D2.1
Weld formation models

General information	
Grant agreement number	755500
Start date of the project	01/09/2017
Project duration	54 months
Due date of the deliverable	28/02/2021
Actual submission date	05/01/2022
Lead beneficiary	N°8 – BZN

Keywords
Dissimilar metal weld, weld formation, welding simulation, thermal simulation, SPPARKS

Type	Meaning	
R	Document, report	x
DEM	Demonstrator, pilot, prototype	
DEC	Websites, patent fillings, videos, etc.	
OTHER	Software, technical diagram, etc.	

Dissemination Level		
PU	Public	x
CO	Confidential, only for members of the consortium (including the Commission Services)	

TABLE OF CONTENT

Table of Tables

Table of Figures

1 Summary

The main objective of this document is to present the models developed by BZN for the prediction of grain orientation during welding. This task includes the results of the developed code using commonly used finite element (FE) models as well as the relatively new modelling approaches based on Kinetic Monte Carlo (MC) technique. The grain orientation under various welding conditions was simulated and compared with the corresponding experimental data. On the macro-scale a single and multi-physics FE model is utilized to predict the thermal history of the molten pool during solidification and parallel on the meso-scale a 3D MC model has been developed to predict the competitive grain growth on the micro-scale and the numerical results are integrated into the Kinetic Monte Carlo algorithm. The macro-scale model provides reasonable predictions for the grain orientation after the solidification, which is found to be profoundly affected by the welding parameters.

The modelling of the welding process is essential to understanding the weld pool formation. In welding, the heat source interacts with the material, resulting in three distinct regions in the weldment. These are the fusion zone. The fusion zone is created by heating above the melting point during the welding process. The weld microstructure development in the fusion zone is more complicated because of physical processes that occur due to the interaction of the heat source with the metal during welding, including re-melting, heat and fluid flow, vaporization, dissolution of gasses, solidification, subsequent solid-state transformation, stresses, and distortion. These processes and their interactions profoundly affect weld pool solidification and microstructure. Using correct heat source models used in finite element analysis (FEA) is a common problem as the variation of welding variables and parameters have a large influence on the weld pool shape and size.

The aim of the current research program is to develop a welding process model for austenitic weldment. A numerical study is performed focusing on the effect of the uncertainly of the welding parameters and its effect on the size of fusion zone, grain orientation and grain distribution during welding. During the solidification process, the austenitic phase forms long columnar grains which grow along the directions of maximum heat loss during cooling. Generally, during the process of welding, beads are produced, in which grains grow along the maximum thermal gradient when cooling. This approach is sufficient to determine a reasonably realistic grain orientation. The approach is implemented in the used FE code.

2 Introduction

Several approaches have been proposed in the literature to model weld microstructures considering specific simplifying hypotheses in grains geometry or orientation [1]-[10]. One method is based on molecular dynamics theory. Molecular dynamics simulations give information at the atomic scale. This includes a comprehensive tracking of all atomic positions and velocities from which detailed defect structures and system thermodynamics can be calculated. This approach has been successfully used to determine diffusivities that can be incorporated into meso-scale simulation techniques such as kinetic Monte Carlo [11]-[15].

The finite element method is useful for solving problems in solid mechanics, heat transport, and other fields. However, finite element simulation is primarily a continuum technique and is limited to suitability, by itself, to simulate the microstructural mass transfer and evolution inherent in the welding process. Therefore, a method situated between the scope of molecular dynamics and finite element methods is required. Kinetic Monte Carlo is an ideally suited simulation technique for modelling grain growth under welding. Nevertheless, the austenitic phase forms long columnar grains which grow along the directions of maximum heat loss during welding. Generally, the grains grow along the maximum thermal gradient when cooling. The adaptation of this approach to the commercial FE code is not too difficult and it is more than adequate to determine one a realistic grain orientation and size.

The evolution of grain orientation and size during solidification of a welding pool is simulated by considering heat transfer from the liquid to the base metal in the molten pool and solidification parameters. The developed efficient 2D and 3D finite element model consists of only the heat transfer simulation to predict the molten pool shape and temperature field and a developed model of grain formation in the molten pool. In an arc weld, a highly localized concentrated energy source fuses the filler metal and nearby base metal to form a weld pool. The shape and temperature distribution of a weld pool are dependent on the type of welding processes, welding parameters (welding velocity, diameter of wire, wire feed rate, voltage, current density, angle of workpiece inclination, etc.) and the weld materials (heat transfer, specific heat, latent heat). One of the major challenges of computational weld mechanics is to simulate the geometry and temperature distribution of a weld pool accurately and efficiently.

There are several approaches to model the effects of the heat source on a weld pool. One of the simpler ways is to prescribe either the thermal flux or power density distribution, or the temperature distribution in the weld pool by a distribution function that intends to model the heat input. The double-ellipsoid model provided by Goldak et al. is the most realistic model for welding with simple shapes. This model is convenient, accurate and efficient in an application of a welding process as long as the geometry of the liquid–air surface of the molten pool does not want to be considered.

In this study, 3-D macro and multi-scale models were developed to study the grain formation in the SMAW, GTAW and SAW welding process with consideration of the parameters which modifies the shape of the weld pool. The multi-scale model is composed of two sub-models on different scales. On the macro-scale, a thermal and coupled thermo-mechanical model is used to simulate the evolution of the molten pool. The model provides information about the molten pool shape. Such information is first applied to a meso-scale MC model to simulate the competitive grain growth in the molten pool and predict the grain distribution in the entire final welds. Both 2D and 3D simulations were investigated, and experimental measurements were performed on WP2 samples to validate the model. The 2D simulation results were also compared with the existing MINA code.

3 Finite Element Analysis

A complex finite element model has been developed in MSC.Marc software to simulate welding processes. Thermal and coupled transient thermo-mechanical analyses are performed, which is a comprehensive technique for welding simulations used to determine and evaluate the temperature fields on the molten pool and grain orientation and grain size. The most important features of the method implemented in the code are presented below.

3.1 Estimation of the welding parameters

The first step of the simulation was the determination of the welding parameters, which are essential for the numerical analysis. For this, a WP2 weld sample was made. The examined weld can be seen in Figure 1. Based on the received welding report it is built up from 12 layers with different numbers of welds on the given layers:

- 1-4 layers: One weld
- 5-7 layers: Two weld
- 8-10 layers: Three weld
- 11-12 layers: Four weld

The detailed welding report can be seen in Figure 2, it includes welding position, the filler materials and some of the welding parameters, the calculation of the missing values is shown in the following.

Figure 1: Welding sample

Welding Report

Page
of

Weld no.: ADVISE Sample 01
Dimension: 517x39
Material: 1.4550

Buildup sequence see additional sheet

Weld Sequence	Welding Process / Welding Position	Number of Layer / Pass No.	Filler material: Designation / Size	Welding Consumables: Shielding Gas l/min	Welding Consumables: Purge Gas l/min	Welding Consumables: Gas Nozzle Size No.	Welding Parameter: Kind of current / Polarity	Welding Parameter: I [A] / U [V]	Welding Parameter: Electrode runout length mm	Welding Parameter: Burn-off Time s	Welding Parameter: Inclination Electrode °	Preheat Temperature / Interpass Temperature [°C]	Welder Name	Comments	Date
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	141	1	EAS 2-IG	Argon	-	6	DC	120	40-60	60	60-80		Skowron		
	PA	1	ø 2.4 mm	10	-		-					25		Root Pass	2018
2	111	2	ESAB OK 61.30	-	-	-	DC	140	130	60	60-80		Skowron		
	PA	2	ø 4.0 mm	-	-		+					25		Filler Pass	2018
2	111	3	ESAB OK 61.30	-	-	-	DC	180	160	65	60-80		Skowron		
	PA	3	ø 5.0 mm	-	-		+					40		Filler Pass	2018
2	111	4	ESAB OK 61.30	-	-	-	DC	180			60-80		Skowron		
	PA	4	ø 5.0 mm	-	-		+					65		Filler Pass	2018
2	111	5	ESAB OK 61.30	-	-	-	DC	180			60-80		Skowron		
	PA	5	ø 5.0 mm	-	-		+					80		Filler Pass	2018
2	111	5	ESAB OK 61.30	-	-	-	DC	180			60-80		Skowron		
	PA	6	ø 5.0 mm	-	-		+					90		Filler Pass	2018
2	111	6	ESAB OK 61.30	-	-	-	DC	180			60-80		Skowron		
	PA	7	ø 5.0 mm	-	-		+					70		Filler Pass	2018
2	111	6	ESAB OK 61.30	-	-	-	DC	180			60-80		Skowron		
	PA	8	ø 5.0 mm	-	-		+					85		Filler Pass	2018
2	111	7	ESAB OK 61.30	-	-	-	DC	180			60-80		Skowron		
	PA	9	ø 5.0 mm	-	-		+					95		Filler Pass	2018
2	111	7	ESAB OK 61.30	-	-	-	DC	180			60-80		Skowron		
	PA	10	ø 5.0 mm	-	-		+					100		Filler Pass	2018
2	111	8	ESAB OK 61.30	-	-	-	DC	180			60-80		Skowron		
	PA	11	ø 5.0 mm	-	-		+					105		Filler Pass	2018
2	111	8	ESAB OK 61.30	-	-	-	DC	180			60-80		Skowron		
	PA	12	ø 5.0 mm	-	-		+					105		Filler Pass	2018

Figure 2: Received Welding Report

The missing welding parameters can be calculated in two ways, the first is the estimation according to the standard and the second is the calculation of the welding parameters based on the welding report. During the welding of the sample two processes were used the gas tungsten arc welding (GTAW) and the shielded metal arc welding (SMAW). Based on the [17] reference the following equations can be used for the calculation of the voltage. In case of GTAW:

$$U = 13 + 0.012I, \qquad (1)$$

where the maximal welding speed must be smaller than $v = 4$ mm/s, and efficiency of this process is $\eta = 0.6$.

The voltage for SMAW is calculated with the following equation:

$$U = 20 + 0.04I \qquad (2)$$

Here the maximal weld speed is the same as in the case of GTAW, and the efficiency is higher: $\eta = 0.8$. From this, the welding heat input can be calculated with the following equations:

$$P = \eta UI \qquad (3)$$

Based on these equations the value of the voltage for the different current values can be calculated. The results are included in Table 1.

Filler Material	Current	Voltage	Welding Power
EAS 2-IG	120 A	14.44 V	1039.68 W
ESAB OK 61.30	140 A	25.6 V	2867.2 W
ESAB OK 61.30	150 A	26 V	3120 W
ESAB OK 61.30	180 A	27.2 V	3916.8 W

Table 1: Calculated welding parameters

With the given electrode runout length and the burn-off time the speed of the welding can be calculated, and with the help of it, the weld heat input also can be determined. The relationship between the cross-sectional area of weld filler metal introduced per pass and the electrical heat input per unit length is known for MMA welds in austenitic steels, it is shown in Figure 3.

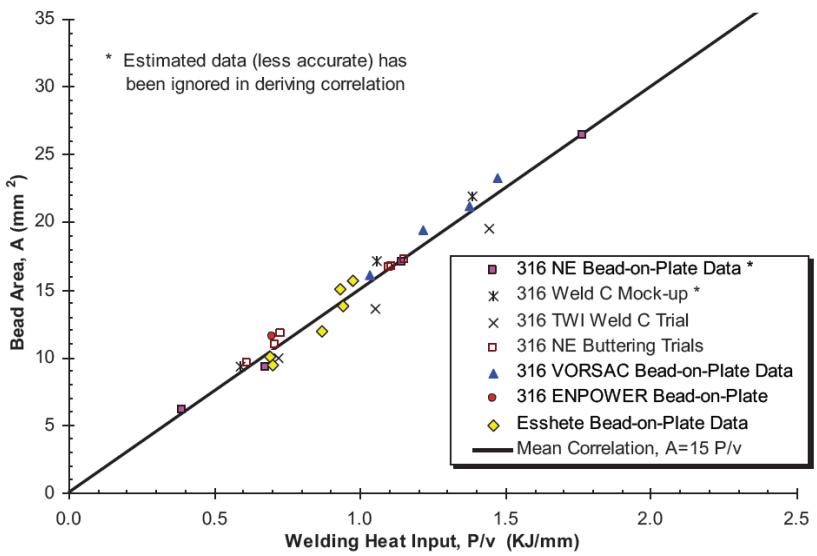


Figure 3: Electrical heat input vs weld filler metal correlation for AISI Type 316 and Esshete 1250 stainless steel MMA weld metal [19]

Based on this diagram the welding heat input and the nugget areas for the different values of current can be estimated. These values are summarized in Table 2. It has to be noted that the equation given in reference [19] does not take into account the efficiency of the welding, therefore the real nugget areas are given in the last column of the table.

Filler Material	Electrode runout length [mm]	Burn-off Time [s]	Welding speed [mm/s]	Welding Heat input [kJ/mm]	Nugget Area from diagram [mm^2]	Real Nugget area [mm^2]
EAS 2-IG	50	60	0.83	1.248	18.714	31.19
ESAB OK 61.30	130	60	2.17	1.323	19.85	24.812
ESAB OK 61.30	125	60	2.08	1.498	22.464	28.08
ESAB OK 61.30	150	60	2.5	1.248	18.72	23.4
ESAB OK 61.30	160	65	2.46	1.591	23.868	29.835

Table 2: Calculated nugget areas based on the welding parameters

Another way of calculating the nugget areas is from the rate of the welded volume and the electrode length. For the welding two types of the electrode was used, the EAS 2-IG and the ESAB OK 61.30. Based on reference [18] the length of the ESAB OK 61.30 electrode is 350 mm, however during welding 50 mm of it are not used. Therefore, from these data the estimated volume and the nugget area is shown in Table 3.

Column 1	Current [A]	Diameter [mm]	Electrode Length [mm]	Electrode Runout length [mm]	Volume [mm^3]	Nugget area [mm^2]
EAS 2-IG	120	2.4	-	50	-	- *
ESAB OK 61.30	140	4	300	130	3769.91	28.999
ESAB OK 61.30	150	4	300	125	3769.91	30.159
ESAB OK 61.30	150	4	300	150	3769.91	25.133
ESAB OK 61.30	180	5	300	160	5890.47	36.816

Table 3: Calculated nugget areas based on the size of the electrode *determined from full welded area

Based on which method is used there is almost a 20% difference in the values of the nugget areas, therefore without exact welding parameters there is a high degree of uncertainty.

3.2 Parameters of Welding Spots

In case of welding simulations, the most widely used mathematical model was created by J. Goldak et al, where the weld heat sources are based on a Gaussian distribution of power density in space [20]. They proposed a double ellipsoidal geometry, this way the arc welding processes and the laser, electron beam processes can also be modelled with the changing of the size and shape of the heat source.

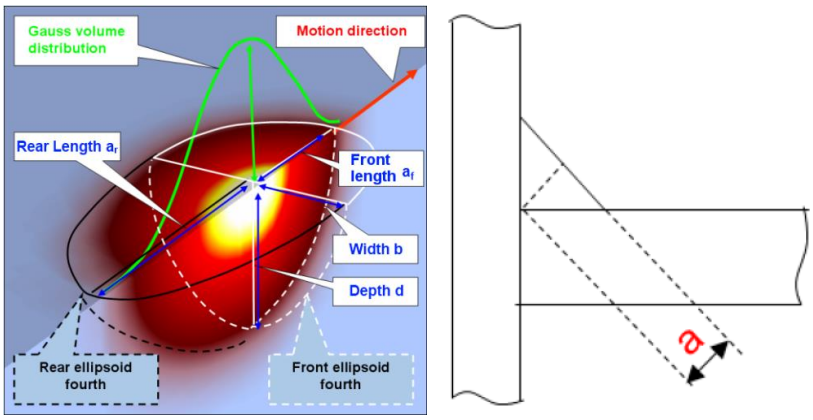


Figure 4: Heat source model (left) design throat (right)

The first concept for welding simulations was the usage of an ellipsoidal heat source, however the results were not consistent with the experimental results, the temperature gradient in front of the heat source was not as steep as expected. Therefore, in the new model the combination of two ellipsoidal sources was suggested. The geometry parameters and the Gauss volume distribution can be seen in Figure 4. The power density distribution for the front quadrant can be written down with the following equation [20]:

$$q(x,y,z,t) = \frac{6\sqrt{3}f_f Q}{ba_f d\pi\sqrt{\pi}} e^{-3x^2/b^2}\, e^{-3y^2/d^2} e^{-3[z+v(\tau-t)]^2/{a_f}^2} \qquad (4)$$

In case of the rear quadrant a similar equation can be used for the calculation of the power density distribution:

$$q(x,y,z,t) = \frac{6\sqrt{3}f_r Q}{ba_r d\pi\sqrt{\pi}} e^{-3x^2/b^2}\, e^{-3y^2/d^2} e^{-3[z+v(\tau-t)]^2/{a_r}^2}, \qquad (5)$$

where

f_f is the fraction of the heat deposited in the front quadrant,

f_r is the fraction of the heat deposited in the rear quadrant,

Q energy input rate [W],

τ lag factor for defining the position of the source at time $t = 0$,

v is the welding speed [m/s],

a_f is the front length of the heat source,

a_r is the rear length of the heat source,

b is the width of the heat source,

d is the depth of the heat source.

Here the equations are written up in a fixed (x, y, z) coordinate system. The transformation between the fixed and moving coordinate systems is the following:

$$\xi = z + v(\tau - t) \qquad (6)$$

From these equations it can be seen that for the simulation of welding the front length, the rear length, the width and the depth of the heat source has to be known.

The depth and the width can be identified from a micrograph, the other values can be estimated with the following rules:

$$a_f = 0.6b \qquad (7)$$

$$a_r = 2.0 \dots 2.5b \qquad (8)$$

If a first estimation is based on a construction plan, then the so-called design throat, can be used for the estimation of the other geometrical parameters (Figure right side). In this case the depth and the width of the heat source is given with the following simple equation:

$$d = a + 2.0 \dots 5.0\ mm \qquad (9)$$

$$b = a + 1 \dots 2\ mm \qquad (10)$$

The welding parameters which have to be included in a welding simulation are shown in Figure 5.

Figure 5: Flux parameters in Marc Mentat

3.3 Welding simulation

Several 2D FE models have been developed in MSC.Marc software to simulate the grain orientations. The original theory which considers the mechanical effect is used to simulate the grain orientation during the welding of the presented WP2 sample.

The coupling of the physics means that the calculated temperature gradient field during the welding is modified by the locally deformation gradient, because the mechanical effect must be considered. However, the coupled simulation is very time consuming in case of 3D, therefore only 2D analyses were performed.

It is well known that the shape and temperature distribution of a weld pool is dependent on the type of welding processes, welding parameters (welding velocity, diameter of the wire, wire feed rate, voltage, current density, angle of workpiece inclination, etc.) and the weld materials (heat transfer, specific heat, latent heat), therefore systematic sensitivity analysis were performed to find the critical parameters, which need to be known to predict the most realistic grain orientation distribution.

The determination of the welding parameters was presented below, the unknown thermal and mechanical material parameters which were applied during the simulations were determined using the JMatPro software based on the chemical composition of the used material.

In the current coupled simulations, quadrilateral solid elements are used in the finite element model. Thermal boundary conditions are defined for heat flow calculations. The initial temperature (room temperature or interpass temperature) of nodes is specified before the first load step of the thermo-mechanical analysis. Nodal temperatures of not yet deposited weld passes are prescribed in the first step of the calculation to avoid ill-conditioned matrices. A combined temperature-dependent heat transfer coefficient was used during the simulation to model the effect of convection and radiation to ambient air. The film coefficient is assumed to be 20 $Wm^{-2}K^{-1}$, while emissivity is taken as a temperature-independent value with a magnitude of 0.8. On the other hand, moving volumetric heat sources induce heat generation which is defined as element body force load during the transient thermo-mechanical analysis. The double ellipsoidal heat source model is implemented in the current analysis.

The sensitivity analysis focuses on the effect of parameters which modifying the shape of the weld pool. Solely Goldak's s double ellipsoidal heat source model is assumed in the calculations, as the only aim of the sensitivity analysis is to evaluate the magnitude of differences in weld pool sizes. Besides that, the effect of the mesh size was also analysed to find the ideal mesh distribution for later calculations.

In one part of the welding simulations the received geometry was used, the layout of the welding can be seen in Figure 6.

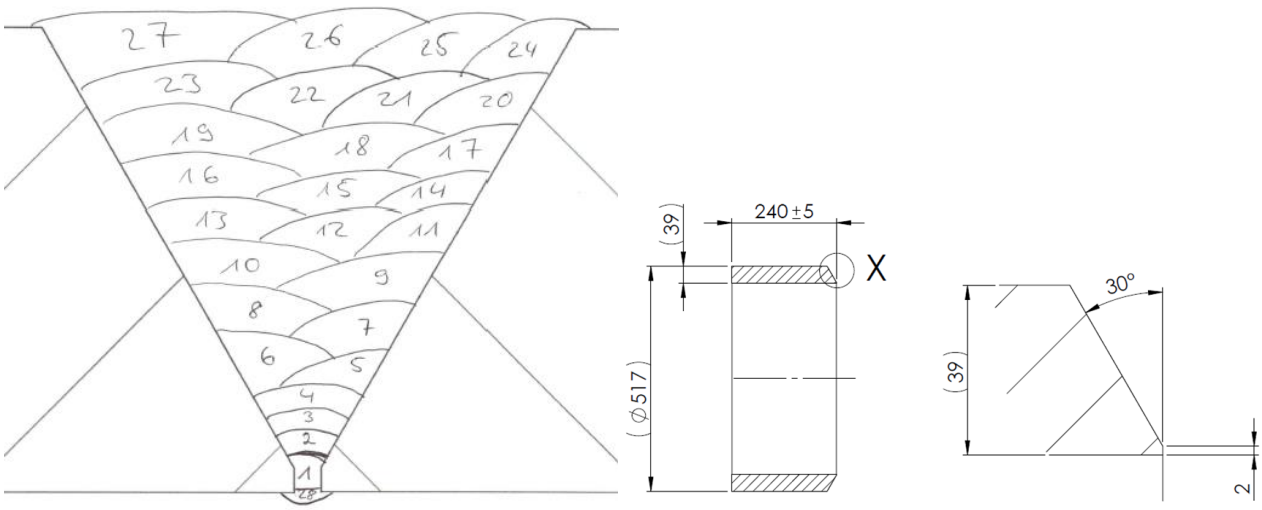


Figure 6: Welding layout

For most of the models the 2D axial symmetric model was used, and for every model several cases were simulated. In case of the first model the examined cases are listed in Table 4.

Case	Mesh	Heat Source	Weld path
1	Normal	Estimated	Normal
2	Finer	Estimated	Normal
3	Extremely fine	Estimated	Normal
4	Normal	Estimated	Modified
5	Normal	Estimated	Modified
6	Normal	Calculated	Normal

Table 4: Simulation versions of Model-1

The finite element mesh of the first model (Model-1) can be seen in Figure 7. For Model-1 six simulations were made, where first the mesh independency was examined, following with a modified weld path, in case 4 all of the weld path was changed for case 5 only the weld paths of #5-27 were modified.

In the first five cases for the heat source the estimated data were used, and finally a simulation was made for the calculated heat source parameters too. In this model the following assumption and settings were made:

- 2D axial symmetric model
- Elements: 2716 (in case of the normal mesh)
- Nodes: 2800 (in case of the normal mesh)
- Average mesh size: 1mm (in case of the normal mesh)
- Back weld (#28) neglected
- No electrode tilts

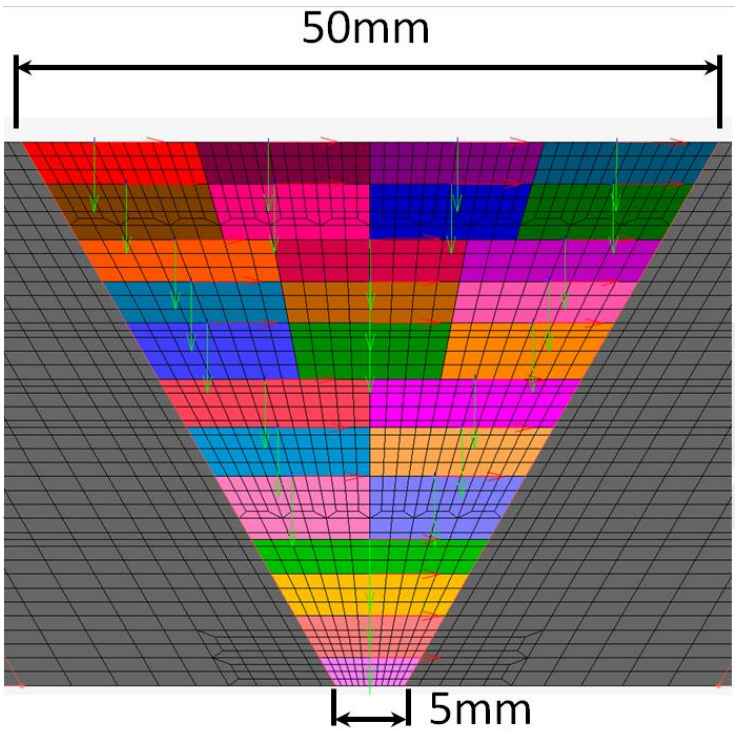


Figure 7: Finite element mesh of Model-1

In case of the second model (Model-2) the same mesh was used as for Model-1. Therefore, in this case too the 2D axial symmetric model was applied. The difference between the two models lies in the effect of the electrode tilting. While in Model-1 it is not considered in Model-2 the electrode is tilted between #5-27 passes. The angle of the tilting is between 10° and 15°. It can be seen in Figure 8. The finished simulations with this model are listed in Table 5. In this case only the size of the mesh was changed.

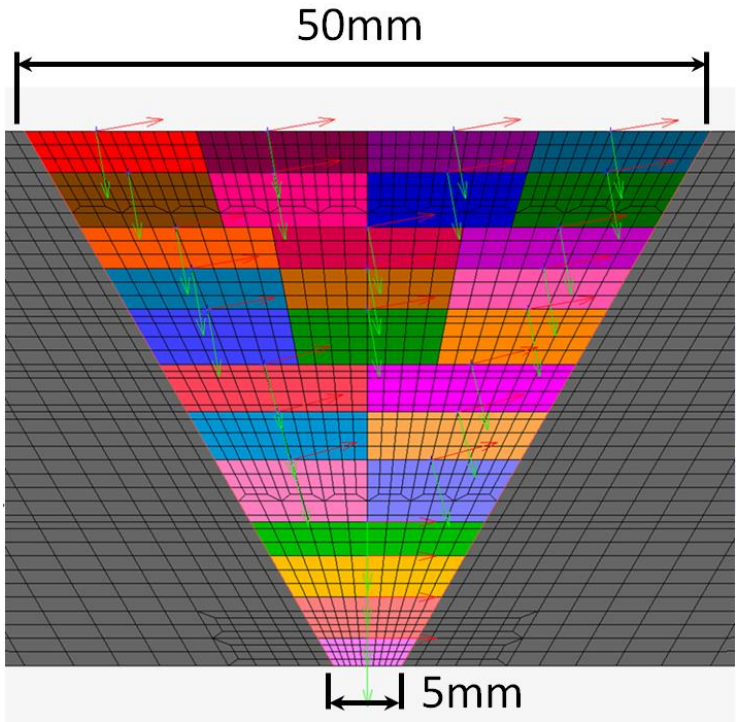


Figure 8: Finite element mesh of Model-2

Case	Mesh	Heat Source	Weld path
7	Normal	Estimated	Normal
8	Finer	Estimated	Normal
9	Extremely fine	Estimated	Normal

Table 5: Simulation versions of Model-2

As the mock-up included a back weld the simulations were also made for this case, and the model was modified. This model can be seen in Figure 9. In this model the following options were set:

- 2D axial symmetric model
- Elements: 1914
- Nodes: 1982
- 2 mm root gap
- Real geometry
- Average mesh size: 1.5mm
- Back weld (#28) included
- Electrode tilted (10-15°) between #5-27 passes

For Model-3 normal weld path and the estimated heat source data were used.

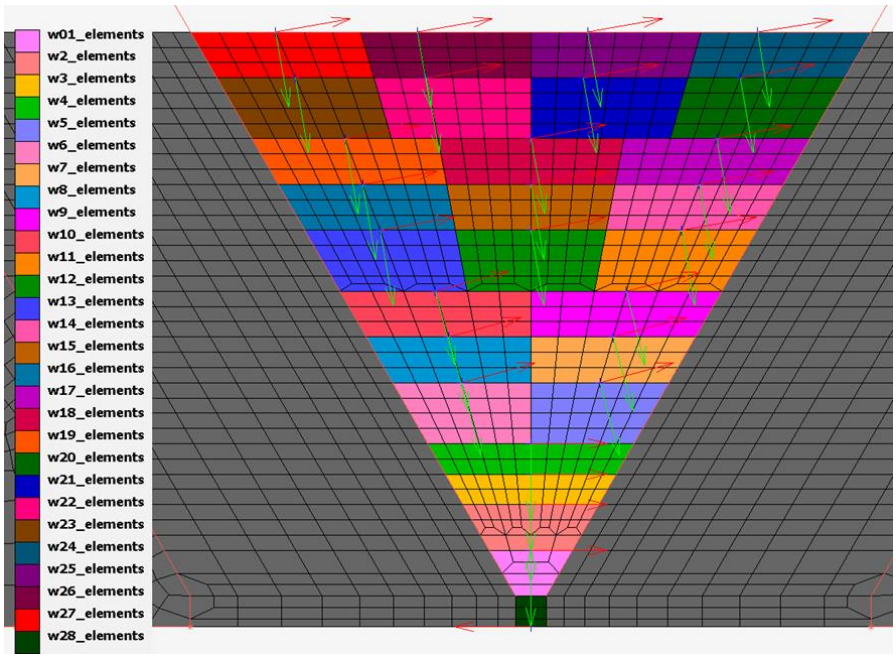


Figure 9: Finite element mesh of Model-3

For Model-4 and Model-5 for the heat source shape the geometry generated in MINA was used. MINA is simplified software for grain orientation modelling developed by EDF and LCND [21]. This is shown in Figure 10. As with the Model-1 and Model-2, the back weld was neglected here too.

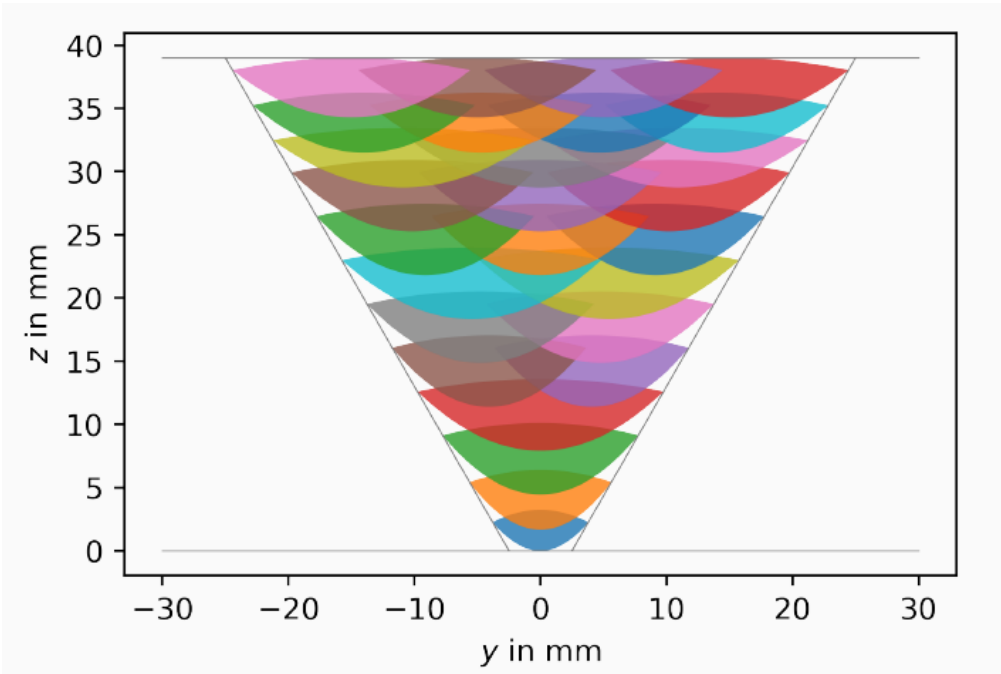


Figure 10: MINA geometry

The generated heat source shape was imported into the MSC.Marc Mentat software; it is illustrated in Figure 11. In this version the following assumptions were made:

- 2D axial symmetric model
- Elements: 2976
- Nodes: 3055
- Average mesh size: 1mm
- Back weld (#28) neglected
- No electrode tilts

For Model-4 only the heat source shape was used from MINA and the position of the weld paths was modified, however in case of Model-5 beside the shape the weld path was also included from MINA.

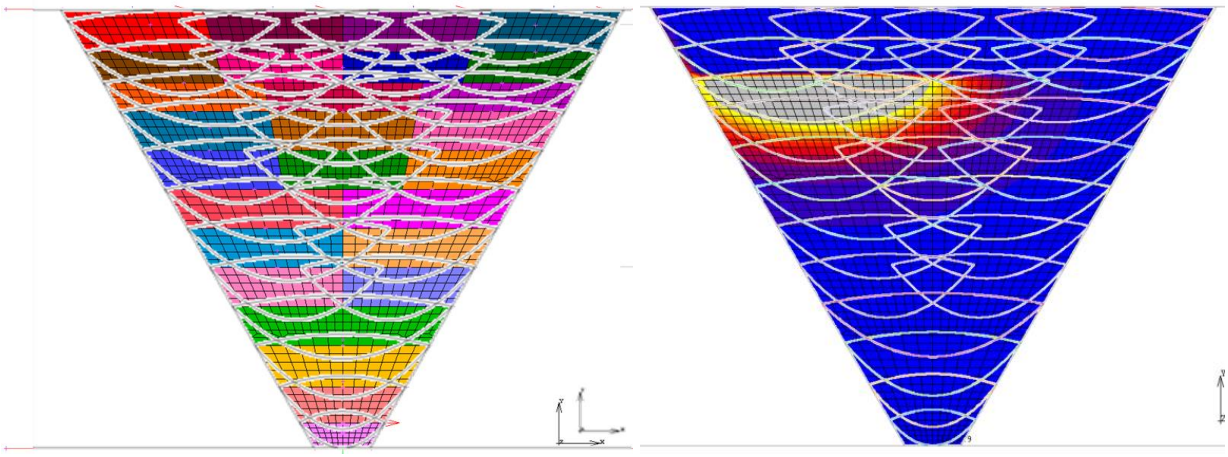


Figure 11: Heat source shape for Model-4 and Model-5

Two more models were generated, for these the heat source model was validated in 3D with the help of Simufact.Welding software. The generated molten volume of the 3D welding is shown in Figure 12.

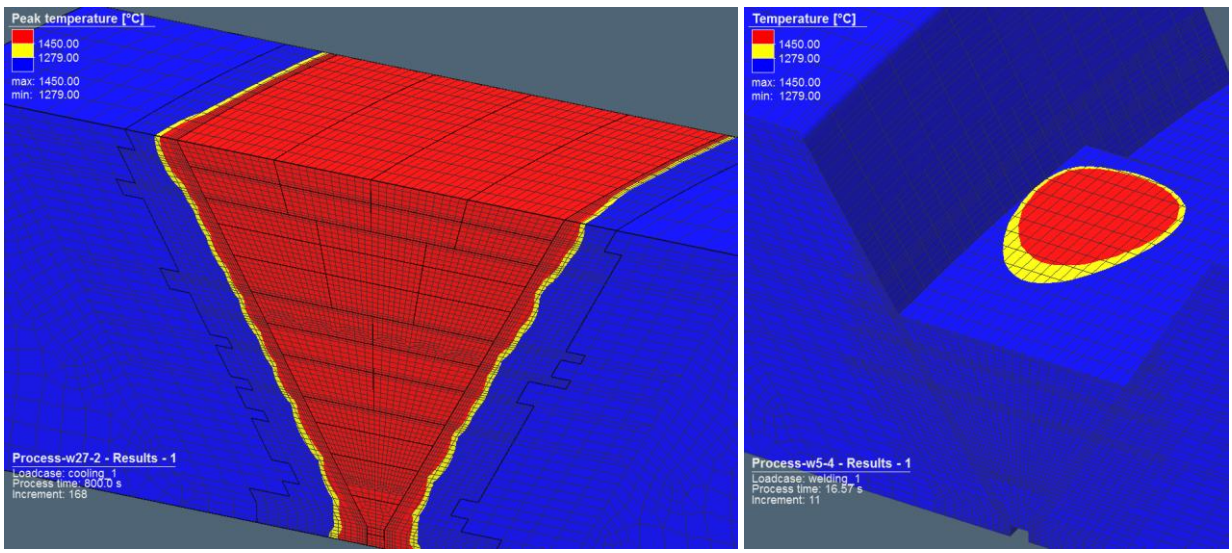


Figure 12: Validation of heat source in 3D

The difference between these models and the previous ones is that here a 2 mm root gap was left, it is well visible in Figure 13, which shows the meshing of Model-6 and that there is no tilting of the electrode in this case. The applied assumptions are the following:

- 2D axial symmetric model
- Elements: 2260
- Nodes: 2330
- Average mesh size: 1mm
- Real geometry
- 2 mm root gap
- No electrode tilt
- Usage of the calculated weld speed and heat input

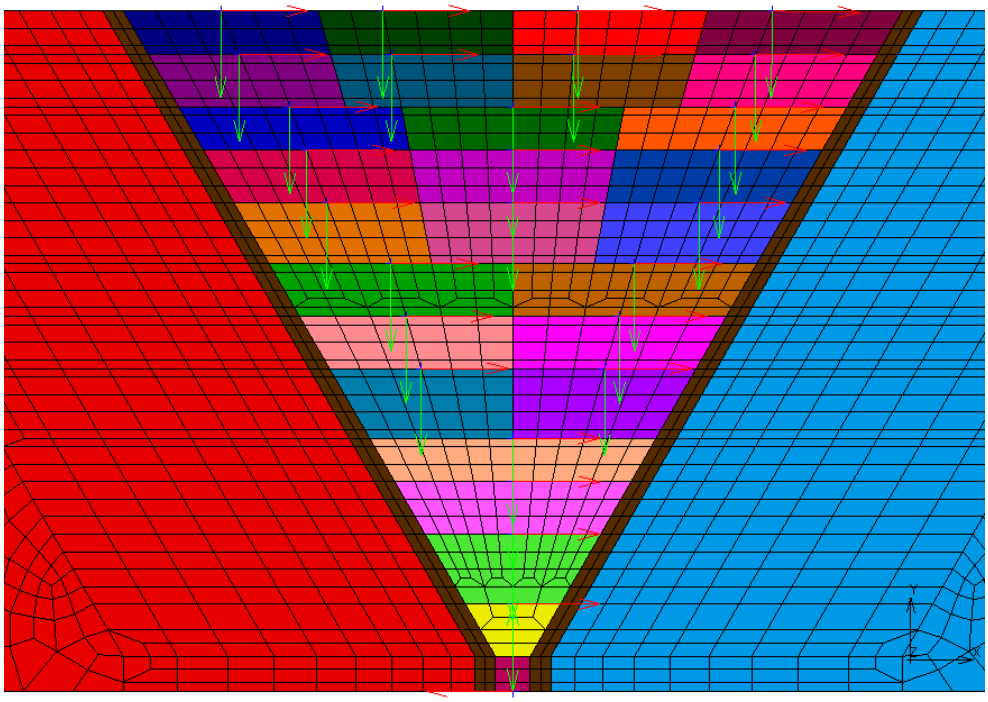


Figure 13: Finite element mesh of Model-6

The difference between Model-6 and Model-7 is only the electrode tilting, in case of Model-7 between the #9 and #27 passes the electrode was tilted with 15° and for this model a normal and an extremely fine mesh was also used. The simulation of the modified electrode tilting was necessary because the changing of the tilt can be seen in the prepared mock-up too (Figure 14) therefore it is essential to know the influence of this parameter.

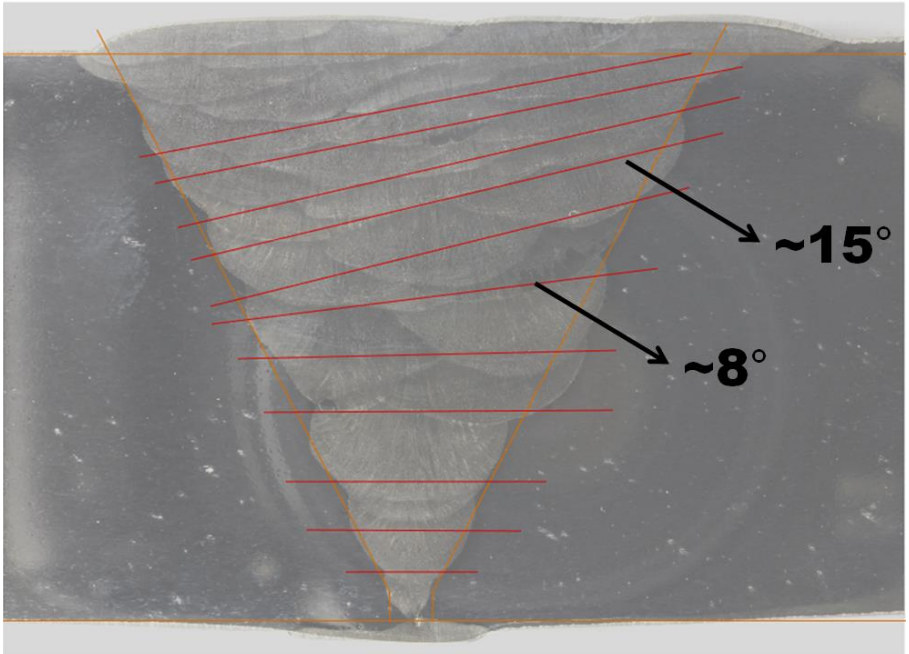


Figure 14: Position of the weld paths in the prepared weld sample

The cases for Model-3 to Model-7 are summarized in Table 6.

Model	Case	Mesh	Heat Source	Weld path
Model-3	10	Normal	Estimated	Normal
Model-4	11	Normal	Estimated	Modified
Model-5	12	Normal	Estimated	Normal
Model-6	13	Normal	Calculated	Normal
Model-7	14	Normal	Calculated	Modified
	15	Extremely fine	Calculated	Modified

Table 6: Further simulation cases

3.4 Results of the finite element simulations

In the previous section the different types of models were introduced and in the following one, the received results will be shown and compared to estimate the effects of the modified parameters. The grain orientation field is presented on FE mesh. The comparison of the different models (with different mesh distribution) is visualised by Scilab software. Using a grid with equidistant point distribution the result of sample points was interpolated from the FE node results. Figure 15 and Figure 16 shows the changes of the grain angles of Case 1, in the following the behaviour of this grain angle is going to be compared in more detail. Most of these comparisons will be done between Model-1 and the later made versions.

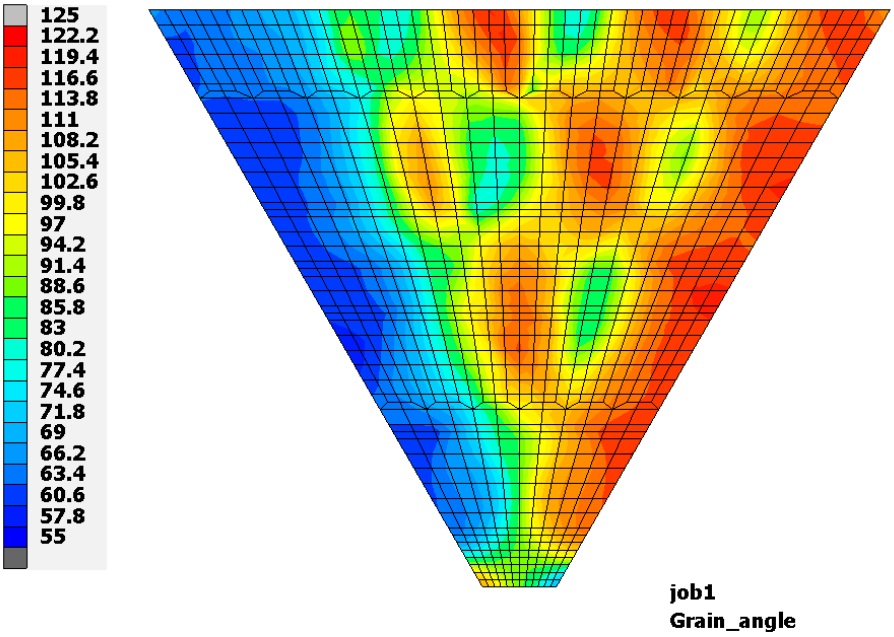


Figure 15: Discretization on FE mesh

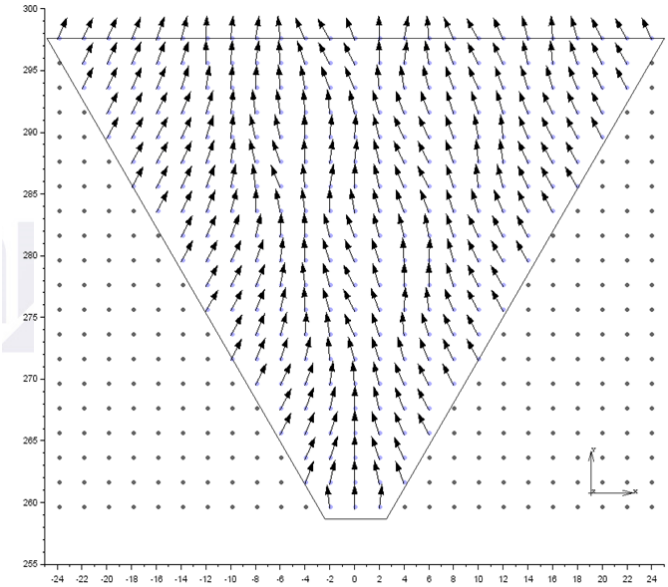


Figure 16: Discretization on sample points: 2 mm

3.4.1 Effect of the mesh size

First of all, the effect of the mesh density was examined in case of Model-1. The mesh size and the simulation cases are listed in Table 7. It is important in every finite element simulation to examine the mesh independency of the model. The results of the models should be independent from the size of the mesh; therefore the simulations were made with three mesh sizes. It should be known that because of the denser mesh the size of the weld pool will be closer to the real shape, however this difference is very small, with the finer meshes the weld pool shape is drawn out more accurately, however because of the larger elements number the calculation time would also increase.

Case	Mesh	Size
1	Normal	1 mm
2	Finer	0.5 mm
3	Extremely fine	0.25 mm

Table 7: Different mesh sizes for Model-1

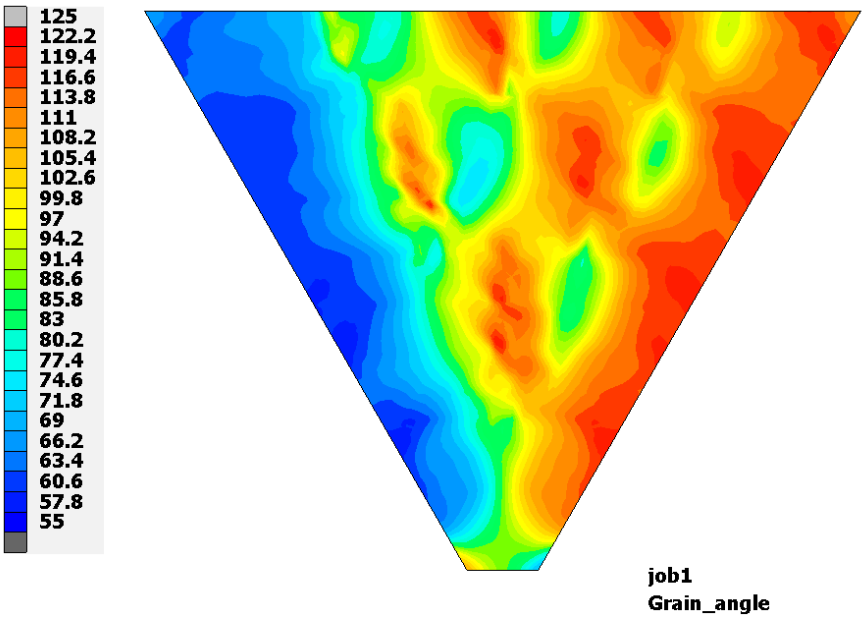


Figure 17: Grain angle distribution of Case 2

In Figure 17 and Figure 18 the grain angles are introduced for cases 2 and 3. Compared to Figure 15, the basic layout is similar however some differences can be noticed.

