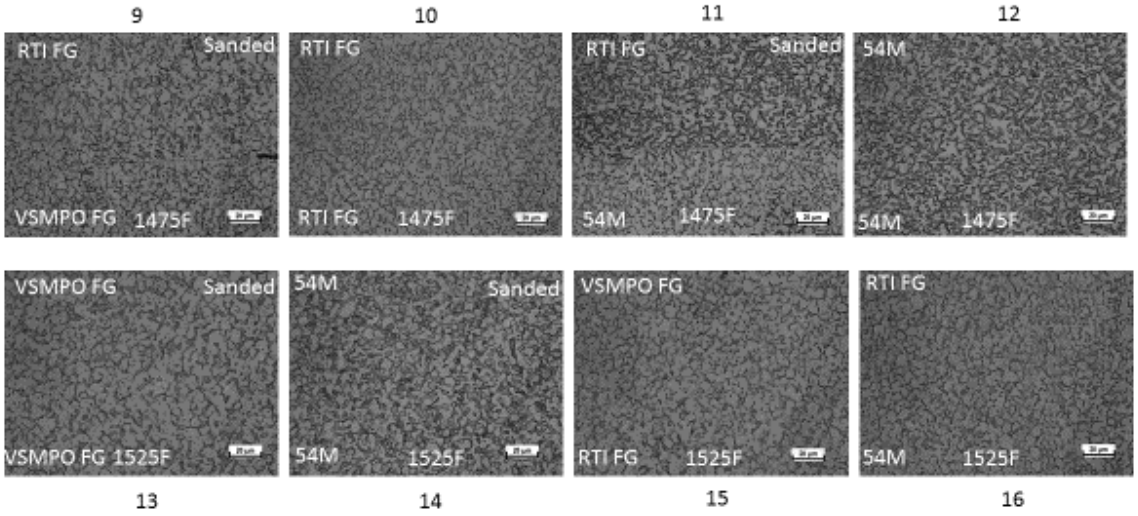


B Dissimilar Bonding TIM3 Results

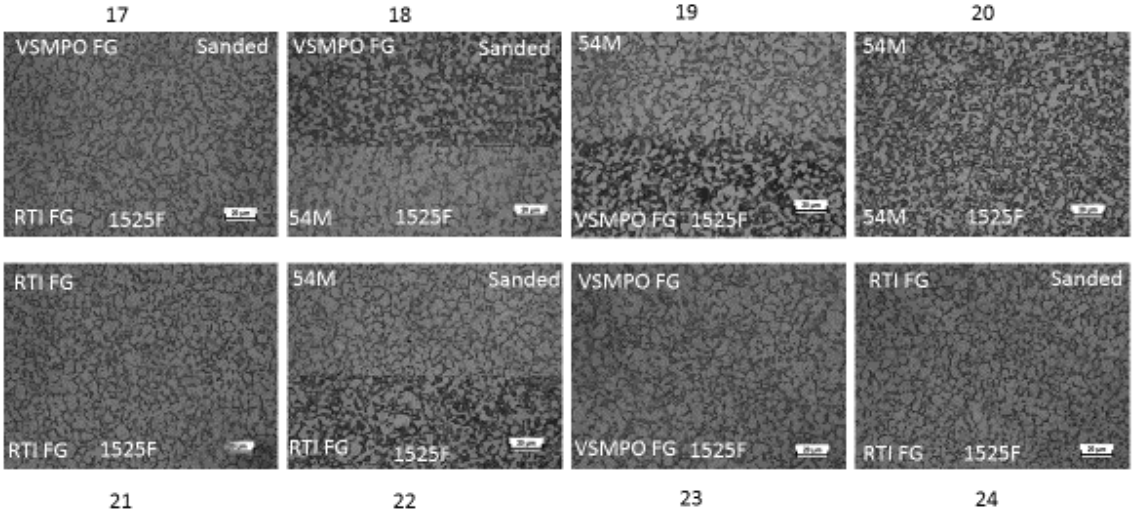
Microstructural Results 1-8



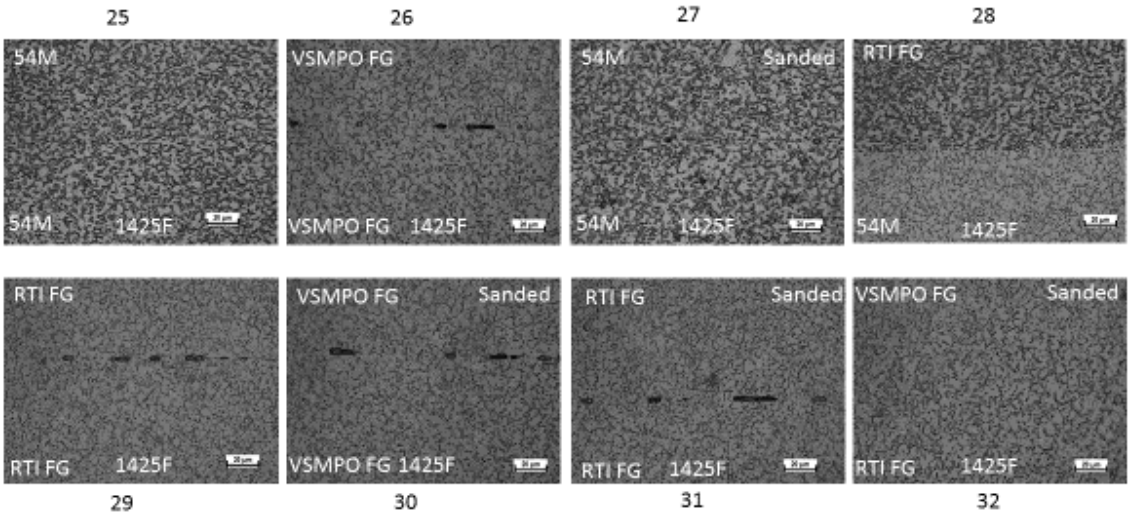
Microstructural Results 9-16



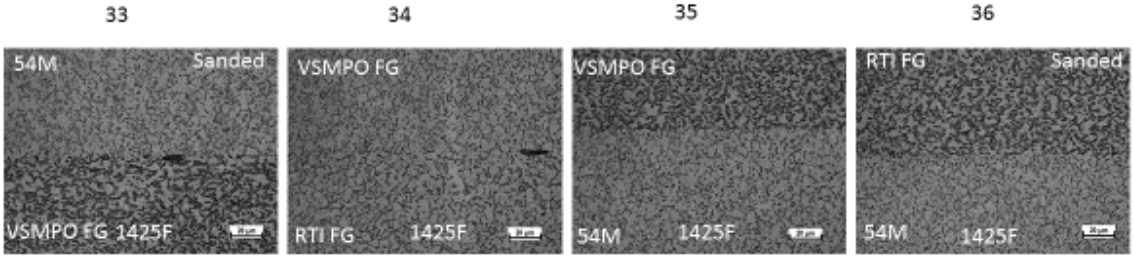
Microstructural Results 17-24



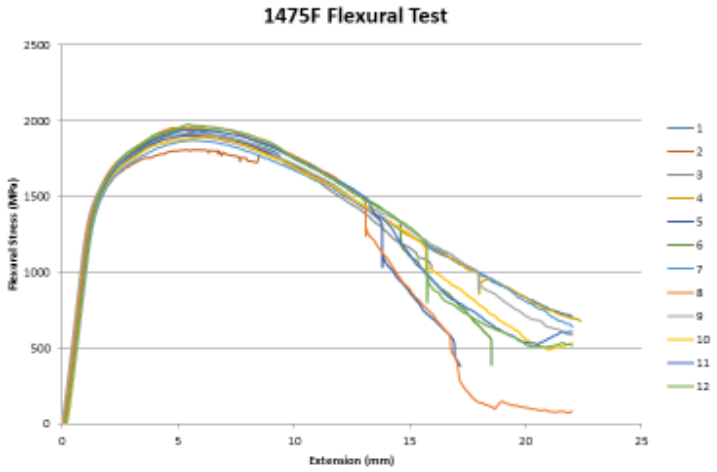
Microstructural Results 25-32



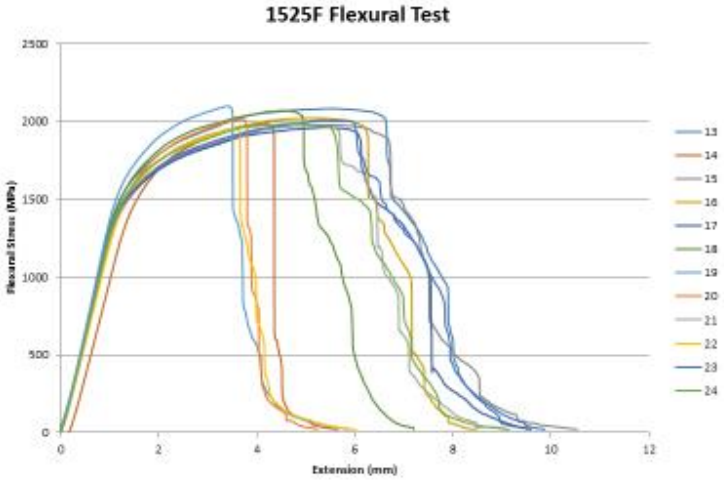
Microstructural Results 33-36



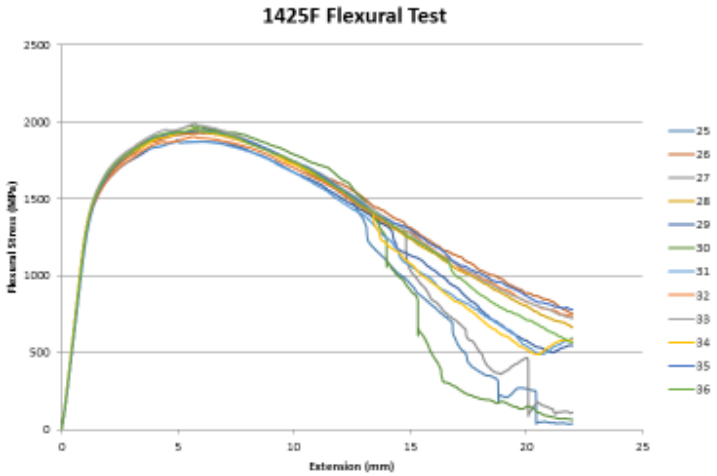
Structural Results 1475F



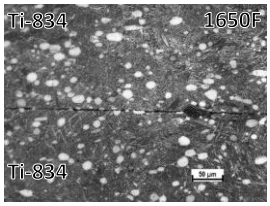
Structural Results 1525F

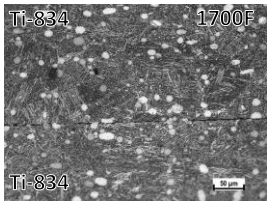


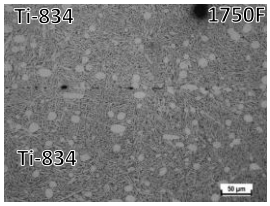
Structural Results 1425F

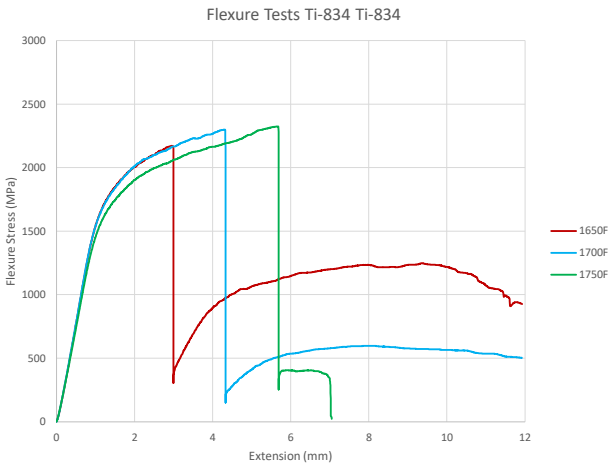


Ti-834 To Ti-834



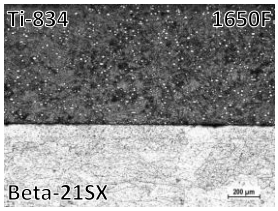


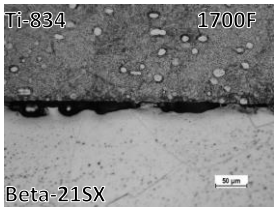


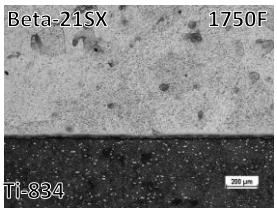


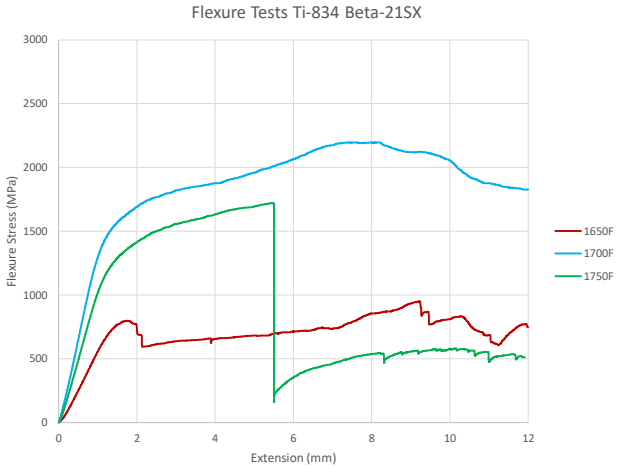
Very poor bonding and early fractures across all temperatures

Ti-834 To Beta-21SX



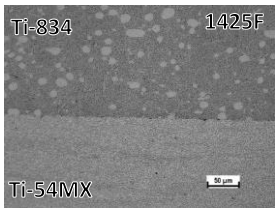


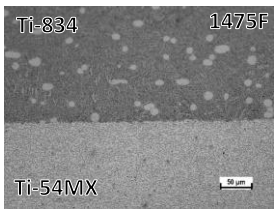


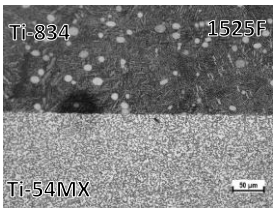


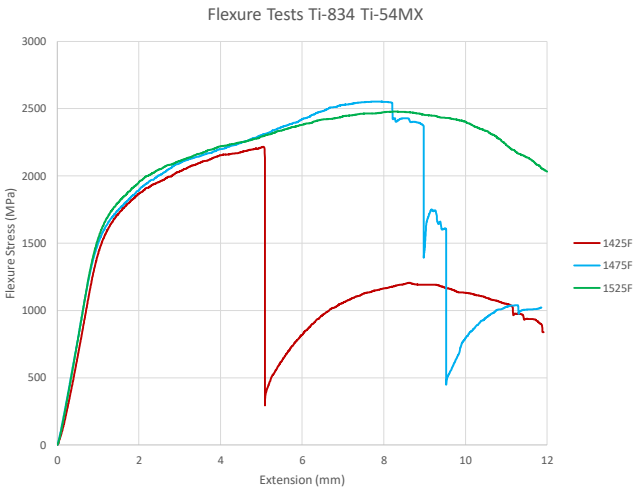
Beta 21SX didn't bond well at all. The 1650F test coupon had large sections completely delaminated.

Ti-834 To Ti-54MX



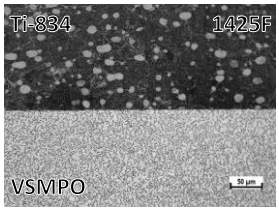


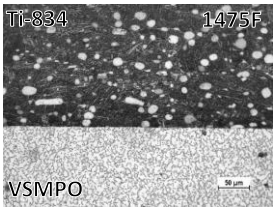


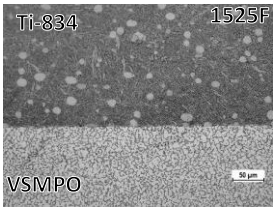


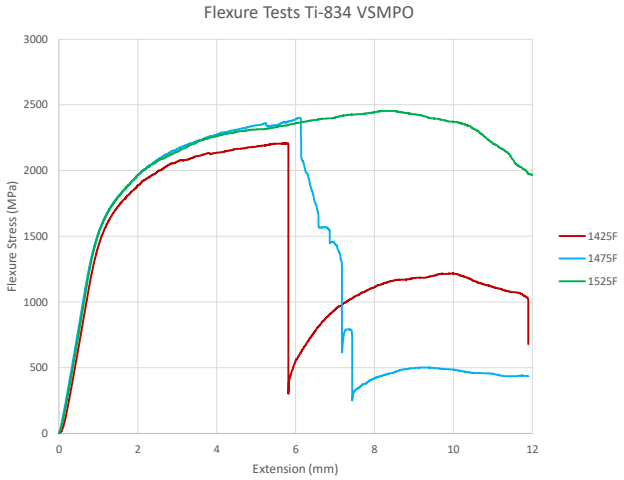
834 had many voids when combined with 54MX with some visible remaining at 1525F. Delamination and early fracture occurred at the two lowest temperatures.

Ti-834 To VSMPO



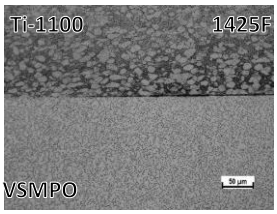


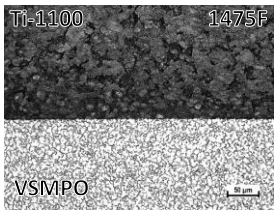


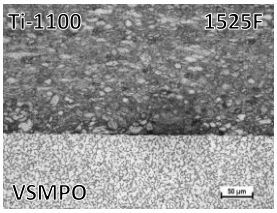


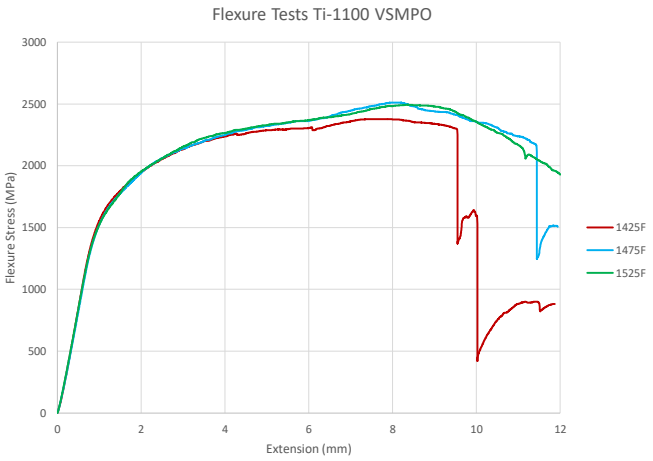
Poor bonding occurred at the two lowest temperatures but was improved for 1525F with only a few visible voids. Delamination again at the two lowest temperatures.

Ti-1100 To VSMPO



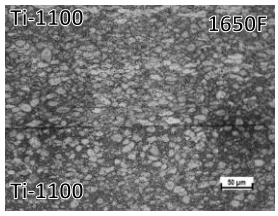


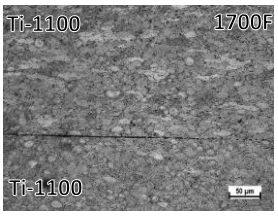


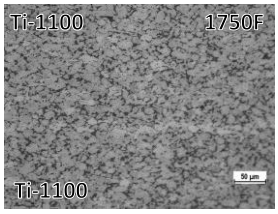


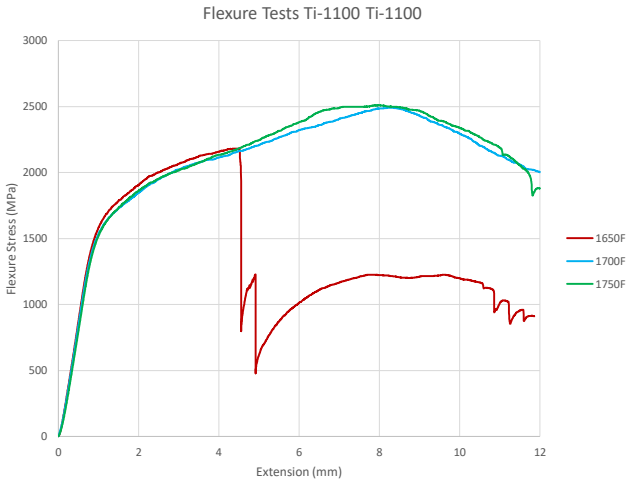
All samples had void problems with delamination occurring at the two low temperatures.

Ti-1100 To Ti-1100

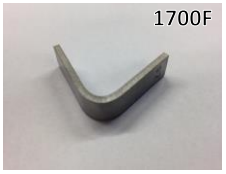




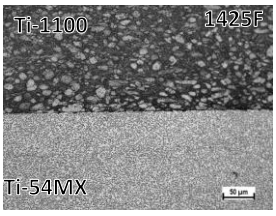


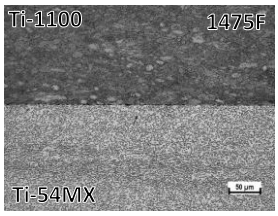


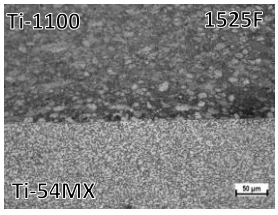
Extensive voids at the two lowest temperatures with some significant voids remaining at 1750F. Delamination occurred at 1650F.

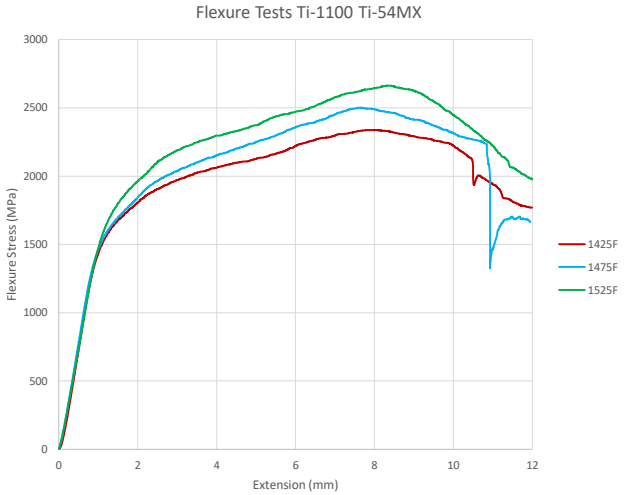


Ti-1100 To Ti-54MX



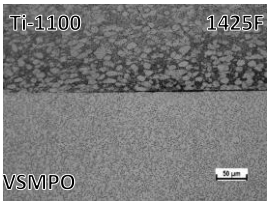


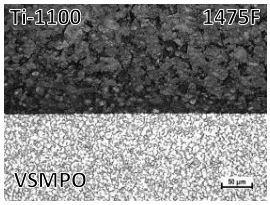


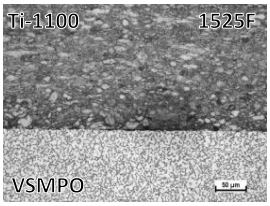


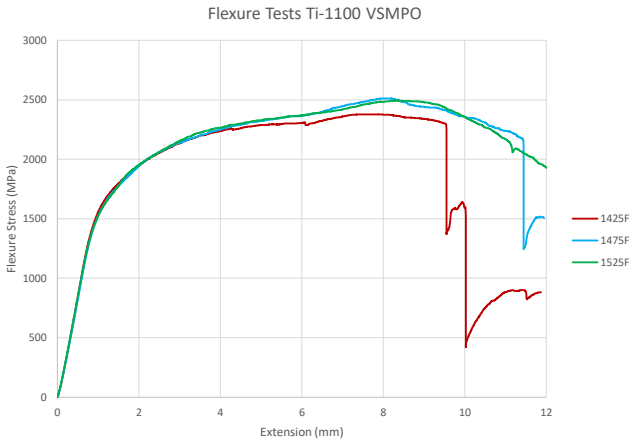
Significant voids at the low temperatures. 1475F delaminated while the 1425F sample was just beginning to delaminate at the end of the test. Still some voids left at 1525F.

Ti-1100 To VSMPO



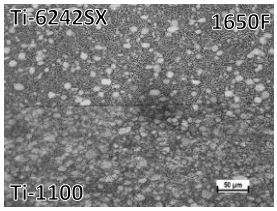


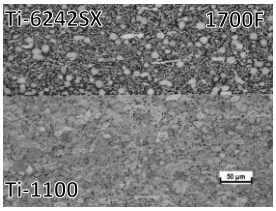


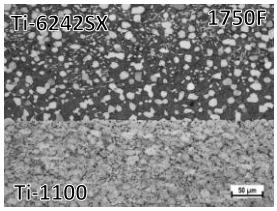


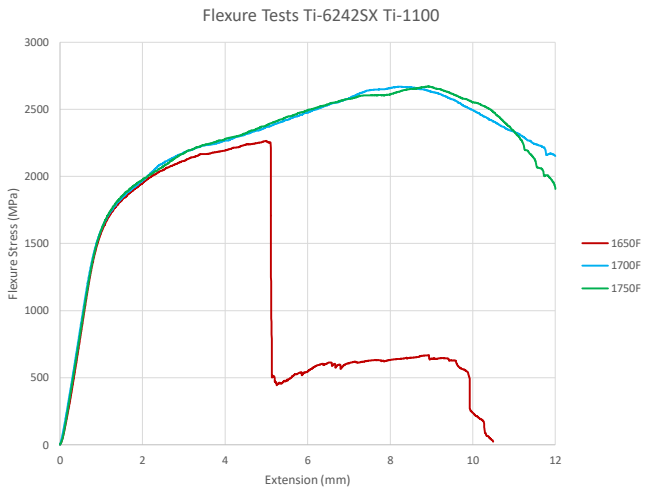
All samples had void problems with delamination occurring at the two low temperatures.

Ti-6242SX To Ti-1100



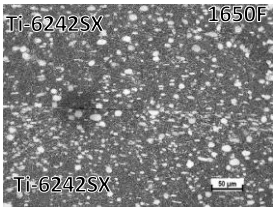


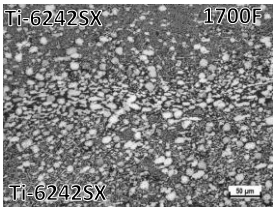


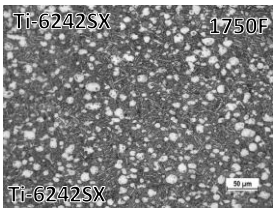


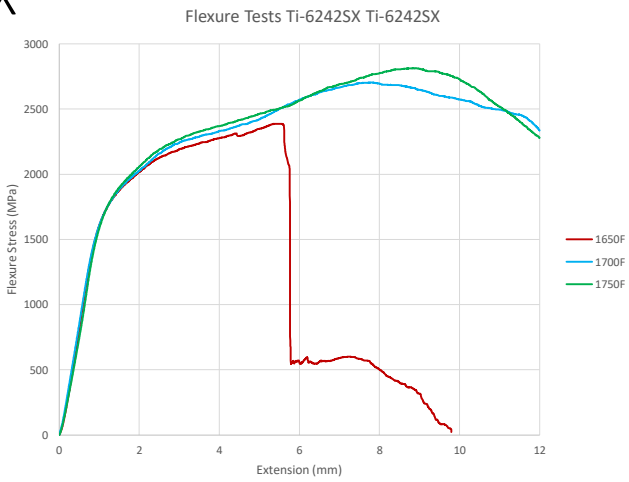
Poor bonding and delamination at 1650F but excellent bonding at the other temperatures

Ti-6242SX To Ti-6242SX



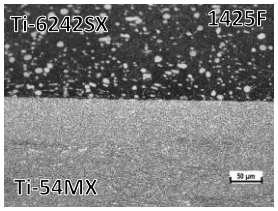


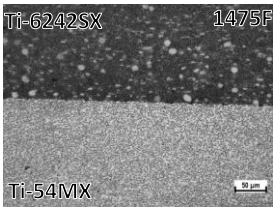


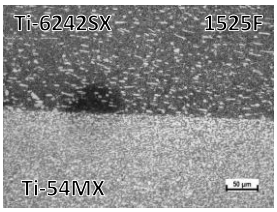


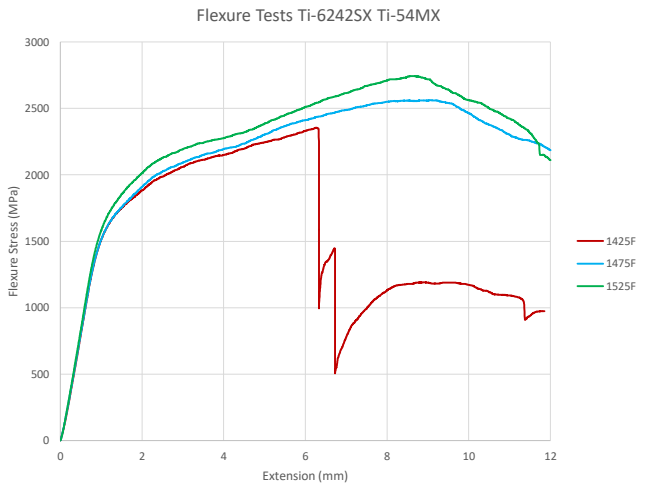
Bonding was poor at 1650F but significantly improved for 1700F and 1750F. The 1650F sample delaminated only slightly during breaking.

Ti-6242SX To Ti-54MX



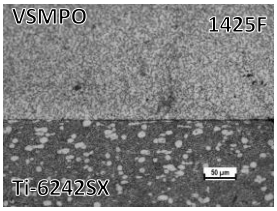


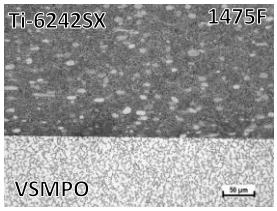


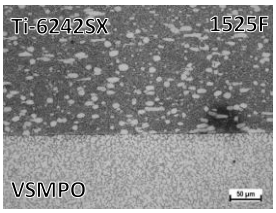


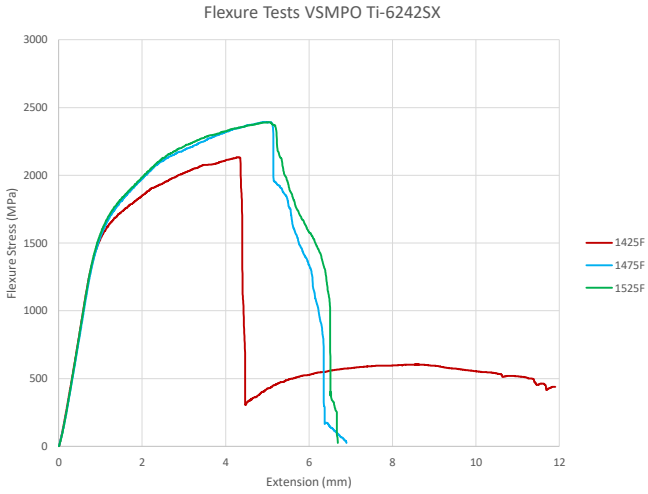
1425F had some significant voids but it was improved for 1475F and good quality at 1525F.

VSMPO To Ti-6242SX



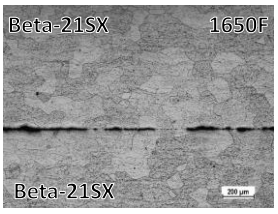


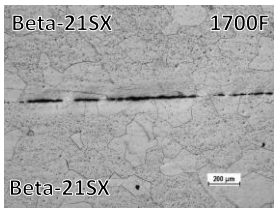


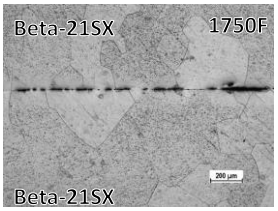


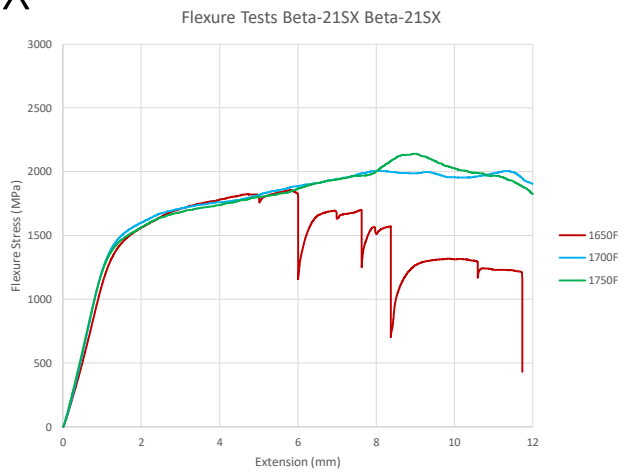
Significant voids occurred in all temperatures with early fracture in all samples.

Beta-21SX To Beta-21SX



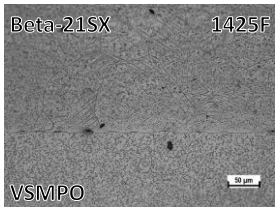


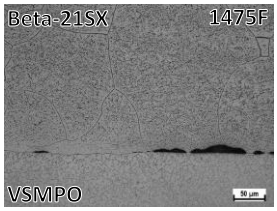


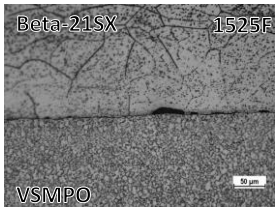


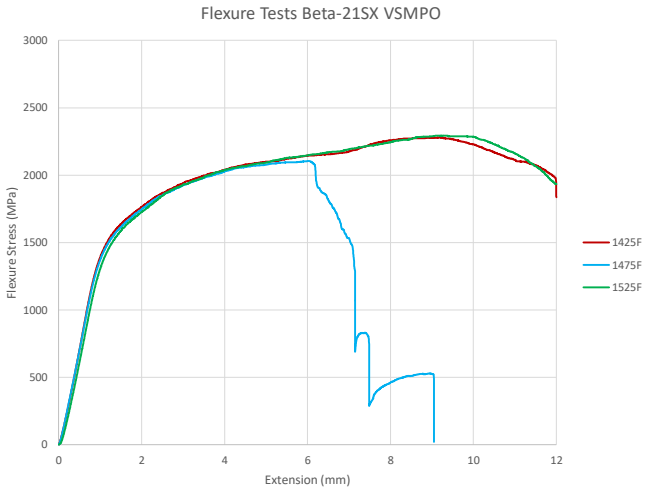
Beta 21SX had consistently poor bonding even to itself.

Beta-21SX To VSMPO



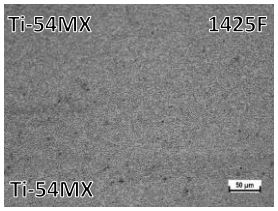


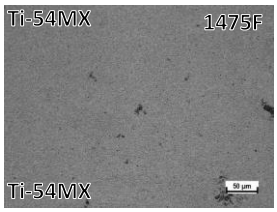


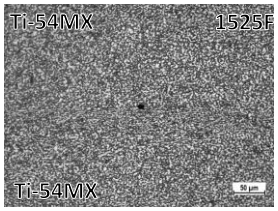


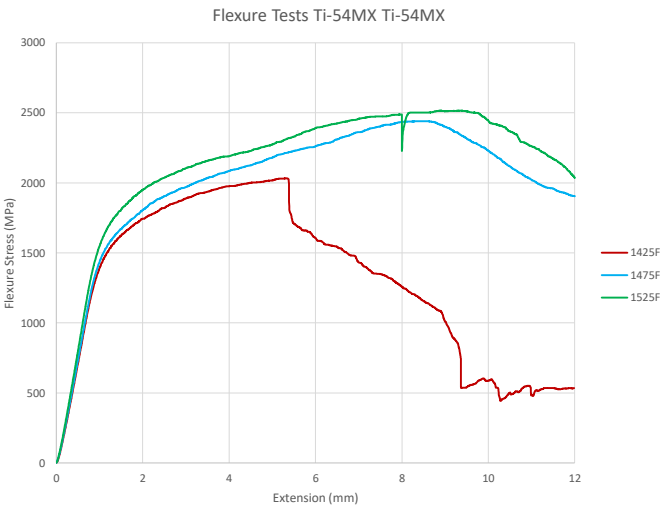
Beta 21SX was improved by combining with VSMPO and formed one of the only quality 1425F bonds. However, large voids remain at the other temperatures.

Ti-54MX To Ti-54MX



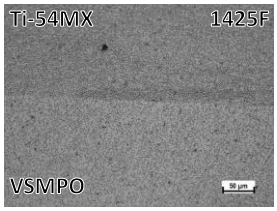


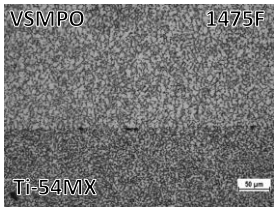


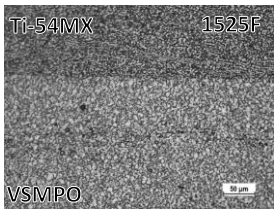


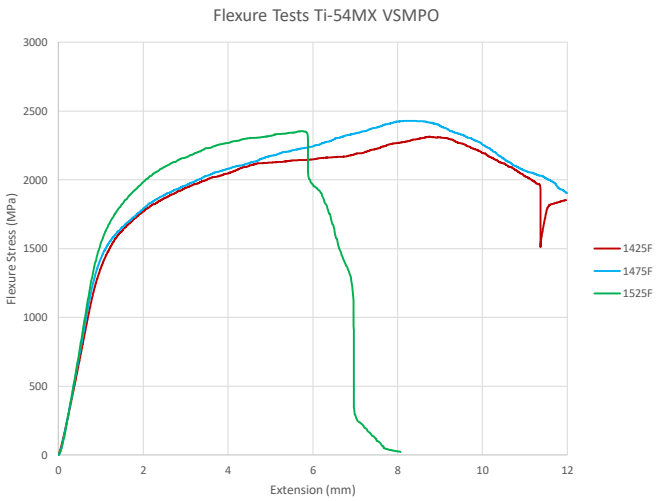
There were a fair number of very small voids in the 1425F but good quality at the other temperatures.

Ti-54MX To VSMPO









Very similar behavior to the rehearsal run with some small voids at 1425F, good bonding at 1475F, and good bonding but some temperature effects at 1525F that cause it to fracture early.

C TIM3 Ultrasonic Results

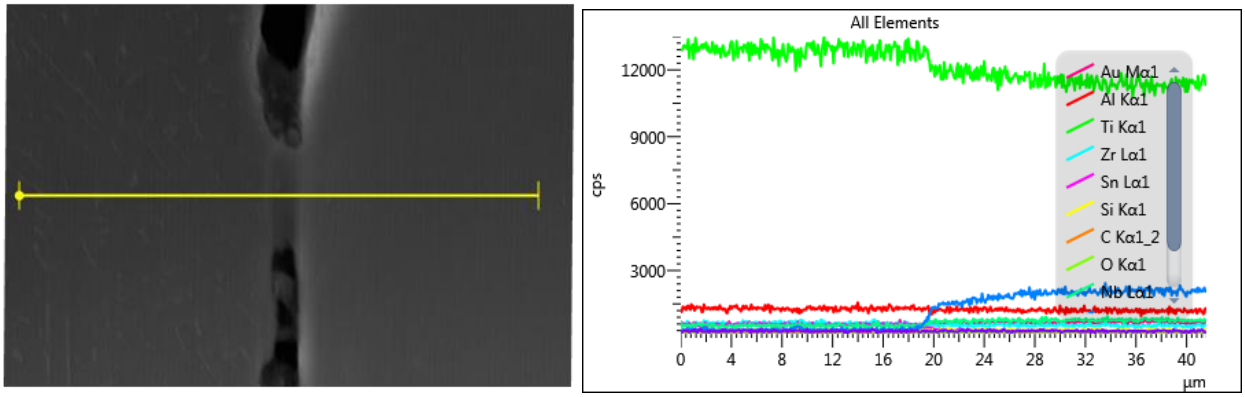
Sample	Temp (F)	Material 1	Material 2	NDT results (C-scan)	Type	NDT notes
1	1425	8	8	No defects found	Manual	
2	1425	8	beta	*Inconclusive*	Manual	Sample not flat
3	1425	8	M	***No defects found***	***Auto***	
4	1425	8	V	***No defects found***	***Auto***	
5	1425	11	11	No defects found	Manual	
6	1425	11	M	***No defects found***	***Auto***	
7	1425	11	V	***No defects found***	***Auto***	***Bent, error due to 2 adjustments***
8	1425	62	11	***No defects found***	***Auto***	***Variation due to gate adjustment***
9	1425	62	62	No defects found	Manual	
10	1425	62	M	***No defects found***	***Auto***	

11	1425	V	62	***No defects found***	***Auto***	***Sample not flat; error along left edge***
12	1425	beta	beta	Inconclusive	Manual	Inconclusive due to rough surface
13	1425	beta	V	***No defects found***	***Auto***	***Error along right edge***
14	1425	M	M	No defects found	Manual	
15	1425	M	V	***No defects found***	***Auto***	***Error due to rough surface along edge***
16	1475	8	8	No defects found	Manual	
17	1475	8	beta	*Inconclusive*	Manual	
18	1475	8	M	***No defects found***	***Auto***	
19	1475	8	V	***No defects found***	***Auto***	***Error due to bond line along one edge***
20	1475	11	11	No defects found	Manual	
21	1475	11	M	***No defects found***	***Auto***	
22	1475	11	V	***No defects found***	***Auto***	***Errors due to 2 adjustments***
23	1475	62	11	***No defects found***	***Auto***	
24	1475	62	62	No defects found	Manual	
25	1475	62	M	***No defects found***	***Auto***	
26	1475	62	V	***No defects found***	***Auto***	
27	1475	beta	beta	Inconclusive	Manual	Inconclusive due to rough surface
28	1475	beta	V	*Inconclusive*	Manual	Sample not flat
29	1475	M	M	No defects found	Manual	
30	1475	V	M	***No defects found***	***Auto***	***2 protrusions on top***
31	1525	8	8	No defects found	Manual	
32	1525	beta	8	*Inconclusive*	Manual	Sample not flat

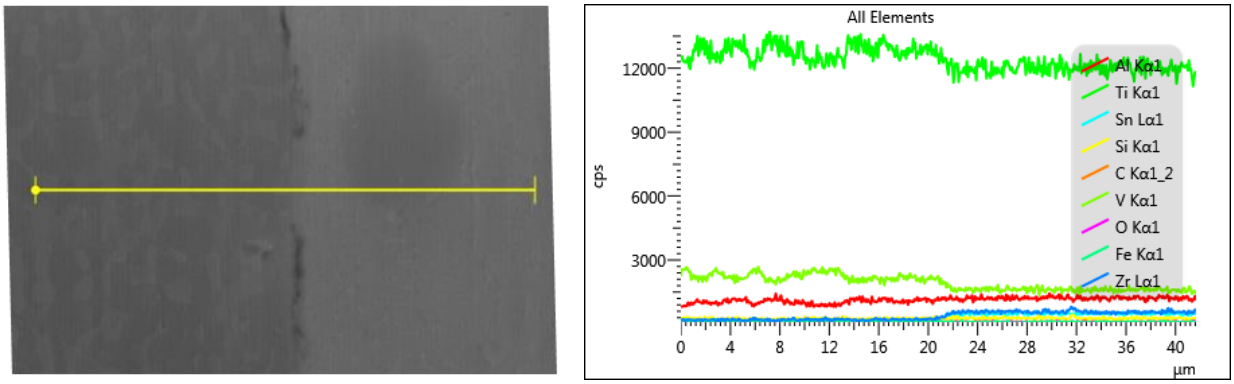
33	1525	8	M	Not tested	Manual	*Tested after cutting*
34	1525	8	V	***No defects found***	***Auto***	
35	1525	11	11	No defects found	Manual	
36	1525	11	M	***No defects found***	***Auto***	
37	1525	11	V	***No defects found***	***Auto***	
38	1525	62	11	***No defects found***	***Auto***	***Sample bent, scanned vertically upside down***
39	1525	62	62	No defects found	Manual	
40	1525	62	M	***No defects found***	***Auto***	***Sample bent, error due to adjustment***
41	1525	62	V	***No defects found***	***Auto***	
42	1525	beta	beta	Inconclusive	Manual	Inconclusive due to rough surface
43	1525	beta	V	***No defects found***	***Auto***	
44	1525	M	M	No defects found	Manual	
45	1525	M	V	***No defects found***	***Auto***	

D EDX TIM3 Samples

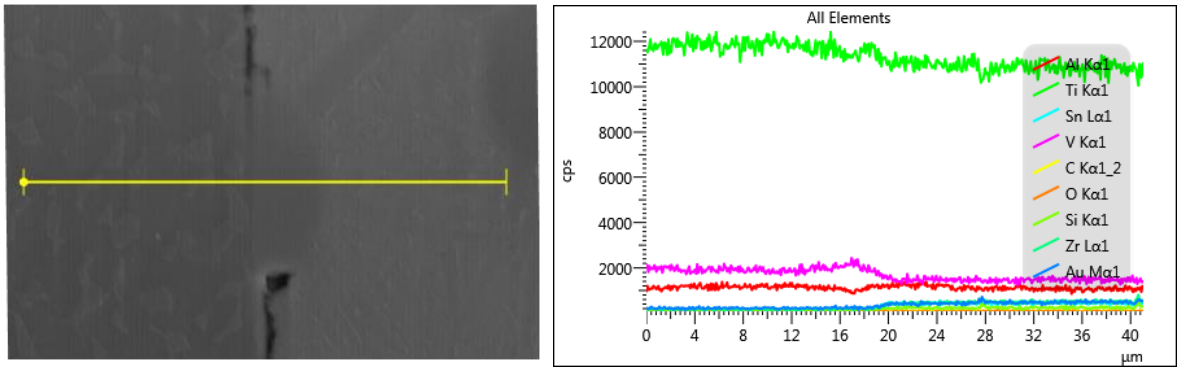
Sample 2 – Ti-834 to Beta-21SX 1650F



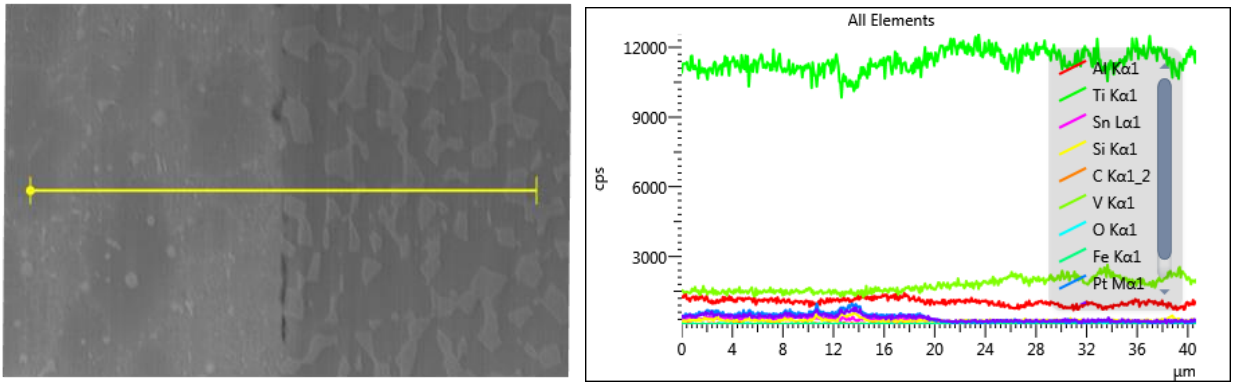
Sample 3 – Ti-834 to Ti-54MX 1425F



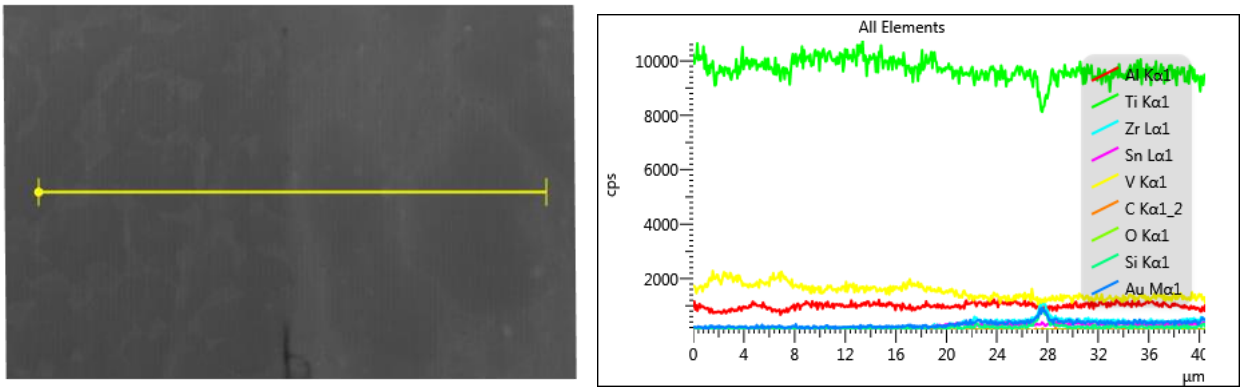
Sample 4 – Ti-834 to VSMPO 1425F



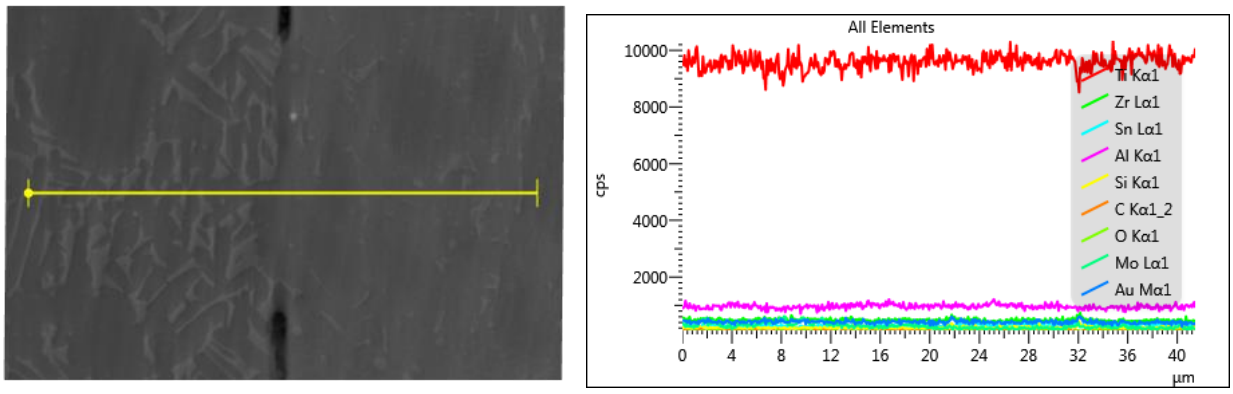
Sample 6 – Ti-1100 to Ti-54MX 1425F



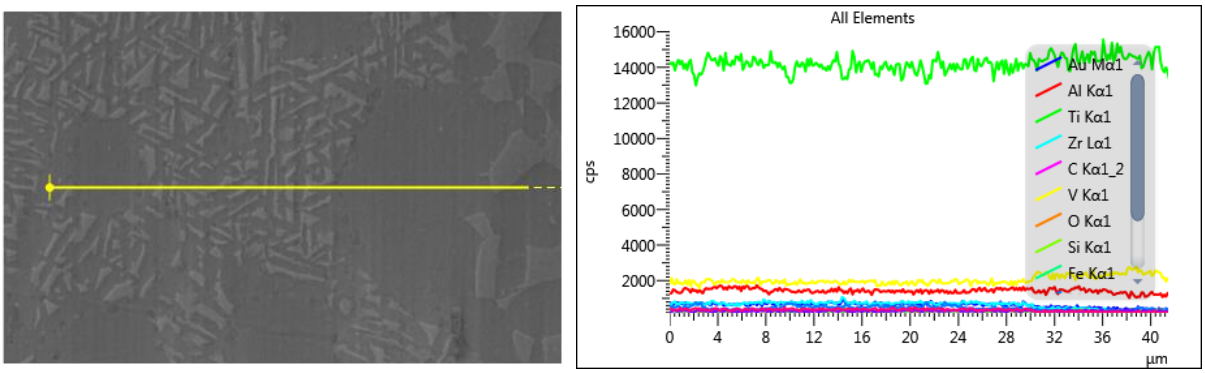
Sample 7 – Ti-1100 to VSMPO 1425F



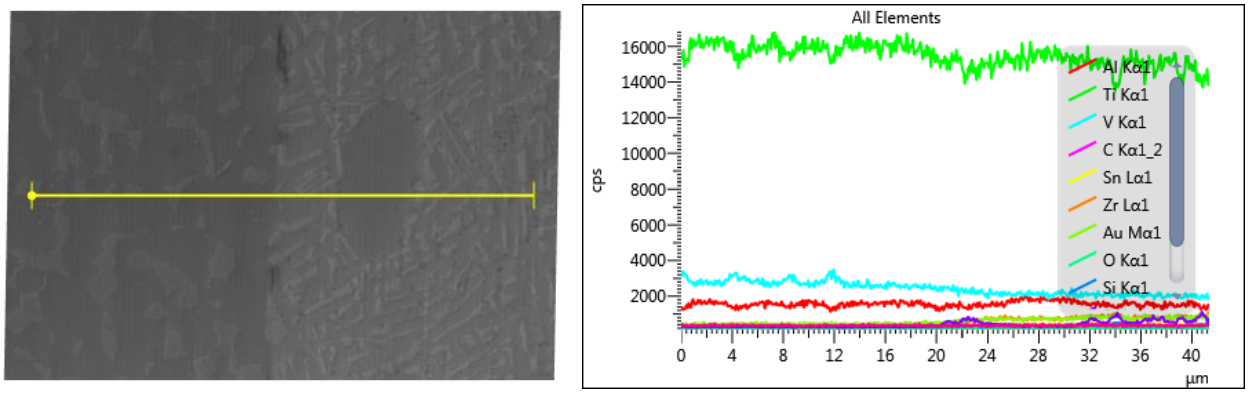
Sample 8 – Ti-1100 to Ti-6242SX 1650F



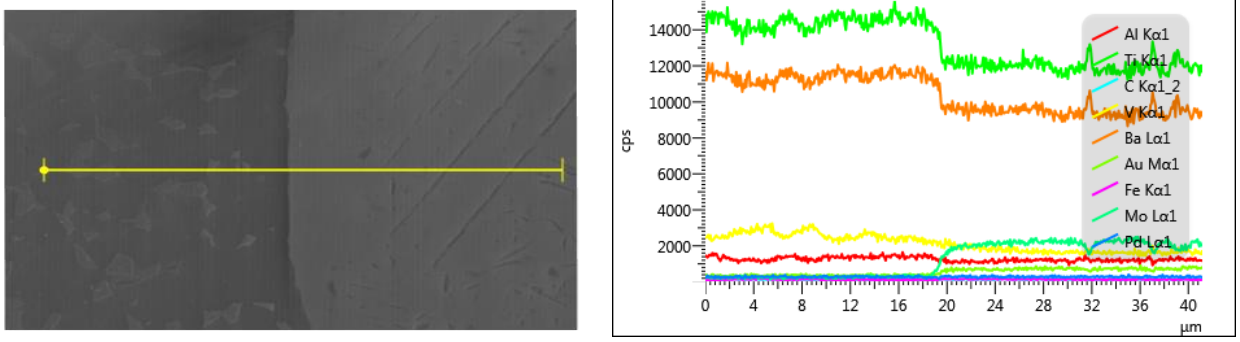
Sample 10 – Ti-1100 to Ti-54MX 1425F



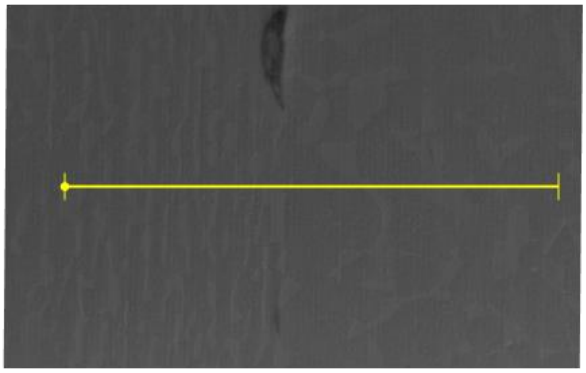
Sample 11 – VSMPO to Ti-6242SX 1425F

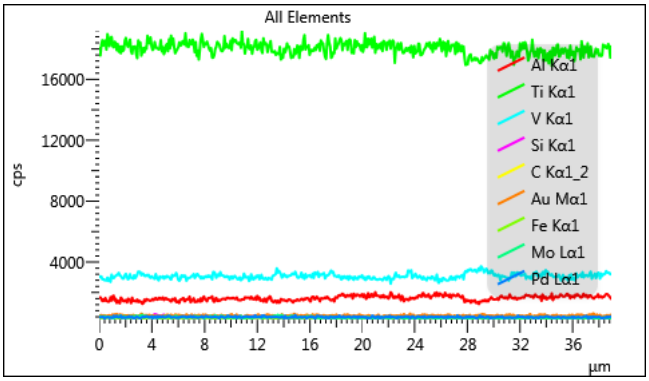


Sample 13 – Beta-21SX to VSMPO 1425F (Barium is a false positive from Titanium and can be ignored)

