Pacific Northwest
NATIONAL LABORATORY

PNNL-32702

Modeling and Simulation of Austenitic Welds and Coarse-grained Specimens: Part II

March 2022

Richard E. Jacob
Matthew S. Prowant
Michael S. Hughes
Aaron A. Diaz

Prepared for the U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Under Contract DE-AC05-76RL01830
Interagency Agreement: NRC-HQ-60-17-D-0010

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-32702

Modeling and Simulation of Austenitic Welds and Coarse-grained Specimens: Part II

March 2022

Richard E. Jacob
Matthew S. Prowant
Michael S. Hughes
Aaron A. Diaz

Prepared for the U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Under Contract DE-AC05-76RL01830
Interagency Agreement: NRC-HQ-60-17-D-0010

Carol A. Nove, Contracting Officer Representative

Pacific Northwest National Laboratory
Richland, Washington 99354

Summary

The Pacific Northwest National Laboratory (PNNL) is conducting confirmatory research for the U.S. Nuclear Regulatory Commission (NRC) to evaluate commercially available nondestructive examination (NDE) modeling and simulation software used in the nuclear industry. Simulation results from ultrasonic testing (UT) models can inform the design and qualification of inspection techniques and help interpret inspection results. CIVA is a modeling and simulation package developed by the French Alternative Energies and Atomic Energy Commission (CEA). CIVA was selected for this study because it is readily available and has been used for NDE in the US commercial nuclear power industry. This report is focused on completing the efforts initiated in the previous PNNL report to evaluate UT modeling and simulation performance, reliability, and accuracy in relation to common inservice inspection (ISI) scenarios in nuclear power plants (NPP). This work will be used to provide guidance when establishing methods to perform and evaluate simulations for more standardized model implementation, simulation analysis, and interpretation of results.

NPP licensee requests for relief from ISI requirements are submitted for review to the NRC Office of Nuclear Reactor Regulation (NRR). Relief requests may contain modeling and simulation results to help establish the technical basis of the request. This project is intended to provide NRR with the fundamental understanding needed to effectively evaluate such licensee submittals. In addition, industry representatives have indicated the desire to determine if simulations can be used in lieu of examinations of physical mockups to reduce the time and cost associated with inspection qualification.

The ongoing work at PNNL aims to define best practices for using computational models to evaluate UT scenarios on NPP components. The current report is a continuation of work initiated in Jacob et al. (2020) where CIVA was used to evaluate simulated flaw responses and ultrasonic beam models in austenitic welds, dissimilar metal welds (DMW), and cast austenitic stainless steel (CASS) materials. The current report also addresses additional topics of noise and attenuation, particularly in models of CASS materials.

In Jacob et al. (2020), it was shown that representing coarse-grained equiaxed materials with three-dimensional (3D) Voronoi models saves time while producing more realistic results than two-dimensional (2D) models based on actual CASS geometries. Work was also started on a coarse-grained columnar CASS model. Section 3.0 of the current report wraps up the modeling work on coarse-grained materials. Results again shows that the 3D Voronoi model produce more realistic simulation results than the 2D specimen-based model when compared to empirical beam maps. In addition, the simulated sound fields are measurably different between the equiaxed and columnar models. When applying coarse-grained models to ISI scenarios, the actual grain structure will likely be unknown. Therefore, simulations with multiple specimen models are recommended in order to predict a range of flaw responses since the grain structure may have an impact on the flaw response.

PNNL developed an austenitic weld model from electron backscatter diffraction (EBSD) images acquired on a laboratory weld sample. When the previous technical letter report (TLR) was published, the full weld model was not yet ready for simulations. In Section 4.0 of the current report, PNNL shows beam simulation results using the full EBSD-based weld model and compares results to model variations. The variations included: 1) multiple grain sizes, 2) random Euler angle assignments with the same crystalline shapes as the EBSD model, and 3) altering the crystalline shapes by using a Voronoi geometry but using the same Euler angle

assignments as the EBSD model. Results showed that the specific Euler angle assignments affected the beam scatter but not as strongly as changing the crystalline shapes. The irregular shapes in the Voronoi geometry added more interfaces with the same number of regions; the added interfaces significantly increased the beam scatter. Overall, results show that the specific details of a weld model may not be critical, and a lack of specific grain information should not stop someone from using a weld model. Ideally, the simulation results should be compared to empirical results. As with coarse-grained models, an absence of empirical results will require using multiple models to bound the possible outcomes.

Sections 5.0 and 6.0 examine the effects of adding noise and attenuation to models. Empirical UT scans of coarse-grained materials typically include coherent signal reflections from grain boundaries; such signals appear as structural noise. Structural noise in CASS can affect the ability to detect and characterize flaws; therefore, accurately predicting flaw detection in such materials depends on the ability to accurately simulate structural noise. In Section 5.0, options are investigated for simulating structural noise in CIVA for a range of stainless steel materials. Simulated noise fields are compared to empirical noise to illustrate that CIVA can simulate noise in many scenarios. Attenuation is another key attribute of UT scans, and the severity of attenuation depends strongly on the material properties and probe frequency. Section 6.0 describes how CIVA implements attenuation in simple materials and how attenuation in more coarse-grained materials modeled by Voronoi regions can be simulated.

Section 7.0 summarizes the findings of this report. In a future report, PNNL will consider the findings from their previous and ongoing modeling and simulation activities (including the current report) to draw conclusions and make recommendations on the use of modeling and simulation in nuclear NDE. The goal is to provide the industry and NRC staff guidance on modeling and simulation best practices.

Appendix A describes insight into some CIVA functions and features. This is a continuation of CIVA-specific guidance that was provided in Jacob et al. (2020). Appendix B is a copy of an internal PNNL technical report describing the conversion of Euler angles between different reference frames. Euler angle conversions are important for developing models based on empirically-measured crystalline orientations or for converting angles from one modeling software convention to another.

Acronyms and Abbreviations

2D	two dimensional
3D	three dimensional
BW	backwall
CAD	computer-aided design
CASS	cast austenitic stainless steel
CEA	French Alternative Energies and Atomic Energy Commission
CS	carbon steel
DMW	dissimilar metal weld
EBSD	electron backscatter diffraction
EDM	electrical discharge machined
FEM	finite element modeling
FSH	full-screen-height
GEIT	General Electric Sensing & Inspection Technologies
ISI	inservice inspection
NDE	nondestructive examination/evaluation
NPP	nuclear power plant
NRC	US Nuclear Regulatory Commission
NRR	NRC Office of Nuclear Reactor Regulation
PA	phased array
PNNL	Pacific Northwest National Laboratory
RES	NRC Office of Nuclear Regulatory Research
SDH	side-drilled hole
SNI	Sensor Networks, Inc.
SNR	signal-to-noise ratio
SS	stainless steel
TLR	technical letter report
TRL	transmit-receive-longitudinal
UT	ultrasonic testing
WSS	wrought stainless steel

Contents

Figures

Tables

1.0 Introduction

The US Nuclear Regulatory Commission (NRC) has identified nondestructive evaluation (NDE) modeling and simulation tools as a technical area of focus for confirmatory research. The NRC Office of Nuclear Regulatory Research (RES) initiated a task for the Pacific Northwest National Laboratory (PNNL) to conduct a multi-phased technical assessment of various modeling and simulation tools and to address modeling issues of increasing levels of complexity. PNNL was directed to focus on ultrasonic testing (UT) modeling packages, in particular UltraVision (Zetec, Inc.), CIVA (EXTENDE, Inc.), and OnScale Solve (OnScale, Inc.). The approach to the work has included:

- Beam and flaw-response models of multiple component/material/flaw configurations;
- Empirical data acquisition to compare and validate the model results; and
- A gap analysis of where the models fail to provide effective and reliable results.