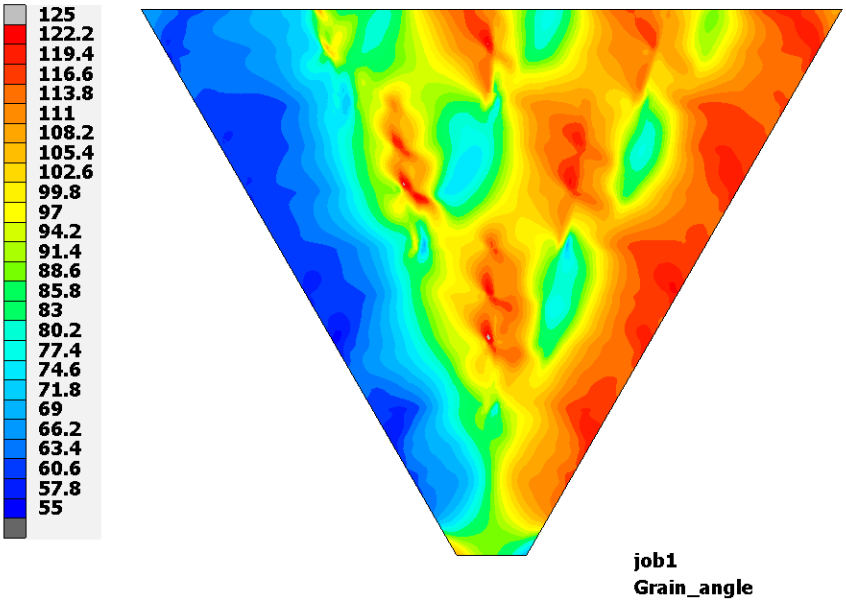


Figure 18: Grain angle distribution of Case 3

For the better understanding in these cases a relative and an absolute error map was made for cases 2 and 3 too which can be seen in Figure 19 and Figure 20.

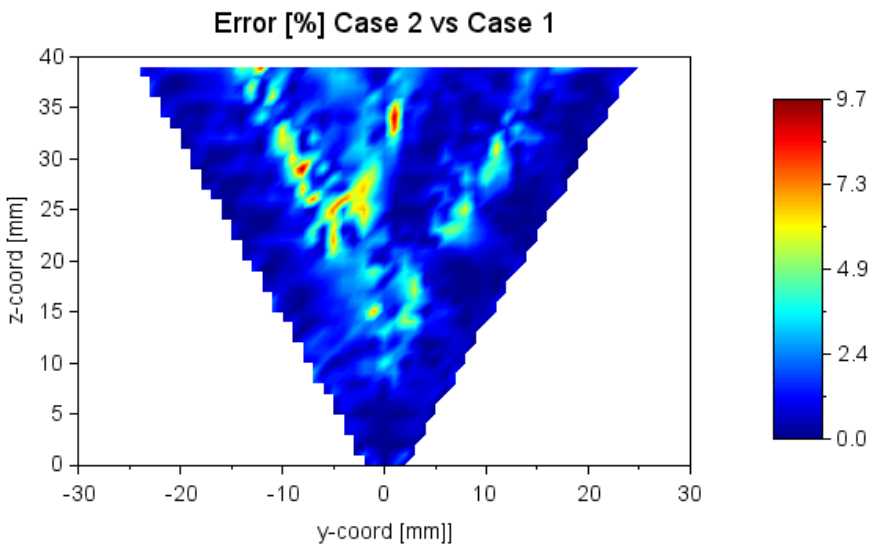


Figure 19: Error between Case 2 and Case 1 simulations

In the case of the finer mesh, the maximal difference from the normal mesh is 9.7%, and it only appears in a few spots, in case of the extremely fine mesh however this value reaches 15%. These values are still acceptable.

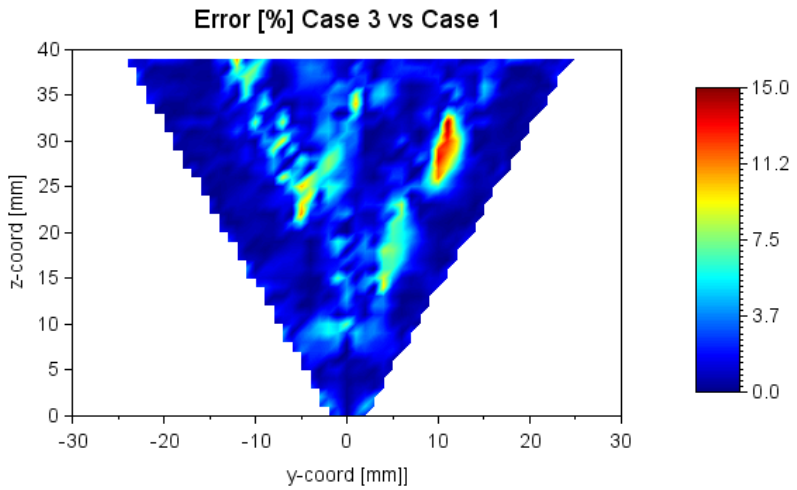


Figure 20: Error between Case 3 and Case 1 simulations

3.4.2 Effect of the geometry

It was very important to have a better idea about what type of simplifications can be applied for a welding simulation, which would not cause drastic changes in the grain distribution. Therefore, after the creation of an estimated, simpler geometry the simulations were made on the real geometry as well. The real geometry was used in case of Model-3, 6 and 7. The distribution of the grain angle is shown in Figure 21 and Figure 22.

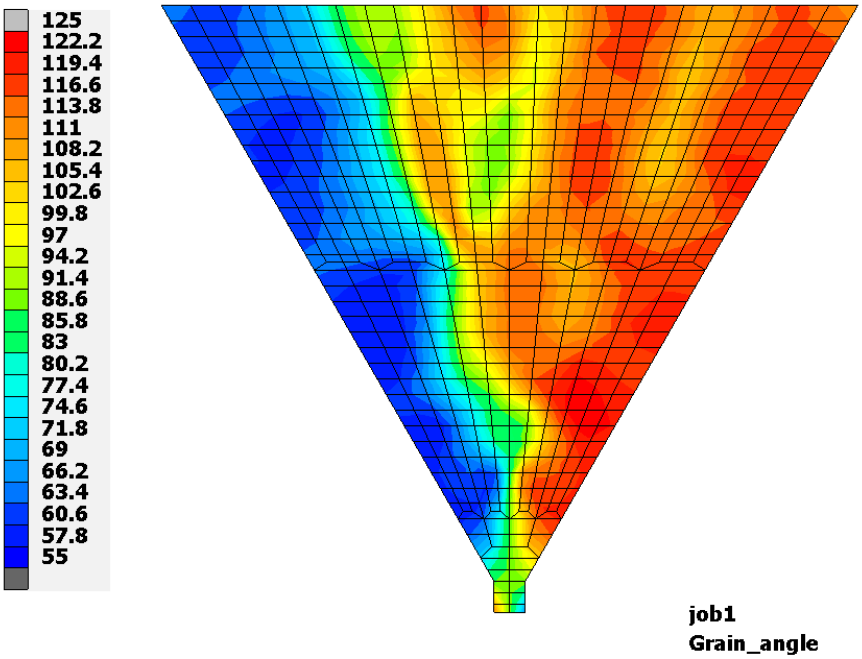


Figure 21: Distribution of grain angle of Model 3

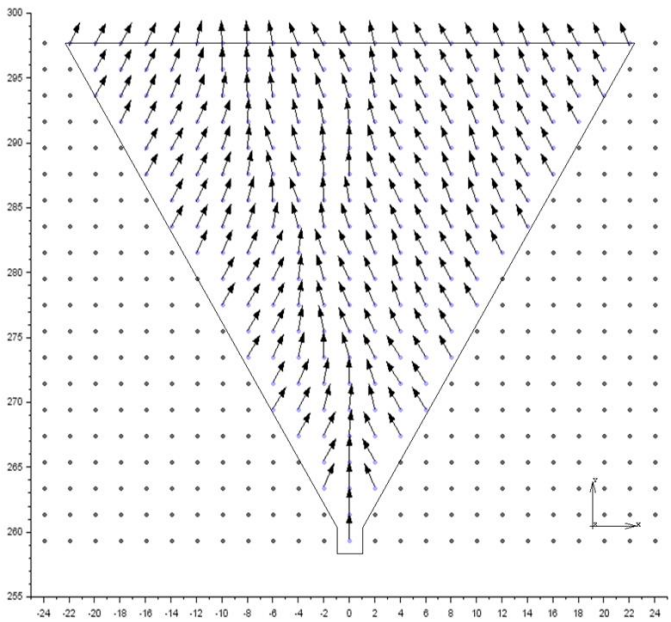


Figure 22: Vector plot of grain orientation on sample points with 2mm

In case of Model-3 the results were compared with Model-1 and Model-2, the differences between these simulations are listed in Table 8.

Case	Geometry	Electrode tilt
1	Estimated	No
7	Estimated	Between #5-27 passes
10	Real	Between #5-27 passes

Table 8: Comparison of Model-3 with different cases

Case 1 and 10 have two differences: one is the geometry and another one is the tilting of the electrode; these two parameters affect the results greatly as can be seen in Figure 23. The error between the two simulations is 35%, however from this result the effect of the usage of the real geometry cannot be estimated.

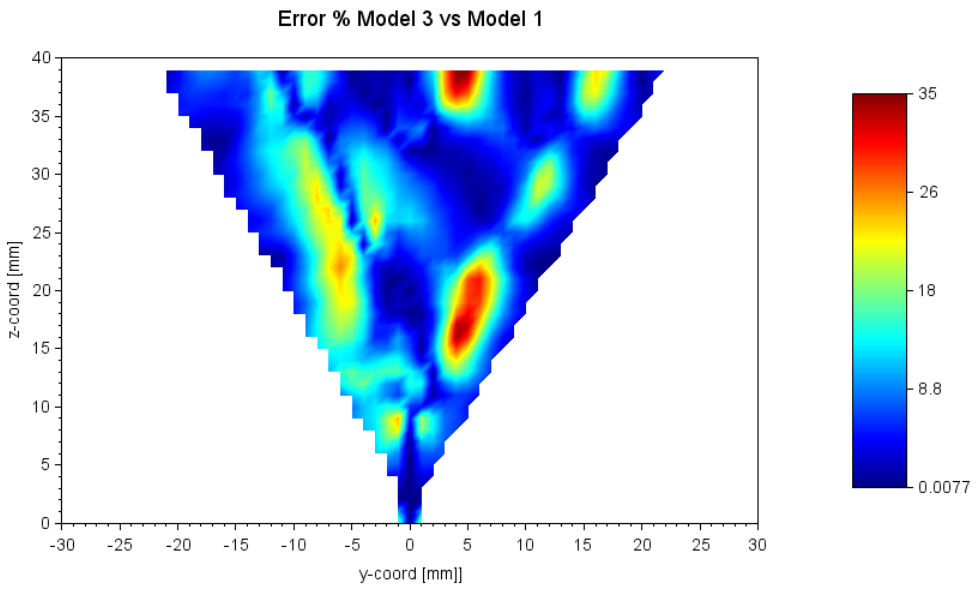


Figure 23: Error between Model 3 and Model 1

To have a better understanding about the influence of geometry, the results of Model-3 were compared with Model-2 too. The error between these cases is illustrated in Figure 24. These two simulations only differ in their geometry; therefore, the effect of the geometry can be better estimated. There is a 26% error between these cases which cannot be neglected. It can be assumed from these results, that the usage of the right geometry is essential in the estimation of the grain angle distribution; the model should not be simplified further as the received results could be affected greatly.

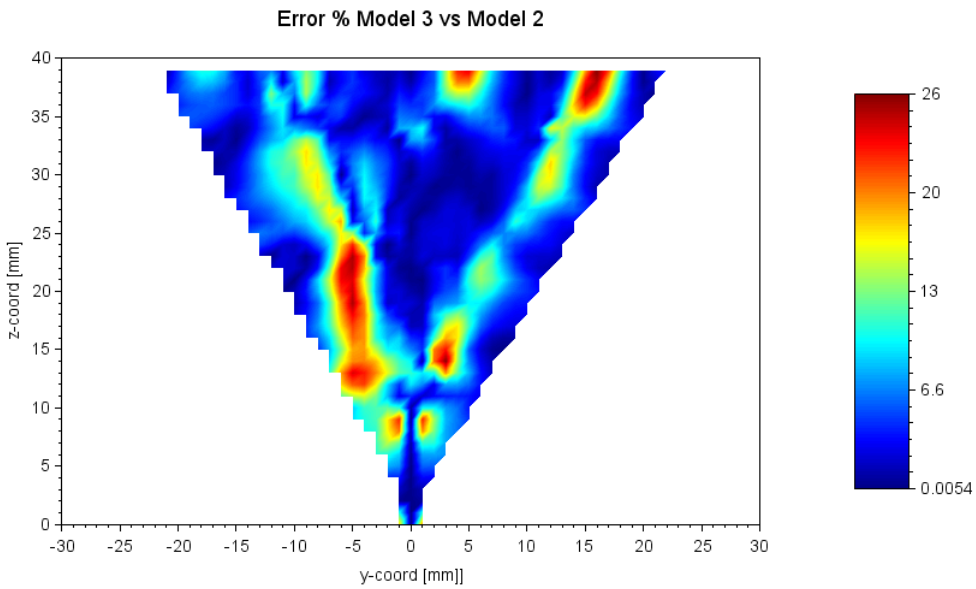


Figure 24: Error between Model 3 and Model 2

Based on the received data the real weld built-up was created in Model-7 where beside using the real geometry the illustration of the real welding was also examined, from it could be estimated that the electrode tilting starts at #9 pass and it becomes larger while it reaches the last, #27 pass. The distribution of the grain angle is shown in Figure 25.

Case	Geometry	Heat source	Electrode tilt
7	Estimated	Estimated	Between #5-27 passes
15	Real	Calculated	Between #9-27 passes

Table 9: Comparison of case 7 and 15

The results were also compared with Model-1 to see the differences between the results of the estimated and the real geometry. The error is shown in Figure 26. There are locations within the welding where the results are highly different. The error reaches 50%.

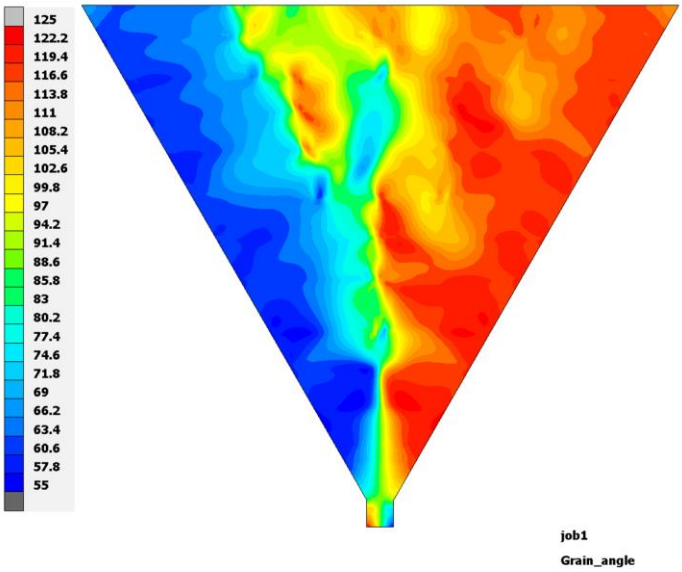


Figure 25: Grain angle of Model-7

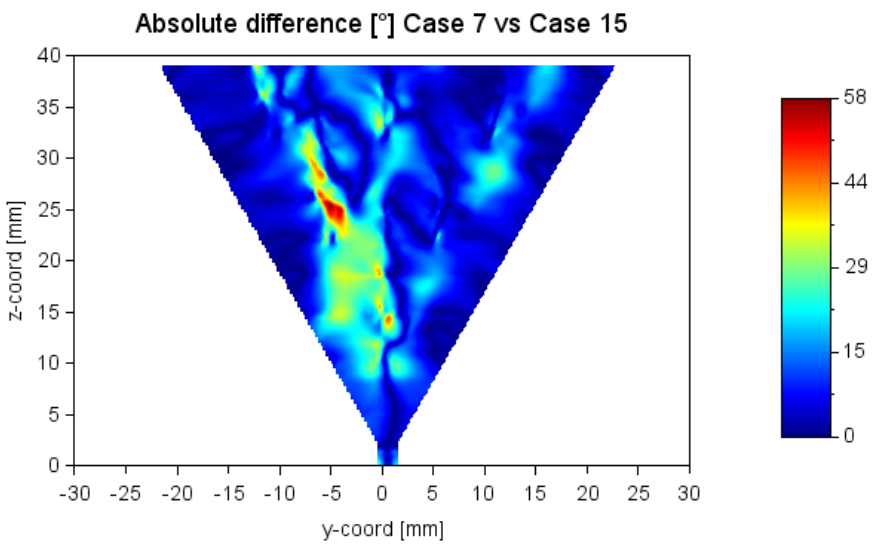


Figure 26: Absolute difference between Case 7 and Case 15

3.4.3 Effect of the weld path

First, the weld path effect in Model-1 is going to be compared. In case 1 the weld path was normal and the position of it was centre, however for cases 4 and 5 the weld paths were modified, the modifications are summarized in Table 10.

Case	Weld path	Position
1	Normal	Centre
4	Modified	All path position change (horizontal, left 1mm)
5	Modified	Path position change (from #5 to #27 expanded horizontally 1 mm)

Table 10: Types of weld path for Model-1

The modification of the weld paths resulted in large errors between Case 4 and Case 1, the difference between the two simulations is shown in Figure 27. at the side of the weld there is no or very small difference between the two cases, however in the middle part of the weld at some places the values can have more than 13.7% error, this would result in wrongly simulated grain angles, therefore it is very essential that the input data are as accurate as possible because this parameter has a large effect on the distribution of the grains.

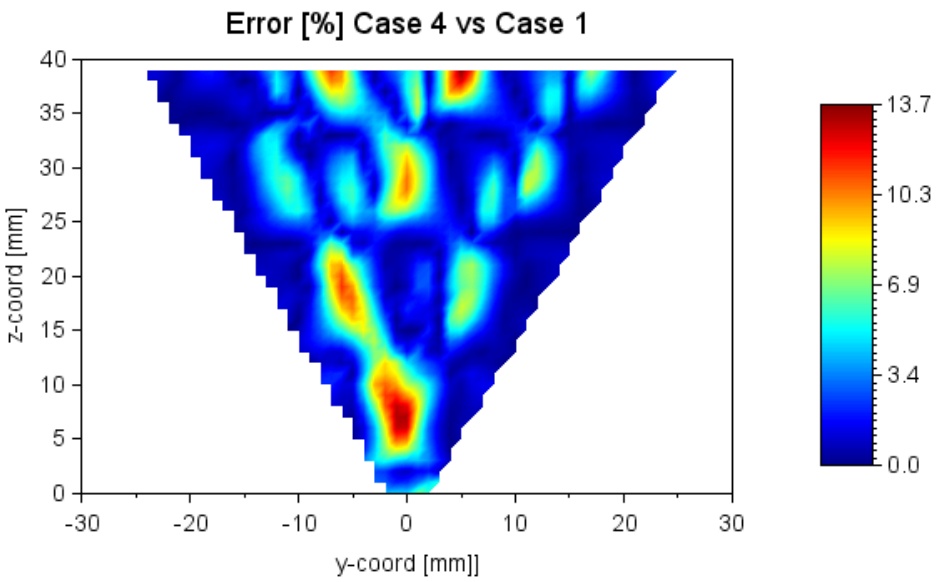


Figure 27: Error between Case 4 and Case 1 simulation

This error can be noticed in Case 5 too, here the path position was only changed from the #5 path, however in this case too, the difference has exceeded 13%, as shown in Figure 28.

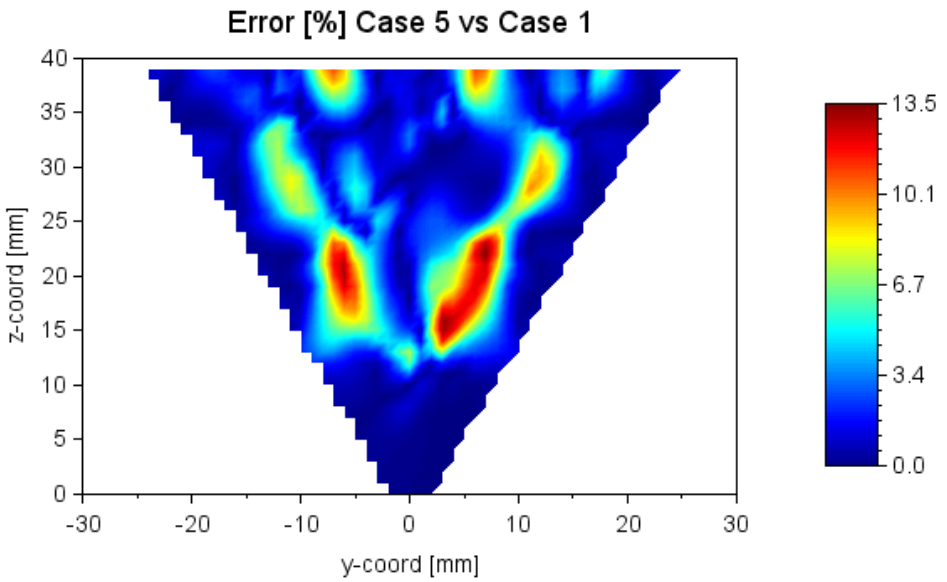


Figure 28: Error between Case 5 and Case 1 simulations

3.4.4 Effect of the heat source

Finally, in case of Model-1 the heat source effect was also considered. The shows the grain angle distribution of Case 1 and Case 6. In Case 1 the estimated values were used for the heat source and in Case 6 the calculated ones. The difference can be seen very well even comparing these two pictures.

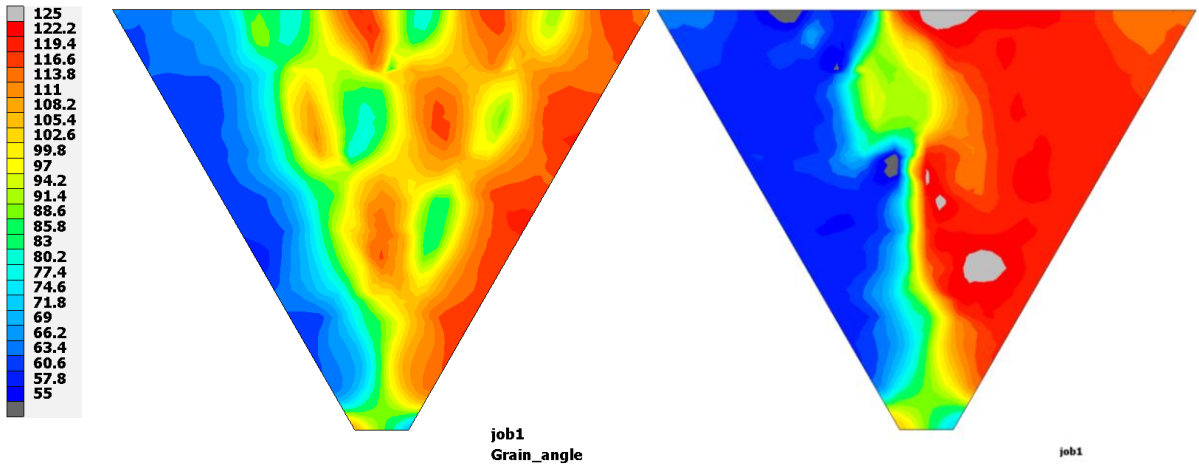


Figure 29: Grain angle distribution (Case 1 – left; Case 6 – right)

To have a better picture of it the error between these cases was also illustrated (see Figure 30).

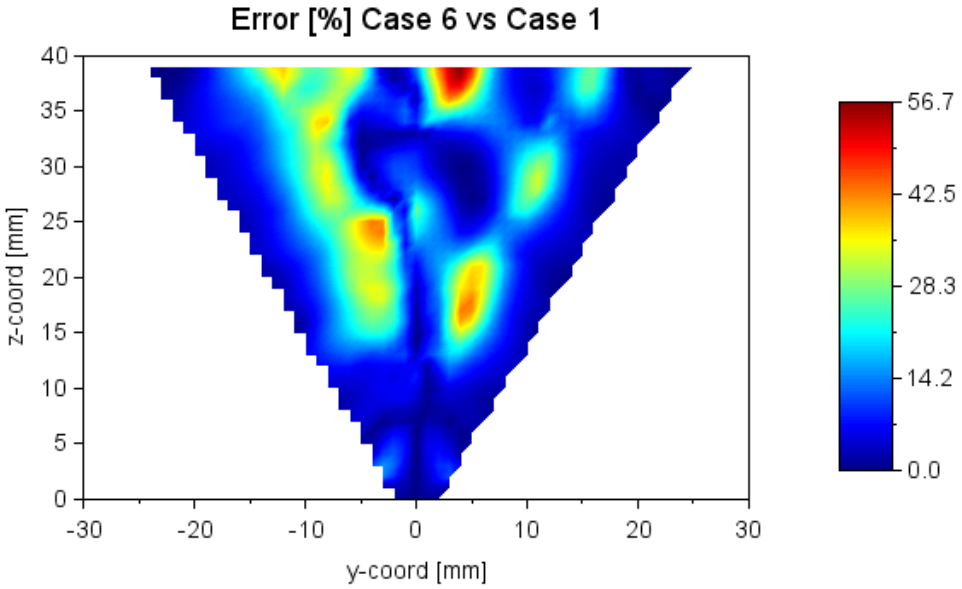


Figure 30: Error between Case 6 and Case 1 results

The largest difference is 56.7%, there are almost no parts on the welding where the grain angles are equal for both cases. For most parts of the weld there is a 14% difference between the angles. From only this one result it can be seen that the heat source model has to be very accurate because even if in the nugget areas there is only a few mm^2 difference, this could result in totally different grain distributions.

3.4.5 Effect of the electrode tilting

Between Model-1 and Model-2 the difference was in the electrode tilting, in the first case it was not included for Model-2 the tilting of the electrode was taken into consideration, therefore between #5-27 passes the electrode was tilted with 10° to 15°. The received results are shown in Figure 31 and Figure 32.

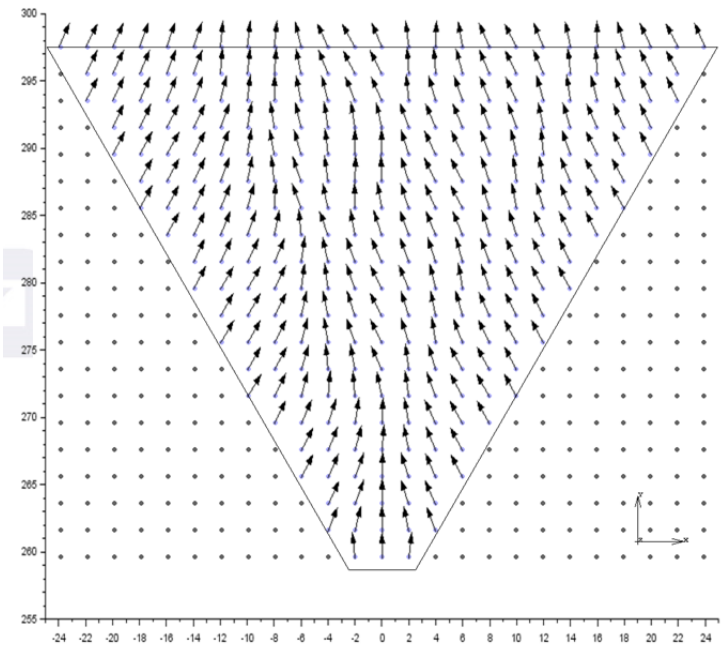


Figure 31: Distribution of grain angle with arrows of Case 7

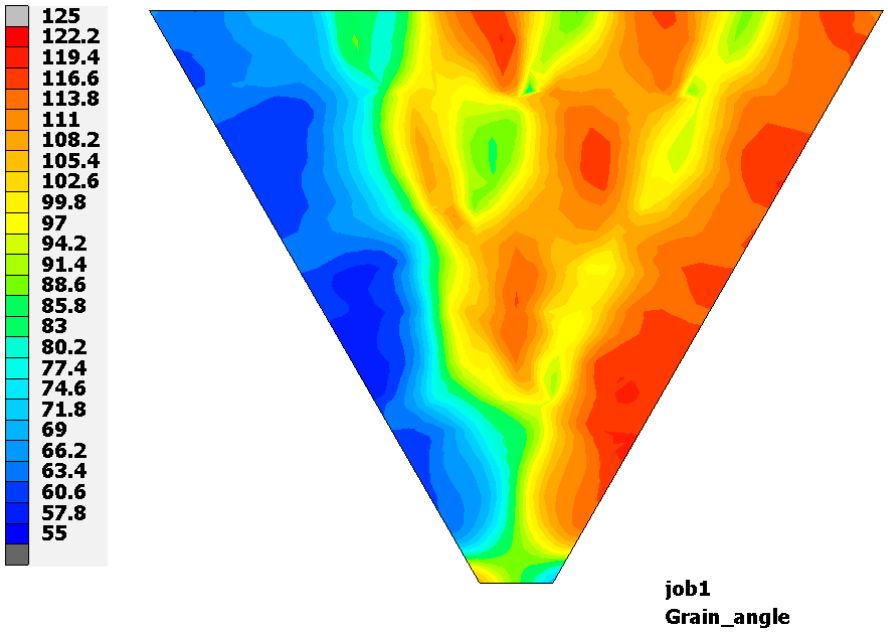


Figure 32: Distribution of grain angle of Case 7

The result was compared with Case 1 where the tilting of the electrode was not used. The difference between the two simulations is shown in Figure 33, where the largest error is 19.3%.

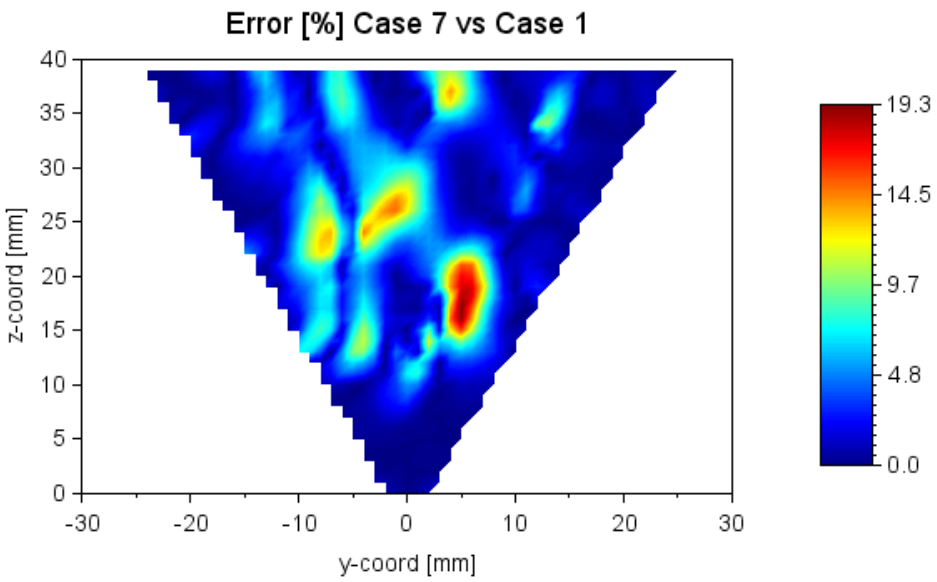


Figure 33: Error [%] Case 7 and Case 1

For Model 2 the simulation was done with extremely fine mesh too. The grain angle of this case is shown in Figure 34.

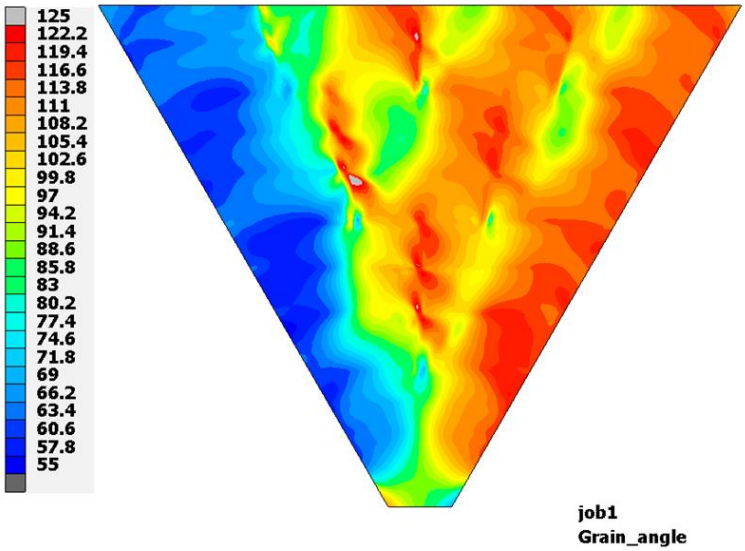


Figure 34: Distribution of grain angle of Case 9

This model was compared to Case 3 where the extremely fine mesh was used for Model-1. The error between these two models is much easier noticeable than in the previous case. Here the error between the two simulations reaches values as high as 30.6%, as can be seen in Figure 35. This and the previous results show how important it is to have knowledge about the electrode position, it has a great influence on the resulted welding therefore it cannot be neglected.

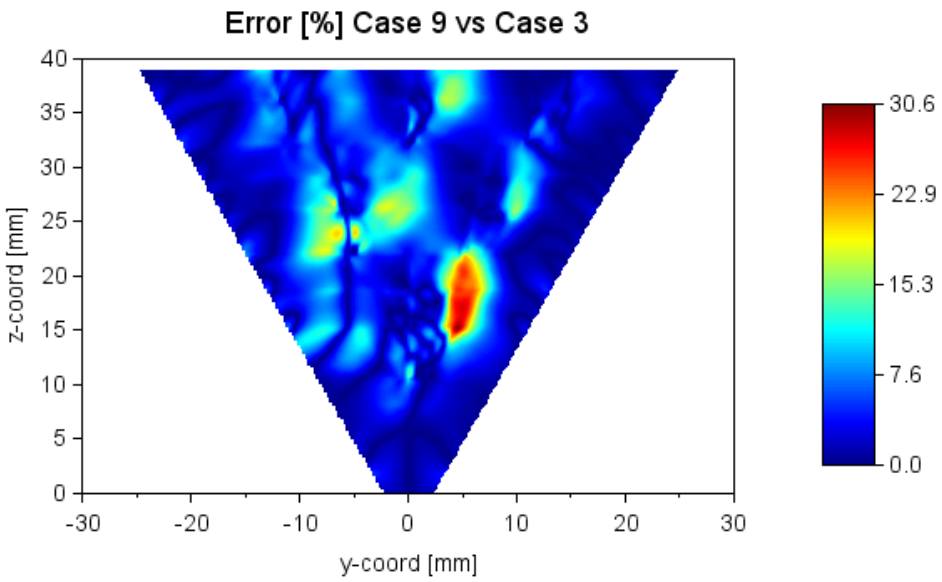


Figure 35: Error [%] Case 9 and Case 1

Based on the received data the real weld built-up was created in Model-7 where beside using the real geometry the illustration of the real welding was also examined from these it could be estimated that the electrode tilting starts at #9 pass and it becomes larger while it reaches the last, #27 pass. The distribution of the grain angle is shown in Figure 21 and Figure 25.

Case	Geometry	Heat source	Electrode tilt
10	Real	Estimated	Between #5-27 passes
15	Real	Calculated	Between #9-27 passes

Table 11: Comperation of Model 7 with different cases

Case 15 and Case 10 differ in the starting point of the electrode tilting. In Case 10 the tilting is started from the #5 pass whereas in Case 15 it was postponed and only starts from the #9. In addition, the two simulations included equal heat sources. The results were compared, and it showed large errors between the two simulations, which are as high as 41.5%, which is visible in Figure 36. This difference appears in the lower middle part of the welding which is predictable as the difference between the two models is at this location. However, this example also shows how highly it affects the final results if the electrode tilting is not or wrongly considered in the simulation.

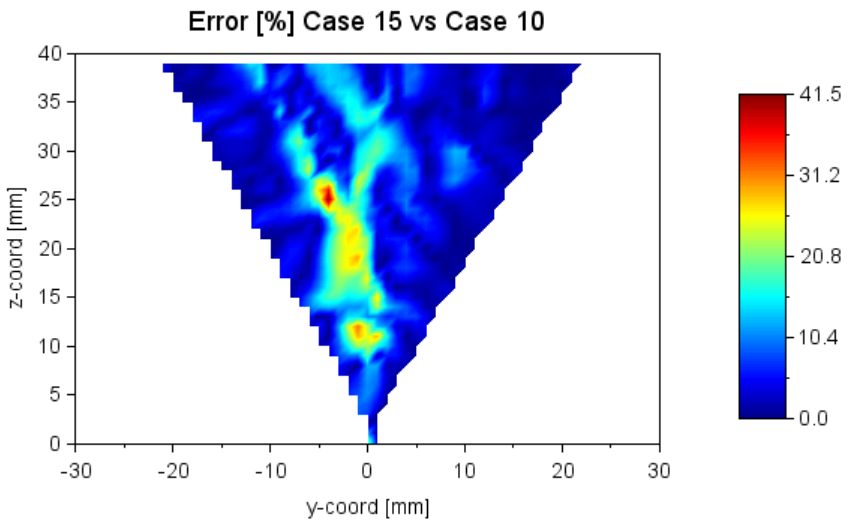


Figure 36: Comperation of Case 15 and Case 10

It has been demonstrated that the simulation of the grain orientation is a faceted problem, with many influencing parameters. The results show that the sensitivity of the unknown welding parameters and the geometry representation are crucial. Therefore, later 3D simulations considered only those welding parameters which modify the weld pool shape during welding, focusing on the applied power and electrode tilting effect.

3.4.6 Comparison with MINA simulation

In order to compare the MINA grain orientation prediction with the presented FE simulations two type FE model was built with MINA heat source shape. Although the MINA code has been dedicated to shielded metal arc welding and flat welding, the comparison with the developed FE algorithm is relevant [21].

The predicted grain orientation distribution using MINA is presented in Figure 37. At the same scale and equal sample point result of FE simulation also shown in Figure 38.

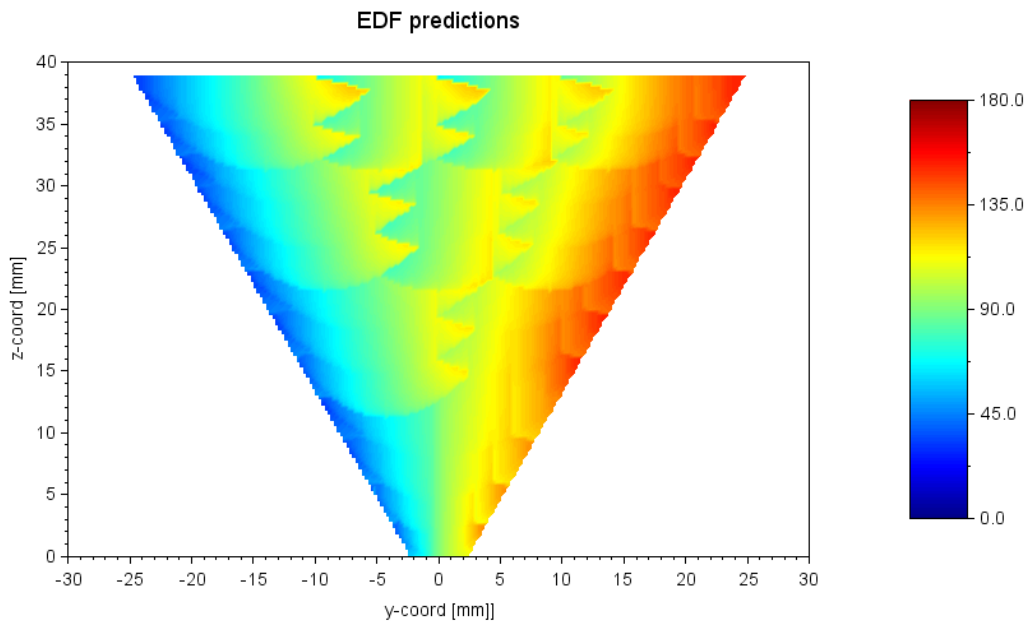


Figure 37: EDF prediction using MINA

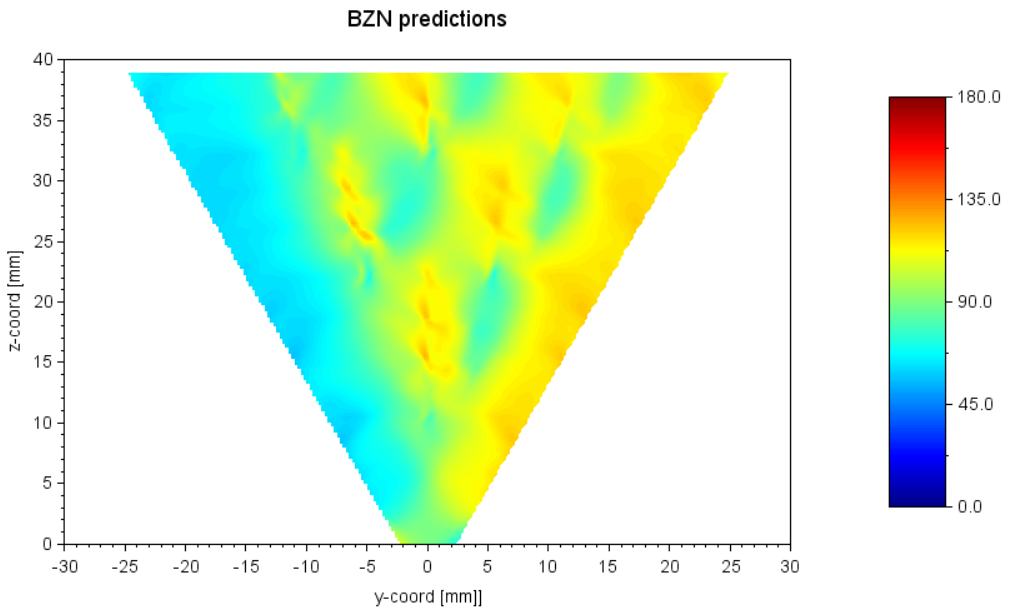


Figure 38: BZN prediction of Model-1, Case 3

The comparison shows that relatively large errors exist between the two simulations, which reach values as high as 56%, as can be seen in Figure 39. The deviations are localised in the bevel side area (left and right zones) and at the location of the higher welding pass number, which is most probably a consequence of the different physics representations, because the FE simulation considers the mechanical effect during the welding. The mechanically deformed welding passes and the deformation of the bevel angle were probably re-melted at the outside edge of the regions, resulting in these deviations.

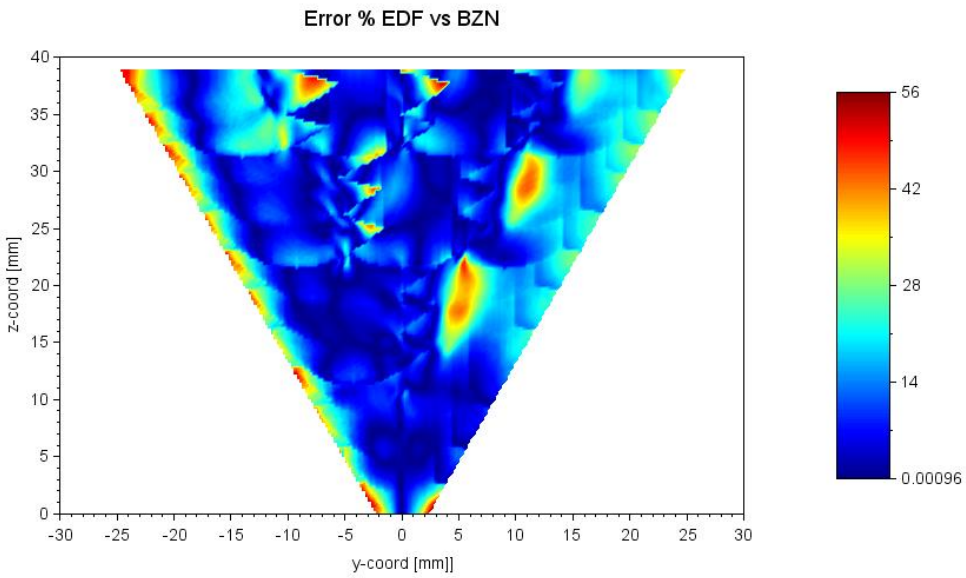


Figure 39: Comperation of the EDF and BZN results (Model-1 Case 3)

3.5 Thermal simulation of welding process

Based on the result presented, BZN revised the original theory which applied the simulation of the grain orientation. Because during the welding, beads are produced by the applied heat source in which grains generally grow along the maximum thermal gradient when cooling [16]. This approach is adequate to determine a realistic grain orientation without considering the effect of geometry distortion. Furthermore, using this approach more welding processes and complex geometry also can be examined. The approach is implemented in the MSC.Marc FE code used and further extended with the prediction of grain size. Neglecting the physics of mechanics, the computation time is significantly reduced, therefore large 3D computations can be realized.

User-supplied subroutines were used to implement the grain orientation, size, and growing calculation models of the overall welded area. The calculation of the grain orientation is based on the MINA model [21]:

$$\begin{gathered} n = 0, \alpha_0 = \alpha_{lower\ box} \\ t_n = \begin{cases} cos(\alpha_{n-1} - \alpha_g);\ \ if\ (\alpha_{n-1} - \alpha_g) < \pi/2 \\ 0\ ;\ else \end{cases} \\ \alpha_n = t_n + (1 - t_n)\alpha_g \end{gathered} \tag{11}$$

where α_n represents the angle of grain, $\alpha_g$is the actual temperature gradient direction, t_n is a parameter which represents the changing of the orientation, n is the number of applied iterations. This parameter n represents the transition between the microscopic and macroscopic scales.

Unfortunately, the value of n depends on the cooling rate during the FE calculation, therefore value of n can be determined as a function of temperature change (a temperature decrease of two degrees Celsius increases the value of n by one). However, the changes in the temperature gradient were only considered between the start and finish of the solidification.