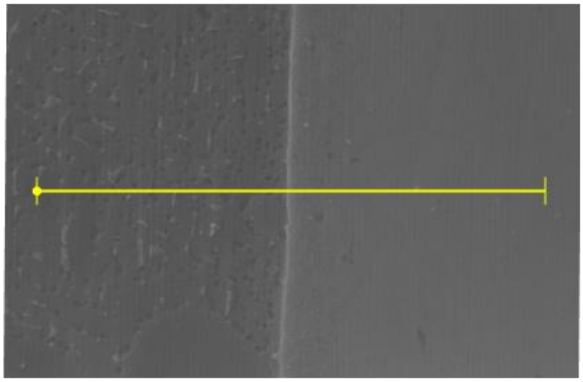


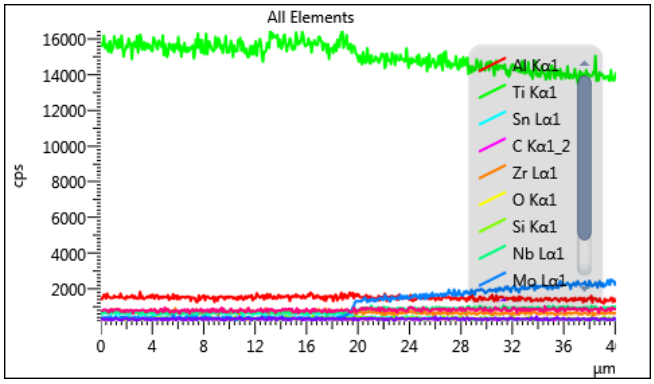
Sample 15 – Ti-54MX to VSMPO 1425F



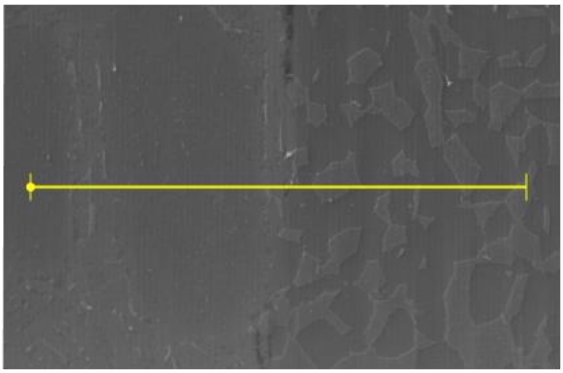


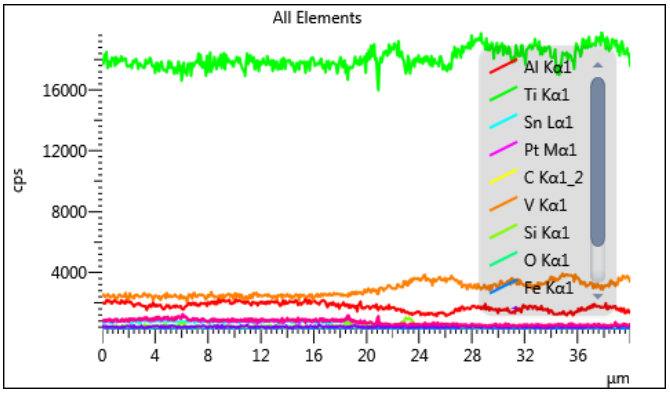
Sample 17 – Ti-834 to Beta-21SX 1700F



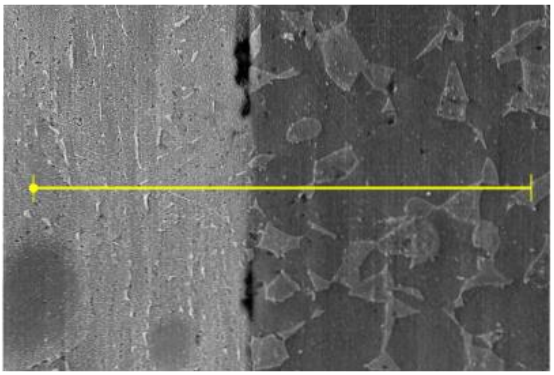


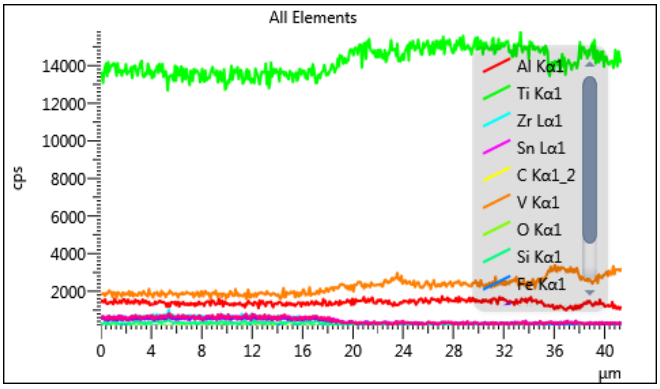
Sample 18 – Ti-834 to Ti-54MX 1475F



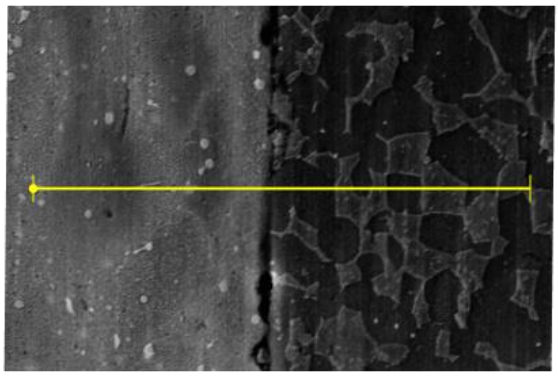


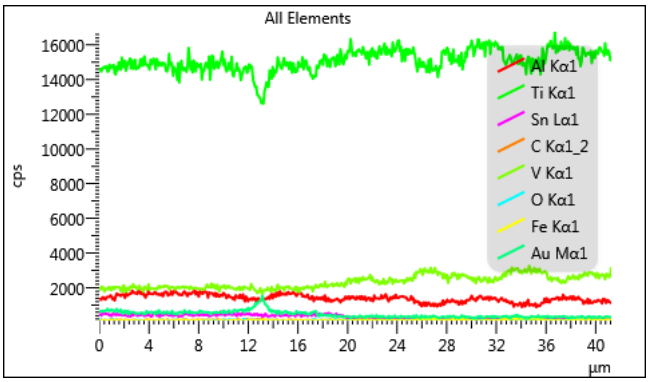
Sample 19 – Ti-834 to VSMPO 1475F



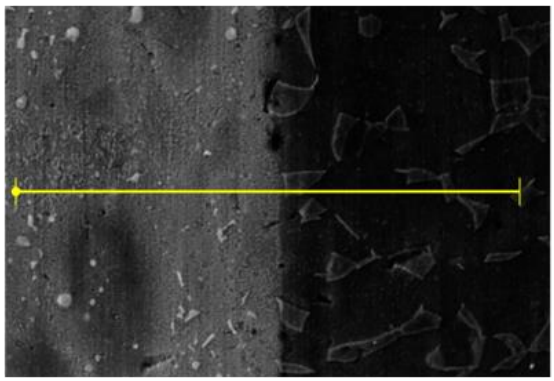


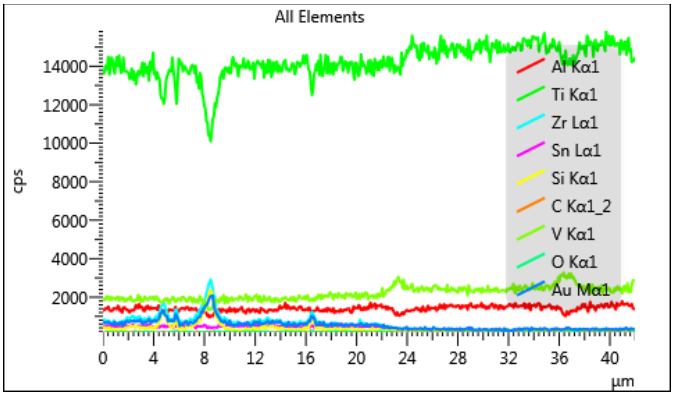
Sample 21 – Ti-1100 to Ti-54MX 1475F



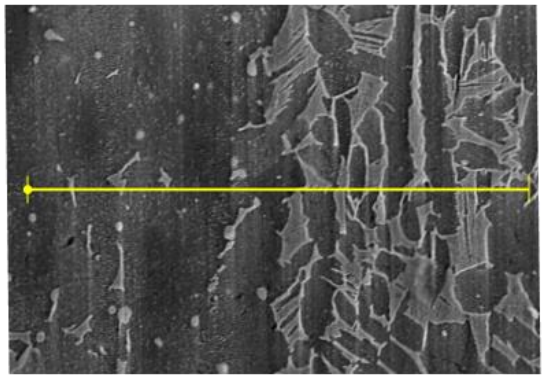


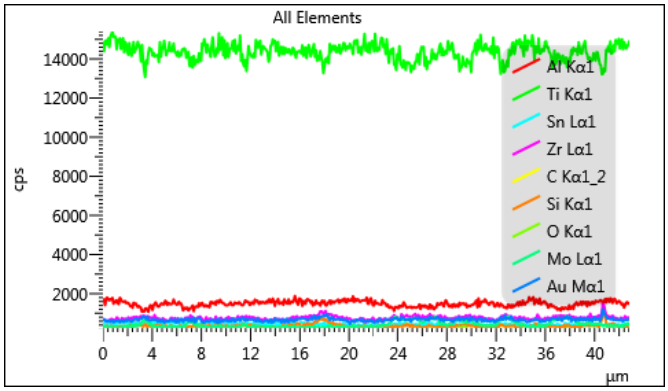
Sample 22 – Ti-1100 to VSMPO 1475F



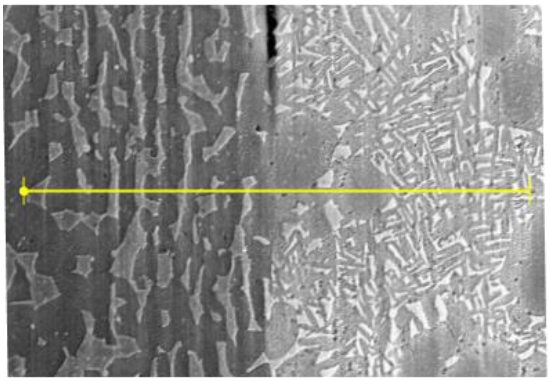


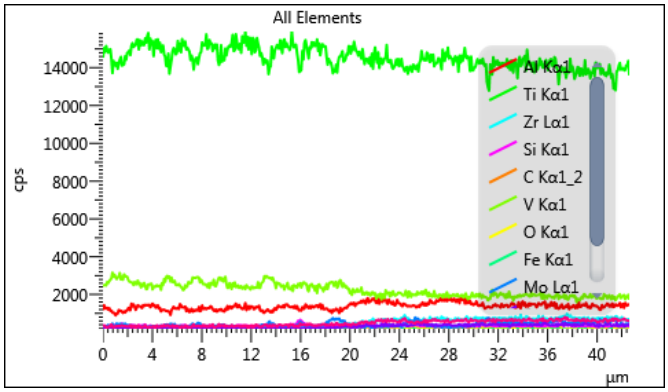
Sample 23 – Ti-1100 to Ti-6242SX 1700F



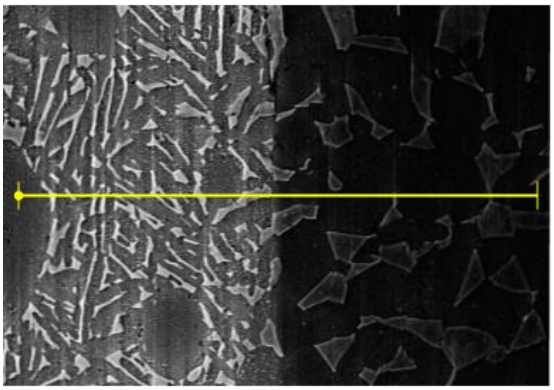


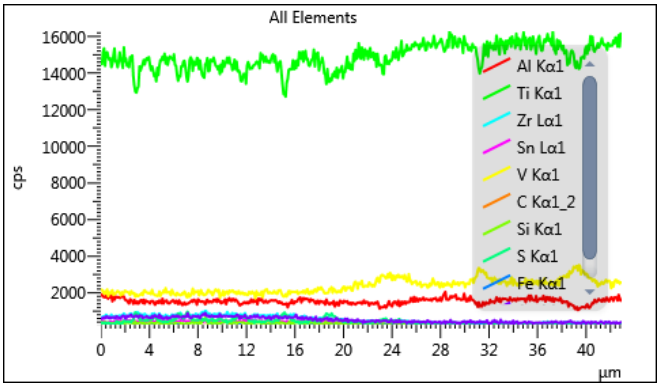
Sample 25 – Ti-1100 to Ti-54MX 1475F



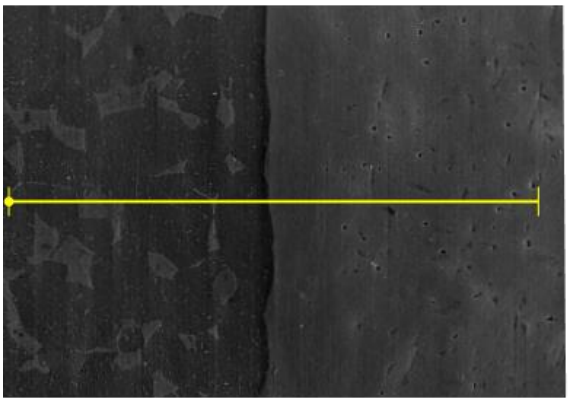


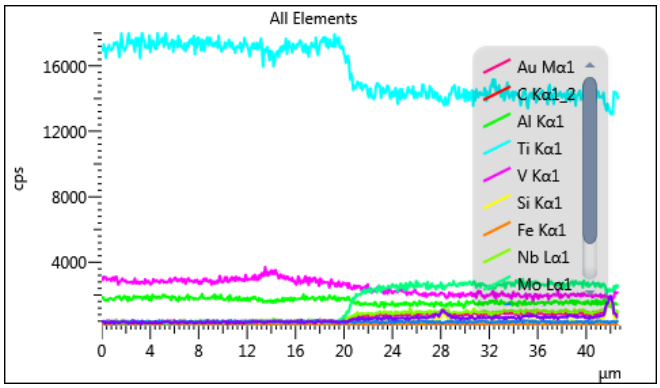
Sample 26 – VSMPO to Ti-6242SX 1475F



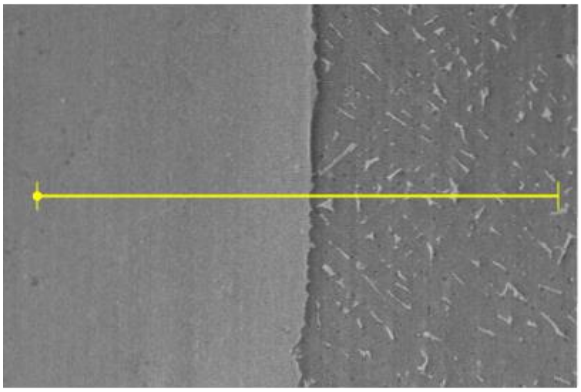


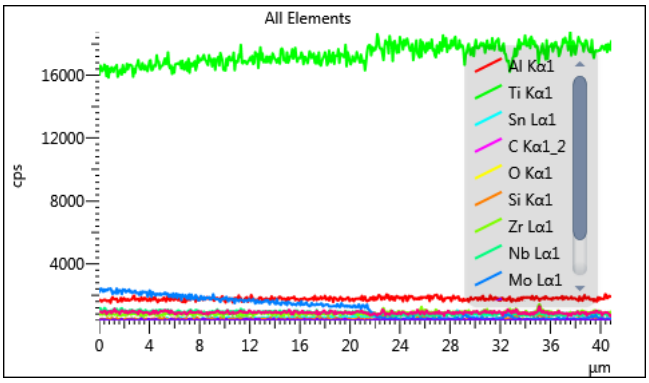
Sample 28 – Beta-21SX to VSMPO 1475F



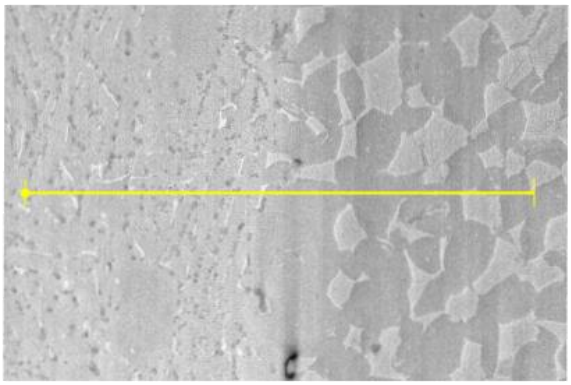


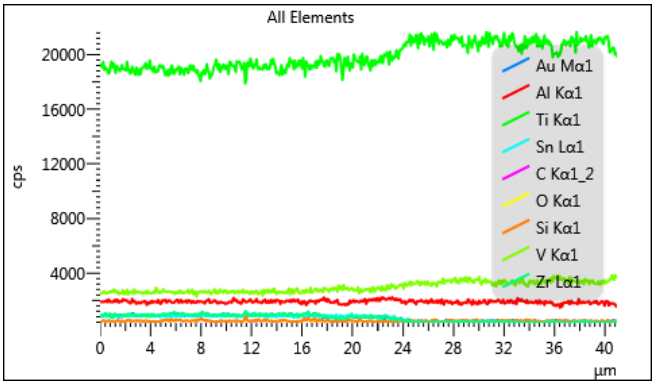
Sample 32 – Ti-834 to Beta-21SX 1750F



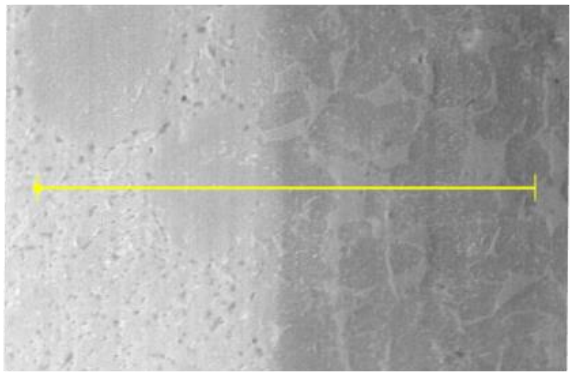


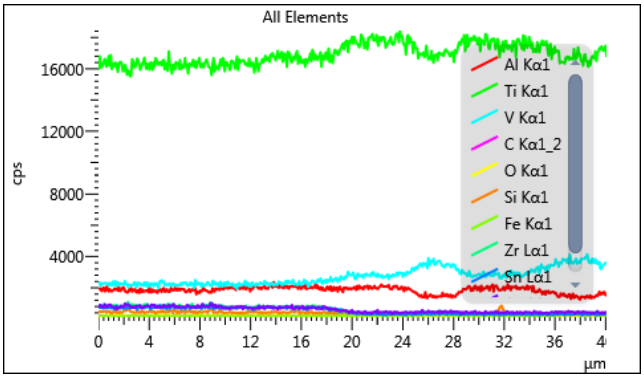
Sample 33 – Ti-834 to Ti-54MX 1425F



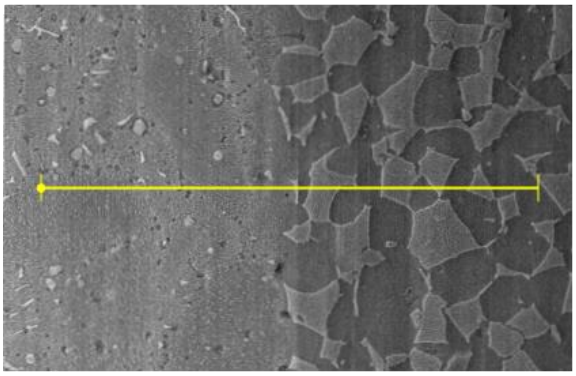


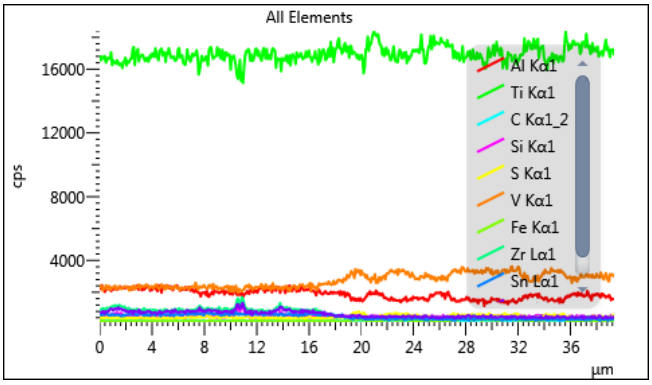
Sample 34 – Ti-834 to VSMPO 1425F



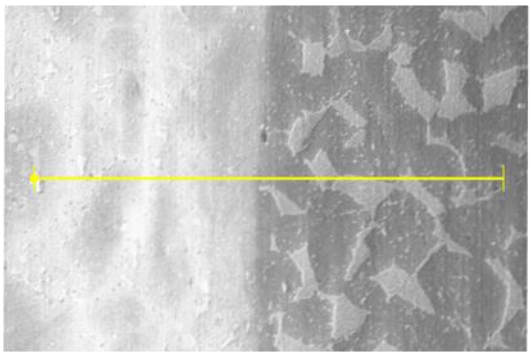


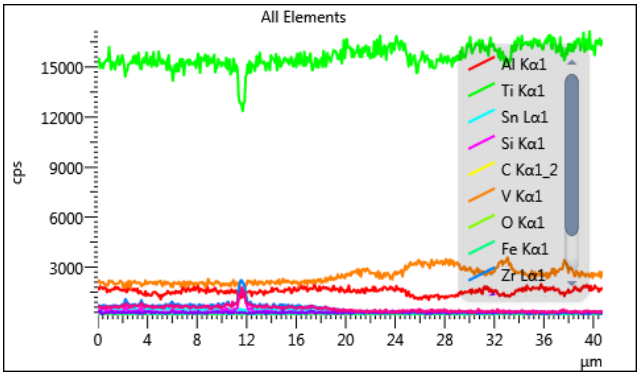
Sample 36 – Ti-1100 to Ti-54MX 1425F



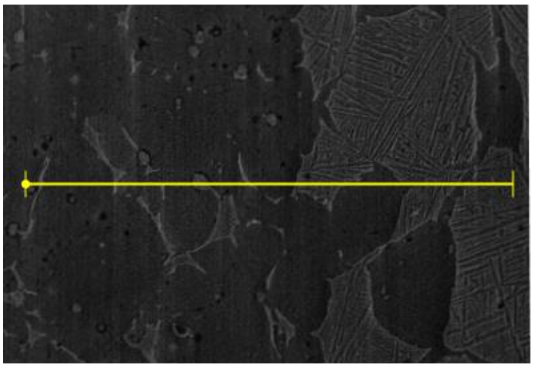


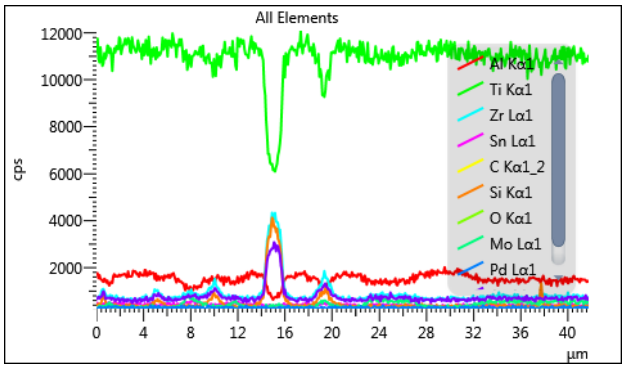
Sample 37 – Ti-1100 to VSMPO 1425F



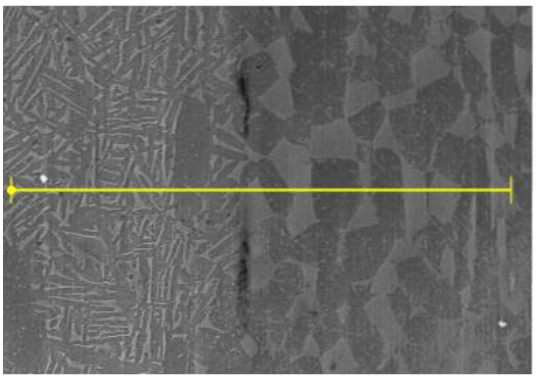


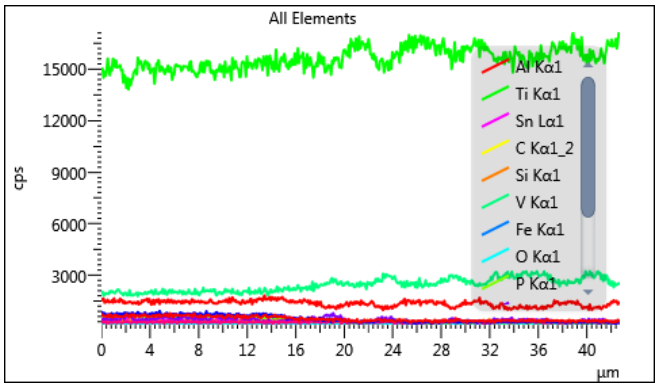
Sample 38 – Ti-1100 to Ti-6242SX 1750F



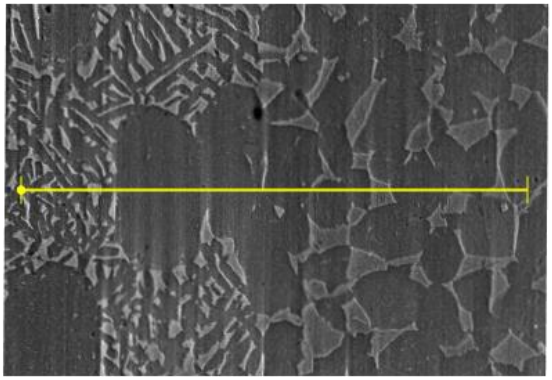


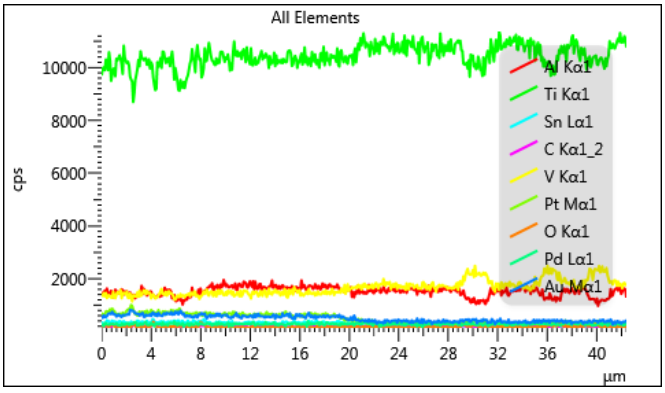
Sample 40 – Ti-1100 to Ti-54MX 1425F



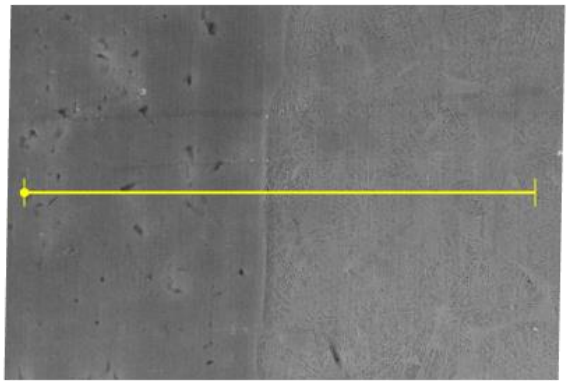


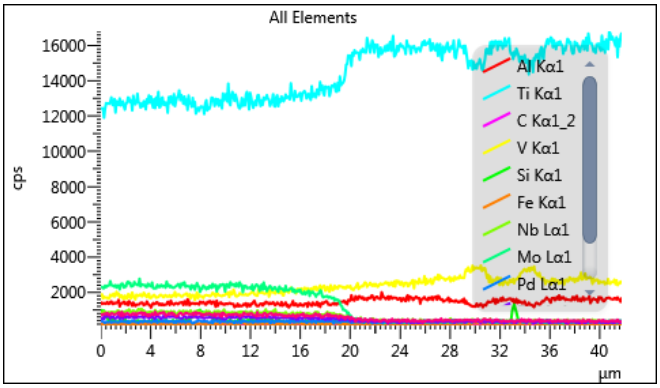
Sample 41 – VSMPO to Ti-6242SX 1425F





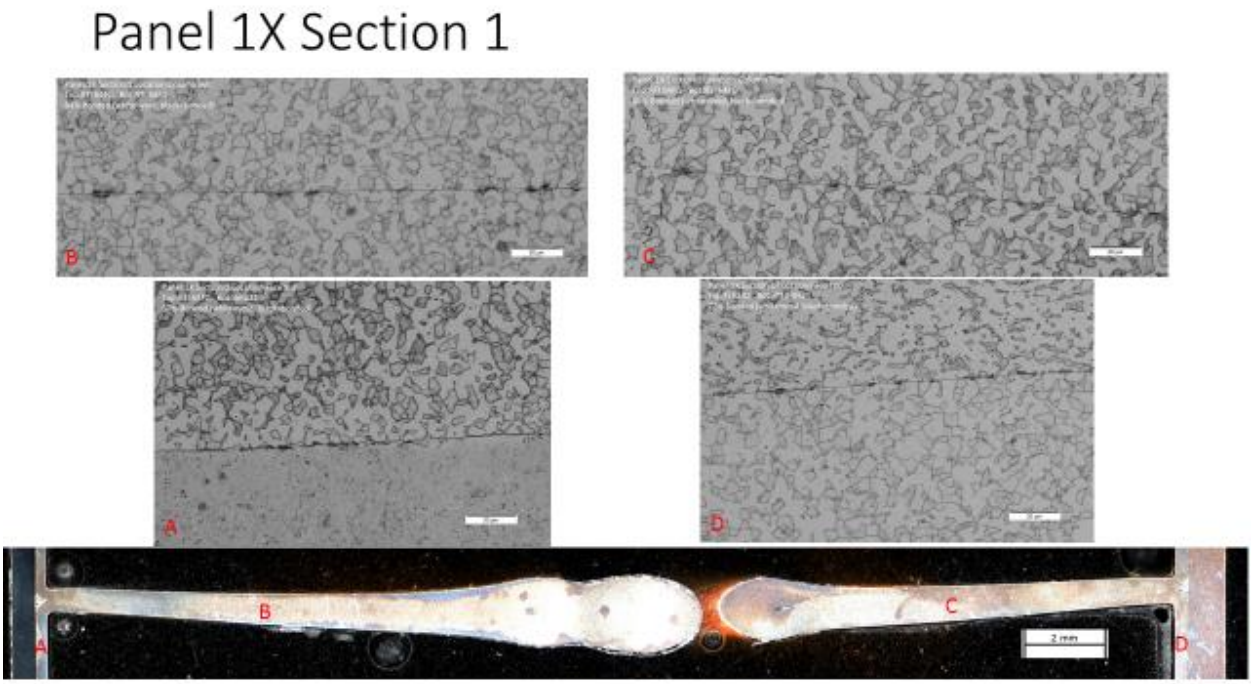
Sample 43 – Beta-21SX to VSMPO 1425F



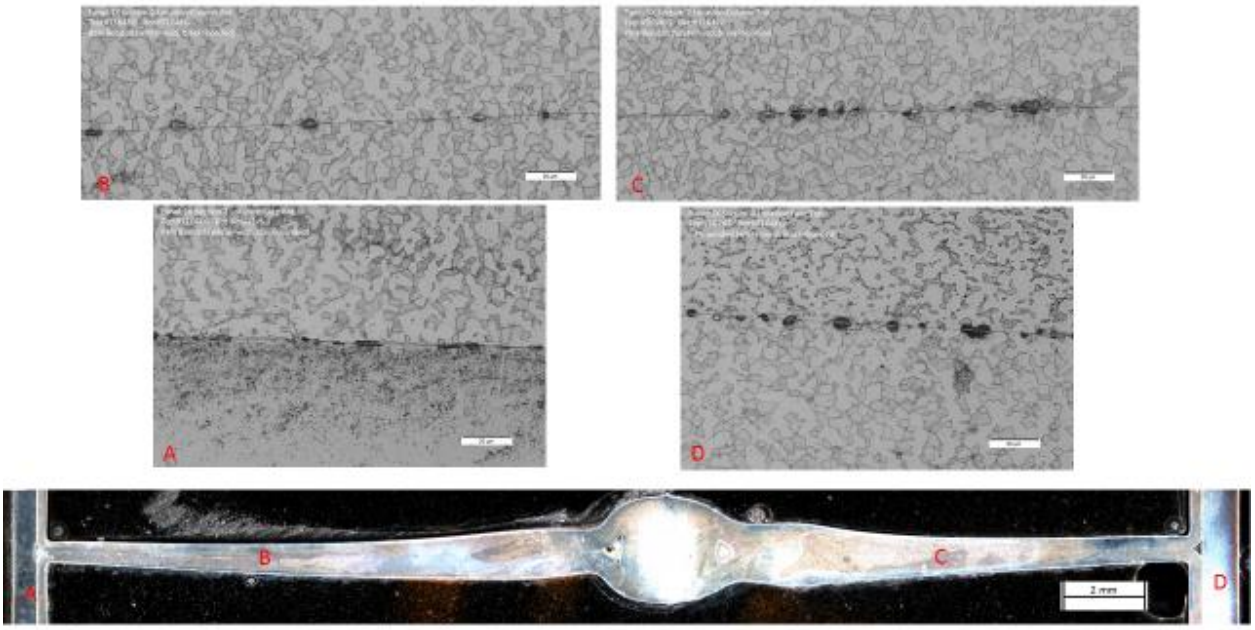


E Four-sheet Process Characterization

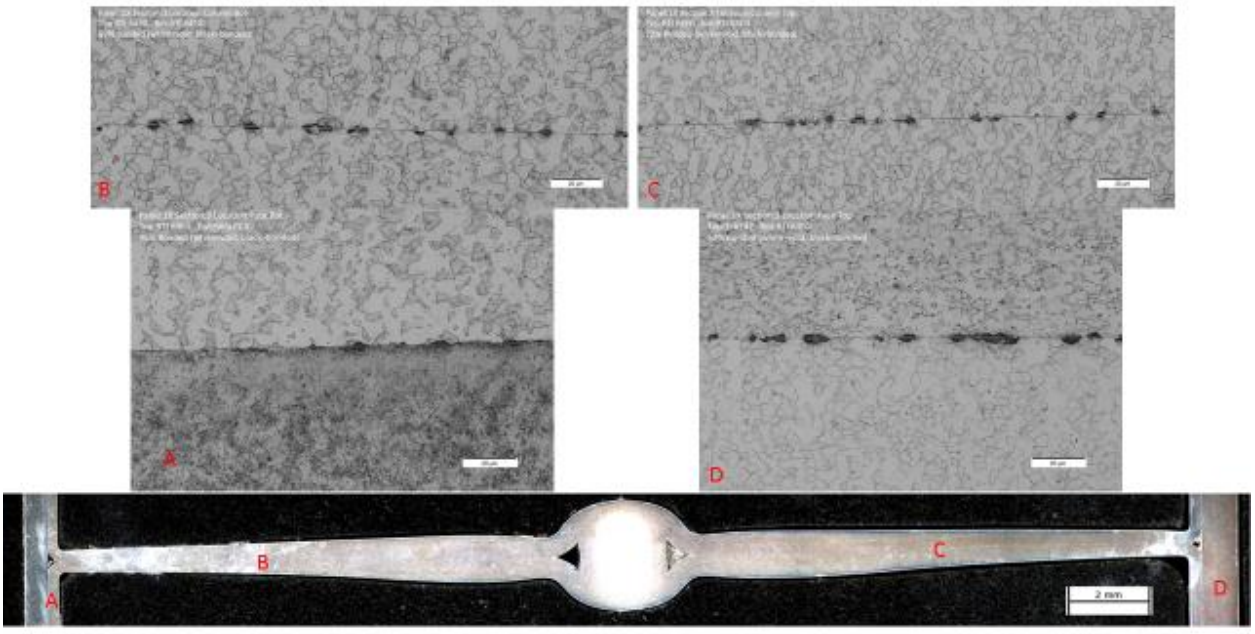
BA18 Panel Group 1



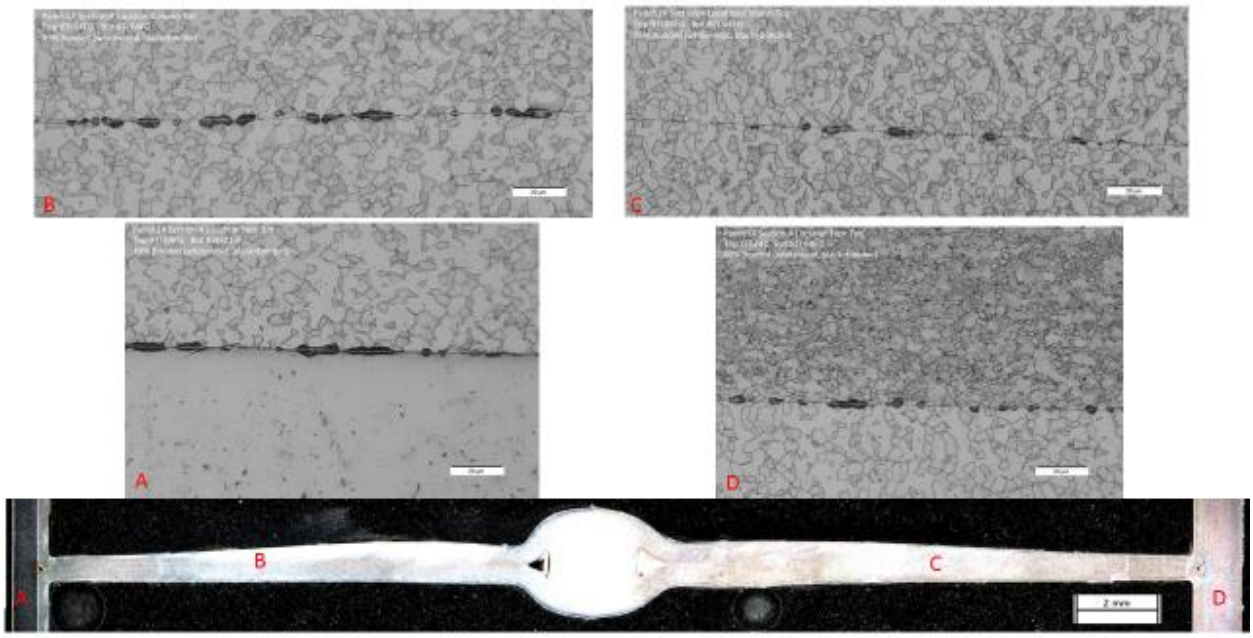
Panel 1X Section 2



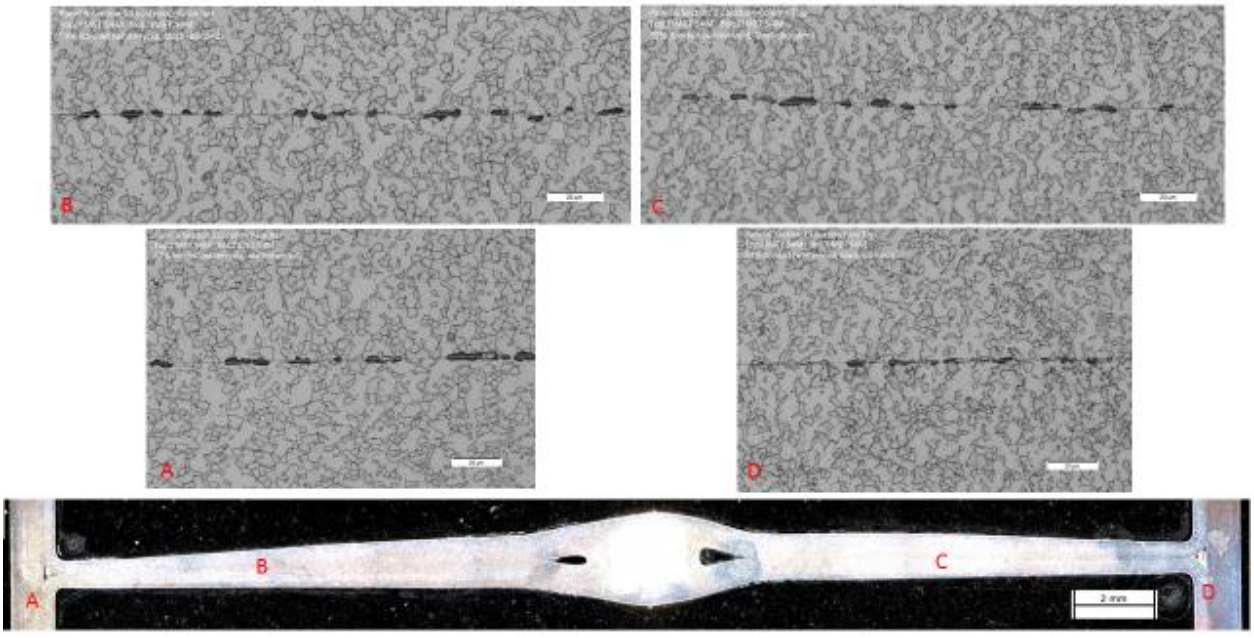
Panel 1X Section 3



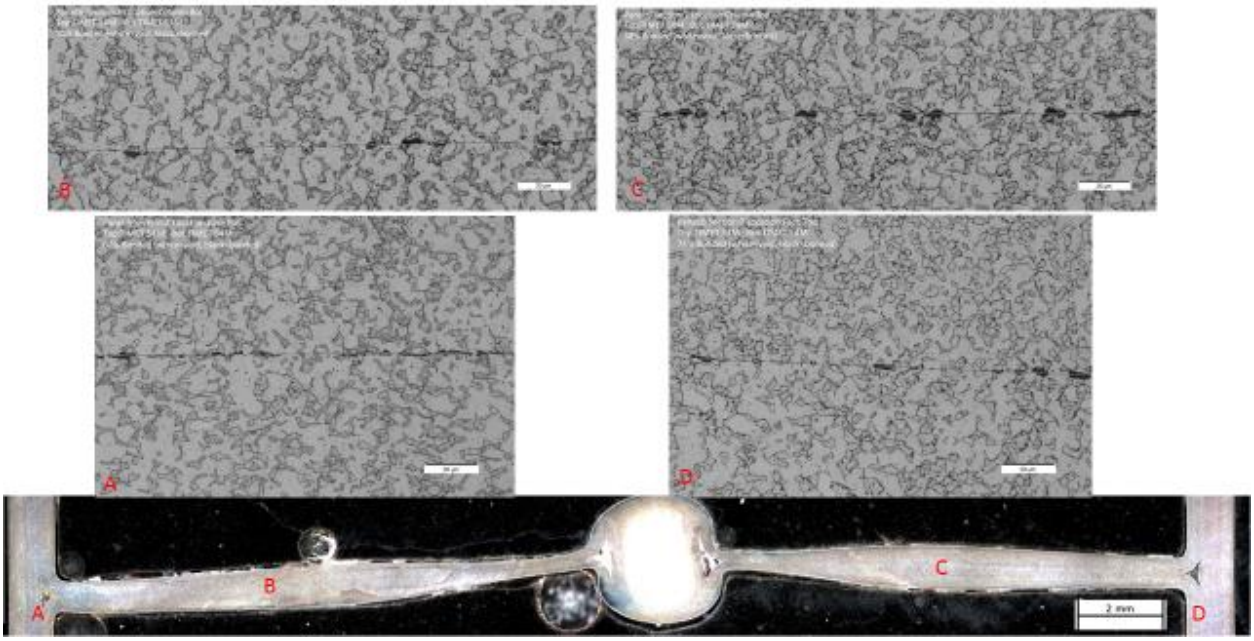
Panel 1X Section 4



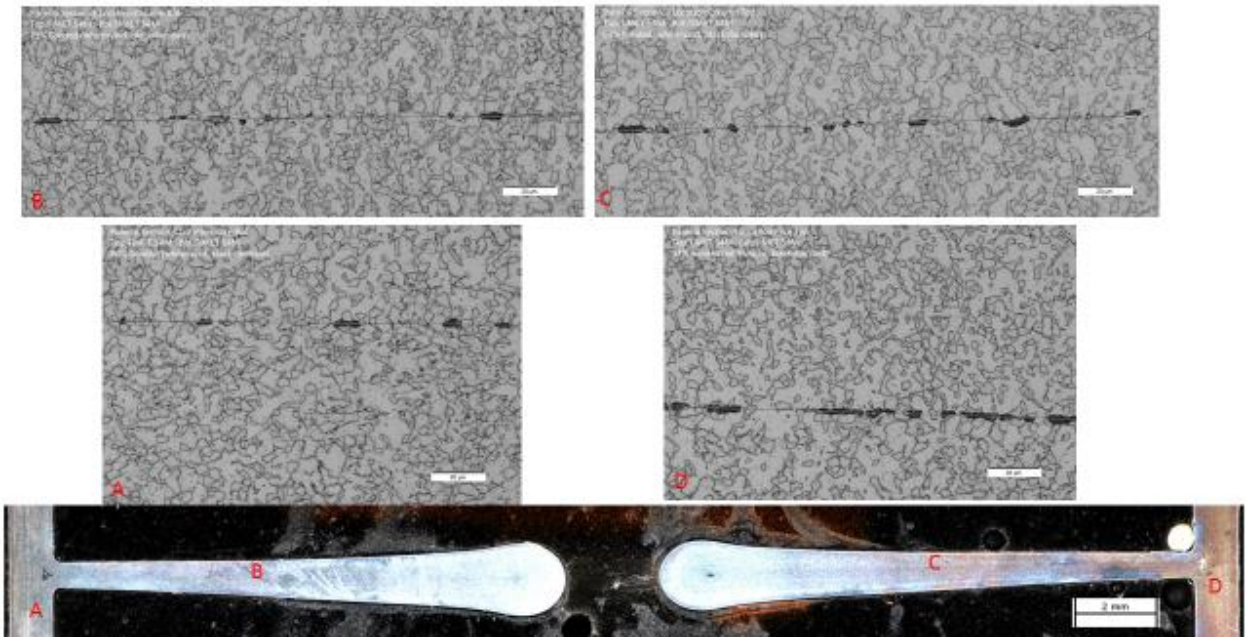
Panel 6 Section 1



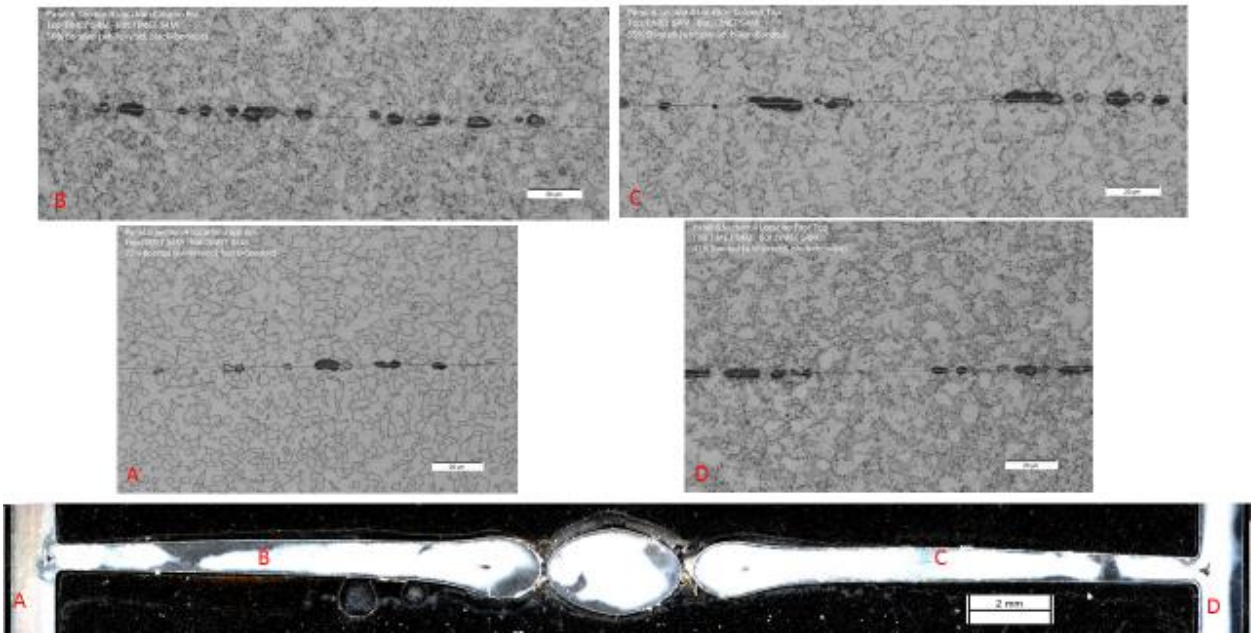
Panel 6 Section 2



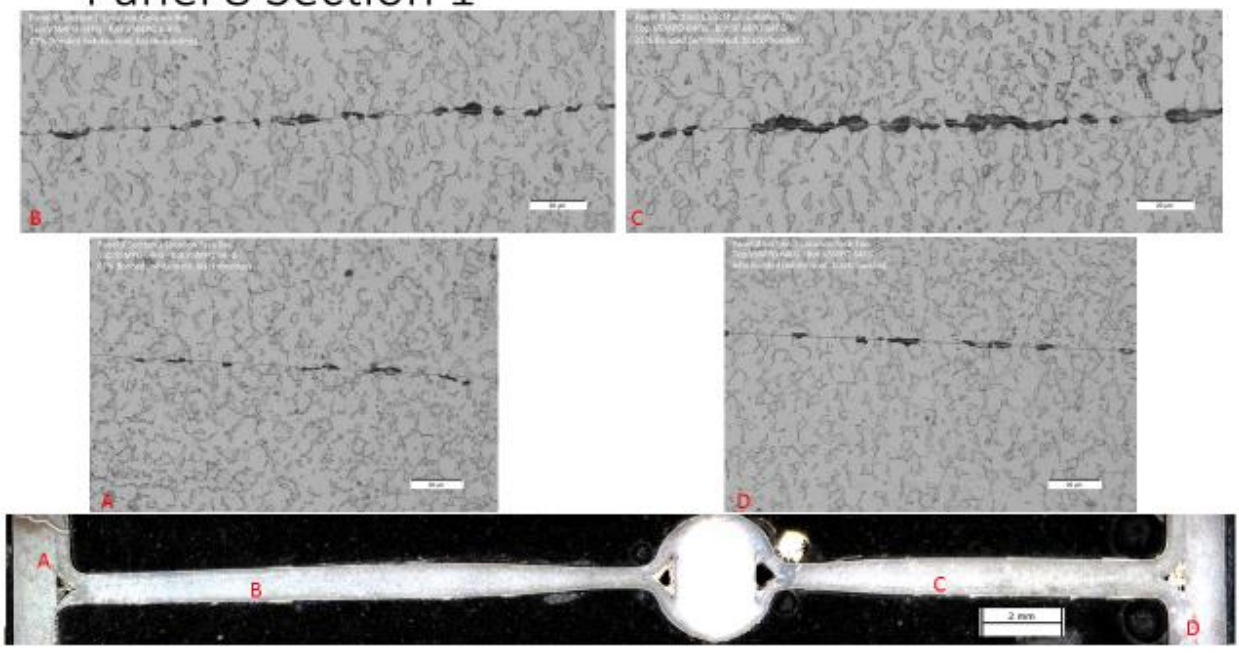
Panel 6 Section 3



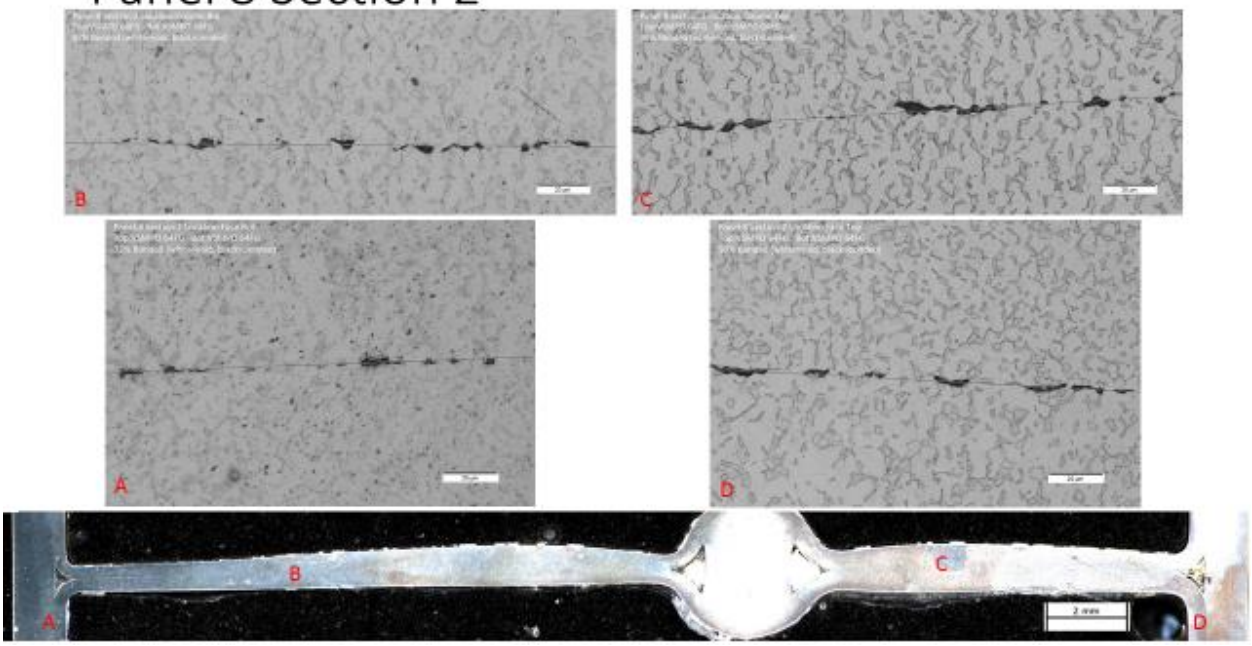
Panel 6 Section 4



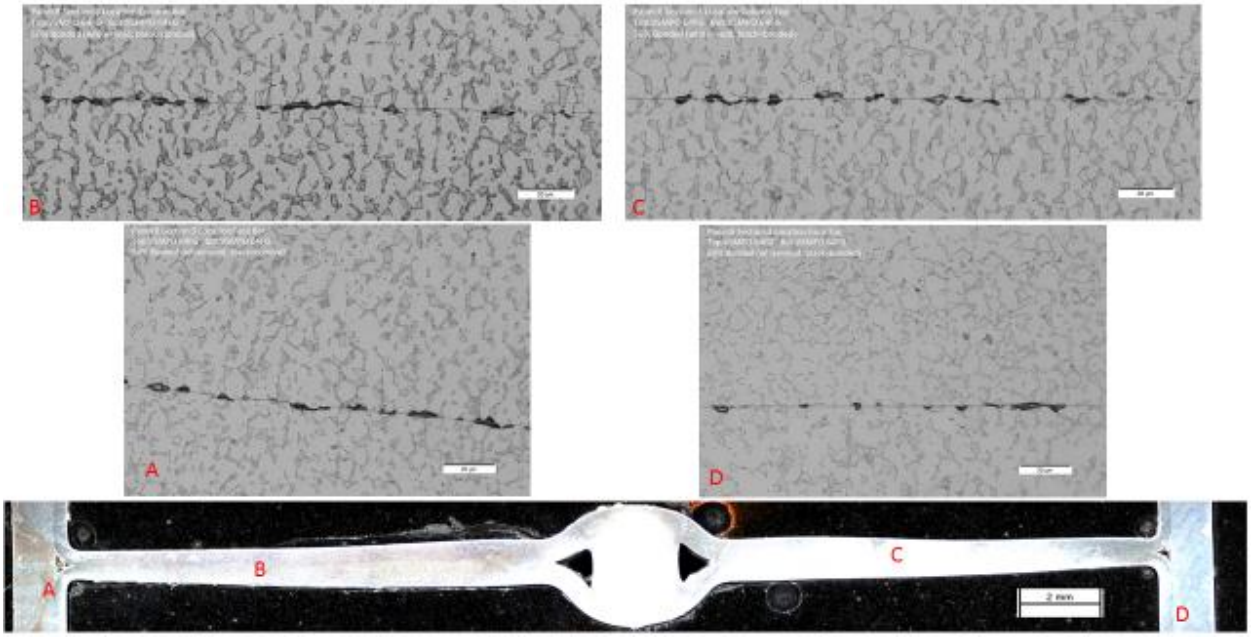
Panel 8 Section 1



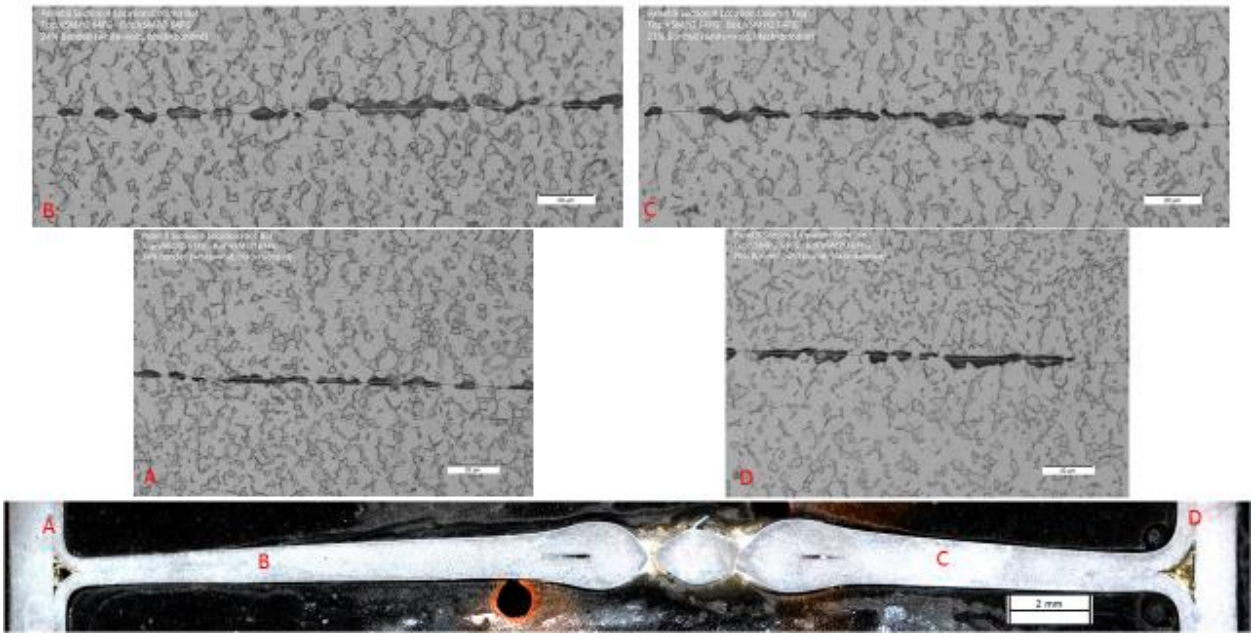
Panel 8 Section 2



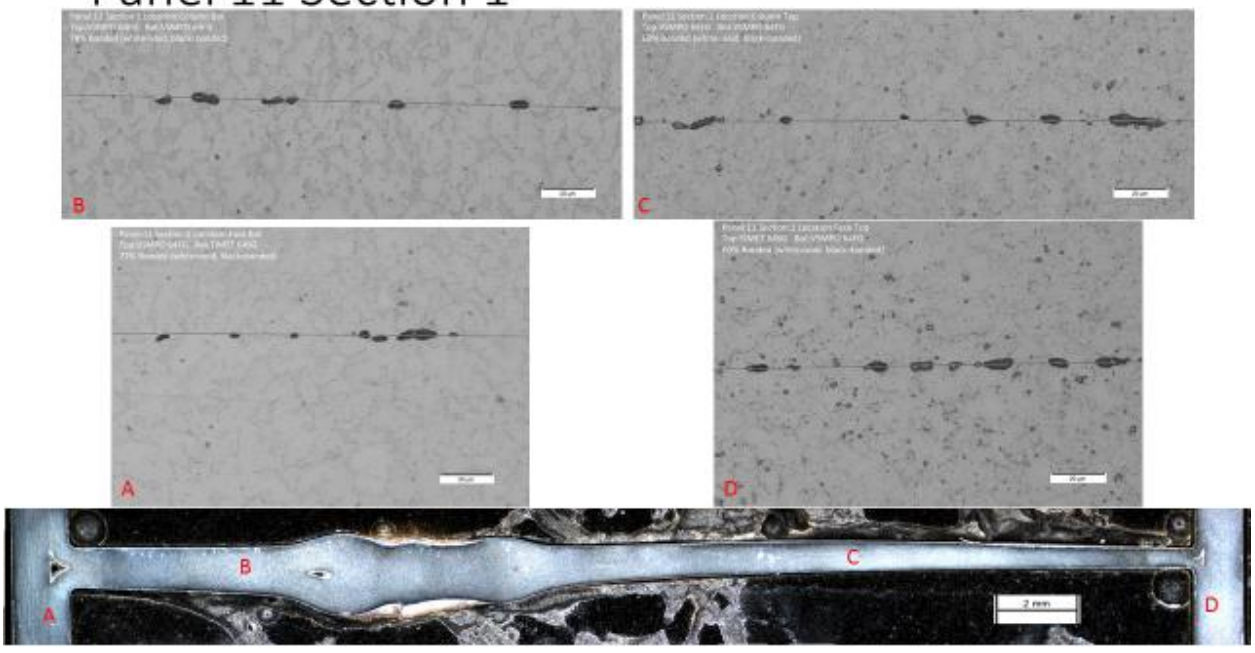
Panel 8 Section 3



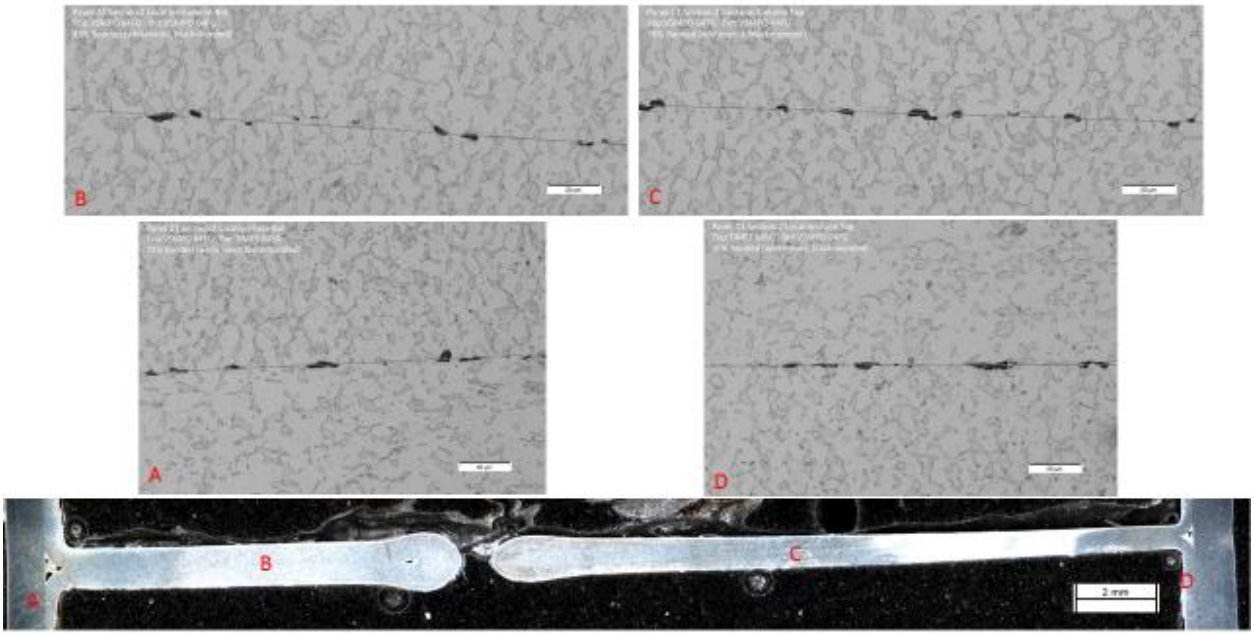
Panel 8 Section 4



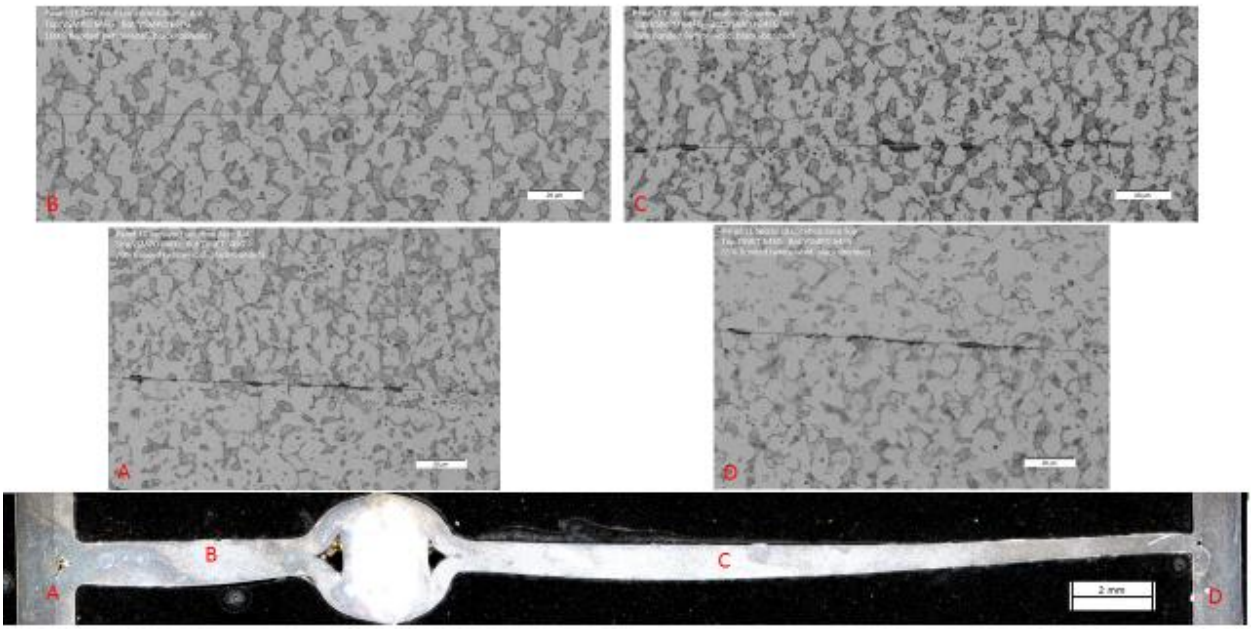
Panel 11 Section 1



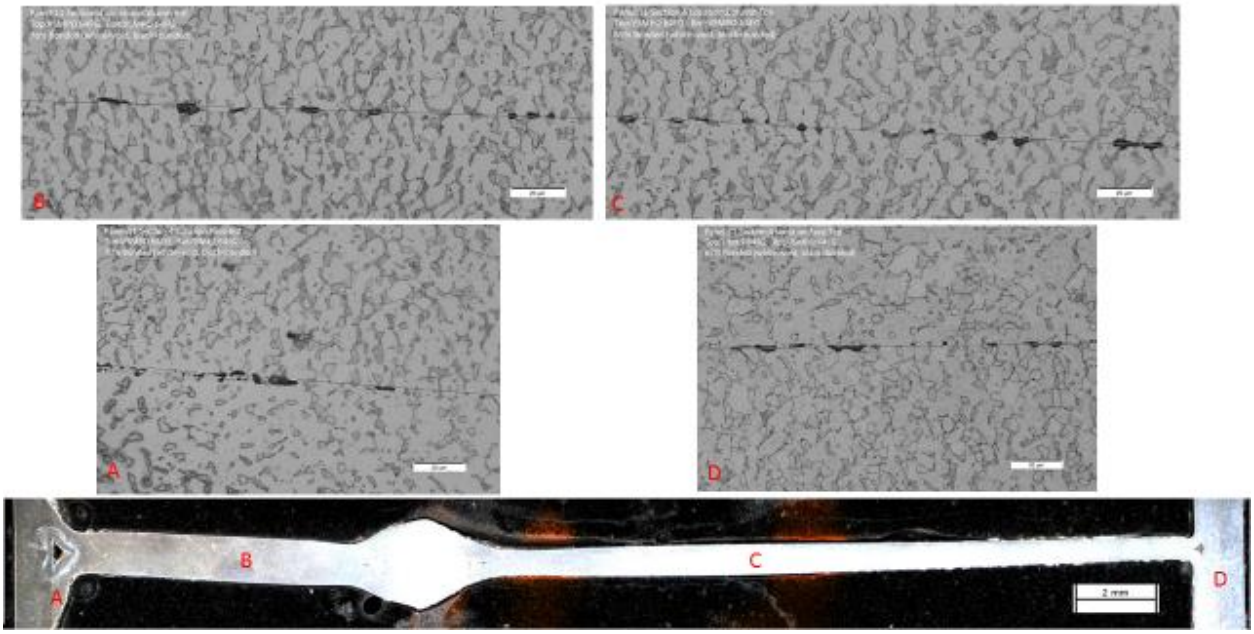
Panel 11 Section 2

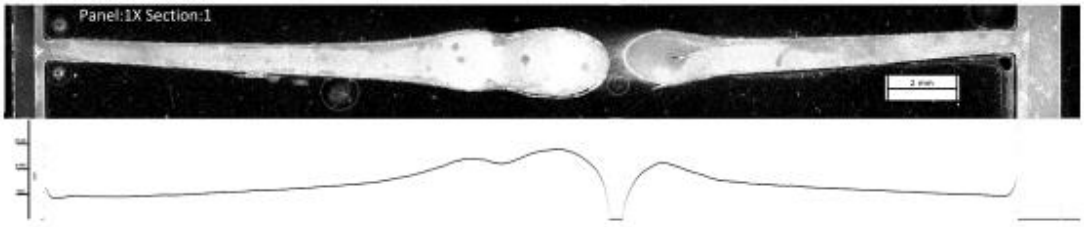


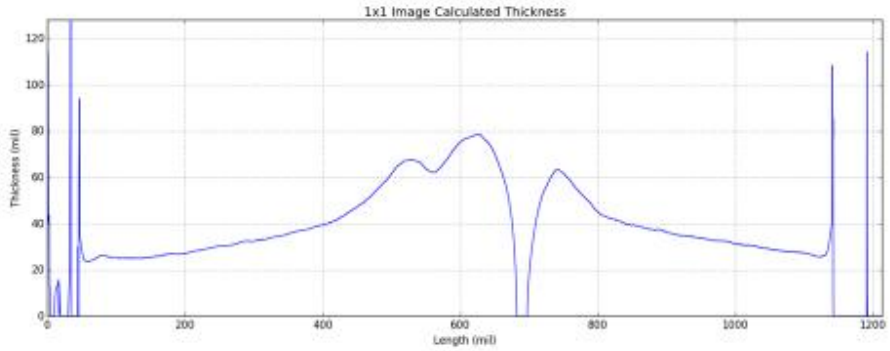
Panel 11 Section 3

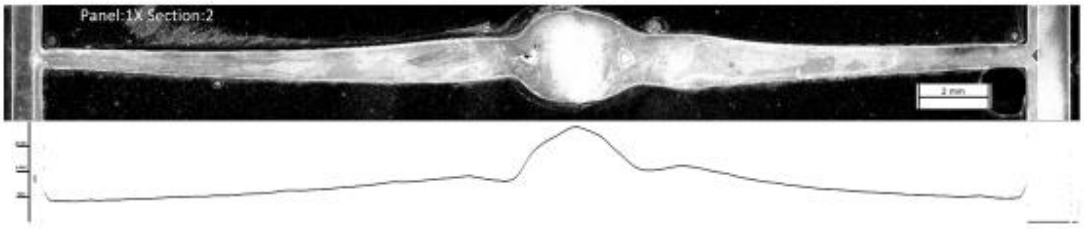


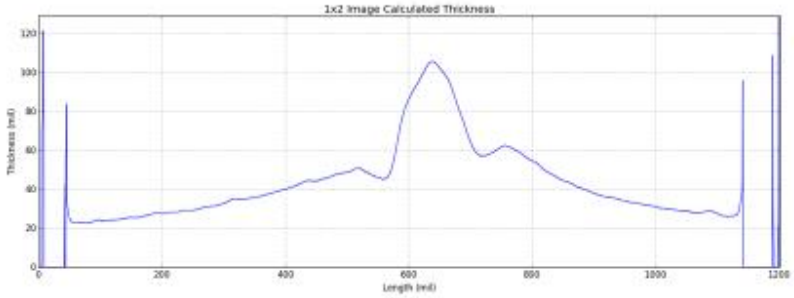
Panel 11 Section 4

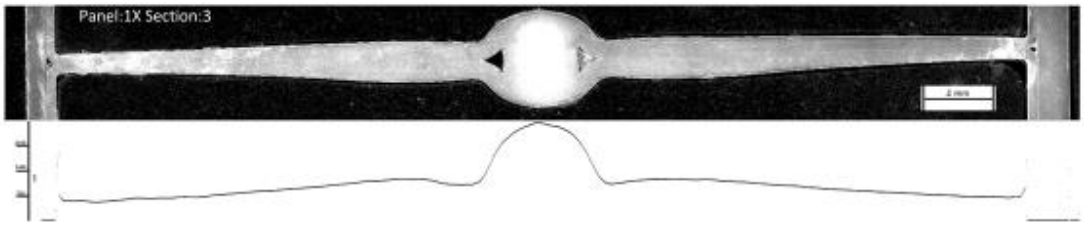


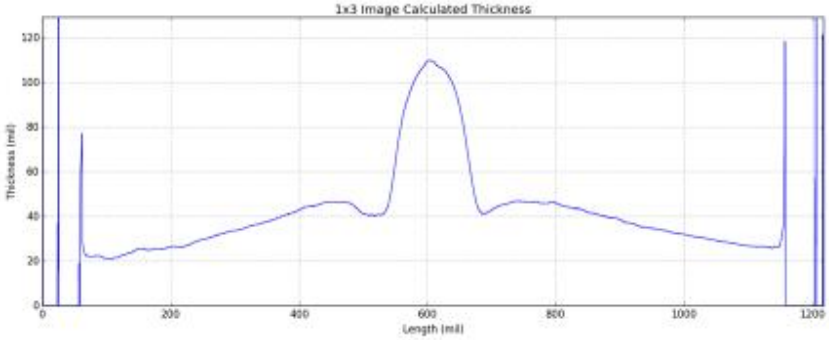


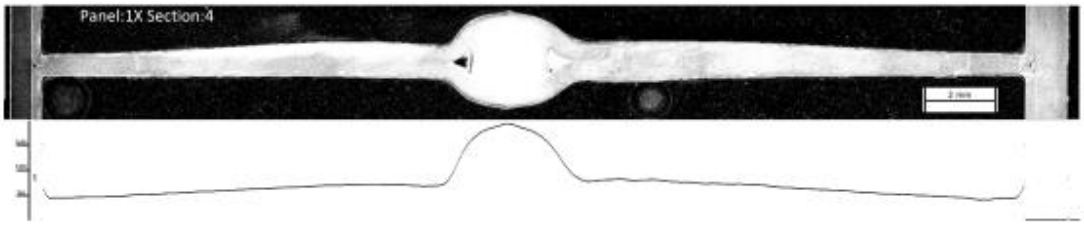


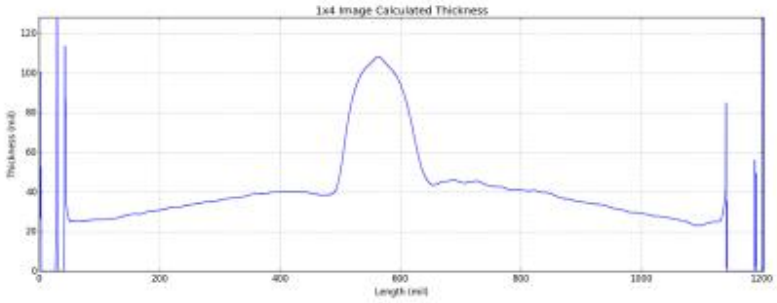

Panel:1X Section:2
2 mm

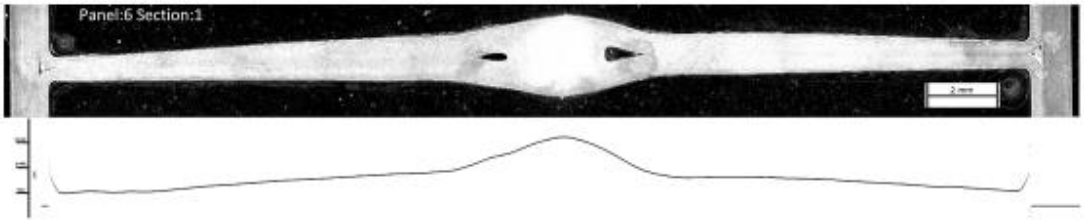

1x2 Image Calculated Thickness
Thickness (mil)
Length (mil)
0
20
40
60
80
100
120
200
400
600
800
1000
1200