To date, PNNL has published a series of three technical letter reports (TLR) on the modeling and simulation assessment (Dib et al. 2017; Dib et al. 2018; Jacob et al. 2020). These reports have covered issues including:

- Quantitative metrics for measuring simulation outcomes and comparing results to empirical data;
- Understanding uncertainties in simulation results and the role of verification and validation;
- Understanding how variability in parameter selection affects variability in simulation outcomes;
- Exploring the relationship between beam simulations and flaw response models and whether the former can be used as a surrogate for the latter;
- Developing and testing models of austenitic welds, coarse-grained materials, and dissimilar metal welds;
- Comparing beam simulations generated by UltraVision and CIVA.

The current report is the fourth in the series and focuses on wrapping up efforts to assess CIVA modeling of coarse-grained materials, austenitic welds, specimen noise, and sound field attenuation. A future report will address flaw response simulations and will explore UT modeling using OnScale Solve. The modeling and simulation work will then be concluded with publication of a NUREG/CR on UT modeling, identifying gaps and best practices for the nuclear NDE industry.

The NRC Office of Nuclear Reactor Regulation (NRR) receives licensee requests for relief from inservice inspection (ISI) requirements. In the event that licensees use modeling and simulation as technical justification in relief requests, PNNL anticipates that NRR would use these reports to help evaluate the requests.

In commercial nuclear NDE, models can be used to help develop inspection procedures, calculate beam coverage, predict flaw detection, and inform probe and mockup design. Ultimately, the goal of modeling is to save time, money, and resources while maintaining or improving plant safety. The use of NDE modeling and simulation continues to grow, but modeling has some significant limitations:

- Human factors cannot be effectively modeled, as they are dependent on the skill level, experience, and physical environment of the NDE examiner;
- Models may be invalid for certain scenarios if they contain approximations or simplifications;
- Incomplete knowledge of the specimen material, grain orientation and microstructure, and/or geometry can have a strong impact on simulation accuracy;
- Unanticipated variables can occur in the field, such as noisy scan data, scan range limitations, and spurious signals like those from weld root, counterbore, or fabrication defects.

Modeling and simulation limitations should be well understood so that results can be interpreted correctly. Ideally, simulation outcomes are accompanied by empirical data for validation. Empirical data can also help catch unanticipated errors or oversights so that models can be corrected and refined, thus improving the usefulness and reliability of the models.

The previous PNNL modeling and simulation report (Jacob et al. 2020) focused on using CIVA to model austenitic welds and coarse-grained materials, such as cast austenitic stainless steel (CASS). These materials are challenging for typical inservice examinations due to beam scatter, attenuation, and redirection. That report described methods for developing and testing specimen models that have anisotropic and complex microstructures. The current report continues that work, as described in the paragraphs below.[1]

Section 2.0 of this report wraps up the modeling work on coarse-grained materials. In the previous PNNL TLR, it was shown that representing coarse-grained equiaxed materials with three-dimensional (3D) Voronoi models saves time while producing more realistic results than two-dimensional (2D) models based on actual CASS geometries. In the current report, the work was extended to a coarse-grained columnar CASS model that was described in the previous report. Results again show that the 3D Voronoi model produces better simulation results than the 2D specimen-based model when compared to empirical beam maps. Results also show that the beam simulations with the equiaxed and columnar grain structures are measurably different, as observed empirically. When applying coarse-grained models to ISI scenarios, the grain structure will likely be unknown. Therefore, simulations with multiple scenarios are recommended in order to predict the range of responses.

PNNL developed an austenitic weld model from electron backscatter diffraction (EBSD) images acquired on a laboratory weld sample. At the time the previous TLR had been published, the full weld model was not yet ready for simulations. In Section 4.0 of the current report, PNNL shows beam simulation results using the full EBSD-based weld model and compares results to variations on the model. The variations included 1) random Euler angle assignments with the same crystalline shapes as the EBSD model and 2) altering the crystalline shapes by using a Voronoi geometry with the same Euler angle assignments as the EBSD model. Results showed that the specific Euler angle assignments affected the beam scatter but not as strongly as changing the crystalline shapes. The Voronoi geometry had more interfaces with the same number of regions; the added interfaces significantly increased the beam scatter. Overall, results show that the specific details of a weld model may not be critical and should not stop someone from using a weld model in simulations. Ideally, the model should be informed by

[1] Jacob et al. (2020) primarily focused on CIVA 2017. A new version of CIVA was released near the conclusion of that work, so the current report uses CIVA 2020. CIVA 2021 was then released after work in the current report was completed.

empirical results. As with coarse-grained models, the absence of empirical results will require using multiple models to bound the possible outcomes.

Finally, in Sections 5.0 and 6.0 the effects of adding noise and attenuation to models are examined. Empirical UT scans of coarse-grained materials typically include signals reflected from grain boundaries; such signals appear as structural noise. The ability to accurately predict flaw detection in such materials depends on the ability to accurately simulate structural noise. In Section 5.0, options are investigated for simulating structural noise in a range of stainless steel materials. Simulated noise fields are compared to empirical noise to illustrate that CIVA can effectively simulate noise in many scenarios. Attenuation is another key attribute of UT scans, and the severity of attenuation depends strongly on the material properties and probe frequency. Section 6.0 describes how CIVA implements attenuation in simple materials and how attenuation in more coarse-grained materials modeled by Voronoi regions can be simulated.

2.0 Terminology and Concepts

PNNL uses the term “model” when referring to the digital representation of a specimen, probe, flaw, etc. or combination thereof. “Model” also describes the mathematical framework used for computation. PNNL uses “simulation” when referring to the computational execution of a model scenario with a particular set of input parameters. A model provides an imitation or representation of the real-world scenario, whereas a simulation uses the model to predict behavior under user-defined conditions.

CIVA is an NDE simulation software package published by EXTENDE, Inc. and originally developed by the French Alternative Energies and Atomic Energy Commission (CEA) for simulating inspections of nuclear power plants. CIVA is used globally in the nuclear industry. It has a UT-specific module with a library of standard geometries and transducers common in UT inspections, which simplifies the modeling effort for most users. CIVA uses a semi-analytical approach and multiple approximations to calculate 3D sound fields in beam simulations and flaw-response simulations. As described in Mahaut et al. (2010), CIVA uses a “pencil” method, or beam-calculation approach, to calculate sound fields and sound propagation. This method uses a high-frequency approximation, which requires the wavelength to be on the order of, or smaller than, the size of flaws or model geometry features. With this approach, CIVA is able to compute results with good accuracy in a relatively short time, albeit with some limitations. PNNL has primarily used CIVA for the modeling and simulation work under the NRC contract because the nuclear NDE industry is a primary target of CIVA applications.

The finite element method (FEM) is a more common approach to modeling because it is much more flexible than CIVA. However, FEM typically requires some computational or programming expertise, and FEM packages usually lack a UT-specific module with pre-defined probes and geometries as found in CIVA. The FEM approach discretizes the model into elements by use of a mesh, then solves the sound field for each element by a series of boundary-value problems using a system of differential equations. FEM has more flexibility than CIVA because it is not bound by the high-frequency approximation and other model assumptions, but simulations typically require much more computing power and time, particularly for 3D computations. There are dozens of commercial and open-source FEM software packages, many with toolboxes specific to ultrasonic simulation. Examples of common commercial FEM software include: OnScale Solve, Ansys (Ansys, Inc.), COSMOL Multiphysics (COSMOL, Inc.), and Abaqus (Dassault Systemes).