According to a model proposed by Andersen and Grong, grain growth can be described by the simple classic isothermal grain growth equation tailored for non-isothermal heating and cooling conditions [22]:

$$\frac{d\overline{D}}{dt} = Ae^{\left(-\frac{Q}{RT}\right)}\left(\frac{1}{\overline{D}} - \frac{1}{k}q_p\right)^{\left(\frac{1}{n}-1\right)} \tag{12}$$

where, A denotes a kinetic constant that describes the grain boundary mobility, Q is the apparent activation energy for grain and R is the gas constant. According to Bernhard and Jürgen adopting the time exponent $n = 0.5$ and disregarding the influence of precipitates leads to the following simplified form [23]:

$$\frac{d\overline{D}}{dt} = Ae^{\left(-\frac{Q}{RT}\right)}\left(\frac{1}{\overline{D}}\right) \tag{13}$$

where $A = 0.004\frac{m^2}{s}$, $D_0 = 100\mu m$ and Q is depend on the equivalent carbon content in case of austenitic steel:

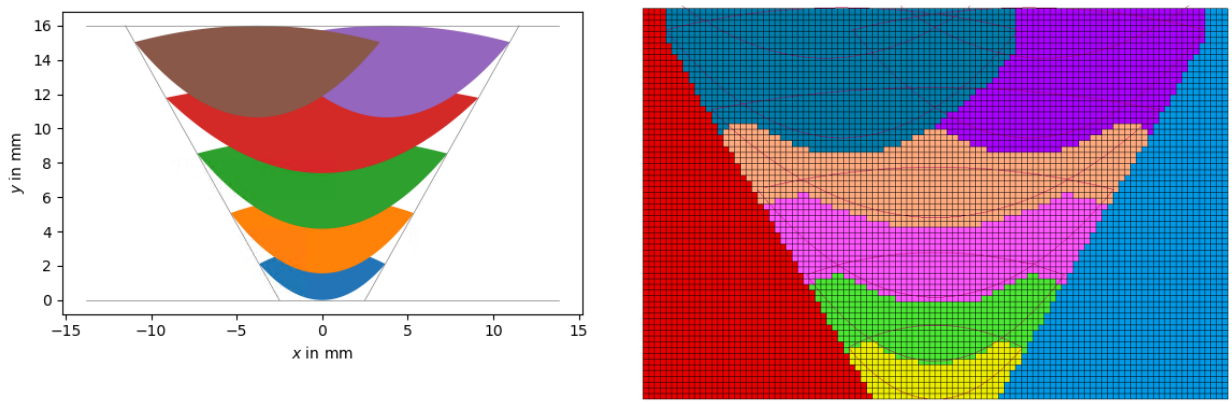
$$Q = 167686 + 40562(mass\ pct\ c_p) \tag{14}$$

Unfortunately, the validity of the model is limited to initial cooling rates of around $10\frac{^\circ C}{s}$, therefore an extension of the higher range is required for welding. Since the local temperature during the solidification of the weldment is not a simple time-dependant function, the calculated temperature evolution must be solved numerically during the process [23][24].

3.5.1 Comparison between the new approximation and MINA code

In order to check the new approximation which is included in the MSC.Marc code a new simplified FE model was built based on MINA bead shapes. The welding model includes only 6 passes, as can be seen in Figure 40, which is sufficient to verify the validity of the Fortran code.

It must be considered that the exact weld shape of MINA is very difficult to determine in the MSC calculation, because the molten shape strongly depends on the applied heat source parameters and material properties used (mass, heat transfer, specific heat). Therefore, the worst and the best cases are shown in Figure 41 .

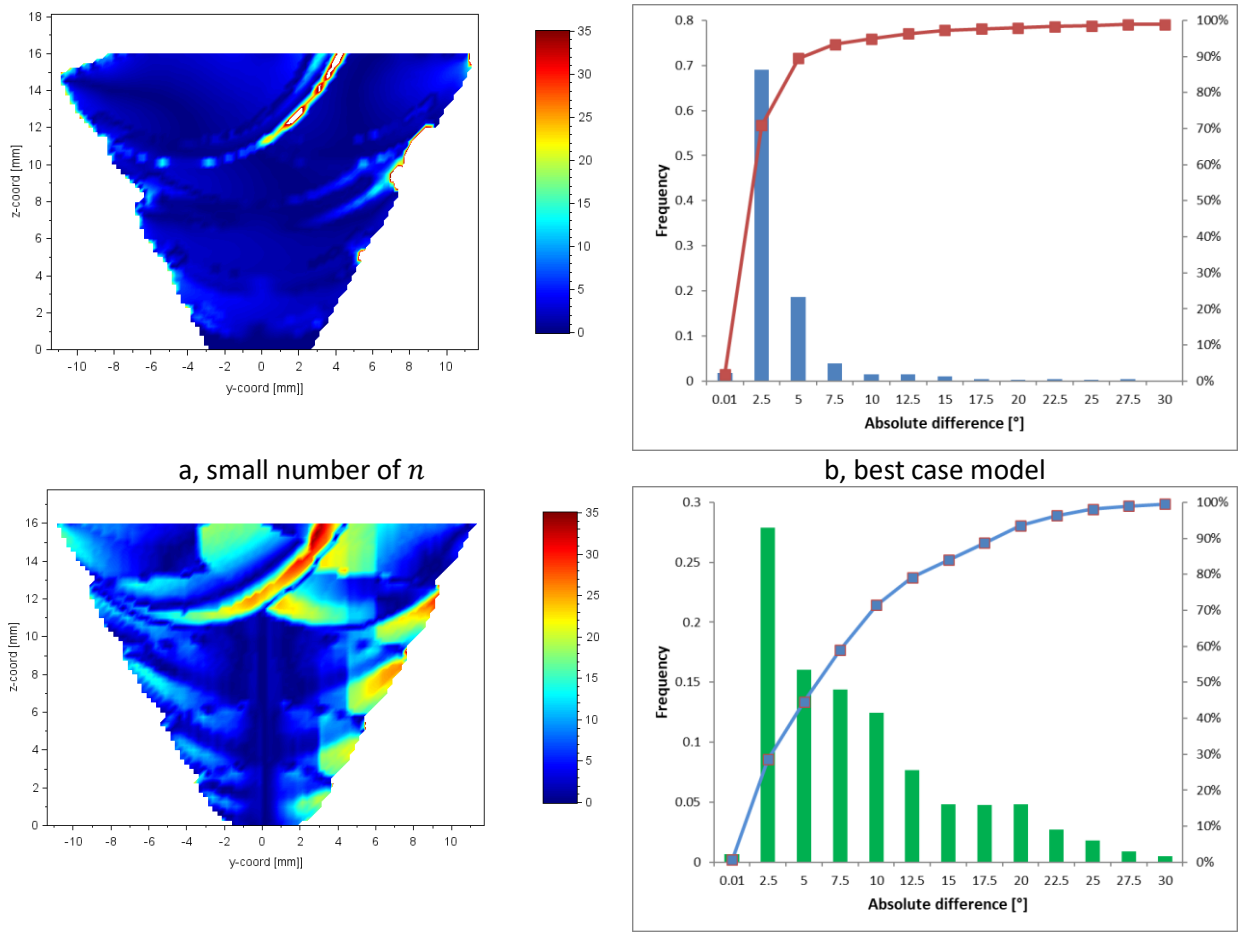


a, The bead shapes b, FE representation of the MINA shapes

Figure 40: FE model of 6 pass weld

Only the fully molten region is used for comparison. The effect of the weld pool differences between the models is clearly visible, originally most of the errors coming from these regions. Furthermore, the iteration parameter is also important in the FE calculation, if the number is too high, then α_n tends very rapidly moving away from α_{n-1} , which means that a grain tends to the new temperature gradient direction (in Figure 41/c), therefore it should be taken into account in the FE calculations.

The error between the best case model and MINA is acceptable in this model which is used in further calculations.



a, small number of n b, best case model

c, large number of n d, worst case model

Figure 41: Comparison between the simulations

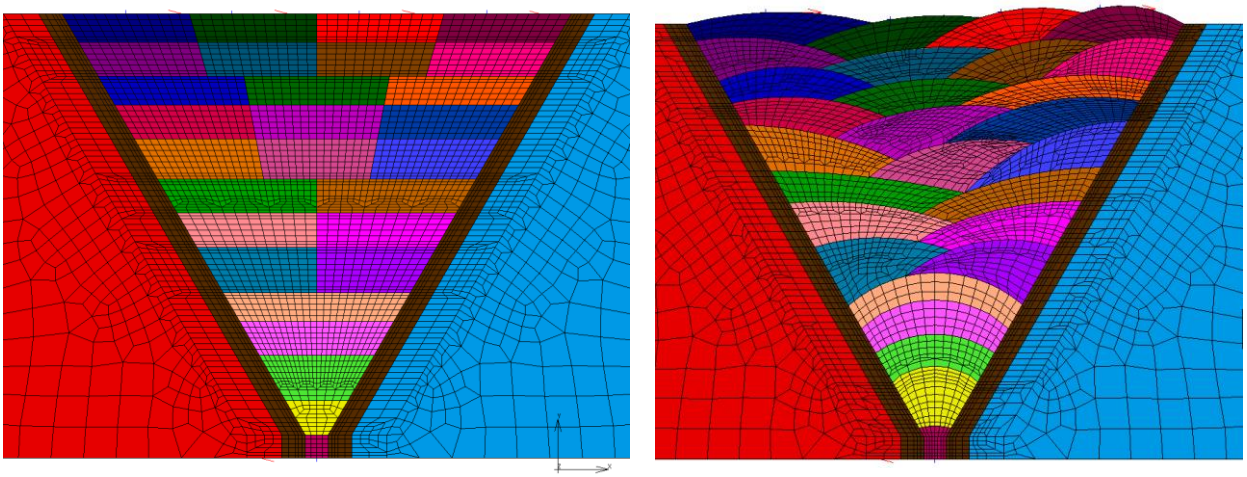
3.5.2 Sensitivity analyses on FE simulation of the weld pool shape

The previously described results have demonstrated the uncertainties and their effect on the weld pool formation using the original theory of grain orientation calculation. It is well known that the shape and temperature distribution of a weld pool is dependent on the type of welding processes, welding parameters and the weld materials. During the solidification of the weld pool, the ratio of temperature gradient and solidification speed along the welding pool boundary have an important influence on the weld solidification structure morphologies. The heat (heat source) and mass (molten volume) transfer in the weld pool determine the weld profile, which is related to the solidification conditions along the welding pool boundary, and affect the weld solidification structures. The investigation of the effect of the welding parameters on the weld profile in the weld pool is useful to understand the formation mechanism of the various weld solidification structures. For all arc welding processes which were analysed, the weld pool size is determined by the welding line energy and the welding speed. In order to analyse the morphological

evolution of the weld solidification structures, the weld geometry should remain constant under different welding conditions. Generally, higher welding power is applied with increasing welding speed to obtain the same shape of the weld pool. In this section, the effect of changes in the welding speeds and the applied welding power were analysed using the FE model of the original WP2 sample and have been done also in 3D FE model.

3.5.2.1 Thermal analysis on WP2 circumferential weld joint

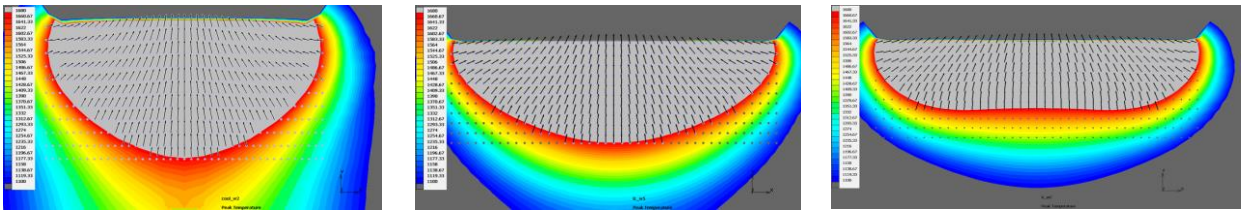
Numerical simulations were performed in two-dimensional axisymmetric model, which have been presented before. In addition to the analysis of the weld macrostructure a new 2D FE model was built according to fusion lines of the WP2 MU. The original model was used to model the effect of line energy modification and includes the effect of tilting during the welding and a change of the welding power.



a, old approach b, new approach

Figure 42: 2D FE models of WP2 sample

On the other hand, in case of SMAW we need to consider another effect during the simulations. In the welding process, the so-called weaving technique has been used presumably. By using the weave bead method, weld beads are deposited in a zigzag formation. This is a common technique for making wide weld rows. An exact simulation of the weaving motion was impossible to achieve, its effect has been analysed applying multiple heat source during the calculation using the new FE model. The simple effects of the weaving technique on the pool shape and the temperature gradient orientations are shown in Figure 43.



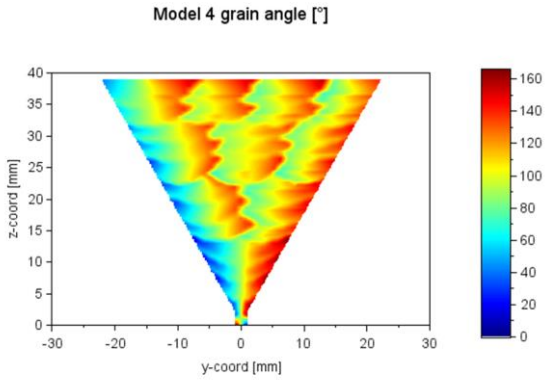
a, no weawing b, weawing: 2 heat source c, weawing: 3 heat source

Figure 43: Weld pool formation in case of SMAW

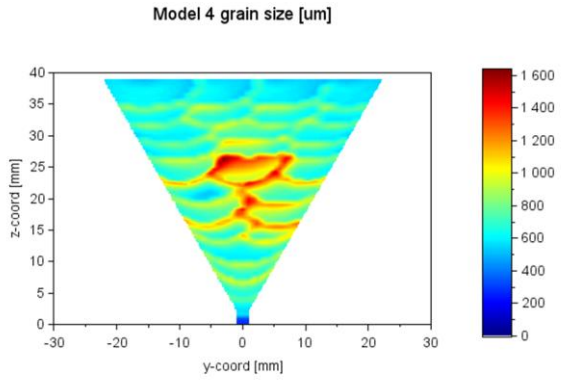
The effect of the unknown or presumed welding parameters were investigated during the 2D calculation, such as:

- Model-4 – it is the original model; the FE model was built on the original geometry using simplified weld areas. The sizes of the cross-sections of the beads and the heat sources were determined according to estimated welding parameters. After the 3rd row were used double heat source to achieve the melting of the beads. – This model using as references.
- Model-5 – it is based on Model 4, but after simulating the 4th row the effect of +8° electrode tilt was considered. It means that the heat flow direction is modified, therefore the shape of the molten regions also changed, and the direct power is decreased. (Figure 45)
- Model-6 – in this model after the 4th row the effect of -5° electrode tilt was considered. The analysed effect is the same as for the previous model. (Figure 46)
- Model-7 – in this model increased welding power was used. It means the line energy increased 10 percentages, therefore all molten regions also increased. (Figure 47)

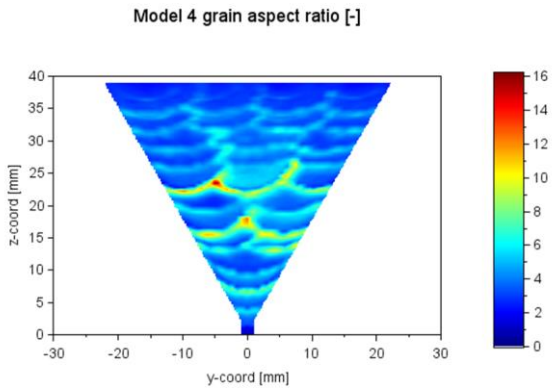
All simulation results were visualized on an equidistant grid with 0.25mm resolution. The calculation of the grain size distribution on a 1x1mm^2 grid resolution was considered. The results of the old approach (Model-4) are shown in Figure 44.



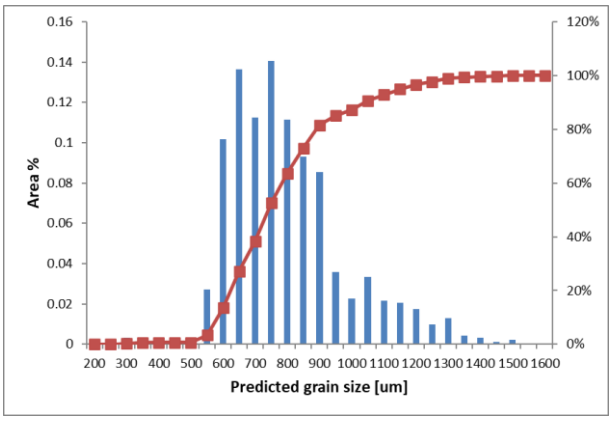
a, The predicted grain angle



b,The predicted grain size



c, The predicted grain aspect ratio

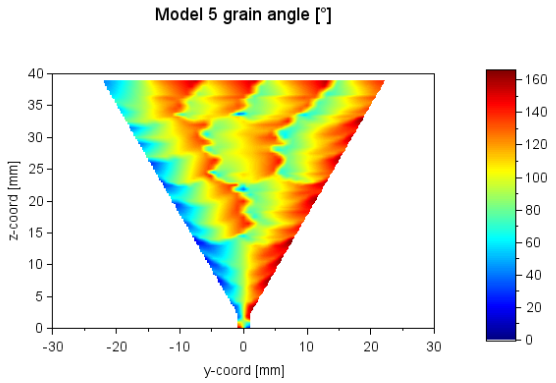


d, Grain ditribution on 1x1mm

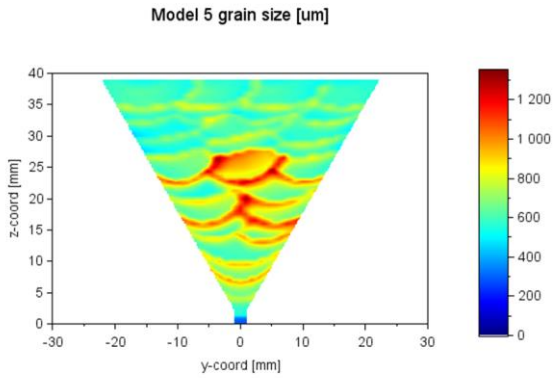
Figure 44: The result of Model-4

In case of the first variation the same mesh was used as for Model 4. The difference between the two models lies in the effect of the electrode tilting. While in the original model it is not considered, in this model the electrode is tilted between #5-27 passes. The angle of the tilting is equal to 8° for the tilted passes. The results are shown in Figure 45. In this figure it is clearly shown that after the 4th pass the

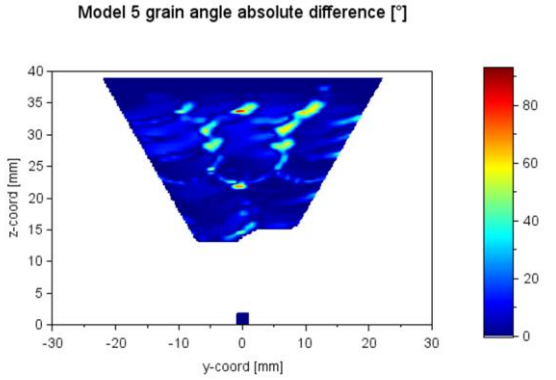
differences of grain orientation between the two models are considerably large locally. The predicted grain size is changed significantly however the maximum values of the grain aspect ratio are the same.



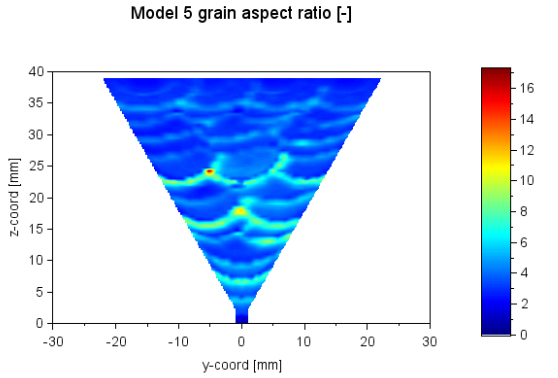
a, The predicted grain angle



b,The predicted grain size



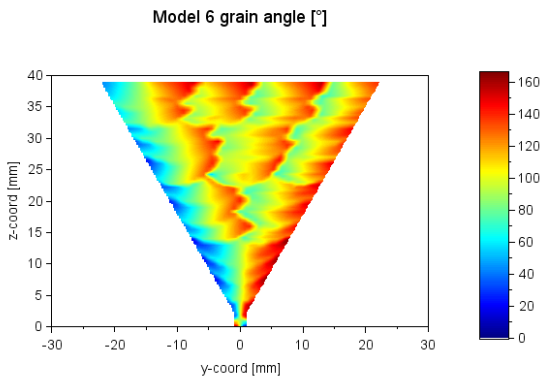
c, Grain orientation differences between model 4 and model 5



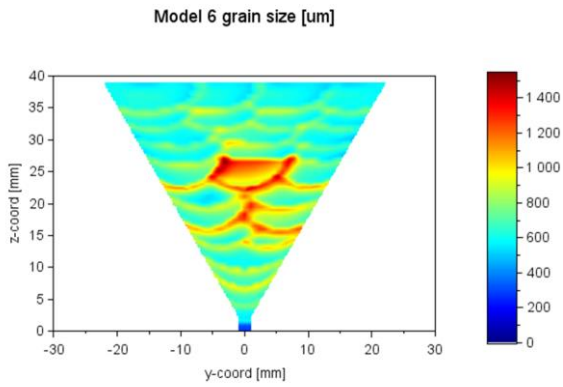
d, The predicted grain aspect ratio

Figure 45: The result of Model-5

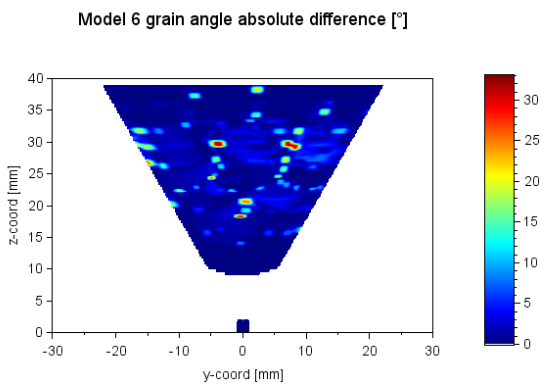
In case of Model-6 also the electrode is tilted between #5-27 passes. The angle of the tilting is equal to -5°. The differences of grain orientation are lower than expected, and the changes of grain size distribution are not very significant.



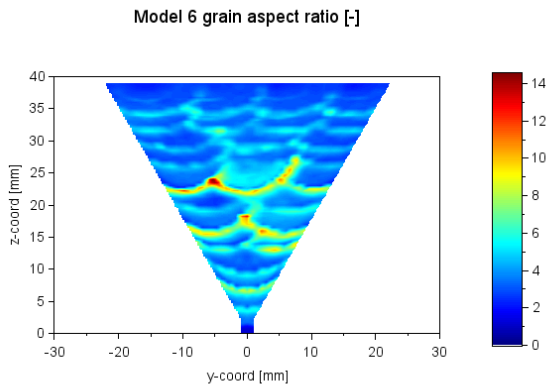
a, The predicted grain angle



b,The predicted grain size



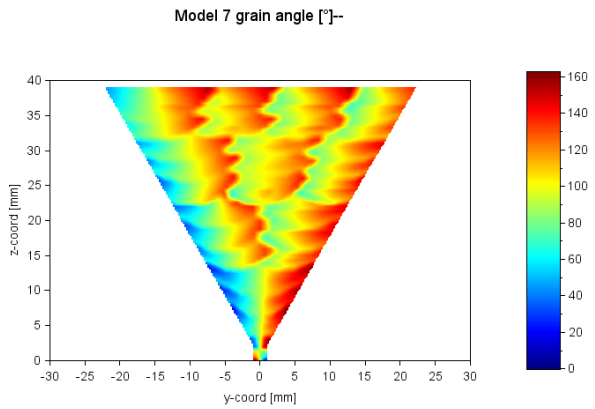
c, Grain orientation differences between Model-4 and Model-6



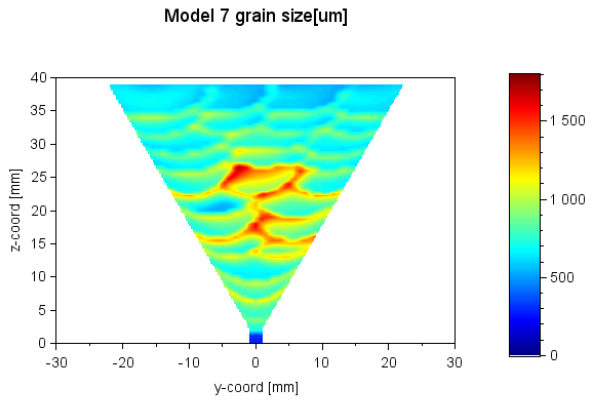
d, The predicted grain aspect ratio

Figure 46: The result of Model 6

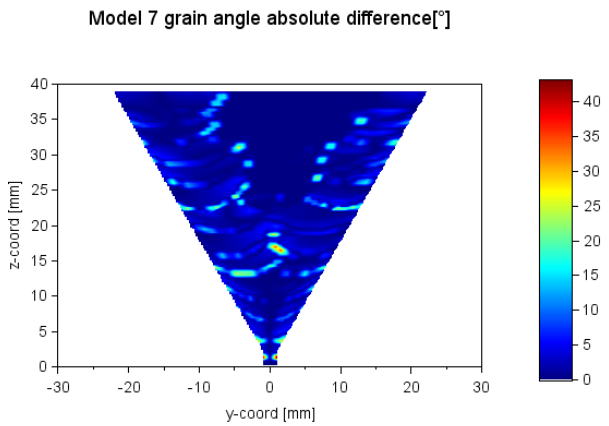
In case of Model-7 the applied power is changed. Increasing the heat input the shape of the molten pool is also changed. Comparing the calculated grain orientations, the deeper regions are clearly visible, but the global differences are not too significant. Changing the applied line energy, the global grain size is increased.



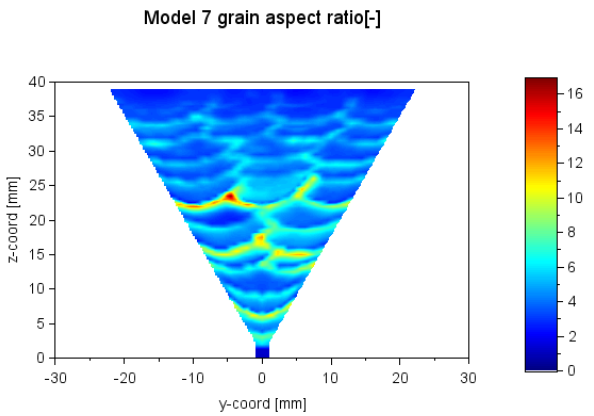
a, The predicted grain angle



b,The predicted grain size



c, Grain orientation differences between Model-4 and Model-7

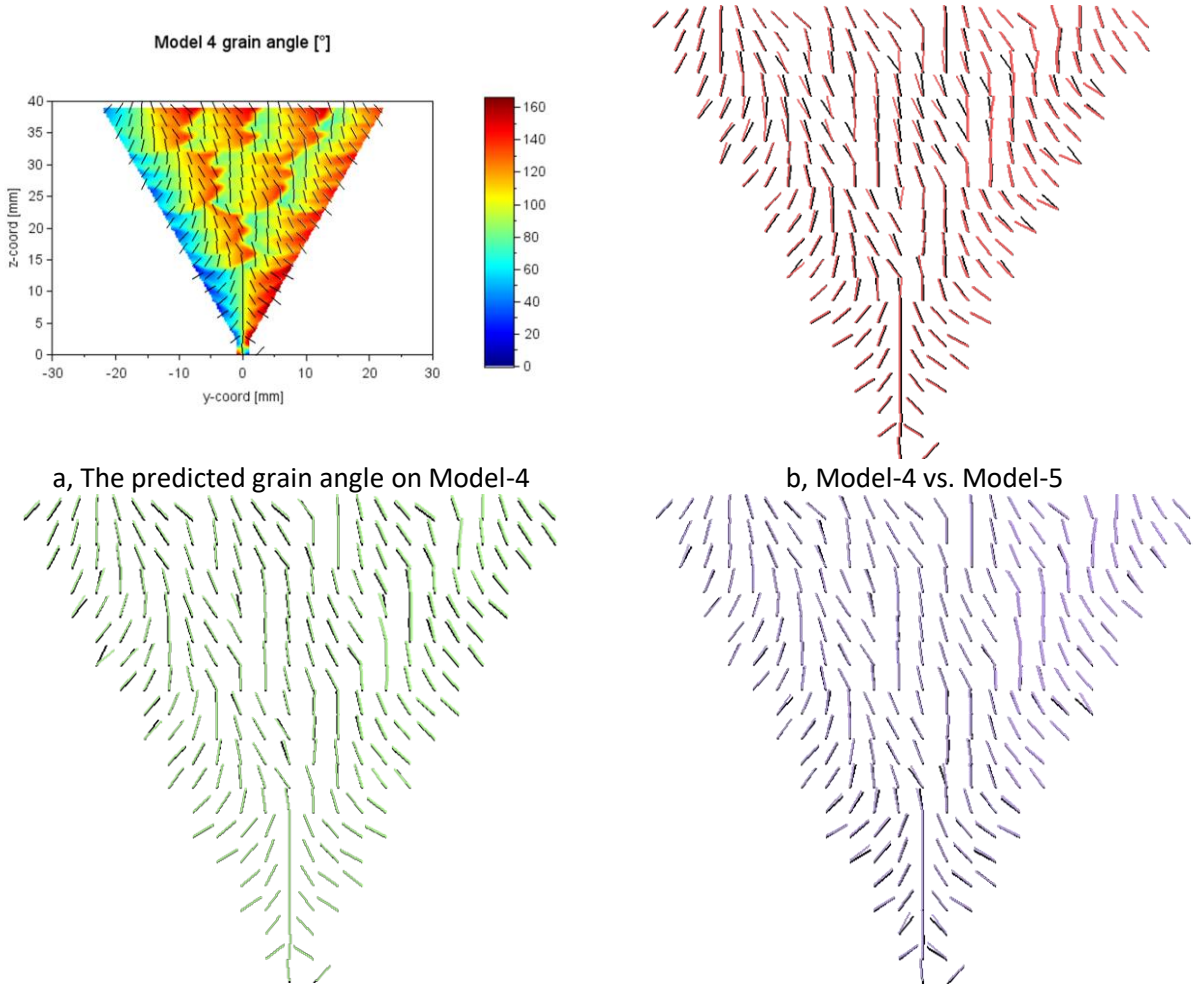


d, The predicted grain aspect ratio

Figure 47: The result of Model 7

To compare the different cases with the original model, it is necessary to use some simplifications. The grain orientation matrix is calculated on the same FE mesh and post processed on the same grid with 0.25x0.25 mm scale, but for the accurate ultrasound modelling a 2x2 mm scale is sufficient, therefore the calculated results are transformed to 1x1mm and 2x2mm scale using area-based averaging.

Using the same 2x2 mm scale the compared results show that globally, the electrode tilt significantly modifies the grain orientation (Figure 48/b).



a, The predicted grain angle on Model-4

b, Model-4 vs. Model-5

c, Model 4 vs. Model 6

d, Model 4 vs. Model 7

Figure 48: The comparison of grain orientation at 2x2 mm scale

Comparing the result of the Model-5 (red) with available measured grain orientation (orange) presented by ICL is shown in Figure 49. The predicted grain orientation on this perpendicular plane shows reasonable agreement with the EBSD scans.

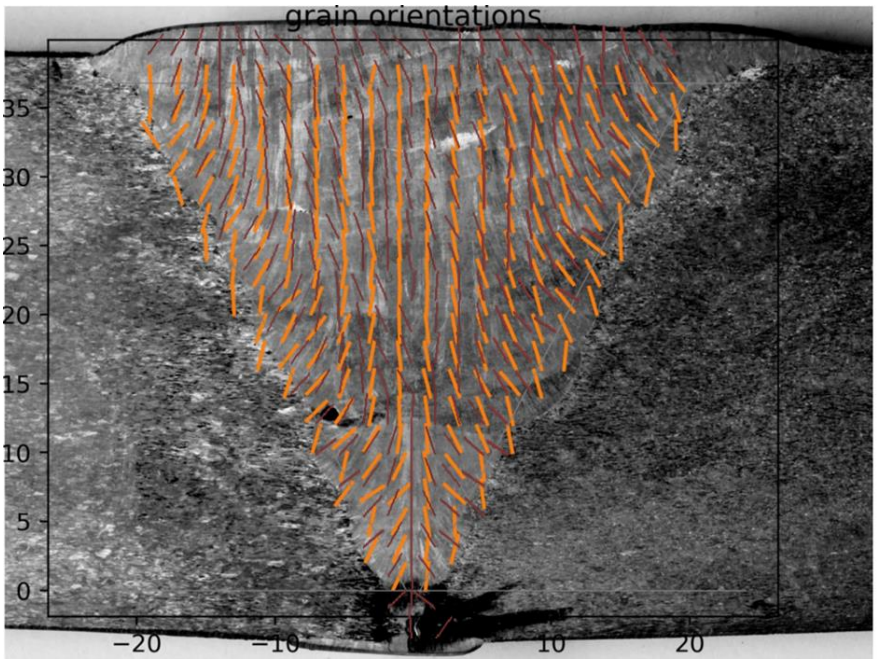
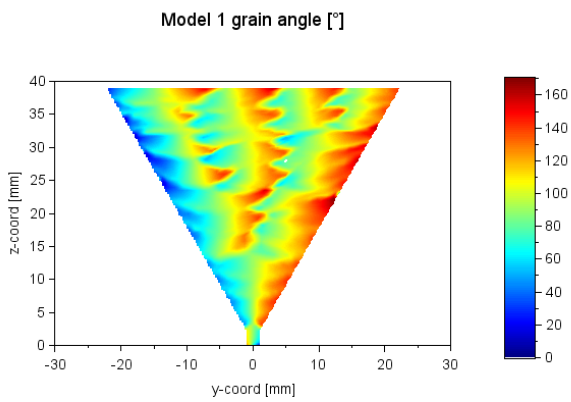


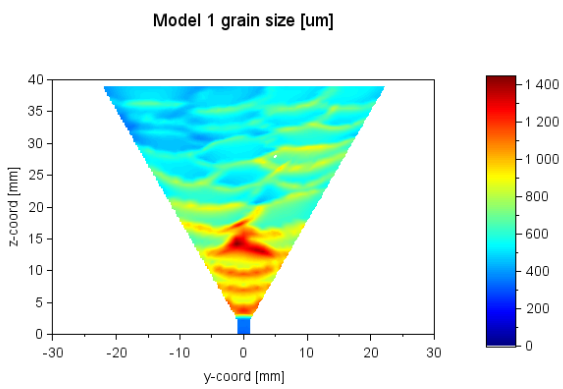
Figure 49: The measured and predicted grain orientation

The second model:

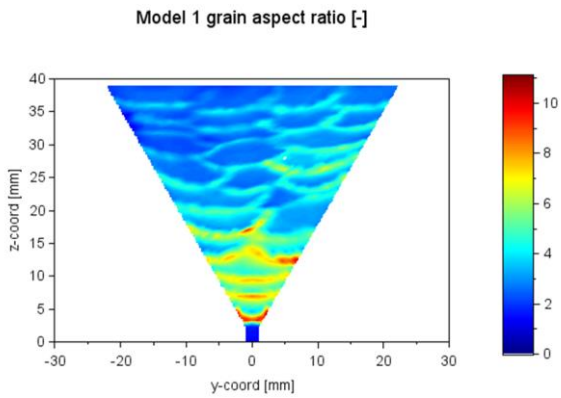
- Model-1 – this is a new approach model. The FE model was built on the original geometry using bead shapes from the microstructure. The total bead area was determined based on literature and known welding parameters. After the 3rd row the weaving technique was considered, according to the literature. (Figure 50)
- Model-2 - in this model smaller weaves were used (half electrode size) during the simulation. This means larger heat input and deeper penetration of the heat source. (Figure 51)
- Model-3 – in this model large weaves were considered (the size of approximated beads). It means lower heat input and shallow penetration. (Figure 52)

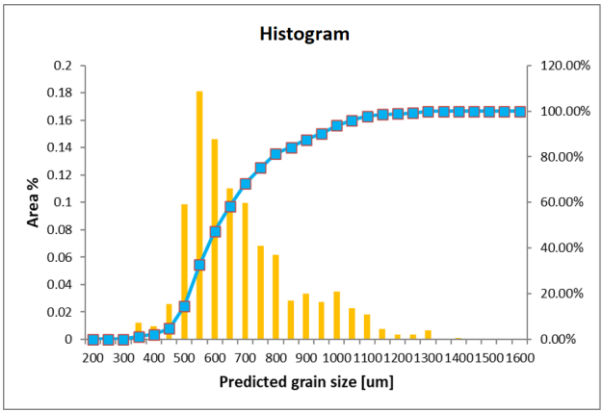


a, The predicted grain angle



b,The predicted grain size



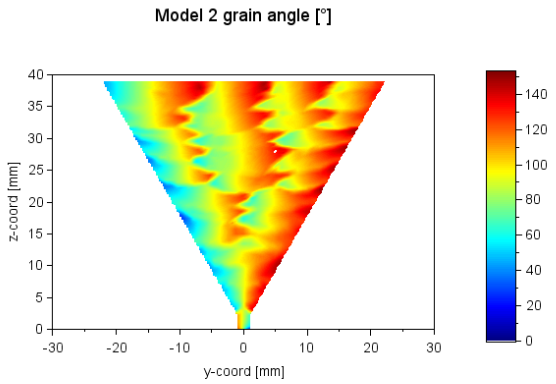


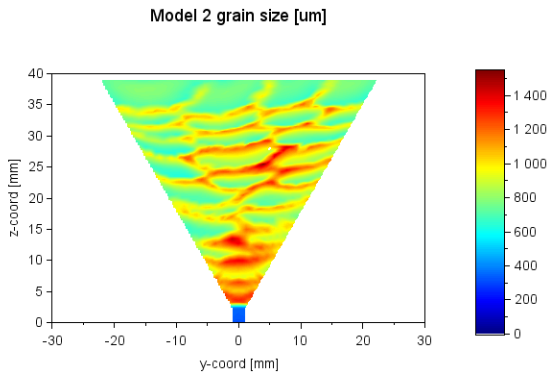
c, The predicted grain aspect ratio

d, Grain ditribution on 1x1mm

Figure 50: The result of Model-1

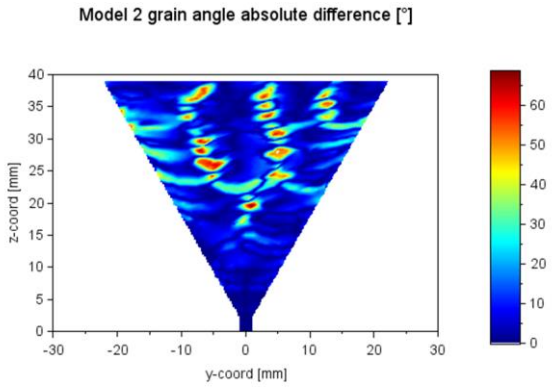
In case of Model-2 the welding technique is changed which is considered in the FE model using double heat source. Decreasing the weaving of the electrode the shape of the molten pool is also changed.

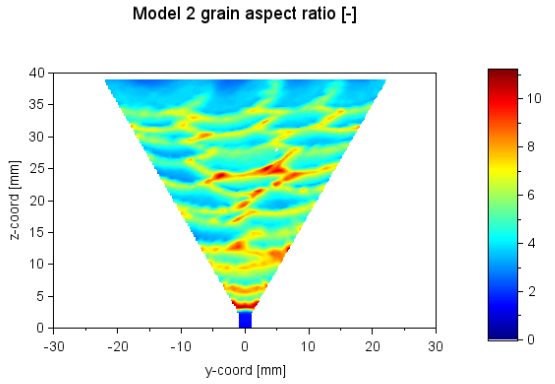




a, The predicted grain angle

b,The predicted grain size





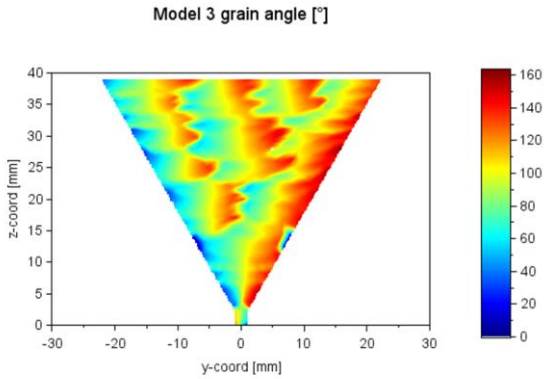
c, Grain orientation differences between Model-1 and Model-2

d, The predicted grain aspect ratio

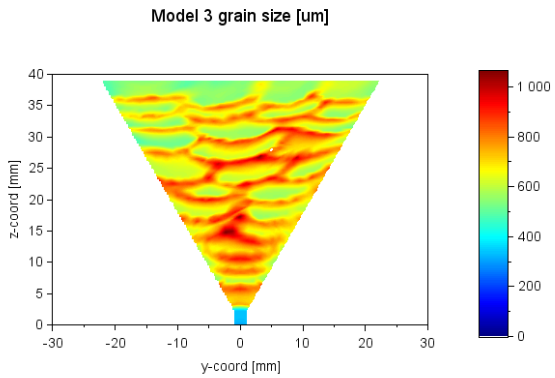
Figure 51: The result of Model-2

Comparing the calculated grain orientations, the deeper regions are also clearly visible, and the local differences are significant. This change of the weaves means that the applied line energy increases, and as a result the global grain size is increased.

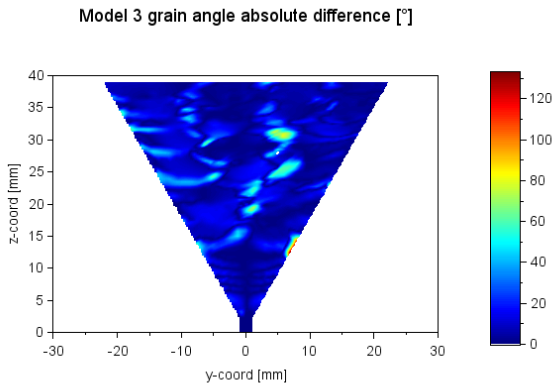
Analysing the result of Model-3, Figure 52 shows the opposite effect. As the electrode wave increases, the shape of the molten pool is shallower than before, therefore the applied line energy decreases; and as a consequence, the global grain size is also decreasing.



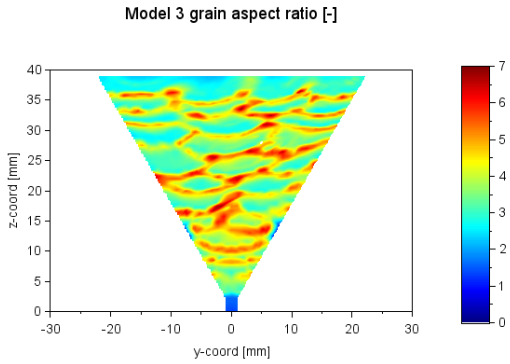
a, The predicted grain angle



b, The predicted grain size



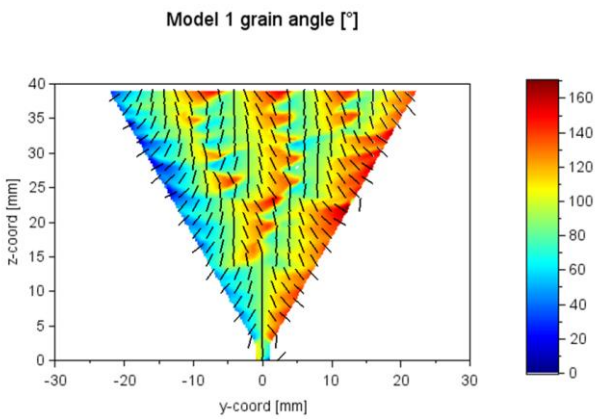
c, Grain orientation differences between model 1 and Model-2



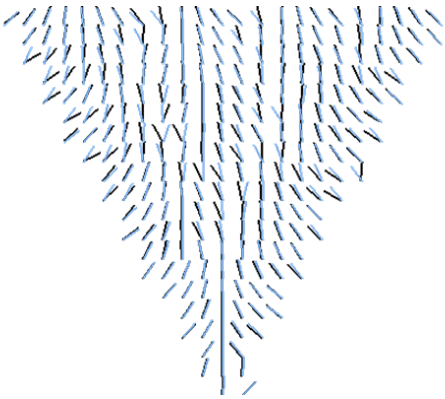
d, The predicted grain aspect ratio

Figure 52: The result of Model-3

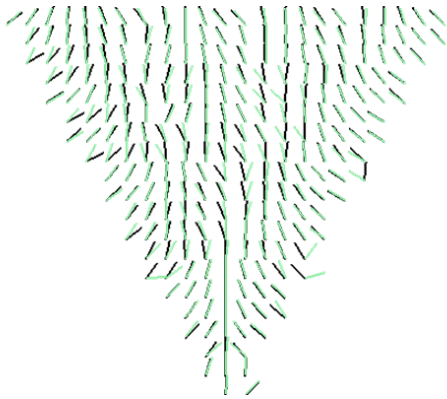
Comparing the result of grain orientation on 2x2 mm scale (Figure 53) the applied welding technique significantly affected the grain orientation. This means that for accurate modelling of the grain orientation it is necessary to consider the technique used during the given welding process.



a, The predicted grain angle on Model-1



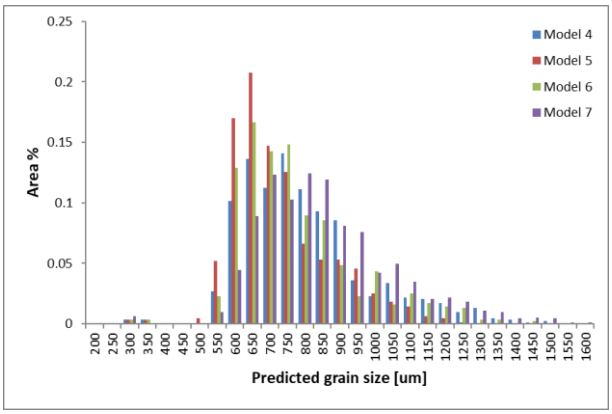
c, Model-1 vs. Model-2



c, Model-1 vs. Model-3

Figure 53: Compariosn of the predicted grain angles

Furthermore, comparing the effect of the analysed unknown welding parameters on the grain size and grain aspect ratio, the associated histograms are shown in Figure 54 and Figure 55. As expected, the effects which increase the line energy of the welding also increased the grain size but using the weaving technique and/or using the electrode tilt the grain size and grain aspect ratio is decreased.



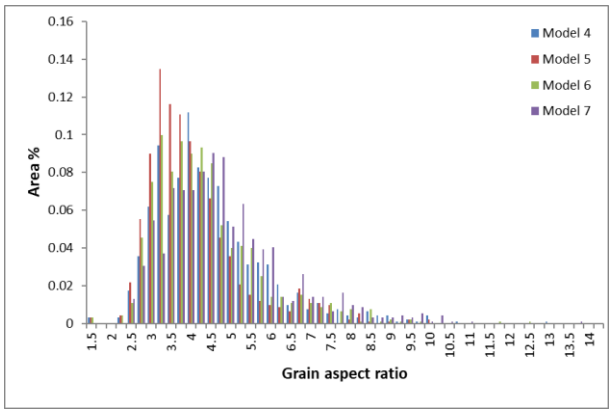


Figure 54: The predicted grain size and aspect ratio from the original model

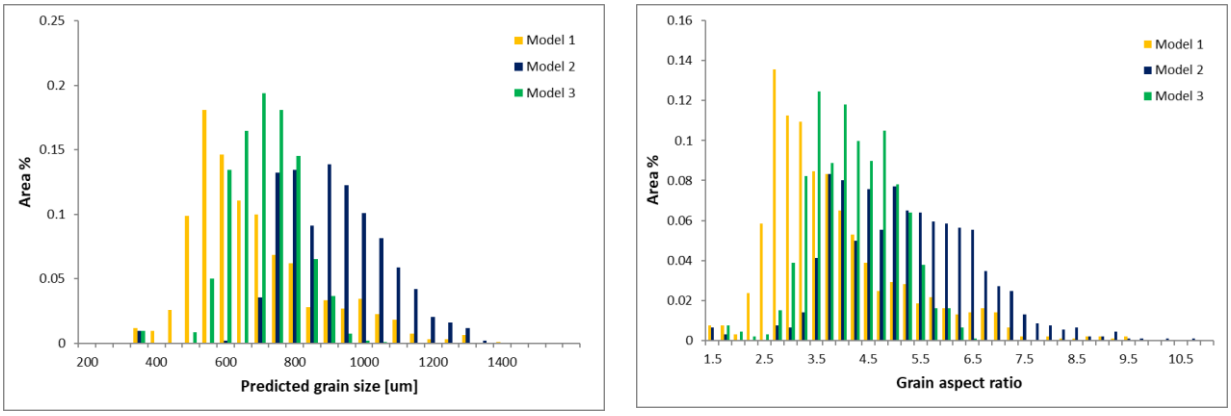


Figure 55: The predicted grain size and aspect ratio from the new model

It has been demonstrated that the determination of the grain orientation is a faceted problem, with many influencing parameters. The effect of the applied welding technique has been investigated systematically. A sensitivity analysis is performed focusing on the effect of electrode tilt and the characteristic parameters of the double ellipsoidal heat source which modifies the shape of the weld pool in 2D FE analysis.