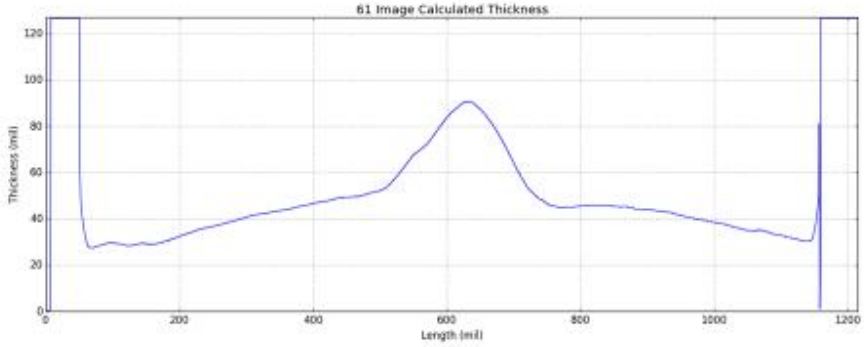

Panel:1X Section:3
2 mm

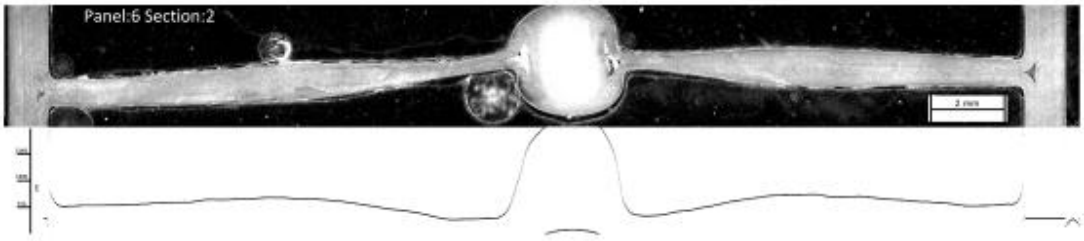

1x3 Image Calculated Thickness
Thickness (mil)
Length (mil)
0
20
40
60
80
100
120
200
400
600
800
1000
1200

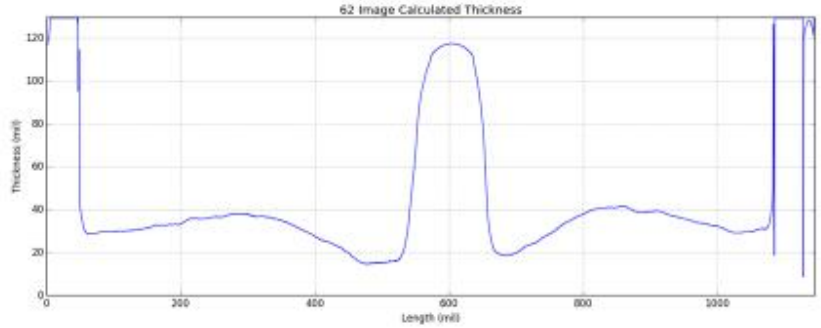

Panel:1X Section:4
2 mm

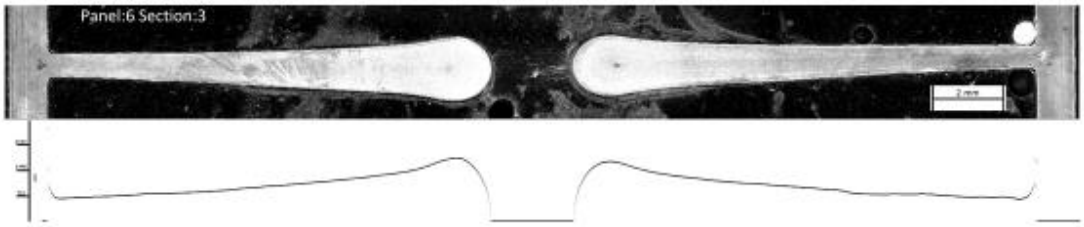

1x4 Image Calculated Thickness
Thickness (mil)
Length (mil)
0
20
40
60
80
100
120
200
400
600
800
1000
1200

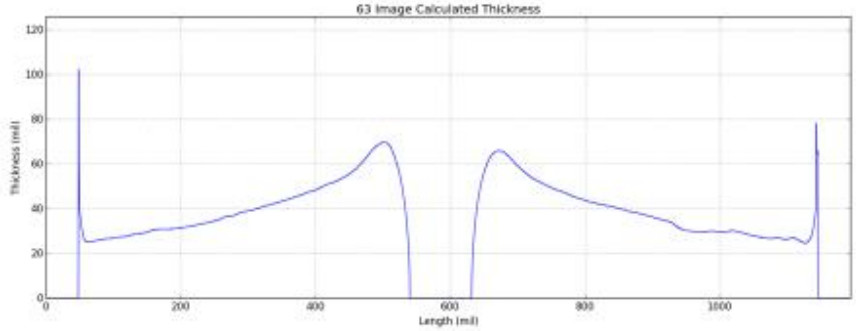

Panel:6 Section:1
2 mm

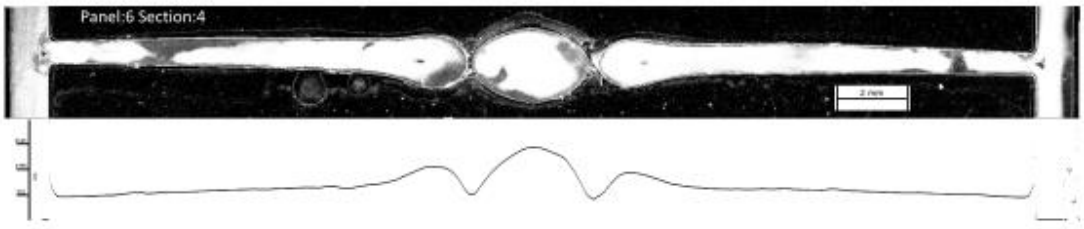

61 Image Calculated Thickness
Thickness (mil)
Length (mil)
0
20
40
60
80
100
120
200
400
600
800
1000
1200

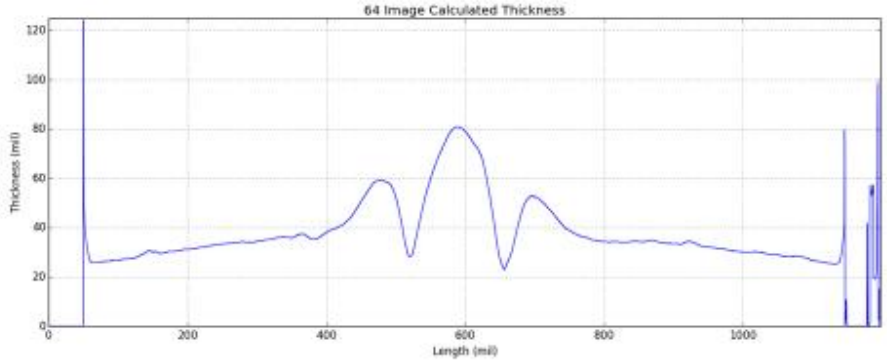

Panel:6 Section:2
2 mm

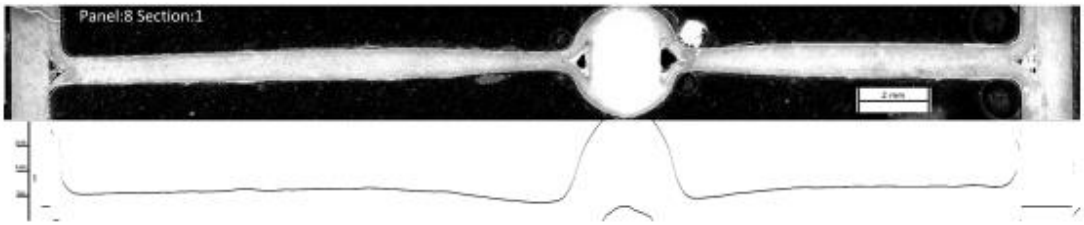

62 Image Calculated Thickness
Thickness (mil)
Length (mil)
0
20
40
60
80
100
120
200
400
600
800
1000

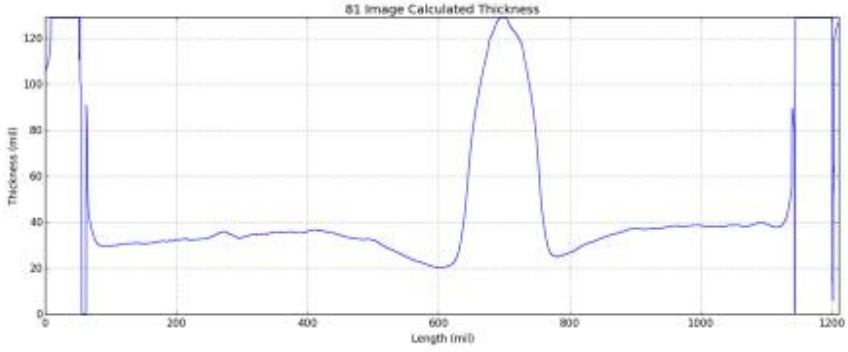

Panel:6 Section:3
2 mm

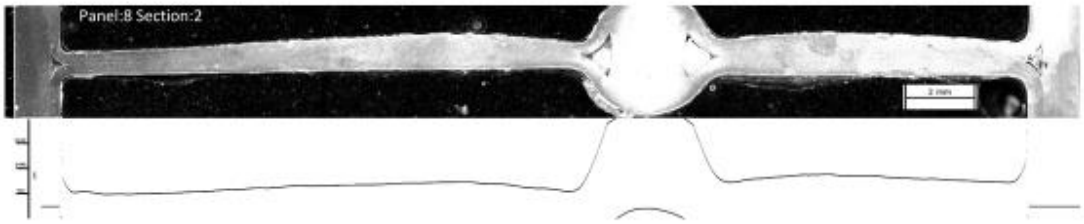

63 Image Calculated Thickness
Thickness (mil)
Length (mil)
0
20
40
60
80
100
120
200
400
600
800
1000

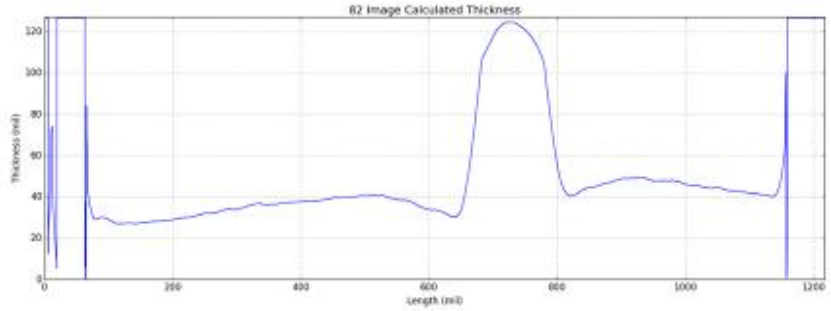

Panel:6 Section:4

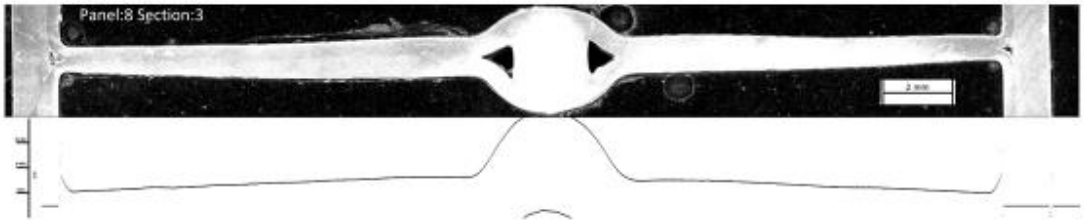

64 Image Calculated Thickness
Thickness (mil)
Length (mil)
0
20
40
60
80
100
120
200
400
600
800
1000

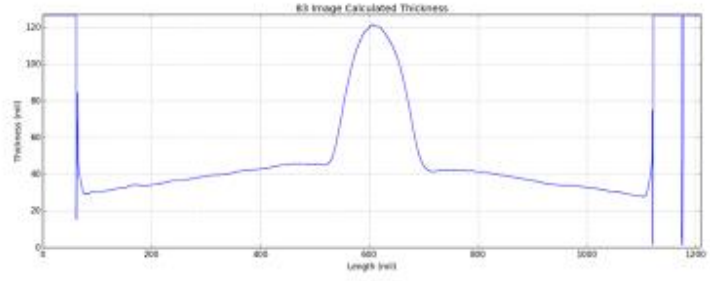

Panel:8 Section:1

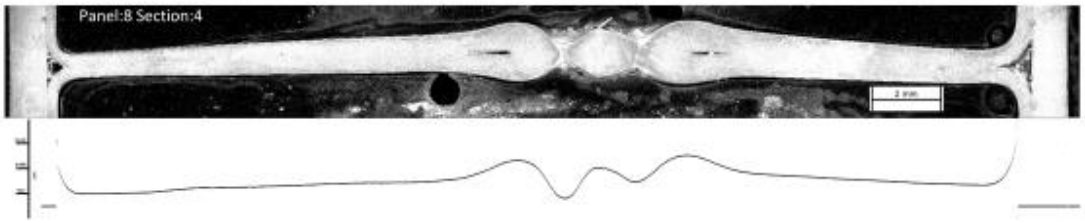

81 Image Calculated Thickness
Thickness (mil)
Length (mil)
0
20
40
60
80
100
120
200
400
600
800
1000
1200

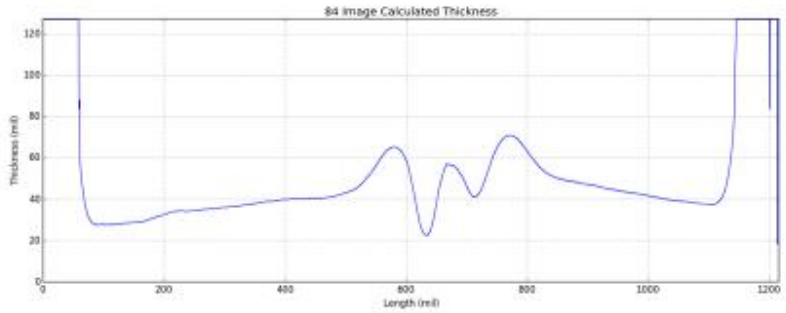

Panel:8 Section:2

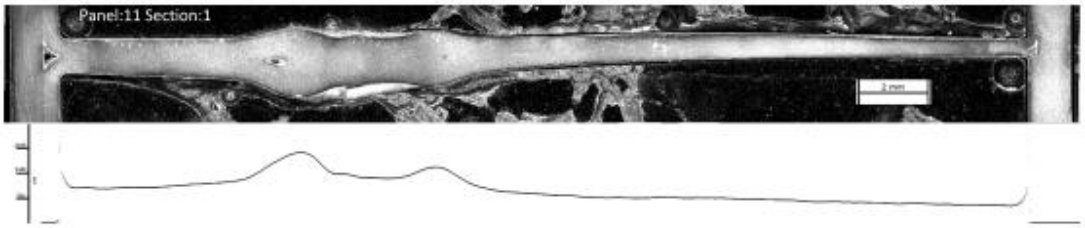

82 Image Calculated Thickness
Thickness (mil)
Length (mil)

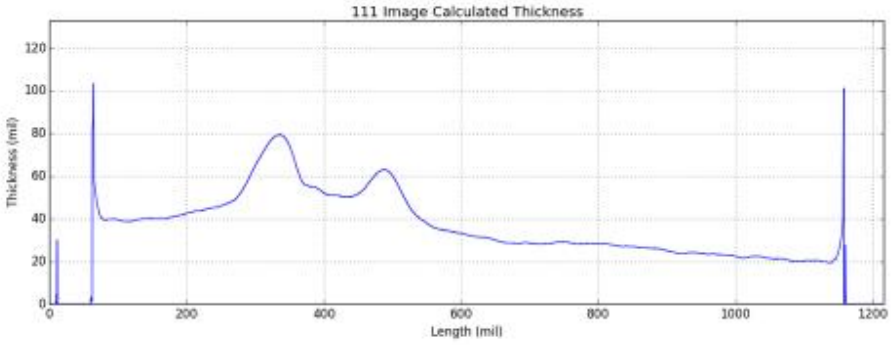

Panel:8 Section:3

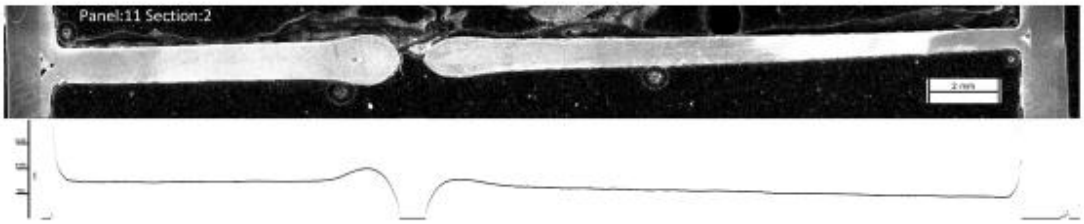

83 Image Calculated Thickness
Thickness (mil)
Length (mil)

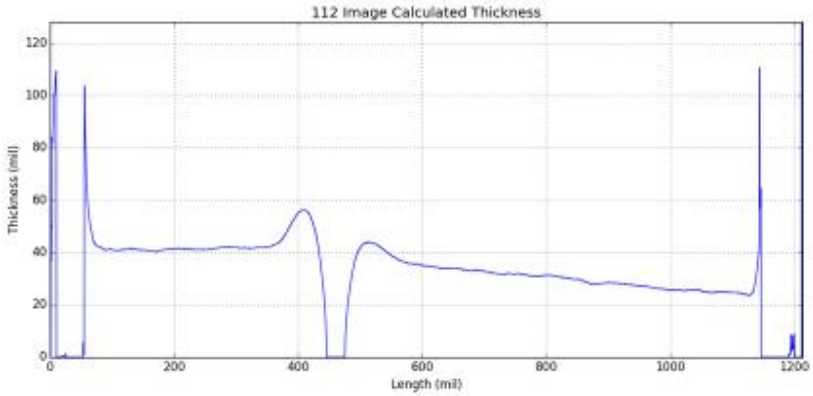

Panel:8 Section:4


84 Image Calculated Thickness
Thickness (mil)
Length (mil)

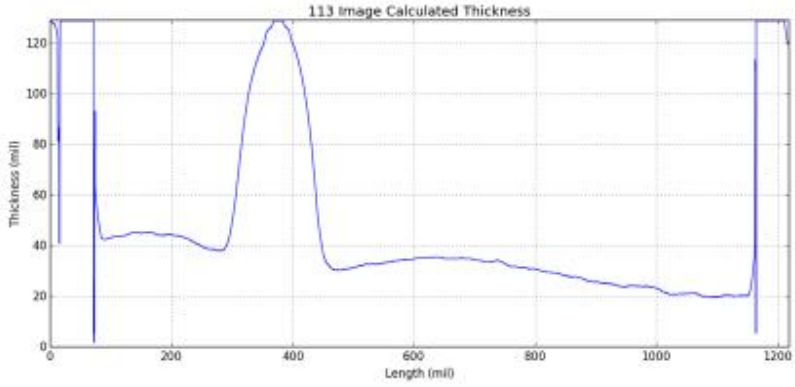

Panel:11 Section:1

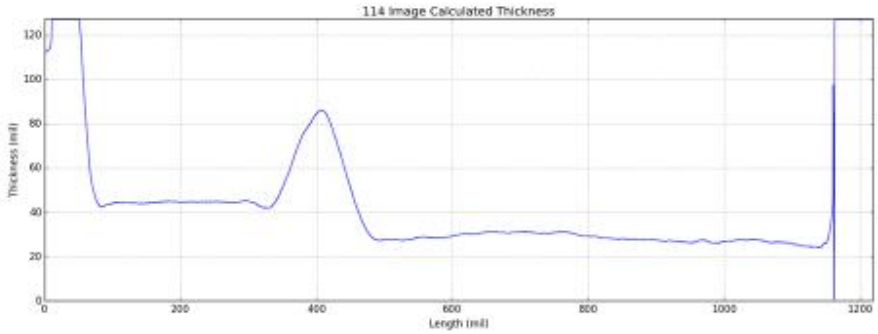

111 Image Calculated Thickness
Thickness (mil)
Length (mil)


Panel:11 Section:2

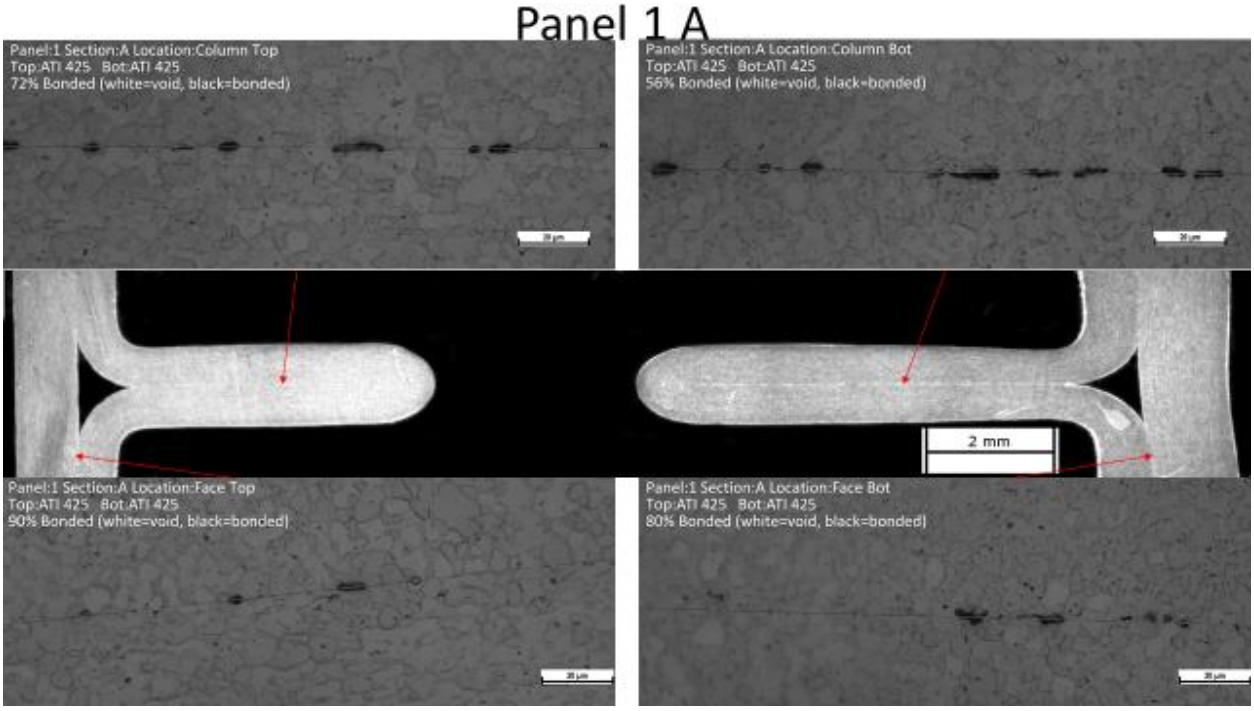

112 Image Calculated Thickness
Thickness (mil)
Length (mil)
0
20
40
60
80
100
120
0
200
400
600
800
1000
1200


Panel:11 Section:3

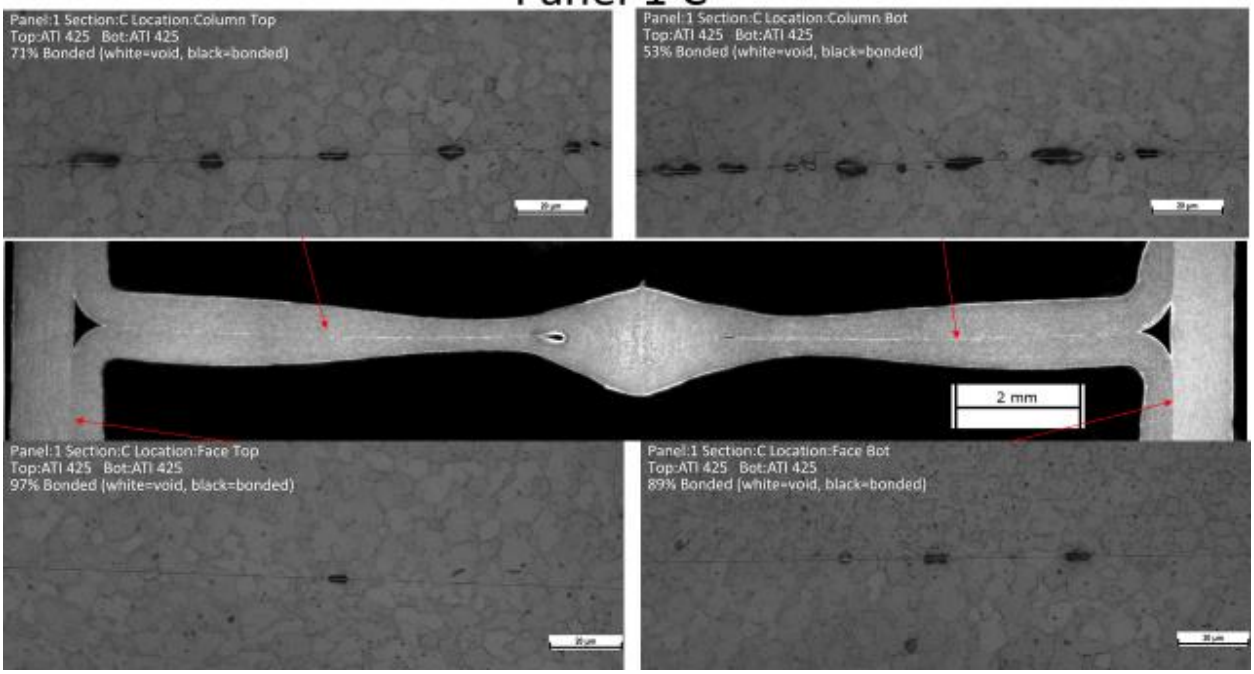

113 Image Calculated Thickness
Thickness (mil)
Length (mil)
0
20
40
60
80
100
120
0
200
400
600
800
1000
1200



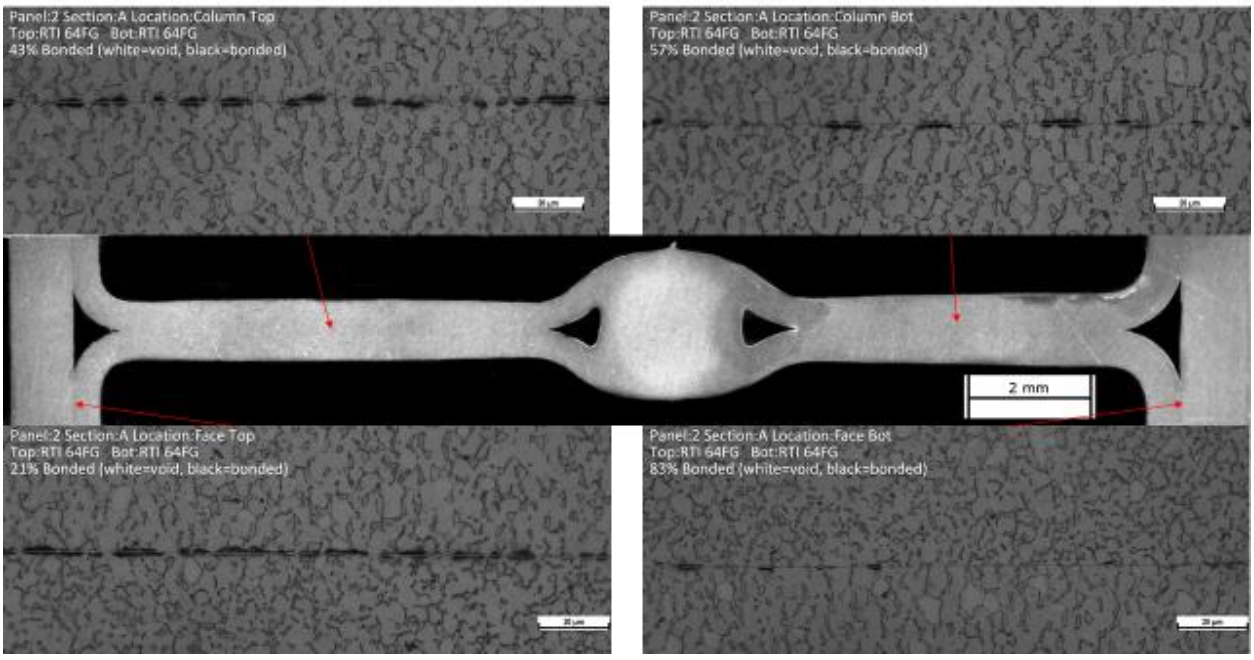
BA18 Panel Group 2



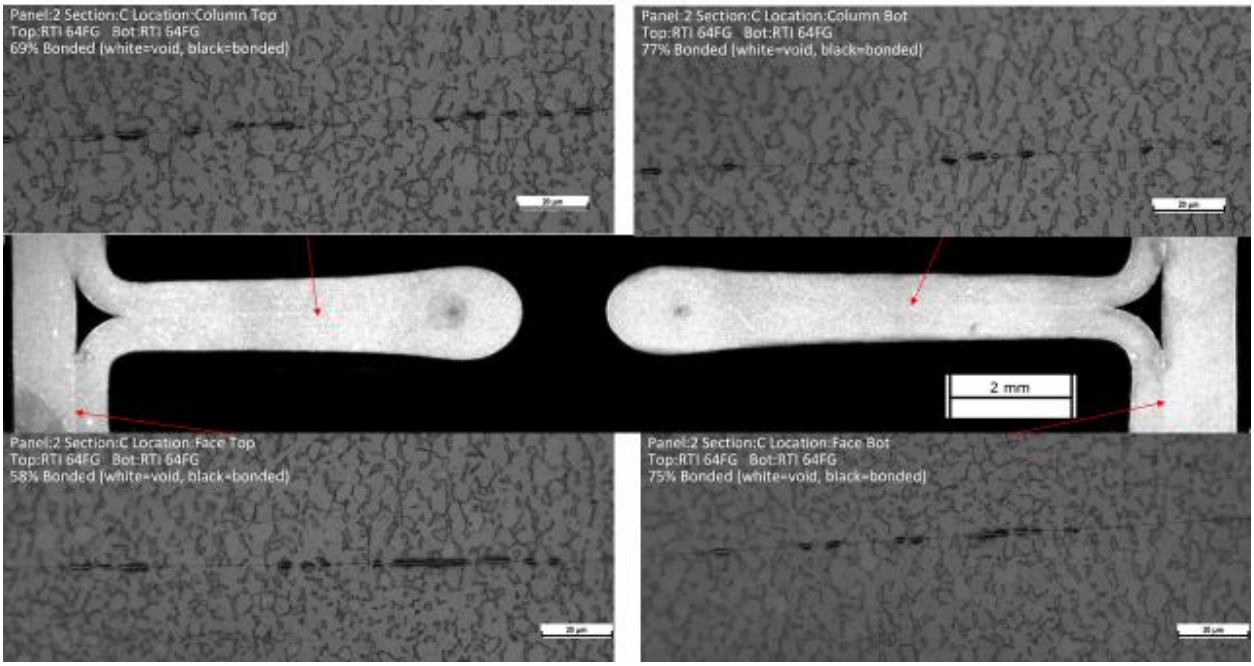
Panel 1 C



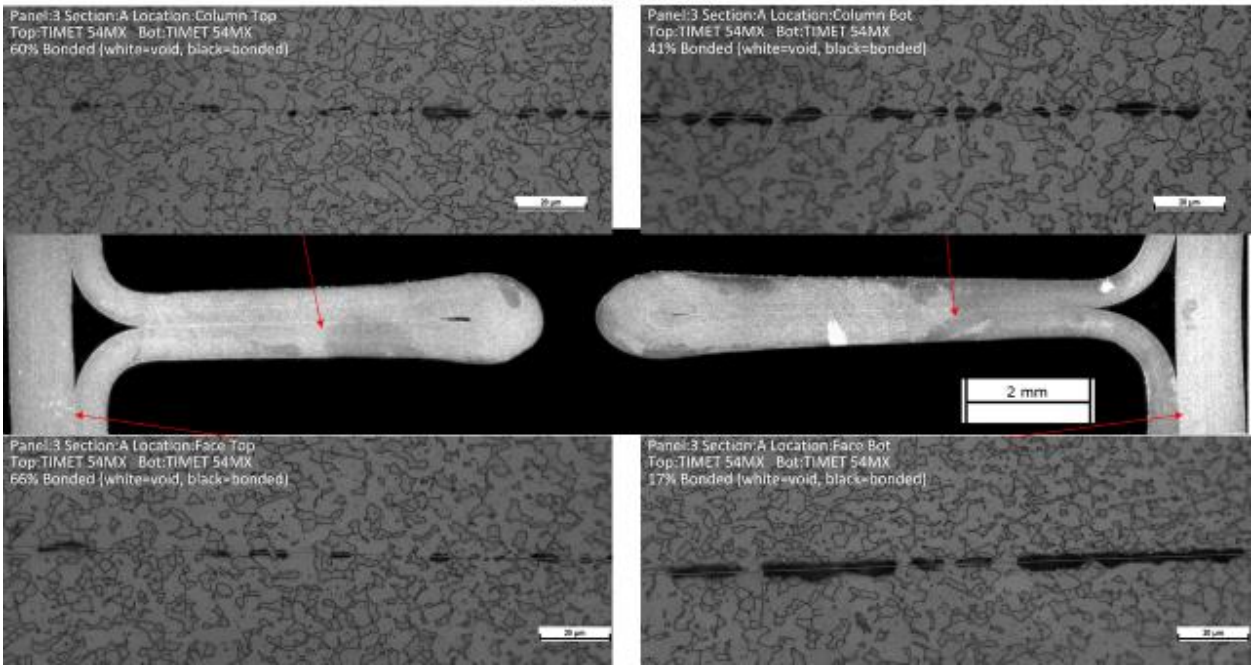
Panel 2 A



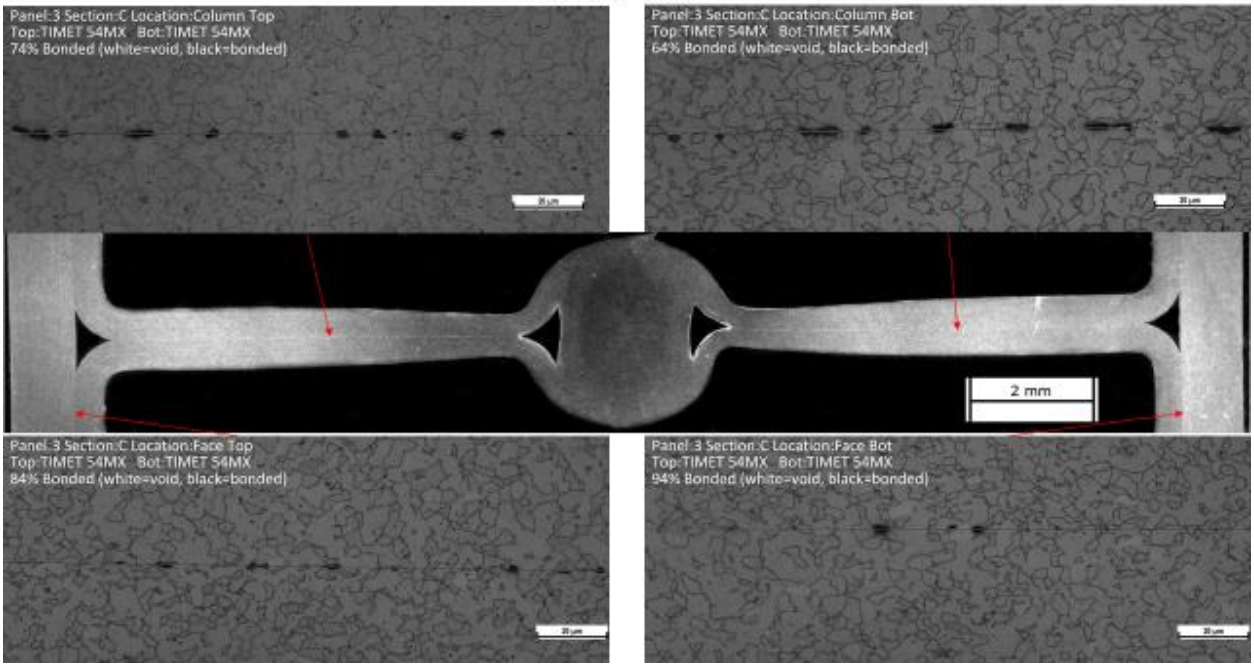
Panel 2 C



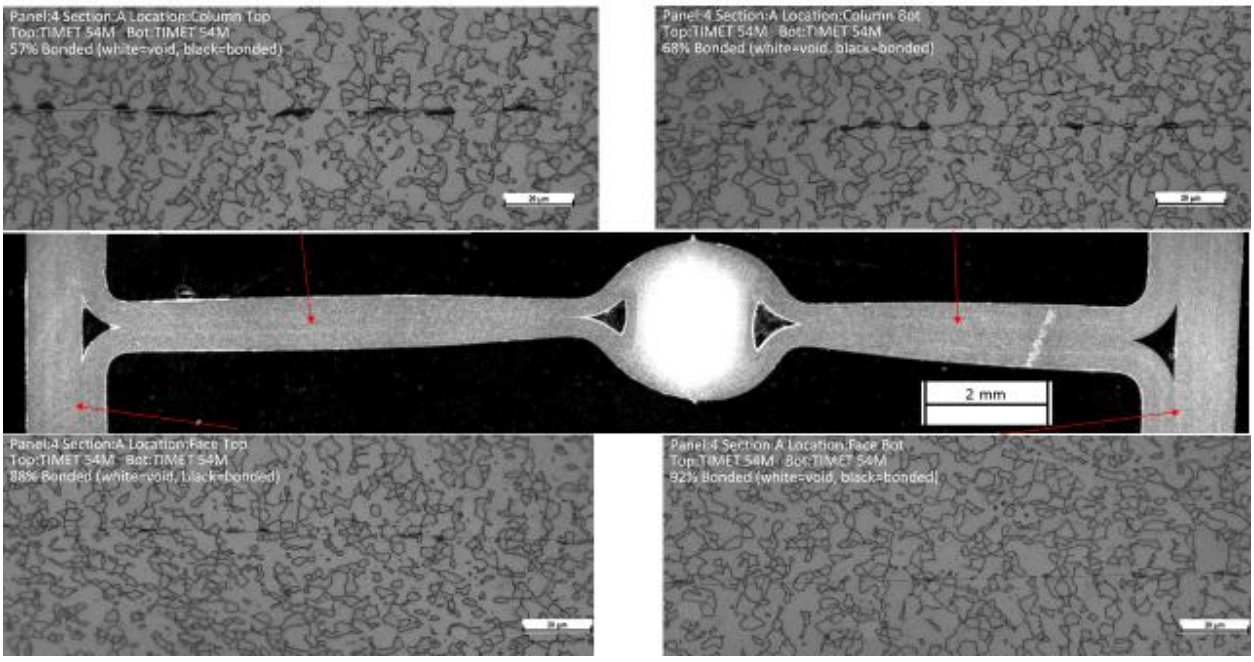
Panel 3 A



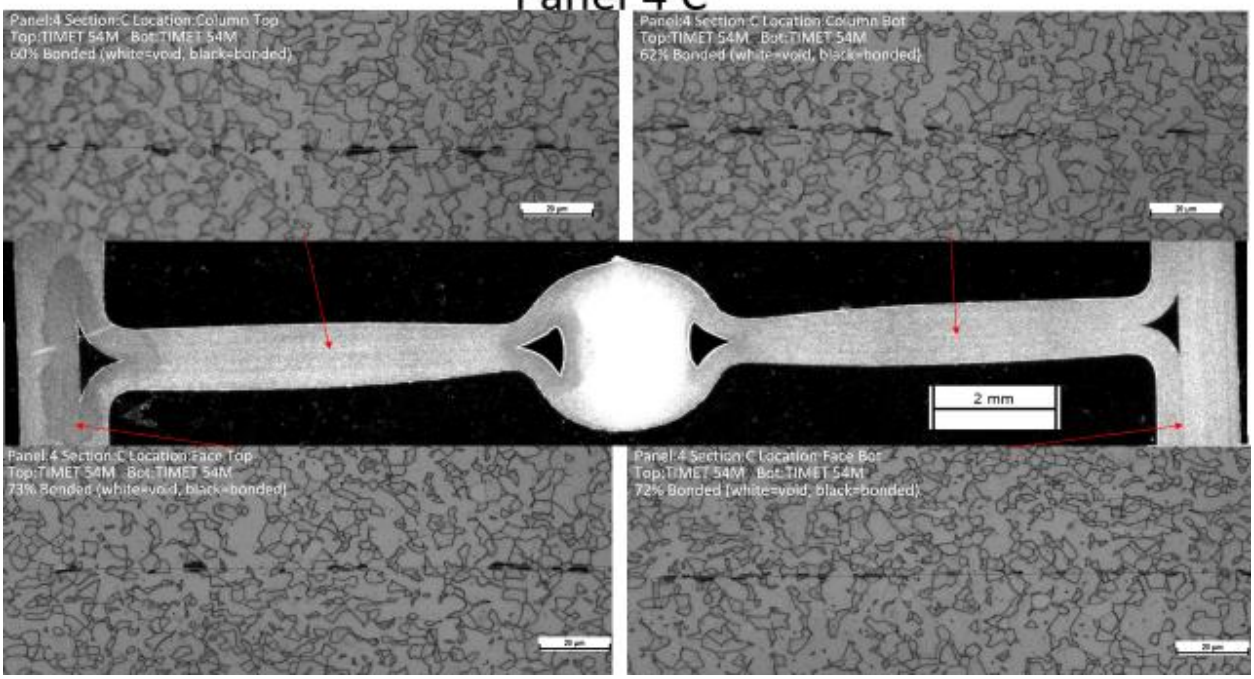
Panel 3 C



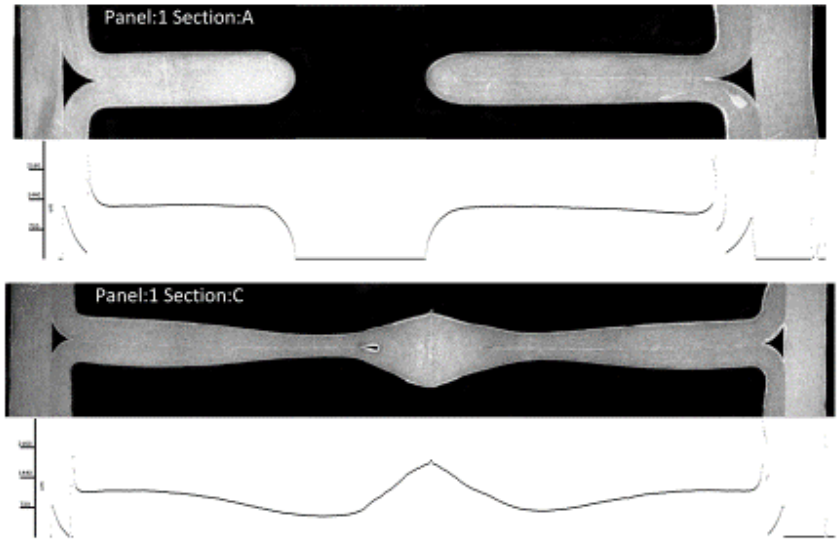
Panel 4 A



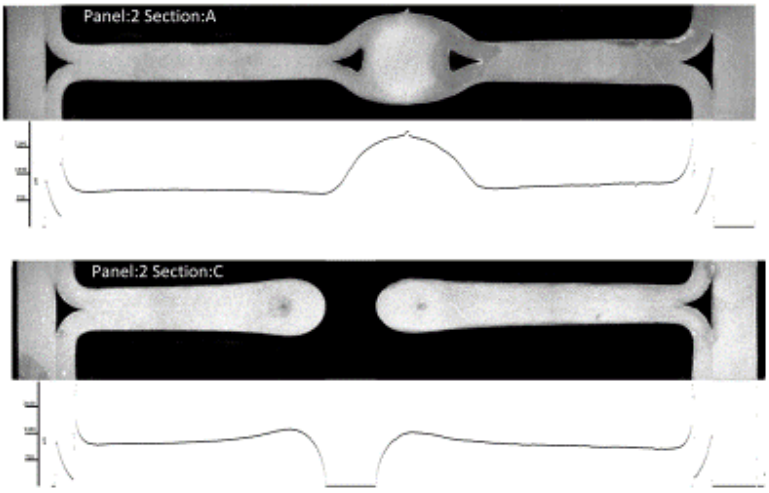
Panel 4 C



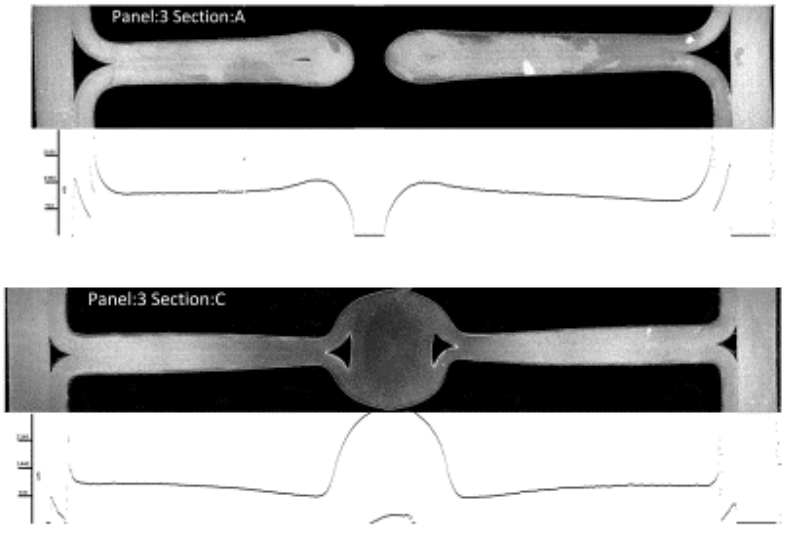
Thickness Profiles Panel 1



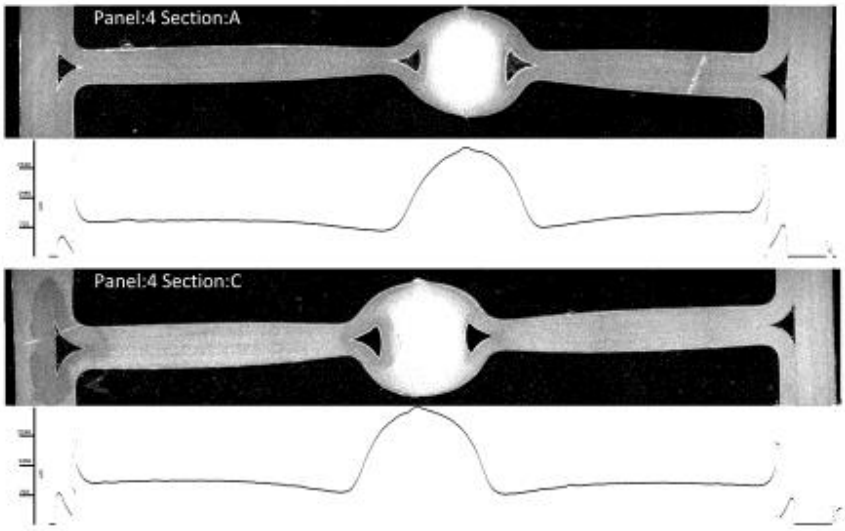
Thickness Profiles Panel 2



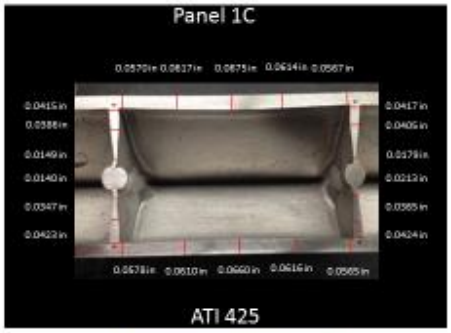
Thickness Profiles Panel 3



Thickness Profiles Panel 4



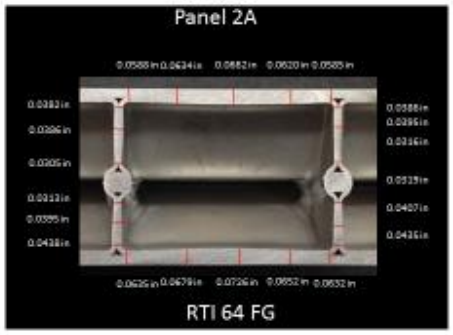
Cross Sections Panel 1

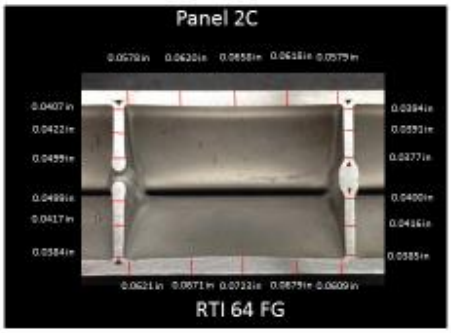


Note: on panel 1A there was not enough material and the bottom was measured on the slanted portion of the trimming. The measurements are larger than the internal panel would be.

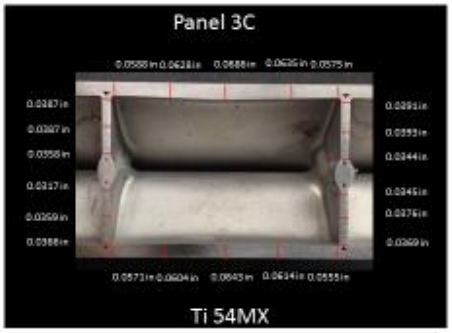
measurement location

Cross Sections Panel 2

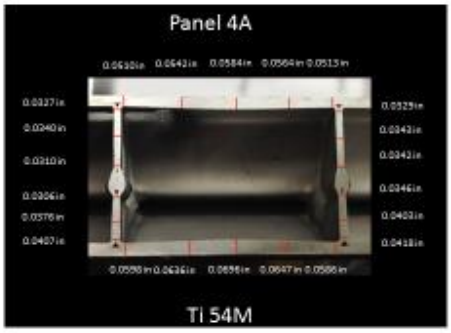


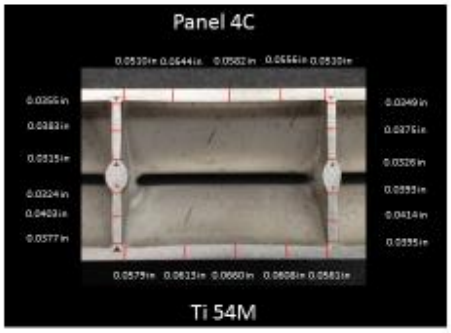


Cross Sections Panel 3



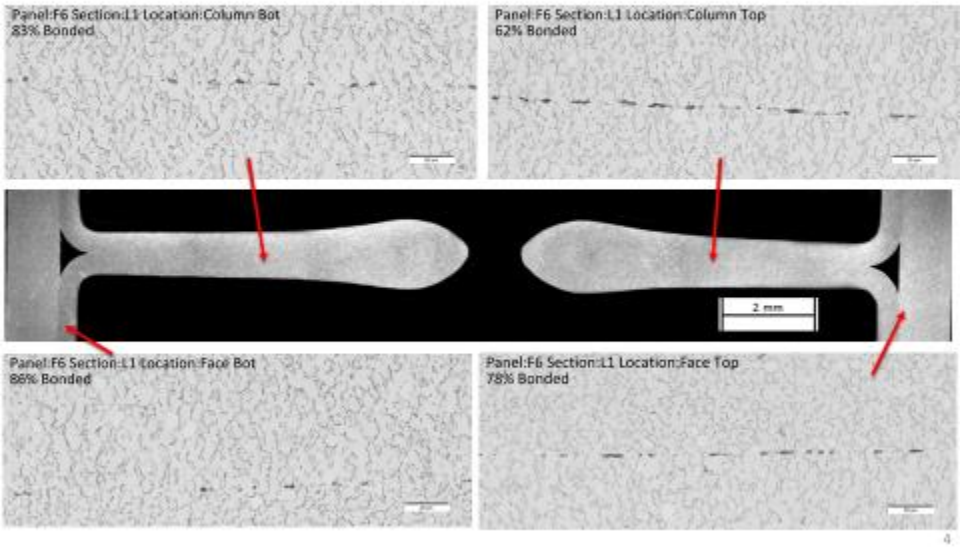
Cross Sections Panel 4



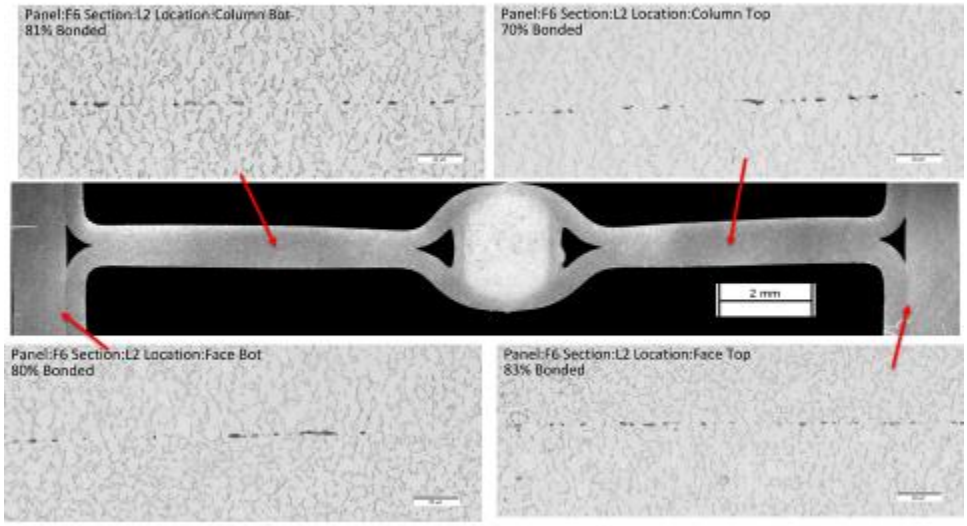


BA18 Panel Group 3

F6 Long Cell Hole (F6L1)

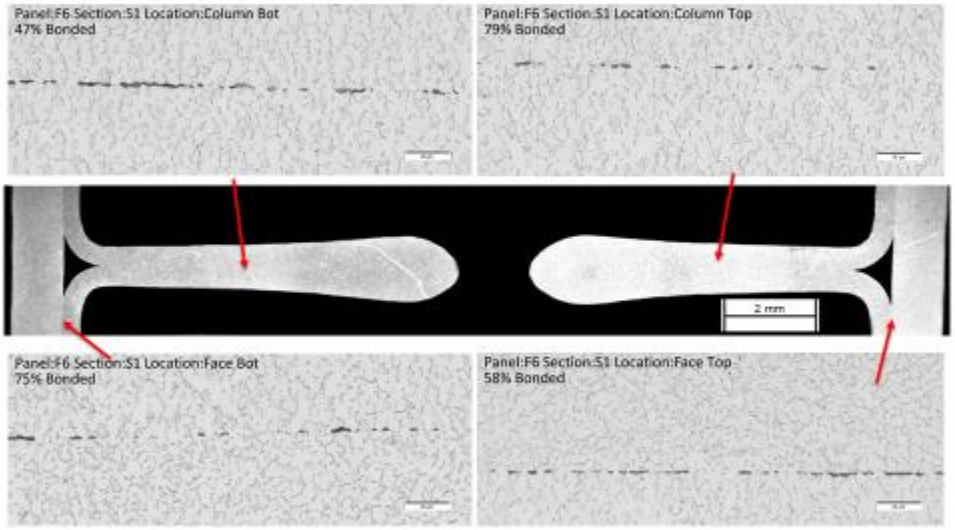


F6 Long Cell Weld Nugget F6L2)



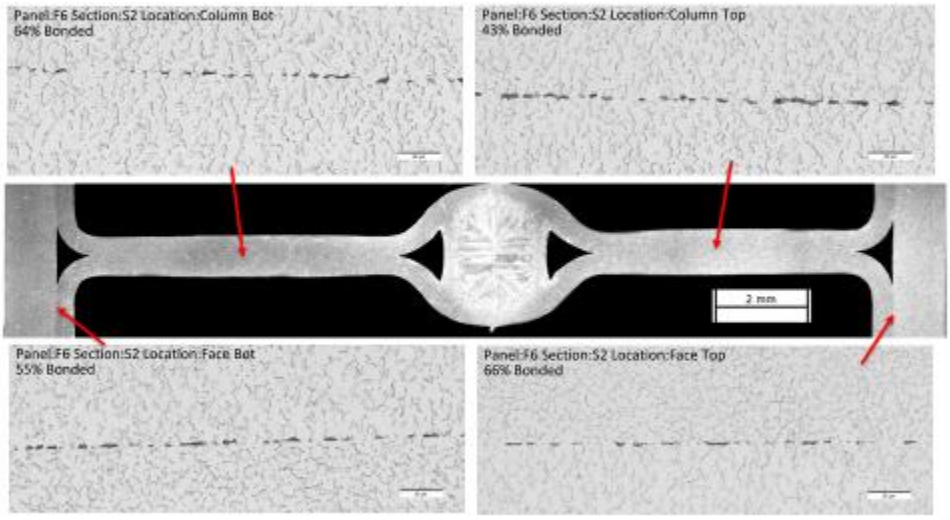
5

F6 Short Cell Hole (F6S1)



6

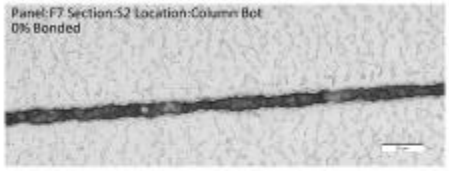
F6 Short Cell Weld Nugget (F6S2)



7

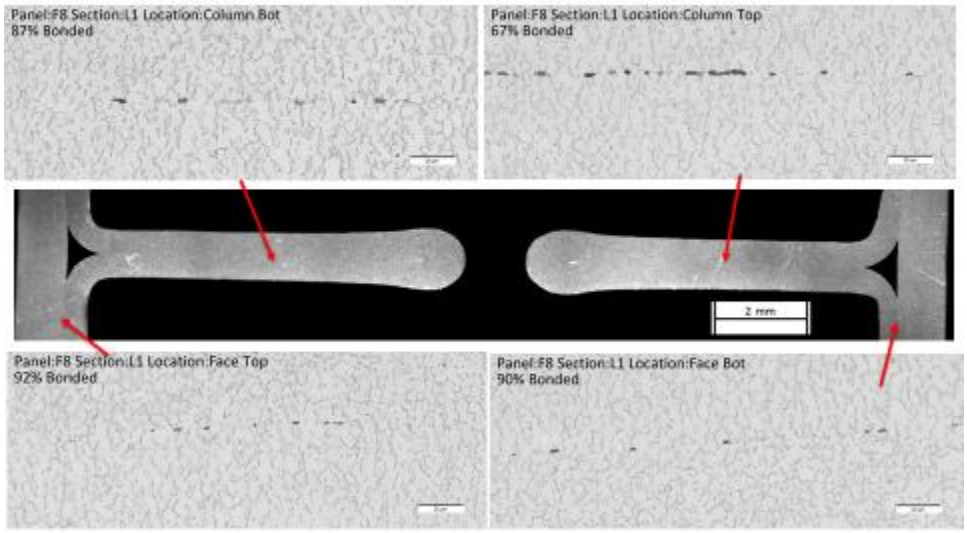
F7

Only the weld nugget of F7 was bonded. Almost all of the face sheets fell off after waterjet cutting and the columns could easily pull apart.