It is necessary to expand a bit on the high-frequency approximation in CIVA. The high-frequency approximations do not cause CIVA to fail at a particular cut-off frequency, nor does CIVA warn the user when the approximation may be violated. Rather, simulation results may gradually become less reliable as the degree to which the approximation is violated increases. The user can run simulations that violate the model approximations, so it is the user’s responsibility to understand the model limitations and to use appropriate model parameters. In PNNL’s experience, most modeling scenarios are executed well at frequencies of 2 MHz and higher (the wavelength in steel at 2 MHz is about 3 mm [0.12 in.]). Some scenarios at 1 MHz are also possible, but results should be scrutinized. Frequencies below 1 MHz are not practical for most scenarios unless specimen and flaw sizes are increased proportionally. For example, low-frequency (~500 kHz) simulations in CASS models are generally not feasible in CIVA, and, therefore, PNNL has avoided such simulations. Metamodels or parametric studies can be run in CIVA to explore model limitations for a given scenario.

Empirical data are data acquired in the field or laboratory. For PNNL, the vast majority of empirical data are collected in the laboratory under controlled conditions. The laboratory setting allows for examination of specifically-designed mockups and for acquisition of more data than in the field, as scans can be repeated as necessary using a variety of probes or scan parameters without time or dose constraints commonly experienced in the field. Laboratory work also allows for destructive testing of mockups to examine grain structures, weld profiles, and flaw shapes. However, laboratory data can lack the realism of field data, especially when surrogate flaws are used, such as electrical discharge machined (EDM) notches or saw cuts.

3.0 Working with Cast Austenitic Stainless Steel and Dissimilar Metal Weld Models

3.1 Summary of Previous Work

CASS materials pose unique challenges for UT inspections. CASS grain structures redirect and attenuate the sound field, resulting in significant beam scatter, high noise levels, and poor sound field penetration (Jacob et al. 2019). Each CASS component has a unique grain structure that distorts the sound field differently, making it impossible to use modeling and simulation to exactly predict sound coverage and flaw echo responses. The challenges are extended to dissimilar metal weld (DMW) models, particularly those adjacent to CASS base metal, as the weld geometry and buttering have their own unique grain structures. Specimen properties typically have the largest effect on simulation accuracy, so it is important to use a model of the specimen that most closely emulates the material's effect on the sound field. Since it is impossible to exactly replicate the material properties (e.g., sizes, shapes, and crystalline orientations) of the grains, the properties must be approximated in simplified models.

Jacob et al. (2020) showed a step-by-step method for creating a CASS specimen model from a cut and polished section. The method used photography and image filters to extract grain boundaries and then assigned Euler angles to each grain. A coarse-grained CASS model was created, and beam simulations were generated in CIVA to test against a comparable Voronoi model. Voronoi models are convenient because they are a built-in option in CIVA for certain geometries (Jenson et al. 2010). Simulation results on both models were qualitatively compared to empirical beam maps. Results showed that developing a realistic coarse-grained CASS model is feasible in CIVA and that beam simulation results appear to provide a realistic representation of beam scatter. Results also showed that the Voronoi model is an excellent proxy for the specimen-based model. Indeed, the Voronoi model has many advantages over the specimen-based model: the Voronoi model was much faster to create, it was inherently 3D, and the simulation times were shorter.

Previous work with DMW models has been performed using various approaches to modeling the weld material. In the simplest approach, a homogeneous weld region was used to join two homogeneous materials (Kumar et al. 2021). Although each material in the model was assigned different material properties, no effort was made to simulate granular structure. The next simplest approach was to use the Ogilvy model of weld grain orientation with homogeneous buttering material. The resulting model is a built-in feature in CIVA, so it is easy to implement. The Ogilvy model has been used in DMW simulations (Gardahaut et al. 2014; Kim et al. 2016). Results in CIVA and using FEM showed no beam scatter, but some beam redirection was observed. Another implementation of DMW models divided the weld and buttering into regions of predominant dendrite orientation (Gardahaut et al. 2012). Note that dendritic orientation (defined by the long axis of the dendrite) is not the same as crystalline orientation, which is determined by the Euler angles. Beam simulation results in CIVA showed increased scatter over the Ogilvy model, but minimal beam redirection was observed as compared to the Ogilvy model. In another study, a group performed CIVA flaw-response modeling on a DMW model where the weld was divided into predominantly vertically-oriented regions, each of which was assigned a different anisotropy (Szávai et al. 2016). Although good agreement with experimental scans was reported, this weld model also did not appear to produce significant beam scatter. With the development of a CASS model and an austenitic weld model (as described in Jacob et al. (2020)), a realistic CASS-carbon steel (CS) DMW model is now straightforward to construct by combining the different model elements of CASS, weld, and buttering.

3.2 Columnar Cast Austenitic Stainless Steel Model

In Jacob et al. (2020), a columnar specimen model was created from a section of specimen B519-C, which was originally used in the Programme for the Inspection of Steel Components (PISC-III) CASS round robin work conducted in the 1980's (Bates et al. 1987; Heasler et al. 1993). Figure 3.1 shows a polished and chemically-etched section of B519-C. Simulations using the specimen model were attempted but would not run to completion in CIVA 2017. With the release of CIVA 2020, however, the simulations were attempted again and completed successfully.

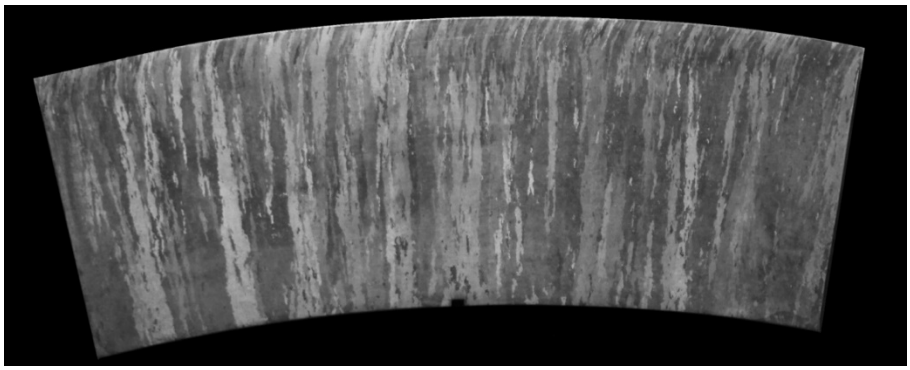

Figure 3.1. Photo of specimen B519-C illustrating a fine-grained columnar microstructure.

Figure 3.2 shows an example of a columnar model developed from B-519C (top left). Ten different sets of Euler angles, as taken from Chen et al. (2015), were assigned to the grains; each color represents a different set of angles. The Euler angle assignments were randomly assigned with the constraint that neighboring regions cannot have the same set of angles. Ten models were created, each of the same geometry and Euler angles but with scrambled Euler angle assignments. The model was 2D, so the geometry was extruded in the third dimension. The probe was oriented such that the sound propagation was parallel to the extruded direction, and a sound field was captured in a plane perpendicular to the extruded dimension. Ten beam simulations were performed using a 1 MHz, 45° phased-array probe and a CIVA accuracy factor of 16. The same settings were used for these simulations as were used in previous CASS simulations based on the model of the equiaxed specimen AAD-3 (Figure 3.2, top right). A 1 MHz probe was used so that comparisons could be made to previously-acquired empirical data on the same specimen prior to it having been cut (see Section 3.3). These settings are detailed in Jacob et al. (2020).