3.5.2.2 3D thermal analysis on a multipass welding

This example represents a multilayer welding model with two layers (Figure 56). The aim was to check the developed code in case of 3D simulation on a simple model and to present the available results. The FEM pre-processing, calculations and post-processing have been carried out by MSC.Marc and Scilab softwares.

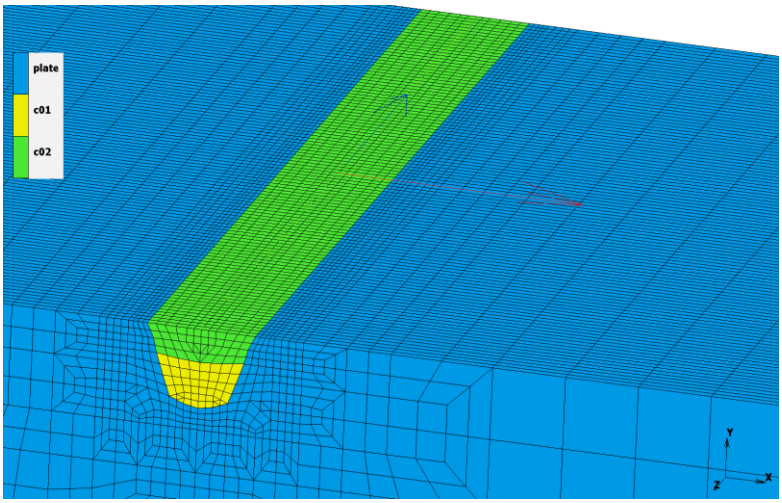
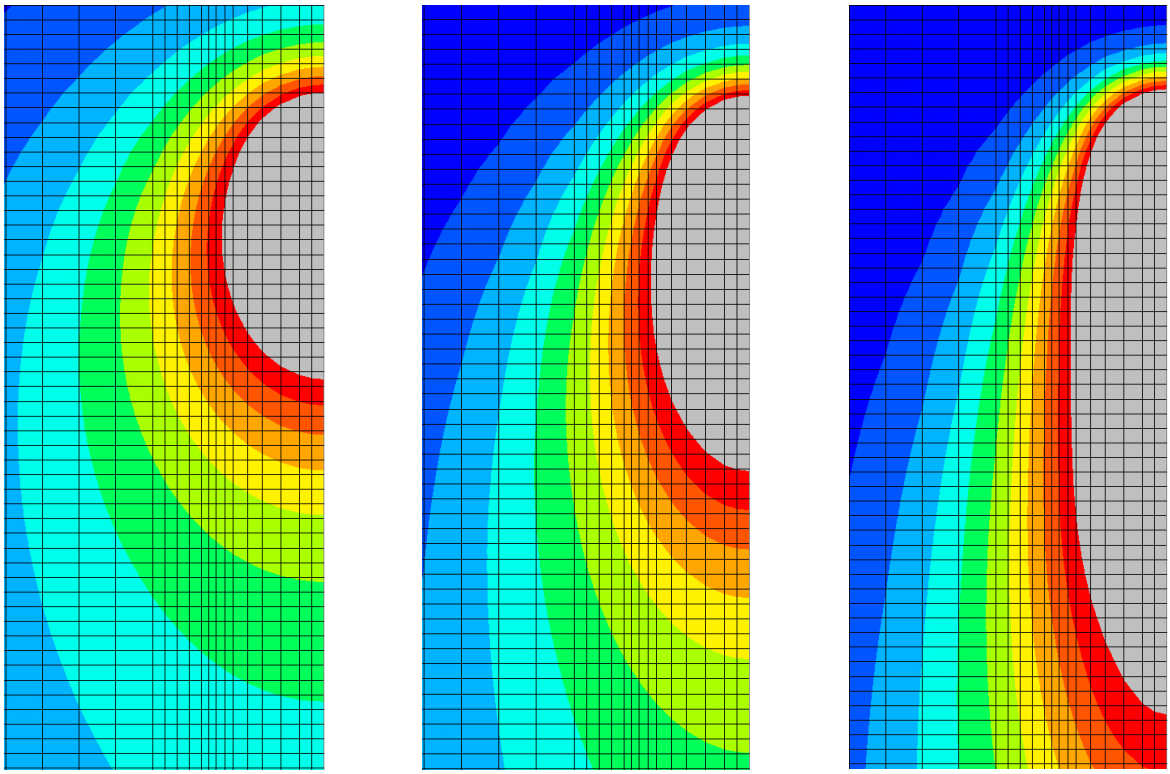


Figure 56: The FE model of multilayer welding

MSC.MARC supports two methods for the creation of filler elements: the quiet element method and the deactivated element method. The analysis used the deactivated element technique. Using MSC.MARC code to implement material addition or removal from the model is very suitable for simulating welding processes. The technique requires the complete model, including all material volume during the whole process, to be defined in advance and meshed. In the deactivated element method, filler elements are initially deactivated in the analysis and are not shown on the post processed file. When the elements are physically created by the moving heat source, they are activated in the model and appear on the post file. During the analysis, a full Newton–Raphson iterative solution technique with a direct sparse matrix solver is used for obtaining a

solution. During the thermal analysis, the temperature and the temperature-dependent material properties change very rapidly. Thus, it is believed that a full Newton–Raphson technique using modified material properties gives more accurate results. All free surfaces of all FE-models are given a convective heat loss with a heat transfer coefficient, h=25W/mK and a radiation heat loss using an emissivity coefficient, ε=0.8. Inactive elements have been activated initially to simulate the addition of filler material. The temperature-dependent material properties of analysed austenitic steel were generated with JMatPro software based on its chemical composition.

According to the previous section, higher welding currents are applied at increasing welding speed. In the present example, the welding speeds used in the calculation are 0.67, 2, and 4 mm/s, respectively, which is the standard speed for SMAW. Also, the welding power changes correspondingly when the same pool is formed. The simulated results of the weld pool shape for various welding conditions are shown in Figure 57.



a, v=0.67 mm/s b, v=2 mm/s c, v=4 mm/s

Figure 57: The results of the weld pool shape for different welding parameters (top view)

It can be seen that with the increase of the welding speed and welding power, the weld pool was elongated along the welding direction, and the weld pool shape changes gradually from the elliptical to the strongly elongated shaped. The solidification conditions along the molten pool boundary are related to the weld pool shape. Based on the discussions in previous sections, the morphological evolution of the weld solidification structures is strongly depending on different welding parameters. The entire weld has been simulated in this section from to start of the arc to the end to analyse which cases are needed the 3D calculation to determine the grain orientation. The analysis is focused on where the welding process is stable and ignores the effect of the arc start/end.

The grain orientation produced in multilayer welding simulations is presented in Figure 58. It can be seen that in the case of flat welding the welding speed is not dominant for the grain orientation in the cross

section of the beads. The important parameter is only the line energy which produced the same weld pool shape in the cross section. On the contrary, the grain orientation is strongly depending on the speed along the welding direction. In a longitudinal plane, the simulated results show vertically aligned grains if the applied welding speed is higher. This figure also shows that the grain structure has good repeatability along the welding direction, indicating that a 2D FE model can reproduce the results of a 3D model in the transverse plane.

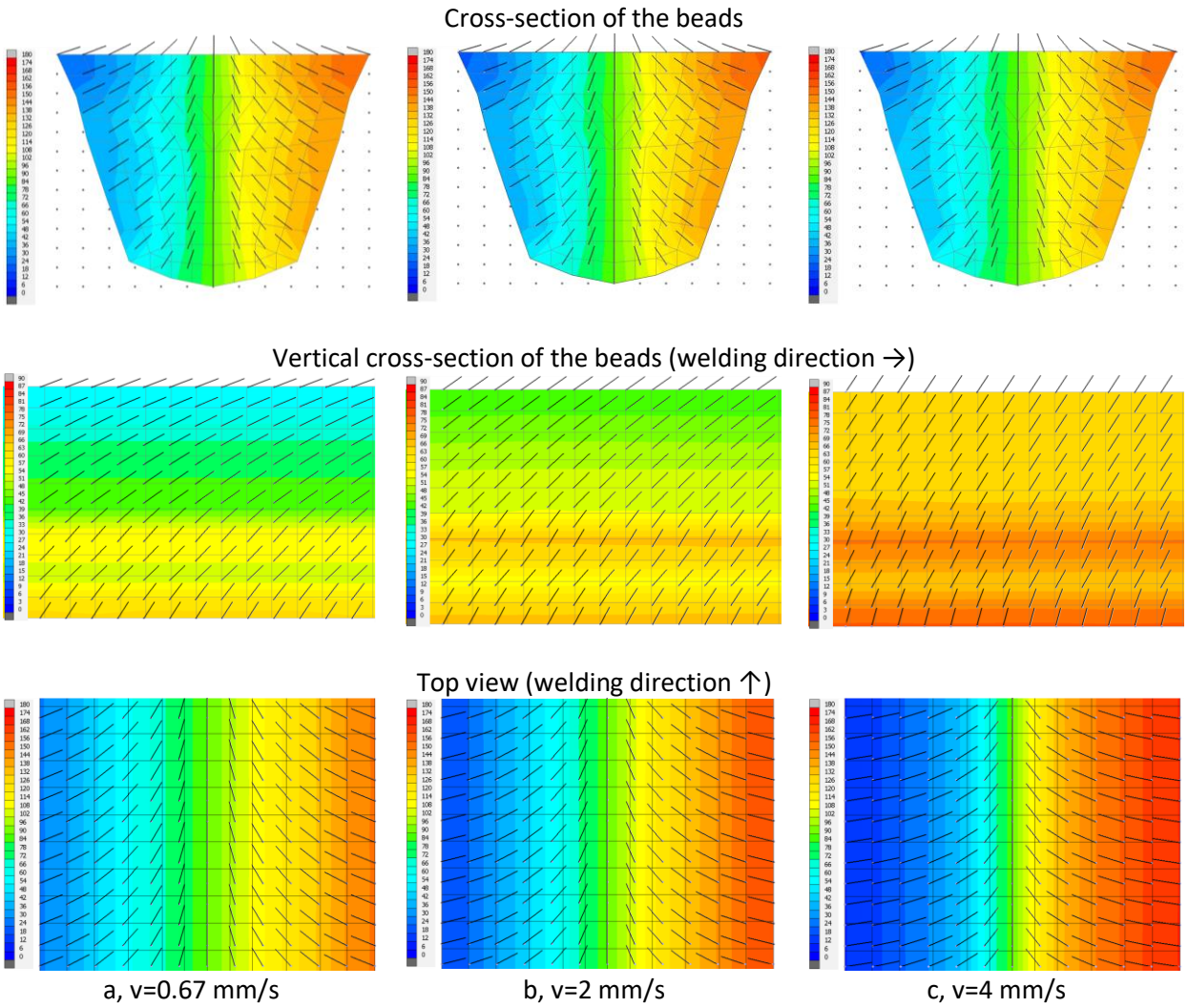


Figure 58: The predicted grain orientations [°]

Moreover, the finer less elongated grains are obtained at this higher welding speed.

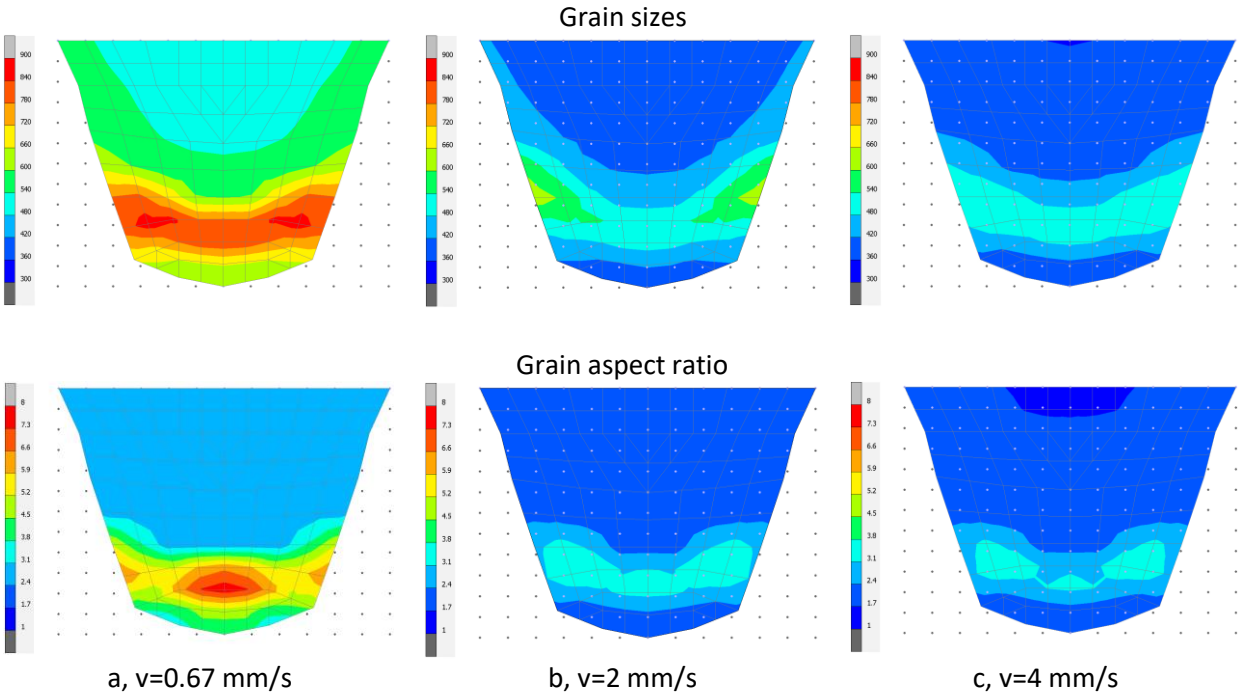


Figure 59: The predicted grain size [µm] and grain aspect ratio [-]

This result demonstrates that this FE simulation technique can reasonably predict grain orientation in both 2D and 3D cases during the multipass welding.

3.5.3 3D simulation on WP2 MU

Based on the axisymmetric model, the WP2 mock-up is also simulated by 3D thermal analysis. The simulations are performed on simplified geometries (Figure 60) because the temperature field changes between the beginning and the end of the arc start are the same. The FEM pre-processing, calculations and post-processing have been carried out by MSC.Marc and Scilab softwares.

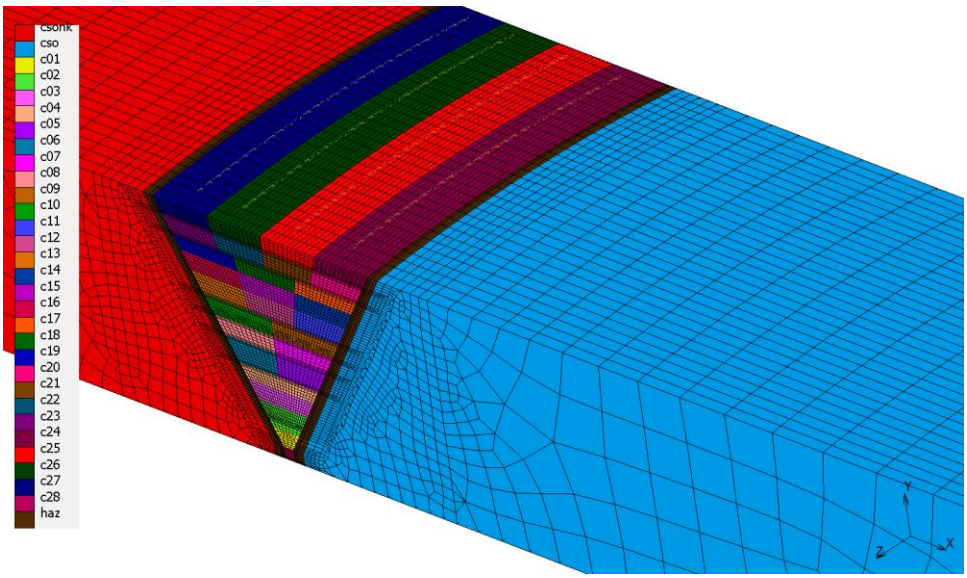


Figure 60: Simplified 3D FE model

The analysis was carried out using the same parameters which were using previously in case of 2D simulations. In case of multilayer welding using SMAW process, it is important to know the location where the process starts and finish, because in this region the local heat input is different from the other regions. In the present 3D simulation, all welding paths start and end at the same position, which does not correspond to reality, but allows to represent the differences in grain orientation.

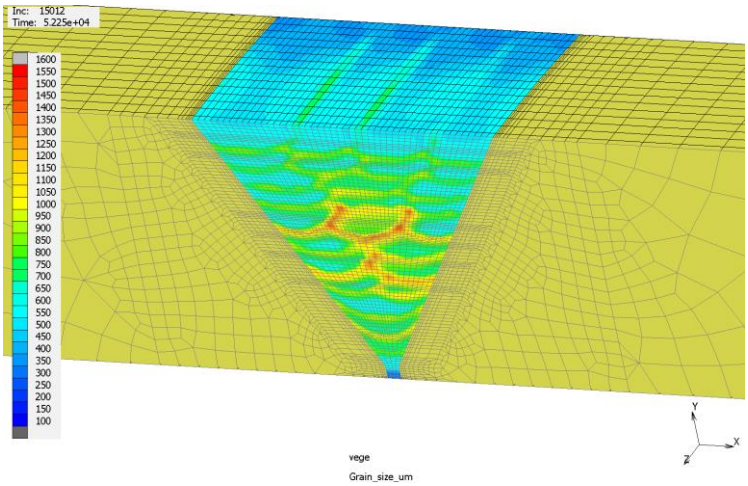
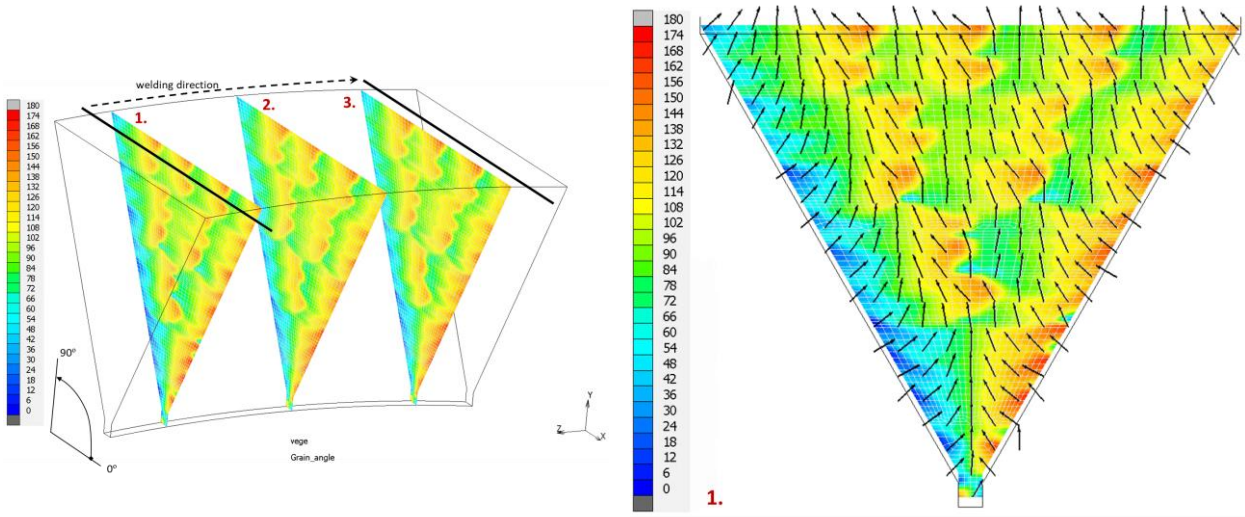


Figure 61: Grain distribution on 3D FE model

The calculated grain orientation results are visualised on the welded region (see Figure 61) using parallel planes with the global coordinate system. The distributions of the grain orientation on the cross-sections of the welded are shown in Figure 62.



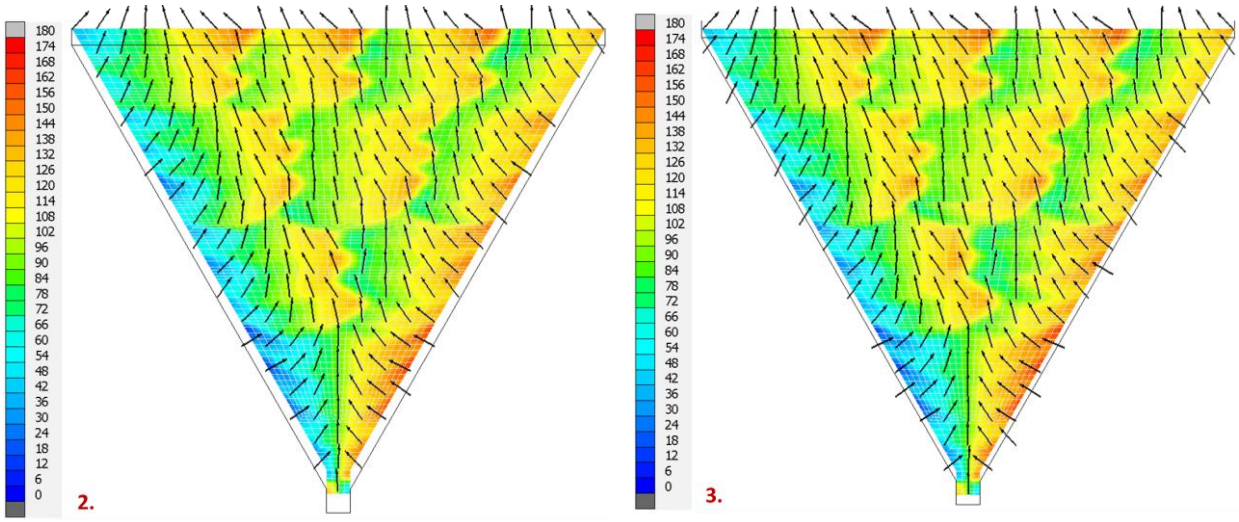


Figure 62: Grain orientation on weld cross-sections

Figure 62/1. represents the grain orientation next to where the arc started and Figure 62/3. next to where the arc finished. Comparing the orientation, the local differences between the cross-sections are significant, but when analysing the global orientation on a 2x2mm grid, the differences are negligible. However, Figure 63 shows the deviation between 2D and 3D simulations. It was clearly demonstrated that as expected, in the centre region of the weld the differences were not very significant but were locally high in the start and end position.

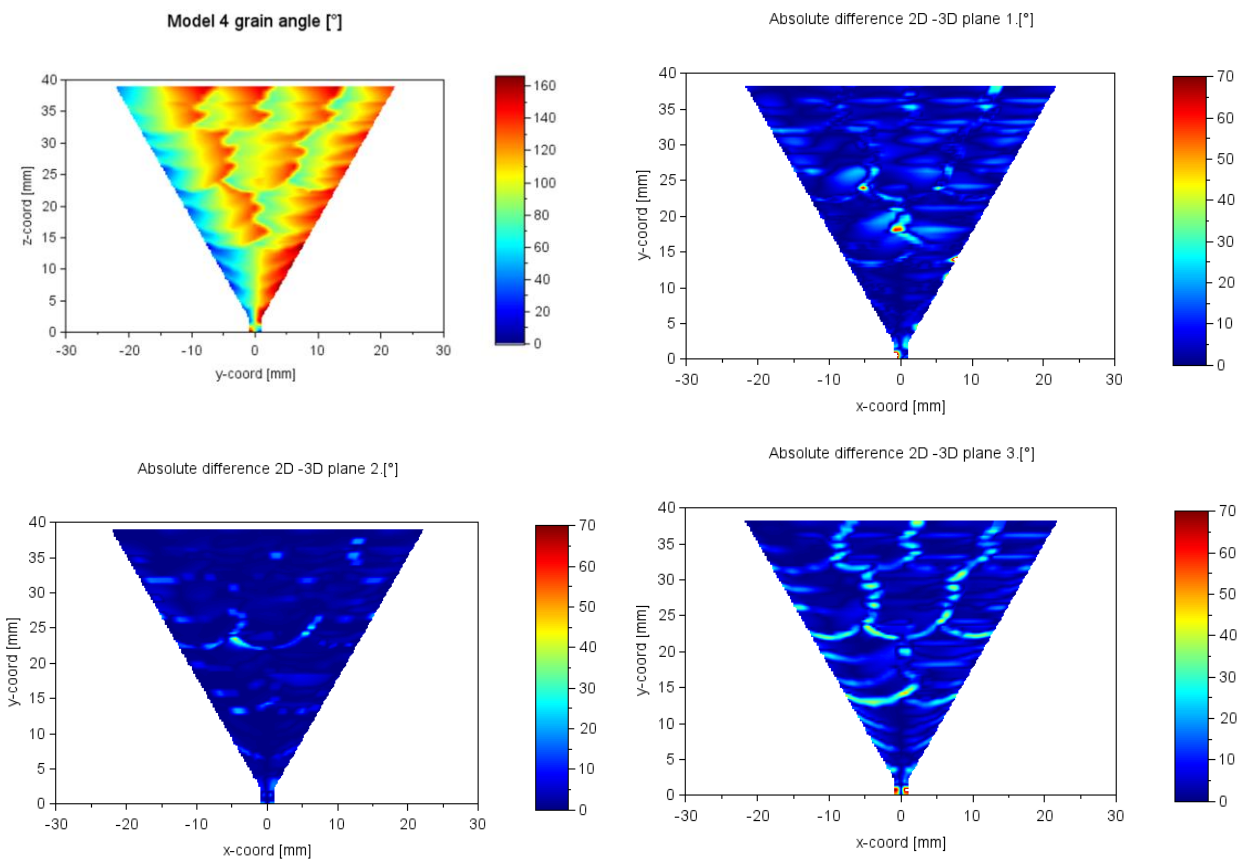


Figure 63: Comparison between 2D and 3D results

Figure 64 and Figure 65 show the derived orientation of the grains in the vertical and horizontal cross-sections of the welded region. In the presented longitudinal planes, the calculated results show vertically aligned grains where the heat source is moving. Along the weld paths, the grains are oriented in almost the same direction which is about 78° in the present calculation, but the orientation differs significantly in the weld start and end positions.

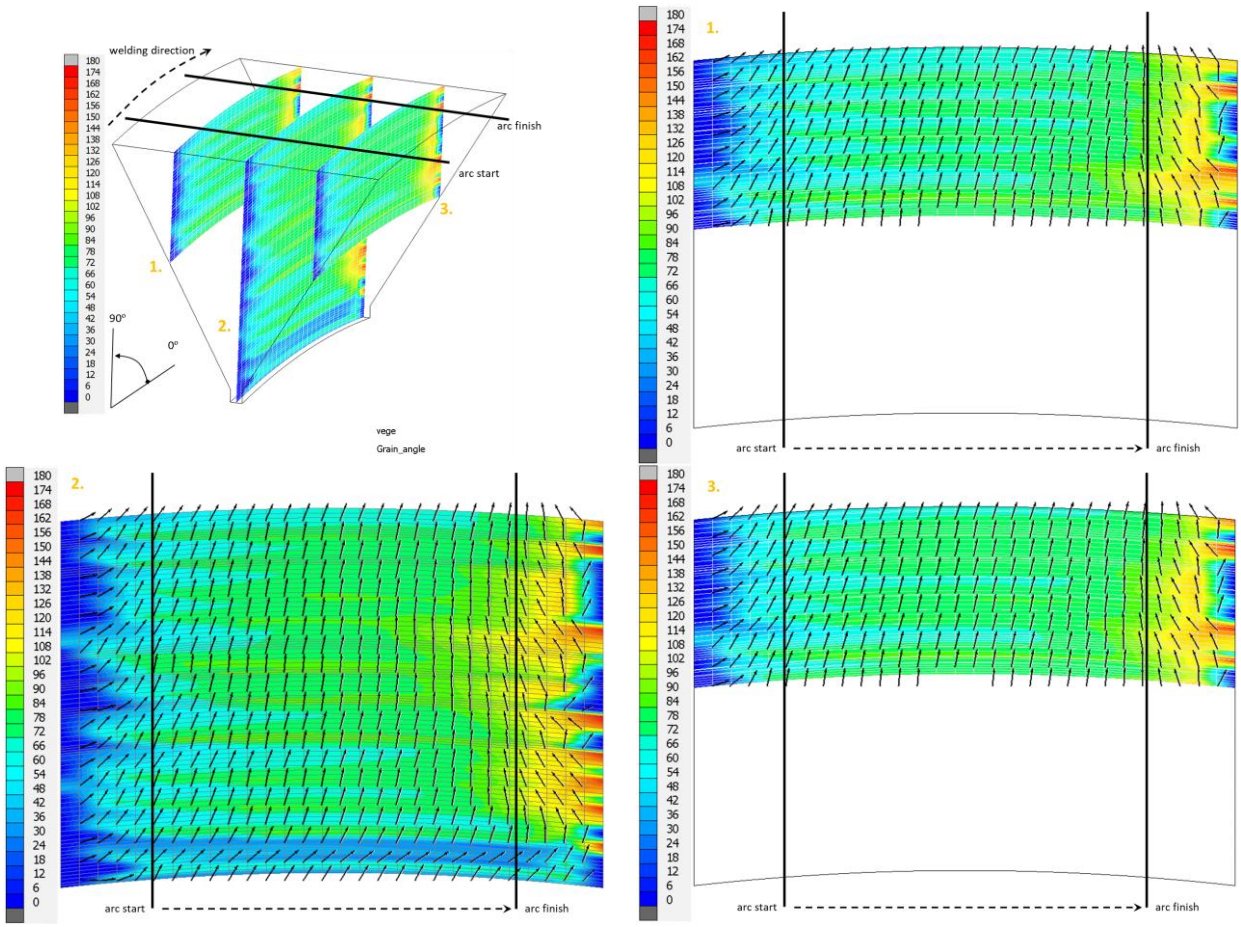
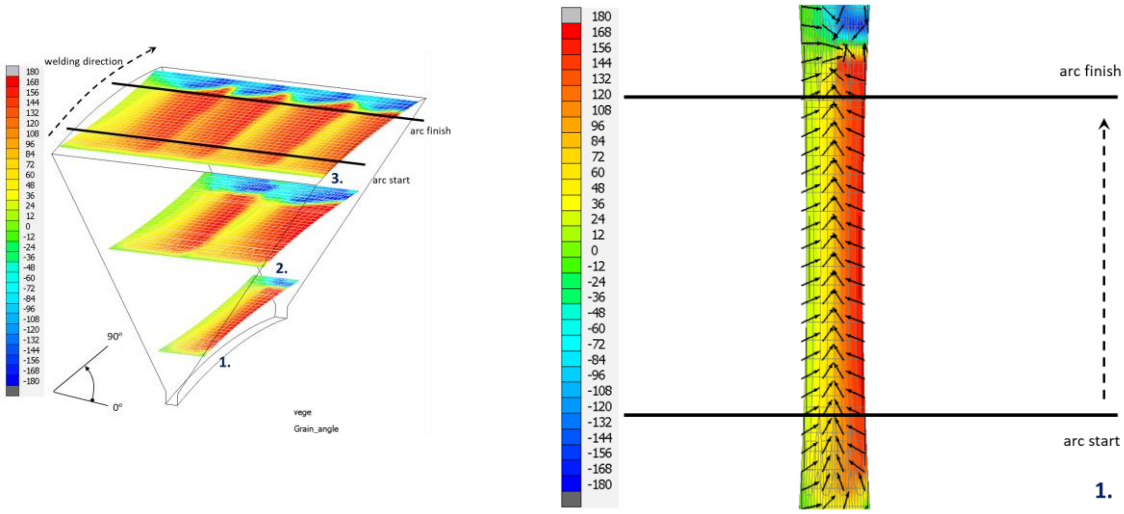


Figure 64: Grain orientation on vertical cross-sections of weld

In the horizontal planes the grains are oriented according to the moving heat source boundary. The effect of the beginning and end of welding is clearly visible in the calculated result.



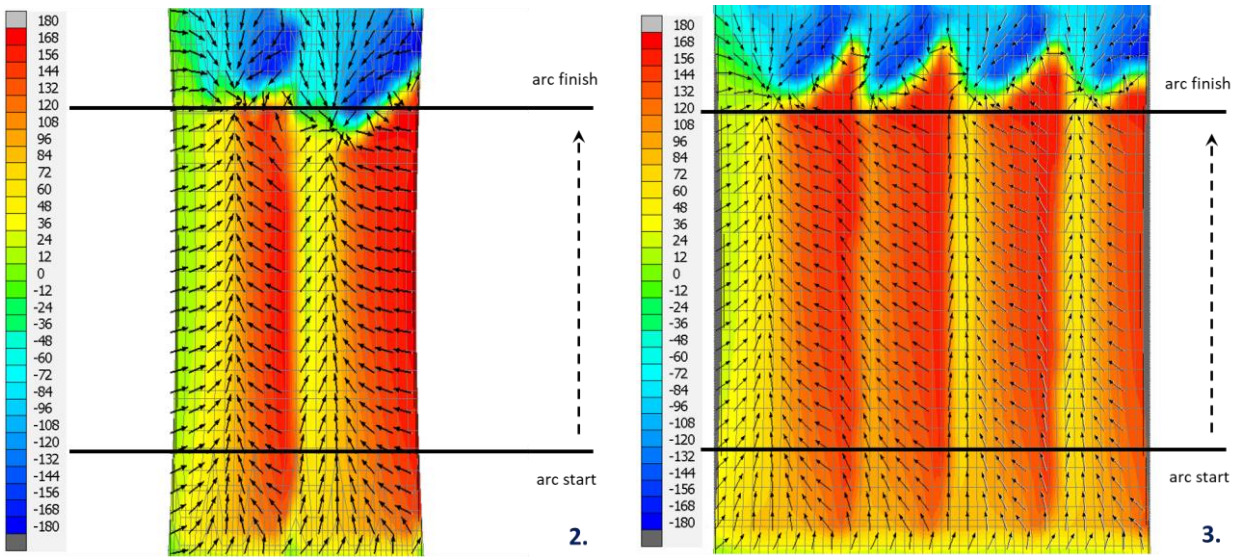


Figure 65: Grain orientation on horizontal cross-sections of weld

3.5.4 2D simulation on DMW replica

The mock-up reflects the DMW configuration of the VVER-440 reactors connecting the reactor pressure vessel with the hot-cold leg. It involves a bimetallic fusion weld with three buttering layers towards the ferritic side. The experimental mock-up with 40mm thickness is shown in Figure 66. Originally, steel 15H2MFA was the manufacturing material for the nozzle rings of DN 500 nozzles. The safe-ends of DN 500 nozzles were manufactured from austenitic steel 08H18N10T. However, this type of steel is no longer available so it is substituted with X6CrNiTi18-10 (1.4541), which has a similar chemical composition, as can be seen in Table 12. The base materials are widely different (15H2MFA ferritic steel and X6CrNiTi18-10 austenitic steel) therefore in order to perform the welding a transient cladding or cushion has to be implemented.

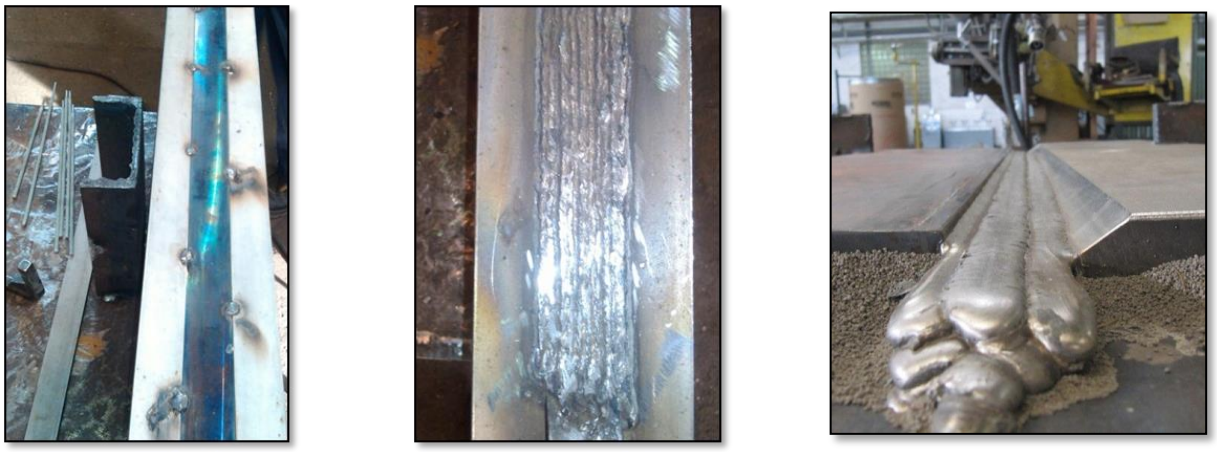


Figure 66. The mock-up manufacturing

This cushion has two layers of material. The thickness of the first layer is 3 ± 1 mm and was welded with EA-395/9 Ø 4 mm covered stick electrodes. After the first layer was ground to meet the thickness criteria, the second layer was welded using EA-400/10T Ø4 mm covered stick electrodes (the maximal temperature at the time of welding was 100°C on the same conditions as at the 1st layer). The total thickness of the layers have to be 9±1 mm (Figure 68). The welding parameters are detailed in Table 13.

	Wt %							
	C (%)	Si (%)	Mn (%)	Cr (%)	Ni (%)	Mo (%)	Co (%)	Ti (%)
EA-395/9	0.08	0.35	1.2	13.5	23	4.5	0.08	-
EA-400/10T	0.07	0.5	1.5	17	9.5	2	-	-
316L	0.015	0.4	1.75	18.5	12	2.75	-	-
X6CrNiTi18-10	0.08	0.08	1.5	17	9	-	-	0.4

Table 12: Chemical composition of stainless steels

	Welding Process	Current (A)	Voltage (V)	Type of Current and Polarity	Heat Input (kJ/mm)
1st layer	SMAW	120-130	24-25	DCEP	~0,61
2nd layer	SMAW	130-140	25-26	DCEP	~0,69

Table 13: Welding parameters of the cladding

The welding was performed in two steps and without pre-heating or heat treatment. Originally the root weld was welded manually from the root side by GTAW method using Sv- 04H19N11M3 Ø 1,6 mm electrodes in flat position.

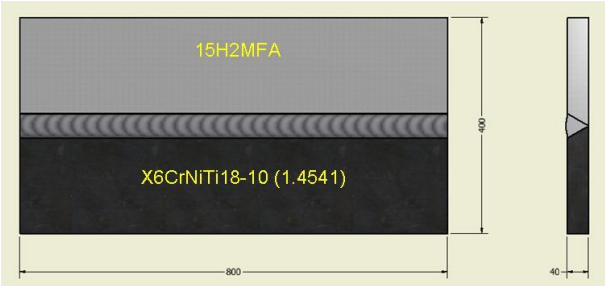


Figure 67. Dimensions of the specimen

Unfortunately, this type of electrode is no longer commercially available, so a slightly different type of electrode is used, namely Lincoln TIG 316L. As can be seen in Table 12, its chemical composition is almost the same as Sv-04H19N11M3. The filling weld and the capping were welded by SAW method using Sv-04H19N11M3 Ø 3.2 mm electrodes and OF-6 flux in horizontal position (Figure 68).

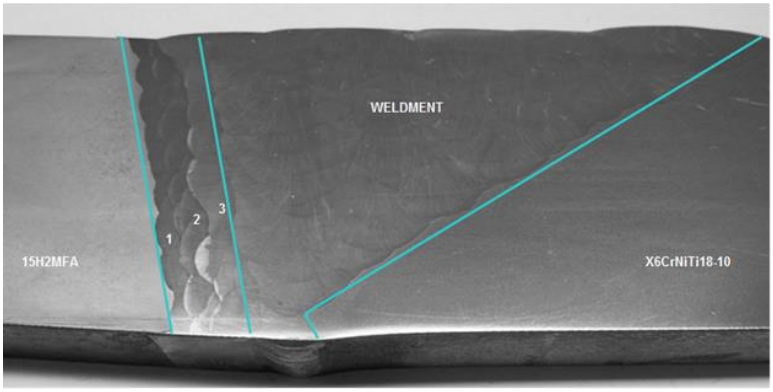


Figure 68. Macrostructures of butt-welded joint

However, this type of electrode and flux is also outdated and therefore no longer available. The substitution was LNS 316L with P240 flux. The chemical composition of the original and the substitute material has an almost perfect match and the P240 is also a highly basic fluoride agglomerated flux, with three bases, just like the original OF-6. The welding parameters are detailed in Table 14.

	Welding Process	Current (A)	Voltage (V)	Type of Current and Polarity	Heat Input (kJ/mm)
Root weld	GTAW	50-60	12-13	DCEN	~0,1
Filling weld	SAW	320-330	28-29	DCEP	~1,16

Table 14: Welding parameters

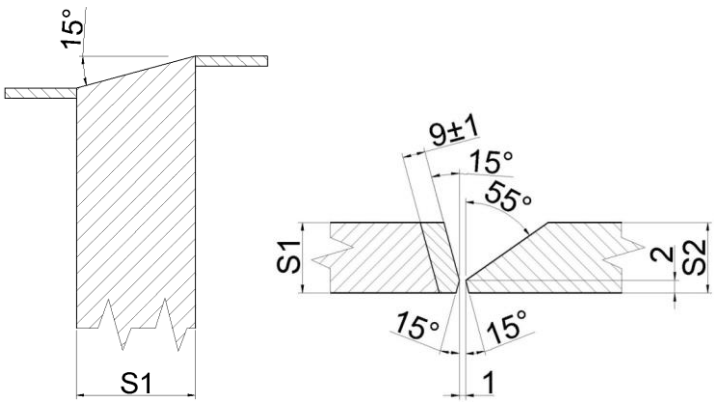


Figure 69: Technical drawing of the simulated MULTIMETAL DMW MU

The welding of DMW mock-up is simulated using 2D thermal finite element model. Work tasks:

- Simulate the cladding process
- Simulate the butt-weld process

The FEM pre-processing, calculations and post-processing have been carried out by MSC.Marc software. The associated FE model can be shown in Figure 70.

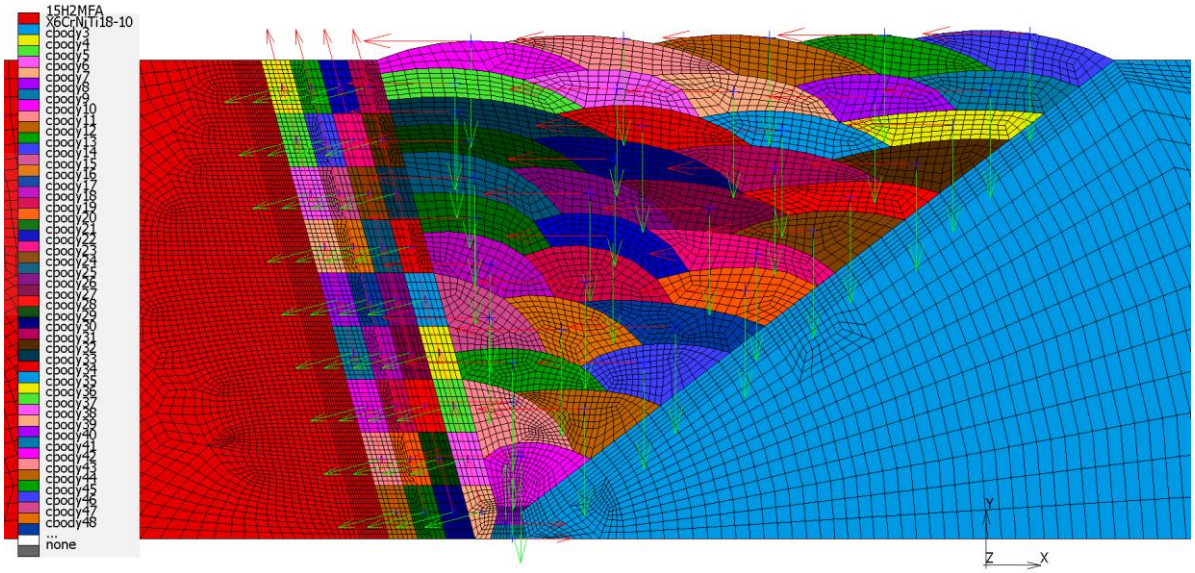


Figure 70. FE mesh for DMW

The cladding layer is divided into 9 beads along with the plate thickness (40 mm). The number of cladding layers was 4. In case of the welding process, the total numbers of simulated passes are 39 with left-to-right order according to the welding procedure. Between welding of the layers ~5min cooling time is considered because the interpass cooling temperatures are also important factors. Interpass temperature was 250°C in the first cladding layer and it was under 100°C in the other cladding layers and all layers of the butt-weld. For DMW simulation, the FE models are created by 4-noded linear thermal elements, element number is 12645, the node number is 12854. Due to anticipated high temperature gradients near the heat source, a relatively fine mesh is used there. The average value of the mesh size was 0.5 mm. Element sizes increase progressively with distance from the heat affected and mixture zone. The bead sizes and the heat source parameters are derived according to literature which has been presented before. All free surfaces of the FE model are given a convective heat loss with a heat transfer coefficient, h=25W/mK and a radiation heat loss using an emissivity coefficient, ε=0.8. The temperature-dependent material properties of analysed austenitic steels were generated with JMatPro software based on its chemical composition.

Unfortunately, during the welding process the MU was slightly distorted, which means that the original angle of the bevel changed from 55° to 53°. Therefore, for the analysis of the grain orientation two different geometries were used during the calculation. The first model is based on the originally designed geometry; the second considers the mechanical effect of the welding by contracting the welded region according to the preliminary coupled thermomechanical calculation. The effect of the slag was also analysed with a temperature-dependent heat transfer coefficient. Its effect represents the third model variation. Ferrite side phase transformation and grain evolution are not analysed in the current model.

Well known variations may arise during the manual welding process, the effect of changes of the pass order from to original welding procedure has not been analysed in the present simulation. The passing order of the butt-welding is also simulated using the original welding procedure.

Figure 71. Distortion after welding

The calculated grain orientation is shown also on the associated FE mesh, but all simulation results were visualized on the equidistant grid with 0.25mm resolution. The calculation of the grain size and grain aspect ratio distribution was analysed 1x1mm^2 areas which are included the HAZ and mixture regions.

Figure 72. shows the result of the first VVER model by using the modified geometry and considering the preheating of the ferritic steel before welding the first buttering layer. It can be observed that the horizontally oriented grains in the buttering layers now are oriented toward the centre line of the weld as solidification develops.