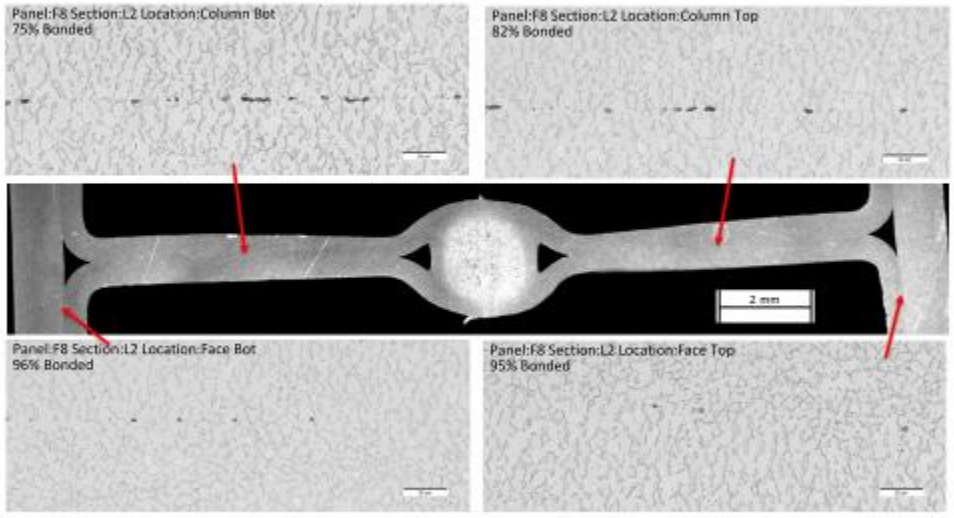
8

F8 Long Cell Hole (F8L1)



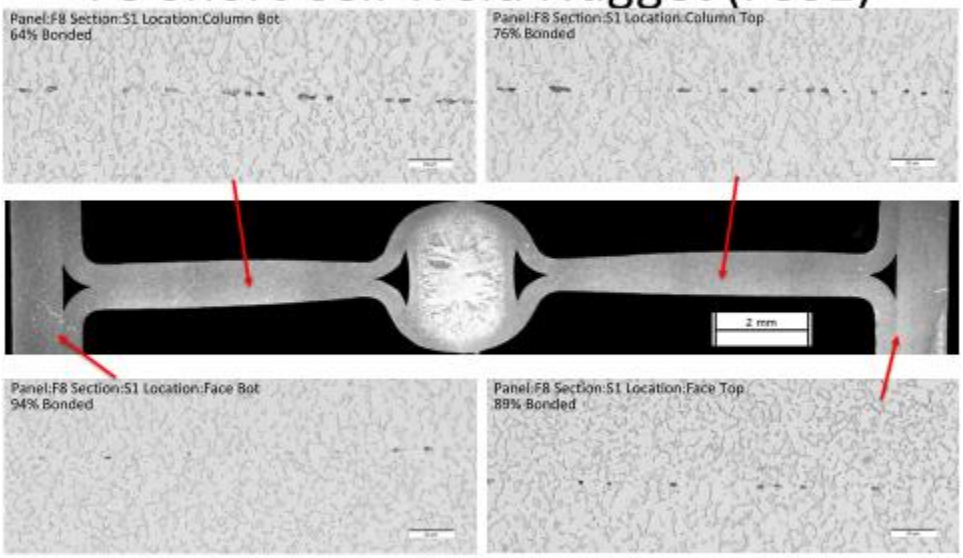
9

F8 Long Cell Weld Nugget (F8L2)



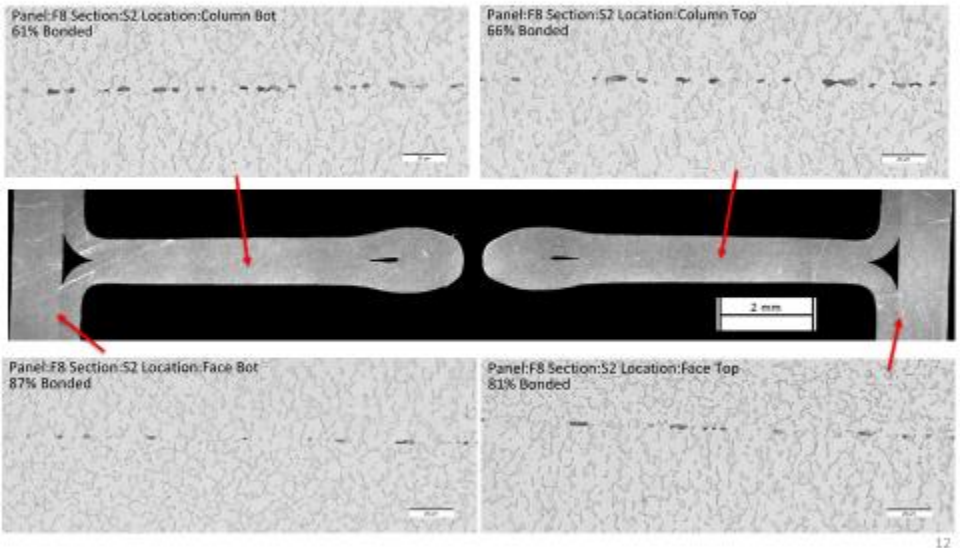
10

F8 Short Cell Weld Nugget (F8S1)



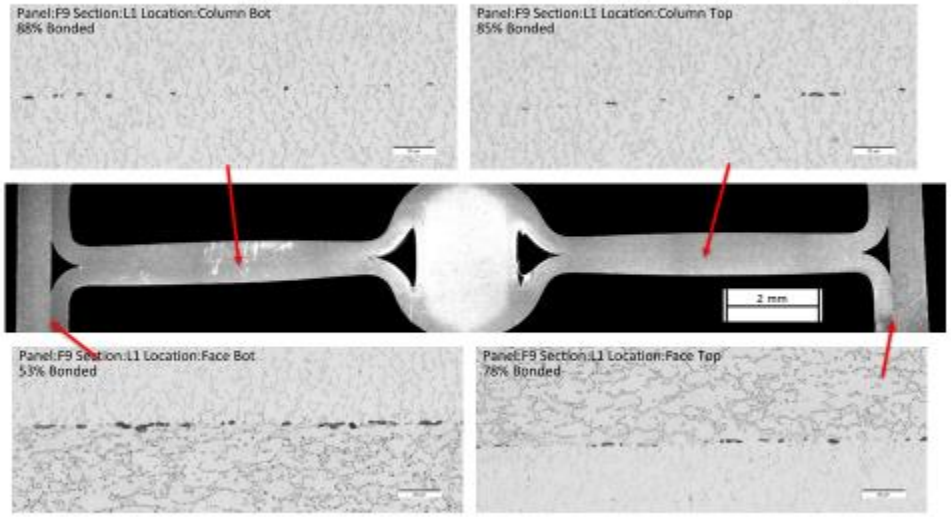
11

F8 Short Cell Hole (F8S2)



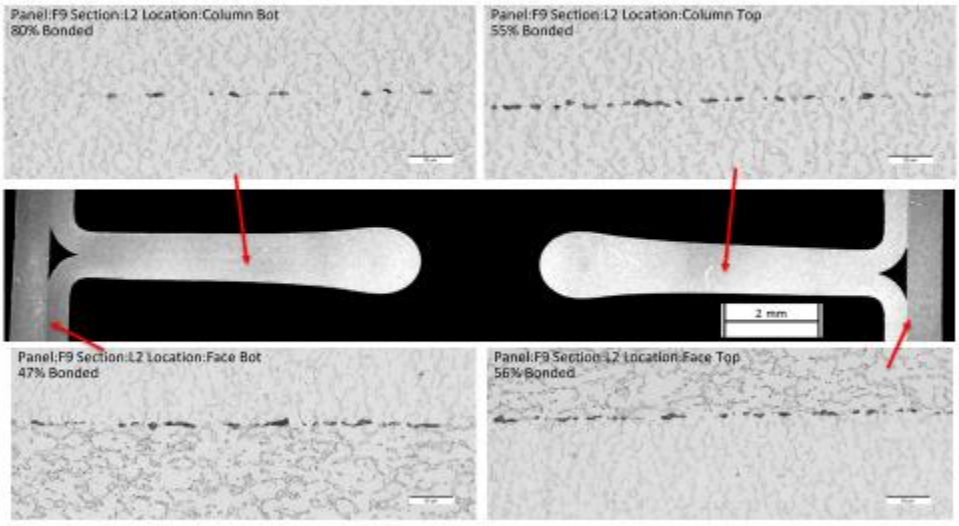
12

F9 Long Cell Weld Nugget (F9L1)



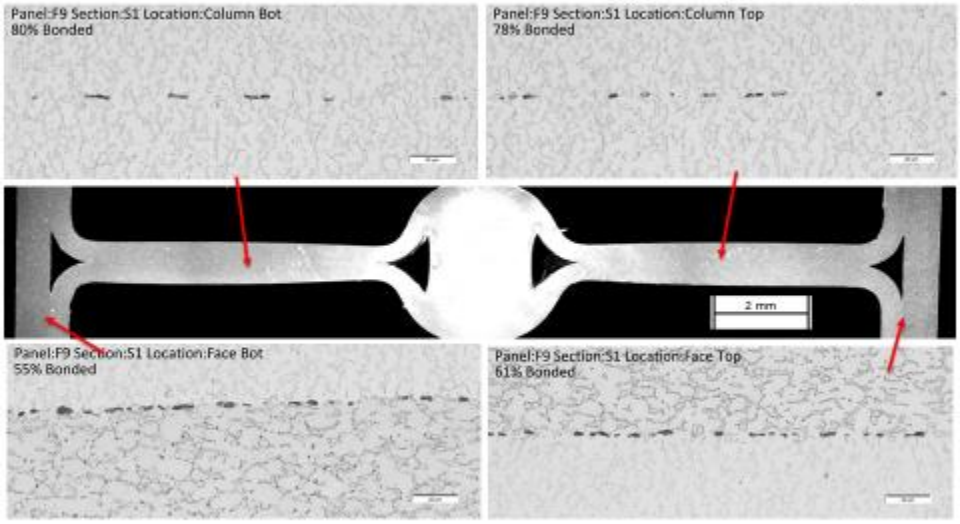
13

F9 Long Cell Hole (F9L2)



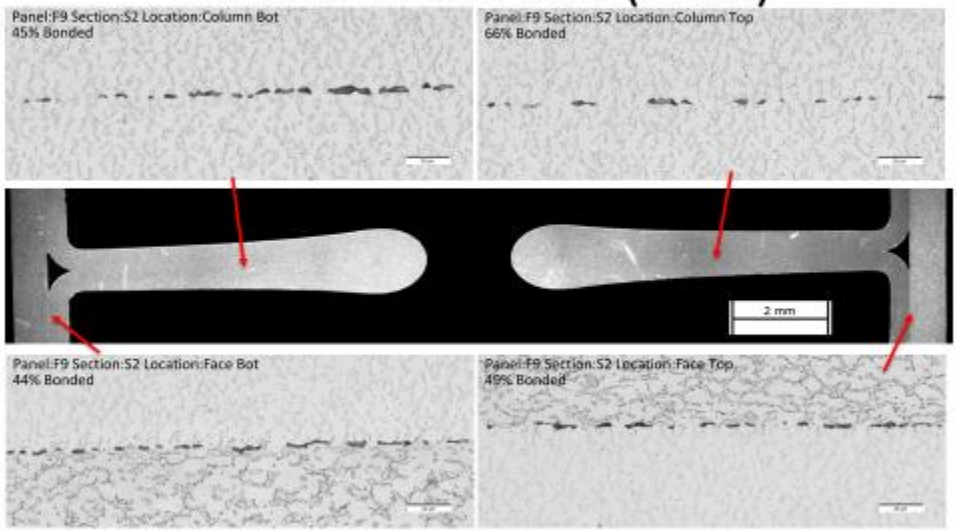
14

F9 Short Cell Weld Nugget (F9S1)



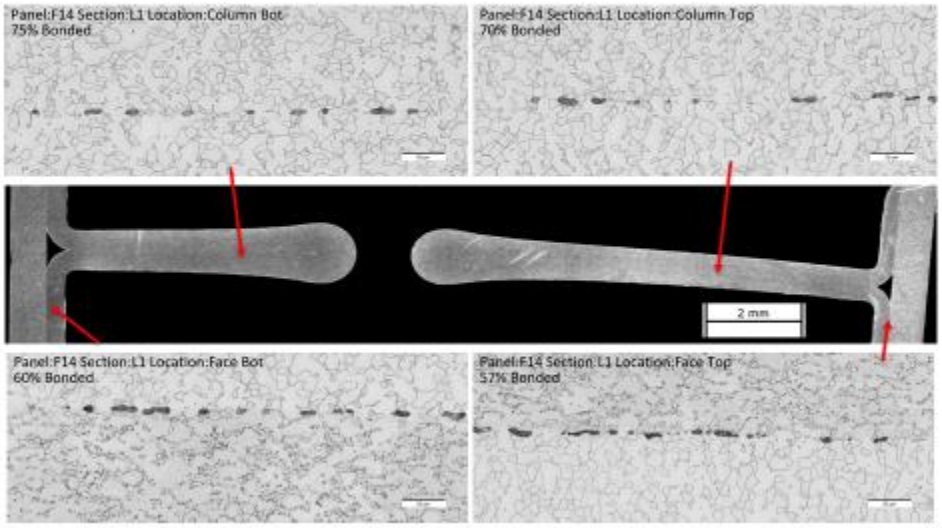
15

F9 Short Cell Hole (F9S2)



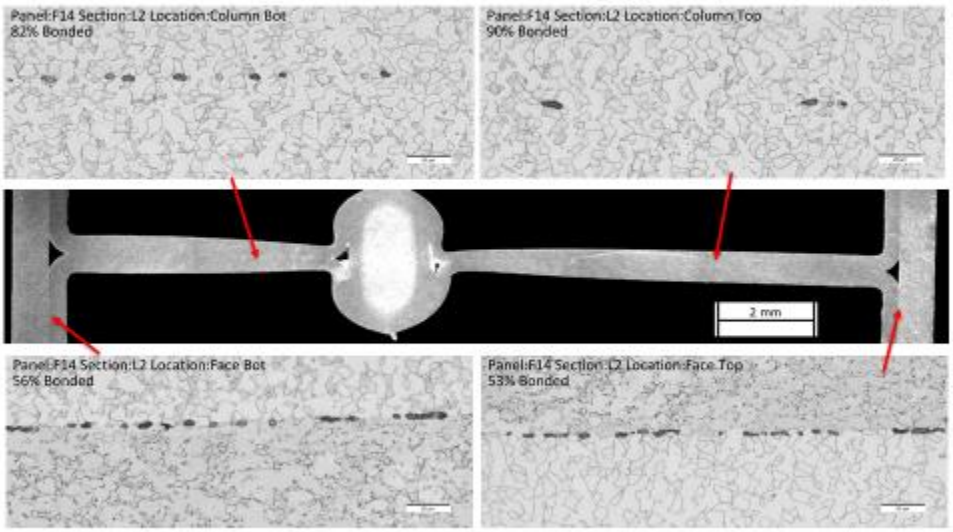
16

F14 Long Cell Hole (F14L1)



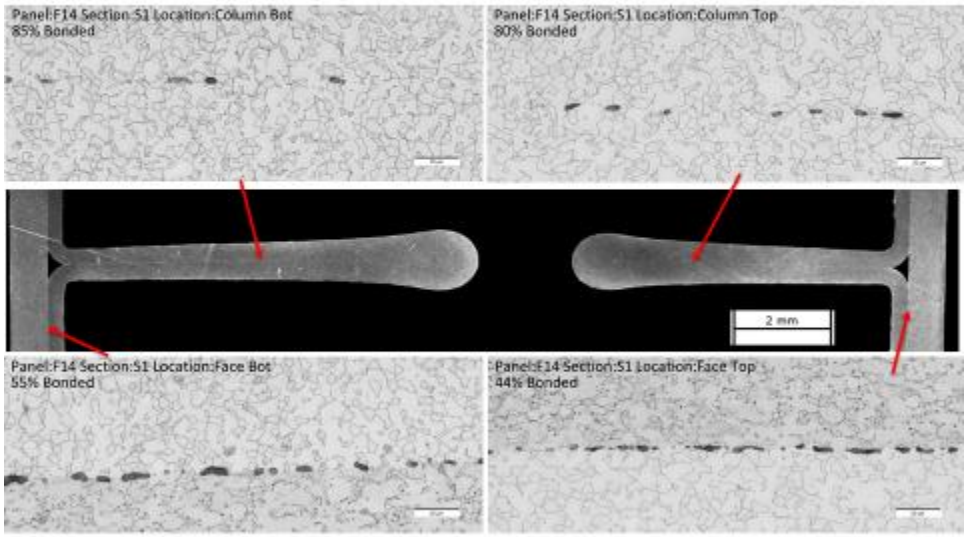
17

F14 Long Cell Weld Nugget (F14L2)



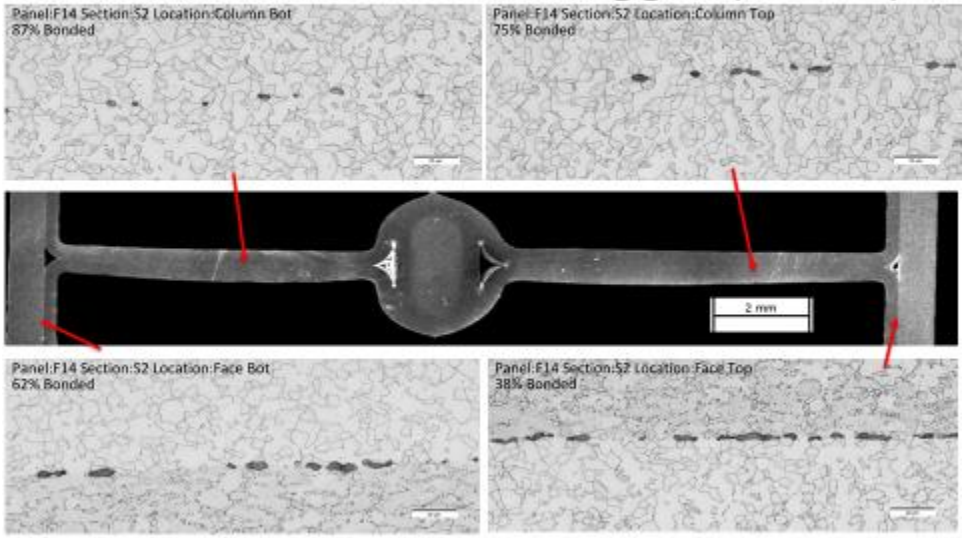
18

F14 Short Cell Hole (F14S1)



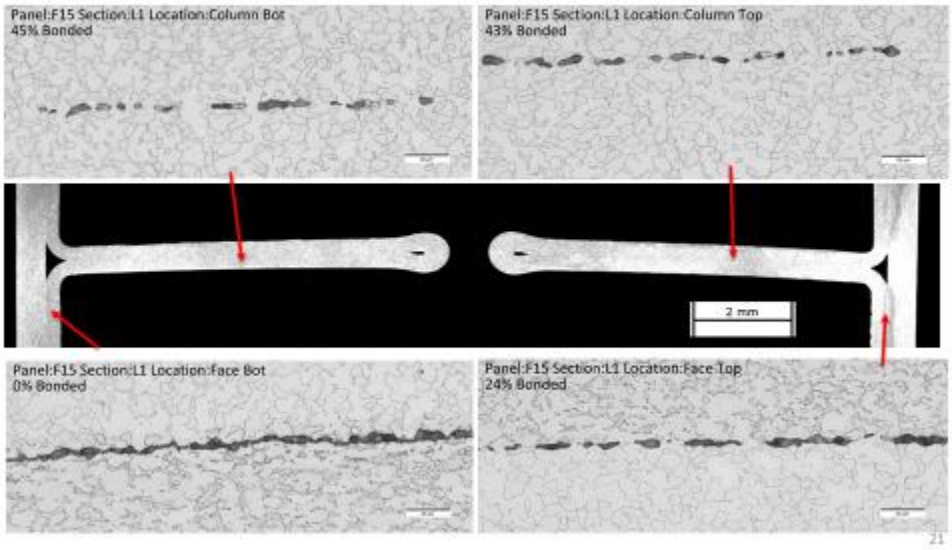
19

F14 Short Cell Weld Nugget (F14S2)



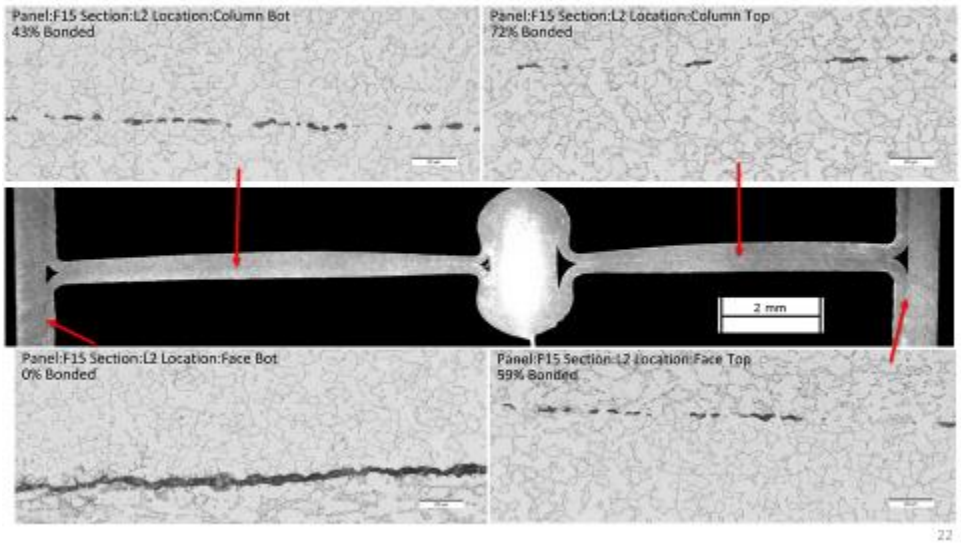
20

F15 Long Cell Hole (F15L1)



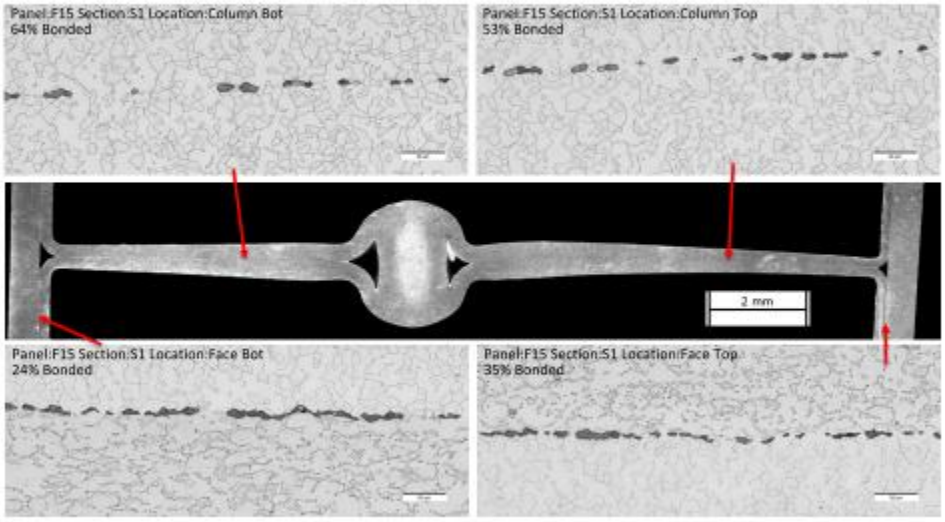
21

F15 Long Cell Weld Nugget (F15L2)



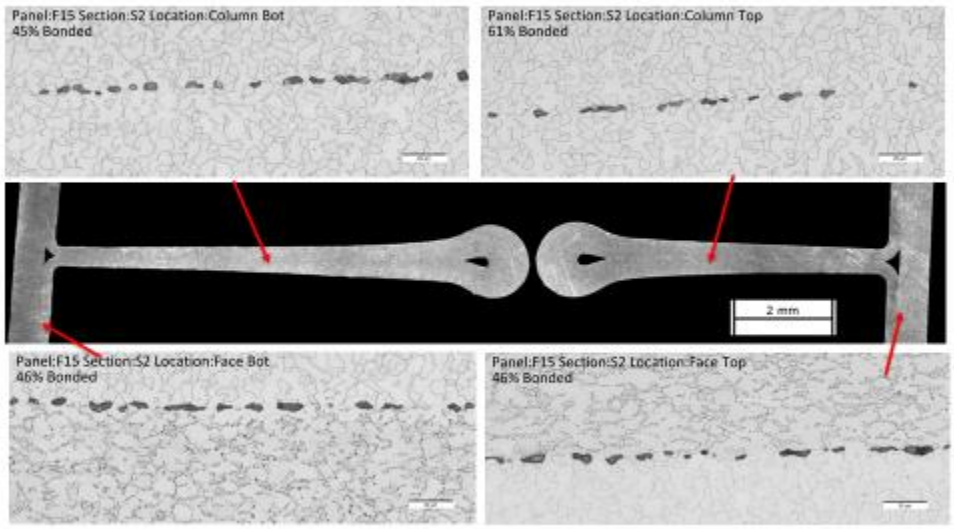
22

F15 Short Cell Hole (F15S1)



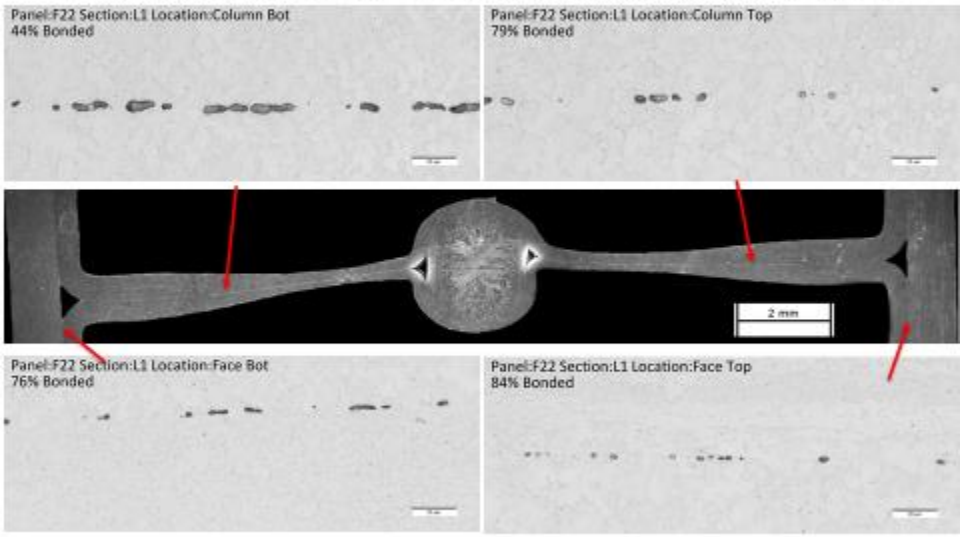
23

F15 Short Cell Weld Nugget (F15S2)



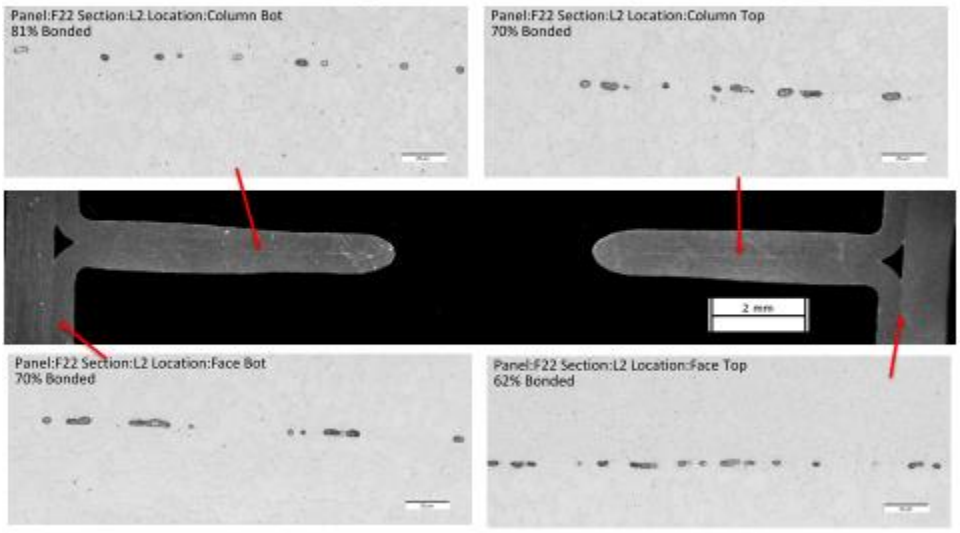
24

F22 Long Cell Weld Nugget (F22L1)



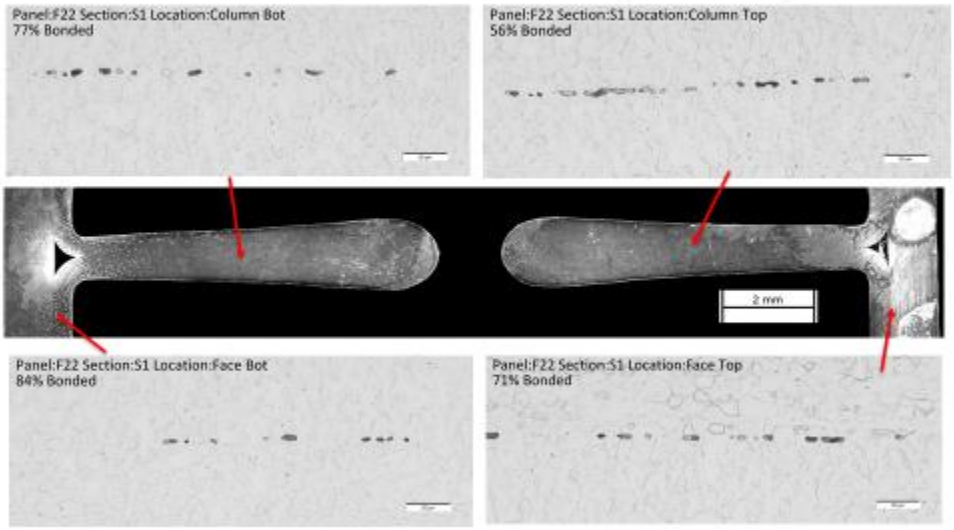
25

F22 Long Cell Hole (F22L2)



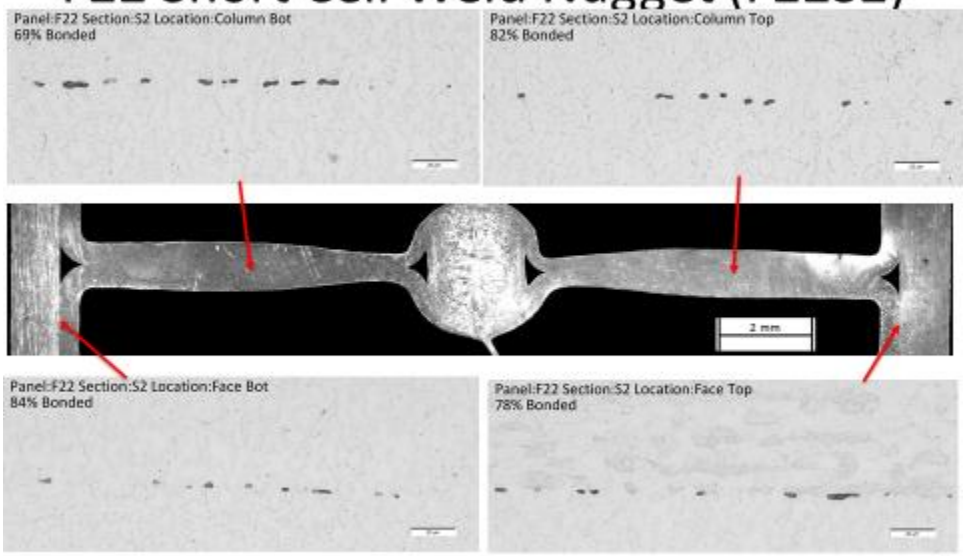
26

F22 Short Cell Hole (F22S1)



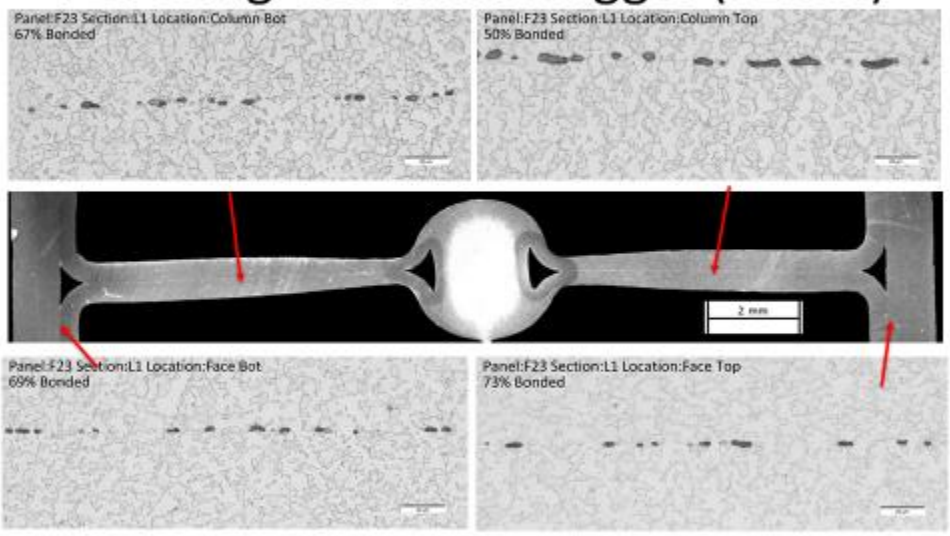
27

F22 Short Cell Weld Nugget (F22S2)



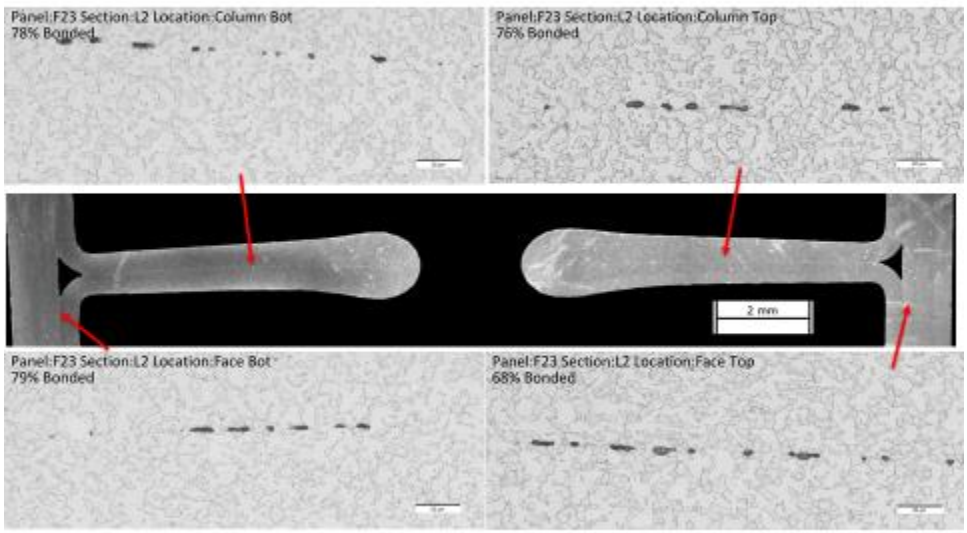
28

F23 Long Cell Weld Nugget (F23L1)



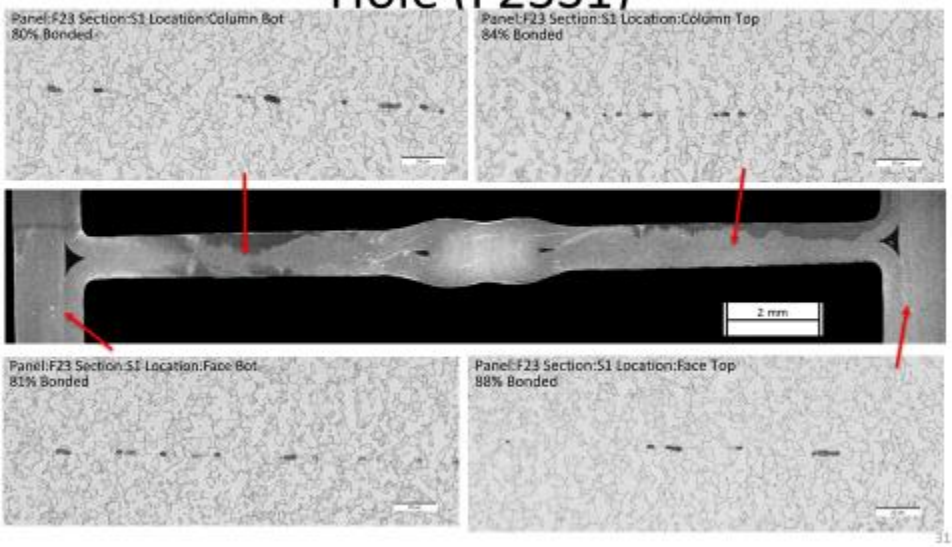
29

F23 Long Cell Hole (F23L2)



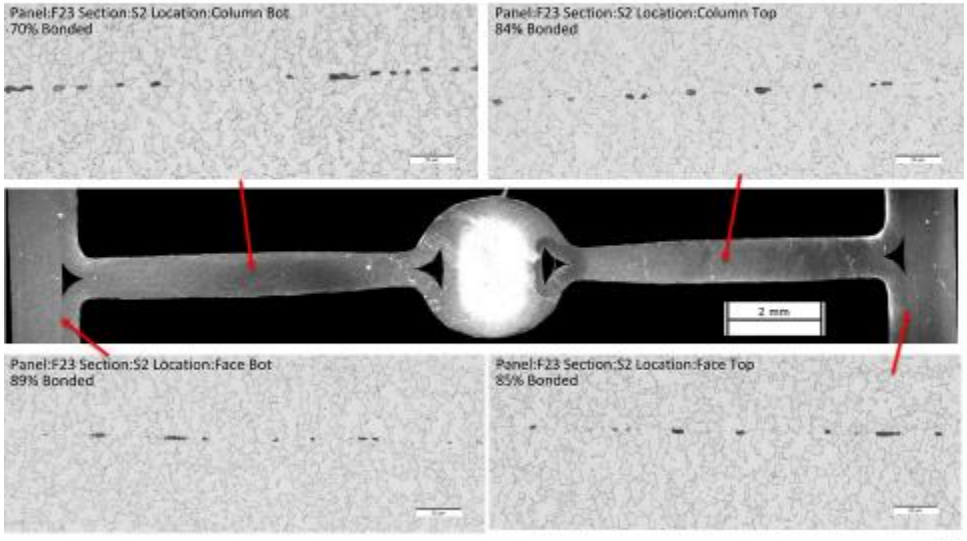
30

F23 Short Cell Between Nugget and Hole (F23S1)



31

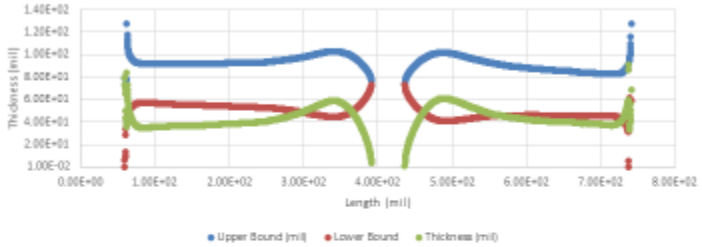
F23 Short Cell Weld Nugget (F23S2)



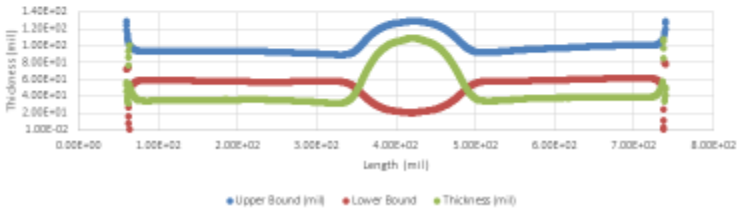
32

Thickness Profiles F6 Long Cell

F6L1



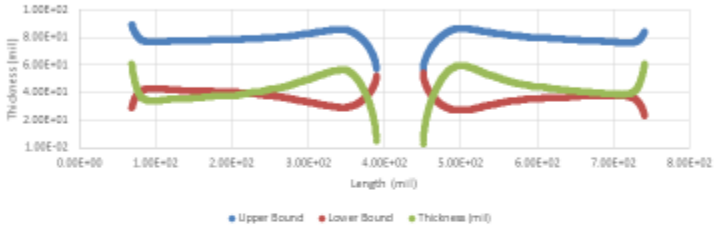
F6L2



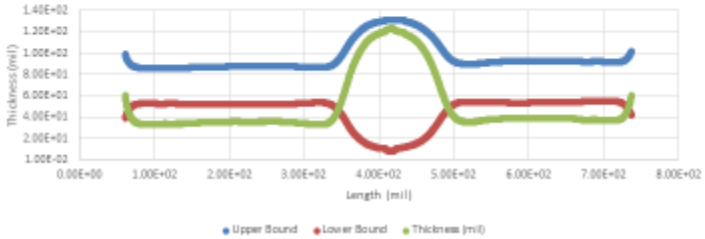
36

Thickness Profiles F6 Short Cell

F6S1



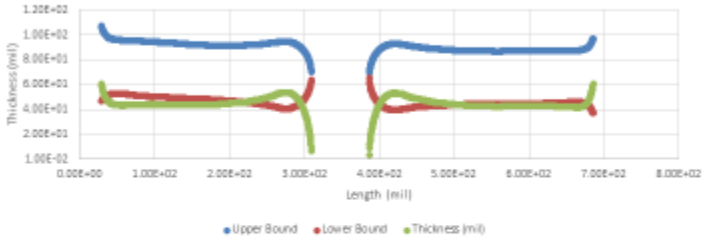
F6S2



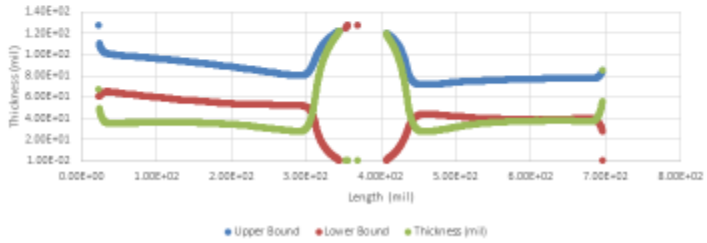
37

Thickness Profiles F7 Long Cell

F7L1



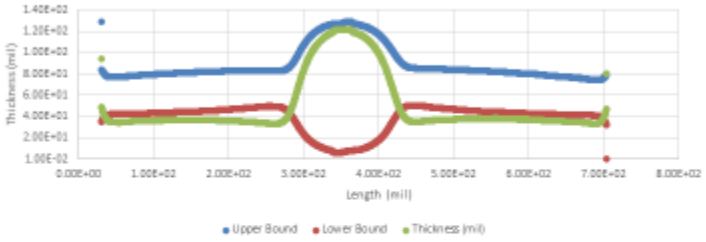
F7L2



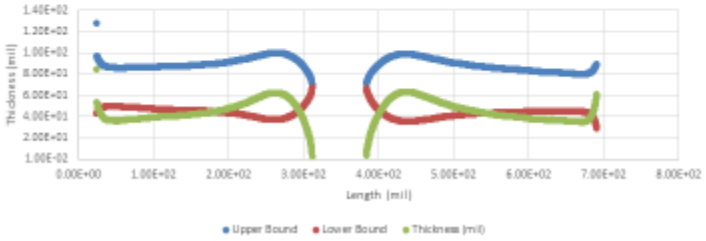
38

Thickness Profiles F7 Short Cell

F7S1



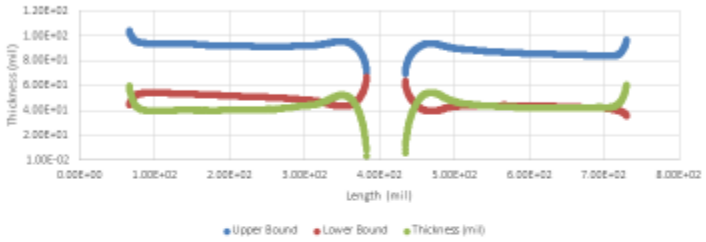
F7S2



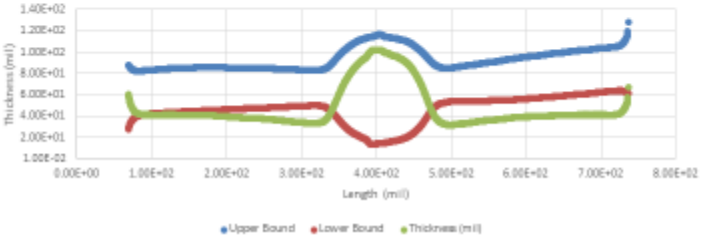
39

Thickness Profiles F8 Long Cell

F8L1



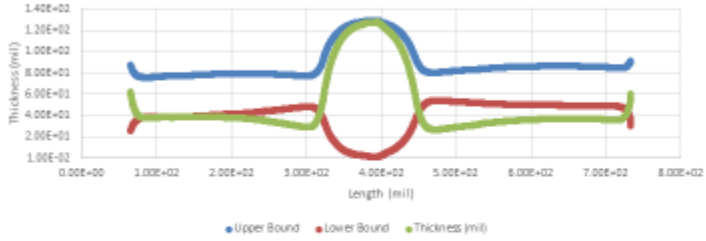
F8L2



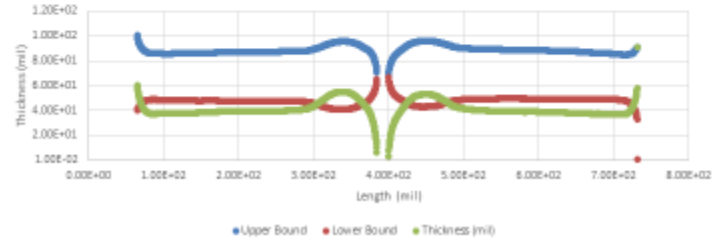
40

Thickness Profiles F8 Short Cell

F8S1



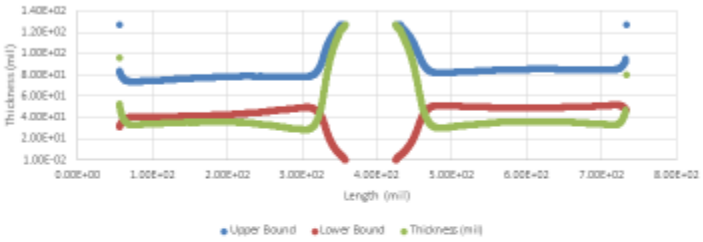
F8S2



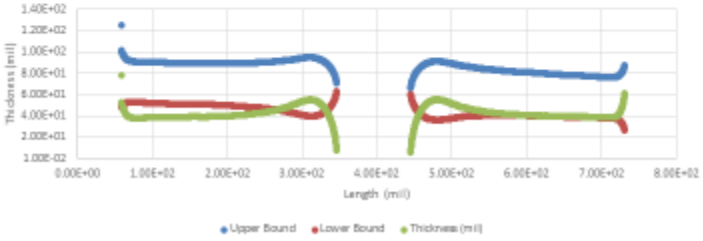
41

Thickness Profiles F9 Long Cell

F9L1



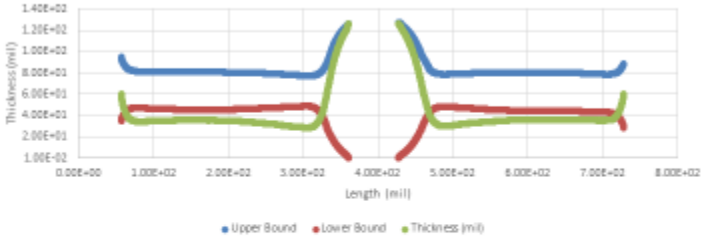
F9L2



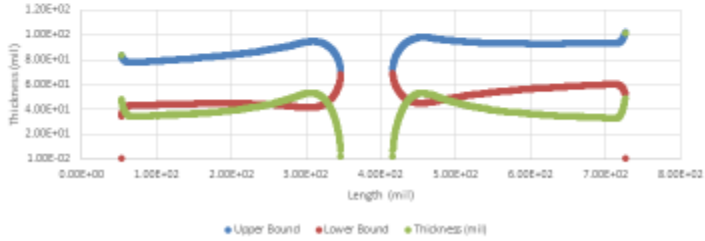
42

Thickness Profiles F9 Short Cell

F9S1



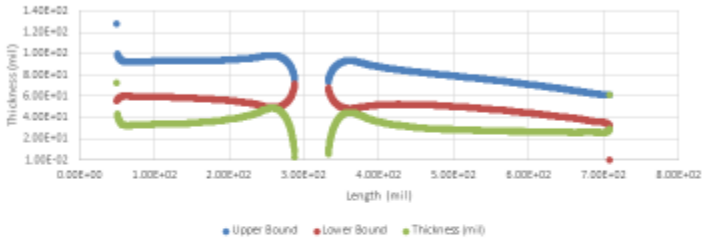
F9S2



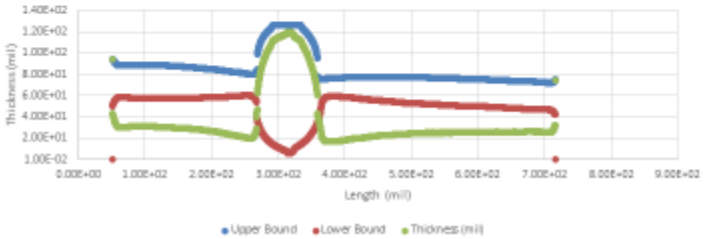
43

Thickness Profiles F14 Long Cell

F14L1



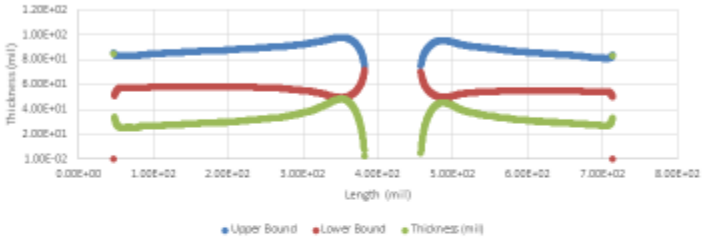
F14L2



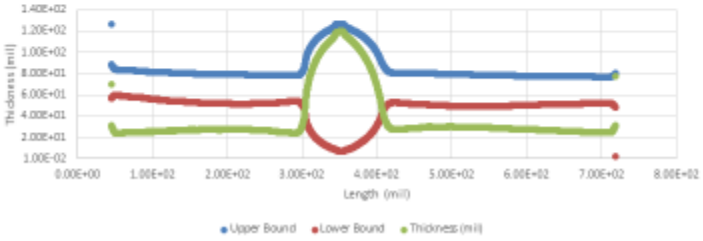
44

Thickness Profiles F14 Short Cell

F14S1



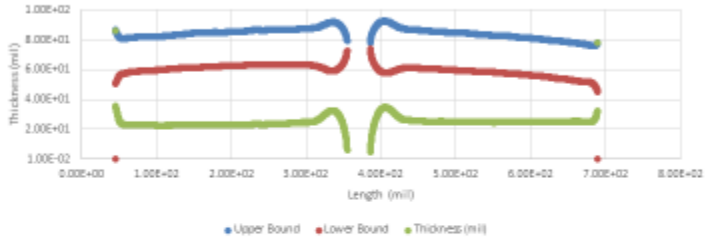
F14S2



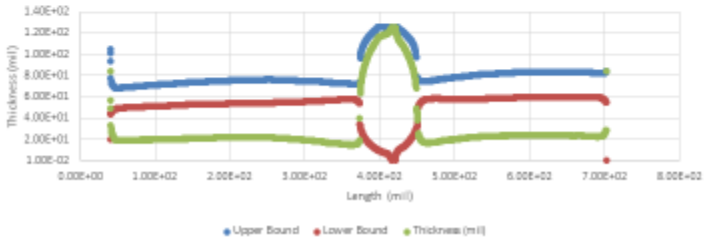
45

Thickness Profiles F15 Long Cell

F15L1



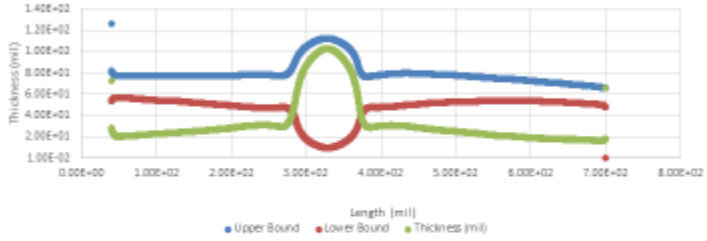
F15L2



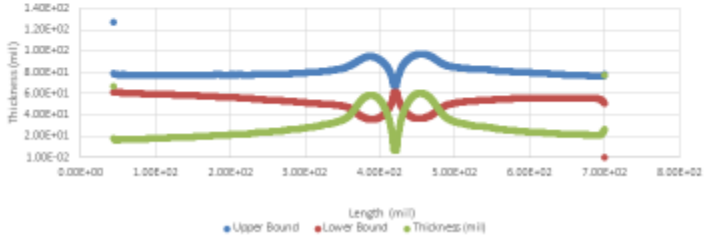
46

Thickness Profiles F15 Short Cell

F15S1

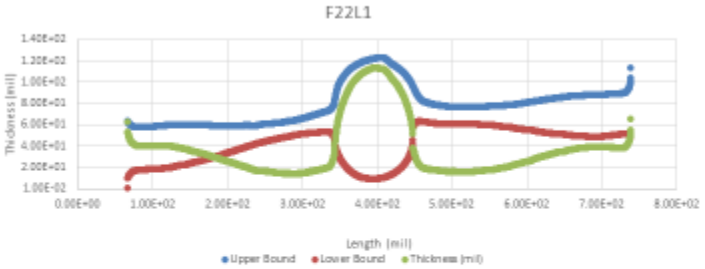


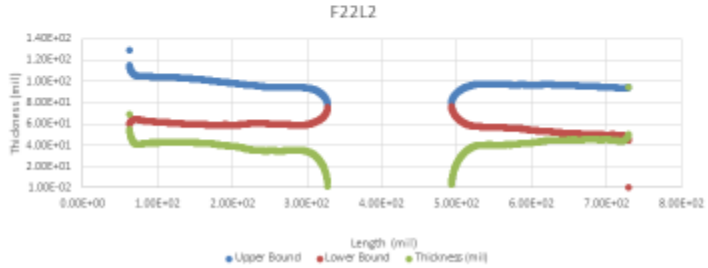
F15S2



47

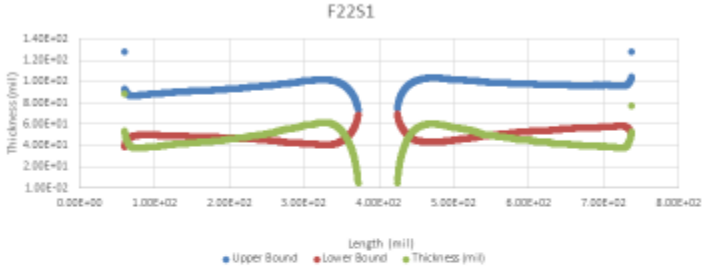
Thickness Profiles F22 Long Cell

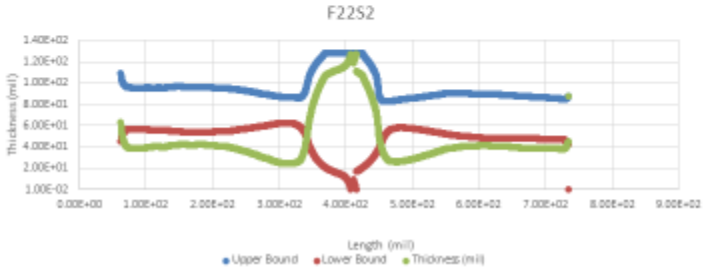




48

Thickness Profiles F22 Short Cell

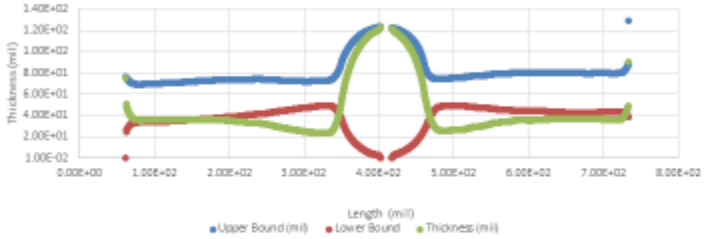




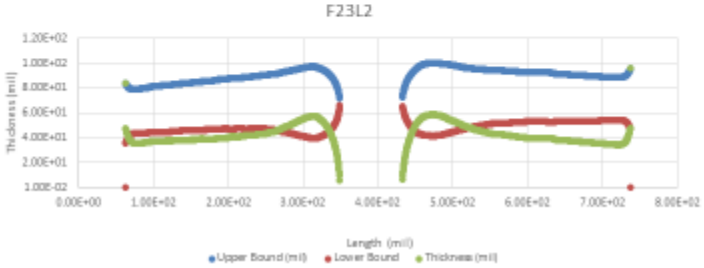
49

Thickness Profiles F23 Long Cell

F23L1



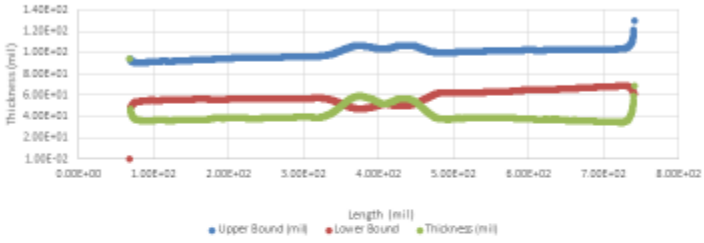
F23L2



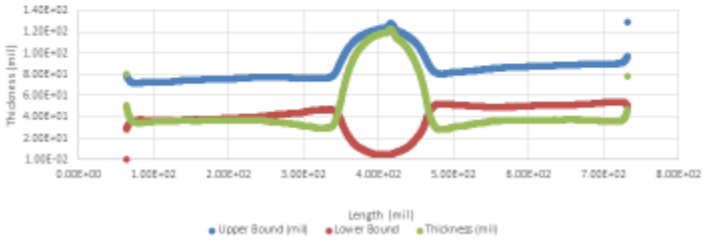
50

Thickness Profiles F23 Short Cell

F23S1

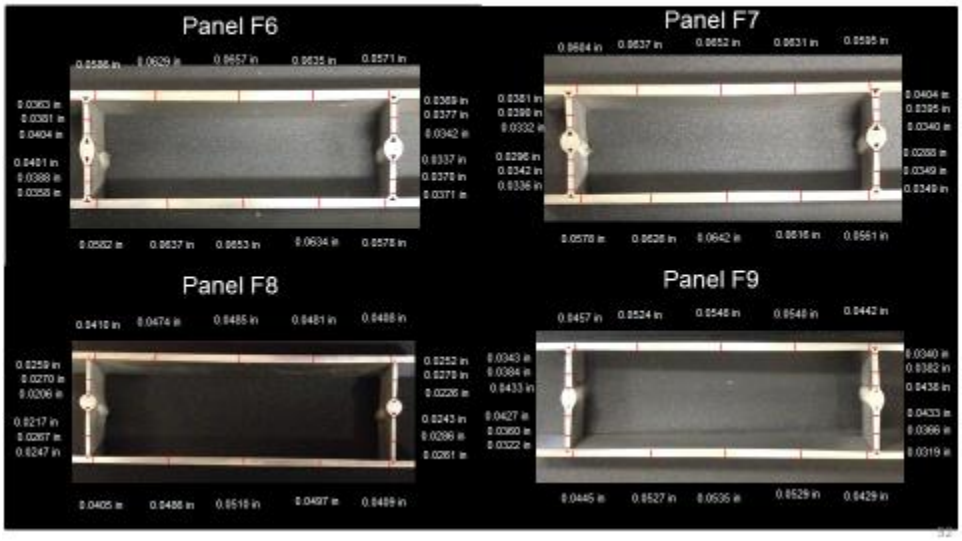


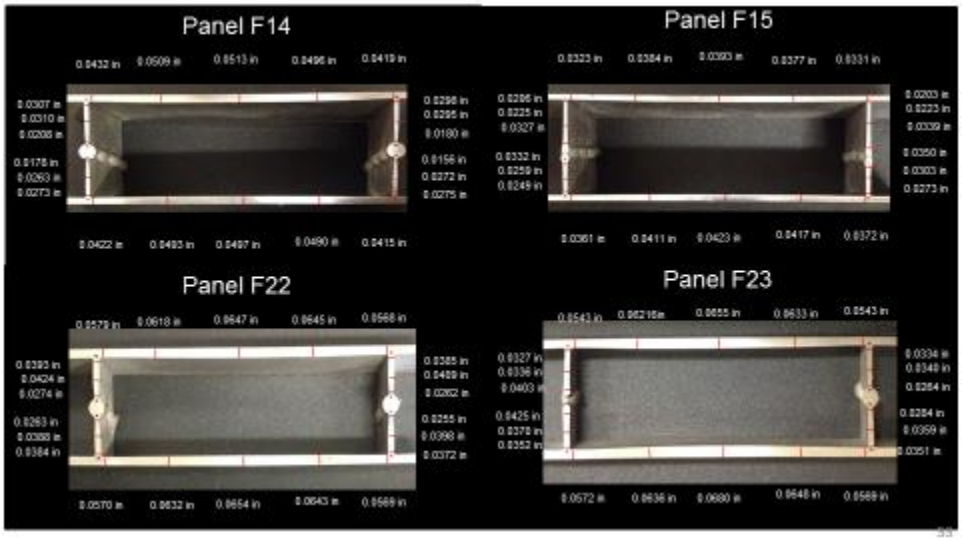
F23S2



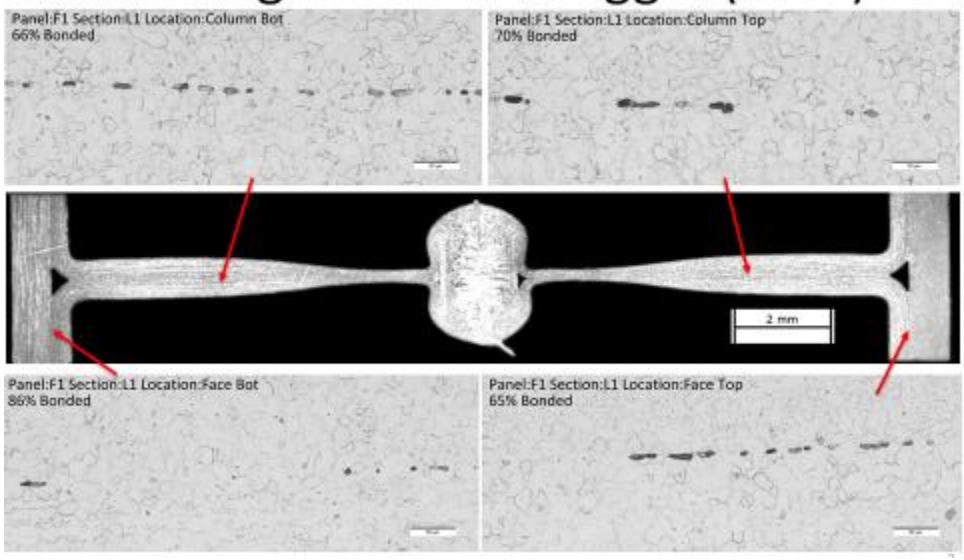
51

Panel Measurements

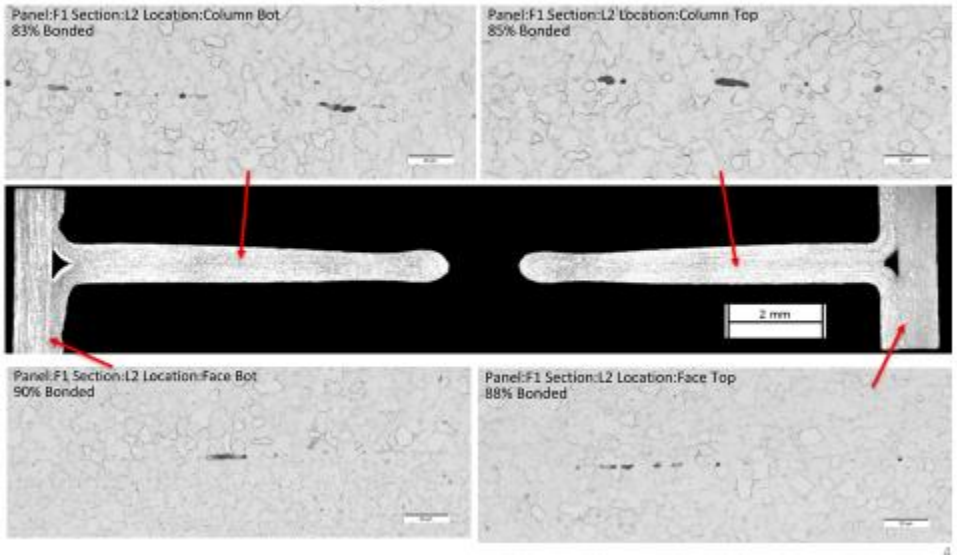




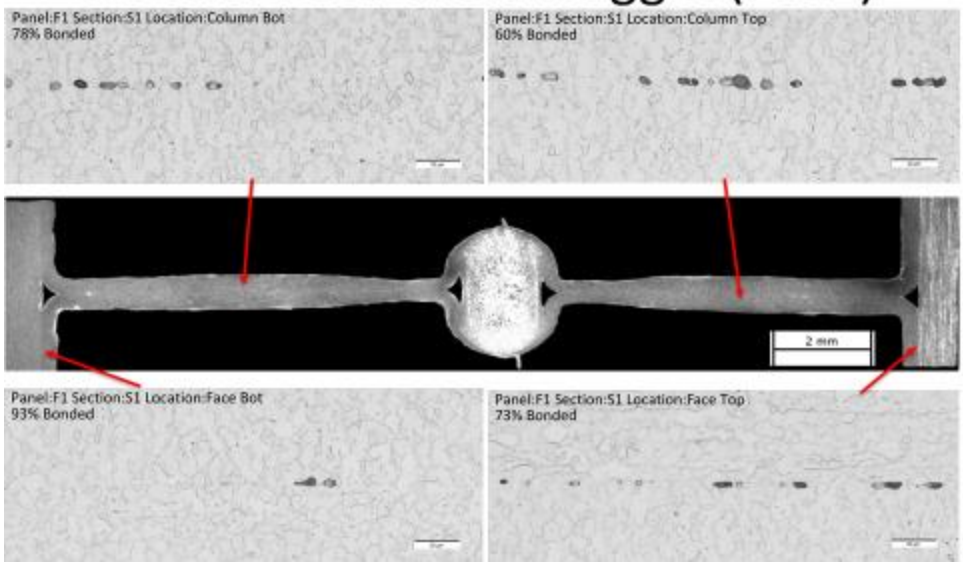
F1 Long Cell Weld Nugget (F1L1)