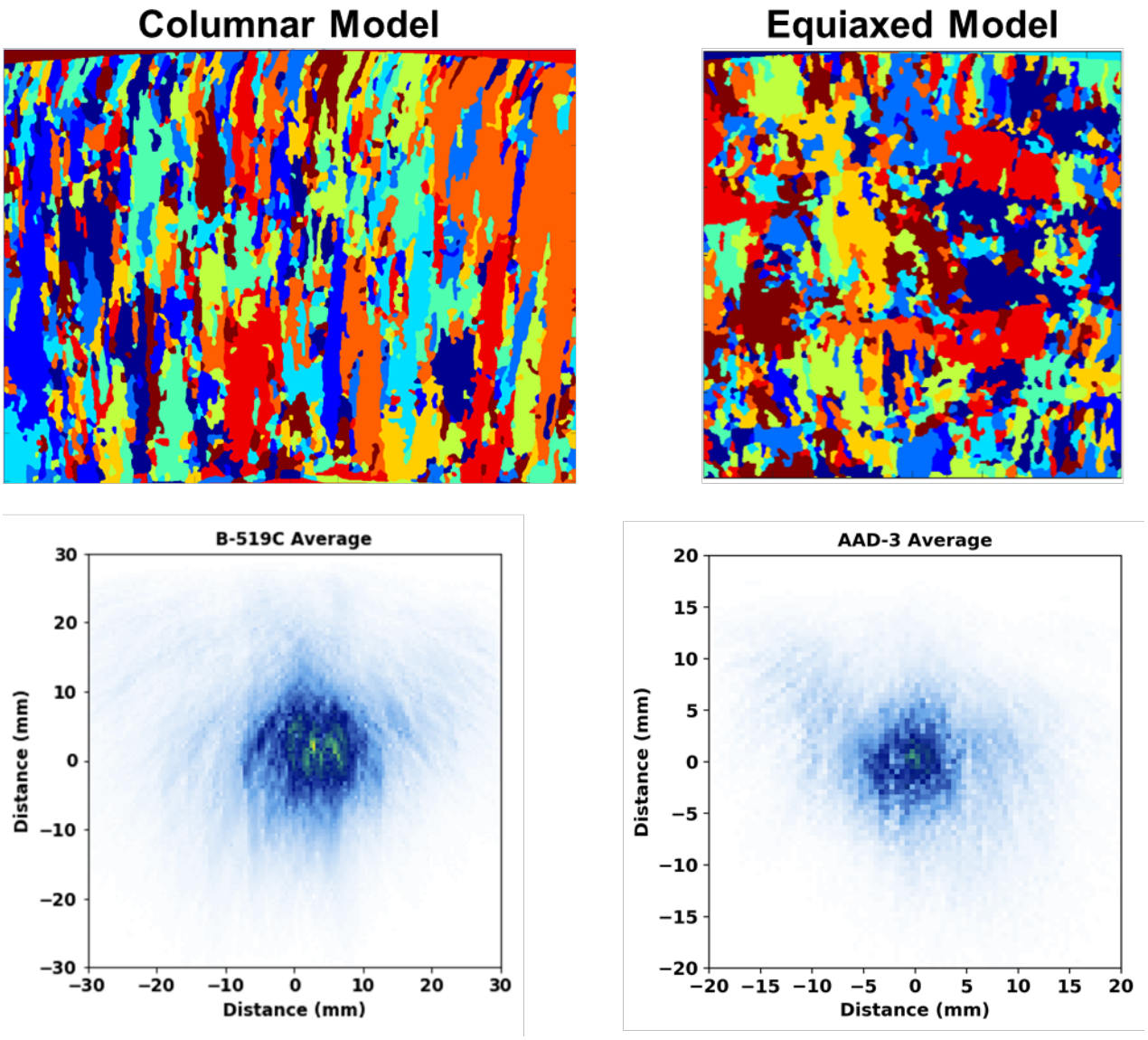

Figure 3.2. Columnar and equiaxed microstructural geometries. The columnar geometry was extracted from the photographs of B519-C. The equiaxed geometry is reproduced from Jacob et al. (2020). Each color represents a different set of Euler angles. The averages of 10 beam simulations are shown for comparison.

Figure 3.3 shows the results of the ten B519-C simulations. The images were divided into quadrants to illustrate that beam redirection often resulted in asymmetric sound fields. In many cases, the bulk of the beam was skewed to the right of center. Also, the sound field appears slightly elongated in the vertical direction as compared to the coarse-grained equiaxed CASS beam maps (see the simulation results in Figure 3.2). Overall, the B519-C simulation results are only slightly different in appearance from those of AAD-3.

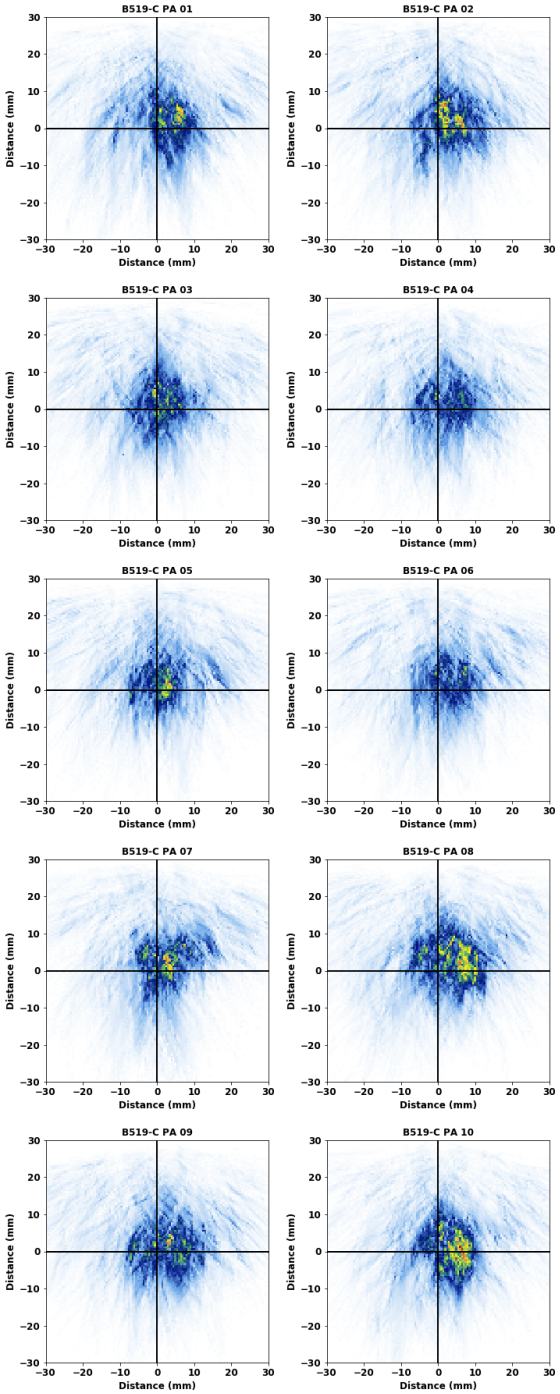

Figure 3.3. The B519-C geometry with assigned Euler angles. Each color represents a different set of angles. A total of 10 sets was used.

Following the B519-C simulations, Voronoi models were generated to imitate columnar CASS material. In a previous CASS model of equiaxed grains, the CIVA Voronoi model was elongated in order to imitate the extrusion of the 2D specimen-based model. The elongation was done by giving the Voronoi grains a high aspect ratio. The aspect ratio can only be applied in one dimension, so for columnar Voronoi grains the aspect ratio must be used to make the grains columnar. Thus, the grains cannot be elongated again in the third dimension to mimic the

extruded 2D B519-C geometry. In the final model, the columnar Voronoi grains were 3D and the specimen-based grains were extruded 2D. Figure 3.4 shows a side-by-side comparison of the cross-sections of B519-C (left) and a columnar Voronoi model (right). The Voronoi specimen was 100 mm × 80 mm × 60 mm (3.9 in. × 3.1 in. × 2.4 in.) and had 10,000 regions, so the average grain volume was 48 mm^3 (0.003 $in.^3$). The Voronoi aspect ratio was 7.