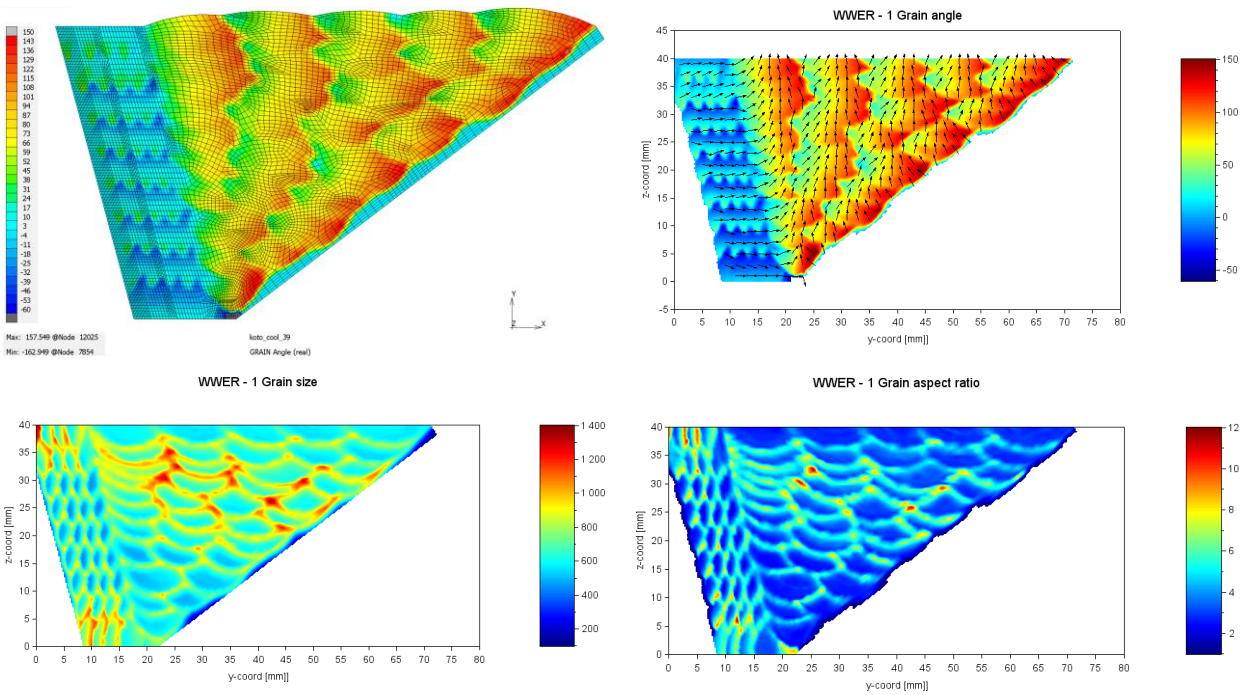


Figure 72. The result of WWER – 1 model

Figure 72 also shows the estimated grain size [µm] and aspect ratio of grains. It can be shown that the grains with higher equivalent diameter are locatedat the start and end point of the buttering layer, and in case of the buttweld after the 15th pass highly elongated grains can be found, with high equivalent diameter according to the simulation. The associated histogram is presented in Figure 73.

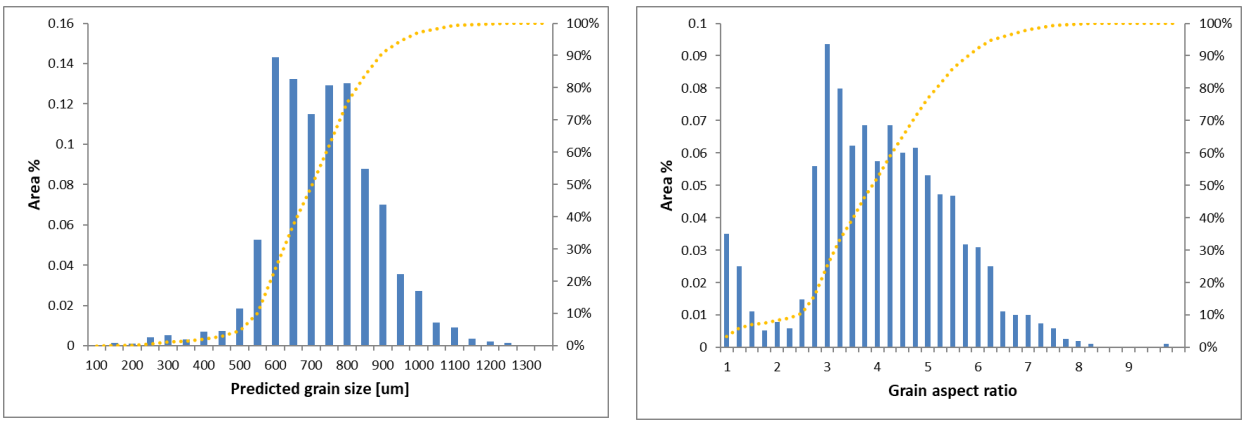


Figure 73: The predicted grain size and aspect ratio from the WWER - 1 model

Figure 74 and Figure 75 represent the results of the analysed model variations. In these cases, the originally planned geometry was used during the calculation, and we did not consider the effect of the preheating of the ferritic part. The difference between the two models is the applied heat transfer coefficient on the outer surfaces of the weld beads which consider the effect of the slag during the welding process. In case of the VVER-3 model 20% lower heat transfer coefficient was used to the ambient than the case of VVER-2 model.

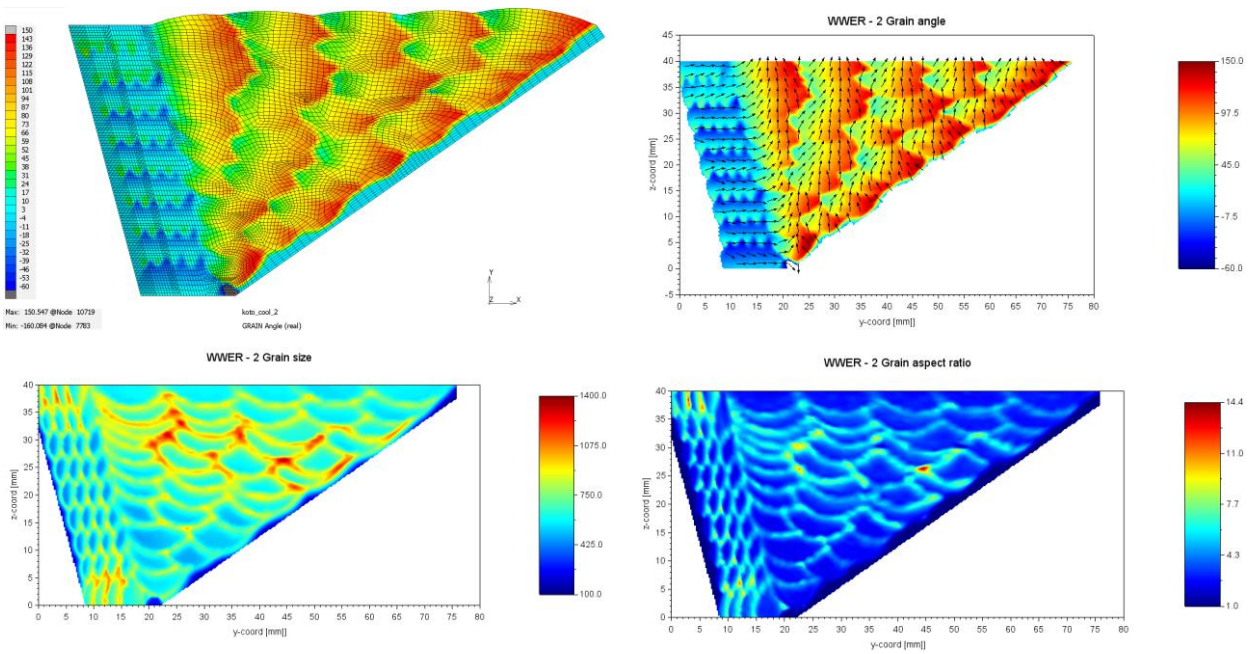


Figure 74. The result of VVER – 2 model

The results show that as the heat transfer coefficient decreases, the grain size and grain aspect ratio change significantly in the welded region, and the orientation of the grains are also changed significantly.

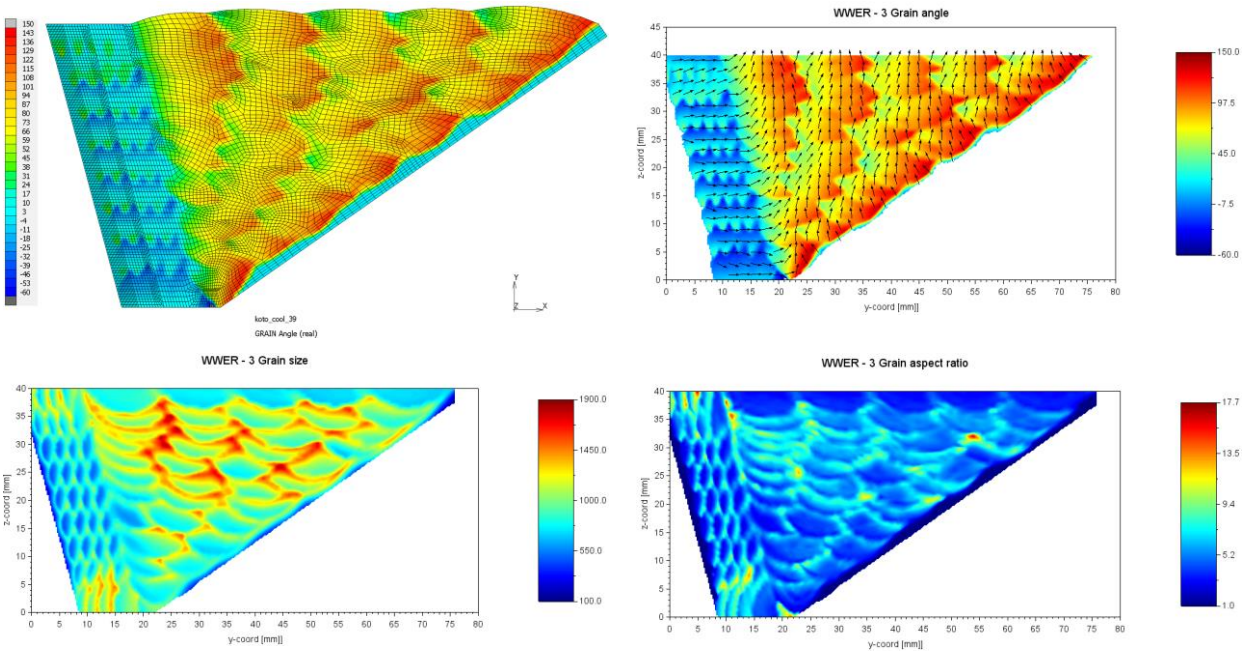


Figure 75. The result of VVER – 3 model

Figure 76 shows the associated histograms that the average predicted grain diameter for the VVER-3 model is approximately 30% higher than for the original VVER-2 model.

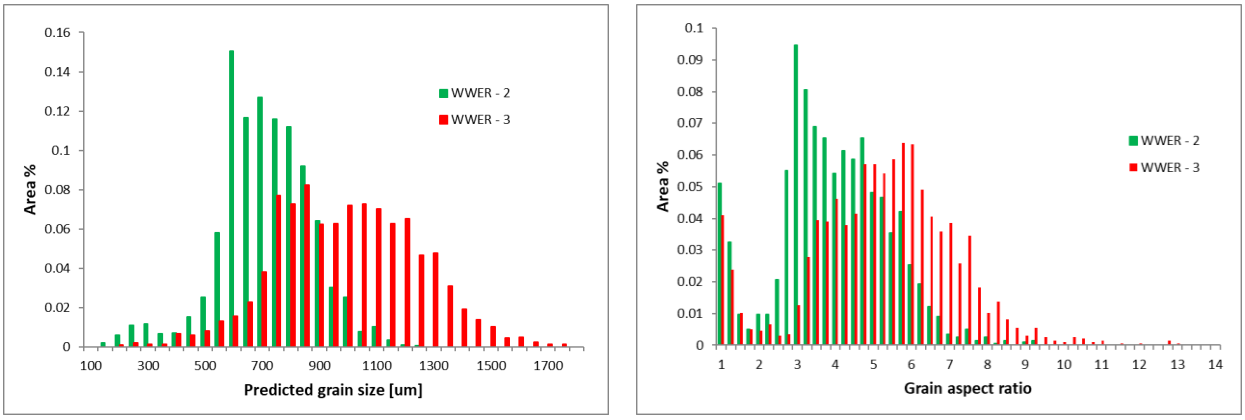


Figure 76: The predicted grain size and aspect ratio from the VVER – 2 and 3 model

Modifying the heat transfer coefficient on the bead's surfaces, the shape of the molten pool is also changed. Comparing the calculated grain orientations, the deeper regions are clearly visible.

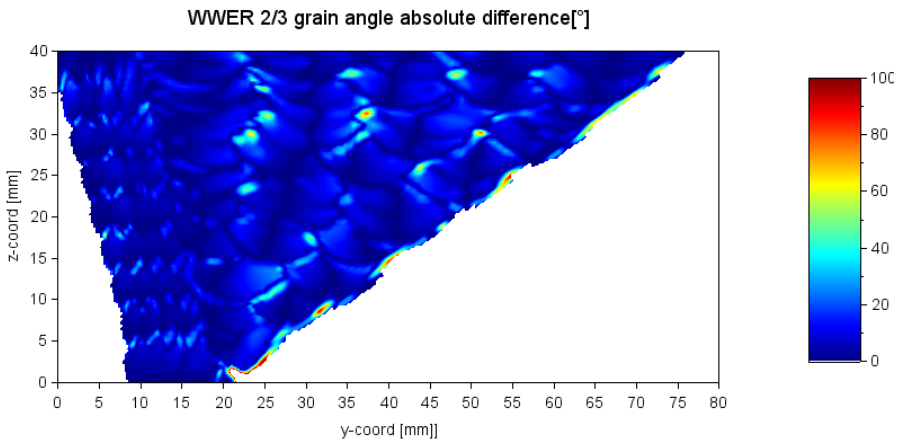


Figure 77: Grain orientation differences

In order to decide which effect is dominant in case of VVER welding, further EBSD analysis and simulations are needed to compare the calculated results with the real macrostructure.

3.6 Conclusions of FE analysis

In this chapter, the finite element method has been utilized exclusively to simulate the welding processes. We have systematically analysed the uncertain parameters of the welding process and their effect on the grain formation.

During the thermal simulations, the following parameters were systematically investigated in case of WP2 sample:

- Tilting of the electrode →modifies the flow direction, the shape id of the molten region and the direct power (-5° and +8°) [Model 5 and 6, see in Figure 45 and Figure 46]
- Increased welding power (10% increase the line energy) → larger molten region. [Model 7, see in Figure 47]
- Modification of bead geometry based on the microstructure of the original geometry, with weaving technique:
  - With medium weaves [Model 1, see in Figure 50]
  - With small weaves → larger heat input and deeper penetration of the heat source [Model 2, see in Figure 51]

- With large weaves → lower heat input and shallow penetration [Model 3, see in Figure 52].

In case of Model 4, 5, 6 and 7 simplified weld shape was generated for the simulation, while in case of model 1, 2 and 3 the bead shapes were created based on the microstructure of the original geometry. Table 15. shows the average differences and the standard deviation between the original and the modified models. The numbers of model were already introduced earlier therefore it will not be listed here. The grain orientation matrix was calculated on the same FE mesh and post processed on the same grid with 0.25x0.25 mm scale, but for the accurate ultrasound modelling a 2x2 mm scale is sufficient, therefore the calculated results were transformed to 1x1mm and 2x2mm scale using area-based averaging. It can be seen that the largest difference in the grain orientation was in the case of the model where electrode tilting was used in the positive side, however in case of negative tilting smaller differences could be detected. The increased welding power also did not modify the grain orientations in large scale.

The usage of weaving technique and newly built FE model showed larger differences than the previously introduced welding parameters.

Model	1x1 mm		2x2 mm	
	Average difference [°]	Standard deviation [°]	Average difference [°]	Standard deviation [°]
5 vs. 4	6.18	8.53	5.4	6.94
6 vs. 4	1.72	3.04	1.48	2.5
7 vs. 4	2.28	3.25	1.8	2.11
3D vs. 4	2.08	2.93	1.77	2.48
1 vs. 4	12.87	11.02	11.8	9.6
2 vs. 4	10.42	10.86	9.21	9.5
3 vs. 4	11.95	12.43	10.97	13.42

Table 15: Effect of the welding parameters

The method which is presented before is easily applicable to modelling grain formation under different welding processes (SMAW, GTAW and SAW), but it is also extendable for other processes like laser welding or casting simulation. Solvers and necessary constitutive equations are implemented in commercial FE software. Thus, only the problem formulation needs to be considered when creating models that take into account different welding techniques, parameters, and the calculated results should also be visualized. However, FEM is very computer intensive, and parallelization of the codes gives only moderate improvements. As larger 3D models with more weld rows are studied, more memory and much more time are needed. This raises the question of the extent to which FEM can be utilized to model grain orientation and microstructures, because the FE uses continuous field on macro-scale, but the presented macro-scale modelling technique provides reasonable predictions of solidification conditions, thereby giving a sufficiently realistic grain orientation.

4 Kinetic Monte-Carlo simulation

In this section before the introduction of the simulation results and the usage of the developed code a brief literature review is shown, which contains the most famous simulation modes of grain growth. It is separated into two groups, namely the thermal and the microstructure modelling methods. From these models, the Monte Carlo method was chosen, which is implemented in the SPPARKS open-source code. SPPARKS is a fully parallel Monte-Carlo code that can be used for on-lattice and off lattice applications.

Among the several SPPARKS examples one can find welding preconditioned ones too. SPPARKS is able to simulate welding. It is well-tested and has noticeably confident results. The only drawback is that it cannot solve multilayer weldments, therefore a refinement of the code was necessary to achieve the desired results. The improvements, the validation of the code and also the results of the simulation carried out with the developed code is introduced hereafter.

4.1 Modelling methods in the literature

Almost all materials go through forming and joining procedures. Both methods have a great influence on the final structure and properties of the material. When welding is applied, the microstructure of the developed material gets destroyed and transformed to an as-cast state. The heat affected zone also experiences notable changes which result in the degradation of material properties. Therefore, the modelling of the microstructure of these types of welded joints is an important and extensively researched topic. With modelling several insights can be achieved which would not be possible with traditional experimental methods. In case of welding, several material phenomena can occur for example grain growth, recrystallisation, and multiple phase transformations. Therefore, in the following two main types of modelling methods are introduced, the thermal modelling methods and the microstructure modelling methods.

4.1.1 Thermal modelling methods

The thermal models can be separated into two large groups, the analytical and the numerical methods. In case of analytical models one or more functions are taken into consideration and these functions are valid over the whole domain where the precondition is that only one closed solution exists. In contrast to this, with numerical models more complex problems can be solved too, as these methods discretise them into smaller parts where the aim is to converge towards a solution.

One of the first analytical models is from Rosenthal who examined an electric arc weld and modelled it with a moving point source while assumed quasi-steady state [21]. He assumed every material property as constant, however even with this assumption the results in case of low temperature were in good agreement with the experimental results. Pavelic improved the model and characterized the heat source with a Gaussian distribution [26]. These models have been improved throughout the years; however, they became too complex to be solved with analytical methods. Nowadays analytical models are still pursued for laser heat sources, an example of this is the model proposed by Ducharme et al [27].

As already mentioned, the numerical thermal models are used for more complex problems where an analytical solution cannot be achieved. For these models the geometry of the weld pool is also included, one of the first in these types of models was made by Okada et al [28]. This model gave good results for lower temperatures however in case of peak temperatures it fell behind the experimental results. Wahab et al [29][30] proposed a model where the entire weld pool cavity was used, this was designed for gas-

metal-arc welding. The heat source is built of three parts, which represent the liquified filler metal and the actual arc energy. The arc energy is separated into an ellipsoid surface source and a spherical source. For high heat input values, the model gives satisfactory results, however on lower values the results can differ almost 50% from the measured values. For deep penetration welding several models were developed for example the model of Sudnik et al [31], who proposed the usage of sub-models. Another example is the work of Babu et al [32], which reduced the problem to one dimension with the assuming that the structure is built up from thin sheets, it can be seen in Figure 78. The solution contains a changeable efficiency coefficient, which can be set for every thin sheet. The sheets at the top of the keyhole have higher efficiency than the ones at the bottom. This model is only valid for thin structures because in case of thicker workpieces the heat conduction in the vertical direction must be considered.

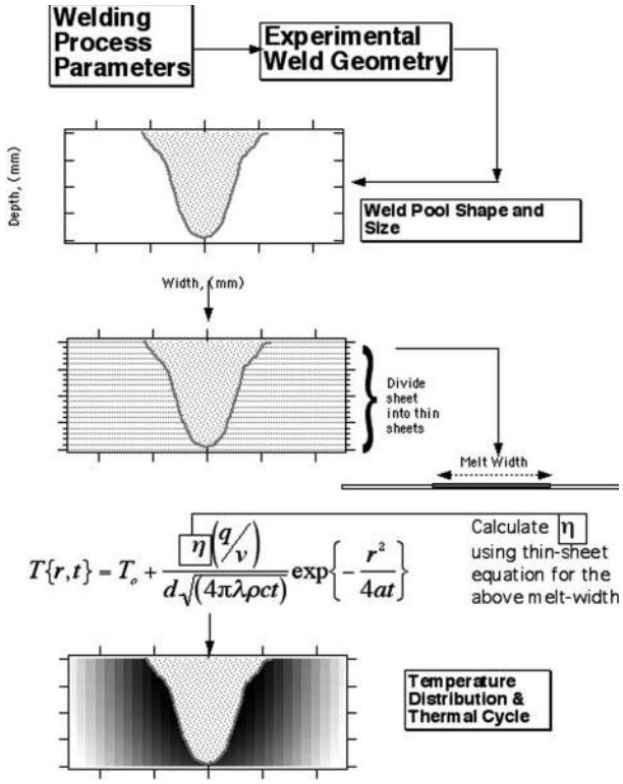


Figure 78: methodology for calculating temperature distribution with the Babu model [32]

Mahrle et al. [33] used mathematical models which describe the fluid flow in the weld pool. With this method they wished to generate accurate temperature fields. In case of high temperatures, the received results were in good agreement with the experimental data, however on low temperatures larger deviation was observed. Fuhrich et al [34] also studied the problem of the keyhole shaped weld pool. In his research he assumed fixed geometry for the inner wall of the keyhole, and its temperature was maintained at the boiling temperature with the help of the boundary conditions. Therefore, the melting temperature isotherm determined the welding pool geometry. The results showed the formation of the flow patterns in case of the keyhole weld pool, wherewith the depth of this pattern changed due to the surface tension. This method simplifies the calculations but can lead to the wrong weld pool shape. It is very rare to simulate the weld pool as free moving surfaces as it needs enormous computational efforts. However, some examples can be found for this like the model proposed by K_i et al [35]. They used Gaussian distribution for the thermal energy input where the beam radius was constant in the workpiece. One of the most well-

known thermal models for welding is the Goldak double ellipsoid method. The model is introduced in detail in chapter 3.2.

4.1.2 Microstructure modelling methods

In case of microstructure modelling two types of methods can be separated, implicit and explicit modelling. In case of implicit modelling the microstructure morphology is not explicitly recovered, the results usually give information about the phase fraction and the average grain size. Many models of this type have introduced simplifications, for example on the morphological features. Several methods can be listed in this group which even nowadays give valuable result about the mechanisms of microstructure evolution during welding. The explicit modelling method inherently yields the features of the microstructure. These methods demand more computational effort. Most of these models can be used to create 2- and 3-dimensional solutions.

Implicit models: One of the basic models of the implicit methods is the empirical model. Before the presence of numerical analysis, the created models were based on experimental results where the temperature of heating and cooling was controlled. A well-known model determines the microstructure of the weldment with the Continuous Cooling Time (CCT) diagrams. Several continuous cooling experiments must be carried out and examined to make cooling trajectories. These go through regions of phases and from this the final microstructure can be concluded. These models mainly use the Scheil additivity to determine the final phase fractions which depend on the time spent in any given temperature range [36]. One of the problems with this method was mentioned by Stadtaus et al [37]. The problem is that the model is not only composition dependent, but the peak temperature dependence should also be included. With the help of neural network models CCT diagram interpolation was done based on measured CCT diagrams. The results proved the statement of Stadtaus [38]. Another empirical model is the Johnson-Mehl-Avrami-Kolmogorov (JMAK) equation. The equation describes the relationship between the isothermal phase transformations and the phase fraction which is a function of time. The model was also adapted to the non-isothermal case, the Scheil additivity was applied in this case too.

The second model group of the implicit method is the diffusion-controlled transformation models. In several phase transformations the diffusion mechanism is slower than other mechanisms, therefore it becomes the rate-controlling process. The first well-known model based on this theory was made by Zener [39]. These models were not specifically made for welding; however, they can be used universally. An important subclass of them considers interface mobility as kinetics determining mechanism, an example of these models being the work of Svoboda et al [40] and Sietsma and van der Zwaag [41]. Svoboda et al developed an analytical expression for the interference position using multiple diffusion fields, influences of lattice strains and finite interface mobility. The model was usable for one-dimension problems. The model of Zwag was more general. It was able to capture mixed mode transformations. The usability of the model is limited because it does not consider interfacial energy. Beside the general solutions there are some models which were developed for the examination of weldings, these usually give results for given alloy types. For example, Katzarov et al [42] created a model for the $\beta \rightarrow \alpha$ transformation in the Ti6Al4V alloy. Here the morphology of the α-phase is the function of the cooling rates. If the cooling process is slow then plate-like α-phased regions will be created, otherwise fine needles of α are produced. Because of the low mobility of the vanadium the diffusion of this component governs the mechanism. The model was implemented in the finite element method where the Eulerian settings were used. Volume elements were used for the handling of the mobile phase. The model can be used well between 1025-1275 K temperature range and 5 Ks^{-1} to 50 Ks^{-1} cooling rate.

Hemmer and Grong [43] examined welded dual-phase steels; their model included consideration of state variables. This approach includes path dependency which is a crucial part of welding simulation because of the movement of the weld pool, therefore the heating and cooling of the workpiece have significant effects on the developed microstructure. The dissolution of austenite due to heating, the formation of δ-ferrite and their decomposition back to austenite are considered in the model. The model was further developed by Hemmer who made more assumptions to simplify the model. These assumptions include the occurring of phase transformations only in series, therefore every austenite must deform into δ-ferrite before the grain growth can start and that the main component which governs the dissolution is the diffusion of nitrogen. This model gives good results for the microstructure of duplex steels, however some of the previously mentioned assumptions limit the usability of the model, for example the occurrence of the reaction being only in series is not fully right, also the usage of a Rosenthal heat source results in inaccurate data for high temperatures. The model was further modified by Myhr et al [44] who were examining the microstructure of Al-Mg-Si alloys. They assumed that the precipitates were spherically shaped which made the problem into a one-dimensional diffusion problem. These ideas were implemented into other research works too, where two aluminium alloys were examined. They assumed that the solidification interface was in local equilibrium, with this the diffusion flux equations could be solved. The following assumptions were also made constant cooling rates, initiation of eutectic phase between the dendrites, and α-phase dendritic growth as a diffusion problem at half the distance of the adjacent dendrites.

Explicit models: As already mentioned above the explicit models reproduce the microstructure morphology too. In the following, the most well-known methods related to this group are introduced briefly. The first model is called the Monte Carlo (MC) method, which can be used for the simulation of microstructure evolution. Fixed grid points with their own orientation are used. This way the simulation uses a large finite number of orientations, which have small effect on the grain growth (as in reality their number of orientations is infinite). Sista And DebRoy used 48 orientations, therefore each grid point was assigned a random orientation number between 1 and 48, this solution gave good results [45]. To simulate grain growth and recrystallization, lattice points are given a probability of changing orientation in each time step. This can be calculated with the following equation:

$$\begin{aligned} p &= 1 && \Delta E \leq 0 \\ p &= \exp\left(-\frac{\Delta E}{kT}\right) && \Delta E > 0 \end{aligned} \qquad (15)$$

where ΔE represents the change of energy because of the change in orientation, T is the temperature, k is the Boltzman constant, p is the probability of orientation change. With a regular orthogonal grid in 3D a point will be assigned with 26 first-, second- and third- nearest neighbours. The grain boundary energy in this form can only be used for grain growth and is treated as isotropic. This version of the MC model also has no coupling to diffusion equations. Yang et al [46] and Shi et al [47] proposed MC models which are able to follow the changes in the heat affected zone. One of the greatest disadvantages of the MC models is that calculation is not coupled inherently to the real-time. There are attempts to make a connection with the usage of empirical relations to grain growth laws. With this the MC dimensionless time (t_{MC}) can be written in the following way:

$$(t_{MC})^{n^G n_1^G} = \frac{L_0^{n^G}}{(K_1^G \lambda)^{n^G}} + \frac{K^G}{(K_1^G \lambda)^{n^G}} \sum \left[\Delta t_i \exp\left(-\frac{Q}{RT}\right)\right] \qquad (16)$$

where L_o is the original grain size, n^G is the growth exponent, K^G is a material constant, t real-time, Q is the activation energy for grain growth, K_1^G and n_1^G are model constants, λ is the grid spacing. Here the main assumption is that the material follows the classical grain growth law, however this equation is valid for isothermal conditions, therefore it is questionable if it can be used for the welding process too.

The second main group of explicit models is the Cellular Automata Method. Micro scale simulations can be created with the usage of this solution. Individual lattice points are the main components of the simulation, and they are described by a state variable. The value of this variable is chosen from a discrete set of states and is evaluated with probabilistic transformation rules. Kumar et al [48] modelled the nucleation and growth of ferrite, where the nucleation rate had a Gaussian distribution over the nucleation temperature range, and it was created at pre-given nodes in the mesh. The growth of ferrite was based on a diffusion-controlled rate. In this case a single grain was examined. It was further developed by Zhang et al. They extended the grid to 200x200 grid, which equals 125 µm. In this case the grid points were switched in a probabilistic way when nucleation was present. The biggest shortcomings of these models are that the interfacial energy is not taken into consideration and that complete homogenisation is assumed, which means that the model can only be used for very low cooling rates. There are also more developed models where the coupling of cellular automates and finite element models was achieved. Chen et al. [58] developed this type of model where they analysed the temperature profile in the welding region with finite element method and for the grain growth the cellular automaton was used, this solution can simulate a large number of grains, however the changes in the heat affected zones are not considered.

The third group covers phase-field methods. This model assumes a state variable which is defined over the whole computational domain. It can depend on conserved variables (compositions) or non-conserved variables (lattice structure, orientation). A diffuse border exists where the state variable can change between 0 and 1, which represents the α and β phases. Theoretical state variables can be used which can include the changes of chemical, structural, and crystallographic fields. These variables are called phase-field variables. The model can be used in many microstructures modelling processes like grain growth simulations, martensitic transformations in ceramics.

4.2 Introduction of built-in simulation methods

The simulation results shown in this section are achieved by using an open-source package developed by Sandia National Laboratories, namely Steve Plimpton, Aidan Thompson and Alex Slepoy. This software package called SPPARKS - an acronym for Stochastic Parallel PARticle Kinetic Simulator.

SPPARKS is a fully parallel Monte-Carlo code that can be used for on-lattice and off lattice applications. It features Kinetic MC (KMC), Rejection Kinetic MC (RKMC), Metropolis MC (MMC), and several KMC solvers.

In a generic sense, the solvers catalogue a list of "events", each with an associated probability, choose a single event to perform, and advance time by the correct amount. Events may be chosen individually at random, or by sweeping over sites in a more ordered fashion [49][50][51].

The MC algorithms are greatly beneficial not only in calculating solitaire games or at CERN's LHC, but at predicting grain growth during solidification. Kinetic means before MC, that it is time (at least simulation step time) dependent.

Among the several SPPARKS examples one can find welding preconditioned one too. SPPARKS is capable of simulating welding on the before described basics [52]. It is well-tested and has noticeably confident results. The only drawback is that it cannot solve multilayer weldments by default as in our case, therefore the C++

code was modified, as the code is designed to be modifiable and can be extended with new functionality. This extension was done at BZN.

For the extension of the code two main parts of the SPPARKS were used, the potts/weld and the potts/additive application styles.

4.2.1 Welding simulation in SPPARKS

The model includes the simulation of melting, solidification, and solid-state microstructural evolution of material in the fusion and heat-affected zones of a weld. The model does not include thermal behaviour, it utilizes user input parameters to specify the weld pool and heat-affect zone properties. Material re-solidifies along the trailing surface of the weld pool and undergoes solid-state evolution in regions immediately surrounding the pool. The moving weld pool leads to the unique grain microstructures which can be seen in the fusion zone (FZ) of welds, characterized by curved elongated grains whose morphologies mainly depend on the processing conditions. Not only the resultant grain size but also its shape has a great influence on the mechanical and other properties of the material, therefore it is essential to have more knowledge about the microstructure of a weld, which highly depends on the welding parameters, for example the type of heat source (laser, flame, electric arc).

4.2.1.1 Potts/weld application

A modified Potts Monte Carlo model is used for the simulation of the grain evolution. For weld simulation the process temperature profile is necessary which depends on the position and the time. In this model these parameters can be reached with the translation of an idealized weld pool and temperature gradient through the simulation region. Melting and solidification is achieved with the exceeding or dropping below the melting temperature. The grain growth simulation includes a temperature dependent grain boundary mobility beside the normal curvature-driven grain growth model. Heat transfer, dendritic growth and fluid motion in the weld pool are not simulated in the model, however they are implicitly considered with the temperature profile specification.

The 3D microstructure digitalization is done on a regular lattice of spatial sites, each site has a unique identifier or “spin” which indicated which grain a given sites belong to. The sites which are continuous and belong to the same spin form a grain. The driving force of grain growth and evolution is the grain boundary energy. The grain growth is calculated with the minimalization of total grain boundary energy. The equation of it is the following [53]:

$$E = \frac{J}{2}\sum_{i}^{N}\sum_{j}^{n}\left(1-\delta_{ij}\right) \qquad (17)$$

where J is the bond energy, i is the index of each site (1 to N), j (1 to n) is the neighbour site of i, δ_{ij} is the number of neighbours of the site i . The grain boundary is received when dissimilar spins are neighbour sites. SPPARKS is using the rejection Kinetic Monte Carlo algorithm for potts/weld, with this the curvature driven grain growth can be simulated. In this type of simulation, the algorithm selects a site and tries to change its spin to a randomly selected spin which is resulted from the set of dissimilar neighbours. The probability of this event can be calculated with the following equation [53]:

$$P = \begin{cases} 1 & \Delta E \leq 0 \\ e^{-\frac{\Delta E}{kT_s}} & \Delta E > 0 \end{cases} \qquad (18)$$

From this equation, it can be seen that the change of spin occurs when $\Delta E \leq 0$, if $\Delta E > 0$ then it will appear at the probabilistic rate given in the previous equation. This solution gives the microstructure of the base metal, it is performed until the desired average base metal grain size is achieved. The temperature dependency of the grain growth can be simulated with a temperate dependent grain boundary mobility term $M(T)$, and it can be written in the following way [53]:

$$P = \begin{cases} M(T) & \Delta E \leq 0 \\ M(T)e^{-\frac{\Delta E}{kT_s}} & \Delta E > 0 \end{cases} \qquad (19)$$

Here $M(T)$ changes between 0 and 1, for room temperature the value of it is 0, therefore grain growth does not occur. This mobility equation can be written in the following way [53]:

$$M(T) = M_0 \exp\left(-\frac{Q}{RT}\right) \qquad (20)$$

where M_0 is the Arrhenius pre-factor, Q is the activation energy for grain boundary motion, R is the gas constant, T is the local temperature caused by proximity to the solidification boundary.

For the changes in spin a similar algorithm is used in the following way:

- Generation of random number R (between 0 and 1)
- Spin change can occur if:
- $\Delta E \leq 0$ and $R \leq M$
- $\Delta E > 0$ and $R \leq P$
- If these do not occur, the original spin remains.

Figure 79 shows the schematic picture of the moving weld pool. It can be seen that when the sites enter the weld pool melt ($T \geq T_m$); grain growth is not allowed therefore there are no grains in the melt and a random spin is assigned to the given site. Every site where the temperature is equal or bigger than the melting temperature belongs to the welding pool. When they exit from the pool ($T < T_m$) they solidify and receive a spin which was randomly chosen from the spins of neighbour sites.

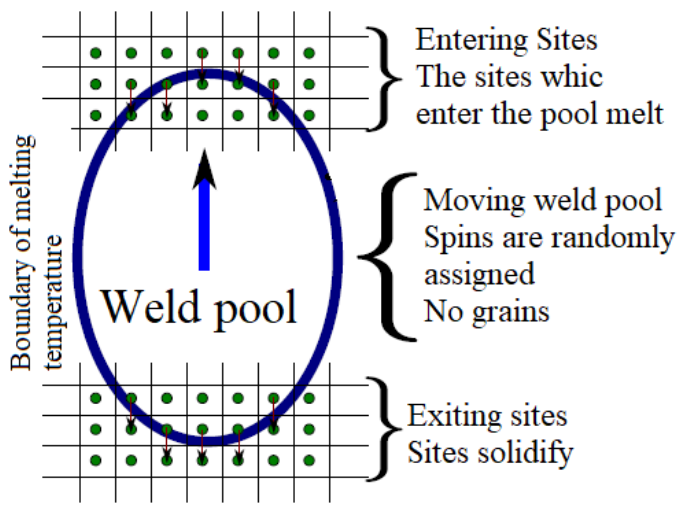


Figure 79: Sites entering into and exiting from weld pool [52]

The unit of the weld speed (v_p) is given in lattice sites per Monte Carlo step. It is related to the real welding speed with a c lattice constant the unit of it is microns per sites. This is calculated with the ratio of the weld pool width to the number of lattice sites of the weld pool in the simulation. The unit of the simulation time is Monte Carlo steps (MCS), it is related to the real-time with the following constant:

$$\gamma = c\frac{v_p}{V} \qquad (21)$$

where V is the real weld speed in microns per second. The unit of γ is seconds per MCS, and it gives the relation between the simulation (t) and the real-time (t_r) in the following way:

$$t_r = \gamma t \qquad (22)$$

For this model the weld microstructure is most significantly affected by the initial base metal grain size, the weld speed and the heat affected zone. These parameters have a large effect on the size and shape of the grains. With the rising of the HAZ width or the decreasing of the weld speed larger grains are formed.

Connecting the previously introduced algorithm to the potts/weld application style it can be said that it is an on-lattice application, where the simulation of grain growth is achieved. Two sheets of material can be welded together, where the thickness of them is identical and the two sheets are touching. This defines the joint to be welded. The welding process is simulated with the moving of a weld pool which is translated along the given joint. The speed of it is defined with the *velocity* parameter. Two types of geometry can be defined in this case, an elliptical pool (*weld_shape_ellipse*) or a teardrop shaped pool (*weld_shape_teardrop*). With these the pool size and shape at the top surface of the plates can be given. One of the most important difference between the two application styles is that in case of the potts/additive 3D simulation can be achieved, therefore more layers can be defined, however in case of the potts/weld application style it is assumed that the weld fully penetrates the thickness of the plates, and at the bottom part of the plates the pool size is determined by an α parameter which varies between 1 and 0. Curvature of the pool in the plate thickness direction is controlled by the β parameter which also varies between 0 and 1. The effect of variation of α and β is shown in Figure 80.

If the spin at a lattice site is inside the volume of the melt pool, then the model randomizes the spin. With this the simulation of melting and re-solidification can be achieved. When the end of the melt pool is reached, a rejection kinetic Monte Carlo event is performed at the site, and the spin is flipped to the value of one of its neighbours. In the heat affected zone the mobility of these sites is decreasing linearly with increasing distance from the melt pool surface, where the maximal mobility 1 is given for the melt pool boundary and 0 is given for the outer boundary of HAZ. Outside of the melt pool and heat affected zone, grain boundary mobility is set to 0, and grain evolution does not occur.

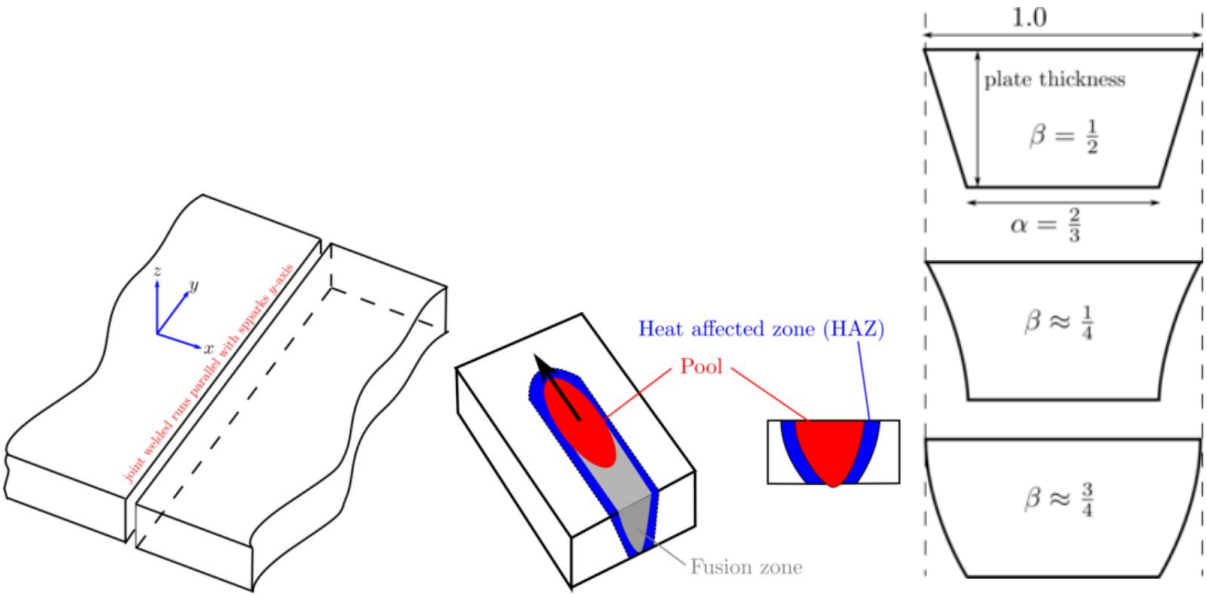


Figure 80: Built-up of potts/weld application style and effects of parameters [51]

In case of the potts/weld application, the values defined in Table 16 can be applied.

Name	Description
nspins	number of possible spins
yp	initial melt pool position along y-axis
alpha	controls relative size of melt pool shape at bottom compared to top
beta	Bezier control point parameter that defines curvature of melt pool shape through thickness
velocity	velocity of melt pool motion (lattice sites per Monte Carlo step)
haz	width of the heat affected zone (haz) surrounding the melt pool

Table 16: Syntax of potts/weld command [51]

Here it must be taken into account that the value of the thickness of the heat affected zone is given (haz). To separate the weld pool and the HAZ, a notation describing the placement of points compared to the weld pool has to be included. $\mathbb{D}$ is the bounded spatial domain of the entire simulation. $\mathbb{L}$ represents the set of points which are inside the weld pool. Those points which are part of $\mathbb{D}$, not included in $\mathbb{L}$ and are within a distance haz of the pool surface, belong to the domain of $\mathbb{V}$ and are part of the HAZ. The set of points belonging to the HAZ can be defined the following way:

$$HAZ := \{x : x \in \mathbb{V} \text{ and } d(x) \leq haz\} \qquad (23)$$

where $d(x)$ gives the Euclidean distance to the pool surface In case of an $x \in \mathbb{V}$ point. The grain boundary mobility can be described with the following equation, however in this model it is not a requirement as the distance of the HAZ is given [51].

$$M(x) = 1 - \frac{d(x)}{haz} \qquad (24)$$

4.2.1.2 Potts/additive application

The potts/additive command is an on-lattice application. This model uses the same Monte Carlo Potts model just like the previously introduced potts/weld application, however it contains several modifications. In this case the microstructure will only change within and immediately outside of the molten zone. With this the formation of a molten pool and the heat affected zone through a given domain can be simulated. Rasterization is achieved by specifying layer-by-layer patterns that can be combined into a comprehensive pattern that determines the translation of the molten zone over the entire simulation range. The application allows any number of paths in each layer and any number of layers in each pattern. For the characterization of the molten pool a double ellipsoid is used. Two of their axes are equal (namely the *spot_width* and the *melt_depth* parameters). Based on the position of the ellipsoid the third axis is defined by *melt_tail_length* or *cap_height*. Similar parameters can be given for the heat affected zone too (*HAZ, tail_HAZ, depth_HAZ*, and *cap_HAZ*). The placements of these eight parameters are shown in next figure. The name and the description of the usable parameters in the potts/additive command can be seen in the Table. For the Goldak's double-ellipsoid heat source model similar parameters are used. However, unlike Goldak's model, here a pre-determined molten zone and HAZ geometry is applied instead of energy flux. With this kind of solution, the specification of molten zones with widely varying geometries including long tails and/or abbreviated "caps" can be achieved without specific knowledge of the material's thermo-fluidic properties.

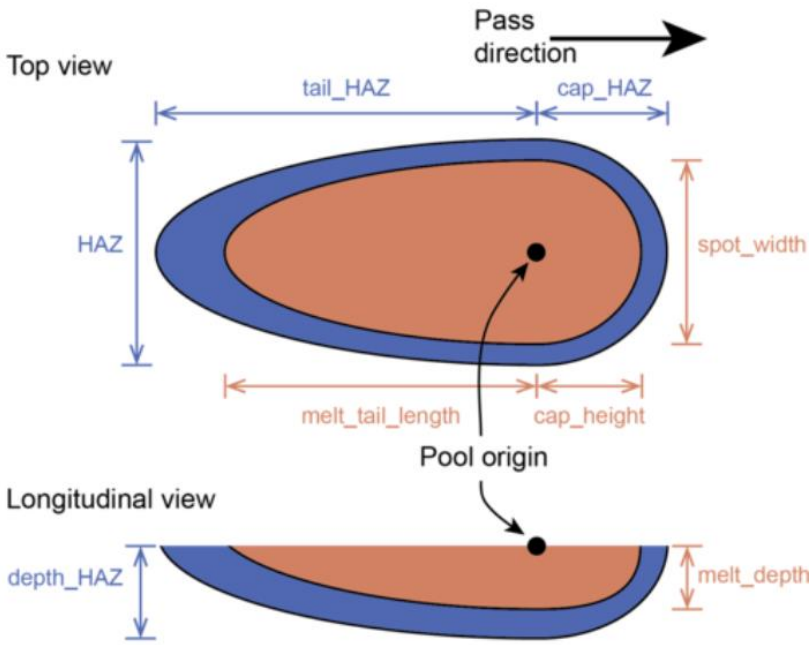


Figure 81: The placement of the potts/additive parameters [51]

Besides the geometric properties the *exp_factor* variable also must be specified. It gives the value of the coefficient in the mobility equation [51]:

$$M_0 = \exp(-exp\_factor \times x) \qquad (25)$$

here x is the shortest distance from the lattice site to the molten pool boundary. Similarly to the potts/weld application, the probability of accepting the change the spin of a selected lattice is given by the Metropolis functions which are modified with the nonuniform mobility equation.

These applications are only compatible with square and square cubic lattices require a rejection KMC (rKMC) algorithm.

Name	Description
nspins	number of possible spins
spot_width	maximum width of the melt pool
melt_tail_length	maximum length of the melt pool trailing the melt spot
melt_depth	maximum depth of the melt pool
cap_height	maximum length of the melt pool leading the melt spot
HAZ	width of the heat affected zone (HAZ) surrounding the melt pool (must be larger than width)
tail_HAZ	Length of the haz trailing the meltpool (must be larger than tail_length)
depth_HAZ	depth of the heat affected zone (HAZ) below the melt pool (must be larger than depth)
cap_HAZ	Length of haz leading the melt pool (must be larger than cap_length)
exp_factor	Coefficient that controls the rate of exponential decay of the haz mobility gradient
am pass	Specify pass parameters used to construct cartesian_layer.
am path	Specify arbitrary paths via start/end points on a layer; sequence of am paths are used to construct a am path_layer
am cartesian_layer:	A scan pattern on rectangular layer constructed from am pass and other parameters
am build	May be used for combinations of layers that comprise a pattern.