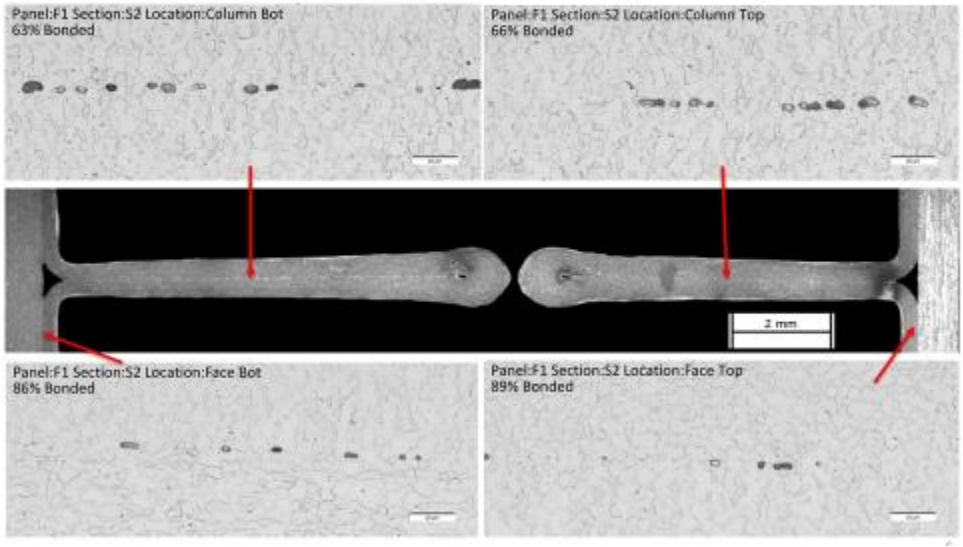
F1 Long Cell Hole (F1L2)



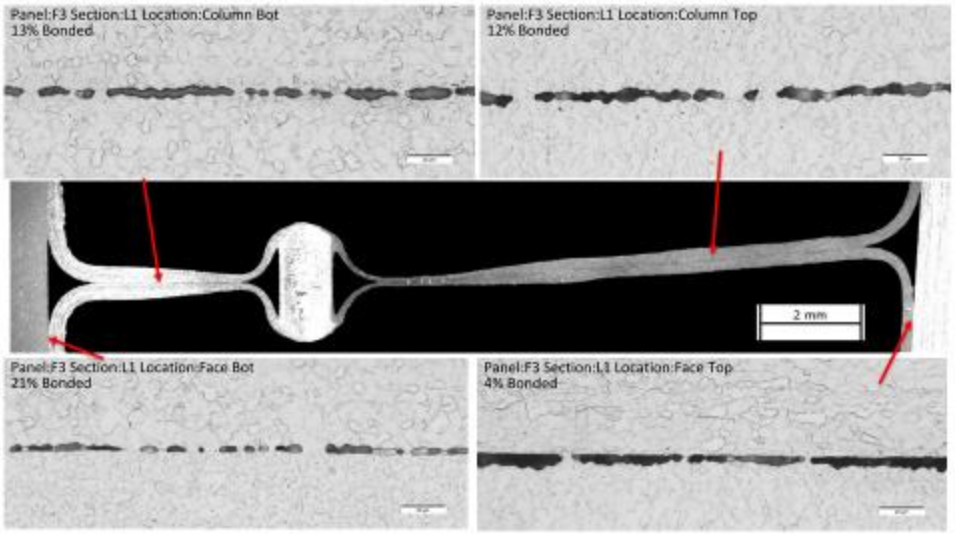
F1 Short Cell Weld Nugget (F1S1)



F1 Short Cell Hole (F1S2)

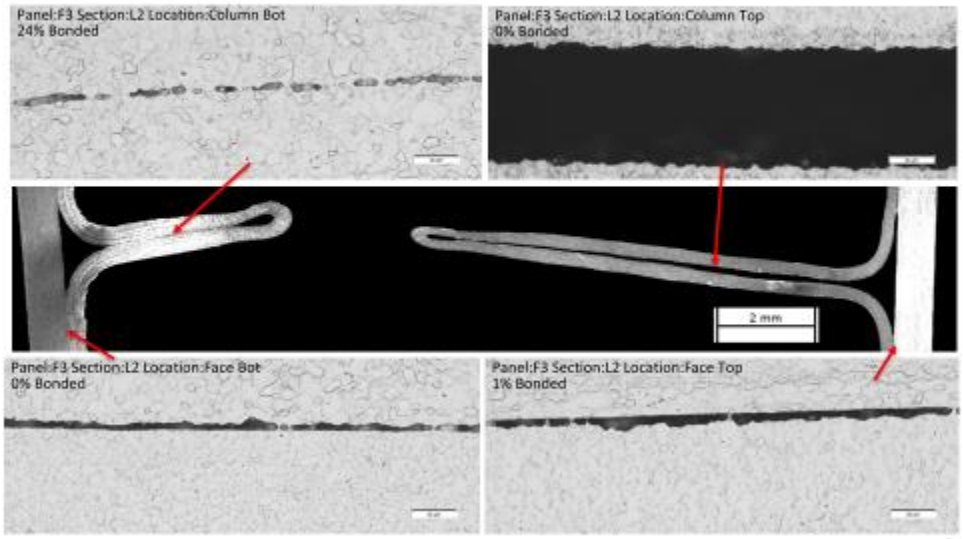


F3 Long Cell Weld Nugget (F3L1)

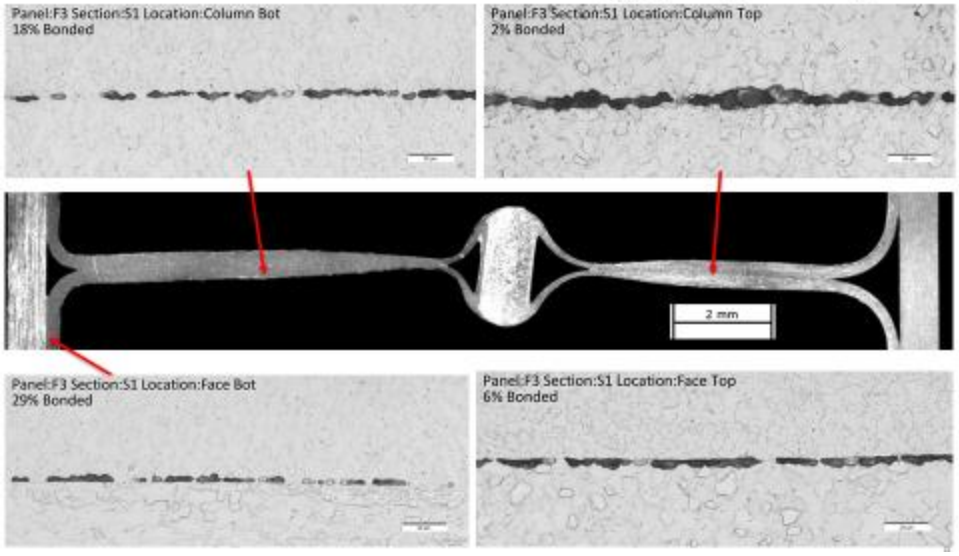


7

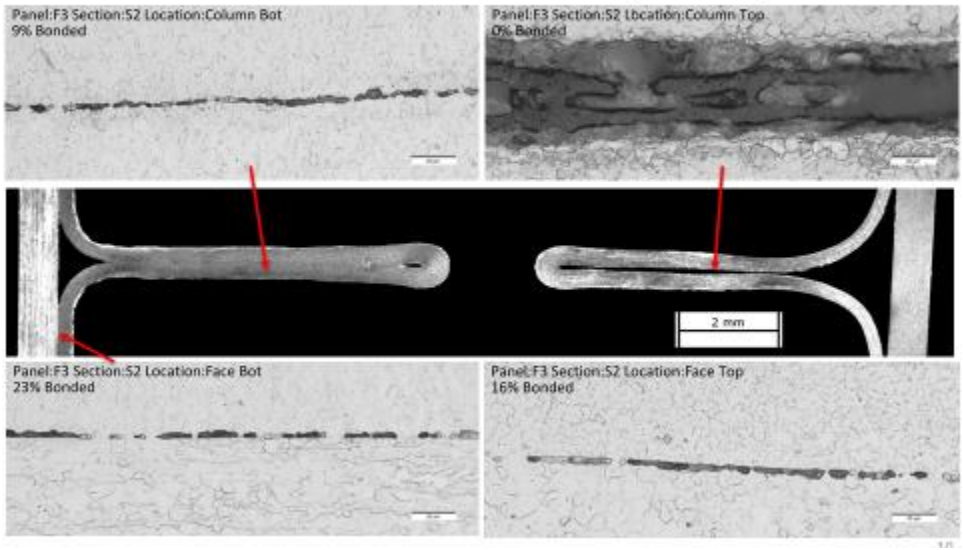
F3 Long Cell Hole (F3L2)



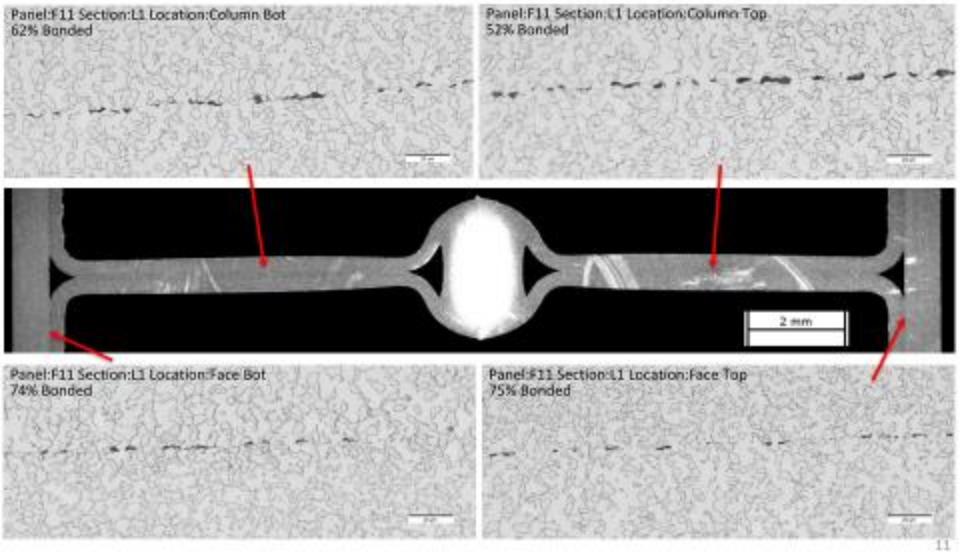
F3 Short Cell Weld Nugget (F3S1)



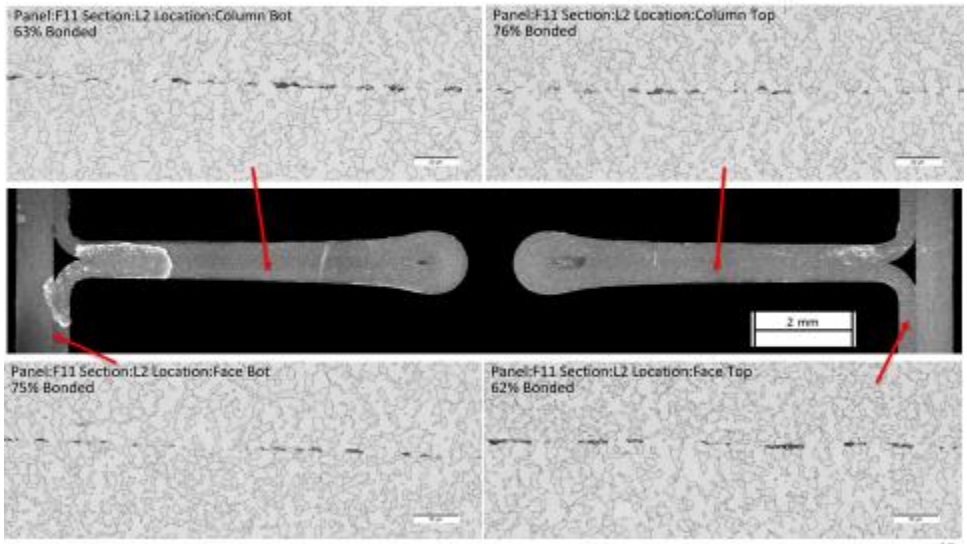
F3 Short Cell Hole (F3S2)



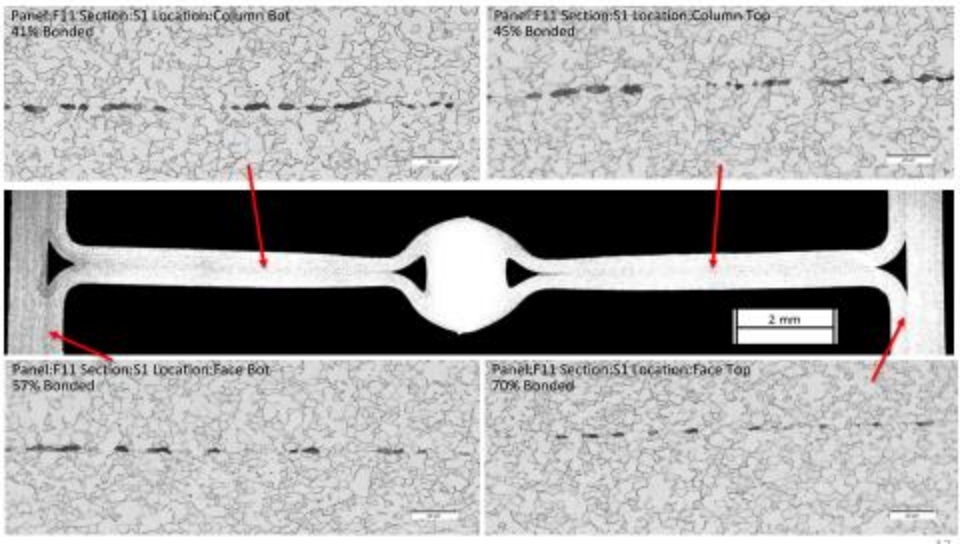
F11 Long Cell Weld Nugget (F11L1)



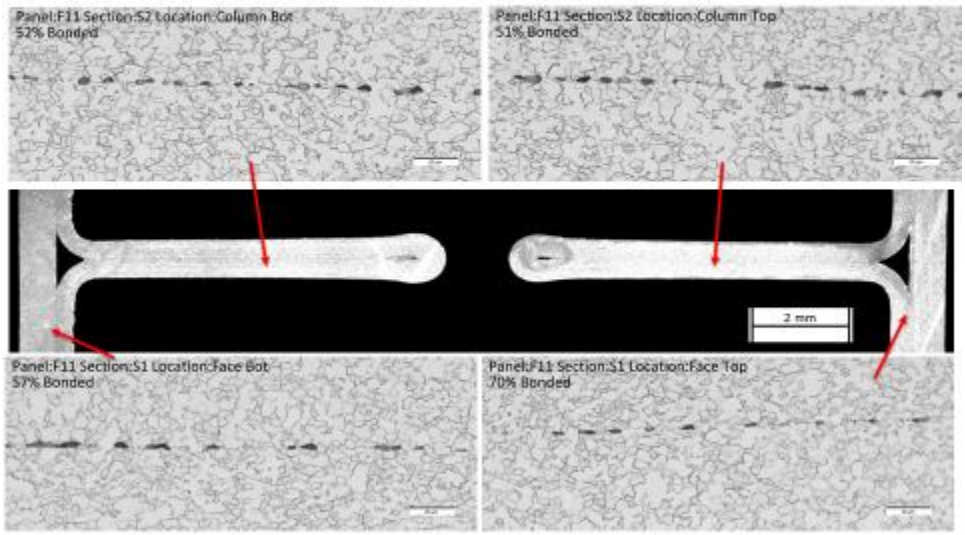
F11 Long Cell Hole (F11L2)



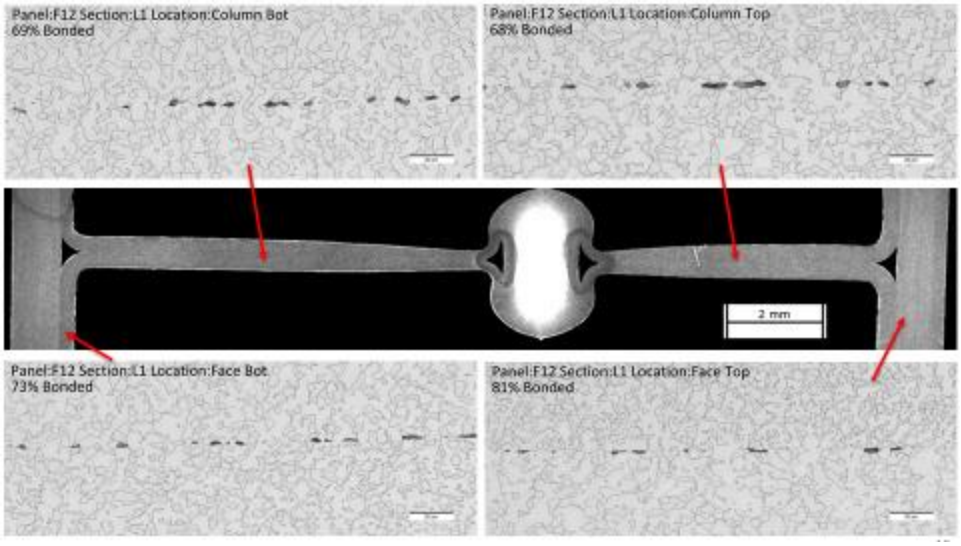
F11 Short Cell Weld Nugget (F11S1)



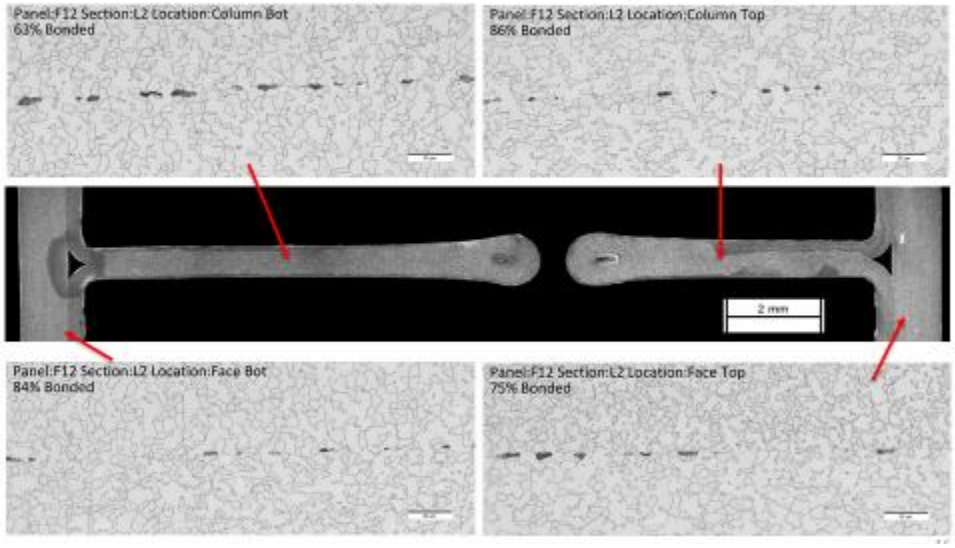
F11 Short Cell Hole (F11S2)



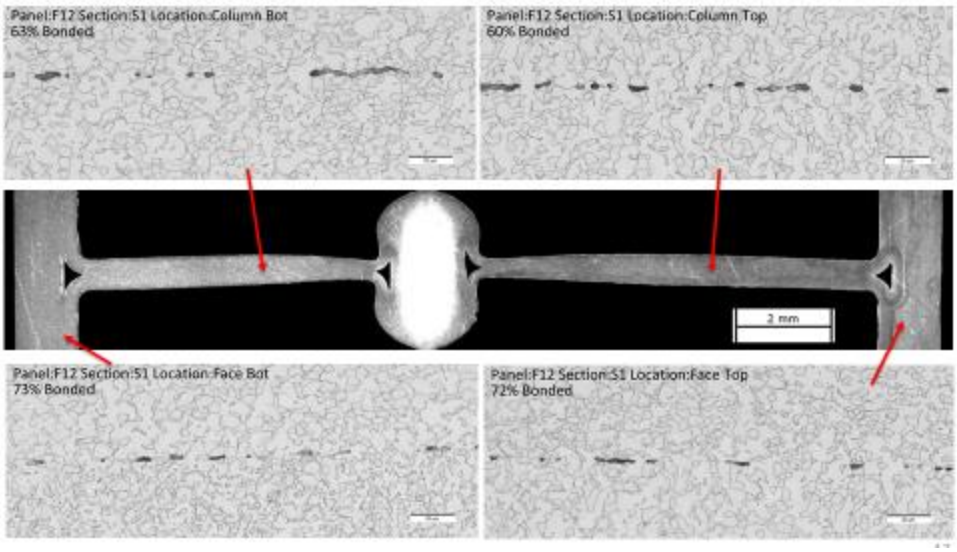
F12 Long Cell Weld Nugget (F12L1)



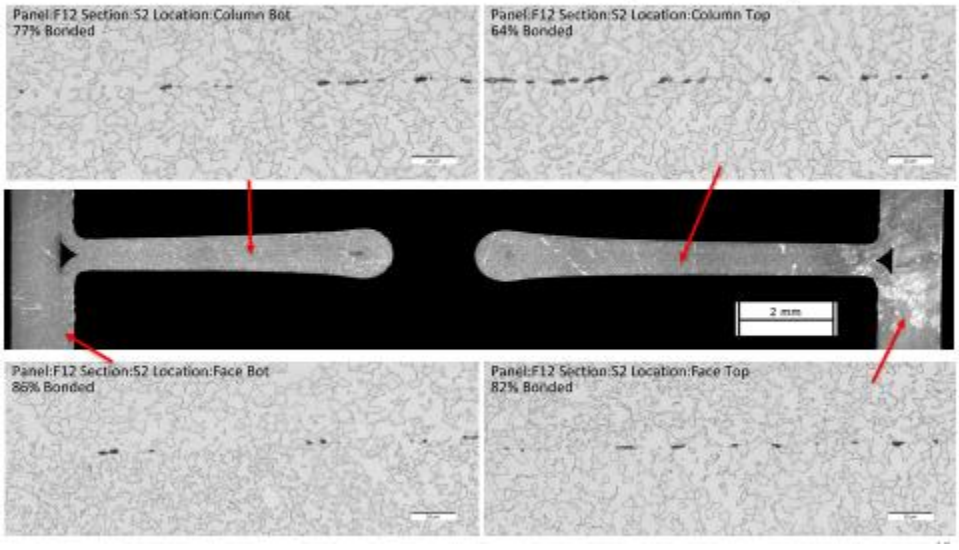
F12 Long Cell Hole (F12L2)



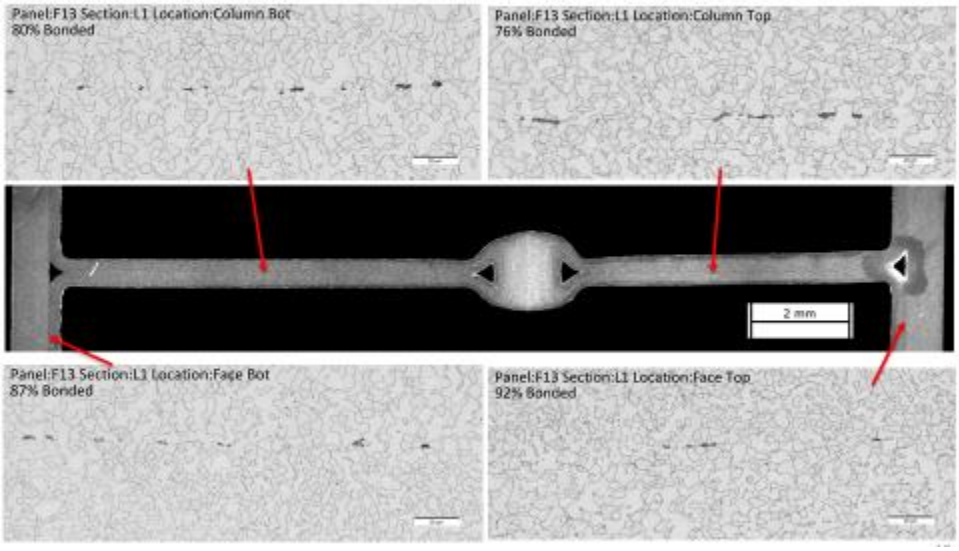
F12 Short Cell Weld Nugget (F12S1)



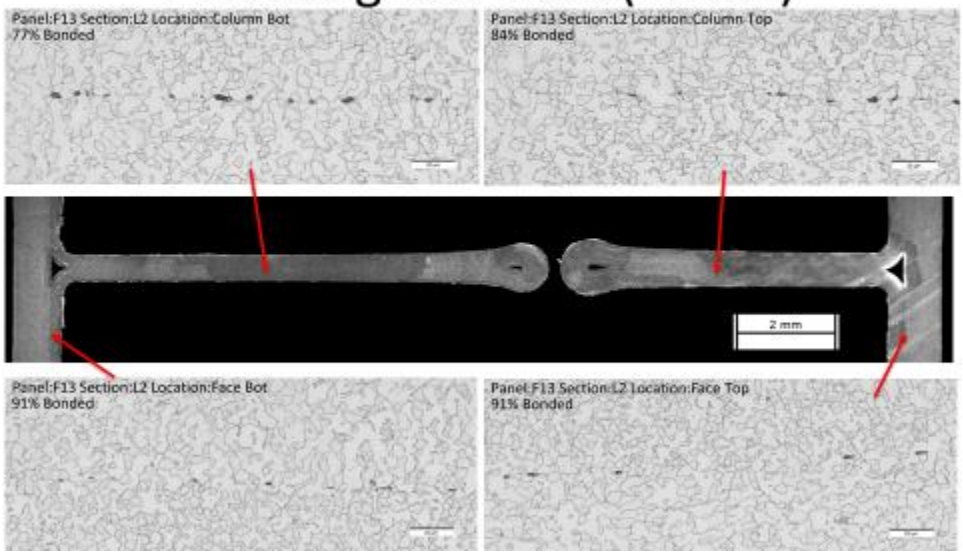
F12 Short Cell Hole (F12S2)



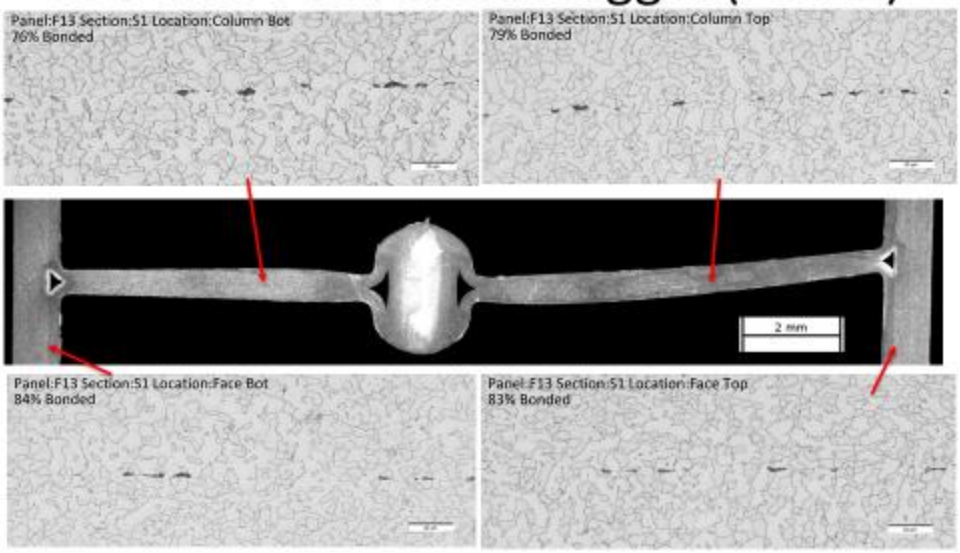
F13 Long Cell Weld Nugget (F13L1)



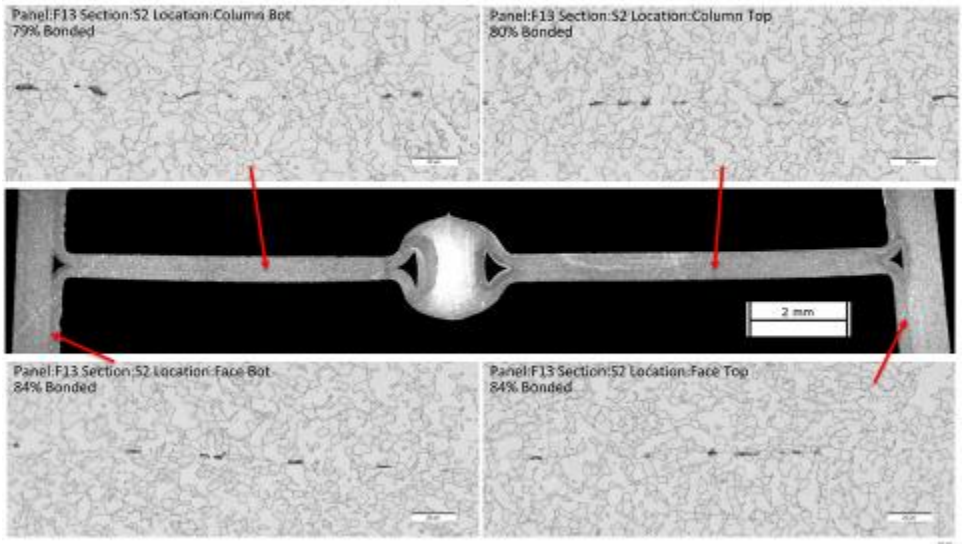
F13 Long Cell Hole (F13L2)



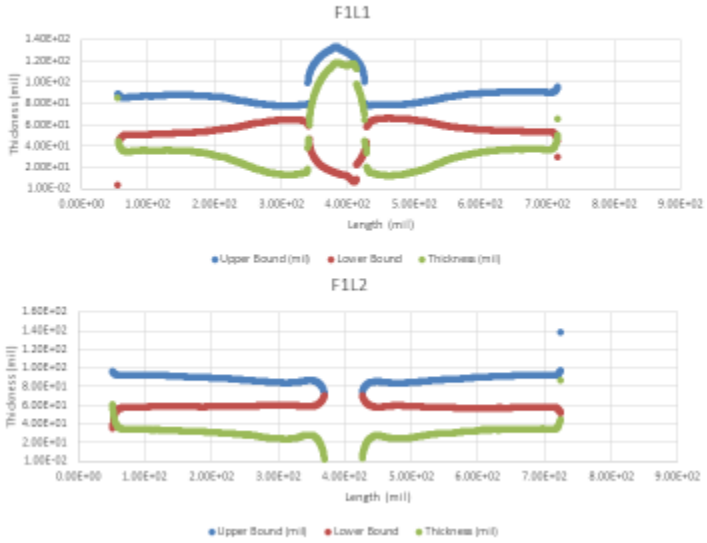
F13 Short Cell Weld Nugget (F13S1)



F13 Short Cell Weld Nugget (F13S2)

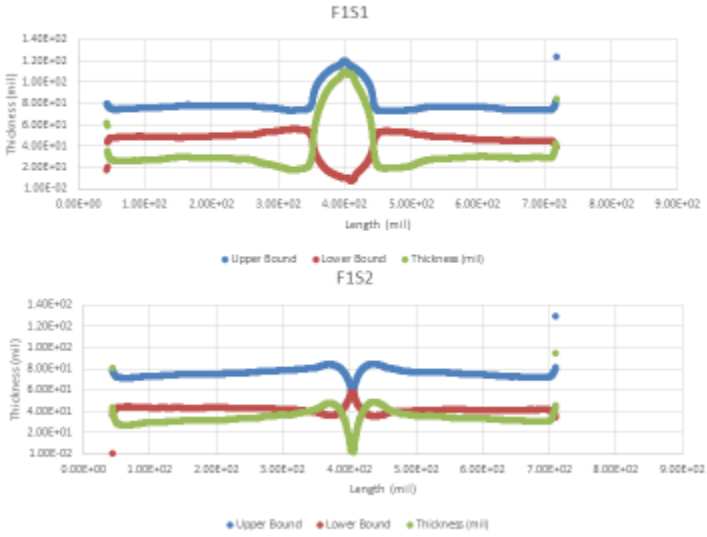


Thickness Profiles F1 Long Cell



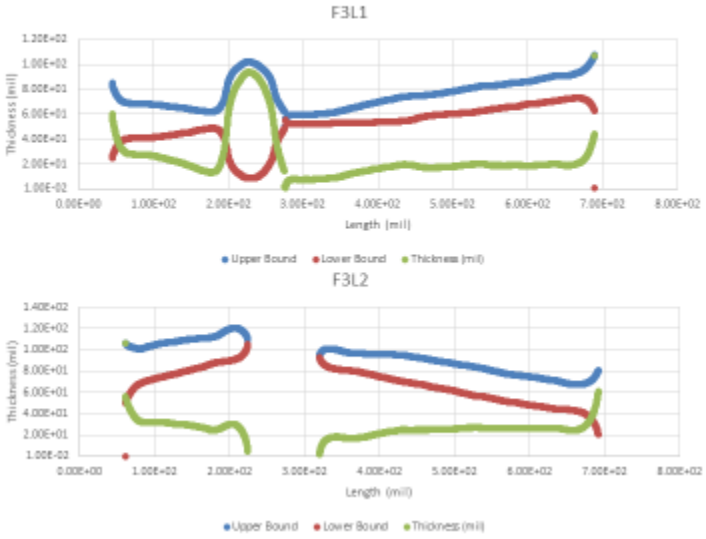
25

Thickness Profiles F1 Short Cell



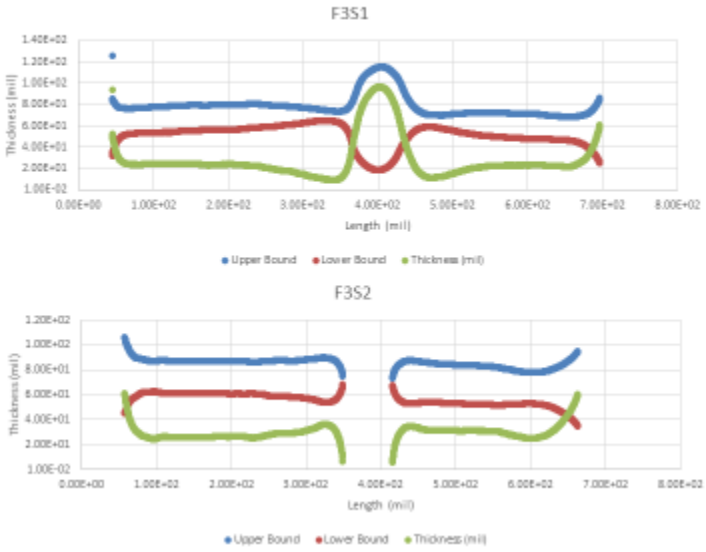
26

Thickness Profiles F3 Long Cell



27

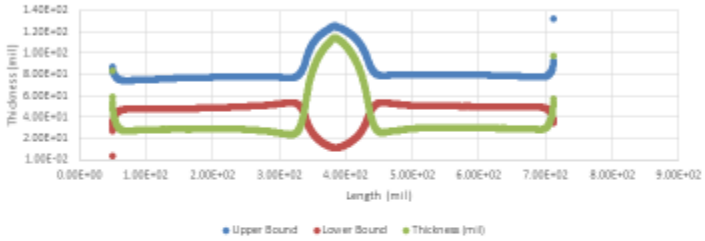
Thickness Profiles F3 Short Cell



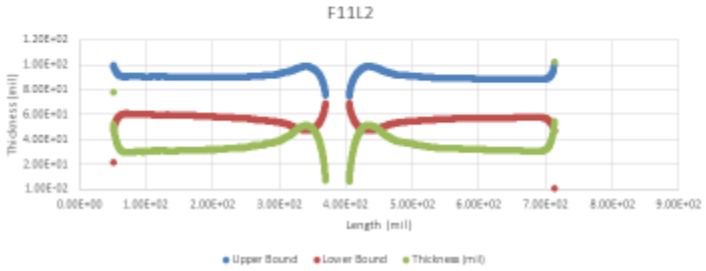
28

Thickness Profiles F11 Long Cell

F11L1



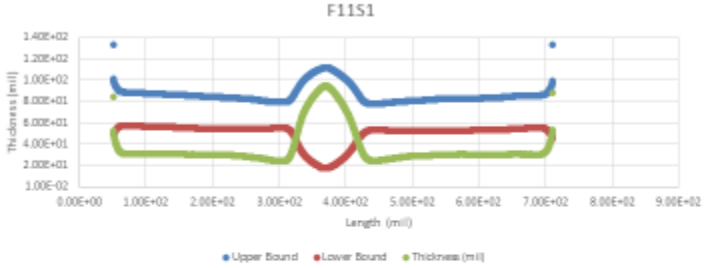
F11L2



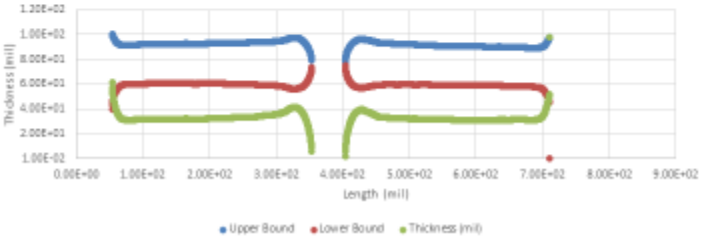
29

Thickness Profiles F11 Short Cell

F11S1



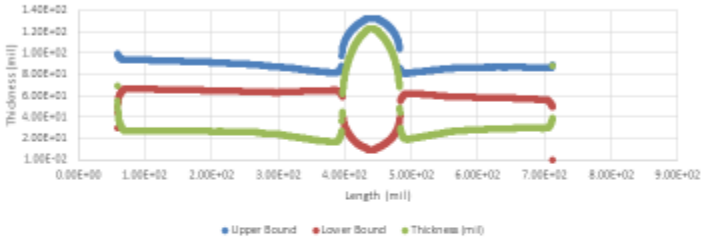
F11S2



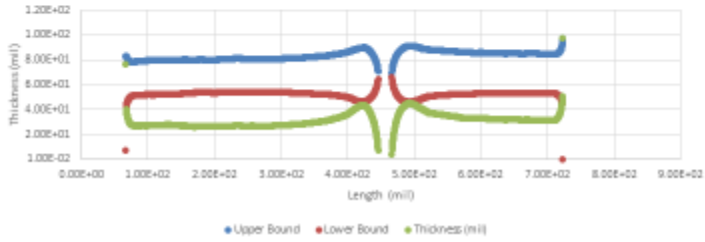
30

Thickness Profiles F12 Long Cell

F12L1



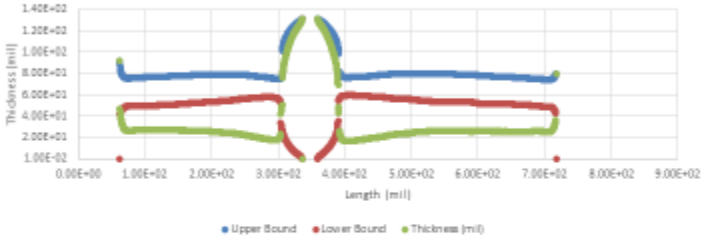
F12L2



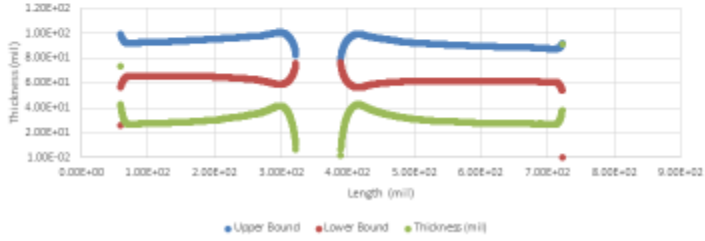
31

Thickness Profiles F12 Short Cell

F12S1

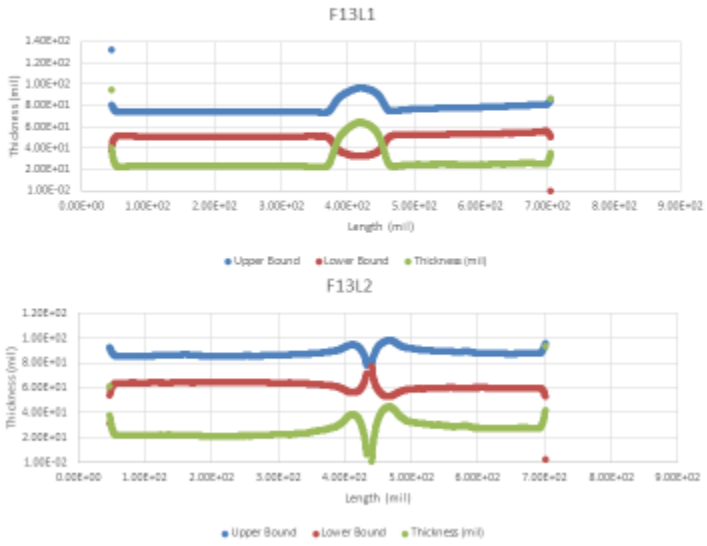


F12S2



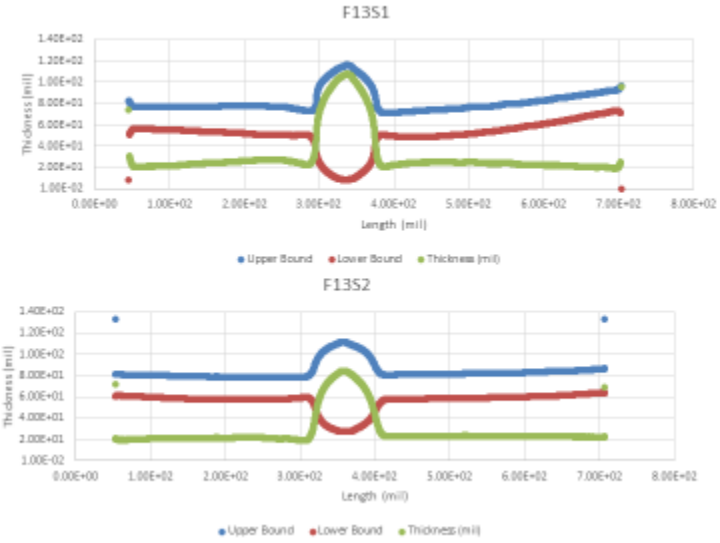
32

Thickness Profiles F13 Long Cell

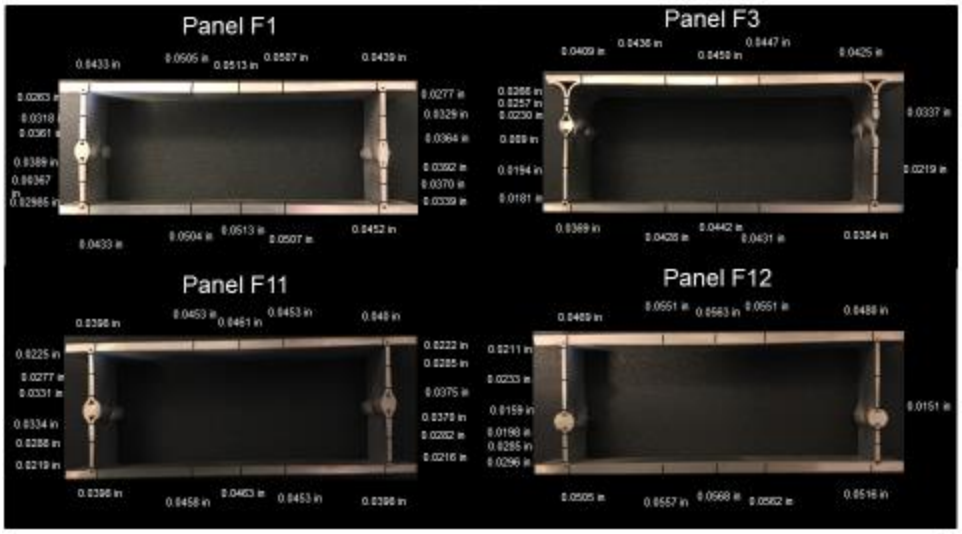


33

Thickness Profiles F13 Short Cell

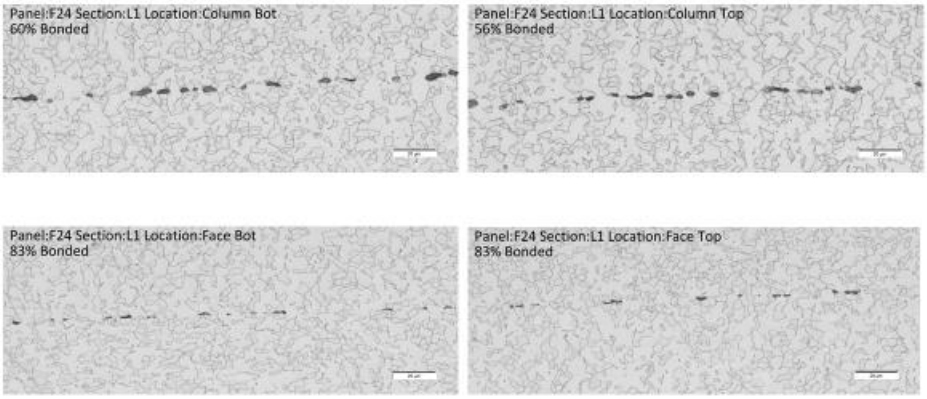


34

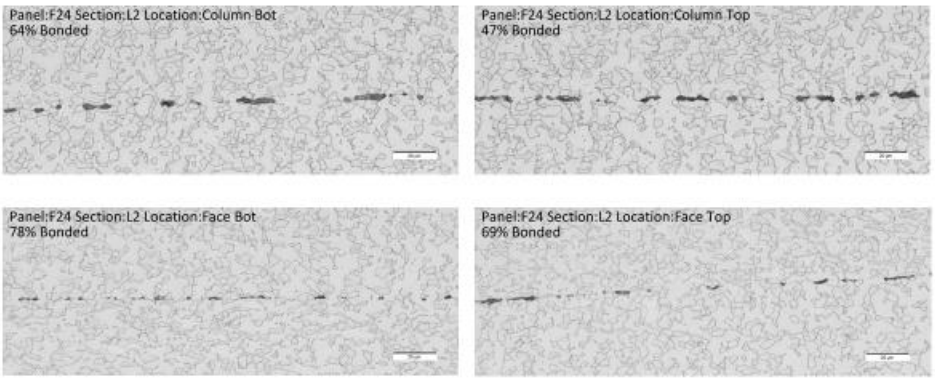

Panel F1
Panel F3
Panel F11
Panel F12


Panel F13

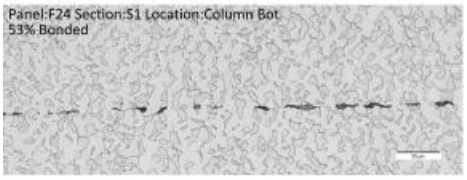
F24 Long Cell Weld Nugget (F24L1)

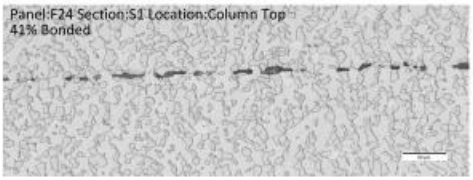


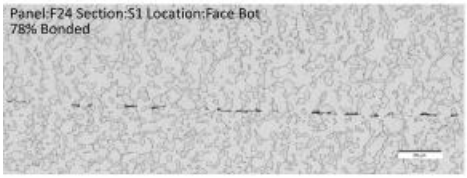
F24 Long Cell Hole (F24L2)

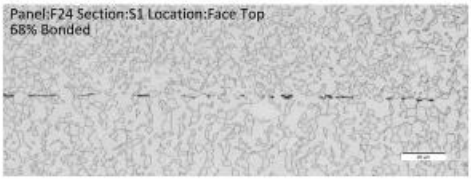


F24 Short Cell Weld Nugget (F24S1)

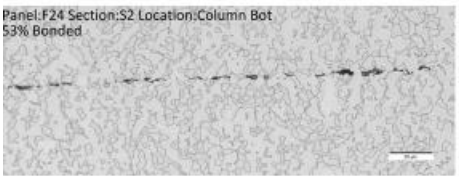


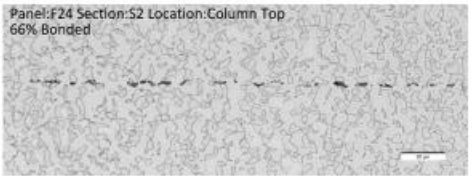


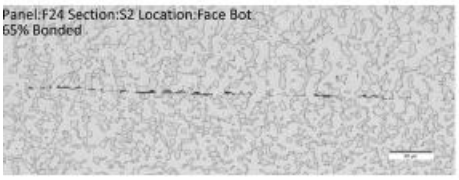




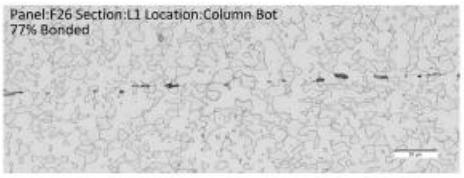
F24 Short Cell Hole (F24S2)

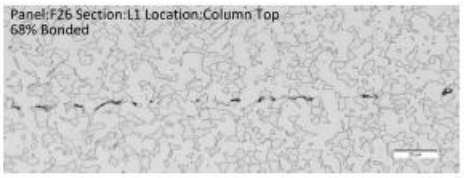


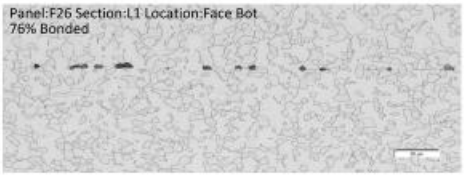


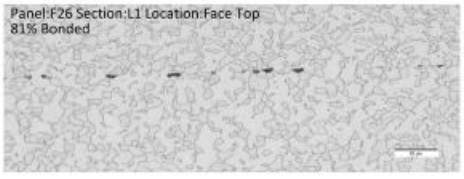


F26 Long Cell Weld Nugget (F26L1)

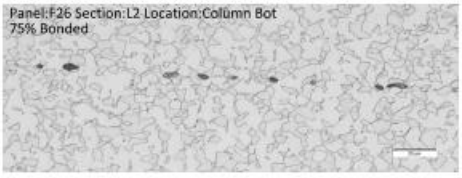


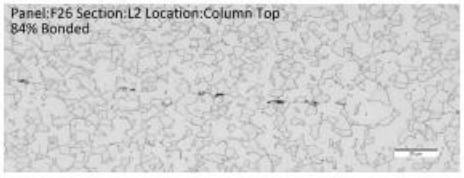


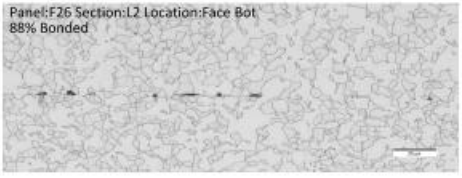




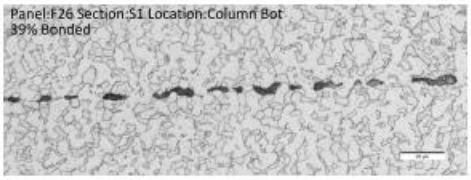
F26 Long Cell Hole (F26L2)

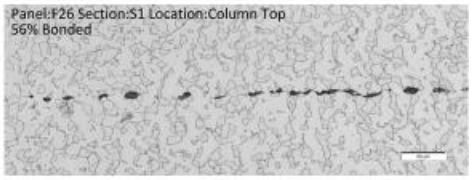


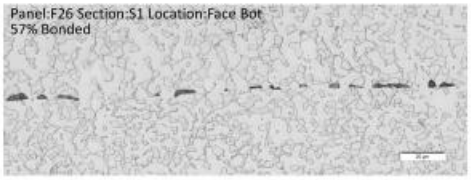


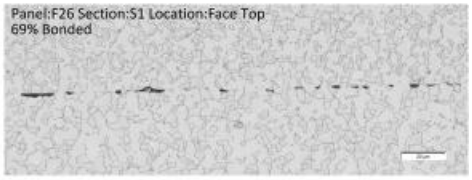


F26 Short Cell Weld Nugget (F26S1)

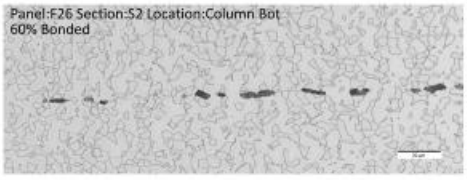


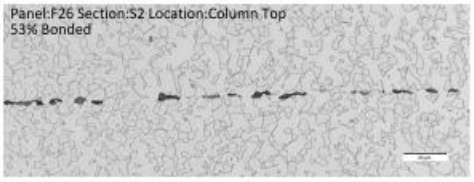


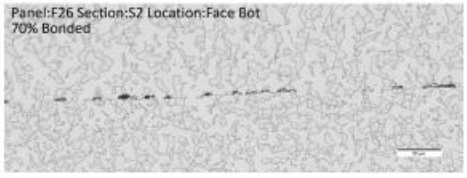


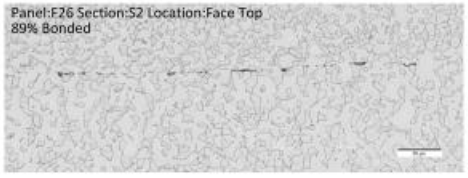


F26 Short Cell Hole (F26S2)