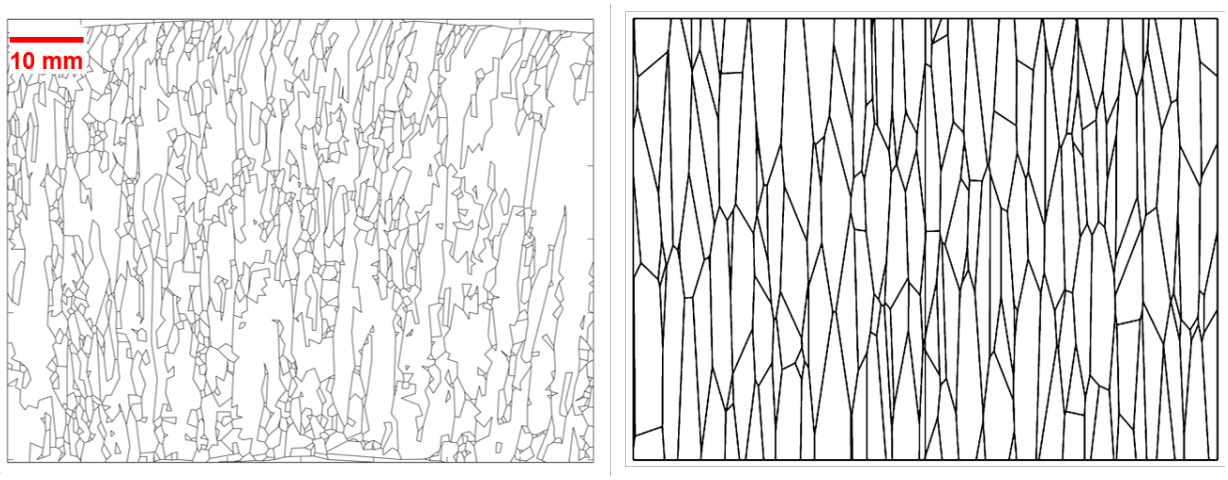

Figure 3.4. B519-C model (left) and Voronoi (right).

To simulate sound scatter in Voronoi regions, CIVA varies the speed of sound from grain to grain instead of using Euler angles. The average speed of sound of the specimen can be set along with a range of velocities, defined by the parameter ΔV. Sound speeds within the ΔV range are randomly assigned to the regions. For example, if the specimen sound velocity is set to 6,000 m/s and ΔV is set to 10% (or 600 m/s), then each Voronoi region will be randomly assigned a velocity between 5,400 m/s and 6,600 m/s. In the AAD-3 equiaxed case, the value of ΔV was varied and was found to significantly affect the simulation outcomes. The same approach was done with the columnar model. Three different values of ΔV were tried: 4%, 6%, and 8%; the results are shown in Figures 3.5, 3.6, and 3.7, respectively. Note that 10 simulations were initially run with ΔV=4%. This was reduced to six simulations each of ΔV=6% and ΔV=8% to save time. The ΔV was randomized for each simulation.

One minor problem occurred in the way that the Voronoi specimen model varied from simulation to simulation. For the equiaxed cases in Jacob et al. (2020), the same Voronoi geometry was maintained for each simulation, but the velocity assignments were randomized. This was akin to using the same AAD-3 geometry but assigning different Euler angles for each simulation. For the columnar cases, however, a bug in CIVA 2020 prevented randomization of the velocity assignments with the same Voronoi geometry; therefore, each Voronoi columnar simulation had a different randomly-generated geometry. (Note: this bug was brought to the attention of EXTENDE and fixed in a CIVA 2020 service pack update). Ideally the geometry would be the same for each simulation; in this case the Voronoi geometry was an uncontrolled variable, which can complicate comparisons of simulation results. However, with many Voronoi regions, the differences from simulation to simulation caused by the geometry differences are expected to be small.

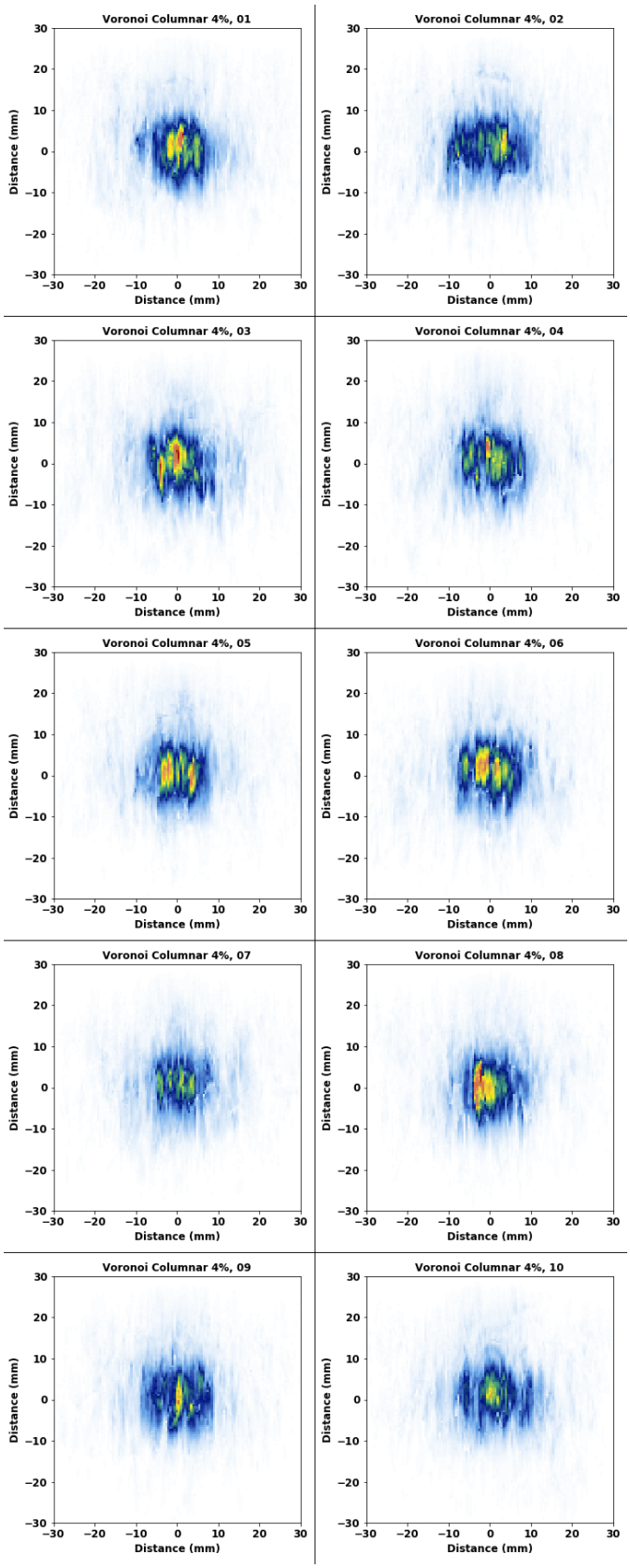

Figure 3.5. Beam simulations with the columnar Voronoi model, ΔV=4%.