Table 17: Parameters of potts/additive applicaton [51]

4.3 Post processing SPPARKS results

Before the introduction of the modified code the software used for the post-processing of the SPPARKS results shall be introduced. SPPARKS outputs its results into dump files, which are basically text files containing the lattice information. The lattice is built up from atoms, which are aligned in a rectangular grid style. Based on atomic spins which denote the type and assigns the atoms to grains, these are also stored in dump files. A text dump file example is shown below:


```
ITEM: TIMESTEP
4    56.6538
ITEM: NUMBER OF ATOMS
2494800
ITEM: BOX BOUNDS
0 150
0 88
0 189
ITEM: ATOMS id type d1 x y z energy
1 2441293 0 0 0 0 0
2 2441293 0 1 0 0 2
3 673488 0 2 0 0 7
……
```

Table 18: SPPARKS dump file example

Since this is relatively low-level volumetric data about the welded grains, we needed to search firstly for open-source solutions in post-processing aka visualizing data. Also, it is essential to visualize the simulated data, because this is the best measure for deciding whether the simulation's result is acceptable or not. If the simulation result found to be passed on a human eye metric, then further investigations are needed at the current parameters. Below we introduce the sequence of programs that were used for processing dump files both for visualizing and data processing.

4.3.1 Ovito

The first and easiest way to check the simulated weld grain shapes is to load the raw dump file into the open source and free visualization program called OVITO. (The Pro version is not free) OVITO is used for the visualization and analysis of atomistic and particle simulation data. With the help of this software better insight can be reached into materials phenomena and physical processes. The program is freely available for all major platforms under an open-source license. It has been used in several computational simulation studies to analyse, understand, and illustrate simulation results. Unique analysis, editing and animation functions are integrated into its easy-to-use graphical user interface [54][55].

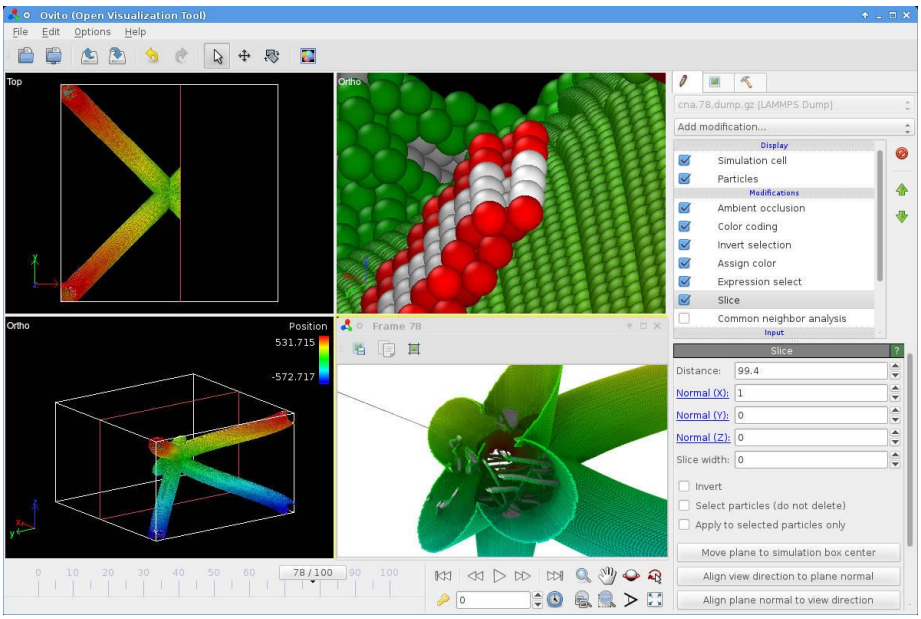


Figure 82: OVITO's interface

4.3.2 DREAM3D

DREAM.3D is completely open source and free for anyone. It can be downloaded from Github. The DREAM3D code mainly developed by US Air Force (AFRL), Navy Research Labs (NRL) contracts, but many contributors can be found on the website [56][57].

The aim of using DREAM3D is to postprocess 3D grain data and get grain statistics, but it does more: BZN also used it for manipulating the rectangular grid at the 2nd weld specimen. Also, it is worth to mention that the code can be used to build initial grain structures, although BZN used SPPARKS' own grain initialization algorithm. It is worth a note that further investigations could be extended in this direction.

DREAM3D code - thanks to the GUI - is ordered into the task pipeline, in which the user is able to select and drag the necessary data processing tasks. There are pre-built pipelines that come with the downloaded software.

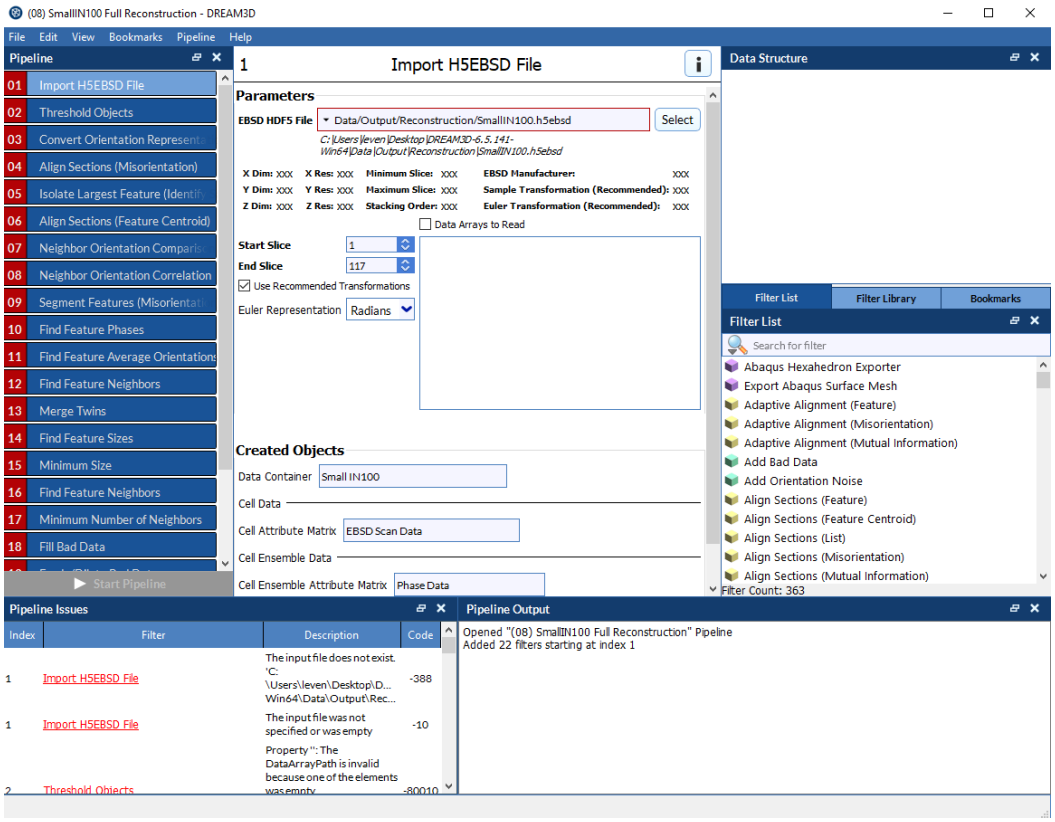


Figure 83: DREAM.3D GUI

4.3.3 Paraview

ParaView is an open-source multiple-platform application for interactive, scientific visualization. It has a client–server architecture to facilitate remote visualization of datasets and generates level of detail (LOD) models to maintain interactive frame rates for large datasets. It is an application built on top of the Visualization Toolkit (VTK) libraries. ParaView is an application designed for data parallelism on shared-memory or distributed-memory multicomputers and clusters. It can also be run as a single computer application. Therefore, Paraview basically is an open source and free high-end data visualization tool for large datasets.

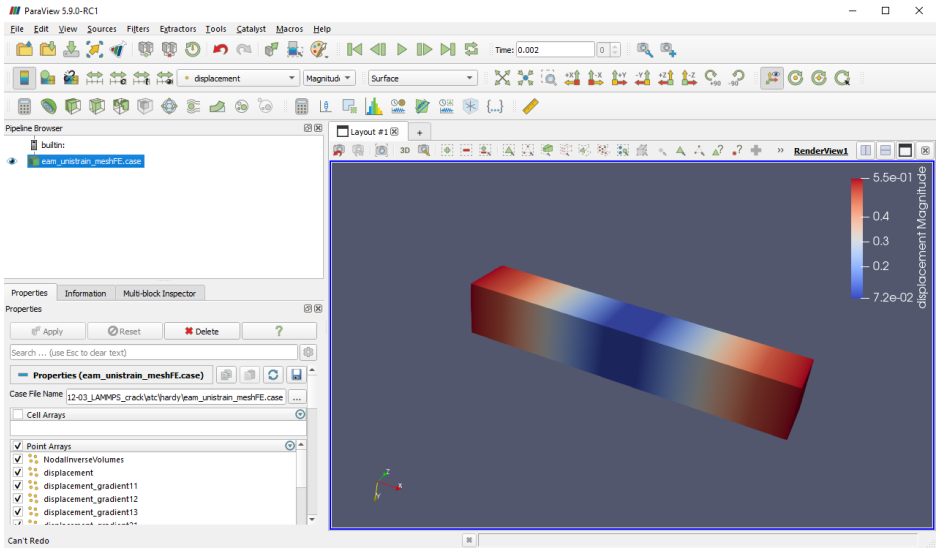


Figure 84: Paraview GUI

BZN produced videos and images mainly, but not exclusively with Paraview. We also did the 2D statistical visualization of grain distributions and other statistics in Paraview.

4.4 C++ code extension

The original SPPARKS code was changed which is shown in Appendix A and B, in order to achieve the results presented in the following section. From the SPPARKS POTTS application style the potts/additive and potts/weld codes were combined. In this way the original functionality of and confidence in the code can be ensured. With the modification BZN is able to create 3D multilayer (ML) welding simulations where the grain structure and its parameters can be extracted.

Basically, the potts/additive code was modified and only a few parts of the potts/weld application were implemented into the code. In SPPARKS every module is built up from several codes which have different extensions. Firstly, the app_style has to be chosen. This command defines what model or application SPPARKS will run. Three kinds of applications can be applied: on-lattice, off-lattice, and general. They differ in the space on which the events occur, on-lattice applications define a set of static sites on this space, the off-lattice applications define a set of mobile sites. Finally in case of the general applications spatial information is not needed. For the additive and weld module the app_style potts command is used, as already mentioned they can be evolved only with the rejection kinetic Monte Carlo algorithm. During the extension of the code the implementations of these commands were not changed. The changes were carried out in the potts/additive files with extensions of .cpp and .h. The modified code of the .h file can be seen in Appendix A, the changes made in the code are marked with yellow colour. As it can be seen only two new variables were given to this part, called *distance* and *q*, they are necessary for the site event rejection function. The *compute_mobility* function also received a third variable, later in the .cpp code the value of *q* is implemented into this, therefore in case of the computation of the mobility not only the distance of the given point can be included but also the size of the heat affected zone.

The largest change in case of the .cpp code was the exchange of the equation of the grain boundary mobility coefficient which appears in the mobility equation. The original equation ($M_0 = \exp(-exp_{factor} \times x)$) did not represent well the process of welding as it was developed for the additive process, therefore this was changed to the one which can be found in the potts/weld application; however, some modification was

necessary. The problem was that the *haz* parameter which is the width of the heat affected zone (haz) surrounding the melt pool, could not be implemented in the model in this form, as the potts/additive application includes the size of the HAZ too not only the size of the weld pool. The following equation was included into the model:

$$M_0 = 1 - \frac{d(x)}{q/h_f} \qquad (26)$$

where q represents the distance of the haz from the origin and the h_f is the HAZ width factor. This parameter was not included automatically in the in. code, which is going to be introduced later, it has to be changed manually in the .cpp code, during the simulations it was varied between the range of 2-10. The modifications can be seen in Appendix B. The modified parts of the model were also marked with yellow colour.

4.5 SPPARKS parameter identification

In this section, the results and workflow of results of kMC multilayer weld simulations are introduced. BZN achieved these results using the modified SPPARKS code.

As it was already mentioned, the modifications in the code had to be tested by running test simulations and checking them manually. This means primarily visual inspections on the simulation results were used, in order to decide whether a run produced the expected grain structure or not. This was the first phase in testing the code. Later visual inspection to build the desired multilayer weld structure was still used, this needed parameter identification and tuning.

4.5.1 SPPARKS input files

The SPPARKS input files are in charge to run and control the simulation. They are written in SPPARKS specific script language, i.e. in a text editor. For multilayer welds we had to develop a file structure and sequence of the files read in. The running sequence is shown in the following figure. Basically, the program can be separated into two main parts, the sites initialization and the Potts weld simulation.

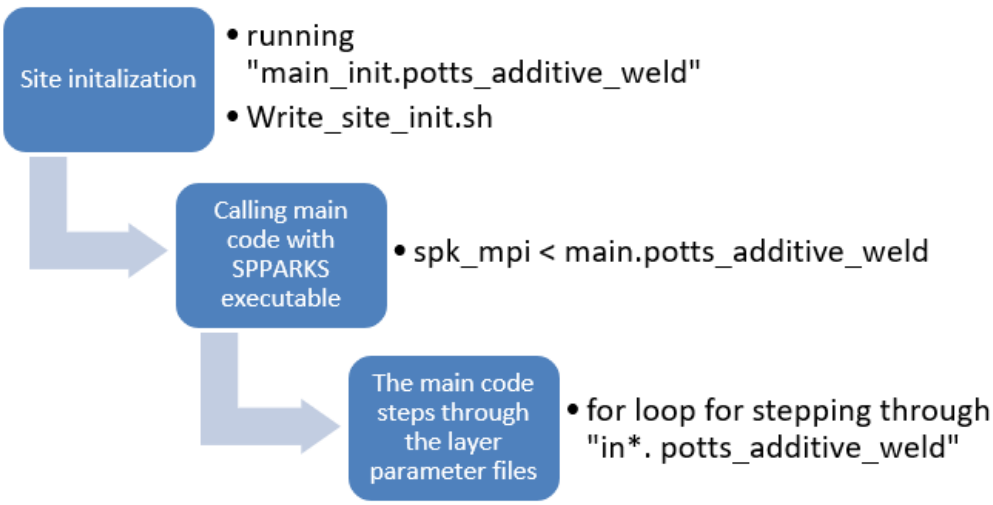


Figure 85: SPPARKS input file structure

The input files contain all necessary data for running a simulation. The first step in order to make a base grain structure on the simulation volume is the site initialization. This is used for setting up the rectilinear 3D space and initializing grains. The main parameters of this file are shown in Table 19.

Variable	Name	Description
Scale	scale	Real world and lattice site scaling factor
Length in direction X	L_x	Length in x direction in 'site'
Length in direction Y	W_y	Length in y direction in 'site'
Length in direction Z	H_z	Length in z direction in 'site'
Temperature constant	temperature	It is a temperature factor for MC simulation

Table 19: Initalization script variables

The main_init.potts_additive_weld is where the parameters of number of sites (the simulation volume), initial grain size, periodicity of the domain, lattice size, and scale are defined. After the determination of the scale factor a *c* constant is given with a unit of micron/site, this gives the relationship between the real size of the grains and the simulated one, in other words it gives the number of microns per lattice site, like it was already introduced in the potts/weld module. It has the following formula:

$$c = \frac{D_R}{D_M} \qquad (27)$$

where D_r is the real grain size and D_M is the dimensionless Monte Carlo simulated grain size (measured in number of grid spacings).

Thereafter the size of the simulation box can be given in mm and can be converted into sites. Finally, the box where the simulation will be carried out can be created. The visual output of this structure is stored in a dump file, which will serve as the input for further weld simulation. An example of the creation of the site initialization and the creation of the dump file can be seen in Appendix C. The results of this example are shown in Figure 86.

With the *scale* and *temp* variables one can change the average grain size of the matrix structure, which will affect the welding simulation. After initialization, the write_site_init.sh shell script file converts the initialized sites file for further processing in SPPARKS as a „.dump" file.

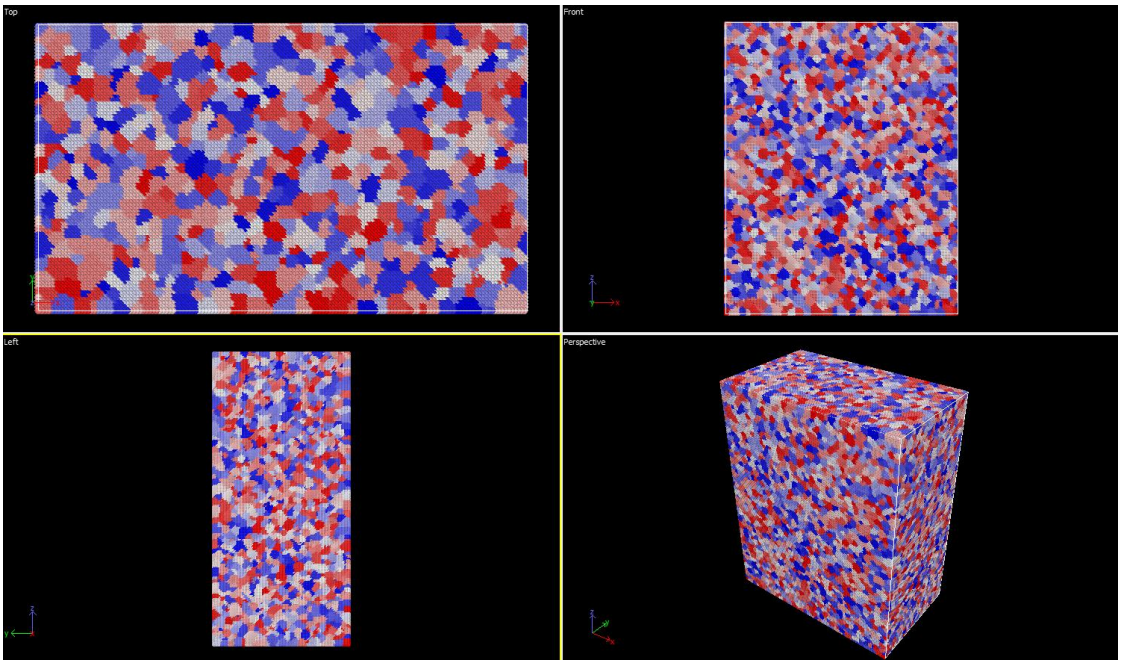


Figure 86: Initalized grain structure

The 'main.potts_additive_weld' is the second script file in order that should be run. This should contain the same 3D geometry properties as the initialization sequence. Besides the parameters listed in the previous step the welding speed is also included in this script file, where two types of speed can be distinguished, the simulated and the real welding speed.

For the determination of the relationship between the simulated welding speed and the real welding speed a new parameter must be introduced, which is called the MC simulation time or MC simulation iteration steps, t_{MCS}. The unit of t_{MCS} is MCS – Monte Carlo steps. Just like it was previously introduced the t_{MCS} is calculated with the help of the simulation welding speed (v_p) the real welding speed (V), the real welding time (t_r) and the factor (c). The form of the related equation is the following:

$$t_{MCS} = \frac{t_r}{v_p}\frac{V}{c} \qquad (28)$$

Variable	Name	Description
Scale	scale	Real world and lattice site scaling factor
Length in direction X	L_x	Length in x direction in 'site'
Length in direction Y	W_y	Length in y direction in 'site'
Length in direction Z	H_z	Length in z direction in 'site'
Temperature constant	temperature	Its a temperature factor for MC simulation
Welding speed simulated	`s1_weldspeeds`	
Welding speed real	`s1_weldspeedr`	

Table 20: Main script variables

The values of length in x, y and z-direction must be converted in this case into sites. An example of these settings is shown in Appendix D, where for the parameters the following values are used. The L_x, W_y, and H_z are converted to sites in the code with the help of the c factor.

Variable	Value	Description
Scale factor	0.1	-
c factor	200	[micron/sites]
L_x	47	[mm]
W_y	25	[mm]
H_z	88	[mm]
temperature	0.1	It is a temperature scaling factor for MC simulation
`s1_weldspeeds`	8.5	[sites/MCS]
`s1_weldspeedr`	2000	[microns/s]

Table 21: Values set in the example

The main.potts_additive_weld file also contains the number of layers, it is marked with "a". It gives the number of in*.potts_additive_weld files. The output of the main routine is the following ML weldment. Take note that the 'in*.potts_additive_weld' file structure is described below.

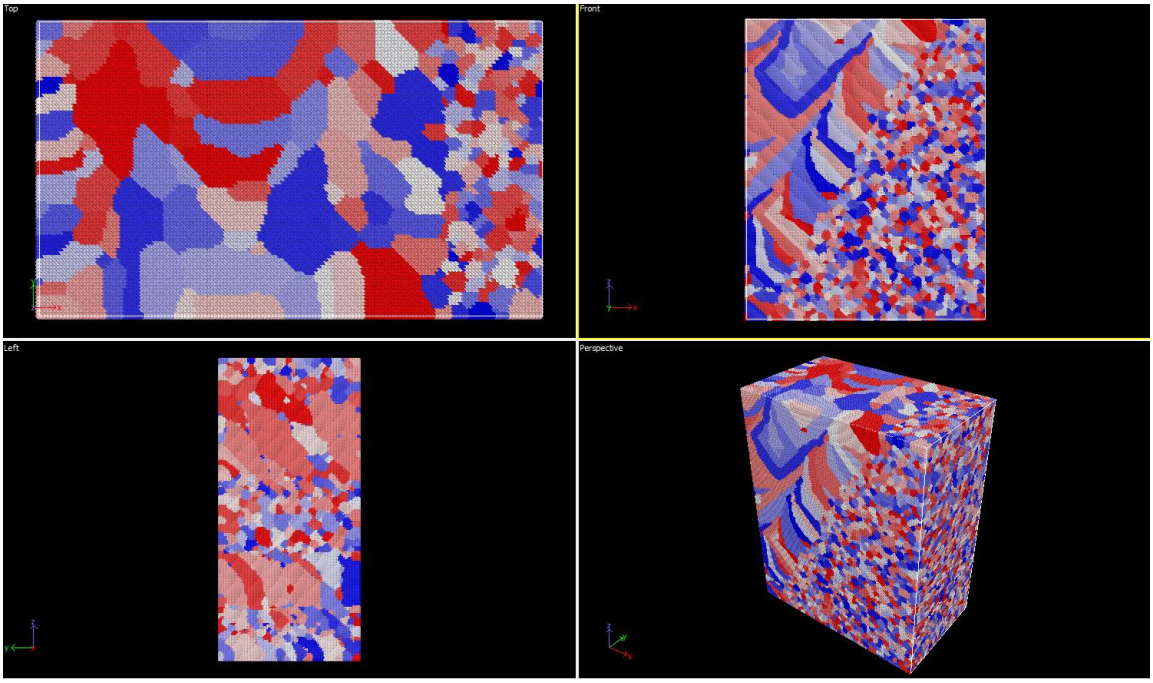


Figure 87: Output of the main routine

An example for the layout of the created 'in*.potts_additive_weld' file can be seen in the Appendix E. The code contains the parameters customized for each layer in Table 22.

Variable	Name	Description
No. Of coloumns	s1_colnums	Parallel weld lines on plane
Spot width	s1_spw	Welding spot width in 'site'
Spot HAZ width	s1_spwhaz	Welding spot HAZ width in 'site'
Spot depth	s1_spd	Welding spot depth in 'site'
Spot depth HAZ	s1_spdhaz	Welding spot HAZ depth in 'site'
Spot tail length	s1_spt	Welding spot tail in 'site'
Spot tail HAZ length	s1_spthaz	Welding spot tail HAZ in 'site'
Spot cap length	s1_spc	Welding spot cap in 'site'
Spot cap HAZ length	s1_spchaz	Welding spot cap HAZ in 'site'
Spot starting position in X	s1_startposx	in 'site'
Distance of the symmetric weld	s1_distance	Total Distance on plane between first line and last weld line
Spot starting position in Z	s1_startposz	Height of the welding plane

Table 22: Parameters included in the 'in*.potts_additive_weld' file

Here as already listed in the table the value of several parameters can be defined. One of the main differences from the original potts/additive is that the number of welds therefore their location in each layer can be defined, and also the movement of the weld line in x and z-direction can be set with the help of *s1_startposx* and *s1_startposz* variables. The last modified file was the run.potts _additive_weld code. The parameters defined in it are listed in Table 23.

Variable	Name	Description
Starting position in y	s1_startposy	Gives the starting position of the weld in the y direction, usually it is 0
	s1_distance	$s1\_startposx \times 2$
The real length of the weld	s1_weldlengthr	Gives the length of all the weld line in 'mm' in the given layer
Real weld time	s1_weldtimer	Gives the real welding time for a layer [s]
The simulated length of the weld	s1_weldlengths	Gives the length of one weld line in 'site'
Simulated weld time	s1_weldtimes	Gives the simulation time of the welding [MCS]
Simulation time	t_simtime	Gives the simulation time where the relationship between the real welding and the simulation is given [MCS]
Frequency of dump	t_dumpfreq	Gives the time of the frequency of writing a dump file

Table 23: Parameters used in the run.potts _additive_weld code

The real length of the given weld is determined with the help of the following equation:

$$s1\_weldlengthr = (W\_y + s1\_spthaz + s1\_spchaz) \times c \times s1\_colnums$$

where W_y is the length in y-direction, $s1\_spthaz$ is the heat affected zone welding spot tail HAZ, $s1\_spchaz$ is the length of the welding spot cap of the HAZ, c is the scale factor, and $s1\_colnums$ is the number of the welding lines in a given layer. Therefore, it gives the length of all the welding line in a layer, where not only the length of the simulation but also the necessary length for the starting and for the finishing of the welding.

The real weld time is calculated with the proportion of the real weld length and the real weld speed.

$$s1\_weldtimer = \frac{s1\_weldlengthr}{s1\_weldspeedr}$$

Similarly, to these equations the simulated length and time can also be calculated. In the later part of the run. potts _additive_weld code the $t\_simtime$ and $s1\_weldlengths$ variables are utilized.

The welding parameters are called into the app_style potts/additive code with the following line:

"app_style potts/additive ${nspins_const} ${s1_spw} ${s1_spt} ${s1_spd} ${s1_spc} ${s1_spwhaz} ${s1_spthaz} ${s1_spdhaz} ${s1_spchaz} 0.0 "

Here the listed arguments in the given sequence are the following:

- Number of spins
- Width of the melt pool
- Length of tail from meltpool midpoint
- How many lattice sites deep the melt pool is
- Height of the cap leading the meltpool
- Size of the HAZ surrounding the melt pool (must be larger than spot_width)
- Length of hot zone behind meltpool (must be larger than melt_tail_length)
- Depth of the hot zone underneath the meltpool (must be larger than melt_depth)
- Size of HAZ infront of the melt pool (must be larger than cap_height)
- Exponential parameter for mobility decay in haz M(d) = exp(-exp_factor * d) (it is 0 as later it is not used)

4.5.2 Geometry parameters and handling

BZN's SPPARKS ML weld simulation is always made on cartesian 3D rectilinear space. The 3D space is built up by finite distance lattice sites in directions denoted conventionally X-Y-Z.

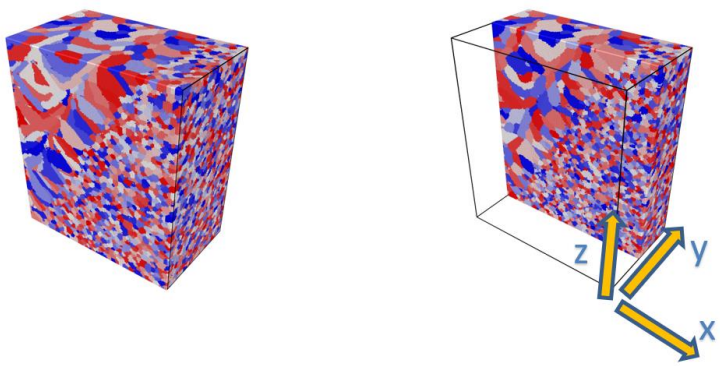


Figure 88: Definition of 3D directions in geometry

BZN is able to simulate on their hardware around ~200-300 million sites, the above shown 12-layer ML weldment is about 100 million sites large. Before the beginning of the simulation, the scaling factor of the real-world weldment and the simulation geometry has to be calculated. For this purpose, Excel was used. An example calculation can be seen below, HAZ depth changed accordingly to get good results, the others are fixed.

Weld direction	y						
Scale factor	0.1						
c [micron/sites]	L_x [mm]	W_y [mm]	H_z [mm]	L_x [sites]	W_y [sites]	H_z [sites]	NSPINS [db]
200	60	3.5	37.70	300	17.5	188.5	989625

Technology Parameters															
		Calculated sizes of the welding pool and HAZ							Simulation sizes of the welding pool and HAZ						
Layer number	Seams number	Spot width [mm]	Spot depth [mm]	Spot tail [mm]	Spot cap [mm]	HAZ width [mm]	HAZ depth [mm]	HAZ cap [mm]	Spot width [sites]	Spot depth [sites]	Spot tail [sites]	Spot cap [sites]	HAZ width [sites]	HAZ depth [sites]	HAZ cap [sites]
1	1	5.40	3.50	6.20	2.20	10.30	3.70	3.70	27.00	17.50	31.00	11.00	51.50	18.50	18.50
2	1	9.40	5.00	12.20	4.00	14.30	4.00	6.00	47.00	25.00	61.00	20.00	71.50	20.00	30.00
3	1	13.00	4.60	14.20	4.00	17.30	3.00	6.00	65.00	23.00	71.00	20.00	86.50	15.00	30.00
4	1	15.60	5.50	15.50	3.00	19.80	2.60	5.70	78.00	27.50	77.50	15.00	99.00	13.00	28.50
5	2	12.25	5.15	15.60	3.75	16.30	4.00	5.85	61.25	25.75	78.00	18.75	81.50	20.00	29.25
6	2	14.55	5.25	14.85	4.00	19.05	3.00	5.90	72.75	26.25	74.25	20.00	95.25	15.00	29.50
7	2	14.55	5.48	15.50	3.50	21.55	3.00	5.70	72.75	27.40	77.50	17.50	107.75	15.00	28.50
8	3	12.27	5.27	15.67	4.67	16.93	3.50	7.00	61.35	26.35	78.35	23.35	84.65	17.50	35.00
9	3	2.80	5.38	15.77	4.00	17.83	3.00	6.40	14.00	26.90	78.85	20.00	89.15	15.00	32.00
10	3	13.83	5.07	14.73	3.67	18.67	2.50	5.97	69.15	25.35	73.65	18.35	93.35	12.50	29.85
11	4	12.85	5.18	15.85	4.00	17.80	3.00	6.13	64.25	25.90	79.25	20.00	89.00	15.00	30.65
12	4	13.25	5.05	15.05	4	18.60	2.5	6.05	66.25	25.25	75.25	20.00	93.00	12.50	30.25

Layer number	Seams number	Starting point [mm]	Distance [mm]	Increment [mm]	High [mm]	Starting point [sites]	Distance [sites]	Increment [sites]	High [sites]
1	1	0.00	0.00	0.00	3.70	0.00	0.00	0.00	18.50
2	1	0.00	0.00	0.00	7.70	0.00	0.00	0.00	38.50
3	1	0.00	0.00	0.00	10.70	0.00	0.00	0.00	53.50
4	1	0.00	0.00	0.00	13.20	0.00	0.00	0.00	66.00
5	2	5.20	10.40	0.00	17.20	26.00	52.00	0.00	86.00
6	2	6.10	12.20	12.20	20.20	30.50	61.00	61.00	101.00
7	2	7.00	14.00	14.00	23.20	35.00	70.00	70.00	116.00
8	3	9.80	19.60	19.60	26.70	49.00	98.00	98.00	133.50
9	3	11.80	23.60	11.80	29.70	59.00	118.00	59.00	148.50
10	3	12.70	25.40	12.70	32.20	63.50	127.00	63.50	161.00
11	4	15.50	31.00	15.50	35.20	77.50	155.00	77.50	176.00
12	4	17.50	35.00	11.67	37.70	87.50	175.00	58.33	188.50

Figure 89: Technology parameters in Excel

4.5.3 Weld parameters

The weld pool geometry is defined in SPPARKS as it was shown previously (Figure 81), in the modified code the same parameters were implemented. There are two other parameters that need to be mentioned for setting the weld simulation: the welding speed in sites/MCS and temperature of the weld.

4.6 Model validation

For the validation of the modified BZN model, the weld simulation parameters, and results of reference [52] were used. In this simulation the shape of the weld pool was fixed, the weld width at the top of the plate was 2.54 mm and at the root it was 1.52 mm. With this the α ratio is 0.6. The width of the weld was fixed to 300 sites. With this information the lattice constant can be calculated, $c = \frac{2.54 \times 1000}{300} = 8.47$ microns per site. The initial microstructure of the base metal was 61 microns. With the lattice constant it can be calculated that in the simulation the average grain diameter is 7.2 sites. Further constant data were the simulation weld speed ($v_p = 10\,\frac{\text{site}}{\text{MCS}}$), the size of the heat affected zone ($haz = 150$ sites) and the plate thickness ($h = 0.7417$ mm). The shape of the weld pool can be seen in Figure 90.

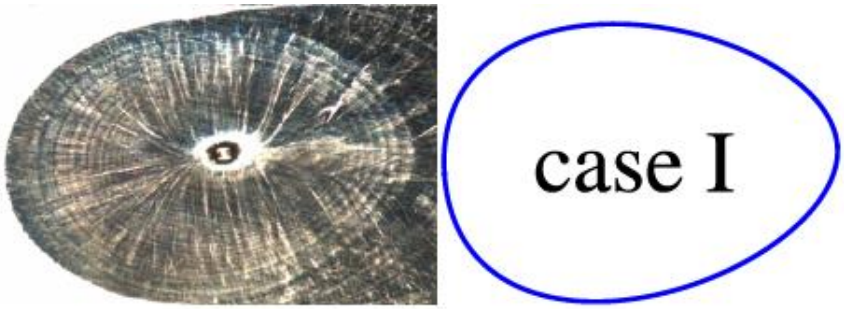


Figure 90: Shape of the weld pool [52]

Datas from [52]	**Weld direction**	**y**	**Scale factor**	**1**				
	c [micron/site]	**L_x [mm]**	**W_y [mm]**	**H_z [mm]**	**L_x [sites]**	**W_y [sites]**	**H_z [sites]**	**NSPINS**
	8.47	5	13.35	0.74	591	1577	66	8.2e7
Datas for the BZN modified model	**Weld direction**	y	**Scale factor**	0.25				
	c [micron/site]	**L_x [mm]**	**W_y [mm]**	**H_z [mm]**	**L_x [sites]**	**W_y [sites]**	**H_z [sites]**	**NSPINS**
	33.88	5	13.35	0.74				1.29e6

Table 24: Parameters used for the validation of the BZN model

The simulations were carried out with varying values of β, which controls the curvature of the pool through the thickness (z-axis) of the domain. The values of β were set to 0.25, 0.5 and 0.75 in case of the reference literature. The results of the simulations can be seen in Figure 91 - Figure 93. It can be seen that the modified model follows quite well the results of the model made with potts/weld.

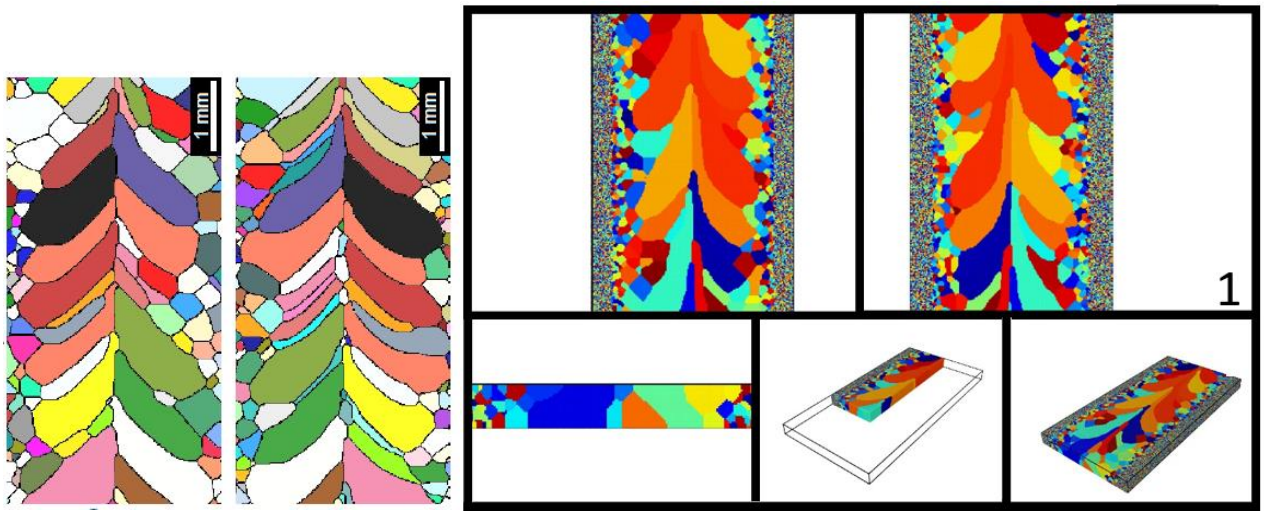


Figure 91: Top and Bottom view in case of $\beta = 0.25$ [52]

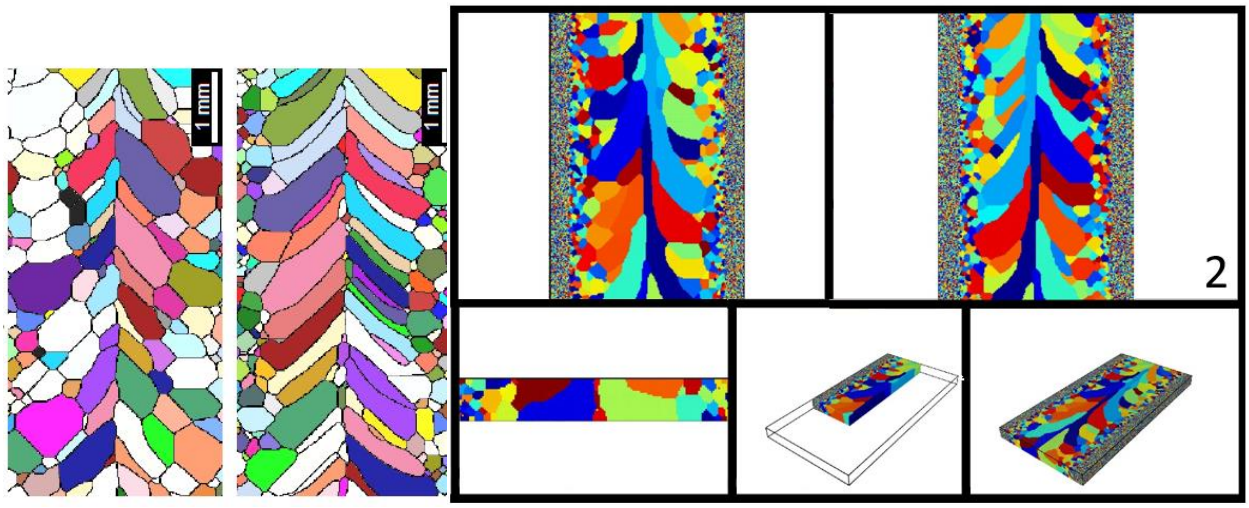


Figure 92: Top and Bottom view in case of $\beta = 0.5$ [52]

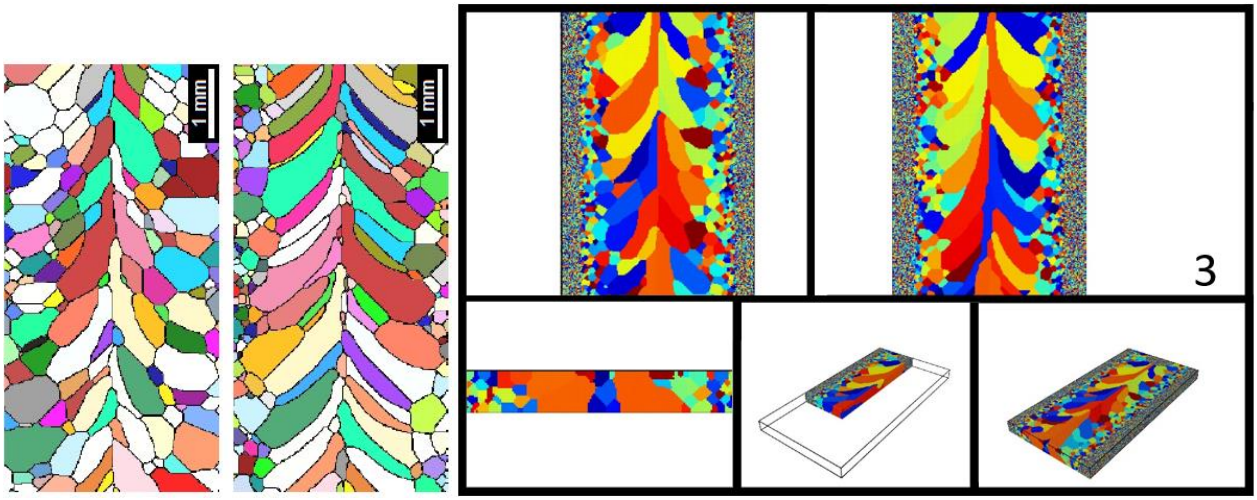


Figure 93: Top and Bottom view in case of $\beta = 0.75$ [52]

4.7 Results of 2D Multilayer simulations

Besides the 3D simulations 2D multilayer simulations were also carried out, where the effect of the different parameters was studied. For the simulation, the geometry given by MU can be seen in Figure 6 and the MINA weld shape was used. Therefore, the dimension of the weld pool, and the examined size was kept constant, and a scale factor was also not changed.

In the simulations the weld speed [sites/MCS] and the HAZ width factor with respect to spot width were varied. The values of the weld speed were 36.5, 37.5, 38.5 and 39.5 sites/MCS. The HAZ factor varied between 2, 5, 8 and 10. The 2D Multilayer simulation results for these parameters are shown in Figure 95. It can be seen that with the increase of the weld speed smaller grains are created.

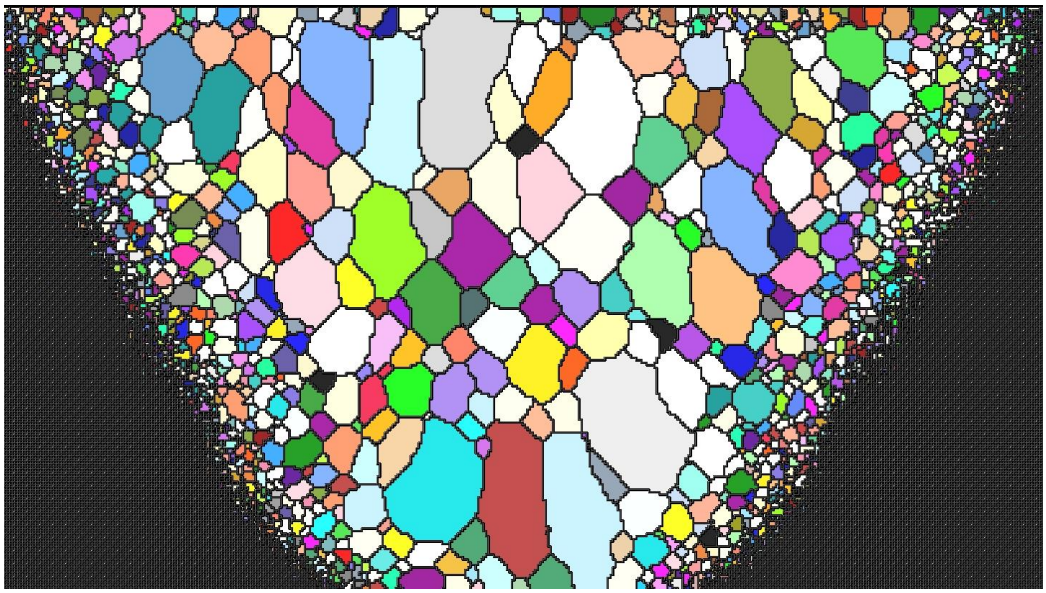


Figure 94: Multilayer SPPARKS BZN modified weld simulation GRAINS

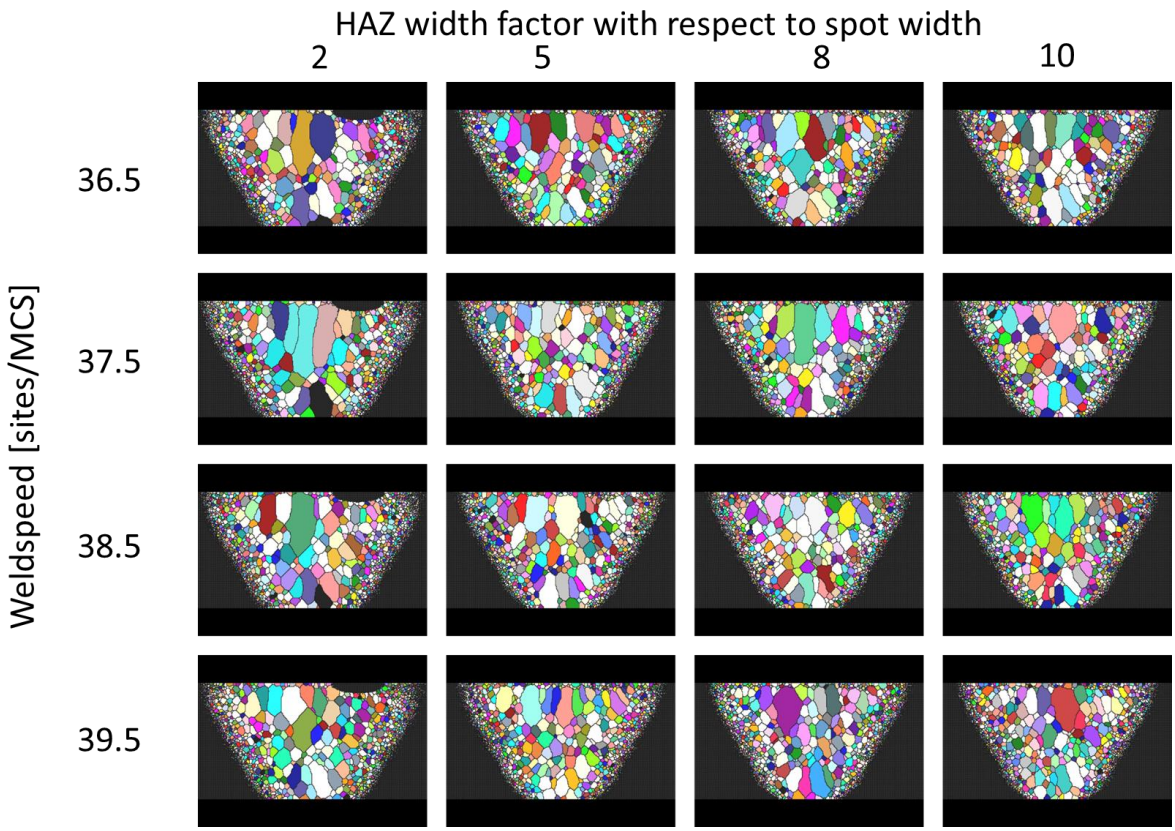


Figure 95: 2D Multilayer - influencing parameters study

The Figure 94 shows the created grains during welding; from these results the orientation of the grains can be obtained. Thereafter the determined orientations were illustrated, and it can be seen in Figure 96, where the blue colour represents the 0°, the red colour the 90° and the white colour the 180°. These data can be used for ultrasonic testing.