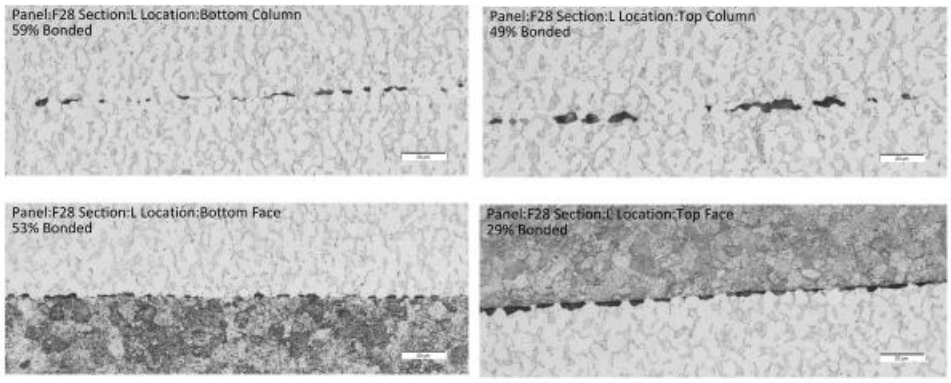




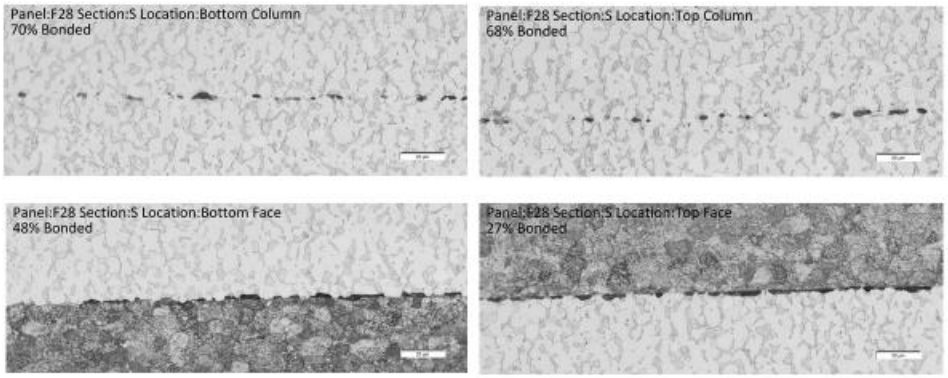


RTI MAI and Additional Panels

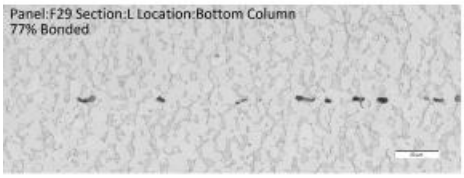
F28 Long

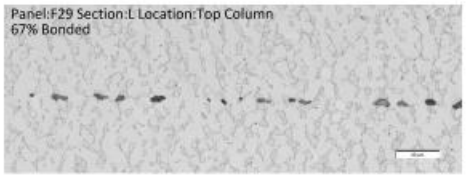


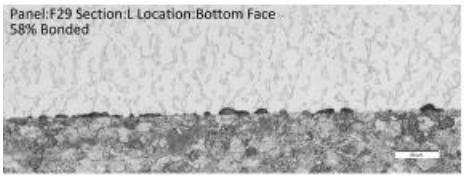
F28 Short

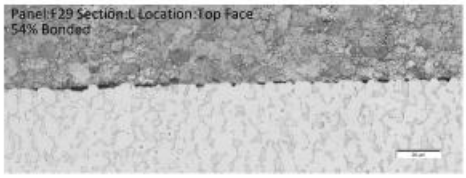


F29 Long

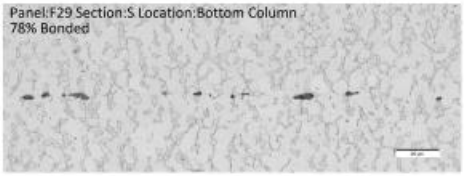


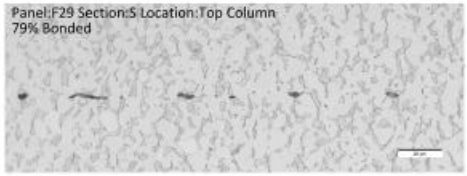


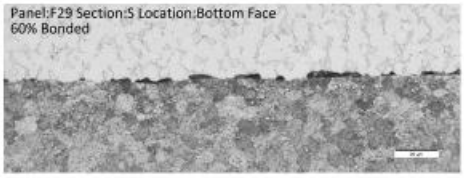


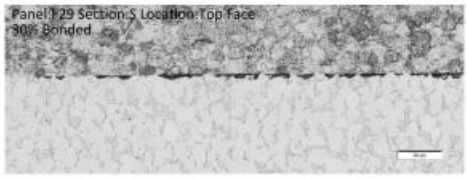


F29 Short

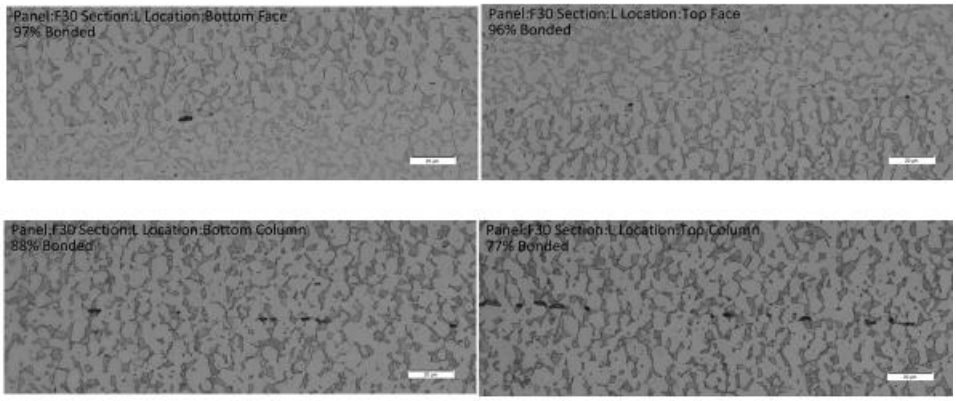




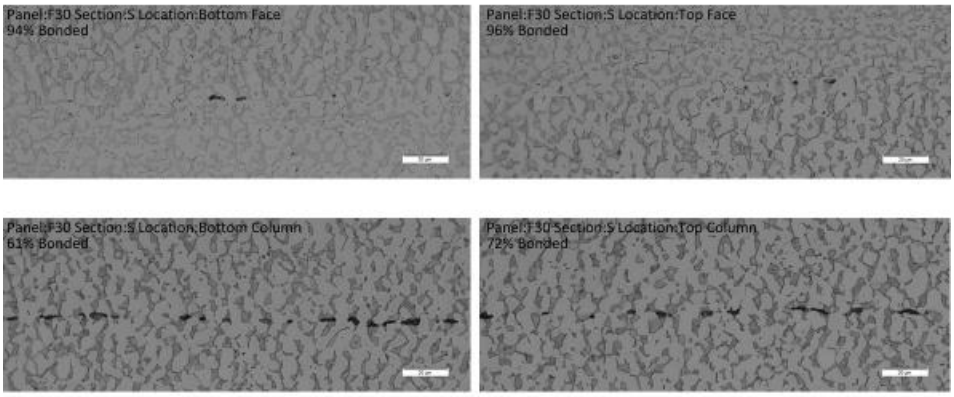




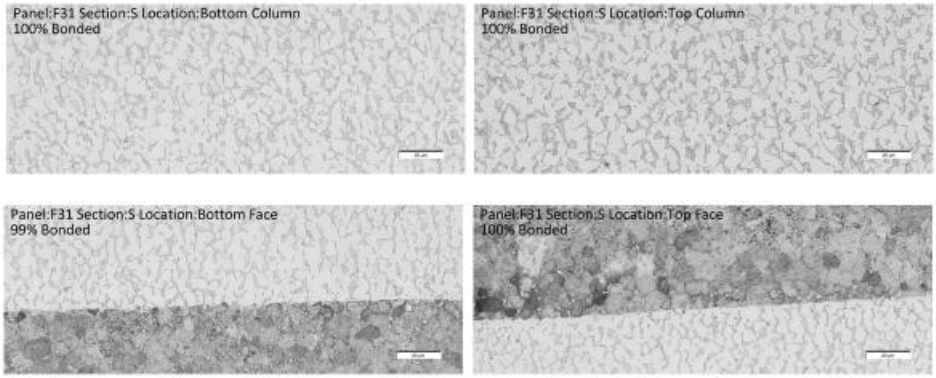
F30 Long



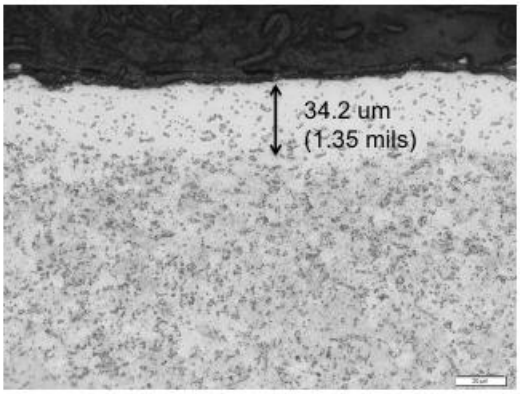
F30 Short



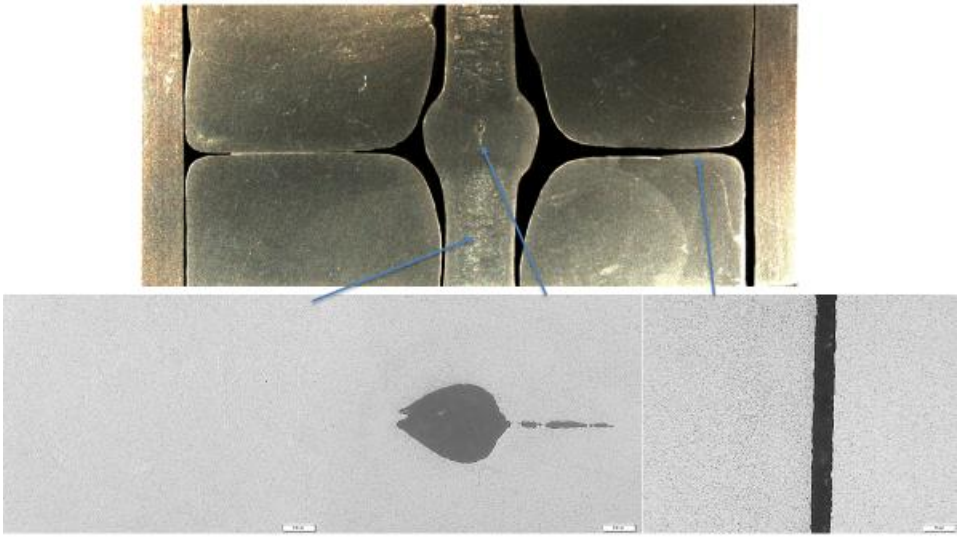
F31 Short



F31 Alpha Case

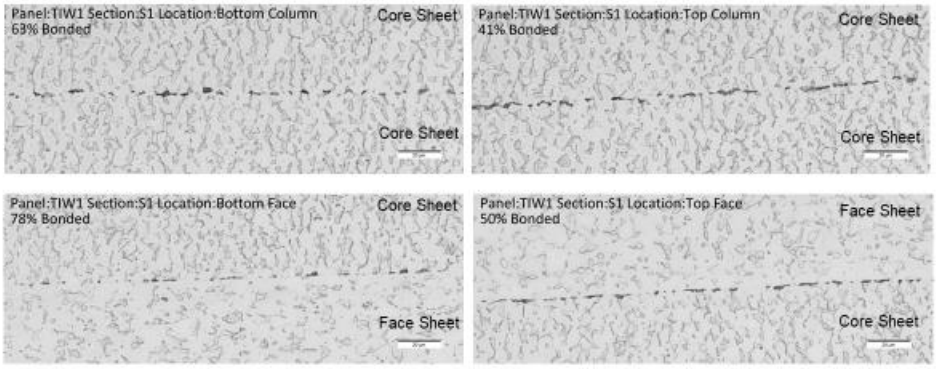


F31 Crossed Nodes

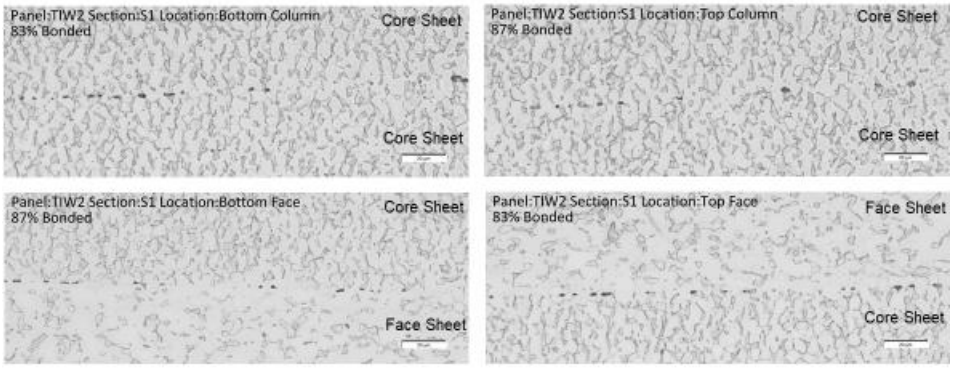


Temperature Study

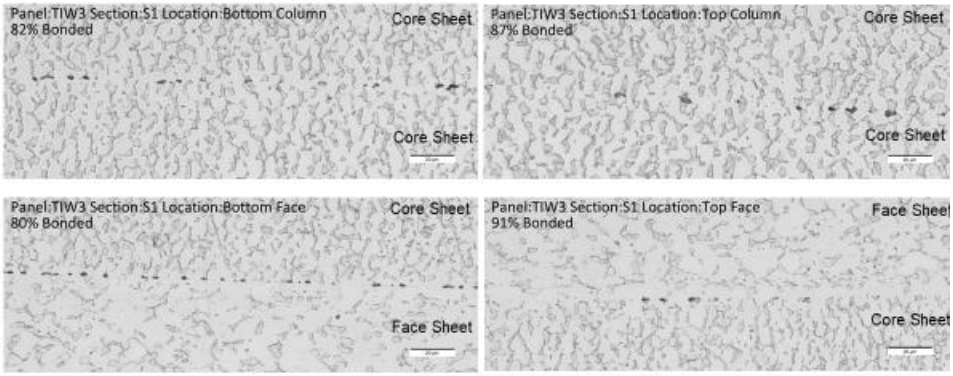
Panel 1 – 1475F Short Cell Weld Nugget



Panel 2 – 1500F Short Cell Weld Nugget



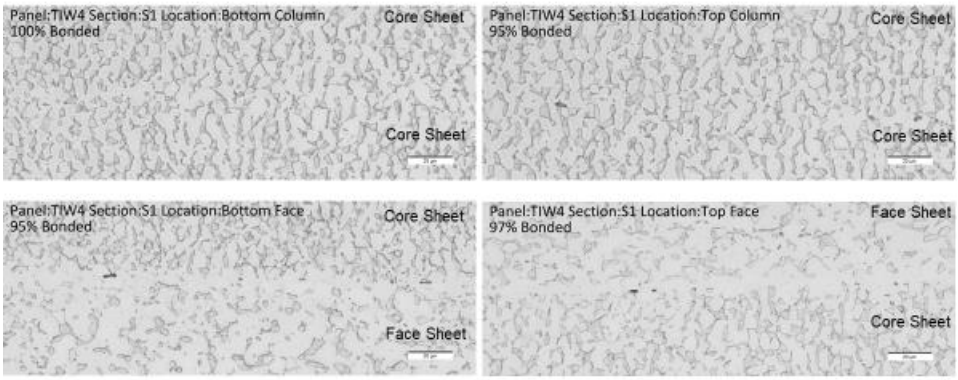
Panel 3 – 1525F Short Cell Weld Nugget



The bottom face of panel 3 had uncharacteristically low bonding and quite a few more voids than the top face. It also had a delamination in one of the trimmings received from Boeing. This may be local contamination and may not represent the normal bonding at this temperature. More samples will help determine this.

7

Panel 4 – 1550F Short Cell Weld Nugget



F Programs

Bondline Calculator

```
# -*- coding: utf-8 -*-
"""
Created on Thu Apr 28 15:18:33 2016

@author: BJF
"""

import numpy as np
import numpy.matlib as M
import matplotlib.pyplot as plt
import math
import pylab
from scipy import ndimage
from scipy import interpolate as interp
from PIL import Image
import cv2
import csv



plt.close(1)
plt.close(2)
plt.close(3)
plt.close(4)
plt.figure(1)
#img=Image.open('64bondline.jpg').convert("L")
#img=np.asarray(img)
micro=10
micro=micro-1

ftype='.tif'
flocation='C:\\Users\\BJF\\OneDrive for Business\\BA18_5\\TIW Temp Study\\Micrographs\\'
setting=np.genfromtxt(flocation+'TIW_settings.csv',delimiter=',')
setting=setting[:,1:]
names=np.genfromtxt(flocation+'TIW_settings.csv',delimiter=',',dtype="S10")
names=names[:,0]
fname=names[micro]#+'50x'
s_mask=7
img=cv2.imread(flocation+fname+ftype,0)
fname=names[micro]
img=img[20:-1,:]
Threshold_ratio=setting[micro,0]
Threshold_value=setting[micro,1]
blursmallvalue=np.int(setting[micro,2])
lowercrop=int(setting[micro,3])
topcrop=lowercrop+8
ftype='.png'

scale=cv2.imread('scalemse.tif',0)
scalemstl=cv2.imread('scalemstl.tif',0)
other=np.copy(img)
pixelsize=20./195
#20x=
plt.subplot(211)
plt.title('Original Image')
plt.imshow(img, cmap='Greys_r')
```

```
blur=cv2.medianBlur(img,15)

blur=blur[img.shape[0]*lowercrop/16:img.shape[0]*topcrop/16,:]
img=img[img.shape[0]*lowercrop/16:img.shape[0]*topcrop/16,:]
#blur=cv2.GaussianBlur(blur,(5,5),0)


#blur[blur>150]=255
#blur[blur<100]=0
#conv0=ndimage.convolve(img,np.array([[1,0,-1],[2,0,-2],[1,0,-1]]))
#conv1=ndimage.convolve(img,np.array([[1,2,1],[0,0,0],[-1,-2,-1]]))
#conv=np.sqrt(conv0**2+conv1**2)
#plt.imshow(blur, cmap='Greys_r')
sobel0=np.abs(cv2.Sobel(blur, cv2.CV_64F, 1, 0, ksize=s_mask))
sobel1=np.abs(cv2.Sobel(blur, cv2.CV_64F, 0, 1, ksize=s_mask))
sobel=np.hypot(sobel1,sobel0)
sobel*=255.0/np.max(sobel)
plt.subplot(212)
#plt.imshow(sobel, cmap='Greys_r')

canny=cv2.Canny(img,100,255)
plt.imshow(canny, cmap='Greys_r')
ret,thresh = cv2.threshold(blur,np.mean(blur)/Threshold_ratio,255,cv2.THRESH_BINARY)
plt.title('Detected Voids After Median Blur and Threshold')
plt.imshow(thresh, cmap='Greys_r')
bl=np.ones((thresh.shape[1]-12,2))
n=6
for i in range(6,thresh.shape[1]-6):
    threshloc=thresh[:,i-1:i+1]
    loc=np.transpose(np.nonzero(threshloc<255))
    bl[i-n,0]=i
    bl[i-n,1]=np.mean(loc[:,0])
    if np.isnan(bl[i-n,1])==True:
        n=n+1
bl=bl[0:(i-n),:]
spl=np.polyfit(bl[:,0],bl[:,1],1)
#y=interp.splev(bl[:,0],spl)
plt.figure(2)
plt.subplot(211)
plt.xlabel('X Pixels')
plt.ylabel('Y Pixels')
plt.title('Curve Fit of Y-Averaged Voids')
plt.plot(bl[:,0],bl[:,1])
ynew=np.polyval(spl,bl[:,0])
plt.plot(bl[:,0],np.polyval(spl,bl[:,0]))

bllocx=np.arange(0,img.shape[1]-1)
bllocy=np.polyval(spl,bllocx)
bllocy=bllocy.astype(int)
#plt.plot(bllocx,bllocy)
overlay=np.copy(img)
overlay[bllocy,bllocx]=0
plt.subplot(212)
plt.title('Original Image with Calculated Bondline')
plt.imshow(overlay, cmap='Greys_r')
plt.figure(3)
plt.subplot(211)
blv=img[bllocy,bllocx]
plt.xlabel('X Pixels Along Bondline')
plt.ylabel('Grayscale (8-bit)')
plt.title('Grayscale Intensity Along Bondline')
plt.plot(bllocx,blv)
```

```
blursmall=cv2.medianBlur(img,blursmallvalue)

blrange=15 #pixels
blrangemat=np.arange(-blrange-1,blrange)[:, np.newaxis]
blvblur=np.min(blursmall[bllocy+blrangemat,np.tile(bllocx,(blrange*2+1,1))],0)
plt.subplot(212)
plt.xlabel('X Pixels Along Bondline')
plt.ylabel('Grayscale (8-bit)')
plt.title('Blurred Grayscale Intensity Along Bondline with Threshold')
plt.plot(bllocx,blvblur)
threshold=(np.max(blvblur)-np.min(blvblur))/Threshold_ratio+np.min(blvblur)
threshold=Threshold_value
plt.plot(bllocx,np.ones((bllocx.shape[0],1))*threshold)
voids=np.linspace(1,blvblur.size,blvblur.size)[blvblur<threshold]
voidpixels=voids.shape[0]+0.0
bondratio=(blvblur.shape[0]-voidpixels)/blvblur.shape[0]
bondoverlay=np.copy(img)
bondoverlay[bllocy,bllocx]=0
bondoverlay[bllocy[blvblur<threshold],bllocx[blvblur<threshold]]=255
plt.figure(4)
plt.title('Original Image with Bondline and Detected Voids   '+"%.0f" % np.round(bondratio*100,1)+'% Bonded'+' (white=void,
black=bonded)')
plt.imshow(bondoverlay, cmap='Greys_r')
diff=cv2.Sobel(blursmall, cv2.CV_64F, 1, 0, ksize=s_mask)
diff=np.abs(diff[bllocy,bllocx])
a=np.core.defchararray.join('Original Image with Bondline and Detected Voids (white=void, black=bonded) ',np.str(bondratio))
#bondoverlay[370:425,1000:1156]=scale  Ramulu Microscope
if fname[-2]=='F':
   locname='Face '
   #sect=fname[-8]
   bondoverlay[650:687,1750:1952]=scale #MSE Microscope
#    bondoverlay[800:800+60,2250:2250+304]=scalemstl #MSTL Microscope
if fname[-2]=='C':
   locname='Column '
   #sect=fname[-7]
   bondoverlay[650:687,1750:1952]=scale  #MSE Microscope
#    bondoverlay[800:800+60,2150:2150+304]=scalemstl #MSTL Microscope
if fname[-1]=='B':
   loc2='Bottom '
if fname[-1]=='T':
   loc2='Top '
sect=fname[-4:-2]

bondoverlay2=np.copy(bondoverlay)
bondoverlay2[bllocy,bllocx]=img[bllocy,bllocx]
write=1
if write==1:

   cv2.imwrite(flocation+fname+'_bond'+ftype,bondoverlay)

   cv2.imwrite(flocation+fname+'_nobond'+ftype,bondoverlay2)

   Panelname=fname[0:-4]
   #
   import PIL
   from PIL import ImageFont
   from PIL import Image
   from PIL import ImageDraw
   img2=Image.open(flocation+fname+'_bond'+ftype)
   img3=Image.open(flocation+fname+'_nobond'+ftype)
   font = ImageFont.truetype("calibri.ttf",70)
   draw = ImageDraw.Draw(img2)
```

```
    draw.text((30, 80),"%.0f" % np.round(bondratio*100,1)+'% Bonded'+' (white=void, black=bonded)',1,font=font)
    draw.text((30, 10),'Panel:'+Panelname+' Section:'+sect+' Location:'+loc2+locname,1,font=font)
    font = ImageFont.truetype("calibri.ttf",20)
    draw.text((30,                150),'threshratio:'+np.str(Threshold_ratio)+'                threshvalue:'+np.str(Threshold_value)+'
blursmall:'+np.str(blursmallvalue),1,font=font)
    #draw.text((30, 80),'Top:TIMET 54M   Bot:TIMET 54M',255,font=font)
    #draw.text((30, 80),'Top:TIMET 54MX   Bot:TIMET 54MX',255,font=font)
    #draw.text((30, 80),'Top:ATI 425   Bot:ATI 425',255,font=font)
    #draw.text((30, 80),'Top:VSMPO 64FG   Bot:VSMPO 64FG',255,font=font)
    #draw.text((30, 80),'Top:VSMPO 64FG   Bot:TIMET 64SG',255,font=font)
    #draw.text((30, 80),'Top:TIMET 64SG   Bot:VSMPO 64FG',255,font=font)
    #draw.text((30, 80),'Top:RTI 64FG   Bot:RTI 64FG',255,font=font)
    #draw.text((30, 80),'Top:RTI 64FG   Bot:Beta21-S',255,font=font)
    #draw.text((30, 80),'Top:Ti 6242   Bot:RTI 64FG',255,font=font)
    #draw.text((1025, 435),'Mag 50x',255,font=font)
    draw = ImageDraw.Draw(img3)
    font = ImageFont.truetype("calibri.ttf",70)
    draw.text((30, 80),"%.0f" % np.round(bondratio*100,1)+'% Bonded',1,font=font)
    draw.text((30, 10),'Panel:'+Panelname+' Section:'+sect+' Location:'+loc2+locname,1,font=font)
    draw = ImageDraw.Draw(img3)
    img2.save(flocation+fname+'_bond'+ftype)
    img3.save(flocation+fname+'_nobond'+ftype)

voiddif=voids[1:]-voids[0:voids.size-1]
voiddif=np.hstack([voiddif,10.])
voidedgeloc=np.linspace(1,voiddif.size,voiddif.size)[voiddif>1]
voidsizes=(voidedgeloc-np.hstack([1,voidedgeloc[0:-1]]))*pixelsize
voidnumber=voidsizes.size
voidstd=np.std
np.savetxt(flocation+fname+'_voids.csv', voidsizes, delimiter=",")
#voidnum=1
#voidsize=1
#for i in range(1,voiddif.size):
#    if voiddif[i]==1:
#        voidsize=voidsize+
#    voiddif[i]
#micrographlist=open(flocation+'micrographlist.txt','rt')



#plt.imshow(img, cmap='Greys_r')
#image,contours,hierarchy=cv2.findContours(canny,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
#contplot=cv2.drawContours(img, contours, -1, (0,255,0), 3)
#rows,cols = thresh.shape[:2]
#vx,vy,x,y = cv2.fitLine(thresh, cv2.DIST_L2,0,0.01,0.01)
#lefty = int((-x*vy/vx) + y)
#righty = int(((cols-x)*vy/vx)+y)
#contplotline = cv2.line(img,(cols-1,righty),(0,lefty),(0,255,0),2)
#plt.imshow(contplot, cmap='Greys_r')
```

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 19 13:51:57 2016

@author: BJF
"""

import numpy as np
import numpy.matlib as M
import math
import pylab
from scipy import ndimage
import matplotlib.pyplot as plt
from PIL import Image
import cv2

plt.close(1)
plt.close(2)
plt.close(3)
fname='F24S1'
floc='C:\\Users\\BJF\\OneDrive for Business\\BA18_5\\MacroImages\\'
imgo = ndimage.imread(floc+fname+'.jpg')
imgcp = ndimage.imread(floc+fname+'.jpg')
fig = plt.figure()
img = cv2.cvtColor(imgo,cv2.COLOR_RGB2GRAY)
imgcp = cv2.cvtColor(imgcp,cv2.COLOR_RGB2GRAY)
blrthrs=[[55,110],[55,110],[55,110],[55,110]]
#pylab.imshow(img)
#pylab.gray()
#pylab.show()

#img=img.astype('uint8')
#imggf = mh.gaussian_filter(img, 1)
#imggf = np.array(imggf, dtype=np.uint8)
#Thres = mh.thresholding.otsu(imggf)
#dx=ndimage.prewitt(img,0)
#dy=ndimage.prewitt(img,1)
#mag=np.hypot(dx,dy)
#mag = mag*255/np.max(mag)
#mag=mag.astype('uint8')
#mag2=ndimage.watershed_ift(img)
#plt.imshow(img)
#plt.gray()
#plt.show()
blur=cv2.medianBlur(img,35)
#blur=blur[img.shape[0]*.1:img.shape[0]*.9,:]
#img=img[img.shape[0]*.1:img.shape[0]*.9,:]
#imgo=imgo[imgo.shape[0]*.1:imgo.shape[0]*.9,:]
s_mask=3
ret,thresh = cv2.threshold(blur,10,255,cv2.THRESH_BINARY) #np.mean(blur)*1.2
canny=cv2.Canny(thresh,10,30)
sobel0=np.abs(cv2.Sobel(blur, cv2.CV_64F, 0, 1, ksize=s_mask))
sobel1=np.abs(cv2.Sobel(blur, cv2.CV_64F, 1, 0, ksize=s_mask))
sobel=np.hypot(sobel1,sobel0)
sobel*=255.0/np.max(sobel)

#lines=cv2.HoughLinesP(canny, 1, np.pi/180, 200,lines,200,50)
#
#for i in range(0,lines.shape[0]):
```

```
#    plt.plot([lines[i,0,0],lines[i,0,2]],[lines[i,0,1],lines[i,0,3]])



#fig = plt.figure(figsize=(250.0, 77.2),dpi=10)
#res=1/2.5
res=7.2/25.4
stx=blur.shape[1]/10
sty=blur.shape[1]*9/10
ymax=blur.shape[0]*9/10
ymin=blur.shape[0]/10
thick=np.zeros((ymax-ymin,2))
xdim=img.shape[1]
plt.figure(1)
plt.imshow(canny, cmap='Greys_r')
UB=np.ones((img.shape[1],2))
Seed=np.array([[img.shape[0]/2,img.shape[1]/4]])
#walk right
#xy=np.copy(Seed)
#UBnd=sobel[0:xy[0,0],xy[0,1]]
#UBmax=np.argmax(UBnd)

#UB[0,:]=[xy[0,1],xy[0,0]+UBmax]

#LB=sobel[xy[0,0]:-1,xy[1,0]]
#for i in range(0,3000):
#    LB=sobel[xy[1,0]]
#Average=np.argmax(canny)
cannybin=canny/255
locy,locx=np.mgrid[0:img.shape[0],0:img.shape[1]]
linelocy=locy*cannybin
linelocx=locx*cannybin
plt.imshow(canny, cmap='Greys_r')



LBnd=np.max(linelocy,0)
Ubs=np.copy(linelocy)
Ubs[Ubs==0]=img.shape[0]-1
UBnd=np.min(Ubs,0)
#for i in range(300,img.shape[1]): #img.shape[1]
#    diffy=np.abs(LBnd[i]-LBnd[i-2])
#    diff=np.sqrt(diffy^2+1)
#    if diff>2:
#        opt=linelocy[:,i]
#        optdiff=opt-LBnd[i-2]
#        #optdiff=np.vstack((np.abs(optdiff),optdiff/np.abs(optdiff)))
#        best=opt[np.argmin(np.abs(optdiff))]
#        LBnd[i]=best
##
#for i in range(300,img.shape[1]):
#    diffy=np.abs(UBnd[i]-UBnd[i-1])
#    diff=np.sqrt(diffy^2+1)
#    if diff>2:
#        opt=linelocy[:,i]
#        optdiff=opt-UBnd[i-3]
#        #optdiff=np.vstack((np.abs(optdiff),optdiff/np.abs(optdiff)))
#        best=opt[np.argmin(np.abs(optdiff))]
#        UBnd[i]=best
#
```

```
#LBnd[LBnd<50]=img.shape[0]-1
UBnd[UBnd>.75*img.shape[0]]=0
middle=(LBnd+UBnd)/2
LBnd2=linelocy-middle
LBnd2[LBnd2<0]=199
LBnd2=np.min(LBnd2,0)+middle
LBnd2=np.int32(LBnd2)

UBnd2=middle-linelocy
UBnd2[UBnd2>0]=-199
UBnd2=np.max(UBnd2,0)+middle
UBnd2=np.int32(UBnd2)


#LBnd=np.vstack((LBnd,locx[0,:].T))
thickness=np.abs(UBnd-LBnd)
imgcp[LBnd,locx[0,:]]=255
imgcp[LBnd-1,locx[0,:]]=0
imgcp[LBnd-2,locx[0,:]]=255
imgcp[UBnd,locx[0,:]]=255
imgcp[UBnd-1,locx[0,:]]=0
imgcp[UBnd-2,locx[0,:]]=255
thickplot=np.ones((400,img.shape[1]))*255
thickplot[399-thickness,locx[0,:]]=0
thickplot[398-thickness,locx[0,:]]=0
thickplot[397-thickness,locx[0,:]]=0
thickplot[396-thickness,locx[0,:]]=0
scale=ndimage.imread('C:\\Users\\BJF\\OneDrive for Business\\BA18_1\\scalelarge.png')
scale=cv2.cvtColor(scale,cv2.COLOR_RGB2GRAY)
thickplot[0:400,0:150]=scale
plt.figure(2)
plt.imshow(np.vstack((imgcp,thickplot)), cmap='Greys_r')
cv2.imwrite(floc+fname+'thplot.png',np.vstack((imgcp,thickplot)))

#plt.plot(thickness)
plt.figure(3)
plt.xlabel('Length (mil)')
plt.ylabel('Thickness (mil)')
plt.title(fname[:-5]+' Image Calculated Thickness')

plt.axis([0, img.shape[1]*res, 0*res, img.shape[0]*res])
plt.grid(True)
plt.plot(locx[0,:]*res,thickness*res)

import PIL
from PIL import ImageFont
from PIL import Image
from PIL import ImageDraw
img=Image.open(floc+fname+'thplot.png')
font = ImageFont.truetype("calibri.ttf",70)
draw = ImageDraw.Draw(img)

#draw.text((300, 10),'Panel:4 Section:A',255,font=font)
img.save(floc+fname+'thplot.png')

lengthx=np.linspace(0,UBnd2.shape[0],UBnd2.shape[0])*res
lengthx=np.reshape(lengthx,(UBnd2.shape[0],1))
Ubound=imgcp.shape[0]*res-UBnd*res
Ubound=np.reshape(Ubound,(UBnd2.shape[0],1))
Lbound=imgcp.shape[0]*res-LBnd*res
Lbound=np.reshape(Lbound,(UBnd2.shape[0],1))
thick=thickness*res
```

```
thick=np.reshape(thick,(UBnd2.shape[0],1))
difference=np.diff(thick,0,1)
transition=np.greater(difference,60)
line=np.reshape(np.arange(np.size(thick)),(np.size(thick),1))
transition=transition*1
transitionloc=transition*line
LB=np.max(transitionloc[0:350,0])
UBannoy=transitionloc[np.size(thick)-350:-1,0]
UBannoy[UBannoy==0]=1e8
UB=np.min(transitionloc[np.size(thick)-350:-1,0])
for i in range(0,np.size(thick)):
    if i>UB:
        thick[i,0]=0
        Ubound[i,0]=0
        Lbound[i,0]=0
    if i<LB:
        thick[i,0]=0
        Ubound[i,0]=0
        Lbound[i,0]=0
    if thick[i,0]==0:
      Ubound[i,0]=0
      Lbound[i,0]=0

output=np.concatenate((lengthx,Ubound,Lbound,thick),1)
np.savetxt(floc+fname+'.csv',output,delimiter=',')

#T = mh.thresholding.rc(img)
#pylab.imshow(img > T)
#pylab.show()

#imgf = mh.gaussian_filter(img, 1.08)
#imgf = np.array(imgf, dtype=np.uint8)
#T = mh.thresholding.otsu(imgf)
#pylab.imshow(imgf>T)
#pylab.show()
#r2=ndimage.binary_dilation(r,iterations=1)
#pylab.imshow(r2)
#pylab.show()

#imgreg=mh.borders(mag)
#pylab.imshow(imgreg)
#pylab.show()

#fimg=pylab.fft2(img)
#fimg[np.real(fimg)<100]=0
#imgft=pylab.ifft2(fimg)
#imgft=np.array(imgft,dtype=np.uint8)
#pylab.imshow(fimg)
#pylab.show()

#fc = mh.wavelet_center(img)
#t = mh.haar(fc, 'D8')
#r = mh.ihaar(fc, 'D8')
#rd = mh.wavelet_decenter(r, fc.shape)
#pylab.imshow(t)
#pylab.show()
```

```
# -*- coding: utf-8 -*-
"""
Created on Fri Jan 20 11:00:24 2017

@author: BJF
"""

import numpy as np
import numpy.matlib as M
import math
import pylab
from scipy import ndimage
import matplotlib.pyplot as plt
from PIL import Image
import cv2

holes=np.ones((1,7))
threshmat=np.genfromtxt('C:\\Users\\BJF\\OneDrive for Business\\Chad\\HoleImages\\Threshmat.csv',delimiter=',')
plt.close(1)

for i in range(0,threshmat.shape[0]):
    panel=threshmat[i,0]
    inum=threshmat[i,1]
    img = ndimage.imread('C:\\Users\\BJF\\OneDrive for Business\\Chad\\HoleImages\\'+'F'+str(int(panel))+'-
'+str(int(inum))+'.jpg')
    #plt.imshow(img)

    thresh=threshmat[threshmat[:,0]==panel,:]
    thresh=thresh[thresh[:,1]==inum,:]
    hsv=cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
    mask=cv2.inRange(hsv,thresh[0,2:5],thresh[0,5:8])
    plt.imshow(mask,cmap='Greys_r')

    blur=cv2.medianBlur(mask,15)
    blur=np.uint8(blur)
    dist_transform = cv2.distanceTransform(blur,cv2.DIST_L2,5)
    ret, sure_fg = cv2.threshold(dist_transform,0.7*dist_transform.max(),255,0)
    sure_fg = np.uint8(sure_fg)
    ret, markers = cv2.connectedComponents(blur)
    markers = markers+1
    [locy,locx]=np.mgrid[0:markers.shape[0],0:markers.shape[1]]
    if np.min(locy[markers==3])>np.max(locy[markers==2]):
        scale=np.min(locy[markers==3])-np.min(locy[markers==2])
    if np.min(locy[markers==3])<np.max(locy[markers==2]):
        scale=np.min(locy[markers==2])-np.min(locy[markers==3])
    if scale==0:
        if np.min(locy[markers==4])>np.max(locy[markers==3]):
            scale=np.min(locy[markers==4])-np.min(locy[markers==3])
        if np.min(locy[markers==4])<np.max(locy[markers==3]):
            scale=np.min(locy[markers==3])-np.min(locy[markers==4])
    measuredweld=threshmat[i,-1]*1e3
    pixeldist=measuredweld/scale

    #plt.imshow(markers)
    for j in range(2,np.max(markers)+1):
        area=np.sum(markers[markers==j])/j*pixeldist**2
        length=(np.max(locy[markers==j])-np.min(locy[markers==j]))*pixeldist
        width=(np.max(locx[markers==j])-np.min(locx[markers==j]))*pixeldist
        aspect=length/width
```

```
        hole=np.array([[panel,inum,j,area,aspect,length,width]])
        holes=np.vstack([holes,hole])
holes=holes[1:,:]
np.savetxt('C:\\Users\\BJF\\OneDrive for Business\\Chad\\HoleImages\\holes.csv',holes,delimiter=',')

areaavgtot=np.average(holes[:,3])
aspectavgtot=np.average(holes[:,4])
areastdtot=np.std(holes[:,3])
aspectstdtot=np.std(holes[:,4])
F6=holes[holes[:,0]==6,:]
F6vals=np.array([[np.average(F6[:,3]),np.std(F6[:,3]),np.average(F6[:,4]),np.std(F6[:,4])]])
F9=holes[holes[:,0]==9,:]
F9vals=np.array([[np.average(F9[:,3]),np.std(F9[:,3]),np.average(F9[:,4]),np.std(F9[:,4])]])
F14=holes[holes[:,0]==14,:]
F14vals=np.array([[np.average(F14[:,3]),np.std(F14[:,3]),np.average(F14[:,4]),np.std(F14[:,4])]])
F15=holes[holes[:,0]==15,:]
F15vals=np.array([[np.average(F15[:,3]),np.std(F15[:,3]),np.average(F15[:,4]),np.std(F15[:,4])]])
F22=holes[holes[:,0]==22,:]
F22vals=np.array([[np.average(F22[:,3]),np.std(F22[:,3]),np.average(F22[:,4]),np.std(F22[:,4])]])
F23=holes[holes[:,0]==23,:]
F23vals=np.array([[np.average(F23[:,3]),np.std(F23[:,3]),np.average(F23[:,4]),np.std(F23[:,4])]])



#plt.hist(F9[:,3], 70, facecolor='g')
```

2D Profile Overlap

```
# -*- coding: utf-8 -*-
"""
Created on Sun Feb 19 22:03:27 2017

@author: BJF
"""
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import expon,halfnorm
from numpy import linspace
from pylab import plot,show,hist,figure,title

plt.close(1)
plt.close(2)
surfneha=np.genfromtxt('C:\\Users\\BJF\\OneDrive for Business\\BA18_2\\SurfaceRoughness\\SurfNeha.csv',delimiter=',')
points=(surfneha.shape[0]-1)
surf=surfneha[1::,:]-np.tile(np.average(surfneha[1:11,:],0),(points,1))-(np.average(surfneha[-11:-1,:],0)-
np.average(surfneha[1:11,:],0))/(points)*np.tile(np.linspace(1,points,points),(surfneha.shape[1],1)).T
#plt.plot(surf[:,0:6]+4*np.tile(np.array([[0,1,2,3,4,5]]),(surf.shape[0],1)))


Tc=773.;
Pr=3.;
T=Tc+273.;
Tm=1973.;
ga300=4.3E4;
gb300=2.05e4;
td_ga=-1.2;
td_gb=-0.5;
G_a=ga300*(1+((T-300)/Tm)*(td_ga));
G_b=gb300*(1+((T-300)/Tm)*(td_gb));
bt=0.15;
al=1-bt;
G=al*G_a+bt*G_b;
G1=6.2E4*(1+((T-300)/Tm)*(1-1.2));
sig_y=1E-3*G;
fo=Pr/sig_y;
voidnum=1
voiddbase=np.zeros((1,3))
surf2mat=2
surf1mat=1
for l in range(0,6):
    surf1=l+surf1mat*6
    for k in range(0,6):
        surf2=k+surf2mat*6
        for n in range(0,10):
            disp=n*points/10
            dif=0
            inc=.01
            for i in range(0,1000):
                olap=surf[:,0]+dif-np.hstack((surf[disp::,surf2],surf[:disp,surf2]))
                foth=olap[olap<0].shape[0]*1.0/points
                if foth>fo:
                    dif=dif+inc
                if foth<fo:
                    dif=dif-inc
                    inc=inc/2
                if np.abs(foth-fo)<=1e-4:
                    break
```

```
            voids=np.copy(olap)
            voids[voids<0]=0.0
            void=np.array([[0.]])
            for j in range(0,points):
                if voids[j]>0:
                    void=np.vstack((void,voids[j]))
                if voids[j]==0:
                    if void.shape[0]>1:
                        void =void[1::,:]
                        voidwidth=void.shape[0]*.5
                        voidheight=np.average(void)
                        voiddbase=np.vstack((voiddbase,np.array([[voidnum,voidheight,voidwidth]])))
                        voidnum=voidnum+1
                        void=np.array([[0.]])

heights = expon.fit(voiddbase[:,1]) # distribution fitting
x = linspace(0,2.5,100)
pdf_fittedw = expon.pdf(x,loc=heights[0],scale=heights[1])
mean1, var1, skew, kurt = expon.stats(loc=heights[0],scale=heights[1],moments='mvsk')
plt.figure(1)
title('Void Height - Exponential Distribution Fit')
plot(x,pdf_fittedw,'r-')
plt.xlim([0,2.5])
hist(voiddbase[:,1],bins=20,normed=True,alpha=.3)

voidwidths=np.copy(voiddbase[:,2])
voidwidths=voidwidths[voidwidths<200]
widths = expon.fit(voidwidths) # distribution fitting
mean2, var2, skew, kurt = expon.stats(loc=widths[0],scale=widths[1],moments='mvsk')
x = linspace(0,200,100)
pdf_fittedh = expon.pdf(x,loc=widths[0],scale=widths[1])
plt.figure(2)
title('Void Width - Exponential Distribution Fit')
plot(x,pdf_fittedh,'r-')
hist(voidwidths,bins=50,normed=1,alpha=.3)
show()
#plt.plot(surf)
```

```
import odbAccess
import numpy as np
names='ti64to64s'
path='C:\\cygwin64\\home\\BJF\\'
nameOfStep='Step-2'
NameOfFile=path+names+'.dat'
Name=path+names+'.odb'
myOdb = odbAccess.openOdb(path=Name)
lastStep=myOdb.steps[nameOfStep]
z=43
lastFrame =myOdb.steps[nameOfStep].frames[z]
nuset= myOdb.rootAssembly.nodeSets['Uppernodes'.upper()]
nlset= myOdb.rootAssembly.nodeSets['Lowernodes'.upper()]
ntop=lastFrame.fieldOutputs['COORD'].getSubset(region=nuset)
nbot=lastFrame.fieldOutputs['COORD'].getSubset(region=nlset)
i=0
y=np.ones((4480/2,1))
x=np.ones((4480/2,1))
for v in range(0,len((y))*2):
    if ntop.values[v].data[2]<0:
        y[i]=ntop.values[v].data[1]#-nbot.values[v].data[1]
        x[i]=ntop.values[v].data[0]
        i=i+1
np.savetxt(path+names+'top'+'.dat',np.hstack((x,y)), delimiter=',')
i=0
y=np.ones((4480/2,1))
x=np.ones((4480/2,1))
for k in range(0,len((x))*2):
    if nbot.values[k].data[2]<0:
        y[i]=nbot.values[k].data[1]#-nbot.values[v].data[1]
        x[i]=nbot.values[k].data[0]
        i=i+1
np.savetxt(path+names+'bot'+'.dat',np.hstack((x,y)), delimiter=',')



# -*- coding: utf-8 -*-
"""
Created on Thu Mar 09 17:11:50 2017

@author: BJF
"""

import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import expon
from numpy import linspace
from pylab import plot,show,hist,figure,title
plt.close(1)
plt.close(2)
name='ti54mto54m_1s'
#olap=np.genfromtxt('C:\\cygwin64\\home\\BJF\\ti54mto54m_1s.dat',delimiter=',')
#nodeloc=np.genfromtxt('C:\\cygwin64\\home\\BJF\\ti64to64_1snodes.dat',delimiter='\t')
nodetop=np.genfromtxt('C:\\cygwin64\\home\\BJF\\'+name+'top.dat',delimiter=',')
#nodebot=np.copy(nodetop)
nodebot=np.genfromtxt('C:\\cygwin64\\home\\BJF\\'+name+'bot.dat',delimiter=',')
nodetop=nodetop[:-1,:]
nodebot=nodebot[:-1,:]
#nodetopvals=np.hstack[nodeloc[nodeloc[:,0]==nodetop],nodeval[nodeloc[:,0]==nodetop]]
```

```
#nodebotvals=np.hstack[nodeloc[nodeloc[:,0]==nodebot],nodeval[nodeloc[:,0]==nodebot]]
#
#ntopv=nodetopvals[np.argsort(nodetopvals[:,1],axis=0),:]
#nbotv=nodebotvals[np.argsort(nodebotvals[:,1],axis=0),:]
olap=(nodetop[:,1]-nodebot[:,1])
olap[olap<.00001]=0
points=len(olap)
voidnum=1
voiddbase=np.zeros((1,3))
voids=np.copy(olap)
void=np.array([[0.]])
for j in range(0,points):
    if voids[j]>0:
        void=np.vstack((void,voids[j]))
    if voids[j]==0:
        if void.shape[0]>1:
            void =void[1::,:]
            voidwidth=void.shape[0]*.5e-3
            voidheight=np.average(void)
            voiddbase=np.vstack((voiddbase,np.array([[voidnum,voidheight,voidwidth]])))
            voidnum=voidnum+1
            void=np.array([[0.]])

heights = expon.fit(voiddbase[:,1]) # distribution fitting
x = linspace(0.00012,np.max(voiddbase[:,1]),100)
pdf_fittedw = expon.pdf(x,loc=heights[0],scale=heights[1])
mean1, var1, skew, kurt = expon.stats(loc=heights[0],scale=heights[1],moments='mvsk')
plt.figure(1)
title('Ti64 to Ti64 Void Height')
plt.ylabel('PDF')
plt.xlabel('Bins (mm)')
plot(x,pdf_fittedw,'r-')
plt.xlim([0,np.max(voiddbase[:,1])])
hist(voiddbase[:,1],bins=5,normed=True,alpha=.3)

voidwidths=np.copy(voiddbase[:,2])
#voidwidths=voidwidths[voidwidths<200]
widths = expon.fit(voidwidths) # distribution fitting
mean2, var2, skew, kurt = expon.stats(loc=widths[0],scale=widths[1],moments='mvsk')
x = linspace(0,np.max(voiddbase[:,2]),100)
pdf_fittedh = expon.pdf(x,loc=widths[0],scale=widths[1])
plt.figure(2)
title('Ti64 to Ti64 Void Width')
plt.ylabel('PDF')
plt.xlabel('Bins (mm)')
plot(x,pdf_fittedh,'r-')
plt.xlim([0,np.max(voiddbase[:,2])])
hist(voidwidths,bins=11,normed='True',alpha=.3)
show()
```

```
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 27 13:18:10 2016

@author: BJF
"""

import numpy as np
import numpy.matlib as M
import math
import pylab
import mahotas as mh
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from scipy import ndimage
from scipy import interpolate as interp
from matplotlib import cm
import cv2
#plt.close(fig1)
#plt.close(fig2)
#fig1 = plt.figure()
#fig2 = plt.figure()
#ax1 = Axes3D(fig1)
#ax2 = Axes3D(fig2)
pi=np.pi
n=200
l=4*pi
ni=1
voidvol=np.zeros((ni,3))
#MCvoid=np.zeros((100,1))
for i in range(0,ni):

    x1,y1=np.mgrid[-l:l:n*1j,-l:l:n*1j]
    r1=np.sqrt(x1**2+y1**2)
#z1=np.zeros((100,100))
#z1=(np.sin(r1)+np.cos(r1))*np.exp(-r1)
    z1=5*(np.sin(x1))/l+np.random.randn(n,n)/100



    x2,y2=np.mgrid[-l:l:n*1j,-l:l:n*1j]
    #r2=np.sqrt((x2+1.5)**2+y2**2)
#z2=(-np.sin(r2)-np.cos(r2))*np.exp(-r2)+2
    z2=(-5*np.sin(x2)-5*np.sin(y2))/l+3 #
    shiftx=np.random.randn()*n*.3
    shifty=np.random.randn()*n*.3
    shiftx=np.abs(np.int(shiftx))
    shifty=np.abs(np.int(shifty))
    z2y=np.vstack((z2[shifty:n,:],z2[0:shifty,:]))
    z2n=np.hstack((z2y[:,shiftx:n],z2y[:,0:shiftx]))
#z2y=np.vstack((z2[shifty:n,:],np.zeros((n-shifty,n))))
#z2n=np.hstack((z2y[:,shiftx:n],np.zeros((n,n-shiftx))))
    z3=z2n-z1
    close=np.min(z3)
    closeid=np.argmin(z3,axis=1)
    closeid2=np.argmin(closeid)

    void=z2n-close-z1
    voidvol[i,0]=shiftx
```

```
    voidvol[i,1]=shifty
    voidvol[i,2]=np.sum(void[n*3/10:n*7/10,n*3/10:n*7/10])
    contact=(np.max(void)-np.min(void))*.5
    MCvoid=z2n-close-z1-contact

MCvoid[MCvoid<0]=0


#plt.pcolor(MCvoid)
#plt.colorbar()
#plt.title('Difference Between Surfaces After 50% Overlap')
#plt.xlabel('X')
#plt.ylabel('Y')
MCdiff=np.abs(np.diff(MCvoid))
MCedges=np.copy(MCvoid)
MCedges[MCedges>0]=255

#MCedges=np.abs(np.diff(MCedges))
#plt.title('Edges of 50% Overlap')
#plt.xlabel('X')
#plt.ylabel('Y')
#plt.pcolor(MCedges)
#plt.colorbar()

sure_bg=MCedges[MCedges<10]
sure_fg=MCedges[MCedges>10]
MCedges=np.uint8(MCedges)
dist_transform = cv2.distanceTransform(MCedges,cv2.DIST_L2,5)
ret, sure_fg = cv2.threshold(dist_transform,0.7*dist_transform.max(),255,0)
sure_fg = np.uint8(sure_fg)
ret, markers = cv2.connectedComponents(MCedges)
markers = markers+1
plt.title('Connected Voids')
plt.pcolor(markers)
```

3D Meshing Algorithm

```
#totalelem=(n-1)*2*(n-1)
#d1=0
#d0=-.1
#r=.9
#ncount=0
#ecount=0
#nodes=np.array([[0,0,0,0]])
#elem=np.zeros((1,16))
#for k in range(0,5):
#    d1=d0-.1*k
#    for i in range(0,n-1):
#        for j in range(0,n-1):
#            xa,ya,za=x1[i,j],y1[i,j],z1[i,j]+d0
#            xb,yb,zb=x1[i+1,j],y1[i+1,j],z1[i+1,j]+d0
#            xc,yc,zc=x1[i+1,j+1],y1[i+1,j+1],z1[i+1,j+1]+d0
#            xd,yd,zd=x1[i,j+1],y1[i,j+1],z1[i,j+1]+d0
##            zavg=np.mean((za,zb,zc,zd))
##            za,zb,zc,zd=za-(zavg-za)*k*r,zb-(zavg-zb)*k*r,zc-(zavg-zc)*k*r,zd-(zavg-zd)*k*r
#                    nodes1=np.array([[1,xa,ya,za+d1],[2,xc,yc,zc+d1],[3,xd,yd,zd+d1],[4,xa,ya,za],[5,xc,yc,zc],[6,xd,yd,zd],[7,xa+(xc-
xa)*.5,ya+(yc-ya)*.5,za+(zc-za)*.5+d1],[8,xa+(xc-xa)*.5,yd,zd+(zc-zd)*.5+d1],[9,xa,ya+(yd-ya)*.5,za+(zd-
za)*.5+d1],[10,xa+(xc-xa)*.5,ya+(yc-ya)*.5,za+(zc-za)*.5],[11,xa+(xc-xa)*.5,yd,zd+(zc-zd)*.5],[12,xa,ya+(yd-ya)*.5,za+(zd-
za)*.5],[13,xa,ya,za+d1/2],[14,xc,yc,zc+d1/2],[15,xd,yd,zd+d1/2],[16,xb,yb,zb+d1],[17,xb,yb,zb],[18,xb,yb+(yc-yb)*.5,zb+(zc-
zb)*.5+d1],[19,xa+(xb-xa)*.5,ya,za+(zb-za)*.5+d1],[20,xb,yb+(yc-yb)*.5,zb+(zc-zb)*.5],[21,xa+(xb-xa)*.5,yb,za+(zb-
za)*.5],[22,xb,yb,zb+d1/2]])
```

```
#           nodes1[:,0]=nodes1[:,0]+ncount
#           elem1=np.array([[1,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15],[2,1,16,2,4,17,5,19,18,7,21,20,10,13,22,14]])
#           elem1[:,0]=elem1[:,0]+ecount
#           elem1[:,1:16]=elem1[:,1:16]+ncount
#           elem=np.vstack((elem,elem1))
#           nodes=np.vstack((nodes,nodes1))
#           ncount=ncount+22
#           ecount=ecount+2
#    d0=d1+d0
#nbot=ncount
#ebot=ecount
#d1=0
#d0=.1
#z2=np.copy(z2n-close)
#for k in range(0,5):
#    d1=d0+.1*k
#    for i in range(0,n-1):
#        for j in range(0,n-1):
#           xa,ya,za=x2[i,j],y2[i,j],z2[i,j]+d0
#           xb,yb,zb=x2[i+1,j],y2[i+1,j],z2[i+1,j]+d0
#           xc,yc,zc=x2[i+1,j+1],y2[i+1,j+1],z2[i+1,j+1]+d0
#           xd,yd,zd=x2[i,j+1],y2[i,j+1],z2[i,j+1]+d0
##            zavg=np.mean((za,zb,zc,zd))
##            za,zb,zc,zd=za+(zavg-za)*k*r,zb+(zavg-zb)*k*r,zc+(zavg-zc)*k*r,zd+(zavg-zd)*k*r
#                nodes1=np.array([[1,xa,ya,za+d1],[2,xc,yc,zc+d1],[3,xd,yd,zd+d1],[4,xa,ya,za],[5,xc,yc,zc],[6,xd,yd,zd],[7,xa+(xc-
xa)*.5,ya+(yc-ya)*.5,za+(zc-za)*.5+d1],[8,xa+(xc-xa)*.5,yd,zd+(zc-zd)*.5+d1],[9,xa,ya+(yd-ya)*.5,za+(zd-
za)*.5+d1],[10,xa+(xc-xa)*.5,ya+(yc-ya)*.5,za+(zc-za)*.5],[11,xa+(xc-xa)*.5,yd,zd+(zc-zd)*.5],[12,xa,ya+(yd-ya)*.5,za+(zd-
za)*.5],[13,xa,ya,za+d1/2],[14,xc,yc,zc+d1/2],[15,xd,yd,zd+d1/2],[16,xb,yb,zb+d1],[17,xb,yb,zb],[18,xb,yb+(yc-yb)*.5,zb+(zc-
zb)*.5+d1],[19,xa+(xb-xa)*.5,ya,za+(zb-za)*.5+d1],[20,xb,yb+(yc-yb)*.5,zb+(zc-zb)*.5],[21,xa+(xb-xa)*.5,yb,za+(zb-
za)*.5],[22,xb,yb,zb+d1/2]])
#           nodes1[:,0]=nodes1[:,0]+ncount
#           elem1=np.array([[1,4,5,6,1,2,3,10,11,12,7,8,9,13,14,15],[2,4,17,5,1,16,2,21,20,10,19,18,7,13,22,14]])
#           elem1[:,0]=elem1[:,0]+ecount
#           elem1[:,1:16]=elem1[:,1:16]+ncount
#           elem=np.vstack((elem,elem1))
#           nodes=np.vstack((nodes,nodes1))
#           ncount=ncount+22
#           ecount=ecount+2
#    d0=d1+d0
#ntop=ncount
#etop=ecount
#a=np.copy(nodes[:,1:4])
#b = np.ascontiguousarray(a).view(np.dtype((np.void, a.dtype.itemsize * a.shape[1])))
#_, idx = np.unique(b, return_index=True)
#_, ivx = np.unique(b, return_inverse=True)
#unique_nodes = nodes[idx]
#rep=idx[ivx]
#elemcolumn=np.copy(elem[:,0])
#for i in range(0,ncount):
#    elem[elem==i]=rep[i]
#elem[:,0]=elemcolumn

#f=open('Interaction.inp','w')
#f.write('\n')
#f.write('*PART, NAME=PartA')
#f.write('\n')
#f.write('\t')
#f.write('*NODE')
#f.write('\n')
#node=1
#for i in range(1,nbot+1):
#    f.write('\t')
```

```
#    f.write(np.str(np.int(unique_nodes[i,0])))
#    f.write(',')
#    f.write(np.str(unique_nodes[i,1]))
#    f.write(',')
#    f.write(np.str(unique_nodes[i,2]))
#    f.write(',')
#    f.write(np.str(unique_nodes[i,3]))
#    f.write('\n')
#
#f.write('\n')
#f.write('\t')
#f.write('*ELEMENT, TYPE=C3D15')
#f.write('\n')
#node=1
#for i in range(1,ebot+1):
#    f.write('\t')
#    for j in range(0,elem.shape[1]):
#        f.write(np.str(np.int(elem[i,j])))
#        f.write(',')
#    f.write('\n')
#
#f.write('*END PART')
#f.write('\n')
#f.write('*PART, NAME=PartB')
#f.write('\n')
#f.write('\t')
#f.write('*NODE')
#f.write('\n')
#node=1
#for i in range(nbot+1,ntop+1):
#    f.write('\t')
#    f.write(np.str(np.int(unique_nodes[i,0])))
#    f.write(',')
#    f.write(np.str(unique_nodes[i,1]))
#    f.write(',')
#    f.write(np.str(unique_nodes[i,2]))
#    f.write(',')
#    f.write(np.str(unique_nodes[i,3]))
#    f.write('\n')
#
#f.write('\n')
#f.write('\t')
#f.write('*ELEMENT, TYPE=C3D15')
#f.write('\n')
#node=1
#for i in range(ebot+1,etop+1):
#    f.write('\t')
#    for j in range(0,elem.shape[1]):
#        f.write(np.str(np.int(elem[i,j])))
#        f.write(',')
#    f.write('\n')
#f.write('*END PART')
##f.write('\t')
#f.write('*ELSET, ELSET=BOT, GENERATE')
#f.write('\n')
#f.write('\t')
#f.write('1,')
#f.write(np.str(ebot))
#f.write(',1')
#f.write('\n')
```

```
#f.write('\n')
#f.write('*PART, NAME=PartB')
#f.write('\n')
#f.write('\t')
#f.write('*ELSET, ELSET=TOP, GENERATE')
#f.write('\n')
#f.write('\t')
#f.write(np.str(ebot+1))
#f.write(',')
#f.write(np.str(etop))
#f.write(',1')
#f.write('\n')
#f.write('*END PART')
#
#f.write('\n')
#f.write('*ELSET, ELSET=TOP, GENERATE')
#f.write('\n')
#f.write(np.str(ebot+1))
#f.write(',')
#f.write(np.str(etop))
#f.write(',1')
#f.write('\n')


f.close()
```

Periodic Boundary Conditions

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 14 07:25:35 2017

@author: BJF
"""
import numpy as np
import interaction

m = mdb.models['Model-1']

names='Eqns'
path='C:\\cygwin64\\home\\BJF\\'
eqns=open(path+names+'.txt','w')
right=m.rootAssembly.sets['botR'].nodes
left=m.rootAssembly.sets['botL'].nodes
nodepair1=np.array([[1,1]])
nodepair2=np.array([[1,1]])
nodepair3=np.array([[1,1]])
nodepair4=np.array([[1,1]])
for v in m.rootAssembly.sets['botR'].nodes:
    rcor=v.coordinates
    nodemin=1000.
    for j in m.rootAssembly.sets['botL'].nodes:
        lcor=j.coordinates
        #diff=rcor-lcor
        vect=np.sqrt((rcor[0]-lcor[0])**2+(rcor[1]-lcor[1])**2+(rcor[2]-lcor[2])**2)
        if vect<nodemin:
            nodemin=np.copy(vect)
            nodemap2=j
    nodemap1=v
    nodepair1=np.vstack((nodepair1,np.array([[int(nodemap1.label),int(nodemap2.label)]])))

right=m.rootAssembly.sets['botU'].nodes
```

```
left=m.rootAssembly.sets['botB'].nodes

for v in m.rootAssembly.sets['botU'].nodes:
    rcor=v.coordinates
    nodemin=1000.
    for j in m.rootAssembly.sets['botB'].nodes:
        lcor=j.coordinates
        #diff=rcor-lcor
        vect=np.sqrt((rcor[0]-lcor[0])**2+(rcor[1]-lcor[1])**2+(rcor[2]-lcor[2])**2)
        if vect<nodemin:
            nodemin=np.copy(vect)
            nodemap2=j
    nodemap1=v
    nodepair2=np.vstack((nodepair1,np.array([[int(nodemap1.label),int(nodemap2.label)]])))


right=m.rootAssembly.sets['topR'].nodes
left=m.rootAssembly.sets['topL'].nodes

for v in m.rootAssembly.sets['topR'].nodes:
    rcor=v.coordinates
    nodemin=1000.
    for j in m.rootAssembly.sets['topL'].nodes:
        lcor=j.coordinates
        #diff=rcor-lcor
        vect=np.sqrt((rcor[0]-lcor[0])**2+(rcor[1]-lcor[1])**2+(rcor[2]-lcor[2])**2)
        if vect<nodemin:
            nodemin=np.copy(vect)
            nodemap2=j
    nodemap1=v
    nodepair3=np.vstack((nodepair1,np.array([[int(nodemap1.label),int(nodemap2.label)]])))
#m.Equation(name='n'+np.str(nodemap1.label)+'_1',terms=((1.0,'nbotR'+np.str(nodemap1.label)+'_'+np.str(nodemap2.label),3),(-
1.0,'ntopR'+np.str(nodemap1.label)+'_'+np.str(nodemap2.label),3)))
right=m.rootAssembly.sets['topU'].nodes
left=m.rootAssembly.sets['topB'].nodes

for v in m.rootAssembly.sets['topU'].nodes:
    rcor=v.coordinates
    nodemin=1000.
    for j in m.rootAssembly.sets['topB'].nodes:
        lcor=j.coordinates
        #diff=rcor-lcor
        vect=np.sqrt((rcor[0]-lcor[0])**2+(rcor[1]-lcor[1])**2+(rcor[2]-lcor[2])**2)
        if vect<nodemin:
            nodemin=np.copy(vect)
            nodemap2=j
    nodemap1=v
    nodepair4=np.vstack((nodepair1,np.array([[int(nodemap1.label),int(nodemap2.label)]])))

nodepair1=nodepair1[1:,:]
nodepair2=nodepair2[1:,:]
nodepair3=nodepair3[1:,:]
nodepair4=nodepair4[1:,:]
#eqns.write(str(nodepair1[100,:]))
eqns.write('*Nset, nset=pair1a, instance=spf-2'+'\n')
for i in range(0,nodepair1.shape[0]):
    eqns.write(str(nodepair1[i,0])+',')
    if (i+1)/16*1.0==(i+1)/16.0:
        eqns.write('\n')
eqns.write('*Nset, nset=pair1b, instance=spf-2'+'\n')
for i in range(0,nodepair1.shape[0]):
    eqns.write(str(nodepair1[i,1])+',')
```

```
    if (i+1)/16*1.0==(i+1)/16.0:
        eqns.write('\n')
eqns.write('\n')
eqns.write('\n')
#
eqns.write('*Nset, nset=pair2a, instance=spf-2'+'\n')
for i in range(0,nodepair2.shape[0]):
    eqns.write(str(nodepair2[i,0])+',')
    if (i+1)/16*1.0==(i+1)/16.0:
        eqns.write('\n')
eqns.write('*Nset, nset=pair2b, instance=spf-2'+'\n')
for i in range(0,nodepair2.shape[0]):
    eqns.write(str(nodepair2[i,0])+',')
    if (i+1)/16*1.0==(i+1)/16.0:
        eqns.write('\n')
eqns.write('\n')
eqns.write('\n')

eqns.write('*Nset, nset=pair3a, instance=spf-1'+'\n')
for i in range(0,nodepair3.shape[0]):
    eqns.write(str(nodepair3[i,0])+',')
    if (i+1)/16*1.0==(i+1)/16.0:
        eqns.write('\n')
eqns.write('*Nset, nset=pair3b, instance=spf-1'+'\n')
for i in range(0,nodepair3.shape[0]):
    eqns.write(str(nodepair3[i,0])+',')
    if (i+1)/16*1.0==(i+1)/16.0:
        eqns.write('\n')
eqns.write('\n')
eqns.write('\n')

eqns.write('*Nset, nset=pair4a, instance=spf-1'+'\n')
for i in range(0,nodepair4.shape[0]):
    eqns.write(str(nodepair4[i,0])+',')
    if (i+1)/16*1.0==(i+1)/16.0:
        eqns.write('\n')
eqns.write('*Nset, nset=pair4b, instance=spf-1'+'\n')
for i in range(0,nodepair4.shape[0]):
    eqns.write(str(nodepair4[i,0])+',')
    if (i+1)/16*1.0==(i+1)/16.0:
        eqns.write('\n')

eqns.write('*Equation'+'\n')
eqns.write('2'+'\n')
eqns.write('pair1a,2,1.0,pair1b,2,-1.0'+'\n')
eqns.write('*Equation'+'\n')
eqns.write('2'+'\n')
eqns.write('pair1a,3,1.0,pair1b,3,-1.0'+'\n')

eqns.write('*Equation'+'\n')
eqns.write('2'+'\n')
eqns.write('pair2a,1,1.0,pair2b,1,-1.0'+'\n')
eqns.write('*Equation'+'\n')
eqns.write('2'+'\n')
eqns.write('pair2a,3,1.0,pair2b,3,-1.0'+'\n')

eqns.write('*Equation'+'\n')
eqns.write('2'+'\n')
eqns.write('pair3a,2,1.0,pair3b,2,-1.0'+'\n')
eqns.write('*Equation'+'\n')
eqns.write('2'+'\n')
eqns.write('pair3a,3,1.0,pair3b,3,-1.0'+'\n')
```

```
eqns.write('*Equation'+'\n')
eqns.write('2'+'\n')
eqns.write('pair4a,1,1.0,pair4b,1,-1.0'+'\n')
eqns.write('*Equation'+'\n')
eqns.write('2'+'\n')
eqns.write('pair4a,3,1.0,pair4b,3,-1.0'+'\n')
eqns.close()
```