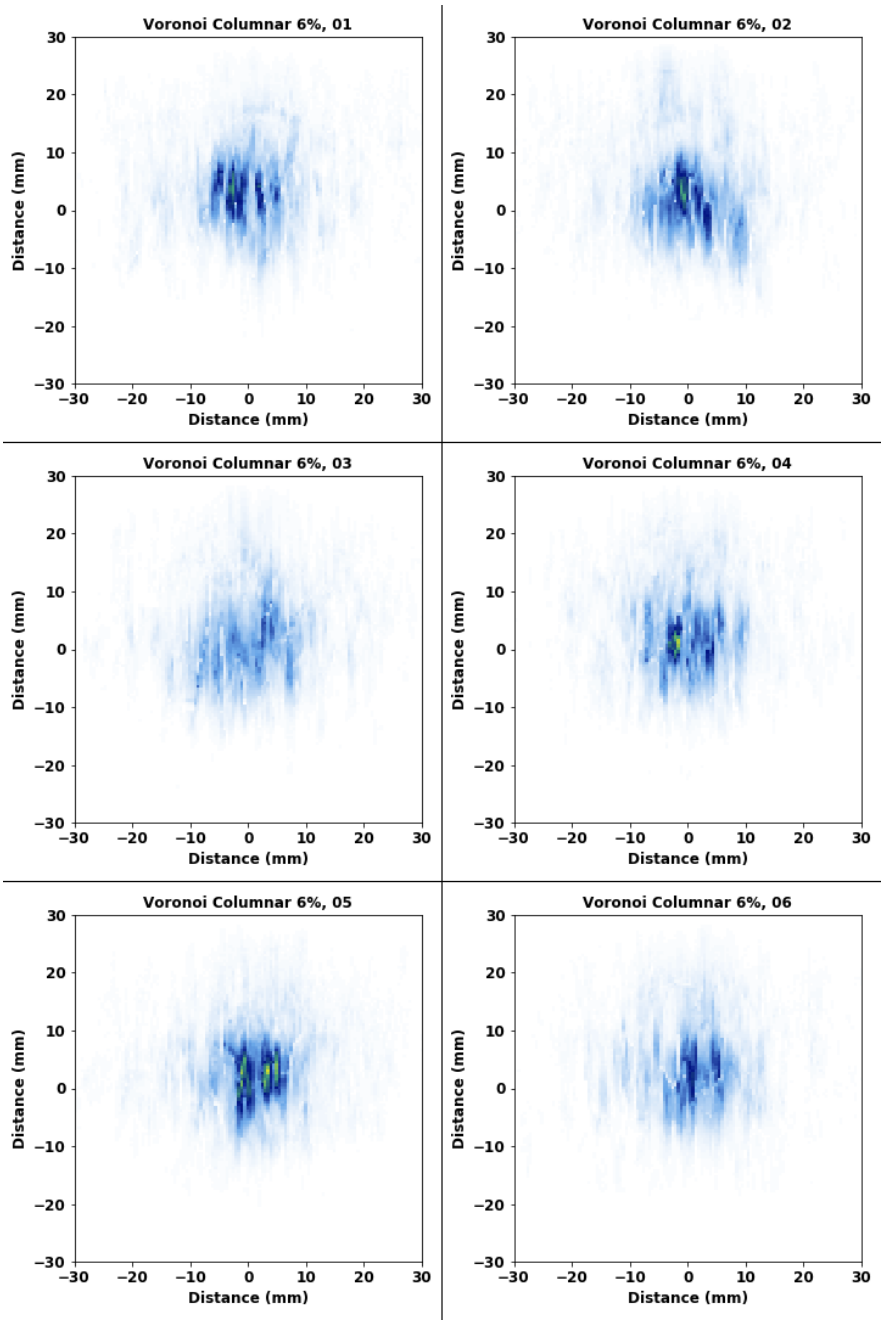

Figure 3.6. Beam simulations with the columnar Voronoi model, ΔV=6%.

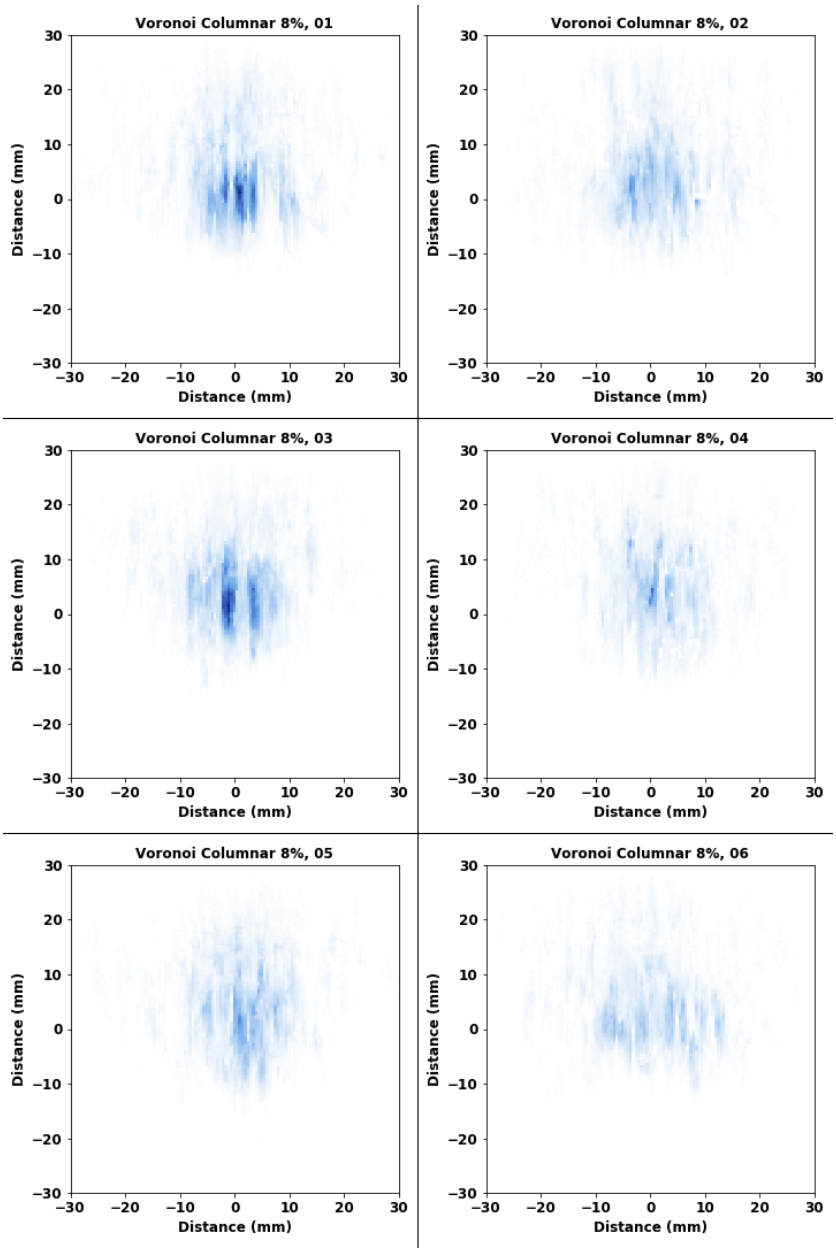

Figure 3.7. Beam simulations with the columnar Voronoi model, ΔV=8%.