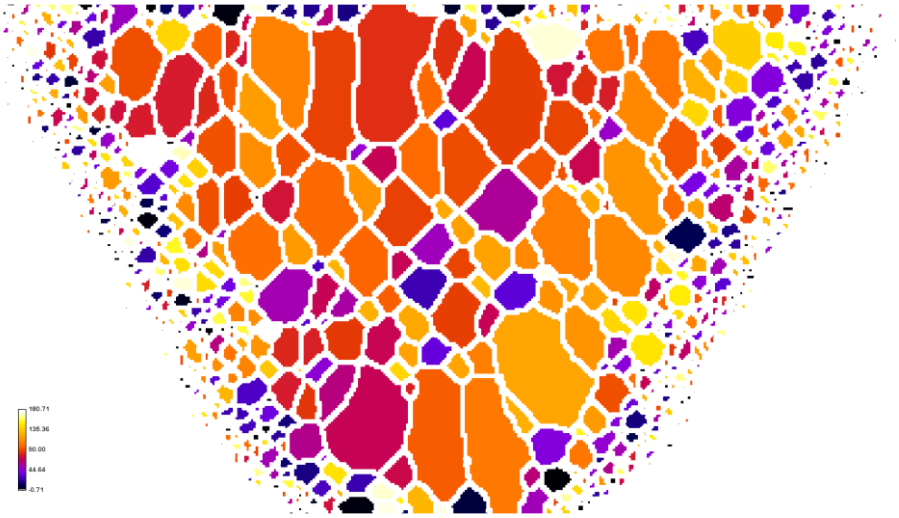


Figure 96: Multilayer SPPARKS BZN modified weld simulation GRAIN ORIENTATION

4.8 Results of 3D Multilayer simulations

4.8.1 Initialization settings

In every simulation, the first step was the set of the initialized grain structure. Figure 97 shows the distribution of the grain size. With the following settings, the most common radius in the grain structure is 0.53 mm:

- Lattice size in micron: 200
- Scaling factor: 0.1
- Box size in sites: 150-87.5-188.5
- Weld speed in micron/sec: 2000
- Weld layer number: 1

Beside the radius of the initialized grains, their sphericity was also investigated with the help of the sphericity index. The sphericity index can be defined as the ratio of the squared volume over the cube of the surface area, normalized such that the value for a ball equal one.

$$sphericity = 36\pi \frac{V^2}{S^3} \qquad (29)$$

From this result can be seen that as the of the radius of the grains increases, their sphericity decreases.

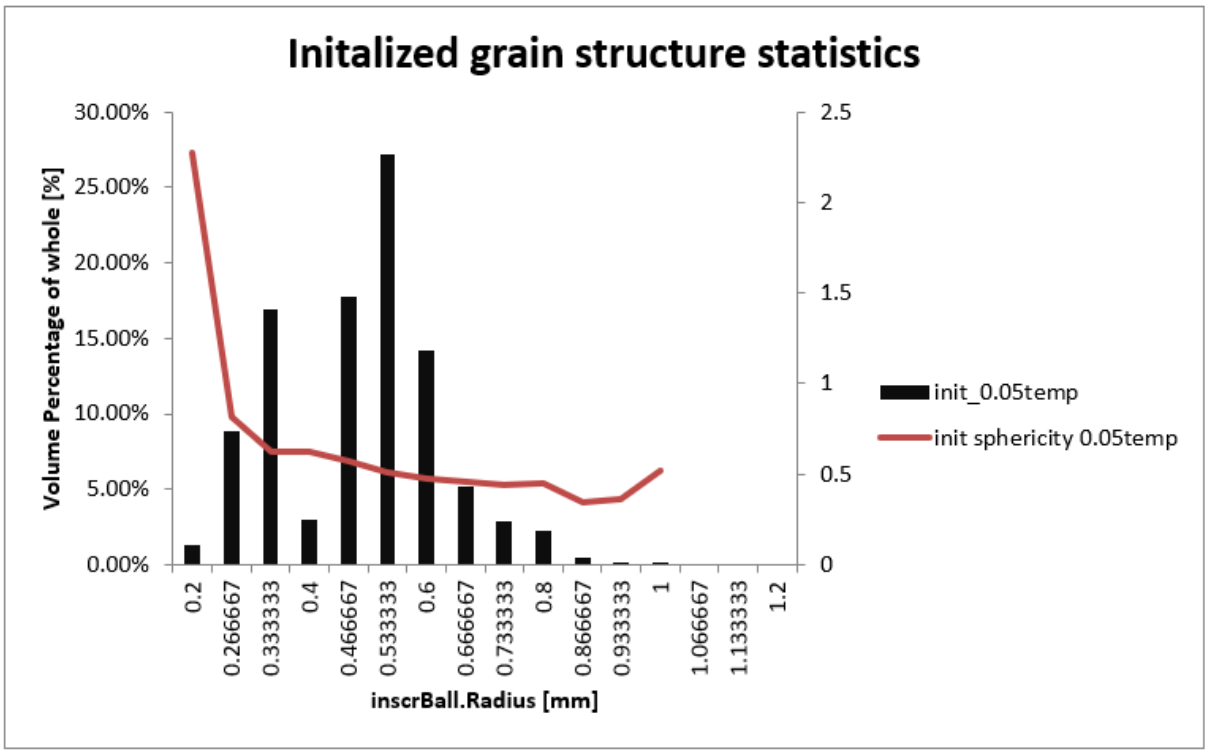


Figure 97: Initalized grain structure statistics

The effect of the temperature settings on SPPARKS initialized grain structure was also investigated. The results are shown below. The figure shows the grain size and the sphericity distribution in the 3D simulation volume. It can be seen that if the temperature is set to a higher value then the initialized grain size would be bigger, without affecting the sphericity of the grains much.

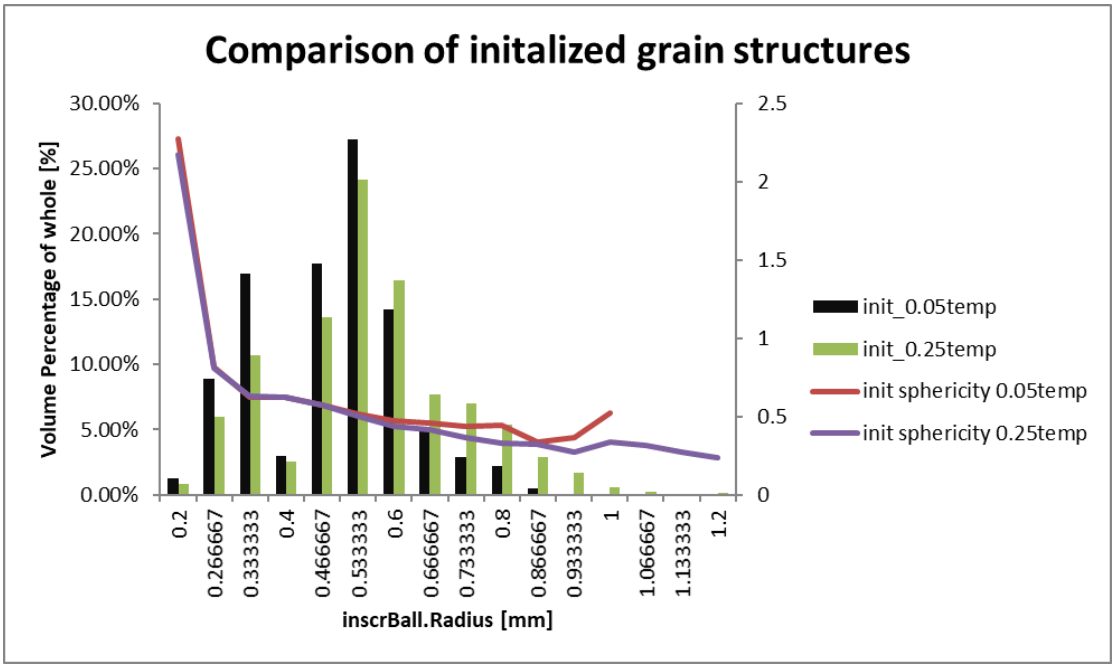
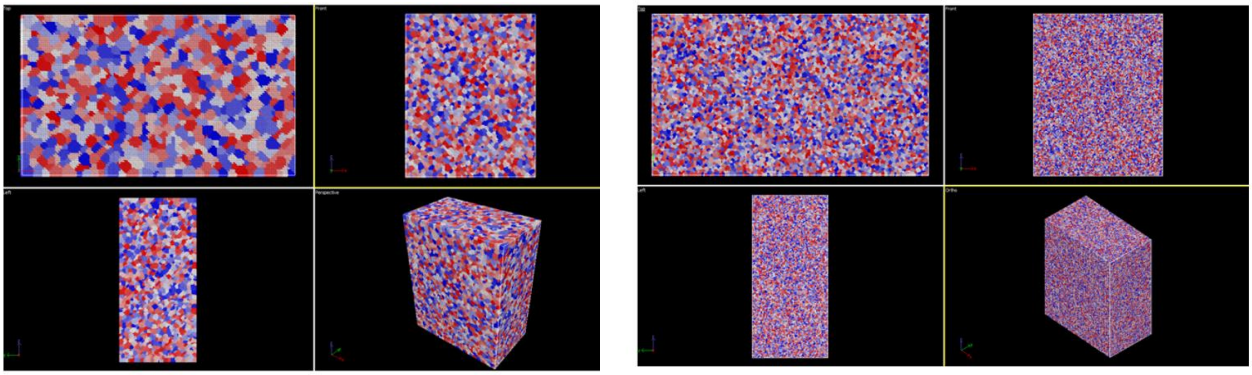


Figure 98: SPPARKS initialized grain distribution with temperature effect

In case of the initialized grain structure, the effect of the scale factor was examined. On the left side of Figure 99, the scale factor was set to 0.1, in case of the picture on the right side 0.3 was chosen. It shows

well that with the increase of the scale factor smaller grains are created. Therefore the following conclusion can be made: the scale parameter changes the size (inscr.ball radius) of grains, so with the increase of the value of scale factor smaller grains are created therefore bigger simulation space and the creation of more grains can be achieved.



200-0.1-150-87.5-188.5-2000-8.5-12-init_0.10temp 66.66666667-0.3-450-262.5-565.5-2000-8.5-12-init_0.10temp

Figure 99: Scale parameter effect on grains

The results were also summarized in a diagram. The statistical evaluation can be seen in Figure 100. In can be seen that there is no large difference between the grain size in mm, also the sphericity of the grains behaves similarly, therefore it can be said that with the variation of this scale factor a larger simulation space can be achieved.

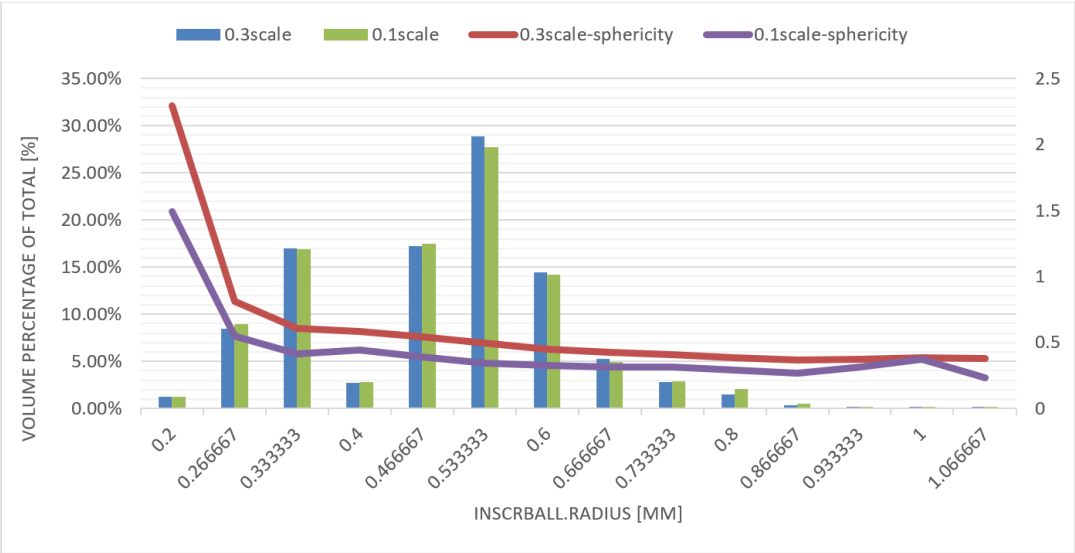


Figure 100: Effect of varying scale parameter on initial grain distribution

From these results, it can be seen that the different settings can have important effects on the initialized grain structure therefore the right parameter settings are essential. In the followings the effects of the different parameters on the welding simulation are introduced.

4.8.2 Weld speed effect on resulting grain structure

The effect of the weld speed was investigated. The simulation weld speed was varied between 17.0, 10.5, 8.5, 6.5 and 4.5 site/MCS. The Statistics for weld speed effect on the distribution can be seen in Figure 102. Besides the five weld speeds, the initial grain size was also illustrated.

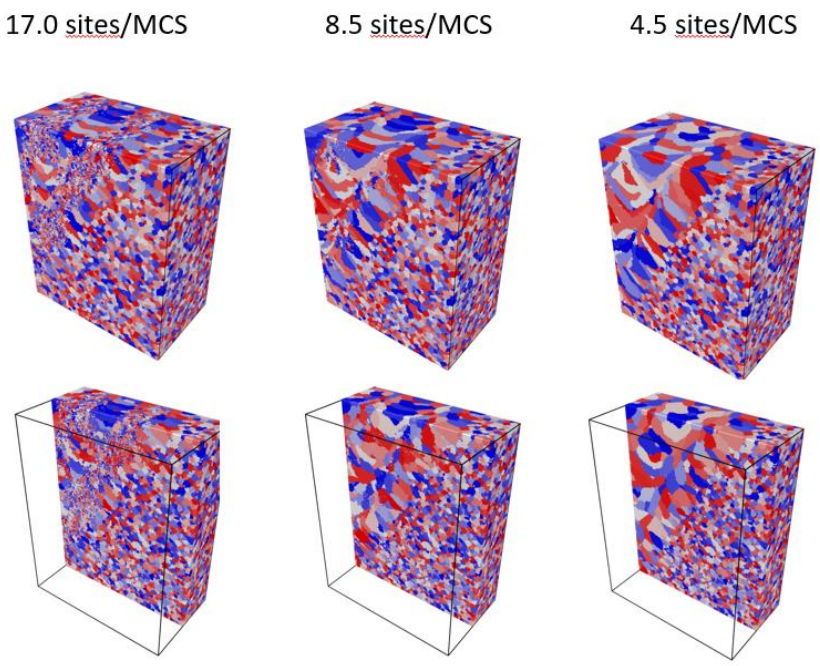


Figure 101: Weld speed effect on weld grain structure

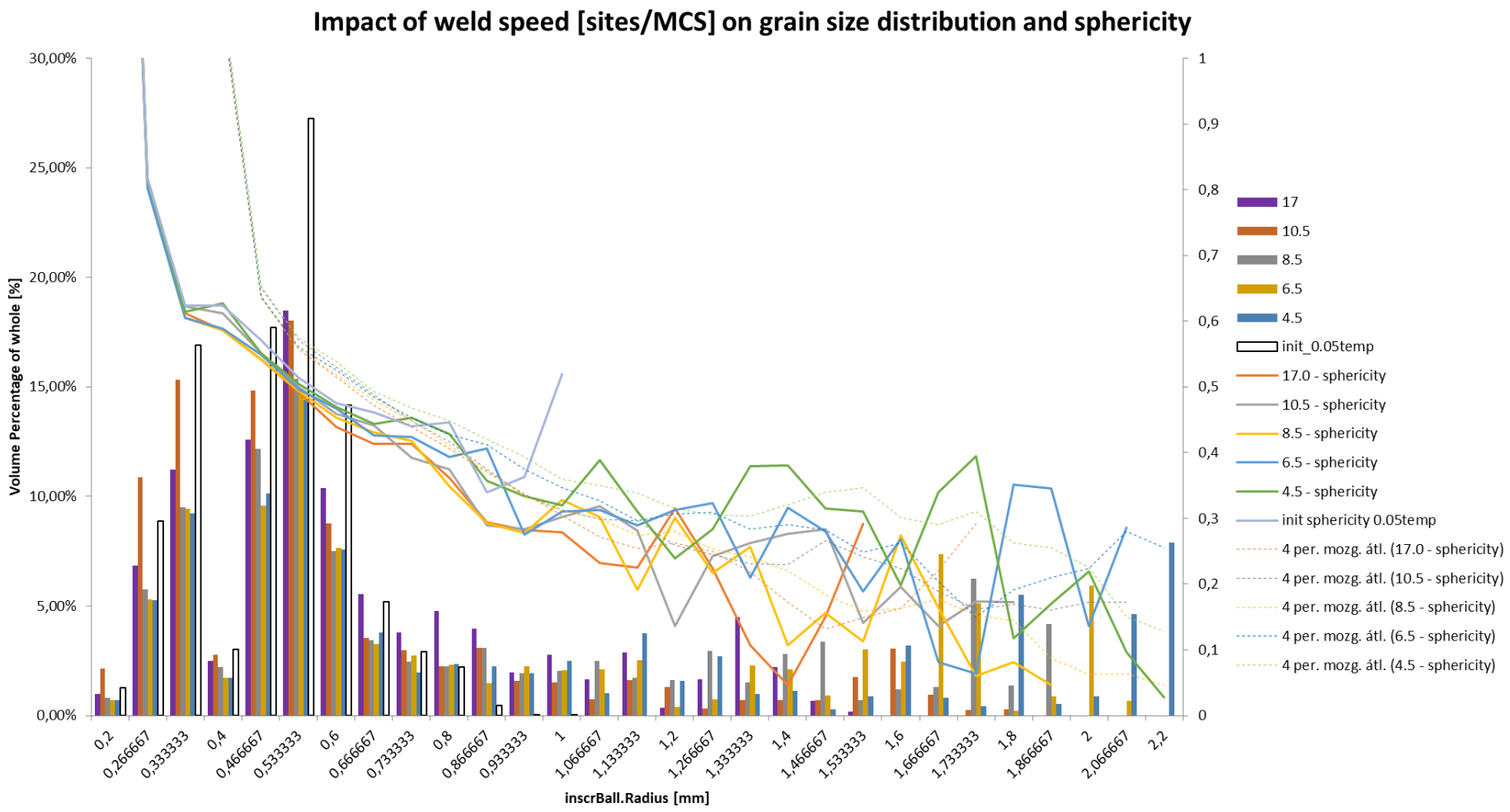


Figure 102: Statistics for weld speed effect on distribution

From the diagram, it can be seen that the basic distribution of grain size is similar in case of the 17.0 and 10.5 sites/MCS weld speeds, in these cases basically no large grains are formed during the simulation, in case of the slower simulation welding speeds larger grains are created, it can be noted that with the decrease of the simulation welding speeds the size of the grains will increase. The sphericity index of the different simulations is also illustrated, in case of the first part similar sphericity and the decreasing nature can be observed however at the larger grains there are numerous local minimum and maximum values.

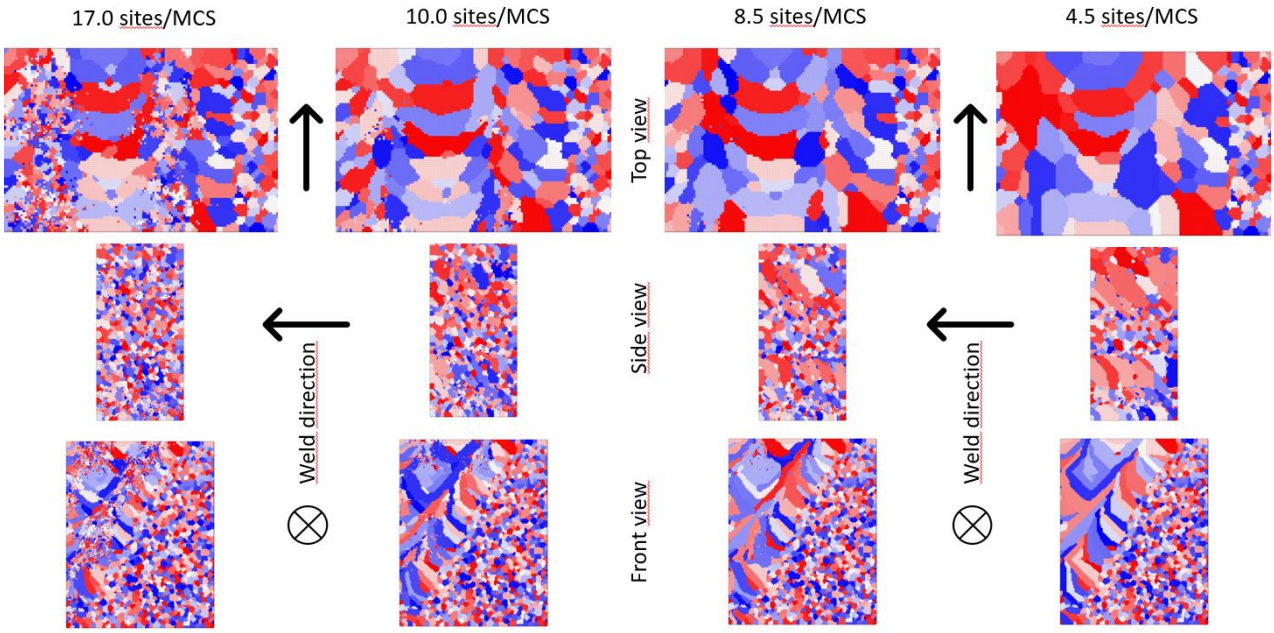


Figure 103: Weld speed effect on weld grain structure in x, y and z direction

Figure 101 and Figure 103 show how the simulation weld speed affects the resulting SPPARKS grain structure. These pictures explain more the previously introduced statistics results and show why in case of the faster simulation weld speeds no large grains occur. It can be seen that in case of the 17.0 sites/MCS weld speed, small grains are created between the larger grains - there are several ‘salt’ or ‘noise’ patterns. This can be observed in every direction. With the decrease of the speed the number of these noise patterns decreases, and therefore the grains will be bigger and smoother, like in the case of 4.5 sites/MCS speed.

From these results the following conclusion can be made: if the speed is lower the grains will be bigger and smoother, if the speed is higher there will be ‘salt’ or ‘noise’ in the result.

4.8.3 Effect of weld temperature change

In case of the fusion zone temperature two values were chosen 0.1 and 0.25. Beside the variation of these values the welding speed was also changed. The visually illustrated grain distribution can be seen in Figure 104.

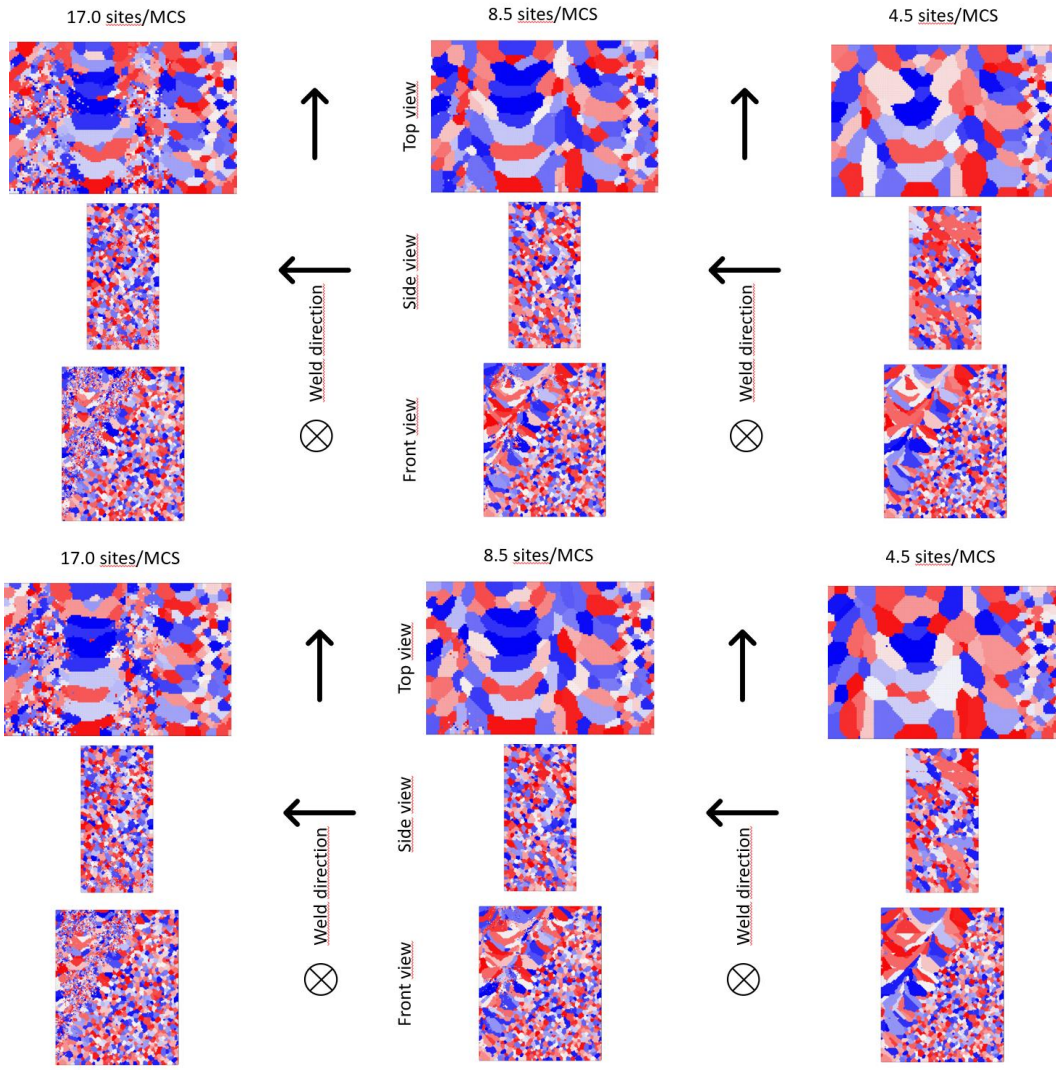


Figure 104: Effect of weld temperature (up 0.10, down 0.25)

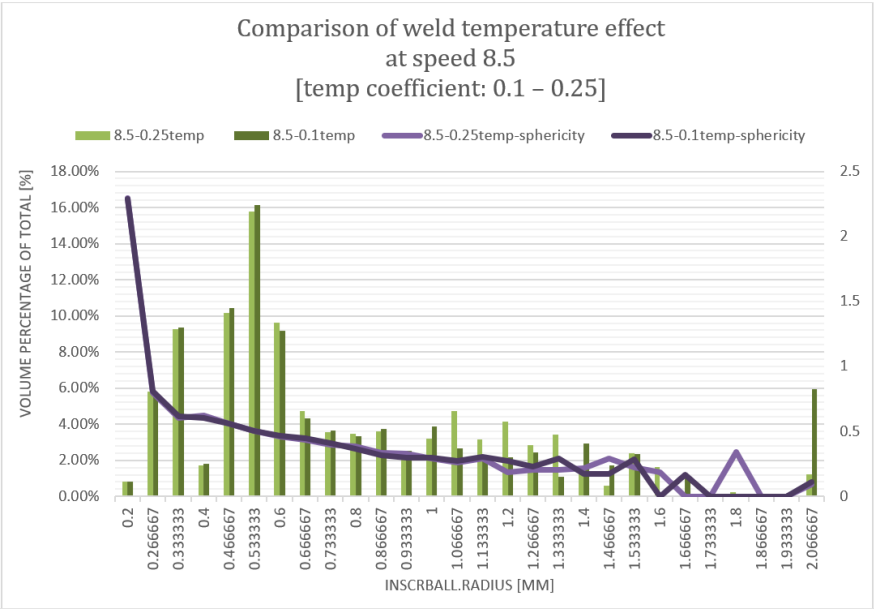


Figure 105: Comparison of weld temperature effect at speed 8.5 sites/MCS

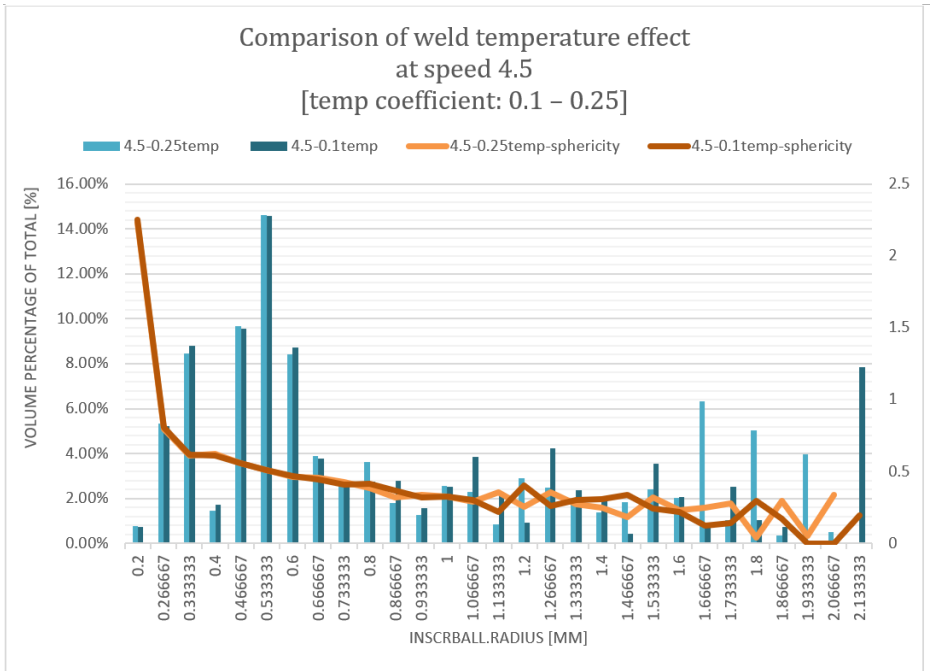


Figure 106: Comparison of weld temperature effect at speed 4.5 sites/MCS

The statistical evaluation of the results can be found in Figure 105 and Figure 106. From the pictures and the statistical results, it can be concluded that with the increase of the temperature of the fusion zone, the resulting grain sizes would be larger. In case of the sphericity of the grains, no large difference can be detected.

4.8.4 Effect of varying scale parameter on resulting grain structure

The effect of the scale parameter variation was not only investigated in the case of the initial grain size but in the case of the welding simulation too. The pictures created about the scale factor of 0.10 and 0.30 can be seen in Figure 107.

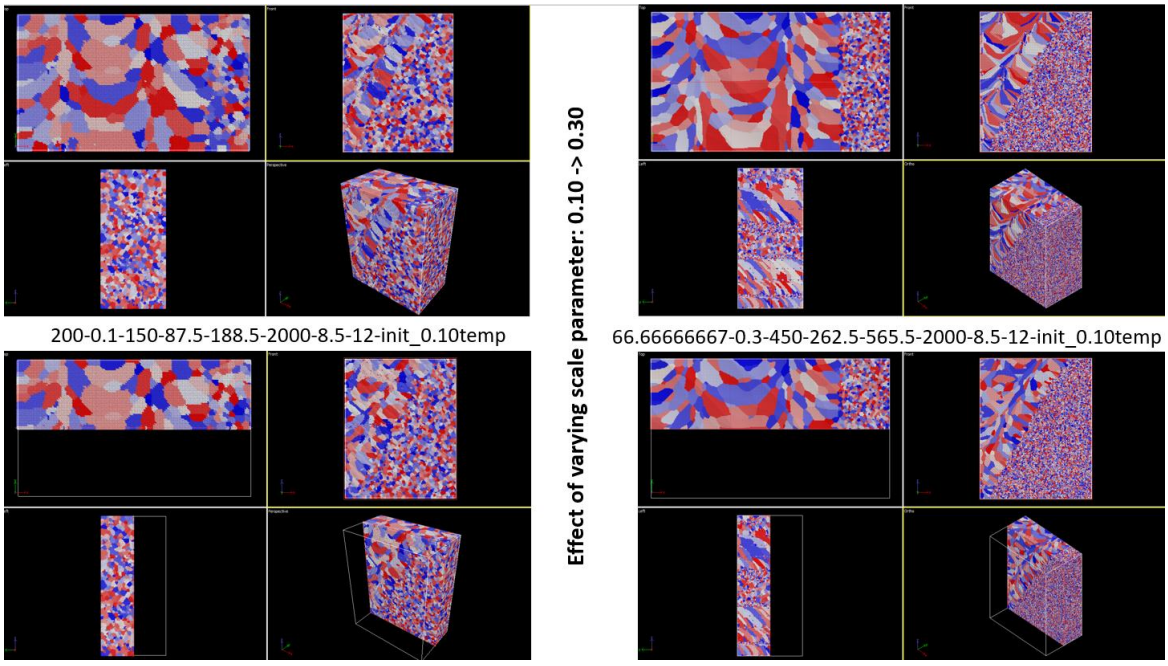


Figure 107: Simulations with different scale parameters

When the scale factor was 0.10 the number of the created sites were around 5.0 million, in case of the 0.30 this number has gone up as high as 133.0 million, therefore the simulation required a larger computer capacity and the simulation ran for a longer time. Figure 108 and Figure 109 show the entire simulated volume with the inclusion of the initial grain sizes.

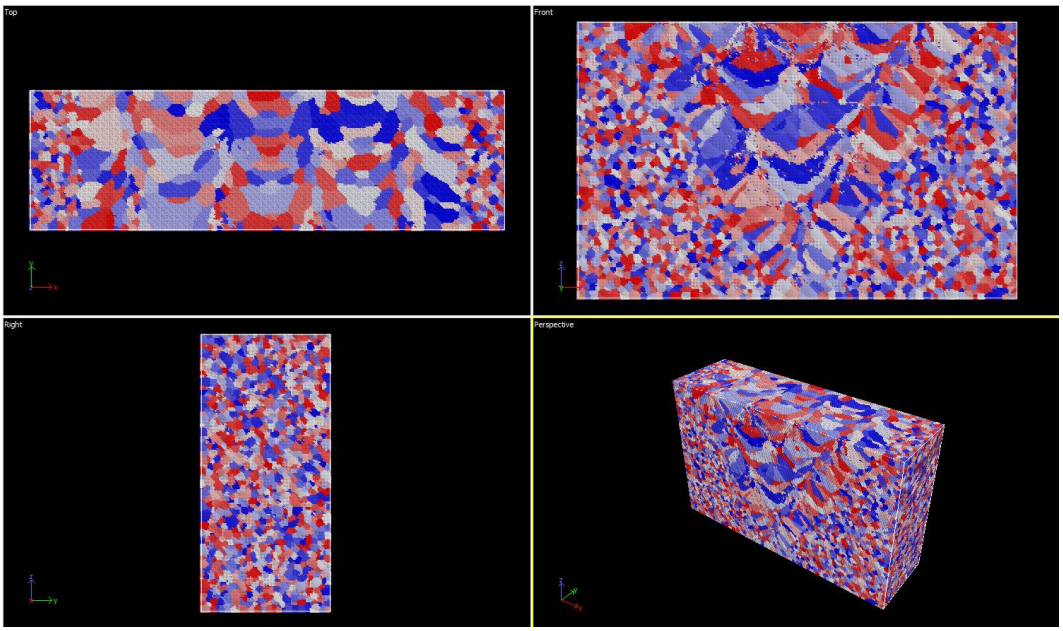


Figure 108: Simulation results on total volume at scale 0.10

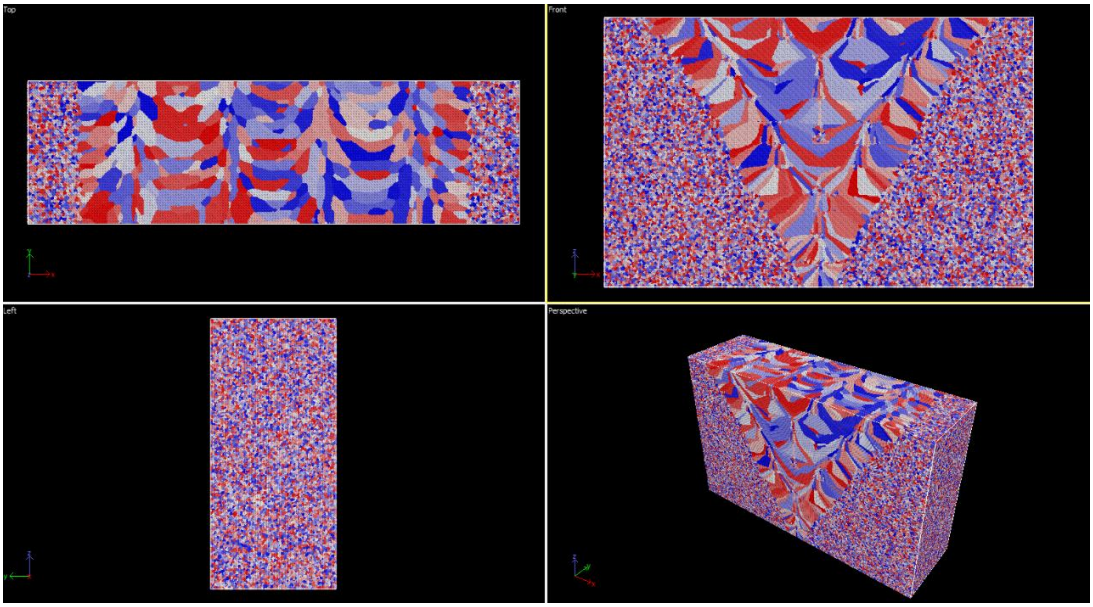


Figure 109: Simulation results on total volume at scale 0.30

These pictures show that with the larger scale factor nicer structure can be created, with the 0.1 scale factor it is more robust, and has several 'noises'. Figure 110 shows the statistical evaluation of the results. It can be seen that with the smaller scale factor the maximal grain size is around 1.6 mm while in case of the larger scale factor the maximal grain size increases to as far as 4.4 mm. For the smaller grain sizes each simulation shows a similar tendency, however because of the 'noises' being present in case of the scale factor of 0.10 large grains cannot be created.

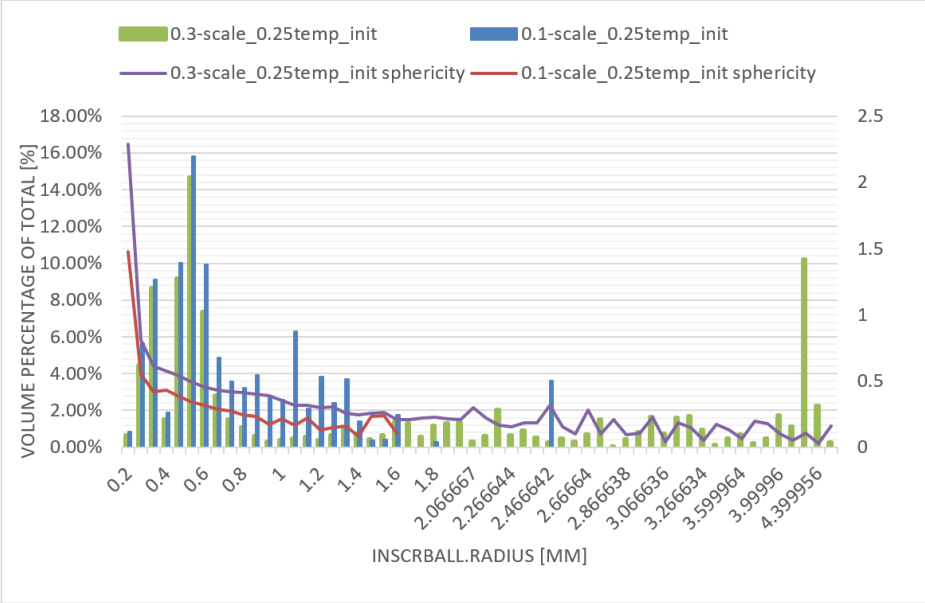


Figure 110: Effect of varying scale parameter on simulated weld grain distribution

Therefore, we can conclude that with the increase of the scale factor we get smoother and nicer results. The bottleneck of 1:1 simulation is the hardware requirements, the simulation of the real-size of the grains without scaling would be very time consuming however with the 0.3 scaling relatively accurate results can be achieved. The scaling of the parameters and the pre-calculation and measuring of the required parameters are essential to receive accurate results.

4.8.5 Effect of initial temperature on final result

The effect of the initial temperature was investigated with two values, 0.10 and 0.25. The results of the simulations can be seen in Figure 111 and Figure 112. It can be seen that with the lower value of the initial temperature scale the simulation will have to create a larger number of small grains, with the increase of this value it can be read from the diagram that in the total volume percentage larger grains appear too. Therefore, the following conclusion can be made: with the increase of the initial temperature scale larger elements can be created.

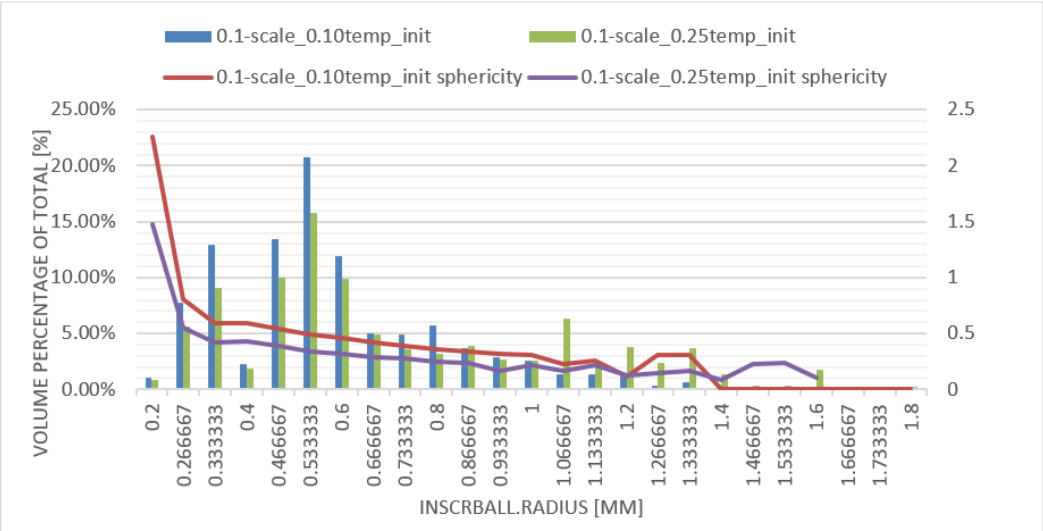


Figure 111: Effect of initial temperature

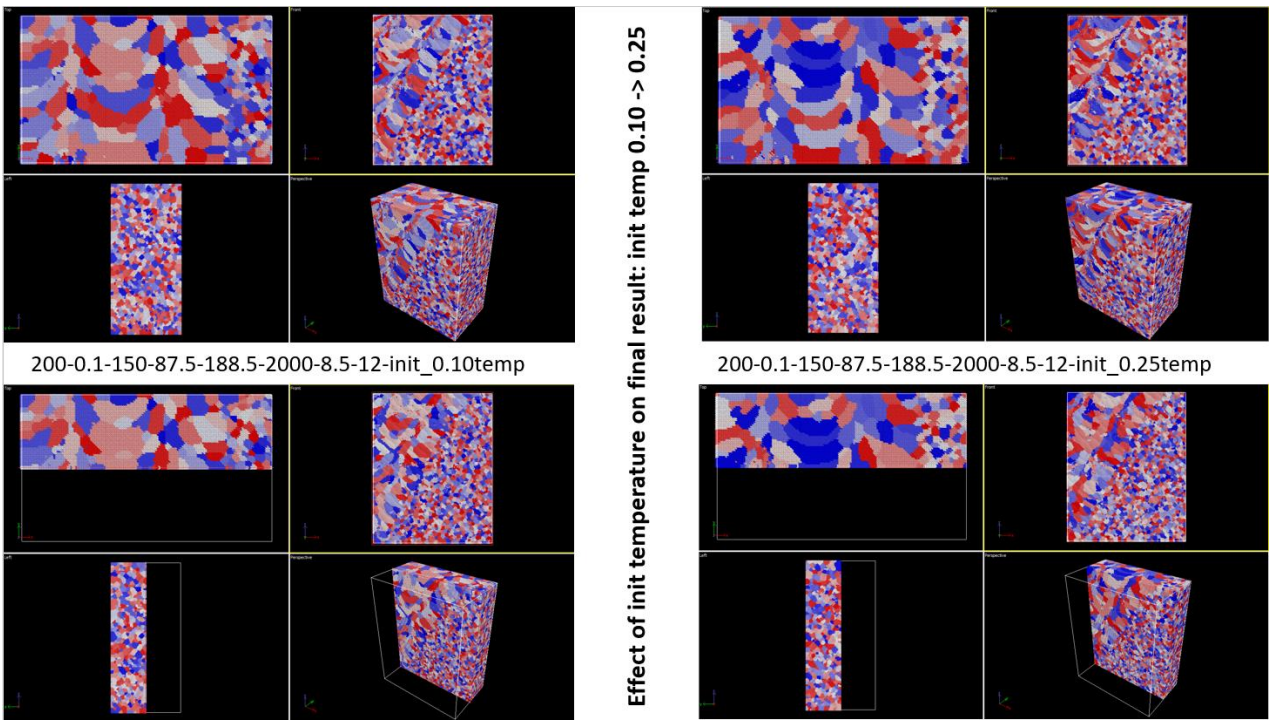


Figure 112: simulation results of the varied initial temperature scale

4.8.6 Effect of the HAZ size

Finally, at 0.30 scale, the effect of the HAZ width and the HAZ depth was also investigated. The original width of the HAZ was taken as 1, compared to these the HAZ value was lowered to 85% of the original value and to 120% of it. The results are shown in Figure 113 and Figure 114.