Stochastic Pilling

```
import numpy as np
import matplotlib.pyplot as plt

#initial cleanup
plt.close("all")

#import settings from file
#csv file doesn't have headers but is formatted, by column, as material 1, material 2, temperature (c), pressure (MPa), height exponential fit (um), width exponential fit (um)
settings=np.genfromtxt('timet.csv',delimiter=',')
if settings.shape[1]<7: #add a column for bonding percentage if there isn't one in timet.csv
    settings=np.hstack((settings,np.zeros((len(settings),1))))

#program settings
inc=.01 #increment size for stepping through the bonding %
n=1000 #number of random voids to compute per process condition
m=1 #number of times to repeat per process condition
ProCondStart=0 #starting process condition
ProCondEnd=3 #ending process condition =len(settings) for all conditions
plotcdf=0 #plot a CDF? 1 for yes, 0 for no
whichcdf=0 #the cdf to plot

allcdf=np.zeros((n,len(settings)*2))#initialize cdf storage

#begin Pilling program
for l in range(ProCondStart,ProCondEnd):

#read material properties from settings for current process condition
    Pr=settings[l,3] #bonding pressure
    Tc=settings[l,2] #bondint temp (celcius)
    T=Tc+273.
    Tm=1973. #melting tempprogram
    G_a=4.3E4*(1+((T-300)/Tm)*(-1.2)); #alpha shear modulus
    G_b=2.05e4*(1+((T-300)/Tm)*(-0.5)); #beta shear modulus
    av_a=1.76E-29 #alpha atomic volume
    av_b=1.81E-29 #beta atomic volume
    R=8.314 #gas constant
    k=1.38*1E-23 #Boltzmann's constant (J/K)
    Dgbdel_a=3.54E-16*np.exp(-97000/(R*T)) #alpha grain boundary diffusion constant
    Dgbdel_b=5.14E-17*np.exp(-153000/(R*T)) #beta grain boundary diffusion constant
    Dv_a=8.6E-10*np.exp(-150000/(R*T)) #alpha volume diffusion constant
    Dv_b=1.9E-7*np.exp(-153000/(R*T)) #beta volume diffusion constant
    Se=1.7 #suface energy
    A=1.2E-9 #creep constant
    n1=1.355 #creep exponent
    del_a=5.9e-10 #alpha grain boundary width
    del_b=5.72e-10 #beta grain boundary width
    Ac_a=A*Dgbdel_a*((G_a*1e6)/(k*T)) #alpha adjusted creep constant
    Ac_b=A*Dgbdel_b*((G_b*1e6)/(k*T)) #beta adjusted creep constant

    aldat=np.array([.42,.33,.49]) #alpha % array for each material
    ddat=np.array([2.4e-6,4.1e-6,5.9e-6]) #grain size array for each material
    al=aldat[np.int(settings[l,0])] #set current alpha %
    d=ddat[np.int(settings[l,1])] #set current grain size
    bt=1-al #compute beta %
    G=al*G_a+bt*G_b #rule of mixtures shear modulus
    sig_y=1E-3*G #compute yield strength from shear modulus
    fo=Pr/sig_y #overlap fraction from stress balance
    fh=1-fo #unbonded fraction
```

```
constants1=np.array([av_a,av_b,T,Pr,R,k,Dgbdel_a,Dgbdel_b,Dv_a,Dv_b,Se,bt,al,d,Tm,A,n1,del_a,del_b,G_a,G_b,Ac_a,Ac_b])
#constants to pass for material 1

   #redo constants for material 2
   al=aldat[np.int(settings[l,1])] #set different alpha %
   bt=1-al #compute beta
   d=ddat[np.int(settings[l,1])] #set different grain sizelen(settings)
   G=al*G_a+bt*G_b #rule of mixtures shear modulus
   sig_y=1E-3*G #compute yield strength from shear modulus
   fo=Pr/sig_y #overlap fraction from stress balance
   fh2=1-fo #unbonded fraction

constants2=np.array([av_a,av_b,T,Pr,R,k,Dgbdel_a,Dgbdel_b,Dv_a,Dv_b,Se,bt,al,d,Tm,A,n1,del_a,del_b,G_a,G_b,Ac_a,Ac_b])
#constants to pass for material 2


# diffusion equations to apply to bonding
   def diffusion( fh,h,ro,constants): #df/dt equation
      av_a,av_b,T,Pr,R,k,Dgbdel_a,Dgbdel_b,Dv_a,Dv_b,Se,bt,al,d,Tm,A,n1,del_a,del_b,G_a,G_b,Ac_a,Ac_b=constants
#import constants into PD
      ht=h #temporary height
      P=Pr*1e6 #adjust pressure to Pa
      sig_r=((P-2*Se/ro)*((np.sqrt(fh)-1)/(1-fh))) #compute cylinder radial stress
      sig_t=((-P-2*Se/ro)*((np.sqrt(fh)+1)/(1-fh))) #compute cylinder tangential stress
      sig_z=(-P/(1-fh)) #compute cylinder end stress
      sig=(np.sqrt(0.5*((sig_r-sig_t)**2+(sig_t-sig_z)**2+(sig_z-sig_r)**2)))*1e-6 #compute equivalent stress
      edot=al*(Ac_a*(sig/G_a)**n1)+bt*(Ac_b*(sig/G_b)**n1) #compute strain rate
      dfdtp = (-edot/(sig))*((3*P*fh**0.5)-((2*Se*fh**0.5)/ro)+((6*Se)/ro)) #change in bonding due to creep
      dfdtgb_a= ((-2*av_a*Dgbdel_a)/(k*T))*(1/((np.log(1/fh))-((1-fh)/2)))*(1/(ro**2))*((1+(ht/d))/ht)*P #change in bonding due
to alpha grain boundary diffusion
      dfdtgb_b= ((-2*av_b*Dgbdel_b)/(k*T))*(1/((np.log(1/fh))-((1-fh)/2)))*(1/(ro**2))*((1+(ht/d))/ht)*P#change in bonding due
to beta grain boundary diffusion
      dfdtgb=al*dfdtgb_a+bt*dfdtgb_b #rule of mixtures grain boundary diffusion
      dfdtvol_a=(-(4*av_a*Dv_a)/(k*T))*(fh/((np.log(1/fh))-((1-fh)/2)))*((1/(ro)**2))*P #change in bonding due to alpha volume
diffusion
      dfdtvol_b=(-(4*av_b*Dv_b)/(k*T))*(fh/((np.log(1/fh))-((1-fh)/2)))*((1/(ro)**2))*P #change in bonding due to beta volume
diffusion
      dfdtvol=al*dfdtvol_a+bt*dfdtvol_b #rule of mixtures volume diffusion
      dtdf=(-sig/edot)/(dfdtp+dfdtgb+dfdtvol) #relate all processes to time
      return dtdf

#statistically apply bonding equations
   fn=np.zeros((1000000,1)) #initialize bonding
   for j in range(0,m): #repeated cdfs to get averaged
      dbtime=np.zeros((n,1)) #initialize bonding time
      h=np.random.exponential(settings[l,4]*1e-6,(n,1)) #select void height from exponential fit
      ro=np.random.exponential(settings[l,5]*1e-6,(n,1)) #select void width from exponential fit
      for i in range(0,n): #compute n random voids per cdf
         dbtime[i]=0 #start timer at zero
         fn[0]=(fh+fh2)/2 #start bonding at an average of the two materials
         for r in range(0,1000000): #increment overlap with diffusion equations
            dtdf1=diffusion(fn[r],h[i],ro[i],constants1) #first surface change
            dtdf2=diffusion(fn[r],h[i],ro[i],constants2) #second surface change
            fn[r+1]=fn[r]-inc #increment bonding
            if fn[r+1]<0: #finish when bonding is complete
               break
            dbtime[i]=dbtime[i]+(dtdf1+dtdf2)/2*inc #increment bonding time at average of the two materials
      if m==1:
         cdf=np.hstack((np.sort(dbtime/3600,0),np.array([np.linspace(1./n,1,n)]).T)) #array CDF
      else:
         cdf[:,0]=cdf[:,0]+np.sort(dbtime/3600,0).flatten()
```

```
    cdf[:,0]=cdf[:,0]/m #average over repeated cdfs
    cdfwork=np.copy(cdf[cdf[:,0]<=3]) #how many of the voids closed in less than three hours
    if cdfwork.shape[0]==cdf.shape[0]: #compute bonding percentage
        bonding=1.0
    else:
        bonding=cdfwork.shape[0]/cdf.shape[0]

    settings[l,6]=bonding #save bonding percentage
    allcdf[:,2*l:2*l+2]=np.copy(cdf) #save CDF for later
    print(bonding) #print bonding percentage for reference

#export bonding percentages
np.savetxt('timet.csv',settings,delimiter=',')

if plotcdf==1:
    plt.style.use('ggplot')
    materials=np.array((['Ti54M','Ti6242','Ti64SG']))
    plt.title('CDF of '+materials[np.int(settings[whichcdf,0])]+'-'+materials[np.int(settings[whichcdf,1])]+' at
'+settings[whichcdf,2].astype('str')+'C and '+settings[whichcdf,3].astype('str')+'MPa')
    plt.plot(allcdf[:,2*whichcdf],allcdf[:,2*whichcdf+1])
    plt.xlabel('Bonding Time (hours)')
```

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import matplotlib.path as path
import matplotlib.animation as animation
number='4'
name='condition'+number
name='ra2'
resultsu=np.load(name+'u.npy')
resultsl=np.load(name+'l.npy')
scale=1e6
mid=np.int(len(resultsu)/2)
movie=0
#plt.style.use('ggplot')
#time=550 #25, 175 500,550

vol=np.sum(resultsu-resultsl)
print(vol)
btime=resultsl.shape[0]*2
print(btime)
wavelength=np.array([[.25,.5,.75,1,2,3,10,20,100],[.462,.436,.399,.373,.372,.358,.351,.352,.352]])

#yupos=resultsu[time,:]
#ylpos=resultsl[time,:]
#Ldist=yupos.shape[0]
#key=0
#bondst=np.zeros((1,2),dtype=int)
#voidbond=np.ones((Ldist))
#for j in range(1,Ldist-1):
#    if yupos[j]-ylpos[j]==0:
#        voidbond[j]=0
#        if np.abs(yupos[j-1]-ylpos[j-1])>0:
#            bondinit=j
#        if yupos[j+1]-ylpos[j+1]>0:
#            bondend=j
#            key=1
#        if key==1:
#            bondst=np.vstack((bondst,np.array([bondinit,bondend])))
#            key=0
#bondst=bondst[1:,:]
#voids=np.transpose(np.vstack((np.append(np.zeros((1),dtype=int),bondst[:,1]+1),np.append(bondst[:,0]-1,Ldist))))
#contact=np.sum(bondst[:,1]-bondst[:,0])
#bondingt=contact*1.0/Ldist
#print(bondingt)

#
plt.figure(figsize=(10, 5), dpi=100, facecolor='w', edgecolor='k')
line1,=plt.plot(scale*resultsu[0,:],'C0')
line2,=plt.plot(scale*resultsl[0,:],'C0')
##line3,=plt.plot(scale*resultsu[time,:],'C1')
##line4,=plt.plot(scale*resultsl[time,:],'C1')
##for i in range(len(bondst)):
##    if bondst[i,0]>800:
##        if bondst[i,1]<1100:
##            line5,=plt.plot(np.linspace(bondst[i,0],bondst[i,1],bondst[i,1]-bondst[i,0]),scale*resultsl[time,bondst[i,0]:bondst[i,1]],'k')
#
#plt.xlim(900,1250)
plt.ylim(-3,3)
plt.xlim(300,650)
```

```
##line3,=plt.plot(scale*resultsu[mid,:],'C2')
##line4,=plt.plot(scale*resultsl[mid,:],'C2')
#line5,=plt.plot(scale*resultsu[-1,:],'C1')
#line6,=plt.plot(scale*resultsl[-1,:],'C1')
##
##plt.legend([line2, line4, line6], ['Original Voids', 'Voids at t=1.5hrs','Final Voids'])
#plt.legend([line2, line6], ['Original Voids','Final Voids'])
###plt.legend([line2, line4,line5], ['Original Voids','Voids at time t='+np.str(time),'Bonded Sections'])
###plt.legend([line2, line4], ['Original Voids','Voids at time t='+np.str(time)])
#plt.xlabel('Horiontal Distance (um)')
#plt.ylabel('Vertical Height (um)')
###plt.title('Condition '+number)
#plt.title('Condition 4 Bonding Example')
###plt.xlim([1750,1900])
#plt.savefig('timeT'+np.str(time)+'.png')

yupos=resultsu[-1,:]
ylpos=resultsl[-1,:]
Ldist=yupos.shape[0]
key=0
bonds=np.zeros((1,2),dtype=int)
voidbond=np.ones((Ldist))
for j in range(1,Ldist-1):
    if yupos[j]-ylpos[j]==0:
        voidbond[j]=0
        if np.abs(yupos[j-1]-ylpos[j-1])>0:
            bondinit=j
        if yupos[j+1]-ylpos[j+1]>0:
            bondend=j
            key=1
        if key==1:
            bonds=np.vstack((bonds,np.array([bondinit,bondend])))
            key=0
bonds=bonds[1:,:]
voids=np.transpose(np.vstack((np.append(np.zeros((1),dtype=int),bonds[:,1]+1),np.append(bonds[:,0]-1,Ldist))))
contact=np.sum(bonds[:,1]-bonds[:,0])
bonding=contact*1.0/Ldist
#print(bonding)
#void closure example
l1=0
l2=1000
vol1=np.sum((resultsu[0,l1:l2]-resultsl[0,l1:l2])*1e-12)
vol2=np.sum((resultsu[-1,l1:l2]-resultsl[-1,l1:l2])*1e-12)
change=vol2/vol1

surf1=resultsu[:,bonds[0,0]:bonds[-1,1]]-np.average(resultsu)
surf2=resultsl[:,bonds[0,0]:bonds[-1,1]]-np.average(resultsl)

#Rau=np.average(np.abs(surf1),axis=1)
#Ral=np.average(np.abs(surf2),axis=1)
#Rtu=np.max(surf1,axis=1)-np.min(surf1,axis=1)
#Rtl=np.max(surf2,axis=1)-np.min(surf1,axis=1)
#Rta=(Rtu+Rtl)/2
#Rta=Rta-np.min(Rta)
#Rta=Rta[3:]/np.max(Rta[3:])
#Raa=(Ral+Rau)/2
#Raa=Raa-np.min(Raa)
#Raa=Raa[3:]/np.max(Raa[3:])
#V=np.sum(resultsu[:,bonds[0,0]:bonds[-1,1]]-resultsl[:,bonds[0,0]:bonds[-1,1]],axis=1)
#dV=V[1:-1]-V[0:-2]
#dV=-dV[1:]/np.max(-dV[1:])
#x=np.arange(0,98)*3./98
```

```
#val=dV/Rta[3:]
##val=val/np.max(val)
##plt.scatter(x,dV[1:]/np.min(dV[1:]))
##plt.scatter(x,Raa[3:]/np.max(Raa[3:]))\
#fit=np.polyfit(x,val,4)
#p=np.poly1d(fit)
#plt.scatter(x,val)
#plt.plot(x,p(x))
#
if movie==1:
    fig, ax = plt.subplots()
    #

    x=np.linspace(0,len(resultsu[0,:]),len(resultsu[0,:]))
    line, = ax.plot(x,resultsu[0,:]*scale,'b')
    line2, = ax.plot(x,resultsl[0,:]*scale,'b')
    line3, = ax.plot(x,resultsu[0,:]*scale,'g')
    line4, = ax.plot(x,resultsl[0,:]*scale,'g')
    def animate(i):
        line.set_ydata(resultsu[i*5,:]*scale)  # update the data
        line2.set_ydata(resultsl[i*5,:]*scale)
        return line,
    #def init():
    #    line.set_ydata(np.ma.array(x, mask=True))
    #    return line,
    ani = animation.FuncAnimation(fig, animate)
    #plt.xlim(900,1250)
    plt.xlabel('Horiontal Distance (um)')
    plt.ylabel('Vertical Height (um)')
    #plt.title('Condition '+number)
    plt.title('Bonding Progression')
    #plt.ylim(-20e-6,20e-6)
    plt.show()

#Timet Pilling Plots
import numpy as np
import matplotlib.pyplot as plt

settings=np.genfromtxt('timet.csv',delimiter=',')
raw=np.reshape(np.genfromtxt('timetraw.csv',delimiter=','), (54,1))
#settings=np.hstack((settings,np.reshape(raw,(54,1))))


i=3 #select material combination to plot (0 through 5 corresponding to names below)
names=np.array(['Ti-54M to Ti-54M','Ti-6242 to Ti-6242','Ti-64 to Ti-64','Ti-54M to Ti-6242','Ti-54M to Ti-64','Ti-6242 to Ti-
64'])
i=i*9
off=.8
plt.close("all")
fig=plt.figure(1,dpi=120)
plt.style.use('ggplot')
name=names[i/9]
plt.title(name)
plt.ylabel('Bonded Fraction')
plt.ylim([0,1.1])
labels = ('Low Pressure', 'Intermediate Pressure', 'High Pressure')
series1y=raw[0+i:3+i,0]
series2y=raw[3+i:6+i,0]
series3y=raw[6+i:9+i,0]
series4y=settings[0+i:3+i,6]
series5y=settings[3+i:6+i,6]
series6y=settings[6+i:9+i,6]
```

```
series1x=np.array([-off,0,+off])
series2x=np.array([3-off,3,3+off])
series3x=np.array([6-off,6,6+off])
xspots=np.hstack((series1x,series2x,series3x))
experiments=settings[0+i:9+i,7]
calculations=settings[0+i:9+i,6]
templabels = ('Low P','Int P','High P','Low P','Int P','High P','Low P','Int P','High P')
plt.bar(xspots,experiments,tick_label=templabels,align='center')
plt.bar(xspots,calculations,align='center',color='#840000')
plt.bar(xspots[experiments<calculations],experiments[experiments<calculations],align='center')
plt.text(-.8, -.1,'Low Temperature',alpha=0.7)
plt.text(1.75, -.1,'Intermediate Temperature',alpha=0.7)
plt.text(5.2, -.1,'High Temperature',alpha=0.7)
plt.legend(['Experiment','Calculation'],loc=4,fontsize='small')
#plt.savefig(name+'.png')
```

```
#interface diffusion
import numpy as np
import matplotlib.pyplot as plt
#import matplotlib.animation as animation
import takdifflib
#import matplotlib.patches as patches
#import matplotlib.path as path
#import matplotlib.animation as animation
plt.close(1)
plt.close(2)
#surfneha=np.genfromtxt('C:\\Users\\BJF\\OneDrive for Business\\BA18_2\\SurfaceRoughness\\SurfNeha.csv',delimiter=',')
#surfneha=np.genfromtxt('SurfNeha.csv',delimiter=',')

#surfpoints=(surfneha.shape[0]-1)
#surf=surfneha[1::,:]-np.tile(np.average(surfneha[1:11,:],0),(surfpoints,1))-(np.average(surfneha[-11:-1,:],0)-
np.average(surfneha[1:11,:],0))/(surfpoints)*np.tile(np.linspace(1,surfpoints,surfpoints),(surfneha.shape[1],1)).T
#surf=surf[4400:5400,:]
dt=1
t=1*dt
#points=20
mesh=2
name='condition9'
surfdifon=0
creepon=1
bndrydifon=0

K=1.3806503e-23
R=8.314
T=273+801.
sigb=-2e6
surf1mat=2
surf2mat=2
surfa=-np.copy(surf[:,surf1mat*6+1])
surfb=np.copy(surf[:,surf2mat*6+1])

class mat1:
#    def __init__(self):
#    def calc(at,T):
    al=.45
    gams=1.7
    omega=(1.76*al+1.81*(1-al))*1e-29
    burga=2.95e-10
    burgb=2.86e-10
    burg=burga*al+burgb*(1-al)
    Tm=1933
    R=8.314
    Ga=43.6e9*(1+(T-300.)/Tm*(-1.2))
    Gb=20.5e9*(1+(T-300.)/Tm*(-0.5))
    G=Ga*al+Gb*(1-al)
    Db=3.6e-16*np.exp(-97e3/R/T)*al+.54e-16*np.exp(-153e3/R/T)*(1-al)
    thickb=2*burg
    Ds=5e-6*1e-10*np.exp(-89e3/R/T)*al+5e-6*1e-10*np.exp(-89e3/R/T)*(1-al)
    thickv=2e-10#*thickb*2
    Dv=thickv*(8.6e-10*np.exp(-150e3/R/T)*al+1.9e-7*np.exp(-153e3/R/T)*(1-al))
    f=2*(5e-6*1e-10*al+5e-6*1e-10*(1-al))/(3.6e-16*al+.54e-16*(1-al))
    Ac=7.7e4*8.6e-10*np.exp(-150e3/R/T)*Ga*burga/K/T*al+1e5*1.9e-7*np.exp(-153e3/R/T)*Gb*burgb/K/T*(1-al)
    m=4.3
class mat2:
    al=.45
```

```
    gams=1.7
    omega=(1.76*al+1.81*(1-al))*1e-29
    burga=2.95e-10
    burgb=2.86e-10
    burg=burga*al+burgb*(1-al)
    Tm=1933
    Ga=43.6e9*(1+(T-300.)/Tm*(-1.2))
    Gb=20.5e9*(1+(T-300.)/Tm*(-0.5))
    G=Ga*al+Gb*(1-al)
    Db=3.6e-16*np.exp(-97e3/R/T)*al+.54e-16*np.exp(-153e3/R/T)*(1-al)
    thickb=2*burg
    Ds=5e-6*1e-10*np.exp(-89e3/R/T)*al+5e-6*1e-10*np.exp(-89e3/R/T)*(1-al)
    thickv=2e-10#*thickb*2
    Dv=thickv*(8.6e-10*np.exp(-150e3/R/T)*al+1.9e-7*np.exp(-153e3/R/T)*(1-al))
    f=2*(5e-6*1e-10*al+5e-6*1e-10*(1-al))/(3.6e-16*al+.54e-16*(1-al))
    Ac=7.7e4*8.6e-10*np.exp(-150e3/R/T)*Ga*burga/K/T*al+1e5*1.9e-7*np.exp(-153e3/R/T)*Gb*burgb/K/T*(1-al)
    m=4.3

fo=-sigb/np.min(np.array([mat1.G*1e-3,mat2.G*1e-3]))
disp=0



surf1,surf2=takdifflib.overlap(surfa,surfb,fo)

#plt.plot(surf1)
#plt.plot(surf2)

#generated surfaces
fogen=.1
genlen=10
array=.01*np.sin(np.linspace(0,np.int(genlen*(1-fogen)),np.int(genlen*(1-fogen)))/np.int(genlen*(1-fogen))*np.pi)
array=np.hstack((array,np.zeros(np.int(genlen*fogen))))
surf1=np.tile(array,np.int(3500/array.shape[0]))
surf1=surf1[np.int(genlen/2):4*genlen]
surf2=-surf1


dx=1e-6/mesh
yuposi=surf1*1e-6
ylposi=surf2*1e-6
xposi=np.arange(yuposi.size)*dx*mesh
xpos=np.arange(yuposi.size*mesh)*dx
yupos=np.interp(xpos,xposi,yuposi)
ylpos=np.interp(xpos,xposi,ylposi)
xpos2=np.copy(xpos)



#geometry and curvatures
ypos=yupos

Ldist=(yupos.size)

L=Ldist*dx


logyu=np.copy(yupos)
logxu=np.copy(xpos)
logyl=np.copy(ylpos)
dVsurf=np.zeros(np.int(t/dt)+1)
```

```
dVbndry=np.zeros(np.int(t/dt)+1)
dVcreep=np.zeros(np.int(t/dt)+1)
#material properties

xposoff1=xpos[0]
xposoff2=xpos[-1]



for i in range(1,np.int(t/dt)+1):

   bonds,voids=takdifflib.bondcompute(Ldist,yupos,ylpos)

   dyu=np.gradient(yupos,dx)
   dyl=np.gradient(ylpos,dx)
   ddyu=np.gradient(dyu,dx)
   ddyl=np.gradient(dyl,dx)
   contact=np.sum(bonds[:,1]-bonds[:,0])
   basestress=sigb*Ldist*dx/contact/dx
#    ktip=np.zeros((voids.shape[0],2))
#    dktip=np.zeros((voids.shape[0],2))
   bndyupos=np.zeros(Ldist)
   bndylpos=np.zeros(Ldist)
   for vd in range(0,voids.shape[0]-1):
      up=len(yupos[voids[vd,0]:voids[vd,1]])
      if up>0:
         if voids[vd,1]<Ldist:
            if voids[vd,0]>0:
               tip2xpos=xpos[voids[vd,1]+1]
               tip2ypos=yupos[voids[vd,1]+1]
               tip1xpos=xpos[voids[vd,0]-1]
               tip1ypos=ylpos[voids[vd,0]-1]
         if voids[vd,1]==Ldist:
            tip2xpos=xpos[voids[vd,1]-1]
            tip2ypos=yupos[voids[vd,1]-1]
            tip1xpos=xpos[voids[vd,0]-1]
            tip1ypos=ylpos[voids[vd,0]-1]
         if voids[vd,0]==0:
            tip2xpos=xpos[voids[vd,1]+1]
            tip2ypos=yupos[voids[vd,1]+1]
            tip1xpos=xpos[voids[vd,0]]
            tip1ypos=ylpos[voids[vd,0]]
         xv=np.hstack((xpos[voids[vd,0]:voids[vd,1]],tip2xpos,xpos[voids[vd,1]:voids[vd,0]:-1],tip1xpos))
         dyv=np.gradient(np.hstack((yupos[voids[vd,0]:voids[vd,1]],tip2ypos,ylpos[voids[vd,1]:voids[vd,0]:-1],tip1ypos)),dx)
         ddyv=np.gradient(dyv,dx)
         kv=ddyv/np.power(1+dyv**2,1.5)
         ktip2=2*np.sqrt(dx)/(yupos[voids[vd,1]]-ylpos[voids[vd,1]])*2
         ktip1=2*np.sqrt(dx)/(yupos[voids[vd,0]]-ylpos[voids[vd,0]])*2
         dkv=np.gradient(kv,dx, edge_order=2)
         ddkv=np.gradient(dkv,dx, edge_order=2)

         if bndrydifon==1:
            voldif=1
            ktip1=(kv[0]+(3*dx/mat1.gams/(xpos[bonds[vd,1]]-xpos[bonds[vd,0]])**2*4/mat1.f*(xpos[voids[vd,1]]-
xpos[voids[vd,0]])*basestress))/(1+3*dx/(xpos[bonds[vd,1]]-xpos[bonds[vd,0]])/mat1.f)
#             =basestress/mat1.gams-(bonds[:,1]-bonds[:,0])/2/3*mat1.f*dkyu[bonds[:,0]]
            bndrystress=basestress
            betaul=-3*mat1.Db/K/T/(xpos[bonds[vd-1,1]]-xpos[bonds[vd-1,0]])**2*4*((up*dx+(xpos[bonds[vd-1,1]]-
xpos[bonds[vd-1,0]]))/2/(xpos[bonds[vd-1,1]]-xpos[bonds[vd-1,0]])*2*bndrystress-ktip1*mat1.gams)
            dusdtul=np.abs((-mat1.thickb/2)*mat1.omega*betaul/2)
            betaur=-3*mat1.Db/K/T/(xpos[bonds[vd,1]]-xpos[bonds[vd,0]])**2*4*((up*dx+(xpos[bonds[vd-1,1]]-xpos[bonds[vd-
1,0]]))/2/(xpos[bonds[vd,1]]-xpos[bonds[vd,0]])*2*bndrystress-ktip1*mat1.gams)
```

```
            dusdtur=np.abs((-mat1.thickb/2)*mat1.omega*betaur/2)
            betall=-3*mat2.Db/K/T/(xpos[bonds[vd-1,1]]-xpos[bonds[vd-1,0]])**2*4*((up*dx+(xpos[bonds[vd-1,1]]-
xpos[bonds[vd-1,0]]))/2/(xpos[bonds[vd-1,1]]-xpos[bonds[vd-1,0]])*2*bndrystress-ktip1*mat2.gams)
            dusdtll=np.abs((-mat2.thickb/2)*mat2.omega*betall/2)
            betalr=-3*mat2.Db/K/T/(xpos[bonds[vd,1]]-xpos[bonds[vd,0]])**2*4*((up*dx+(xpos[bonds[vd-1,1]]-xpos[bonds[vd-
1,0]]))/2/(xpos[bonds[vd,1]]-xpos[bonds[vd,0]])*2*bndrystress-ktip1*mat2.gams)
            dusdtlr=np.abs((-mat2.thickb/2)*mat2.omega*betalr/2)
            dudtu=(dusdtur+dusdtul+voldif*(dusdtul+dusdtur)/mat1.Db*mat1.Dv/mat1.thickb*mat1.thickv*2)/2
            dudtl=(dusdtlr+dusdtll+voldif*(dusdtlr+dusdtll)/mat2.Db*mat2.Dv/mat1.thickb*mat1.thickv*2)/2
            bndyupos[voids[vd,0]:voids[vd,1]]+=-dudtu-dudtu*((xpos[bonds[vd-1,1]]-xpos[bonds[vd-1,0]])+(xpos[bonds[vd,1]]-
xpos[bonds[vd,0]]))/2/(xpos[voids[vd,1]]-xpos[voids[vd,0]])
            bndylpos[voids[vd,0]:voids[vd,1]]+=dudtl+dudtl*((xpos[bonds[vd-1,1]]-xpos[bonds[vd-1,0]])+(xpos[bonds[vd,1]]-
xpos[bonds[vd,0]]))/2/(xpos[voids[vd,1]]-xpos[voids[vd,0]])

    if surfdifon==1:
#       kyu=np.zeros(Ldist)
#       kyl=np.zeros(Ldist)
#       for vdn in range(0,voids.shape[0]):
#           voidspace=np.arange(voids[vdn,0],voids[vdn,1],1)
#           for j in voidspace:
#               if j<Ldist-1:
#                   if j>1:
#                       kyu[j]=takdifflib.curvature(yupos[j-1],yupos[j],yupos[j+1],xpos[j-1],xpos[j],xpos[j+1])
#                       kyl[j]=takdifflib.curvature(ylpos[j-1],ylpos[j],ylpos[j+1],xpos[j-1],xpos[j],xpos[j+1])
#           tip1=voids[vdn,0]-1
#           tip2=voids[vdn,1]+1
#
#           if tip2<Ldist-1:
#               if tip1>1:
#                   kyu[tip1]=takdifflib.curvature(ylpos[tip1+1],yupos[tip1],yupos[tip1+1],-xpos[tip1+1],xpos[tip1],xpos[tip1+1])
#                   kyu[tip1]=kyu[tip1+1]
#                   kyl[tip1]=kyu[tip1]
#                   kyu[tip2]=takdifflib.curvature(yupos[tip2-1],yupos[tip2],ylpos[tip2-1],xpos[tip2-1],xpos[tip2],-xpos[tip2-1])
#                   kyu[tip2]=kyu[tip2+1]
#                   kyl[tip2]=kyu[tip2]
        kyu=ddyu/np.power((1+dyu**2),3/2.)
        kyl=ddyl/np.power((1+dyl**2),3/2.)
        dsu=np.sqrt(dyu**2+dx**2)
        dsl=np.sqrt(dyu**2+dx**2)
        dkyu=np.gradient(kyu,dx)
        dkyl=np.gradient(kyl,dx)

        vyus,betaus=takdifflib.surfdif(dkyu,dyu,dx,mat1.gams,mat1.Ds,K,T,mat1.omega)
        vyuv,betauv=takdifflib.surfdif(dkyu,dyu,dx,mat1.gams,mat1.Dv,K,T,mat1.omega)
        vyls,betals=takdifflib.surfdif(dkyl,dyl,dx,mat2.gams,mat2.Ds,K,T,mat2.omega)
        vylv,betalv=takdifflib.surfdif(dkyl,dyl,dx,mat2.gams,mat2.Dv,K,T,mat2.omega)

        yuposref=np.copy(yupos)
        ylposref=np.copy(ylpos)
        dxupos=np.zeros(Ldist)
        dyupos=np.zeros(Ldist)
        dxlpos=np.zeros(Ldist)
        dylpos=np.zeros(Ldist)
        for l in range(0,voids.shape[0]):

dxupos[voids[l,0]:voids[l,1]]+=vyus[voids[l,0]:voids[l,1]]*np.cos(betaus[voids[l,0]:voids[l,1]])+vyuv[voids[l,0]:voids[l,1]]*np.c
os(betauv[voids[l,0]:voids[l,1]])

dyupos[voids[l,0]:voids[l,1]]+=vyus[voids[l,0]:voids[l,1]]*np.sin(betaus[voids[l,0]:voids[l,1]])+vyuv[voids[l,0]:voids[l,1]]*np.si
n(betauv[voids[l,0]:voids[l,1]])
            dxlpos[voids[l,0]:voids[l,1]]+=-vyls[voids[l,0]:voids[l,1]]*np.cos(betals[voids[l,0]:voids[l,1]])-
vylv[voids[l,0]:voids[l,1]]*np.cos(betalv[voids[l,0]:voids[l,1]])
```

```
        dylpos[voids[l,0]:voids[l,1]]+=-vyls[voids[l,0]:voids[l,1]]*np.sin(betals[voids[l,0]:voids[l,1]])-
vylv[voids[l,0]:voids[l,1]]*np.sin(betalv[voids[l,0]:voids[l,1]])
      yupos=takdifflib.changepos_remesh(dyupos,dxupos,yupos,xpos,xpos2,dt)
      ylpos=takdifflib.changepos_remesh(dylpos,dxlpos,ylpos,xpos,xpos2,dt)

   if creepon==1:
      creepyu=np.zeros((Ldist))
      creepxu=np.zeros((Ldist))
#       creepyl=np.zeros((Ldist))
#       creepxl=np.zeros((Ldist))

      luint,ruint=takdifflib.indexcreeppoints(yupos,xpos,bonds,voids,Ldist)
      llint,rlint=takdifflib.indexcreeppoints(-ylpos,xpos,bonds,voids,Ldist)
      creepyu,creepxu,xdist=takdifflib.creepdeform(1,yupos,xpos,dyu,luint,ruint,basestress,bonds,voids,Ldist,mat1,dx)
      creepyl,creepxl,xdist=takdifflib.creepdeform(-1,ylpos,xpos,dyl,llint,rlint,basestress,bonds,voids,Ldist,mat2,dx)

   #compute left side larger deformation top
#       sigxu=np.zeros(Ldist)
#       xulength=np.zeros(Ldist)
#       dyul=np.gradient(yupos)
#       dyll=np.gradient(ylpos)
#       dhidtu=np.zeros(Ldist)
#       dxidtu=np.zeros(Ldist)
#       for l in range(0,bonds.shape[0]):
#          bondistress=basestress#*(bonds[l,1]-bonds[l,0])/contact
#          luint2=luint[l,:]
#          ruint2=ruint[l,:]
#          luint2=luint2[luint2>0]
#          ruint2=ruint2[ruint2>0]
#          xuldist=np.abs(np.interp(yupos[luint2],yupos[ruint2],xpos[ruint2],right=xpos[-1])-xpos[luint2])
#          ladd=bondistress*(bonds[l,1]-bonds[l,0])*dx/xuldist
#          ladd[np.isnan(ladd)]=0
#          sigxu[luint2]=sigxu[luint2]+ladd
#          xulength[luint2]=xuldist
#          dhidtu[luint2]=-np.abs(dyul[luint2])*mat1.Ac*np.abs(sigxu[luint2]/mat1.G)**mat1.m
#          dxidtu[luint2]=-dhidtu[luint2]*np.abs(xulength[luint2])/2/dyul[luint2]
#          xurdist=np.abs(np.interp(yupos[ruint2],yupos[luint2],xpos[luint2],right=xpos[0])-xpos[ruint2])
#          radd=bondistress*(bonds[l,1]-bonds[l,0])*dx/xurdist
#          radd[np.isnan(radd)]=0
#          sigxu[ruint2]=sigxu[ruint2]+radd
#          xulength[ruint2]=xurdist
#          dhidtu[ruint2]=-np.abs(dyul[ruint2])*mat1.Ac*np.abs(sigxu[ruint2]/mat1.G)**mat1.m
#          dxidtu[ruint2]=-dhidtu[ruint2]*np.abs(xulength[ruint2])/2/dyul[ruint2]
#       dxidtu[np.isnan(dxidtu)]=0
#       for vdn in range(0,voids.shape[0]):
#          maxpos=np.argmax(yupos[voids[vdn,0]:voids[vdn,1]+1])+voids[vdn,0]
#          creepyu[voids[vdn,0]-1:maxpos+1]=np.cumsum(dhidtu[voids[vdn,0]-1:maxpos+1])
##            creepxu[voids[vdn,0]-1:maxpos]=np.cumsum(dxidtu[voids[vdn,0]-1:maxpos])
#          creepxu[voids[vdn,0]-1:maxpos+1]=dxidtu[voids[vdn,0]-1:maxpos+1]
#          temp=dhidtu[maxpos:voids[vdn,1]+1]
#          temp2=np.cumsum(temp[::-1])
#          creepyu[maxpos:voids[vdn,1]+1]=temp2[::-1]
##            creepxu[maxpos:voids[vdn,1]+1]=np.cumsum(dxidtu[maxpos:voids[vdn,1]+1])
#          creepxu[maxpos:voids[vdn,1]+1]=dxidtu[maxpos:voids[vdn,1]+1]
##            creepxu[maxpos]=0
##            creepyu[maxpos]=0
#       #compute left side larger deformation top
#       sigxl=np.zeros(Ldist)
#       xllength=np.zeros(Ldist)
#       for l in range(0,bonds.shape[0]):
#          bondistress=-basestress#*(bonds[l,1]-bonds[l,0])/contact
#          llint2=llint[l,:]
```

```
#           rlint2=rlint[l,:]
#           llint2=llint2[llint2>0]
#           rlint2=rlint2[rlint2>0]
#           xlldist=np.abs(np.interp(-ylpos[llint2],-ylpos[rlint2],xpos[rlint2],right=xpos[-1])-xpos[llint2])
#           ladd=bondistress*(bonds[l,1]-bonds[l,0])*dx/xlldist
#           ladd[np.isnan(ladd)]=0
#           sigxl[llint2]=sigxl[llint2]+ladd
#           xllength[llint2]=xlldist
#           xlrdist=np.abs(np.interp(-ylpos[rlint2],-ylpos[llint2],xpos[llint2],right=xpos[0])-xpos[rlint2])
#           radd=bondistress*(bonds[l,1]-bonds[l,0])*dx/xlrdist
#           radd[np.isnan(radd)]=0
#           sigxl[rlint2]=sigxl[rlint2]+radd
#           xllength[rlint2]=xlrdist
#       dhidtl=np.abs(dyll)*mat2.Ac*np.abs(sigxl/mat2.G)**mat2.m
#       dxidtl=-dhidtl*np.abs(xllength)/2/dyll
#       dxidtl[np.isnan(dxidtl)]=0
#       for vdn in range(0,voids.shape[0]):
#           maxpos=np.argmin(ylpos[voids[vdn,0]:voids[vdn,1]+1])+voids[vdn,0]
#           creepyl[voids[vdn,0]-1:maxpos]=np.cumsum(dhidtl[voids[vdn,0]-1:maxpos])
##            creepxl[voids[vdn,0]-1:maxpos]=np.cumsum(dxidtl[voids[vdn,0]-1:maxpos])
#           creepxl[voids[vdn,0]-1:maxpos]=dxidtl[voids[vdn,0]-1:maxpos]
#           temp=dhidtl[maxpos:voids[vdn,1]+1]
#           temp2=np.cumsum(temp[::-1])
#           creepyl[maxpos:voids[vdn,1]+1]=temp2[::-1]
##            creepxl[maxpos:voids[vdn,1]+1]=np.cumsum(dxidtl[maxpos:voids[vdn,1]+1])
#           creepxl[maxpos:voids[vdn,1]+1]=dxidtl[maxpos:voids[vdn,1]+1]
##            creepxl[maxpos]=0
#           creepyl[maxpos]=0
        yuposref=np.copy(yupos)
        ylposref=np.copy(ylpos)
        yupos=takdifflib.changepos_remesh(creepyu,creepxu,yupos,xpos,xpos2,dt)
        ylpos=takdifflib.changepos_remesh(creepyl,creepxl,ylpos,xpos,xpos2,dt)
#       yupos+=creepyu*dt
#       xpos+=creepxu*dt
#       for vdn1 in range(1,voids.shape[0]):
#           maxpos=np.argmax(yupos[voids[vdn1,0]:voids[vdn1,1]+1])+voids[vdn1,0]
#           yupos[maxpos]=(yupos[maxpos]+(yupos[maxpos-1]+yupos[maxpos+1])/2)/2
#           xpos[maxpos]=(xpos[maxpos-1]+xpos[maxpos+1])/2
#       yupos=np.interp(xpos2,xpos,yupos)
#       xpos=np.copy(xpos2)
#       ylpos+=creepyl*dt
#       xpos+=creepxl*dt
#       for vdn2 in range(1,voids.shape[0]):
#           maxpos=np.argmin(ylpos[voids[vdn2,0]:voids[vdn2,1]+1])+voids[vdn2,0]
#           ylpos[maxpos]=(ylpos[maxpos]+(ylpos[maxpos-1]+ylpos[maxpos+1])/2)/2
#           xpos[maxpos]=(xpos[maxpos-1]+xpos[maxpos+1])/2
#       ylpos=np.interp(xpos2,xpos,ylpos)
#       xpos=np.copy(xpos2)

        dVcreep[i]=np.sum(np.abs(yupos-yuposref))+np.sum(np.abs(ylpos-ylposref))

    if bndrydifon==1:
        yuposref=np.copy(yupos)
        ylposref=np.copy(ylpos)
        bndyupos[np.isnan(bndyupos)]=0
        bndylpos[np.isnan(bndylpos)]=0
        bndyupos[np.isinf(bndyupos)]=-1e-12
        bndylpos[np.isinf(bndylpos)]=1e-12
        yupos+=bndyupos*dt
        ylpos+=bndylpos*dt
        dVbndry[i]=np.sum(np.abs(yupos-yuposref))+np.sum(np.abs(ylpos-ylposref))
```

```
    diff=yupos-ylpos
    avg=(yupos+ylpos)/2
#    yupos[diff<1]=avg[diff<0]
    yupos[diff<0]=avg[diff<0]
    ylpos[diff<0]=avg[diff<0]
    print(i/(t/dt))




#    if np.int(i*dt/(t/points))-i*dt/(t/points)==0:
    logyu=np.vstack((logyu,yupos))
#    logxu=np.vstack((logxu,xpos))
    logyl=np.vstack((logyl,ylpos))

#fig, ax = plt.subplots()
#
#x=xpos
#line, = ax.plot(x,logyu[0,:],'b')
#line2, = ax.plot(x,logyl[0,:],'b')
#def animate(i):
#    line.set_ydata(logyu[i,:])  # update the data
#    line2.set_ydata(logyl[i,:])
#    return line,
##def init():
##    line.set_ydata(np.ma.array(x, mask=True))
##    return line,
#ani = animation.FuncAnimation(fig, animate)
##plt.xlim([.0044,.0055])
#plt.show()

#fig = plt.figure()
#l, = plt.plot([], [], 'k-o')
#x0, y0 = 0, 0
#import matplotlib
#matplotlib.use("Agg")
#import matplotlib.animation as manimation
#FFMpegWriter = manimation.writers['ffmpeg']
#metadata = dict(title='Movie Test', artist='Matplotlib',
#                comment='Movie support!')
#writer = FFMpegWriter(fps=15, metadata=metadata)
#
#with writer.saving(fig, "writer_test.mp4", 100):
#    for i in range(100):
#        line.set_ydata(logyu[i,:])  # update the data
#        line2.set_ydata(logyl[i,:])
#        writer.grab_frame()

#ani.save('Bonding.mp4')
line1,line2,=plt.plot(xpos,logyu[0,:],'C0',xpos,logyl[0,:],'C0')
line3,line4,=plt.plot(xpos,logyu[-1,:],'C1',xpos,logyl[-1,:],'C1')
plt.legend([line2, line4], ['Original Voids', 'Voids after diffusion'])
plt.xlabel('Horiontal Distance (m)')
plt.ylabel('Vertical Height (m)')
plt.title(name)
np.save(name+'u',logyu)
np.save(name+'l',logyl)
#plt.xlim([.004725,.004925])
#
#plt.figure(2)
#bondplot=2
```

```
#plt.plot(xpos,logyu[-1,:],'C1')
#plt.scatter(xpos[ruint[bondplot]],logyu[-1,ruint[bondplot]],c='C1')
#plt.legend(['Void Surface','Creep Points of Interest'])
#plt.scatter(xpos[luint[bondplot]],logyu[-1,luint[bondplot]],c='C0')
#plt.scatter(xpos[rlint[bondplot]],logyl[-1,rlint[bondplot]],c='C1')
#plt.scatter(xpos[llint[bondplot]],logyl[-1,llint[bondplot]],c='C0')
#plt.plot(xpos,logyl[-1,:],'C1')
#plt.xlabel('Horiontal Distance (m)')
#plt.ylabel('Vertical Height (m)')
#plt.title('Points That Creep is Calculated Over for a Single Void')
bonds,voids=takdifflib.bondcompute(Ldist,yupos,ylpos)
contact=np.sum(bonds[:,1]-bonds[:,0])
bonding=contact/((bonds[-1,1]+bonds[-1,0])/2-(bonds[0,1]+bonds[0,0])/2)
print(bonding)


#
#plt.figure(num=2,figsize=(4.8,2.37),dpi=200)
#x=np.linspace(0,100,101)*dt/3600
#totalvol=np.sum(logyu[0,:]-logyl[0,:])
#line1,=plt.plot(x,np.cumsum(dVsurf/totalvol*100),'C0')
#line2,=plt.plot(x,np.cumsum(dVbndry/totalvol*100),'C1')
#line3,=plt.plot(x,np.cumsum(dVcreep/totalvol*100),'C2')
#plt.legend([line3, line2, line1], ['Creep', 'Boundary Diffusion','Surface Diffusion'])
#plt.xlabel('Bonding Time (hours)')
#plt.ylabel('Percentage of Volume Closed')
#plt.title('Volume Closure Rates for the Different Bonding Mechanisms')

#plt.axis([4700,5300,-.5,1])
#plt.plot(xpos*1e6,logyu[-1,:]*1e6,'g',xpos*1e6,logyl[-1,:]*1e6,'g')
```

Surface Tracking Library

```
import numpy as np
def overlap(surfa,surfb,fo):
    surfpoints=surfa.shape[0]
    #compute initial overlap of surfaces
    inc=np.abs(np.max(surfa)-np.min(surfa))
    dif=np.mean(surfb)-np.mean(surfa)
    for i in range(0,10000):
        olap=surfa+dif-surfb
        foth=olap[olap<=0].shape[0]*1.0/surfpoints
        if foth<fo:
            dif=dif-inc
        if foth>fo:
            dif=dif+inc
            inc=inc/2
        if np.abs(foth-fo)<=1e-4:
            break

    olap[olap<=0]=0.0
    surfa=surfa+dif
    averagesurf=(surfa+surfb)/2
    surfa[olap==0]=averagesurf[olap==0]
    surfb[olap==0]=averagesurf[olap==0]
    return surfa,surfb

def bondcompute(Ldist,yupos,ylpos):
    #search through mesh for bonded areas
    key=0
    bonds=np.zeros((1,2),dtype=int)
```

```
    voidbond=np.ones((Ldist))
    for j in range(1,Ldist-1):
        if yupos[j]-ylpos[j]==0:
            voidbond[j]=0
            if np.abs(yupos[j-1]-ylpos[j-1])>0:
                bondinit=j
            if yupos[j+1]-ylpos[j+1]>0:
                bondend=j
                key=1
            if key==1:
                bonds=np.vstack((bonds,np.array([bondinit,bondend])))
                key=0
    bonds=bonds[1:,:]
    #note: will fail if overlap starts with a bonded section
    voids=np.transpose(np.vstack((np.append(np.zeros((1),dtype=int),bonds[:,1]+1),np.append(bonds[:,0]-1,Ldist))))
    return bonds,voids

def surfdif(dky,dy,dx,gams,Diffconst,K,T,omega):
    #calculate surface diffusion
    Jys=gams*Diffconst/K/T*dky
    dJys=np.gradient(Jys,dx)
    vys=-omega*dJys
    betas=np.arctan(dy)-np.pi/2
    return vys,betas

def changepos_remesh(delypos,delxpos,ypos,xpos,xpos2,dt):
    #Add the changes in position due to bonding mechanism to x and y posisitions then remesh
    ychgpos=ypos+delypos*dt
    xchgpos=xpos+delxpos*dt
    ynewpos=np.interp(xpos2,xchgpos,ychgpos)
    return ynewpos

def indexcreeppoints(ypos,xpos,bonds,voids,Ldist):
    #find points of creep propagating out from bonds
    lint=np.zeros((bonds.shape[0],Ldist),dtype=int)
    rint=np.zeros((bonds.shape[0],Ldist),dtype=int)
    for l in range(0,bonds.shape[0]):
        lstop=0
        lastmaxl=bonds[l,0]
        rstop=0
        lastmaxr=bonds[l,1]
        for k in range(1,1000000):
            if lstop==0:
                if ypos[bonds[l,0]-k]>ypos[lastmaxl]:
                    lint[l,k-1]=bonds[l,0]-k-1
                    lastmaxl=bonds[l,0]-k
                if ypos[bonds[l,0]-k]==np.max(ypos[0:voids[l,1]+1]):
                    lstop=1
#                 if bonds[l,0]-k==voids[l,0]:
#                     lustop=1
                if bonds[l,0]-k-1==0:
                    lstop=1
            if rstop==0:
                if ypos[bonds[l,1]+k]>ypos[lastmaxr]:
                    rint[l,k-1]=bonds[l,1]+k
                    lastmaxr=bonds[l,1]+k
                if ypos[bonds[l,1]+k]==np.max(ypos[voids[l+1,0]::]):
                    rstop=1
#                 if bonds[l,1]+k==voids[l+1,1]:
#                     rustop=1
                if bonds[l,1]+k+1==Ldist:
                    rstop=1
```

```
            if rstop*lstop==1:
                break
    return lint,rint
#
#def creepslice(ypos,xpos,bonds,voids,Ldist):
#    slen=np.zeros(Ldist)
#    sstress=np.zeros(Ldist)
#    for i in range(0,Ldist):
#        if ypos[i]>ypos[i+1]:
#            c=0
#            while ypos[i]>ypos[i+c]:
#                c=c+1
#            xslice=xpos[i+c]-

#
def creepdeform(ul,ypos,xpos,dy,lint,rint,basestress,bonds,voids,Ldist,mat,dx):
    ypos=ypos*ul
    sigx=np.zeros(Ldist)
    xlength=np.zeros(Ldist)
    dhidt=np.zeros(Ldist)
    dxidt=np.zeros(Ldist)
    creepy=np.zeros((Ldist))
    creepx=np.zeros((Ldist))
    for l in range(0,bonds.shape[0]):
        bondistress=basestress#*(bonds[l,1]-bonds[l,0])/contact
        lint2=lint[l,:]
        rint2=rint[l,:]
        lint2=lint2[lint2>0]
        rint2=rint2[rint2>0]
        for i in lint2:
            ry=np.max(np.where(ypos[i]>ypos[rint2]))
            ry1=rint2[ry]
            ry2=ry1+1
            xldist=(xpos[ry2]-xpos[ry1])/(ypos[ry2]-ypos[ry1])*(ypos[i]-ypos[ry1])+xpos[ry1]-xpos[i]
            ladd=bondistress*(bonds[l,1]-bonds[l,0])*dx/xldist
            sigx[i]=sigx[i]+ladd
            xlength[i]=xldist
#        xldist=np.abs(np.interp(ypos[lint2],ypos[rint2],xpos[rint2],right=xpos[-1])-xpos[lint2])

#        ladd[np.isnan(ladd)]=0


        xrdist=np.abs(np.interp(ypos[rint2],ypos[lint2],xpos[lint2],right=xpos[0])-xpos[rint2])
        radd=bondistress*(bonds[l,1]-bonds[l,0])*dx/xrdist
#        radd[np.isnan(radd)]=0
        sigx[rint2]=sigx[rint2]+radd
        xlength[rint2]=xrdist
    dhidt=np.abs(dy)*mat.Ac*np.abs(sigx/mat.G)**mat.m
    dxidt=dhidt*np.abs(xlength)/2/dy
    dxidt[np.isnan(dxidt)]=0
    for vdn in range(0,voids.shape[0]):
        maxpos=np.argmin(ypos[voids[vdn,0]:voids[vdn,1]+1])+voids[vdn,0]
        creepy[voids[vdn,0]:maxpos+1]=creepy[voids[vdn,0]:maxpos+1]+np.cumsum(dhidt[voids[vdn,0]:maxpos+1])
#            creepxl[voids[vdn,0]-1:maxpos]=np.cumsum(dxidtl[voids[vdn,0]-1:maxpos])
        creepx[voids[vdn,0]:maxpos]=creepx[voids[vdn,0]:maxpos]+dxidt[voids[vdn,0]:maxpos]
        temp=dhidt[maxpos:voids[vdn,1]+1]
        temp2=np.cumsum(temp[::-1])
        creepy[maxpos:voids[vdn,1]+1]=creepy[maxpos:voids[vdn,1]+1]+temp2[::-1]
#            creepxl[maxpos:voids[vdn,1]+1]=np.cumsum(dxidtl[maxpos:voids[vdn,1]+1])
        creepx[maxpos:voids[vdn,1]+1]=creepx[maxpos:voids[vdn,1]+1]+dxidt[maxpos:voids[vdn,1]+1]
        creepy[maxpos]=creepy[maxpos]/2
        creepy[maxpos]=creepy[maxpos]/2
```

```
    return creepy,creepx,xldist

def curvature(y1,y2,y3,x1,x2,x3):
    ma=(y2-y1)/(x2-x1)
    mb=(y3-y2)/(x3-x2)
    xc=(ma*mb*(y1-y3)+mb*(x1+x2)-ma*(x2+x3))/2/(mb-ma)
    yc=-1/ma*(xc-(x1+x2)/2)+(y1+y2)/2
    radius=np.sqrt((y2-yc)**2+(x2-xc)**2)
    curv=1/radius
    return curv
```

```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
from scipy import stats
from scipy import interpolate

plt.close(1)
plt.close(2)
plt.close(3)
plt.close(4)
plot=0
finish=1
ftype='.tif'
flocation='//home//bjferg//Desktop//Extra//'
#fname='F12S1FT50x'
fname='F26L2CBcrop'
s_mask=7
img=cv.imread(flocation+fname+ftype,0)
Threshold_ratio=.945
#x=705
#y=660
#crop=img[y:y+128,x:x+256]
thresh = cv.adaptiveThreshold(img,255,cv.ADAPTIVE_THRESH_GAUSSIAN_C,cv.THRESH_BINARY,49,1)
blur = cv.blur(img,(15,15))
ret,voidthresh = cv.threshold(blur,np.mean(img)/1.27,255,cv.THRESH_BINARY)
#plt.figure(1)
#plt.imshow(img, cmap='Greys_r')
if plot==1:
    plt.figure(4)
    plt.imshow(thresh, cmap='Greys_r')
    plt.figure(1)
    plt.imshow(voidthresh, cmap='Greys_r')
thresh[thresh==0]=150
thresh[thresh>240]=0
#pointsxy=np.genfromtxt(flocation+'Resultslarge.csv',dtype=np.int32,delimiter=',')
pointsxy=np.genfromtxt(flocation+'Resultsupdate.csv',dtype=np.int32,delimiter=',')
#plt.figure(2)

#sobely = cv.Sobel(img,cv.CV_64F,0,1,ksize=15)
#sobelx = cv.Sobel(img,cv.CV_64F,1,0,ksize=15)
#sobel=np.sqrt(sobelx**2+sobely**2)
#edges = cv.Canny(sobel,1,255)
#blursobel=cv.blur(sobel,(2,2))

#plt.imshow(sobelthresh,cmap='Greys_r')
if finish==1:
    gmap=np.zeros(img.shape)+thresh
    gmap[voidthresh==0]=2500
    for j in range(0,pointsxy.shape[0]):
        gmap[pointsxy[j,2],pointsxy[j,1]]=pointsxy[j,0]
    gmaptmp=np.copy(gmap)
    for i in range(1,60):
        for j in range(0,gmap.shape[0]):
            for k in range(0,gmap.shape[1]):
                if gmap[j,k]>0:
                    if gmap[j,k]<800:
                        val=gmap[j,k]
                        if j-1>0:
                            if gmap[j-1,k]==0:
```

```
                        gmaptmp[j-1,k]=val
                    if j+1<gmap.shape[0]:
                        if gmap[j+1,k]==0:
                            gmaptmp[j+1,k]=val
                    if k-1>0:
                        if gmap[j,k-1]==0:
                            gmaptmp[j,k-1]=val
                    if k+1<gmap.shape[1]:
                        if gmap[j,k+1]==0:
                            gmaptmp[j,k+1]=val
            gmap=np.copy(gmaptmp)
            if np.max((gmaptmp==0))==0:
                break
        gmap[gmap==150]=0
        gmaptmp=np.copy(gmap)
        for i in range(1,60):
            for j in range(0,gmap.shape[0]):
                for k in range(0,gmap.shape[1]):
                    if gmap[j,k]>0:
                        if gmap[j,k]<800:
                            val=gmap[j,k]
                            if j-1>=0:
                                if gmap[j-1,k]==0:
                                    gmaptmp[j-1,k]=val
                            if j+1<gmap.shape[0]:
                                if gmap[j+1,k]==0:
                                    gmaptmp[j+1,k]=val
                            if k-1>=0:
                                if gmap[j,k-1]==0:
                                    gmaptmp[j,k-1]=val
                            if k+1<gmap.shape[1]:
                                if gmap[j,k+1]==0:
                                    gmaptmp[j,k+1]=val
            gmap=np.copy(gmaptmp)
            if np.max((gmaptmp==0))==0:
                break
        gmap[gmap==2500]=0

#       gmap[0:10,:]=36
#       gmap[-10::,:]=36
#       gmap=gmap[:,0:np.int(gmap.shape[1]/2.0)]
        if plot==1:
            plt.figure(2)
            plt.imshow(gmap,cmap='gnuplot')
        scale=9.45 #pixels per micrometer
        coarsen=2
        refine=1
        height=gmap.shape[0]/scale
        width=gmap.shape[1]/scale
        xspace=np.linspace(0,1,gmap.shape[1]/coarsen)*width
        yspace=np.linspace(1,0,gmap.shape[0]/coarsen)*height
        xspace2=np.linspace(0,1,gmap.shape[1]*refine)*width
        yspace2=np.linspace(1,0,gmap.shape[0]*refine)*height
        xmap,ymap=np.meshgrid(xspace,yspace)
        xmap2,ymap2=np.meshgrid(xspace2,yspace2)
        gmap2=np.zeros((gmap.shape[0]/coarsen,gmap.shape[1]/coarsen))
        for i in range(0,gmap2.shape[0]):
            for j in range(0,gmap2.shape[1]):
                gmap2[i,j],count=stats.mode(gmap[coarsen*i:coarsen*(i+1),coarsen*j:coarsen*(j+1)],axis=None)
        phimap=np.copy(gmap2)
#       plt.figure(3)
#       plt.imshow(phimap,cmap='gnuplot')
```

```
    phimap[phimap>0]=1
    phimap[phimap<=15]=1
    phimap[gmap2==0]=0
    for gn in range(0,8):
        phimap[(gmap2<=4*(gn+1)) & (gmap2>4*gn)]=gn+1
    phimap[gmap2>=33]=8

#    f1 = interpolate.interp2d(xmap, ymap, gmap, kind='linear')
#    f2 = interpolate.interp2d(xmap, ymap, phimap, kind='linear')
#    gmap3 = f1(xmap2, ymap2)
#    phimap3 = f2(xmap2, ymap2)
    gmap2=gmap2[:,0:gmap2.shape[1]/2]
    phimap=phimap[:,0:phimap.shape[1]/2]
    xmap=xmap[:,0:xmap.shape[1]/2]
    ymap=ymap[:,0:ymap.shape[1]/2]
#    gmap3=gmap3[:,0:gmap3.shape[1]/2]
#    phimap3=phimap3[:,0:phimap3.shape[1]/2]
#    xmap2=xmap2[:,0:xmap2.shape[1]/2]
#    ymap2=ymap2[:,0:ymap2.shape[1]/2]
#    beta=1
#    alpha=5
#    betathresh=496
#    for gn in range(1,np.int(np.max(gmap2))+1):
#        if gn<=betathresh:
#            phimap[gmap2==gn]=beta
#            beta=beta+1
#            if beta==5:
#                beta=1
#        if gn>betathresh:
#            phimap[gmap2==gn]=alpha
#            alpha=alpha+1
#            if alpha==9:
#                alpha=5

#    phimap[gmap2==36]=9

    ebsd=np.zeros((gmap2.shape[0]*gmap2.shape[1],9))
    ebsd[:,0]=np.ones((1,gmap2.shape[0]*gmap2.shape[1]))*2.48663
    ebsd[:,1]=np.ones((1,gmap2.shape[0]*gmap2.shape[1]))*1.84098
    ebsd[:,2]=np.ones((1,gmap2.shape[0]*gmap2.shape[1]))*5.50548
    ebsd[:,3]=np.reshape(xmap,(1,gmap2.shape[0]*gmap2.shape[1]))
    ebsd[:,4]=np.reshape(ymap,(1,gmap2.shape[0]*gmap2.shape[1]))
    ebsd[:,6]=np.reshape(gmap2,(1,gmap2.shape[0]*gmap2.shape[1]))
    ebsd[:,7]=np.reshape(phimap,(1,gmap2.shape[0]*gmap2.shape[1]))
    ebsd[:,8]=np.ones((1,gmap2.shape[0]*gmap2.shape[1]))*0
    if plot==1:
        plt.figure(3)
        plt.imshow(phimap,cmap='gnuplot')
    grainnum=np.max(gmap2)+1
#    step=1/scale*coarsen
    widn=gmap2.shape[1]
    hein=gmap2.shape[0]
    step=height/hein
    voidvol=gmap2[gmap2==0]
    reduction=(widn*hein*1.0-voidvol.shape[0])/widn/hein
    area=phimap[phimap>0]
    beta=area[area<=4]
    alpha=area[area>4]
    betanew=(area.shape[0]*.04-alpha.shape[0]*0)/beta.shape[0]
    alphanew=(area.shape[0]*.06-beta.shape[0]*0)/alpha.shape[0]
    tinewalpha=1-(alphanew+0)
    tinewbeta=1-(betanew+0)
```

```
np.savetxt('//home//bjferg//projects//pfdb//'+'PFhalfC2input.txt',ebsd,delimiter=' ',header='Header: Moose Phase Field Input File \nDate: 12-Sep-2016 20:55:42 \nColumn 1: Euler angle "phi1" (in radians) \nColumn 2: Euler angle "PHI" (in radians) \nColumn 3: Euler angle "phi2" (in radians) \nColumn 4: x-coordinate (in microns) \nColumn 5: y-coordinate (in microns) \nColumn 6: z-coordinate (in microns) \nColumn 7: grain number (integer) \nColumn 8: phase number (integer) \nColumn 9: symmetry class (from TSL) \nPhase 1: Titanium Alpha (symmetry class = 62) \nPhase 2: Titanium Beta (symmetry class = 43) \nPhase 0: Void \nNumber of Grains: '+str(grainnum)+' \nX_Min: 0 \nX_Max: '+str(width)+'\nX_step: '+str(step)+'\nX_Dim: '+str(widn)+'\nY_Min: 0 \nY_Max: '+str(height)+'\nY_step: '+str(step)+'\nY_Dim: '+str(hein)+'\nZ_Min: 0 \nZ_Max: 0\nZ_step: 0\nZ_Dim: 0\n',fmt='%1.4f %1.4f %1.4f %1.4f %1.4f %1.4f %1d %1d %1d')
```