Table 3.1 compares some of the key attributes of the B519-C and Voronoi columnar models. Two significant differences are the time required to open the simulation files and the grain sizes. First, CIVA takes much longer to open and process custom geometries with many interfaces than it does the Voronoi geometries, adding a significant amount of time to model preparation and simulation analysis. Second, to compare the grain sizes, a cross-sectional plane was taken through the Voronoi model, and grain areas were measured. The B519-C model had a smaller average grain size, a larger range of grain sizes (as indicated by the high standard deviation [SD] relative to the mean), and a majority of small grains, as indicated by the low median value (these small grains may not be impactful on the results since the largest grains have the greatest influence on the resulting sound field map (Wan et al. 2017). For the Voronoi model, the average grain size depended on the number of grains and the specimen volume. After some

trial-and-error, the number of Voronoi grains was set at 10,000 to qualitatively match the "typical" appearance of the larger B519-C grains. The number of Voronoi grains can easily be changed in CIVA; however, increasing it substantially will increase the simulation time (and require higher accuracy factors, further lengthening simulation times), while decreasing it will make the grains too large. The absolute number of grains can be reduced while maintaining the same grain size by reducing the specimen volume commensurately, if reducing specimen volume is feasible for the model.

Table 3.1. Comparison of B519-C and columnar Voronoi simulations.

B519-C	Voronoi
2D, extruded in third dimension	True 3D
1,034 total grains	10,000 total grains (this variable can be changed in CIVA to affect the grain sizes)
80 mm × 60 mm (3.1 in. × 2.4 in.) cross-sectional area	80 mm × 60 mm (3.1 in. × 2.4 in.) cross-sectional area
Grain size: average ± SD: 4.7 ± 15.7 mm^2 (0.007 ± 0.024 in.^{2}) median: 1.1 mm^2 (0.0017 in.^{2})	Cross-sectional grain size: average ± SD: 27 ± 17 mm^2 (0.042 ± 0.026 in.^{2}) median: 25 mm^2 (0.039 in.^{2})
Time required for CIVA to open the simulation file: ~45 minutes	Time required for CIVA to open the simulation file: ~5 minutes
Simulation time (accuracy = 16): ~5.5 hours	Simulation time (accuracy = 16): ~4.5 hours

It is important to determine an appropriate accuracy factor in CIVA (accuracy factor essentially controls the mesh density) by running a few test simulations. For example, Figure 3.8 shows the same Voronoi model simulated with accuracy factors of 8 (left), 16 (center), and 32 (right). These simulations ran with times of approximately 1.25 hours, 4.5 hours, and 15 hours, respectively.[2] Clearly, accuracy=8 is inadequate, as many pixels were uncalculated. The simulation with accuracy=16 differs only slightly from that of accuracy=32. Note that in the equiaxed/extruded case, an accuracy of 2 or 3 was adequate for the Voronoi model. However, with the large number of columnar Voronoi regions, a higher accuracy factor was needed. In the end, both models had the same accuracy factor.

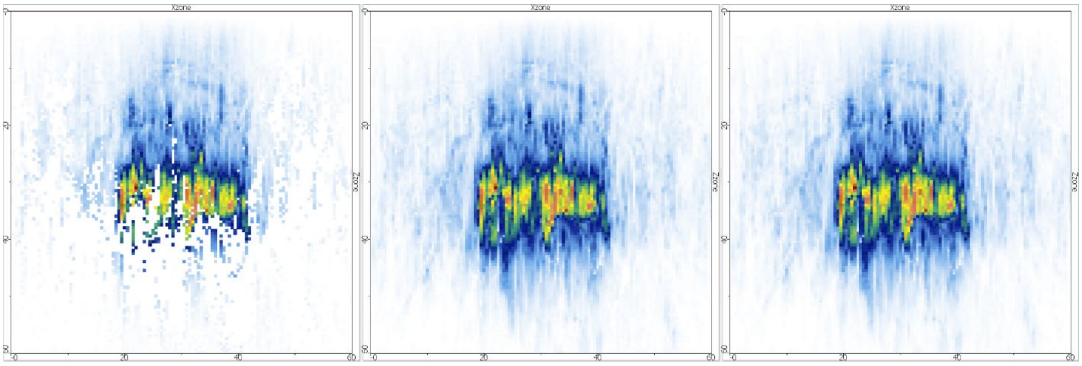

Figure 3.8. Columnar Voronoi simulations with accuracy factors of 8 (left), 16 (center), and 32 (right).

[2] This gives an idea of how the accuracy factor affects simulation times, though they will vary from computer to computer.

3.3 Beam Partitioning in Cast Austenitic Stainless Steel

Beam partitioning is the sectioning of the beam by the CASS grain structure. Beam partitioning is caused by the elongated grains that scatter and redirect the sound field asymmetrically, and it can produce a shadowing effect for certain volumes of the sound field since the energy is diverted elsewhere. PNNL observed that beam partitioning is a signature of columnar grains and developed a metric to quantitatively compare empirical sound fields through different materials (Crawford et al. 2014). The metric is calculated by drawing a horizontal line through the beam map to create a profile, and every instance that the signal intensity exceeds a threshold is counted as a “crossing.” This process is repeated for every horizontal line through the image, and the sum of all crossings is taken. Similarly, the crossings of all vertical lines are counted. The ratio of horizontal to vertical crossings is the crossing ratio, or partitioning metric. PNNL applied the metric to compare the Voronoi simulations, the B519-C simulations, and the empirical beam maps.

Figure 3.9 shows an example of the crossings from a horizontal and vertical line profile through a Voronoi beam simulation. Panel A shows the original image. Panel B shows the image with a median filter applied and horizontal and vertical lines where example profiles were taken. The filter reduces pixel-to-pixel fluctuations while preserving edges and boundaries. Panels C and D are the image profiles through the horizontal and vertical lines, respectively. The black horizontal lines in C and D represent the threshold level (in this case, 70% of the maximum signal), and the red circles show the crossings. In the example in the figure, there are two horizontal crossings and one vertical crossing, resulting in a crossing ratio of 2. All horizontal and vertical profiles of the image are interrogated in order to find an overall ratio. Beam partitioning should result in more horizontal crossings than vertical crossings for columnar materials, resulting in a ratio >1. For equiaxed grains, on the other hand, there should be no preferred direction of beam segmentation, and the ratio should be ≈1.

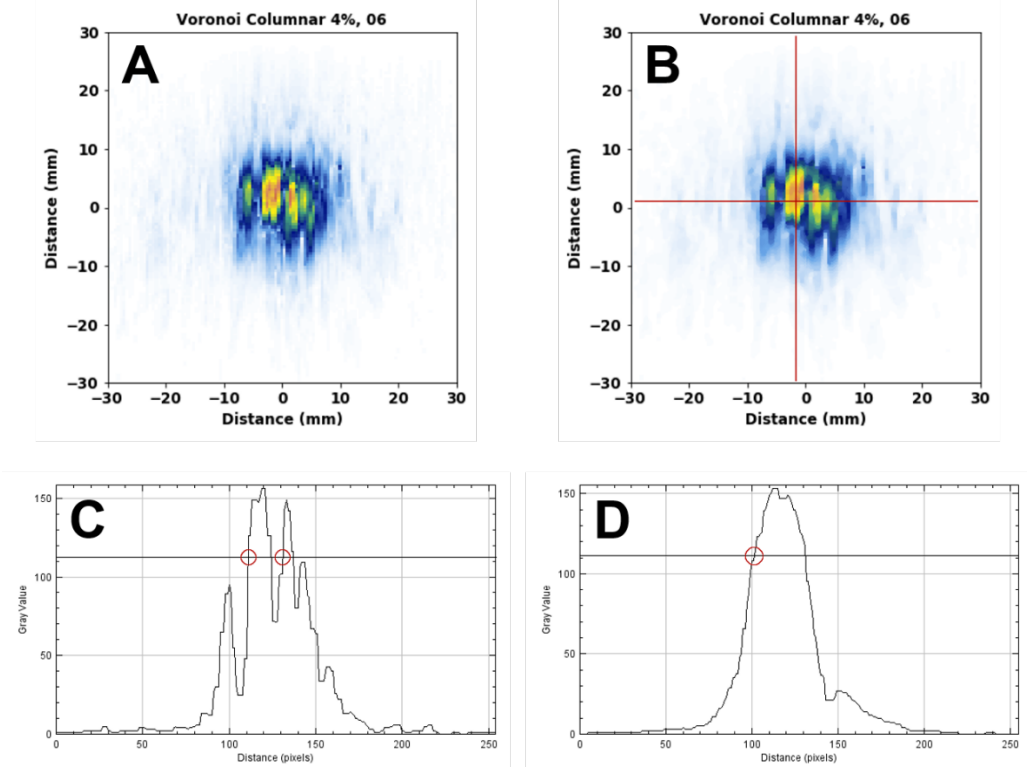

Figure 3.9. Example of how the crossing ratio is calculated. A) A simulated beam map through a columnar Voronoi specimen model. B) The same beam map with a median filter applied and with positions of horizontal and vertical sections indicated. C) The horizontal line profile with a threshold level (black line) and threshold crossings (red circles). D) The vertical line profile with the threshold crossing indicated (red circle).