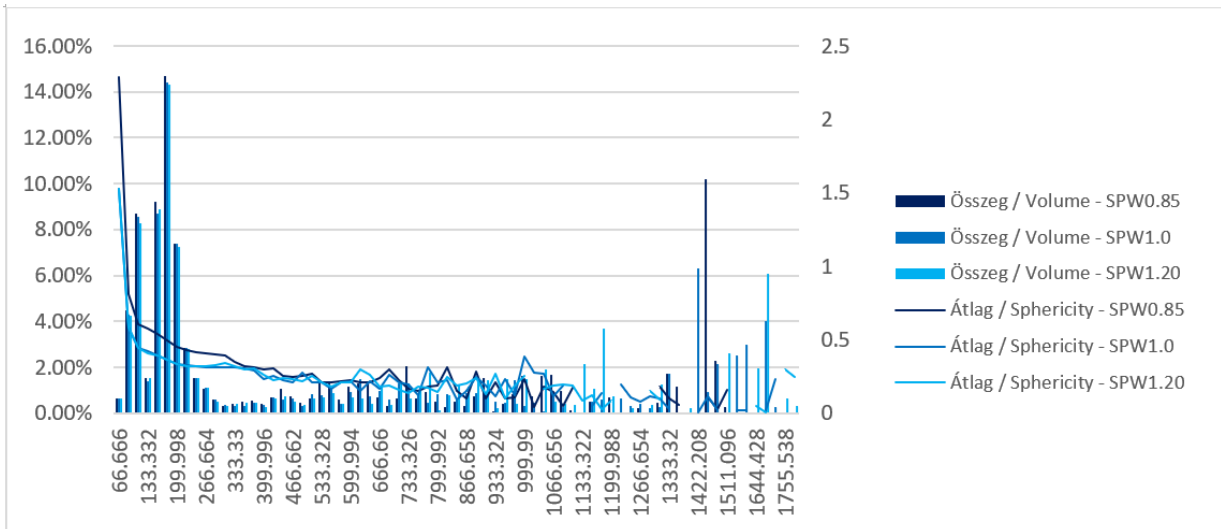


Figure 113: Simulation results at scale 0.30 with symmetry, effect of HAZ width varying (radius is not scaled)

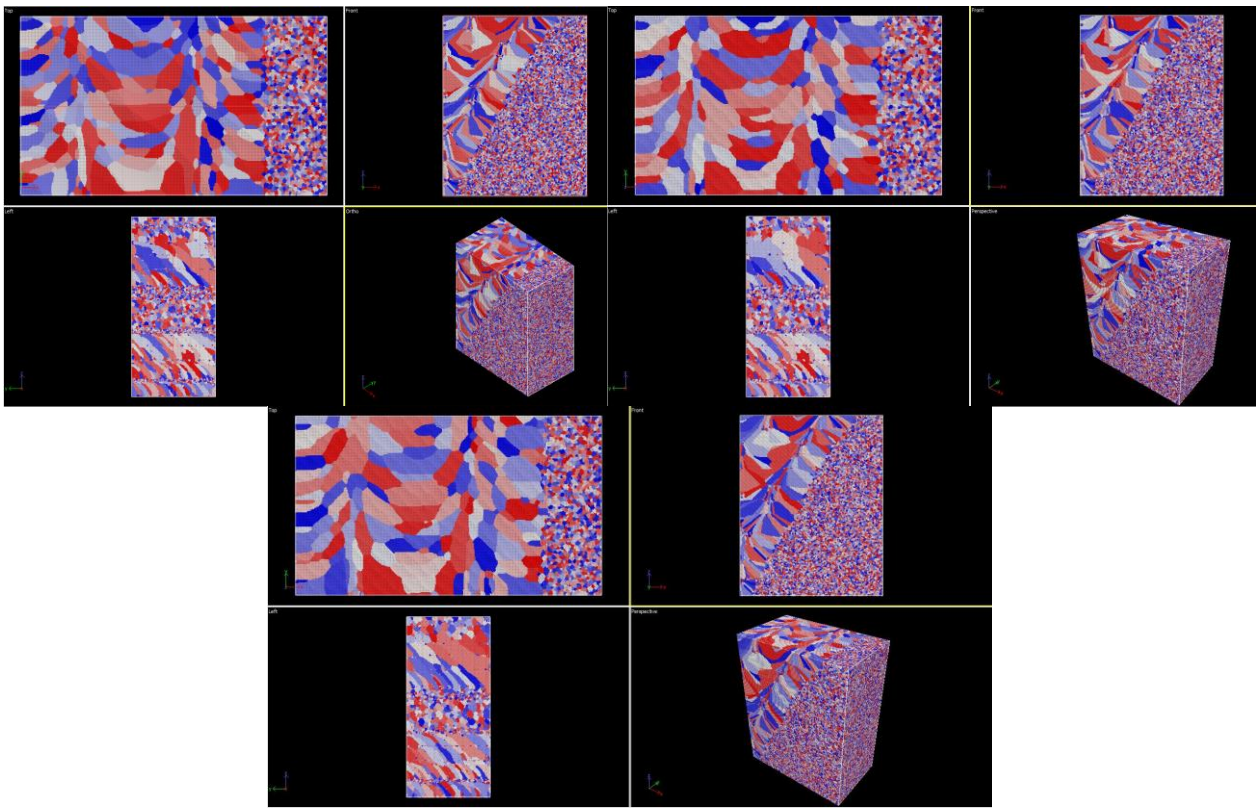


Figure 114: Effect of the width of the HAZ (×0.85, ×1.0, ×1.2)

The depth of the HAZ was also varied and beside the original values the 120% of them were also simulated. The simulation results are shown in Figure 115 and Figure 116. It can be seen that with the larger values (in case of the width and the depth too) more large grains are created, in case of the smaller grains the number of the created grains is similar.

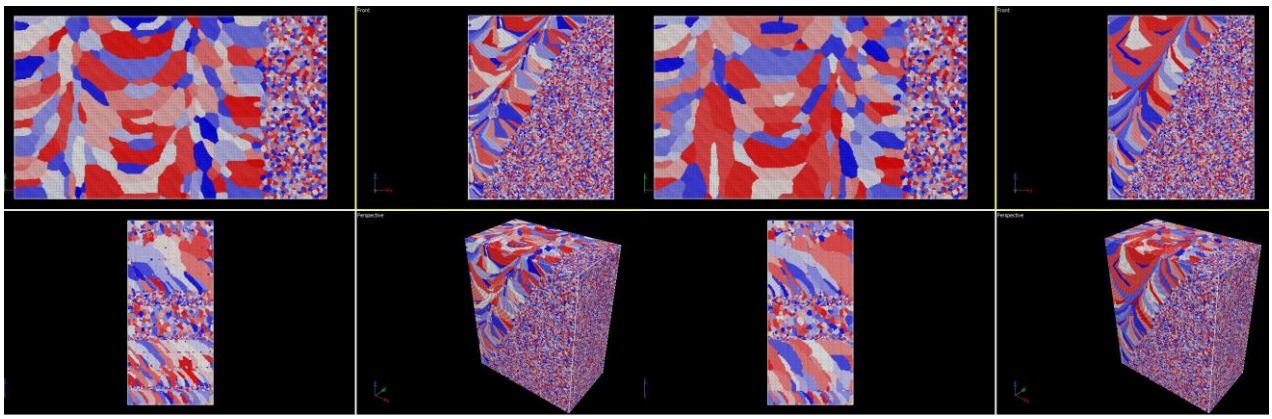


Figure 115: Effect of HAZ depth varying

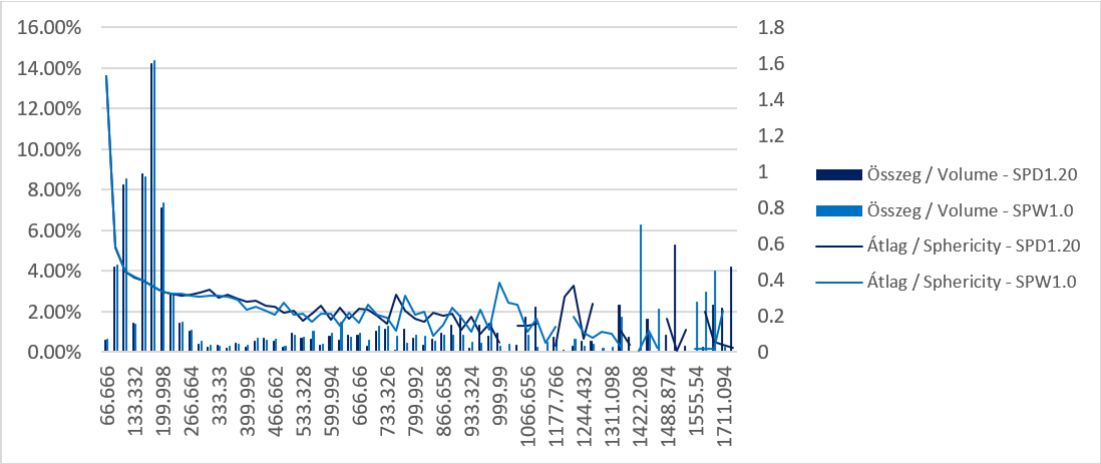


Figure 116: Simulation results at scale 0.30 with symmetry, effect of HAZ depth varying

4.9 Achievements and Conclusions

BZN used the SPPARKS code to develop a model which is able to simulate the formation of grains during a welding process, for this the potts/welds and the potts/additive applications were combined to create a simulation method for multilayer welding processes. In the created model the following parameters can be changed to receive different results:

- Simulation volume: Defines the number of atoms (sites), which affects the resolution of the output. As SPPARKS simulations are dimensionless the output needed to be rescaled or arranged to real scales.
- Grain initialization temperature: Increasing the initialization temperature to create the base matrix of weld increases the initialized grain size, resulting in bigger grains after welding simulation is completed, therefore the base grain size can be changed.
- Set different FZ and HAZ geometry until the results are in good correspondence with the real measurements: Several simulated weld results can be achieved by varying the geometry properties of FZ and HAZ, which have the greatest effect on the resulting weld grain structure. The input of these geometric parameters can be found in „in*." files.
- Weld temperature: similar to the initialization temperature, it affects the size of the grains.
- Change the simulation welding speed [sites/MCS]: The welding speed is defined in [sites/MCS] before the simulation. Based on the value the input script calculates the MCSs needed to simulate the actual layer. The defined speed is the other important factor that greatly influences the resulting grain structure. The lower the speed the greater the grain size will be.

The visualization of the result can be obtained in several programs, BZN mainly used OVITO for this, where the rendering of the sectional images can be created. Thereafter the image sequence can be loaded into ImageJ, where the MorpholibJ package can be used to convert the results into .csv file, which can later be analysed using Excel.

With the modified SPPARKS code by BZN, the SPPARKS user can build any kind of geometric one-directional multilayer weldments. This means the number of layers and the seams per layer, as well as their geometric positions (parameters) can be defined.

To achieve the results previously introduced, the user should follow the steps below:

- Take the weld technology parameters,
- Calculate the FZ and HAZ geometry,
- Input the calculated geometry properties into the input files per layer,
- BUT limit the HAZ depth in the simulation file to the maximum vertical length between layers,

- Input the simulation volume, the grain initialization temperature, and weld temperature, the number of layers, set the scale parameter,
- Set weld speed in sites/MCS.

With the previously listed steps, the users can build multilayer welding simulations with the help of SPPARKS, where they can investigate the microstructure of their own weld.

4.10 Discussion and Limitation

The implementations of the developed combined FE and SPPARKS model are shown to predict the grain structure upon the single linear pass and multiple linear passes GTAW and SMAW processes.

The present SPPARKS model can be used to predict the final grain structure during the fusion welding processes. The application of the present model can take into account different welding processes where epitaxial grain growth and growth competition take place in the parent metal as well as in the newly formed weld bead. The presented results suggest that the used SPPARKS model is very sensitive to the applied initial parameters. This can be seen in Figure 94-Figure 96 and Figure 108-Figure 109. The key parameters are identified above, but the uncertainty of these parameters is not analysed to find the ideal parameter for the WP2 sample. However, validation of this model needs to be developed in the future work.

As a follow up of the project, possible research areas could include the followings: the simulated results should be compared to corresponding experimental results on different austenitic alloys to find the ideal simulation parameters and to predict the realistic grain size and grain orientation which can be used for the modelling of ultrasound propagation.

5 Conclusion

In this work, simulation models have been developed for the formation of welds to cover the range of kinds of welds and welding processes such as SMAW, GTAW and SAW. The developed finite element models include generalised 3D capabilities to consider the variety of features encountered in real complex weld formations, such as dissimilar materials, asymmetric geometries, and buttering layers. The method which is presented before is easily applicable to model the grain formation during welding, but it is extendable for other processes like laser welding or casting simulation. The developed and presented FE modelling technique on macro-scale provides reasonable predictions for the solidification conditions, thereby gives a realistic stiffness tensor orientation and typical grain size and elongation.

On the other hand, BZN used the SPPARKS code to develop a model which can simulate the formation of grains during a welding process on a meso-scale. A novel simulation technique based on a modified Kinetic Monte Carlo Potts model has been developed to simulate the current welding sample of the processed microstructure. The model uses the shape of the molten zone from the FE thermal simulation, the surrounding temperature gradient, and the scan pattern to model microstructural evolution. The flexibility of the model provides, in addition to the minimum calculation requirements, an excellent technique for evaluating the grain structure associated with the use of the processing parameter of the current welding process. The validation of this SPPARKS model needs to be developed in a future work.

6 Bibliography

[1] W. Kurz, B. Giovanola, R. Trivedi, Theory of microstructural development during rapid solidification, Acta Metallurgica 34 pp. 823–830, 1986.

[2] M. H. Burden, J. D. Hunt, Cellular and dendritic growth. II, Journal of Crystal Growth 22 (2), pp. 109–116, 1974.

[3] G. Duggan, M. Tong, D. J. Browne, An integrated meso-scale numerical model of melting and solidification in laser welding, IOP Conference Series: Materials Science and Engineering 27, 2012.

[4] T. Koseki, H. Inoue, Y. Fukuda, A. Nogami, Numerical simulation of equiaxed grain formation in weld solidification, Science and Technology of Advanced Materials 4 pp. 183–195, 2003.

[5] D. J. Browne, J. D. Hunt, A fixed grid front-tracking model of the growth of a columnar front and an equiaxed grain during solidification of an alloy, Numerical Heat Transfer, Part B: Fundamentals 45 pp. 395–419, 2004.

[6] Toloui M, Militzer M. Phase Field Modelling of Microstructure Evolution in the HAZ of X80 Linepipe Steel. ASME. International Pipeline Conference, Volume 3: Materials and Joining pp. 323-328, 2012.

[7] Poole WJ, Militzer MM, Garcin TT. An Integrated Model to Predict Microstructure and Mechanical Properties in the Heat Affected Zone for X80 Linepipe. ASME. International Pipeline Conference, Volume 3: Materials and Joining pp. 301-306, 2012.

[8] Tomohiro Takaki, Toshimichi Fukuoka, Yoshihiro Tomita, Phase-field simulation during directional solidification of a binary alloy using adaptive finite element method, Journal of Crystal Growth, Volume 283, pp 263-278, 2005.

[9] Wajira U. Mirihanage, David J. Browne, Combined analytical/numerical modelling of nucleation and growth during equiaxed solidification under the influence of thermal convection, Computational Materials Science, Volume 46, Issue 4, pp 777-784, 2009.

[10] Pavlyk, Vitaliy; Dilthey, Ulrich, Simulation of weld solidification microstructure and its coupling to the macroscopic heat and fluid flow modelling, Modelling and Simulation in Materials Science and Engineering, Volume 12, Issue 1, pp. S33-S45, 2004.

[11] T. M. Rodgers, J. A. Mitchell, and V. Tikare, T. M. Rodgers, J. A. Mitchell, V. Tikare: A Monte Carlo model for 3D grain evolution during welding, Modelling and Simulation in Materials Science and Engineering, 25, 2017.

[12] T. M. Rodgers, J. D. Madison, V. Tikare, Simulation of metal additive manufacturing microstructures using kinetic Monte Carlo, Computational Materials Science 135, pp. 78-89, 2017.

[13] E. R. Homer, V. Tikare, E. A. Holm, Hybrid Potts-phase field model for coupled microstructural-compositional evolution, Computational Materials Science, 69, 414-423, 2013.

[14] Xiz C, Xin C, Xu H, Madigan B, Monte Carlo simulation and experimental measurements of grain growth in the heat affected zone of 304 stainless steel during multipass welding. Int J Adv Manuf Technol 80:1197–121, 2015.

[15] Sista S, Yang Z, DebRoy T, Three-dimensional Monte Carlo simulation of grain growth in the heat-affected zone of 2.25Cr-1Mo steel weld. Metall Mater Trans B 31B:529–536, 2000.

[16] H Granjon, Fundamentals of Welding Metallurgy, Woodhead Publishing, 1991.

[17] ASM Handbook Volume 6 Welding Brazing and Soldering

[18] https://www.esab.hu/hu/hu/products/filler-metals/covered-stick-electrodes-smaw/stainless-steel-electrodes/ok-61-30.cfm

[19] P J Bouchard, Validated residual stress profiles for fracture assessments of stainless steel pipe girth welds, Int Jnl Press Vess and Piping 84(4), 195-222 (2007)

[20] J. Goldak, A. Chakravarti, M. Bibbya New Finite Element Model for Welding Heat Sources Metallurgical Transactions B, (1984) 15: 299. https://doi.org/10.1007/BF02667333

[21] Moysan J., Apfel A., Corneloup G, Chassignole B, . Modelling the grain orientation of austenitic stainless steel multipass welds to improve ultrasonic assessment of structural integrity. Int J Pressure Vessels Piping 80:77-85 , (2003)

[22] Anderson, I.; Grong, O. Analytical modelling of grain growth in metals and alloys in the presence of growing and dissolving precipitates—II. Abnormal grain growth. Acta Metall. (1995), 43, 2673–2688

[23] Bernhard, C., J. Reiter und H. Preßlinger: A model for predicting the austenite grain size at the surface of continuously cast slabs. Met. Trans. B, 39 (2008), 885–895

[24] Wang, Y.; Ding, M.; Zheng, Y.; Liu, S.; Wang, W.; Zhang, Z. Finite-Element Thermal Analysis and Grain Growth Behavior of HAZ on Argon Tungsten-Arc Welding of 443 Stainless Steel. Metals (2016), 6, 77

[25] D. Rosenthal, Mathematical theory of heat distribution during welding and cutting, Weld Journal 20 220-234 (1941)

[26] V. Pavelic, Experimental and computed temperature histories in gas tungsten arc welding of thin plates, Weld Journal 48 295-305 (1969)

[27] R. Ducharme et al., The laser welding of thin metal sheets – an integrated keyhole and weld pool model with supporting experiments, Journal of Physics D: Applied Physics 27(8) 1619-1627 (1994)

[28] A. Okada, T. Kasugai, K. Hiraoka, Heat-source model in arc-welding and evaluation of weld heat-effected zone, The Iron and Steel Institute of Japan 28(10) 876-882 (1988)

[29] M. A. Wahab, M. J. Painter, Numerical models of gas metal arc welds using experimentally determined weld pool shapes as the representation od the welding heat source, International Journal of Pressure Vessels and Piping 73(2) 153-159 (1997)

[30] M. A. Wahab, M. J. Painter, M. H. Davies, The prediction of the temperature distribution and weld pool geometry in the gas metal arc welding process. Journal of Materials Processing Technology 77(1-3) 233-239 (1998)

[31] W. Sudnik, D. Radaj, W. Erofeew, Computerized simulation of laser beam welding, modelling and verification, Journal of Physics D: Applied Physics 29(11) 2811-2817 (1996)

[32] S. S. Babu, S. A. David, J. M. Vitek, R. W. Reed, Solidification and microstructure modelling of welds in aluminium alloys 5754 and 6111, Science and Technology of Welding and Joining 6(1) 31-40 (2001)

[33] A. Mahrle, J. Schmidt, D. Weiss, Simulation of temperature fields in arc and beam welding, Heat Mass Transfer 36(2) 117-126 (2000)

[34] T. Fuhrich, P. Berger, H. Hügel, Numerical calculation of the weld pool in deep penetration laser welding, Numerical Analysis of Weldability 6 (2001)

[35] H. Ki, P. S. Mohanty, J. Mazumder, Modeling of laser keyhole welding: Part II. simulation of keyhole evolution, velocity, temperature profile, and experimental verification, Metallurgical and Materials Transactions 33(6) 1831-1842 (2002)

[36] Z. Yang, T. Debroy, Modeling macro- and microstructures of gas-metal-arc welded HSLA-100 steel, Metallurgical and Materials Transactions B 30(3) 483-493 (1993)

[37] M. Stadtaus, V. Michailov, H. Wohlfahrt, Simulation of weld shape, microstructure, distortion and residual stresses for laser beam welding, Numerical Analysis of Weldability, 6 (2001)

[38] P. van der Wolk, Modelling CCT-diagrams of engineering steels using neural networks,Phd Technical University of Delft (2001)

[39] C. Zener, Theory of growth of spherical precipitates from solid solution, J App Phys, 20 950-953 (1949)

[40] J. Svoboda, F. D. Fischer, P. Fratzl, E. Gamsjager, N. K. Simha, Kinetics of interfaces during diffusional transformations, Acta Materialia 49(7) 1249-1259 (2001)

[41] J. Sietsma, S. van der Zwag, A concise model for mixed-mode phase transformations in the solid state, Acta Materialia 52(14) 4143-4152 (2004)

[42] I. Katzarov, S. Malinov, W. Sha, Finite element modelling of the morphology of beta to alpha phase transformation in Ti-6Al-4V alloy, Metallurgical and Materials Transactions A 33(4) 1027-1040 (2002)

[43] H. Hemmer, O. Grong, A process model for the heat-affected zone microstructure evolution in duplex stainless steel weldments: Part 1. The Model, Metallurgical and Materials Transactions A 30(11) 2915-2929 (1999)

[44] O. R. Myhr, O. Grong, S. Klokkehaug, H.G. Fjaer, Modelling of the microstructure and strength evolution during ageing and welding of Al-Mg-Si alloys, Numerical Analysis of Weldability – 6 (2001)

[45] S. Sista, T. Debroy, Three-dimensional Monte Carlo simulation of grain growth in zone-refined iron, Metallurgical and Materials Transactions B 32(6) 1195-1201 (2001)

[46] Z. Yang, S. Sista, J. W. Elmer, T. Debroy, Three dimensional Monte Carlo simulation of grain growth during GTA welding of titanium. Acta Materialia 48(20) 4813-4825 (2000)

[47] Y. W. Shi, D. Chen, Y. P. Lei, X.Y. Li, HAZ microstructure simulation in welding of an ultra fine grain steel, Computational Materials Science, 31(3-4) 379-388 (2004)

[48] M. Kumar, R. Sasikumar, P. K. Nair, Competition between nucleation and early growth of ferrite from austenite – studies using cellular automaton simulations, Acta Materialia 46(17) 6291-6303 (1998)

[49] A. Slepoy, A. P. Thompson and S. J. Plimpton, A Constant-Time Kinetic Monte Carlo Algorithm for Simulation of Large Biochemical Reaction Networks, 2008.

[50] S. Plimpton, C. Battaile, M. Chandross, L. Holm, A. Thompson, V. Tikare, G. Wagner, E. Webb, X. Zhou, C. G. Cardona and A. Slepoy, Crossing the Mesoscale No-Man's Land via Parallel Kinetic Monte Carlo, 2009.

[51] Sandia National Labratories, "SPPARKS www page," 12 2020. [Online]. Available: https://spparks.sandia.gov/index.html.

[52] Theron M. Rodgersa, John A. Mitchella and Veena Tikar, A Monte Carlo model for 3D grain evolution, SAND2017-8231J.

[53] T. M. Rodgers, J. D. Madison, V. Tikare, M. C. Maguire, Predicting Mesoscale Microstructural Evolution in Electron Beam Welding, JOM 68, 1419–1426 (2016)

[54] “OVITO,” OVITO Gmbh, [Online]. Available: https://www.ovito.org/. [Accessed 05 12 2020].

[55] A. Stukowski, “Visualization and analysis of atomistic simulation data with OVITO–the Open Visualization Tool,” Modelling and Simulation in Materials Science and Engineering, 2009.

[56] Dream3D, “DREAM.3D,” ARFL, NRL, [Online]. Available: http://dream3d.bluequartz.net/. [Accessed 05 12 2020].

[57] M. A. Groeber and M. A. Jackson, “DREAM.3D: A Digital Representation Environment for the Analysis of Microstructure in 3D,” Integrating Materials and Manufacturing Innovation, 2014.

[58] S. Chen, G. Guillemot, C. A. Gandin, 3D Coupled Cellular Automaton (CA){Finite Element (FE) Modeling for Solidification Grain Structures in Gas Tungsten Arc Welding (GTAW), ISIJ International 54 401-407 (2014)

7 Annex

7.1 Appendix A

Modified code of "app_potts_additive.h"

```
/* ----------------------------------------------------------------------
   SPPARKS - Stochastic Parallel PARticle Kinetic Simulator
   http://www.cs.sandia.gov/~sjplimp/spparks.html
   Steve Plimpton, sjplimp@sandia.gov, Sandia National Laboratories

   Copyright (2008) Sandia Corporation.  Under the terms of Contract
   DE-AC04-94AL85000 with Sandia Corporation, the U.S. Government retains
   certain rights in this software.  This software is distributed under
   the GNU General Public License.

   See the README file in the top-level SPPARKS directory.
------------------------------------------------------------------------- */

#ifdef APP_CLASS
AppStyle(potts/additive,AppPottsAdditive)

#else

#ifndef SPK_APP_POTTS_ADDITIVE
#define SPK_APP_POTTS_ADDITIVE

#include "app_potts.h"
#include "am_raster.h"
#include <stdlib.h>
#include <map>

using std::map;
using RASTER::Pass;
using RASTER::TransversePass;
using RASTER::RectangularLayer;
using RASTER::Pattern;

namespace SPPARKS_NS {

class AppPottsAdditive : public AppPotts {
 public:
  AppPottsAdditive(class SPPARKS *, int, char **);
  virtual void grow_app();
  virtual void init_app();
  virtual void site_event_rejection(int, RandomPark *);
  void input_app(char *, int , char **);
  double compute_mobility(int, double, double);
  void app_update(double);

 //Remove all of the variables we don't actually need
 protected:
  int NXeffective;
  int NYeffective;
  int NZeffective;
  int CurrLayer;
  int CurrPass;

  double *MobilityOut;

 double spot_width;
```

```
 double melt_tail_length;
 double tail_HAZ;
 double exp_factor;
 double cap_height;
 double HAZ;
 double melt_depth;
 double cap_HAZ;
 double depth_HAZ;

 private:
   map<int,Pass> passes;
   map<int,TransversePass> transverse_passes;
   map<int,RectangularLayer> rectangular_layers;
   Pattern pattern;
   RectangularLayer active_layer;
   double *distance;
   double q;
};

}

#endif
#endif

/* ERROR/WARNING messages:

E: Illegal ... command

Self-explanatory.  Check the input script syntax and compare to the
documentation for the command.  You can use -echo screen as a
command-line option when running SPPARKS to see the offending
line.

E: One or more sites have invalid values

The application only allows sites to be initialized with specific
values.

*/
```

7.2 Appendix B

Modified code of "app_potts_additive.cpp"

```
/* ----------------------------------------------------------------------
   SPPARKS - Stochastic Parallel PARticle Kinetic Simulator
   http://www.cs.sandia.gov/~sjplimp/spparks.html
   Steve Plimpton, sjplimp@sandia.gov, Sandia National Laboratories

   Copyright (2008) Sandia Corporation.  Under the terms of Contract
   DE-AC04-94AL85000 with Sandia Corporation, the U.S. Government retains
   certain rights in this software.  This software is distributed under
   the GNU General Public License.

   See the README file in the top-level SPPARKS directory.
------------------------------------------------------------------------- */

/* ----------------------------------------------------------------------
   Contributing author: Theron Rodgers and John Mitchell (Sandia)
------------------------------------------------------------------------- */

#include "string.h"
#include "math.h"
#include "app_potts_additive.h"
#include "random_park.h"
#include "solve.h"
#include "lattice.h"
#include "domain.h"
#include "am_ellipsoid.h"
#include "error.h"
#include <vector>
#include <algorithm>

#include <iostream>

using namespace SPPARKS_NS;
using RASTER::point;
using RASTER::DIR;

/* ---------------------------------------------------------------------- */

AppPottsAdditive::AppPottsAdditive(SPPARKS *spk, int narg, char **arg) :
  AppPotts(spk,narg,arg), passes(), active_layer(), distance(nullptr) {

   // only error check for this class, not derived classes
   if (strcmp(arg[0],"additive") == 0 && narg != 11 )
    error->all(FLERR,"Illegal app_style command");

   nspins = atoi(arg[1]); //Number of spins
   spot_width = atof(arg[2]); //Width of the melt pool
   melt_tail_length = atof(arg[3]); //Length of tail from meltpool midpoint
   melt_depth = atof(arg[4]); //How many lattice sites deep the melt pool is
   cap_height = atof(arg[5]); //Height of the cap leading the meltpool
   HAZ = atof(arg[6]); //Size of the HAZ surrounding the melt pool (must be
larger than spot_width)
   tail_HAZ = atof(arg[7]); //Length of hot zone behind meltpool (must be
larger than melt_tail_length)
   depth_HAZ = atof(arg[8]); //Depth of the hot zone underneath the meltpool
(must be larger than melt_depth)
   cap_HAZ = atof(arg[9]); //Size of HAZ infront of the melt pool (must be
larger than cap_height)
```

```
   exp_factor = atof(arg[10]); //Exponential parameter for mobility decay in
haz M(d) = exp(-exp_factor * d, however in the modified code it is not used
later

   //Define the layer object, this might work better in init_app
   ndouble = 1;
   allow_app_update = 1;
   recreate_arrays();
}

/* ----------------------------------------------------------------------
   Define additional input commands for the AM app
------------------------------------------------------------------------- */

void AppPottsAdditive::input_app(char *command, int narg, char **arg)
{
   if (strcmp(command,"am_pass") == 0) {
      if (narg != 7) error->all(FLERR,"Illegal pass command.");
      char* dir=nullptr;
      double distance=-1.0;
      double speed=-1.0;
      int id=std::atoi(arg[0]);
      if(strcmp(arg[1],"dir")==0){
         dir=arg[2];
      } else {error->all(FLERR,"Illegal pass command. Expected 'dir.'");}
      if(strcmp(arg[3],"distance")==0){
         distance=std::atof(arg[4]);
      } else {error->all(FLERR,"Illegal pass command. Expected
'distance.'");}
      if(strcmp(arg[5],"speed")==0){
         speed=std::atof(arg[6]);
      } else {error->all(FLERR,"Illegal pass command. Expected 'speed.'");}
      DIR d;
      if(strcmp(dir,"X")==0) d=DIR::X;
      else if(strcmp(dir,"Y")==0) d=DIR::Y;
      else {error->all(FLERR,"Illegal pass 'dir' command. Expected 'X|Y.'");}
      passes[id]=Pass(d,distance,speed);

   } else if (strcmp(command,"am_transverse_pass") == 0) {
      if (narg != 5) error->all(FLERR,"Illegal transverse_pass command.");
      double distance=-1.0;
      double increment=-1.0;
      int id=std::atoi(arg[0]);
      if(strcmp(arg[1],"distance")==0){
         distance=std::atof(arg[2]);
      } else {error->all(FLERR,"Illegal transverse_pass command. Expected
'distance.'");}
      if(strcmp(arg[3],"increment")==0){
         increment=std::atof(arg[4]);
      } else {error->all(FLERR,"Illegal transverse_pass command. Expected
'increment.'");}
      transverse_passes[id]=TransversePass(distance,increment);

   } else if (strcmp(command,"am_cartesian_layer") == 0) {
      if (narg != 10) error->all(FLERR,"Illegal cartesian_layer command.");
      double x=0.0, y=0.0;
      int pass_id=-1, transverse_pass_id=-1;
      bool serpentine=false;
      int id=std::atoi(arg[0]);
      if(strcmp(arg[1],"start_position")==0){
         x=std::atof(arg[2]);
         y=std::atof(arg[3]);
```

```
      } else {error->all(FLERR,"Illegal cartesian_layer command. Expected
'start_position.'");}
      if(strcmp(arg[4],"pass_id")==0){
         pass_id=std::atoi(arg[5]);
      } else {error->all(FLERR,"Illegal cartesian_layer command. Expected
'pass_id.'");}
      if(strcmp(arg[6],"transverse_pass_id")==0){
         transverse_pass_id=std::atoi(arg[7]);
      } else {error->all(FLERR,"Illegal cartesian_layer command. Expected
'transverse_pass_id.'");}
      if(strcmp(arg[8],"serpentine")==0){
         //Parsing this as boolean wasn't working, temporarily changed to
integer
         serpentine=std::atoi(arg[9]);
      } else {error->all(FLERR,"Illegal cartesian_layer command. Expected
'serpentine.'");}
      {
         // Create 'RectangularLayer'
         point start(x,y,0);
         Pass p=passes[pass_id];
         DIR dir=p.get_dir();
         double speed=p.get_speed();
         double pass_distance=p.get_distance();
         //Define the "overpass", which will be determined by tail_HAZ +
cap_height
         double overpass = tail_HAZ + cap_HAZ;
         TransversePass tp=transverse_passes[transverse_pass_id];
         double transverse_pass_distance=tp.get_distance();
         double transverse_pass_increment=tp.get_increment();

rectangular_layers[id]=RectangularLayer(start,dir,speed,pass_distance,overpas
s,transverse_pass_distance,transverse_pass_increment,serpentine);
      }

   } else if (strcmp(command,"am_pattern") == 0) {
      int num_layers;
      vector<int> layer_ids;
      double z_start=-1.0;
      double z_increment=-1.0;
      int id=std::atoi(arg[0]);
      if(strcmp(arg[1],"num_layers")==0){
         num_layers=std::atoi(arg[2]);
      } else {error->all(FLERR,"Illegal pattern command. Expected
'num_layers.'");}
      if(strcmp(arg[3],"layer_ids")==0){
      } else {error->all(FLERR,"Illegal pattern command. Expected
'layer_ids.'");}
      int num_args=1+2+1+num_layers+2+2;
      if (narg != num_args) error->all(FLERR,"Illegal pattern command.");
      int iarg=4;
      for(int i=0;i<num_layers;i++,iarg++){
         layer_ids.push_back(std::atoi(arg[iarg]));
      }
      if(strcmp(arg[iarg],"z_start")==0){
         iarg+=1;
         z_start=std::atof(arg[iarg]);
         iarg+=1;
      } else {error->all(FLERR,"Illegal pattern command. Expected
'z_start.'");}
      if(strcmp(arg[iarg],"z_increment")==0){
         iarg+=1;
         z_increment=std::atof(arg[iarg]);
```

```
          iarg+=1;
      } else {error->all(FLERR,"Illegal pattern command. Expected
'z_increment.'");}
      pattern=Pattern(layer_ids,z_start,z_increment);

   } else error->all(FLERR,"Unrecognized command");
}

/* ----------------------------------------------------------------------
   set site value ptrs each time iarray/darray are reallocated
---------------------------------------------------------------------- */

void AppPottsAdditive::grow_app()
{
  spin = iarray[0];
  MobilityOut = darray[0];
  //distance block was given by BZN it is necessary for the
site_event_rejection function.
    distance = darray[0];
}

/* ----------------------------------------------------------------------
   initialize before each run
   check validity of site values
---------------------------------------------------------------------- */

void AppPottsAdditive::init_app()
{
   delete [] sites;
   delete [] unique;
   sites = new int[1 + maxneigh];
   unique = new int[1 + maxneigh];

   dt_sweep = 1.0/maxneigh;

   int flag = 0;
   for (int i = 0; i < nlocal; i++)
    if (spin[i] < 1 || spin[i] > nspins) flag = 1;
   int flagall;
   MPI_Allreduce(&flag,&flagall,1,MPI_INT,MPI_SUM,world);
   if (flagall) error->all(FLERR,"One or more sites have invalid values");

   int next_layer_id=pattern.begin();
   active_layer=rectangular_layers[next_layer_id];

   // Compute distance function based upon initial pool position
   app_update(0.0);
}

/* ----------------------------------------------------------------------
    Use app update to set the new position of the weld pool and determine the
    mobilities for the new configuration
 ---------------------------------------------------------------------- */

void AppPottsAdditive::app_update(double dt)
{
   // Move pool
   if(active_layer.move(dt)){
   } else {
      // Need to update layer
      int next_layer_id=pattern.next();
      active_layer=rectangular_layers[next_layer_id];
```

```
    }
    // WARNING: this should always be run after checking on a move;
    // z-elevation of active layer
    double layer_z=pattern.get_layer_z_elevation();

     //Use the new position as input to the mobility calculation
     //Loop through all of the local sites and assign the new mobilities

     //Specify the shape of the melt pool and then calculate the distance at
each local site.
     RASTER::pool_shape::AmEllipsoid ae(spot_width, melt_depth,
melt_tail_length, cap_height, HAZ, tail_HAZ);

     //Go through all the local sites and calculate the distance.
     double d;
     q = 0;

     for(int i=0;i<nlocal;i++){
         // SPPARKS lattice site
         double XYZ[]={xyz[i][0],xyz[i][1],xyz[i][2]};
         // Lattice point location relative to 'pool' position
         point
xyz_r_p=active_layer.compute_position_relative_to_pool(XYZ,layer_z);

         //Temporary assignment of xo, xo is in the melt pool's reference
frame!
         double xo[]={xyz_r_p[0],xyz_r_p[1],xyz_r_p[2]};


         if(xo[0] < 0 && xo[2] <= 0 && abs(xo[2]) <= depth_HAZ  && xo[0] > -
tail_HAZ && abs(xo[1]) <= HAZ/2.0) {

             //If we're in the fusion zone, calculate distance
             if (abs(xo[1]) <= spot_width * 0.5 && abs(xo[0]) <= tail_HAZ) {
                 d = ae.distance(xo);

                 distance[i]=d;
             }
             //If we're in the HAZ, calculate distance
             else if (abs(xo[1]) <= HAZ/2.0 && abs(xo[0]) <tail_HAZ) {
                 d = ae.distance(xo);
                 distance[i]=d;

             }
         }
         //If we're in front of the pool, look out to a distance cap_HAZ away
         else if (abs(xo[0]) <= cap_HAZ && xo[2] <=0 && abs(xo[2]) <=
depth_HAZ && abs(xo[1]) <= HAZ/2.0) {
             d = ae.distance(xo);
             distance[i]=d;
         }

         //In case of potts/weld application if we are not in the Fusion Zone
then the program calculates the distance for every point:

         else {
             d = -10;
             distance[i]=d;
         }

         if (distance[i] > q) {
             q = distance[i];
```

```
        }
    }



    for(int i=0;i<nlocal;i++){

        //Only calculate mobilities for things inside the HAZ bounds and
below the active layer
        if (distance[i] >= 0) {
            MobilityOut[i] =  compute_mobility(i, distance[i], q);
        }
        //Inside the pool so set mobilty to 1 (which randomizes things)
        else if (distance[i] > -5) {

            MobilityOut[i] = 1;
        }
        //If we're outside the region of interest, make Mobility zero
        else {
            MobilityOut[i] = 0;
        }

    }
   active_layer.move(dt);
}


/* ----------------------------------------------------------------------
 Compute the mobility at the specified lattice site. Returns a double
 between 0 and 1 representing the mobility.
 ---------------------------------------------------------------------- */

double AppPottsAdditive::compute_mobility(int site, double d, double q)  {

    //We're going to take care of categorizing all the little details of the
mobility
    //gradient in app_update, so here we'll just calculate the mobility based
on distance
    /*int n = sizeof(distance) / sizeof(distance[0]);
    double d_max = *std::max_element(&distance[0],&distance[n-1]);*/
    MobilityOut[site] = 1-d/(q/4);

    return MobilityOut[site];
}

/* ----------------------------------------------------------------------
   rKMC method
   perform a site event with no null bin rejection
   flip to random neighbor spin without null bin
   technically this is an incorrect rejection-KMC algorithm
---------------------------------------------------------------------- */
void AppPottsAdditive::site_event_rejection(int site, RandomPark *random) {
   int oldstate = spin[site];
   double einitial = site_energy(site);

   if (MobilityOut[site] < 0.0) {
      MobilityOut[site] = 0.0;
      return;
   }

   if(MobilityOut[site] >= 1.0){
      //Mobility = 0.0;
```

```
        spin[site] = (int) (nspins*random->uniform());
        return;
    }

    // events = spin flips to neighboring site different than self
    int j,m,value;
    int nevent = 0;
    int z = xyz[site][2];

    if(MobilityOut[site] > 0.0){ //&& (MobilityOut[site] < 1.0)) {
        //(spin[i] != nspins) another criteria to exclude gg interaction
        for (j = 0; j < numneigh[site]; j++) {
            value = spin[neighbor[site][j]];
            double j_distance = distance[neighbor[site][j]];
            /*if (value == spin[site] || value == nspins) continue;
                for (m = 0; m < nevent; m++)
                    if (value == unique[m]) break;
                if (m < nevent) continue;*/
            if (value == spin[site] || j_distance < 0.0) continue;
            unique[nevent++] = value;
        }

        if (nevent == 0) return;
        int iran = (int) (nevent*random->uniform());
        if (iran >= nevent) iran = nevent-1;
            spin[site] = unique[iran];
        double efinal = site_energy(site);

        // accept or reject via Boltzmann criterion
        if (efinal <= einitial) {
            if (random->uniform() > MobilityOut[site]){
                spin[site] = oldstate;
            }
        } else if (temperature == 0.0) {
            spin[site] = oldstate;
        } else if (random->uniform() > MobilityOut[site] * exp((einitial-
efinal)*t_inverse)) {
            spin[site] = oldstate;
        }

        if (spin[site] != oldstate) naccept++;

    }
}
```

7.4 Appendix C

Initalization script file, 'main_init.potts_additive_weld'

```
seed          5678910

variable     scale equal 0.1
variable     c equal 20.0/${scale}

variable     L_x equal    47*1000/${c}   #50
variable     W_y equal    25*1000/${c}     #40
variable     H_z_0    equal    21*1000/${c} #37.7
variable     H_z_1    equal    67*1000/${c} #37.7
variable     H_z equal    88*1000/${c} #37.7
variable     nspins_const     equal    ${L_x}*${W_y}*${H_z}

app_style  potts/neighonly ${nspins_const}

dimension     3

# periodic x
# not periodic y
# not periodic z
boundary p n n

# Number of microns per lattice site is approximately 20
lattice       sc/26n 1.0
region        box block -108 293 0 125 0 200
create_box    box
create_sites      box

set site range 1 ${nspins_const}

sweep         raster
# solve_style tree
sector yes

diag_style        energy

temperature   0.10

stats             1.0

# Write 'dump' files; see 'dump' documentation
dump          1 text 1.0 potts_init.dump
dump_modify   1 delay 1.0
diag_style    cluster stats no delay 1.0 delt 1.0 filename potts_init.cluster

# Write 'image'; see 'dump' documentation
#     images written every second
dump top image 1 potts_init.top.*.jpg site site crange 1  3500000 drange 1 1
view 0.0 0.0 boundary site 1 shape cube box no 1 zoom 2.0 size 1024 1024
sdiam 1.05
dump_modify top cwrap yes boundcolor black backcolor black pad 4

run             1.0
```

7.5 Appendix D

Main script file, 'main.potts_additive_weld'

```
# SPPARKS potts/additive test file


# Firstly the previously introduced random lattice site has to be created
with the following command:
# # mpirun -np 4 spk_mpi < main_init.potts_additive_weld
# or if the work is done on custler:
# mpiexec -f /home/john/hosts -n 36 /home/john/new3/spk_mpi <
/home/john/new3/main_init.potts_additive_weld
# Thereafter in the terminal the following command has to be made:
# ./write_site_init.sh potts_init.dump
# And finally it has to be ran with the following:
# mpiexec -f /home/john/hosts -n 36 /home/john/new3/spk_mpi <
/home/john/new3/main.potts_additive_weld -var inarg1 2000 -var inarg2 8.5

variable    a index 12 #here the number of the layers is given

label   loop2

variable    seed_const  equal   123456

variable    scale equal 0.1
variable    c equal 20.0/${scale}

variable    L_x equal   47*1000/${c}  #Length of the box in x [sites]
variable    W_y equal   25*1000/${c}    # Length of the box in y [sites]
variable    H_z_0   equal   21*1000/${c} #37.7
variable    H_z_1   equal   67*1000/${c} #37.7
variable    H_z equal   88*1000/${c} # Length of the box in z [sites]

variable    nspins_const    equal   ${L_x}*${W_y}*${H_z}

#[micron/s]
variable    s1_weldspeedr   equal   ${inarg1} #this and inarg2 is given the
following way: mpiexec -f /home/john/hosts -n 36 /home/john/new3/spk_mpi <
/home/john/new3/main.potts_additive_weld -var inarg1 2000 -var inarg2 8.5


variable    s1_weldspeeds   equal   ${inarg2}
variable    s1_weldspeedy   equal   ${s1_weldspeeds}
variable    t_stats equal   1.0

#dumpfreq
variable    dumpconst       equal   1

#-------------------------------------------

#if ${a} == 1 then "jump main_init.potts_additive_weld"

#label siteinit_back

jump in${a}.potts_additive_weld

label   loop
next a
jump main.potts_additive_weld loop2
```

7.6 Appendix E

'In*.' script file, 'in*.potts_additive_weld'

```
#---
#The following values have to be rewritten in the in?. files for every
welding row:
#---

variable    s1_colnums  equal   1
variable    s1_spw  equal   5.4*1000/${c}
variable    s1_spwhaz   equal   10.4*1000/${c}*1.0
variable    s1_spd  equal   3.5*1000/${c}
variable    s1_spdhaz   equal   5*1000/${c}*1.2
variable    s1_spt  equal   6.2*1000/${c}
variable    s1_spthaz   equal   ${s1_spt}*1.5
variable    s1_spc  equal   2.2*1000/${c}
variable    s1_spchaz   equal   3.7*1000/${c}
variable    s1_startposx    equal   0*1000/${c}
variable    s1_distance equal   ${s1_startposx}*2
if ${s1_colnums} == 1 then "variable    s1_increment equal 0" else "variable
s1_increment equal ${s1_distance}*1.0/(${s1_colnums}-1)"
variable    s1_startposz    equal   3.2*1000/${c}

#---
#These are the parameters calculated for the run file#---

jump run.potts_additive_weld run_weld_additive
```

7.7 Appendix F

'run.' script file, 'run.potts_additive_weld'

```
label run_weld_additive
#---
#The "in?." will step In the upper row of the run file
# In the followings the parameters necessary for the simulation are
calculated
#---
seed         ${seed_const}

variable    s1_startposy      equal 0.0
variable    s1_distance equal ${s1_startposx}*2

#[micron]
variable    s1_weldlengthr    equal
      (${W_y}+${s1_spthaz}+${s1_spchaz})*${c}*${s1_colnums}

#[s]
variable    s1_weldtimer      equal ${s1_weldlengthr}/${s1_weldspeedr}

#[sites]
variable    s1_weldlengths    equal ${W_y}+${s1_spthaz}+${s1_spchaz}

variable    s1_weldtimes      equal ${s1_weldlengths}/${s1_weldspeeds}

variable    t_simtime         equal
      ${s1_weldtimer}*${s1_weldspeedr}/(${c}*${s1_weldspeeds})/${s1_colnums}

variable    t_dumpfreq  equal ${t_simtime}/(${s1_colnums}*${dumpconst})
#---
#The calling of the weld additive unique function:
#And the giving of the welding parameters:
#---

app_style    potts/additive         ${nspins_const}     ${s1_spw}     ${s1_spt}
${s1_spd} ${s1_spc}           ${s1_spwhaz}      ${s1_spthaz}      ${s1_spdhaz}
${s1_spchaz} 0.0
#   |————————————————————————————————————————
#   | nspins       = atoi(arg[1])
#   |————————————————————————————————————————
#   | nspins = atoi(arg[1]); #Number of spins
#   | spot_width = atoi(arg[2]); #Width of the melt pool
#   | melt_tail_length = atoi(arg[3]); #Length of tail from meltpool midpoint
#   | melt_depth = atoi(arg[4]); #How many lattice sites deep the melt pool is
#   | cap_height = atoi(arg[5]); #Height of the cap leading the meltpool
#   |————————————————————————————————————————
#   | HAZ = atoi(arg[6]); #Size of the HAZ surrounding the melt pool (must be
larger than spot_width)
#   | tail_HAZ = atoi(arg[7]); #Length of hot zone behind meltpool (must be
larger than melt_tail_length)
```

```
#   | depth_HAZ = atof(arg[8]); //Depth of the hot zone underneath the meltpool
(must be larger than melt_depth)
#   | cap_HAZ = atoi(arg[8]); #Size of HAZ infront of the melt pool (must be
larger than cap_height)
#   | exp_factor = atof(arg[9]); #Exponential parameter for mobility decay in
haz M(d) = exp(-exp_factor * d)
#   |————————————————————————————————————————
#Define simulation domain and initialize site variables
#————————————————————————————————————————
lattice            sc/26n 1.0
region             box block 0 ${L_x}/2 0 ${W_y} 0 ${H_z}
boundary     p n n
create_box   box
create_sites       box
#Chosing the beginning file, for the first layer the initated region is loaded
for any other layer the final grain result of the previous layer  is loaded
if ${a} == 1 then "read_sites site.init" else "read_sites ${c}-${scale}-${L_x}-
${W_y}-${H_z}-${s1_weldspeedr}-${s1_weldspeeds}.1"

#Define an additive scan pattern using a combination of pass, transverse_pass,
cartesian_layer, and pattern
am_pass 1 dir Y distance ${W_y} speed ${s1_weldspeedy}

print "WELD_SPEED [sites/MCS]: "
print ${s1_weldspeedy}
print "WELD_TIME [MCS]: "
print ${t_simtime}

am_transverse_pass 1 distance ${s1_distance} increment ${s1_increment}

am_cartesian_layer 1 start_position ${s1_startposx} ${s1_startposy} pass_id 1
transverse_pass_id 1 serpentine 0

am_pattern  1  num_layers  1  layer_ids  1  z_start  ${s1_startposz}  z_increment
${H_z}
#Setup the solver type and parameters. Must use a "sweep" style solver
sector             yes
sweep        raster
temperature  0.1

#Specify output commands and styles.

jump main_dump.potts_additive_weld write_out_res

label write_out_res_back

run                      ${t_simtime}
clear
jump main.potts_additive_weld loop
```