```
[Mesh]
  type = EBSDMesh
  filename = 'PFhalfinput.txt'
  uniform_refine=1
  elem_type=QUAD4
[]


[GlobalParams]
  op_num = '10'
  var_name_base = 'gr'
[]

[Variables]
  # concentration
  # chemical potential
  [./wTi]
    order = FIRST
    family = LAGRANGE
  [../]
  [./wAl]
    order = FIRST
    family = LAGRANGE
  [../]
  [./wV]
    order = FIRST
    family = LAGRANGE
  [../]
  [./PolycrystalVariables] # Automatically creates order parameter variables
  [../]
  [./Ti]
    order = FIRST
    family = LAGRANGE
  [../]
  [./Al]
    order = FIRST
    family = LAGRANGE
  [../]
  [./V]
    order = FIRST
    family = LAGRANGE
  [../]
  [./vTi]
    order = FIRST
    family = LAGRANGE
    initial_condition = 0#.01#.184
    #initial_condition =.184
    #scaling=1e8
  [../]
  [./vAl]
    order = FIRST
    family = LAGRANGE
    initial_condition = 0#.01#.012
    #initial_condition = .012
    #scaling=1e8
  [../]
  [./vV]
    order = FIRST
```

```
    family = LAGRANGE
  initial_condition = 0#.01#.008
  #initial_condition = .008
  #scaling=1e8
  [../]
  [./aTi]
    order = FIRST
    family = LAGRANGE
    initial_condition = .9#.91539#.46
    #initial_condition = .46
  #  scaling=1e6
  [../]
  [./aAl]
    order = FIRST
    family = LAGRANGE
    initial_condition = .06#.05797#.03
    #initial_condition = .03
    #scaling=1e6
  [../]
  [./aV]
    order = FIRST
    family = LAGRANGE
    initial_condition = .04#.02903#.02
    #initial_condition = .02
    #scaling=1e6
  [../]
  [./bTi]
    order = FIRST
    family = LAGRANGE
    initial_condition = .9#.76934#.736
    #initial_condition = .736
  [../]
  [./bAl]
    order = FIRST
    family = LAGRANGE
    initial_condition = .06#.15346#.048
    #initial_condition = .048
  [../]
  [./bV]
    order = FIRST
    family = LAGRANGE
    initial_condition = .04#.07720#.032
    #initial_condition = .032
  [../]
  # Lagrange multiplier
  [./lambda]
    order = FIRST
    family = LAGRANGE
  #  initial_condition = 1
    #scaling=1e-10
  [../]
  [./disp_x]
    order = FIRST
    family = LAGRANGE
  [../]
  [./disp_y]
    order = FIRST
    family = LAGRANGE
  [../]
[]
```

```
[BCs]

  [./gr0]
  type = NeumannBC
  variable = gr0
  boundary = 'left right top bottom'
  [../]
  [./gr1]
  type = NeumannBC
  variable = gr1
  boundary = 'left right top bottom'
  [../]
  [./gr2]
  type = NeumannBC
  variable = gr2
  boundary = 'left right top bottom'
  [../]
  [./gr3]
  type = NeumannBC
  variable = gr3
  boundary = 'left right top bottom'
  [../]
  [./gr4]
  type = NeumannBC
  variable = gr4
  boundary = 'left right top bottom'
  [../]
  [./gr5]
  type = NeumannBC
  variable = gr5
  boundary = 'left right top bottom'
  [../]
  [./gr6]
  type = NeumannBC
  variable = gr6
  boundary = 'left right top bottom'
  [../]
  [./gr7]
  type = NeumannBC
  variable = gr7
  boundary = 'left right top bottom'
  [../]
  [./gr8]
  type = NeumannBC
  variable = gr8
  boundary = 'left right top bottom'
  [../]
  [./gr9]
  type = NeumannBC
  variable = gr9
  boundary = 'left right top bottom'
  [../]
  [./Tif]
  type = NeumannBC
  variable = Ti
  boundary = 'left right top bottom'
  value=0
  [../]
#[./front_y]
  #  type = Pressure
  #  variable = disp_y
  #  boundary = top
```

```
  #  component=1
#    factor = -2e3
#  [../]
  [./back_y]
    type = PresetBC
    variable = disp_y
    boundary = bottom
    value = 0
  [../]
  [./front_y]
    type = PresetBC
    variable = disp_y
    boundary = top
    value = 0#.00027e-3
  [../]
  [.dispx]
    type=PresetBC
    variable=disp_x
    boundary='top bottom left right'
    value=0
  [../]
#  [./Periodic]
#  [./all]
#    variable = 'Al V aTi aAl aV bTi bAl bV vTi vAl vV'
#    auto_direction = 'x y'
#  [../]
#[../]
[]

[ICs]

#  [./PolycrystalICs]
  #  [./PolycrystalColoringIC]
  #    polycrystal_ic_uo = voronoi
  #  [../]
  #[../]
[./phase1_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 1
  variable = gr1
[../]
[./phase2_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 2
  variable = gr2
[../]
[./phase3_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 3
  variable = gr3
[../]
[./phase4_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 4
  variable = gr4
[../]
[./phase5_recon]
  type = ReconPhaseVarIC
```

```
  ebsd_reader = ebsd_reader
  phase = 5
  variable = gr5
[../]
[./phase6_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 6
  variable = gr6
[../]
[./phase7_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 7
  variable = gr7
[../]
[./phase8_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 8
  variable = gr8
[../]
[./phase9_recon]
type=ConstantIC
  #type = ReconPhaseVarIC
  #ebsd_reader = ebsd_reader
  #phase = 9
  variable = gr9
  value=0
[../]
[./phase0_recon]
  type = ReconPhaseVarIC2
  ebsd_reader = ebsd_reader
  phasemax = 8
  weights='1 0 0 0 0 0 0 0 0'
  variable = gr0
[../]

[./phaseTi_recon]
  type = ReconPhaseVarIC2
  ebsd_reader = ebsd_reader
  phasemax = 8
  weights='0 .9  .9 .9 .9 .9 .9 .9 .9 0'
  variable = Ti
[../]
[./phaseAl_recon]
  type = ReconPhaseVarIC2
  ebsd_reader = ebsd_reader
  phasemax = 8
  weights='0 .06  .06 .06 .06 .06 .06 .06 .06 0'
  variable = Al
[../]
[./phaseV_recon]
  type = ReconPhaseVarIC2
  ebsd_reader = ebsd_reader
  phasemax = 8
  weights='0 .04  .04 .04 .04 .04 .04 .04 .04 0'
  variable = V
[../]

[]
```

```
[Materials]
  # simple toy free energies
  [./fb]
    type = DerivativeParsedMaterial
    f_name = F1
    material_property_names='FHCP T'
    args = '  vTi aTi bTi vAl aAl bAl vV aV bV'
    #function = '1*((vTi-.92)^2+(vAl-.06)^2+(vV-.04)^2)'
    #function = '1*((vTi-.9)^2+(vAl-.06)^2+(vV-.04)^2)'
    function = '1e2*((vTi)^2+(vAl)^2+(vV)^2)'
    constant_names= 'kb'
    constant_expressions= '5.67e-8'
    #function = 'kb*T*((1-(vTi+vAl+vV))*log(1-(vTi+vAl+vV))+(vTi+vAl+vV)*log(vTi+vAl+vV))'
    derivative_order=2
    #compute=false
  [../]
  [./gtot]
    type = ParsedMaterial
    f_name = Gtot
    material_property_names='FHCP FBCC F1 halpha hbeta hvoid f_el_mat'
    function= 'FHCP*halpha+FBCC*hbeta+f_el_mat+F1*hvoid'
[./F_HCP_A3]
    type = DerivativeParsedMaterial
    block = 0
    f_name=GHCP
    material_property_names='f_el_mat halpha T Vm'
    args = 'aTi aAl aV '
    function = '(8.3145*T*(1.0*if(aAl > 1.0e-13,aAl*log(aAl),0)/(-0.5*1 + 1.5) + 1.0*if(aTi > 1.0e-13,aTi*log(aTi),0)/(-0.5*1 +
1.5) + 1.0*if(aV > 1.0e-13,aV*log(aV),0)/(-0.5*1 + 1.5) + 0.5*if(1 > 1.0e-13,1*log(1),0)/(-0.5*1 + 1.5)) + (aAl*1*if(T >= 298.15
& T < 2900.0,-1.8*T + if(T >= 298.15 & T < 700.0,-8.77664e-7*T^3 - 0.001884662*T^2 - 24.3671976*T*log(T) + 137.093038*T
- 7976.15 + 74092/T,if(T >= 700.0 & T < 933.473,-5.764227e-6*T^3 + 0.018531982*T^2 - 38.5844296*T*log(T) + 223.048446*T
- 11276.24 + 74092/T,if(T >= 933.473 & T < 2900.0,-31.748192*T*log(T) + 188.684136*T - 11278.361 - 1.230622e+28/T^9,0)))
+ 5481,0) + aTi*1*if(T >= 298.15 & T < 4000.0,if(T >= 298.15 & T < 900.0,1.06716e-7*T^3 - 0.004777975*T^2 -
23.9933*T*log(T) + 133.615208*T - 8059.921 + 72636/T,if(T >= 900.0 & T < 1155.0,-9.0876e-8*T^3 - 0.0042033*T^2 -
23.9887*T*log(T) + 132.988068*T - 7811.815 + 42680/T,if(T >= 1155.0 & T < 1941.0,2.02715e-7*T^3 - 0.0081465*T^2 -
14.9466*T*log(T) + 66.976538*T + 908.837 - 1477660/T,if(T >= 1941.0 & T < 4000.0,-3.04747e-7*T^3 + 0.008204849*T^2 -
87.2182461*T*log(T) + 638.806871*T - 124526.786 + 36699805/T,0)))),0) + aV*1*if(T >= 298.15 & T < 4000.0,2.4*T + if(T
>= 298.15 & T < 790.0,1.2175e-7*T^3 - 0.003098*T^2 - 24.134*T*log(T) + 133.346053*T - 7930.43 + 69460/T,if(T >= 790.0 &
T < 2183.0,-6.8e-7*T^3 + 6.25e-5*T^2 - 25.9*T*log(T) + 143.291093*T - 7967.842,if(T >= 2183.0 & T < 4000.0,-47.43*T*log(T)
+ 321.140783*T - 41689.864 + 6.44389e+31/T^9,0))) + 4000,0))/(-0.5*1 + 1.5) + (aAl*aTi*aV*1*(-1/3*aAl + (2/3)*aTi - 1/3*aV
+ 1/3)*if(T >= 298.15 & T < 6000.0,-40*T - 206074,0) + aAl*aTi*1*(aAl - aTi)^2*if(T >= 298.15 & T < 6000.0,-7756,0) +
aAl*aTi*1*(aAl - aTi)*if(T >= 298.15 & T < 6000.0,0.825*T - 3475,0) + aAl*aTi*1*if(T >= 298.15 & T < 6000.0,37.863*T -
134164,0) + aTi*aV*1*if(T >= 298.15 & T < 6000.0,13233,0))/(-0.5*1 + 1.5))'
  [../]
  [./F_BCC_A2]
    type = DerivativeParsedMaterial
    f_name=GBCC
    material_property_names='f_el_mat hbeta T Vm'
    args = 'bTi bAl bV '
    function = '(8.3145*T*(1.0*if(bAl > 1.0e-13,bAl*log(bAl),0)/(-3.0*1 + 4.0) + 1.0*if(bTi > 1.0e-13,bTi*log(bTi),0)/(-3.0*1 +
4.0) + 1.0*if(bV > 1.0e-13,bV*log(bV),0)/(-3.0*1 + 4.0) + 3.0*if(1 > 1.0e-13,1*log(1),0)/(-3.0*1 + 4.0)) + (bAl*1*if(T >= 298.15
& T < 2900.0,-4.813*T + if(T >= 298.15 & T < 700.0,-8.77664e-7*T^3 - 0.001884662*T^2 - 24.3671976*T*log(T) +
137.093038*T - 7976.15 + 74092/T,if(T >= 700.0 & T < 933.473,-5.764227e-6*T^3 + 0.018531982*T^2 - 38.5844296*T*log(T)
+ 223.048446*T - 11276.24 + 74092/T,if(T >= 933.473 & T < 2900.0,-31.748192*T*log(T) + 188.684136*T - 11278.361 -
1.230622e+28/T^9,0))) + 10083,0) + bTi*1*if(T >= 298.15 & T < 4000.0,if(T >= 298.15 & T < 1155.0,-2.78803e-7*T^3 -
0.000663845*T^2 - 25.5768*T*log(T) + 134.71418*T - 1272.064 + 7208/T,if(T >= 1155.0 & T < 1941.0,-8.4534e-7*T^3 +
0.00121707*T^2 - 22.3771*T*log(T) + 105.366379*T + 6667.385 - 2002750/T,if(T >= 1941.0 & T < 4000.0,1.228863e-6*T^3 -
0.02200832*T^2 + 19.0900905*T*log(T) - 182.426471*T + 26483.26 + 1400501/T,0))),0) + bV*1*if(T >= 298.15 & T <
4000.0,if(T >= 298.15 & T < 790.0,1.2175e-7*T^3 - 0.003098*T^2 - 24.134*T*log(T) + 133.346053*T - 7930.43 + 69460/T,if(T
>= 790.0 & T < 2183.0,-6.8e-7*T^3 + 6.25e-5*T^2 - 25.9*T*log(T) + 143.291093*T - 7967.842,if(T >= 2183.0 & T < 4000.0,-
47.43*T*log(T) + 321.140783*T - 41689.864 + 6.44389e+31/T^9,0))),0))/(-3.0*1 + 4.0) + (bAl*bTi*bV*1*(-1/3*bAl - 1/3*bTi
```

```
+ (2/3)*bV + 1/3)*if(T >= 298.15 & T < 6000.0,-150*T + 75972.5,0) + bAl*bTi*bV*1*(-1/3*bAl + (2/3)*bTi - 1/3*bV + 1/3)*if(T >= 298.15 & T < 6000.0,40*T - 113926,0) + bAl*bTi*bV*1*((2/3)*bAl - 1/3*bTi - 1/3*bV + 1/3)*if(T >= 298.15 & T < 6000.0,-100*T + 7315,0) + bAl*bTi*1*(bAl - bTi)^2*if(T >= 298.15 & T < 6000.0,400,0) + bAl*bTi*1*(bAl - bTi)*if(T >= 298.15 & T < 6000.0,4890,0) + bAl*bTi*1*if(T >= 298.15 & T < 6000.0,39.961*T - 132903,0) + bAl*bV*1*(bAl - bV)*if(T >= 298.15 & T < 6000.0,-43*T + 131633,0) + bAl*bV*1*if(T >= 298.15 & T < 6000.0,14.4*T - 136730,0) + bTi*bV*1*(bTi - bV)*if(T >= 298.15 & T < 6000.0,2025.39,0) + bTi*bV*1*if(T >= 298.15 & T < 6000.0,6523.17,0))/(-3.0*1 + 4.0))'
  [../]

  [./F_BCC_A2_parabolic]
    type = DerivativeParsedMaterial
    f_name=FBCC
    material_property_names='f_el_mat hbeta T Vm '
    constant_names= 'FBCC2Ti FBCCTi FBCC2Al FBCCAl FBCC2V FBCCV FBCCall'
    constant_expressions= '10349.1714593 -54797.6671001 74343.5758521 -130740.276508 81391.8335058 -80879.3122606 -54450.82070'
    args = 'bTi bAl bV '
    function = 'Vm*(1/2*FBCC2Ti*(bTi-.76934)^2+FBCCTi*(bTi-.76934)+1/2*FBCC2Al*(bAl-.15346)^2+FBCCAl*(bAl-.15346)+1/2*FBCC2V*(bV-.07720)^2+FBCCV*(bV-.07720)+FBCCall)+f_el_mat*hbeta'
  [../]
  [./F_HCP_A2_parabolic]
    type = DerivativeParsedMaterial
    f_name=FHCP
    material_property_names='f_el_mat halpha T Vm '
    constant_names= 'FHCP2Ti FHCPTi FHCP2Al FHCPAl FHCP2V FHCPV FHCPall'
    constant_expressions= '6695.36454023 -45204.0135888 170103.418816 -151347.949994 302311.555571 -60733.183779 -54450.82070'
    args = 'aTi aAl aV '
    function = 'Vm*(1/2*FHCP2Ti*(aTi-.91539)^2+FHCPTi*(aTi-.91539)+1/2*FHCP2Al*(aAl-.05797)^2+FHCPAl*(aAl-.05797)+1/2*FHCP2V*(aV-.02903)^2+FHCPV*(aV-.02903)+FHCPall)+f_el_mat*halpha'
  [../]

  # constant properties
  [./constants]
    type = GenericConstantMaterial
    prop_names  = 'L    MTi2 MAl2 MV2 T R Vm kappa Gall reduce mov betaalpha Mn'
    prop_values = '6e-3 10 10 10 1025 8.314 1e-13 1.09 -54450.82070 .92118 .1 1 1e12'
  [../]
  [./kappas]
    type = GenericConstantMaterial
    prop_names  = ' kappa11 kappa12 kappa13 kappa21 kappa22 kappa23 kappa31 kappa32 kappa33'
    prop_values = '   1.636      3.8e-15       3.8e-15       3.8e-15       3.8e-15       3.8e-15     3.8e-15       3.8e-15      3.8e-15     '
  [../]
  [./mobilityBCCTi]
    type = DerivativeParsedMaterial
    f_name = MBCCTi
    args = 'Ti Al V'
    material_property_names='T R'
    function=  'Ti*(-151989.95-127.37*T)+Al*(-196948.34-87.54*T)+V*(-329983.49-65.25*T)+Ti*V*(174277.28-62.90*T)*(Ti-V)^0+Ti*V*(-27605.20)*(Ti-V)^1+Ti*Al*V*1721802.68'
  [../]
  [./mobilityBCCAl]
    type = DerivativeParsedMaterial
    f_name = MBCCAl
    args = 'Ti Al V'
    material_property_names='T R'
    function=  'Al*(-215000-80.2*T)+Ti*(R*T*log(5.19e-10*exp(-98000/R/T)+5.51e-6*exp(-204000/R/T)))+V*(-268200-97.2*T)+Al*Ti*(-499946.15+333.87*T)+Al*Ti*(-407271.5+286.27*T)*(Al-Ti)+Al*V*(-751405.09)+Al*Ti*V*(-308901.62)'
  [../]
  [./mobilityBCCV]
    type = DerivativeParsedMaterial
    f_name = MBCCV
    args = 'Ti Al V'
```

```
  material_property_names='T R'
  function=                    'Al*(-325008.12-73.99*T)+Ti*(-179392.69-106.68*T)+V*(-325008.12-73.99*T)+Al*V*(-
1300707.37)+Ti*V*(159507.07-48.62*T)+Ti*V*(-10551.39)*(Ti-V)'
 [../]
 [./mobilityHCPTi]
  type = DerivativeParsedMaterial
  f_name = MHCPTi
  args = 'Ti Al V'
  material_property_names='T R'
  function=        'Al*(-79800+R*T*log(2.38e-5))+Ti*(-303000+R*T*log(1.35e-3))+V*(-258550+R*T*log(4.36e-2))+Al*Ti*(-
442800)'
 [../]
 [./mobilityHCPAl]
  type = DerivativeParsedMaterial
  f_name = MHCPAl
  args = 'Ti Al V'
  material_property_names='T R'
  function=             'Al*(-79800+R*T*log(2.38e-5))+Ti*(-193200+R*T*log(1e-8))+V*(-258550+R*T*log(4.36e-2))+Al*Ti*(-
491950)+Ti*V*587700'
 [../]
 [./mobilityHCPV]
  type = DerivativeParsedMaterial
  f_name = MHCPV
  args = 'Ti Al V'
  material_property_names='T R'
  function=  'Al*(-79800+R*T*log(2.38e-5))+Ti*(-251490-56*T)+V*(-258550+R*T*log(4.36e-2))+Al*Ti*(-529600)+Al*Ti*(-
130180)*(Al-Ti)'
 [../]
 [./mobilityTi]
  type = DerivativeParsedMaterial
  f_name = MTi
  material_property_names='MBCCTi MHCPTi hvoid T R Vm halpha hbeta  h1 h2 h3 h4 h5 h6 h7 h8 h9 Mn betaalpha'

#+10*(gr1*(gr2+gr3+gr4+gr5+gr6+gr7+gr8)+gr2*(gr3+gr4+gr5+gr6+gr7+gr8)+gr3*(gr4+gr5+gr6+gr7+gr8)+gr4*(gr5+gr6+gr7
+gr8)+gr5*(gr6+gr7+gr8)+gr6*(gr7+gr8)+gr7*gr8)
  function=
'Vm*Mn*(hbeta*exp(MBCCTi/R/T)/R/T+halpha*betaalpha*exp(MHCPTi/R/T)/R/T+hvoid*0/Vm*Mn+10*(exp(MBCCTi/R/T)
/R/T+exp(MHCPTi/R/T)/R/T)/2*(h1*(h2+h3+h4+h5+h6+h7+h8)+h2*(h3+h4+h5+h6+h7+h8)+h3*(h4+h5+h6+h7+h8)+h4*(h5+
h6+h7+h8)+h5*(h6+h7+h8)+h6*(h7+h8)+h7*h8)+15*(exp(MBCCTi/R/T)/R/T+exp(MHCPTi/R/T)/R/T)/2*hvoid*(halpha+hbet
a))'
 [../]
 [./mobilityAl]
  type = DerivativeParsedMaterial
  f_name = MAl
  material_property_names='MBCCAl MHCPAl hvoid T R Vm halpha hbeta  h1 h2 h3 h4 h5 h6 h7 h8 h9 Mn betaalpha'
  function=
'Vm*Mn*(hbeta*exp(MBCCAl/R/T)/R/T+halpha*betaalpha*exp(MHCPAl/R/T)/R/T+hvoid*0/Vm*Mn+10*(exp(MBCCAl/R/T
)/R/T+exp(MHCPAl/R/T)/R/T)/2*(h1*(h2+h3+h4+h5+h6+h7+h8)+h2*(h3+h4+h5+h6+h7+h8)+h3*(h4+h5+h6+h7+h8)+h4*(h5
+h6+h7+h8)+h5*(h6+h7+h8)+h6*(h7+h8)+h7*h8)+15*(exp(MBCCAl/R/T)/R/T+exp(MHCPAl/R/T)/R/T)/2*hvoid*(halpha+hb
eta))'
 [../]
 [./mobilityV]
  type = DerivativeParsedMaterial
  f_name = MV
  material_property_names='MBCCV MHCPV hvoid T R Vm halpha hbeta  h1 h2 h3 h4 h5 h6 h7 h8 h9 Mn betaalpha'
  function=
'Vm*Mn*(hbeta*exp(MBCCV/R/T)/R/T+halpha*betaalpha*exp(MHCPV/R/T)/R/T+hvoid*0/Vm*Mn+10*(exp(MBCCV/R/T)/
R/T+exp(MHCPV/R/T)/R/T)/2*(h1*(h2+h3+h4+h5+h6+h7+h8)+h2*(h3+h4+h5+h6+h7+h8)+h3*(h4+h5+h6+h7+h8)+h4*(h5+h
6+h7+h8)+h5*(h6+h7+h8)+h6*(h7+h8)+h7*h8)+15*(exp(MBCCV/R/T)/R/T+exp(MHCPV/R/T)/R/T)/2*hvoid*(halpha+hbeta))
'
 [../]
```

```
[./h1]
  type = SwitchingFunctionMultiPhaseMaterial
  h_name = h1
  all_etas = 'gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9'
  phase_etas = 'gr1'
[../]
[./h2]
  type = SwitchingFunctionMultiPhaseMaterial
  h_name = h2
  all_etas = 'gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9'
  phase_etas = 'gr2'
[../]
[./h3]
  type = SwitchingFunctionMultiPhaseMaterial
  h_name = h3
  all_etas = 'gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9'
  phase_etas = 'gr3'
[../]
[./h4]
  type = SwitchingFunctionMultiPhaseMaterial
  h_name = h4
  all_etas = 'gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9'
  phase_etas = 'gr4'
[../]
[./h5]
  type = SwitchingFunctionMultiPhaseMaterial
  h_name = h5
  all_etas = 'gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9'
  phase_etas = 'gr5'
[../]
[./h6]
  type = SwitchingFunctionMultiPhaseMaterial
  h_name = h6
  all_etas = 'gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9'
  phase_etas = 'gr6'
[../]
[./h7]
  type = SwitchingFunctionMultiPhaseMaterial
  h_name = h7
  all_etas = 'gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9'
  phase_etas = 'gr7'
[../]
[./h8]
  type = SwitchingFunctionMultiPhaseMaterial
  h_name = h8
  all_etas = 'gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9'
  phase_etas = 'gr8'
[../]
[./h9]
  type = SwitchingFunctionMultiPhaseMaterial
  h_name = h9
  all_etas = 'gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9'
  phase_etas = 'gr9'
[../]

# Switching functions for each phase
# hvoid(gr1, gr2, gr3)
[./hvoid]
  type = SwitchingFunctionMultiPhaseMaterial
  h_name = hvoid
  all_etas = 'gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9'
  phase_etas = 'gr0'
```

```
[../]

[./hbeta]
  type = SwitchingFunctionMultiPhaseMaterial
  h_name = hbeta
  all_etas = 'gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9'
  phase_etas = 'gr2 gr1 gr3 gr4'
[../]
[./halpha]
  type = SwitchingFunctionMultiPhaseMaterial
  h_name = halpha
  all_etas = 'gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9'
  phase_etas = ' gr5 gr6 gr7 gr8 gr9'
[../]

# Barrier functions for each phase
[./g0]
  type = BarrierFunctionMaterial
  g_order = SIMPLE
  eta = gr0
  function_name = g0
[../]
[./g1]
  type = BarrierFunctionMaterial
  g_order = SIMPLE
  eta = gr1
  function_name = g1
[../]
[./g2]
  type = BarrierFunctionMaterial
  g_order = SIMPLE
  eta = gr2
  function_name = g2
[../]
[./g3]
  type = BarrierFunctionMaterial
  g_order = SIMPLE
  eta = gr3
  function_name = g3
[../]
[./g4]
  type = BarrierFunctionMaterial
  g_order = SIMPLE
  eta = gr4
  function_name = g4
[../]
[./g5]
  type = BarrierFunctionMaterial
  g_order = SIMPLE
  eta = gr5
  function_name = g5
[../]
[./g6]
  type = BarrierFunctionMaterial
  g_order = SIMPLE
  eta = gr6
  function_name = g6
[../]
[./g7]
  type = BarrierFunctionMaterial
  g_order = SIMPLE
  eta = gr7
```

```
    function_name = g7
  [../]
  [./g8]
    type = BarrierFunctionMaterial
    g_order = SIMPLE
    eta = gr8
    function_name = g8
  [../]



  [./ElasticityTensor]
  type = ComputePolycrystalElasticityTensor2
  grain_tracker = grain_tracker
  length_scale=1e-6
  remap_grains=false
  [../]

  [./stress]
    type = ComputeLinearElasticStress
  [../]
  [./strain]
    type = ComputeSmallStrain
    displacements = 'disp_x disp_y'
  [../]
  [./elastic_free_energy_p]
    type = ElasticEnergyMaterial
    f_name = f_el_mat
    args = 'gr1 gr2 gr3 gr0 gr4 gr5 gr6 gr7 gr8 gr9'
  [../]
[]

[Kernels]
  [./Null]
    type=NullKernel
    variable=gr9
  [../]
  [./TensorMechanics]
    displacements = 'disp_x disp_y'
  [../]

  [./ACInterface0]
    type = ACInterface
    variable = gr0
    kappa_name = kappa11
  [../]
  [./ACInterface1]
    type = ACInterface
    variable = gr1
    kappa_name = kappa
  [../]
  [./ACInterface2]
    type = ACInterface
    variable = gr2
    kappa_name = kappa
  [../]
  [./ACInterface3]
    type = ACInterface
    variable = gr3
    kappa_name = kappa
  [../]
  [./ACInterface4]
```

```
  type = ACInterface
  variable = gr4
  kappa_name = kappa
[../]
[./ACInterface5]
  type = ACInterface
  variable = gr5
  kappa_name = kappa
[../]
[./ACInterface6]
  type = ACInterface
  variable = gr6
  kappa_name = kappa
[../]
[./ACInterface7]
  type = ACInterface
  variable = gr7
  kappa_name = kappa
[../]

# Kernels for Allen-Cahn equation for gr0
[./dgr0dt]
  type = TimeDerivative
  variable = gr0
[../]
[./ACBulkF0]
  type = KKSMultiACBulkF
  variable  = gr0
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g0
  eta_i     = gr0
  wi        = 132.0
  args      = '   gr2 gr3 gr4 gr1 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC0Ti]
  type = KKSMultiACBulkC
  variable  = gr0
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr0
  args      = 'gr3 gr2 gr4 gr1 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC0Al]
  type = KKSMultiACBulkC
  variable  = gr0
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr0
  args      = 'gr3 gr2 gr4 gr1 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC0V]
  type = KKSMultiACBulkC
  variable  = gr0
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr0
  args      = 'gr3 gr2 gr4 gr1 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
```

```
[./multipler0]
  type = MatReaction
  variable = gr0
  v = lambda
  mob_name = L
[../]
  # Kernels for Allen-Cahn equation for gr1
[./dgr1dt]
  type = TimeDerivative
  variable = gr1
[../]
[./ACBulkF1]
  type = KKSMultiACBulkF
  variable  = gr1
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g1
  eta_i     = gr1
  wi        = 3.3e1
  args      = '   gr2 gr3 gr4 gr0 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC1Ti]
  type = KKSMultiACBulkC
  variable  = gr1
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr1
  args      = 'gr2 gr3 gr4 gr0 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC1Al]
  type = KKSMultiACBulkC
  variable  = gr1
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr1
  args      = 'gr2 gr3 gr4 gr0 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC1V]
  type = KKSMultiACBulkC
  variable  = gr1
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr1
  args      = 'gr2 gr3 gr4 gr0 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]


[./multipler1]
  type = MatReaction
  variable = gr1
  v = lambda
  mob_name = L
[../]
# Kernels for Allen-Cahn equation for gr2
[./dgr2dt]
  type = TimeDerivative
  variable = gr2
```

```
[../]
[./ACBulkF2]
  type = KKSMultiACBulkF
  variable  = gr2
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g2
  eta_i     = gr2
  wi        = 3.3e1
  args      = '   gr1 gr3 gr4 gr0 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC2Ti]
  type = KKSMultiACBulkC
  variable  = gr2
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr2
  args      = 'gr1 gr2 gr3 gr4 gr0 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC2Al]
  type = KKSMultiACBulkC
  variable  = gr2
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr2
  args      = 'gr1 gr2 gr3 gr4 gr0 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC2V]
  type = KKSMultiACBulkC
  variable  = gr2
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr2
  args      = 'gr1 gr2 gr3 gr4 gr0 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]


[./multipler2]
  type = MatReaction
  variable = gr2
  v = lambda
  mob_name = L
[../]
# Kernels for Allen-Cahn equation for gr3
[./dgr3dt]
  type = TimeDerivative
  variable = gr3
[../]
[./ACBulkF3]
  type = KKSMultiACBulkF
  variable  = gr3
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g3
  eta_i     = gr3
  wi        = 3.3e1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC3Ti]
```

```
  type = KKSMultiACBulkC
  variable  = gr3
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr3
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC3Al]
  type = KKSMultiACBulkC
  variable  = gr3
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr3
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC3V]
  type = KKSMultiACBulkC
  variable  = gr3
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr3
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]


[./multipler3]
  type = MatReaction
  variable = gr3
  v = lambda
  mob_name = L
[../]
# Kernels for Allen-Cahn equation for gr4
[./dgr4dt]
  type = TimeDerivative
  variable = gr4
[../]
[./ACBulkF4]
  type = KKSMultiACBulkF
  variable  = gr4
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g4
  eta_i     = gr4
  wi        = 3.3e1
  args      = '  gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC4Ti]
  type = KKSMultiACBulkC
  variable  = gr4
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr4
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC4Al]
  type = KKSMultiACBulkC
  variable  = gr4
  Fj_names  = 'F1 FHCP FBCC'
```

```
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr4
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC4V]
  type = KKSMultiACBulkC
  variable  = gr4
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr4
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]

[./multipler4]
  type = MatReaction
  variable = gr4
  v = lambda
  mob_name = L
[../]
# Kernels for Allen-Cahn equation for gr5
[./dgr5dt]
  type = TimeDerivative
  variable = gr5
[../]
[./ACBulkF5]
  type = KKSMultiACBulkF
  variable  = gr5
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g5
  eta_i     = gr5
  wi        = 3.3e1
  args      = '  gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC5Ti]
  type = KKSMultiACBulkC
  variable  = gr5
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr5
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC5Al]
  type = KKSMultiACBulkC
  variable  = gr5
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr5
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC5V]
  type = KKSMultiACBulkC
  variable  = gr5
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr5
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
```

```
[../]

[./multipler5]
  type = MatReaction
  variable = gr5
  v = lambda
  mob_name = L
[../]
# Kernels for Allen-Cahn equation for gr6
[./dgr6dt]
  type = TimeDerivative
  variable = gr6
[../]
[./ACBulkF6]
  type = KKSMultiACBulkF
  variable  = gr6
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g6
  eta_i     = gr6
  wi        = 3.3e1
  args      = '  gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC6Ti]
  type = KKSMultiACBulkC
  variable  = gr6
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr6
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC6Al]
  type = KKSMultiACBulkC
  variable  = gr6
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr6
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC6V]
  type = KKSMultiACBulkC
  variable  = gr6
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr6
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]

[./multipler6]
  type = MatReaction
  variable = gr6
  v = lambda
  mob_name = L
[../]
# Kernels for Allen-Cahn equation for gr7
[./dgr7dt]
  type = TimeDerivative
  variable = gr7
[../]
```

```
[./ACBulkF7]
  type = KKSMultiACBulkF
  variable  = gr7
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g7
  eta_i     = gr7
  wi        = 3.3e1
  args      = '  gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC7Ti]
  type = KKSMultiACBulkC
  variable  = gr7
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr7
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC7Al]
  type = KKSMultiACBulkC
  variable  = gr7
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr7
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC7V]
  type = KKSMultiACBulkC
  variable  = gr7
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr7
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]

[./multipler7]
  type = MatReaction
  variable = gr7
  v = lambda
  mob_name = L
[../]


[./mult_lambda]
  type = MatReaction
  variable = lambda
  mob_name = 9
[../]
#[./dgr8dt]
#  type = TimeDerivative
#  variable = gr8
#[../]
[./ACBulkF8]
  type = KKSMultiACBulkF
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g8
  eta_i     = gr8
```

```
  wi       = 3.3e1
  mob_name  = 1
  args     = '  gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC8Ti]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr8
  mob_name  = 1
  args     = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC8Al]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr8
  mob_name  = 1
  args     = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./ACBulkC8V]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr8
  mob_name  = 1
  args     = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_CoupledACint_8]
  type = SimpleCoupledACInterface
  variable = lambda
  v = gr8
  kappa_name = kappa
  mob_name = 1
[../]
[./mult_ACBulkF7]
  type = KKSMultiACBulkF
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g7
  eta_i     = gr7
  wi       = 3.3e1
  mob_name  = 1
  args     = '  gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC7Ti]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr7
  mob_name  = 1
  args     = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
```

```
[./mult_ACBulkC7Al]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr7
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC7V]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr7
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_CoupledACint_7]
  type = SimpleCoupledACInterface
  variable = lambda
  v = gr7
  kappa_name = kappa
  mob_name = 1
[../]
[./mult_ACBulkF6]
  type = KKSMultiACBulkF
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g6
  eta_i     = gr6
  wi        = 3.3e1
  mob_name  = 1
  args      = '  gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC6Ti]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr6
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC6Al]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr6
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC6V]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
```

```
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr6
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_CoupledACint_6]
  type = SimpleCoupledACInterface
  variable = lambda
  v = gr6
  kappa_name = kappa
  mob_name = 1
[../]
[./mult_ACBulkF5]
  type = KKSMultiACBulkF
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g5
  eta_i     = gr5
  wi        = 3.3e1
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC5Ti]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr5
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC5Al]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr5
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC5V]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr5
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_CoupledACint_5]
  type = SimpleCoupledACInterface
  variable = lambda
  v = gr5
  kappa_name = kappa
  mob_name = 1
[../]
[./mult_ACBulkF4]
```

```
  type = KKSMultiACBulkF
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g4
  eta_i     = gr4
  wi        = 3.3e1
  mob_name  = 1
  args      = '  gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC4Ti]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr4
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC4Al]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr4
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC4V]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr4
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_CoupledACint_4]
  type = SimpleCoupledACInterface
  variable = lambda
  v = gr4
  kappa_name = kappa
  mob_name = 1
[../]
[./mult_ACBulkF3]
  type = KKSMultiACBulkF
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g3
  eta_i     = gr3
  wi        = 3.3e1
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC3Ti]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
```

```
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr3
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC3Al]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr3
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC3V]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr3
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_CoupledACint_3]
  type = SimpleCoupledACInterface
  variable = lambda
  v = gr3
  kappa_name = kappa
  mob_name = 1
[../]
[./mult_ACBulkF2]
  type = KKSMultiACBulkF
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g2
  eta_i     = gr2
  wi        = 3.3e1
  mob_name  = 1
  args      = '  gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC2Ti]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr2
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC2Al]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr2
  mob_name  = 1
```

```
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC2V]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr2
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_CoupledACint_2]
  type = SimpleCoupledACInterface
  variable = lambda
  v = gr2
  kappa_name = kappa
  mob_name = 1
[../]
[./mult_ACBulkF1]
  type = KKSMultiACBulkF
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g1
  eta_i     = gr1
  wi        = 3.3e1
  mob_name  = 1
  args      = '  gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC1Ti]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr1
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC1Al]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr1
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC1V]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr1
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9 Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_CoupledACint_1]
  type = SimpleCoupledACInterface
```

```
  variable = lambda
  v = gr1
  kappa_name = kappa
  mob_name = 1
[../]
[./mult_ACBulkF0]
  type = KKSMultiACBulkF
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid hbeta halpha'
  gi_name   = g0
  eta_i     = gr0
  wi        = 132.0
  mob_name  = 1
  args      = '  gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC0Ti]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vTi aTi bTi'
  eta_i     = gr0
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC0Al]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vAl aAl bAl'
  eta_i     = gr0
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_ACBulkC0V]
  type = KKSMultiACBulkC
  variable  = lambda
  Fj_names  = 'F1 FHCP FBCC'
  hj_names  = 'hvoid halpha hbeta'
  cj_names  = 'vV aV bV'
  eta_i     = gr0
  mob_name  = 1
  args      = ' gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8 gr9  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./mult_CoupledACint_0]
  type = SimpleCoupledACInterface
  variable = lambda
  v = gr0
  kappa_name = kappa11
  mob_name = 1
[../]

# Kernels for the penalty equation enforcing etas sum to 1

[./gr0reaction]
  type = MatReaction
  variable = gr8
  v = gr0
  mob_name = 1
[../]
```

```
[./gr1reaction]
  type = MatReaction
  variable = gr8
  v = gr1
  mob_name = 1
[../]
[./gr2reaction]
  type = MatReaction
  variable = gr8
  v = gr2
  mob_name = 1
[../]
[./gr3reaction]
  type = MatReaction
  variable = gr8
  v = gr3
  mob_name = 1
[../]
[./gr4reaction]
  type = MatReaction
  variable = gr8
  v = gr4
  mob_name = 1
[../]
[./gr5reaction]
  type = MatReaction
  variable = gr8
  v = gr5
  mob_name = 1
[../]
[./gr6reaction]
  type = MatReaction
  variable = gr8
  v = gr6
  mob_name = 1
[../]
[./gr7reaction]
  type = MatReaction
  variable = gr8
  v = gr7
  mob_name = 1
[../]
[./gr8reaction]
  type = MatReaction
  variable = gr8
  mob_name = 1
[../]
[./one]
  type = BodyForce
  variable = gr8
  value = -1.0
[../]
# Phase concentration constraints for material phases

[./null4]
type=NullKernel
variable=lambda
[../]
[./chempot11]
  type = KKSPhaseChemicalPotential
  variable = vTi
  cb       = aTi
```

```
  fa_name  = F1
  fb_name  = FHCP
  args_a='  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./chempot11b]
  type = KKSPhaseChemicalPotential
  variable = aTi
  cb       = bTi
  fa_name  = FHCP
  fb_name  = FBCC
  args_a='  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./chempot22]
  type = KKSPhaseChemicalPotential
  variable = vAl
  cb       = aAl
  fa_name  = F1
  fb_name  = FHCP
  args_a='  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./chempot22b]
  type = KKSPhaseChemicalPotential
  variable = aAl
  cb       = bAl
  fa_name  = FHCP
  fb_name  = FBCC
  args_a='  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./chempot33]
  type = KKSPhaseChemicalPotential
  variable = vV
  cb       = aV
  fa_name  = F1
  fb_name  = FHCP
  args_a='  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./chempot33b]
  type = KKSPhaseChemicalPotential
  variable = aV
  cb       = bV
  fa_name  = FHCP
  fb_name  = FBCC
  args_a='  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]



#lagrange multiplier constraints for material phases

#phase concentration equality



[./phaseconcentrationTi]
  type = KKSMultiPhaseConcentration
  variable = bTi
  cj = 'vTi aTi bTi'
  hj_names = 'hvoid halpha hbeta'
  etas = 'gr0 gr5 gr2'
  c = Ti
[../]
```

```
[./phaseconcentrationAl]
  type = KKSMultiPhaseConcentration
  variable = bAl
  cj = 'vAl aAl bAl'
  hj_names = 'hvoid halpha hbeta'
  etas = 'gr0 gr5 gr2'
  c = Al
[../]
[./phaseconcentrationV]
  type = KKSMultiPhaseConcentration
  variable = bV
  cj = 'vV aV bV'
  hj_names = 'hvoid halpha hbeta'
  etas = 'gr0 gr5 gr2'
  c = V
[../]
[./CHBulkTi]
  type = KKSSplitCHCRes
  variable = Ti
  ca       = aTi
  cb       = bTi
  fa_name  = FHCP
  fb_name  = FBCC
  w        = wTi
  h_name=halpha
  args_a='  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
[./CHBulkAl]
  type = KKSSplitCHCRes
  variable = Al
  ca       = aAl
  cb       = bAl
  fa_name  = FHCP
  fb_name  = FBCC
  w        = wAl
  h_name=halpha
  args_a='  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]
#split two phase cahn hilliard
[./CHBulkV]
  type = KKSSplitCHCRes
  variable = V
  ca       = aV
  cb       = bV
  fa_name  = FHCP
  fb_name  = FBCC
  w        = wV
  h_name=halpha
  args_a='  Ti Al V  vTi aTi bTi vAl aAl bAl vV aV bV'
[../]

[./dcdtTi]
  type = CoupledTimeDerivative
  variable = wTi
  v = Ti
[../]
[./dcdtAl]
  type = CoupledTimeDerivative
  variable = wAl
  v = Al
[../]
[./dcdtV]
```

```
    type = CoupledTimeDerivative
    variable = wV
    v = V
  [../]
  [./Tiparsed]
    type = SimpleCHInterface
    variable = Ti
    mob_name = MTi
    kappa_name = kappa_c
  [../]
  [./Alparsed]
    type = SimpleCHInterface
    variable = Al
    mob_name = MAl
    kappa_name = kappa_c
  [../]
  [./Vparsed]
    type = SimpleCHInterface
    variable = V
    mob_name = MV
    kappa_name = kappa_c
  [../]
  [./ckernelTi]
    type = SplitCHWRes
    mob_name = MTi
    variable = wTi
    args='vTi aTi bTi'
  [../]
  [./ckernelAl]
    type = SplitCHWRes
    mob_name = MAl
    variable = wAl
    args='vAl aAl bAl'
  [../]
  [./ckernelV]
    type = SplitCHWRes
    mob_name = MV
    variable = wV
    args='vV aV bV'
  [../]
[]
#[Functions]
#  [./penalty]
#    type = PiecewiseLinear
#    x = '0   3   5'
#    y = '0 0 -.01'
#  [../]
#[]
[AuxVariables]
  [./bnds]
  [../]
  [./unique_grains]
  order = CONSTANT
  family = MONOMIAL
  [../]
  [./var_indices]
  order = CONSTANT
  family = MONOMIAL
  [../]
  [./vonmises_stress]
  order = CONSTANT
  family = MONOMIAL
```

```
  [../]
  [./void111]
  order = CONSTANT
  family = MONOMIAL
  [../]
  [./freeE]
  order = CONSTANT
  family = MONOMIAL
  [../]
  [./mob]
  order = CONSTANT
  family = MONOMIAL
  [../]
  [./Estrain]
  order = CONSTANT
  family = MONOMIAL
  [../]
  [./Tirat]
  [../]
  [./Alrat]
  [../]
  [./Vrat]
  [../]

[]


[AuxKernels]
  [./BndsCalc]
    type = BndsCalcAux
    variable = bnds
    execute_on = 'initial timestep_end'
  [../]
  [./void111]
    type = RankFourAux
    variable = void111
    rank_four_tensor = elasticity_tensor
    index_l = 0
    index_j = 0
    index_k = 0
    index_i = 0
  [../]
  [./unique_grains]
  type = FeatureFloodCountAux
  variable = unique_grains
  execute_on = timestep_end
  flood_counter = grain_tracker
  field_display = UNIQUE_REGION
  [../]
  [./var_indices]
    type = FeatureFloodCountAux
    variable = var_indices
    execute_on = timestep_end
    flood_counter = grain_tracker
    field_display = VARIABLE_COLORING
  [../]
  [./vonmises_stress]
    type = RankTwoScalarAux
    variable = vonmises_stress
    rank_two_tensor = stress
    scalar_type = VonMisesStress
    execute_on = timestep_end
```

```
  [../]
  [./mobility]
   type = MaterialRealAux
   property = MTi
   variable = mob
   execute_on = 'initial timestep_end'
  [../]
  [./strainenergy]
   type = MaterialRealAux
   property = Gtot
   variable = Estrain
   execute_on = 'initial timestep_end'
  [../]
  [./Tiratio]
  type = ParsedAux
  args = 'Ti Al V gr0'
  function = 'if(Ti+Al+V>.05,Ti/(Ti+Al+V),Ti)'
  #function='(1-gr0)*(Ti/(abs(Ti)+abs(Al)+abs(V)))'
  variable = Tirat
[../]
[./Alratio]
type = ParsedAux
args = 'Ti Al V gr0'
#function = 'if(Ti+Al+V>.05,Ti/(Ti+Al+V),Al)'
function='(1-gr0)*(Al/(abs(Ti)+abs(Al)+abs(V)))'
variable = Alrat
[../]
[./Vratio]
type = ParsedAux
args = 'Ti Al V gr0'
#function = 'if(Ti+Al+V>.05,Ti/(Ti+Al+V),Al)'
function='(1-gr0)*(V/(abs(Ti)+abs(Al)+abs(V)))'
variable = Vrat
[../]
[./freenergy]
type=KKSMultiFreeEnergy
variable=freeE
hj_names='hvoid hbeta hbeta hbeta hbeta halpha halpha halpha halpha '
Fj_names='F1 FBCC FBCC FBCC FBCC FHCP FHCP FHCP FHCP '
w=3.3e1
gj_names='g0 g1 g2 g3 g4 g5 g6 g7 g8'
interfacial_vars='gr0 gr1 gr2 gr3 gr4 gr5 gr6 gr7 gr8'
kappa_names='kappa kappa kappa kappa kappa kappa kappa kappa kappa'
execute_on = 'initial timestep_end'
[../]
[]



[UserObjects]
  [./ebsd_reader]
    type = EBSDReader
  [../]
#  [./voronoi]
#    type = PolycrystalVoronoi
#    grain_num = 9 # Number of grains
#    coloring_algorithm = jp # bt will assign one grain to each op if they are the same
#    rand_seed = 1
#  [../]
  [./euler_angle_file]
    type = EulerAngleFileReader
    file_name = grn_36_rand_2D.tex
```

```
  [../]
  [./grain_tracker]
  type = GrainTrackerElasticity2
  compute_var_to_feature_map = true
  fill_method = symmetric9
 C_ijkl1 = '1e-3 1e-3 1e-3 1e-3 1e-3 1e-3 1e-3 1e-3 1e-3'
  C_ijkl2 = '153.11e-3 72.05e-3 72.05e-3 153.11e-3 72.05e-3 153.11e-3 40.53e-3 40.53e-3 40.53e-3'
  C_ijkl3 = '164.56e-3 77.44e-3 77.44e-3 164.56e-3 77.44e-3 164.56e-3 43.56e-3 43.56e-3 43.56e-3'
  euler_angle_provider = euler_angle_file
  remap_grains=false
  [../]
[]

[Executioner]
  type = Transient
  #solve_type = 'JFNK'
   solve_type = PJFNK
  petsc_options_iname = '-pc_type -pc_hypre_type -pc_hypre_boomeramg_strong_threshold'
  petsc_options_value = '  hypre    boomeramg                 0.7'
  l_max_its = 15 # Max number of linear iterations
  l_tol = 1e-10 # Relative tolerance for linear solves
  nl_max_its = 15 # Max number of nonlinear iterations
  nl_rel_tol = 1e-1 # Absolute tolerance for nonlienar solves
  #nl_abs_tol =1e-2

  end_time=10800
  #dt=.1
  [./TimeStepper]
  type = SolutionTimeAdaptiveDT
  dt =1e-2
  optimal_iterations=3
  [../]
  [./TimeIntegrator]
     #type = ImplicitEuler
     type = BDF2
    #type = CrankNicolson
    # type = ImplicitMidpoint
    # type = LStableDirk2
    # type = LStableDirk3
    # type = LStableDirk4
    # type = AStableDirk4
    #
    # Explicit methods
    #  type = ExplicitEuler
    # type = ExplicitMidpoint
    # type = Heun
    # type = Ralston
  [../]
[]
[Adaptivity]
max_h_level = 1
initial_steps=1
refine_fraction=0.2
coarsen_fraction=0.4
initial_marker=combo
marker=combo
[Markers]
  [./marker]
  type = ValueRangeMarker
  lower_bound = 0.05
  upper_bound = 0.95
  variable = gr0
```

```
  third_state = DO_NOTHING
  [../]
  [./marker1]
  type = ValueRangeMarker
  lower_bound = 0.05
  upper_bound = 0.95
  variable = gr1
  [../]
  [./marker2]
  type = ValueRangeMarker
  lower_bound = 0.05
  upper_bound = 0.95
  variable = gr2
  [../]
  [./marker3]
  type = ValueRangeMarker
  lower_bound = 0.05
  upper_bound = 0.95
  variable = gr3
  [../]
  [./marker4]
  type = ValueRangeMarker
  lower_bound = 0.05
  upper_bound = 0.95
  variable = gr4
  [../]
  [./marker5]
  type = ValueRangeMarker
  lower_bound = 0.05
  upper_bound = 0.95
  variable = gr5
  [../]
  [./marker6]
  type = ValueRangeMarker
  lower_bound = 0.05
  upper_bound = 0.95
  variable = gr6
  [../]
  [./marker7]
  type = ValueRangeMarker
  lower_bound = 0.05
  upper_bound = 0.95
  variable = gr7
  [../]
  [./marker8]
  type = ValueRangeMarker
  lower_bound = 0.05
  upper_bound = 0.95
  variable = gr8
  [../]
  [./combo]
  type = ComboMarker
  markers = 'marker marker1 marker2 marker3 marker4 marker5 marker6 marker7 marker8'
  [../]
[../]
[]

[Preconditioning]
  active = 'full'
  [./full]
    type = SMP
  #  full = true
```

```
  coupled_groups='Ti,wTi,vTi,aTi,bTi        Al,wAl,vAl,bAl,aAl        V,wV,vV,bV,aV        gr0,gr1,gr2,gr3,gr4,gr5,gr6,gr7,gr8,gr9
disp_x,disp_y'
  [../]
[]

[Outputs]
  file_base=flux2b2cdiffalla
  exodus=true

  [./debug2] # This is only a test, this should be turned on via the [Debug] block
    type = VariableResidualNormsDebugOutput
  [../]
[]
```

```
[Mesh]
  type = EBSDMesh
  filename = 'test2.txt'
[]
#[MeshModifiers]
#  [./scale]
#    type = Transform
#    transform = SCALE
#    vector_value = '1e-6 1e-6 1e-6'
#  [../]
#[]
[GlobalParams]
  op_num = '9'
  var_name_base = 'gr'
[]

[Variables]
  [./disp_x]
  [../]
  [./disp_y]
  [../]
  [./PolycrystalVariables] # Automatically creates order parameter variables
  [../]
[]

[AuxVariables]
  [./vel_x]
  [../]
  [./accel_x]
  [../]
  [./vel_y]
  [../]
  [./accel_y]
  [../]
  [./C1111]
    order = CONSTANT
    family = MONOMIAL
  [../]
  [./var_indices]
    order = CONSTANT
    family = MONOMIAL
  [../]
[]

[ICs]
[./phase1_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 1
  variable = gr1
[../]
[./phase2_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 2
  variable = gr2
[../]
[./phase3_recon]
  type = ReconPhaseVarIC
```

```
  ebsd_reader = ebsd_reader
  phase = 3
  variable = gr3
[../]
[./phase4_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 4
  variable = gr4
[../]
[./phase5_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 5
  variable = gr5
[../]
[./phase6_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 6
  variable = gr6
[../]
[./phase7_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 7
  variable = gr7
[../]
[./phase8_recon]
  type = ReconPhaseVarIC
  ebsd_reader = ebsd_reader
  phase = 8
  variable = gr8
[../]
[./phase0_recon]
  type = ReconPhaseVarIC2
  ebsd_reader = ebsd_reader
  phasemax = 8
  weights='1 0 0 0 0 0 0 0'
  variable = gr0
[../]
[]

[Kernels]
  [./DynamicTensorMechanics]
    displacements = 'disp_x disp_y'
    alpha = 0.3
  [../]
  [./inertia_x]
    type = InertialForce
    variable = disp_x
    velocity = vel_x
    acceleration = accel_x
    beta = 0.3025
    gamma = 0.6
    alpha = 0.3
  [../]
  [./inertia_y]
    type = InertialForce
    variable = disp_y
    velocity = vel_y
    acceleration = accel_y
```

```
    beta = 0.3025
    gamma = 0.6
    alpha = 0.3
  [../]
  [nullgr0]
  type = NullKernel
  variable = gr0
  []
  [nullgr1]
  type = NullKernel
  variable = gr1
  []
  [nullgr2]
  type = NullKernel
  variable = gr2
  []
  [nullgr3]
  type = NullKernel
  variable = gr3
  []
  [nullgr4]
  type = NullKernel
  variable = gr4
  []
  [nullgr5]
  type = NullKernel
  variable = gr5
  []
  [nullgr6]
  type = NullKernel
  variable = gr6
  []
  [nullgr7]
  type = NullKernel
  variable = gr7
  []
  [nullgr8]
  type = NullKernel
  variable = gr8
  []
[]

[AuxKernels]
  [./accel_x]
    type = NewmarkAccelAux
    variable = accel_x
    displacement = disp_x
    velocity = vel_x
    beta = 0.3025
    execute_on = timestep_end
  [../]
  [./vel_x]
    type = NewmarkVelAux
    variable = vel_x
    acceleration = accel_x
    gamma = 0.6
    execute_on = timestep_end
  [../]
  [./accel_y]
    type = NewmarkAccelAux
    variable = accel_y
    displacement = disp_y
```

```
    velocity = vel_y
    beta = 0.3025
    execute_on = timestep_end
  [../]
  [./vel_y]
    type = NewmarkVelAux
    variable = vel_y
    acceleration = accel_y
    gamma = 0.6
    execute_on = timestep_end
  [../]
  [./C1111]
    type = RankFourAux
    variable = C1111
    rank_four_tensor = elasticity_tensor
    index_l = 0
    index_j = 0
    index_k = 0
    index_i = 0
  [../]
  [./var_indices]
    type = FeatureFloodCountAux
    variable = var_indices
    execute_on = timestep_end
    flood_counter = grain_tracker
    field_display = VARIABLE_COLORING
  [../]
[]


[BCs]
  [./top_y]
    type = DirichletBC
    variable = disp_y
    boundary = top
    value=0.0
  [../]
  [./top_x]
    type = DirichletBC
    variable = disp_x
    boundary = top
    value=0.0
  [../]
  [./right_x]
    type = DirichletBC
    variable = disp_x
    boundary = right
    value=0.0
  [../]
  [./left_x]
    type = DirichletBC
    variable = disp_x
    boundary = left
    value=0.0
  [../]
  [./bottom_x]
    type = DirichletBC
    variable = disp_x
    boundary = bottom
    value=0.0
  [../]
  #[./bottom_y]
```

```
  #  type = FunctionPresetBC
  #  variable = disp_y
  #  boundary = bottom
  #  function = displacement_bc
  #[../]
  [./bottom_y]
    type = Pressure
    variable = disp_y
    boundary = bottom
    component=1
    function = displacement_bc
  [../]
[]

[Materials]
[./ElasticityTensor]
type = ComputePolycrystalElasticityTensor2
grain_tracker = grain_tracker
[../]
  [./strain]
    type = ComputeSmallStrain
    block = 0
    displacements = 'disp_x disp_y'
  [../]

  [./stress]
    type = ComputeLinearElasticStress
    store_stress_old = true
    block = 0
  [../]

  [./density]
    type = GenericConstantMaterial
    block = 0
    prop_names = 'density'
    prop_values = '4046e-24'
  [../]
[]

[Executioner]
type = Transient
#solve_type = 'PJFNK'
solve_type = 'NEWTON'
petsc_options_iname = '-pc_type -sub_pc_type   -sub_pc_factor_shift_type'
petsc_options_value = 'asm       ilu            nonzero'
l_max_its = 30
nl_max_its = 30
#l_tol = 1.0e-8
#nl_rel_tol = 1.0e-6
nl_abs_tol = 1.0e-4

  start_time = 0
  end_time = 20e-7
  dt = 2.5e-9
[]

[Functions]
  [./displacement_bc]
    type = PiecewiseLinear
    data_file = 'waveform.csv'
    format = columns
  [../]
```

```
[]

[UserObjects]
  [./ebsd_reader]
    type = EBSDReader
  [../]
  [./euler_angle_file]
    type = EulerAngleFileReader
    file_name = grn_36_rand_2D.tex
  [../]
  [./grain_tracker]
  type = GrainTrackerElasticity2
  compute_var_to_feature_map = true
  fill_method = symmetric9
 C_ijkl1 = '1e-3 1e-3 1e-3 1e-3 1e-3 1e-3 1e-3 1e-3 1e-3'
  C_ijkl2 = '153.11e-3 72.05e-3 72.05e-3 153.11e-3 72.05e-3 153.11e-3 40.53e-3 40.53e-3 40.53e-3'
  C_ijkl3 = '164.56e-3 77.44e-3 77.44e-3 164.56e-3 77.44e-3 164.56e-3 43.56e-3 43.56e-3 43.56e-3'
 euler_angle_provider = euler_angle_file
  remap_grains=false
  [../]
[]

[Postprocessors]
  [./_dt]
    type = TimestepSize
  [../]
  [./nodal_sum]
  type = NodalSum
  variable = disp_y
  boundary=bottom
  execute_on = 'initial timestep_end'
[../]
[]

[Outputs]
  exodus = true
  print_perf_log = true
  file_base=ultrasonic_square_novoid2
  #[./debug2] # This is only a test, this should be turned on via the [Debug] block
  #  type = VariableResidualNormsDebugOutput
  #[../]
[]
```

Bryan Ferguson

5267 Shilshole Ave NW – 253-245-9150 – bjferg@uw.edu

Skills

Experimental

Metallurgical sample preparation and analysis, SEM, EDX, microhardness, ultrasonic inspection, machining and structural testing, alloy preparation, XRD, electrochromic and dye sensitized solar cell fabrication.

Theoretical

Development of models in python, matlab, C++, Fortran, ANSYS, ABAQUS, MooseFramework, LAMMPS, and Gaussian. Creating inventive models using analytical equations, statistical methods, computer vision, finite element methods, phase field, molecular dynamics, and SCF/DFT ab initio calculations.

Research Experience

Superplastic Forming and Diffusion Bonding of Titanium – PI: Ramulu Mamidala

2015 - Current

Worked in conjunction with Boeing on an Air Force Metals Affordability Initiative with the goal of providing scientific understanding of a four-sheet SPF/DB titanium engine component. Work includes characterizing a complex process by preparing and analyzing fabricated specimens for bonding quality and microstructure, creating programs to detect features in micrographs to get a robust experimental dataset, and analysis using SEM and EDX. New structural tests were developed using FEM and experiments. Theoretical work includes developing models with a variety of methods including statistically implementing analytical equations, phase field, FEM, and molecular dynamics.

Ferromagnetic Shape Memory Alloy Orthodontic Device – PI: Minuru Taya

2013 – 2015

Designed a unique, magnetically actuated, ferromagnetic shape memory alloy orthodontic bracket in cooperation with a California based company. Scope included designing a long range magnetic actuator and a high strength bracket that could open and close under a magnetic field. Work included FePb synthesis, XRD analysis, structural FEM, and coupled magnetic and structural multiphysics.

Dye Sensitized Solar Cell Electrochromic Window – PI: Minuru Taya

2011 – 2012

Research into the design and fabrication of a combination dye sensitized solar cell and electrochromic window. Used density functional theory to compute absorption bands in different charge states to select potential dyes.

Work Experience

Ferguson Power Systems
2008-2010
Started a solar power company with a contract for producing power in California. Designed, tested, and prototyped ideas in an effort to find a solar thermal power solution to the contract requirements.

Process Engineer at Certainteed
2006-2008
Process engineer at a roofing manufacturing company. Designed and oversaw construction of a custom cooling system.

Education

M.S. in Mechanical Engineering
University of Washington, Seattle, WA
June 2014

B.S. in Mechanical Engineering
University of Portland, Portland, OR
June 2006

Ph.D. in Mechanical Engineering
University of Washington, Seattle, WA
Graduating Winter 2020

M.S. in Applied Mathematics
University of Washington, Seattle, WA
Graduating December 2019 (all credits currently completed)

References

Dr. Ramulu Mamidala
Professor of Mechanical Engineering
University of Washington
ramulum@uw.edu

Dr. Daniel Sanders
Senior Technical Fellow
The Boeing Company
Daniel.g.sanders@boeing.com

Publications and Conferences

B. Ferguson and M. Ramulu, "Preliminary Results for Dissimilar Titanium Alloy Diffusion Bonding Using a Surface Roughness Finite Element Model" ASM International Aeromat 2017

Ferguson, Bryan, and M. Ramulu. "Surface tracking of diffusion bonding void closure and its application to titanium alloys." International Journal of Material Forming (2019): 1-15.

Ferguson, Bryan, and M. Ramulu. " Investigation of the Complexities Inherent in Manufacturing Near-Unconstrained Superplastic Parts by Experiments and Simulation." In Preparation

Ferguson, Bryan, and M. Ramulu. " Structural Testing of Diffusion Bonding: Simulation and Experimental Verification." In Preparation

Ferguson, Bryan, and M. Ramulu. " Effects of Materials and Dissimilarity on Titanium Diffusion Bonding." In Preparation

Ferguson, Bryan, and M. Ramulu. " Improved Stochastic Dissimilar Diffusion Bonding Model with Experimental Validation." In Preparation

Ferguson, Bryan, and M. Ramulu. "Phase Field Simulation of Diffusion Bonding and the Effects of Crystal Texture on Bonding Performance." In Preparation