Table 3.2 shows the mean crossing ratios of the Voronoi columnar, B519-C (columnar), and AAD-3 (coarse-grained equiaxed) simulations. Also included are approximate values taken from the 1 MHz, 45° longitudinal-wave data shown in Figure 6.12 of Crawford et al. (2014). Note that the high number of samples taken (every row and column of each image) resulted in low SD. In the table, the SDs are presented with an extra level of precision, i.e., to the second decimal point to avoid misrepresentations from rounding (e.g., all SDs would have to be rounded up to 0.1 because rounding down to 0.0 would not make physical sense, but rounding all to 0.1 would mask the significance of the SD).

Table 3.2. Comparison of crossing ratios.

Case	Mean Crossing Ratio ± SD
Voronoi columnar ΔV=4%	2.3 ± 0.05
Voronoi columnar ΔV=6%	2.4 ± 0.06
Voronoi columnar ΔV=8%	2.2 ± 0.05
B519-C (simulated)	2.0 ± 0.02
B519-C (from Crawford et al.)[(a)]	2.4 ± 0.04
AAD-3 (simulated)	2.0 ± 0.04
AAD-3 (from Crawford et al.)[(a)]	1.4 ± 0.03

(a) approximate, as read from Figure 6.12 in Crawford et al. (2014)

The crossing ratios of AAD-3 and B519-C simulations did not differ from each other. The simulations were not ideal representations of the different grain structures because the models were inherently 2D and were extruded in the third dimension. For AAD-3 (simulated), the ratio is higher than empirical, and for B519-C (simulated), the ratio is lower than empirical. Notably, the Voronoi columnar simulations and the B519-C empirical results agree. Importantly, the crossing ratio of the Voronoi models was largely independent of ΔV, so the optimal value of ΔV should be determined based on other factors, such as signal intensity or scatter. The ΔV was not optimized here, but the optimization process is shown in Jacob et al. (2020).

3.4 Dissimilar Metal Weld Model

DMW models can be particularly challenging because there are four distinct regions that need to be modeled: both parent materials, the weld region, and the buttering. Each region has a distinct grain pattern and material properties that will affect the sound field differently. PNNL attempted two approaches to making a DMW model. The first was to cut, polish, and etch a DMW section from the large-bore mockup described in Jacob et al. (2019). The section was then photographed to extract the grain boundaries using the method outlined in Jacob et al. (2020). Figure 3.10 is an example of a photograph taken with ambient light (top), the light source to the left (bottom left), and the light source to the right (bottom right). The bottom photos were converted to greyscale and masked to retain the CASS, the weld, and the buttering sections. For CASS specimens B519-C and AAD-3, this process resulted in a suitable grain representation. However, the DMW had far too many grains, including many small grains that were difficult to outline. All of the small grains resulted in more line segments than CIVA could efficiently handle, yet combining the small grains into larger grains would have changed the scattering and sound propagation effects appreciably (Wan et al. 2017).

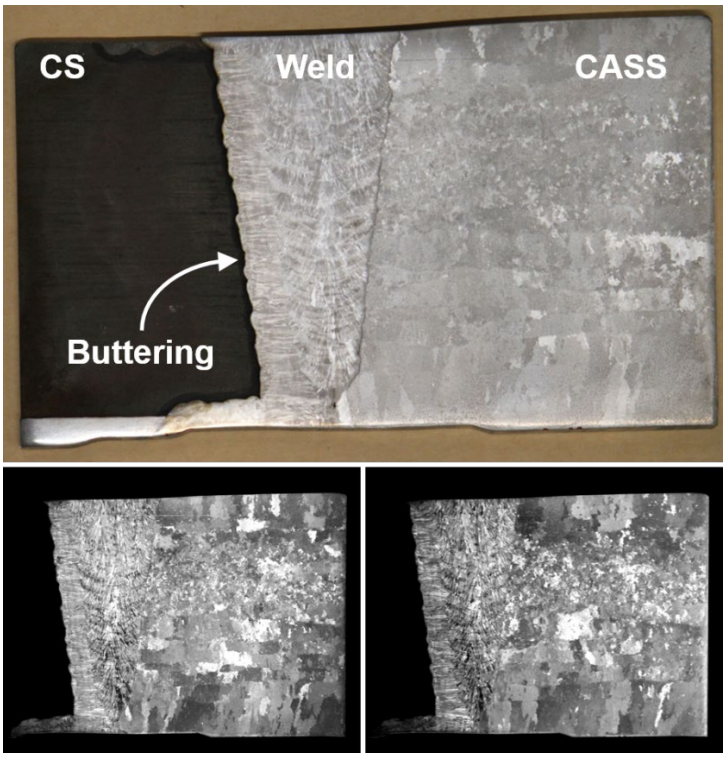

Figure 3.10. Photographs of a polished and etched section from a DMW mockup. Top: a photograph taken with ambient light. Bottom left: the light source to the left. Bottom right: the light source to the right.

The second approach was to build onto the electron backscatter diffraction (EBSD) weld model started in the previous TLR and continued in Section 4.0 (cf. Figure 4.1) by adding Voronoi regions on the CASS side and a buttering region on the CS side. Recall that in CIVA 2020 the Voronoi option is only available in simple specimen models. Therefore, Voronoi regions were created in another platform and imported into CIVA using a method such as manual tracing or digitization in computer-aided design (CAD) software. A drawback of this approach is that the Voronoi regions will be 2D, but this is consistent with the weld model, which is also 2D. Tracing in the CIVA CAD tool is a reasonable option if the number of grain boundaries is not too large and if the process only needs to be done once or twice.

PNNL used the second approach to generate a DMW model. First, a 2D image with Voronoi regions was generated in ImageJ,[3] then this image was converted to a CAD file in MATLAB and merged with the 256-pixel EBSD weld model. Voronoi line segments that intersected the weld were terminated at the weld boundary. The geometry was then imported into CIVA to test that CIVA could read it. At this stage, the CIVA CAD editor was used to reduce the total number of line segments—and therefore the computational burden on CIVA—by manually combining small regions with neighboring regions and by straightening some segments. At this stage, a buttering region was hand-drawn. An isotropic region was then added to the buttering side to represent CS or wrought stainless steel (WSS), and another isotropic region was added to the CASS end to extend the specimen to accommodate the probe. The final geometry was exported from CIVA, and MATLAB was used to assign Euler angles to each grain of the CASS and butter regions (the Euler angles of the EBSD weld model were retained in the weld region). Bunge

[3] ImageJ is Java-based image analysis software available for free at https://imagej.nih.gov/ij/.

Euler angles were taken from Chen et al. (2015) and converted to the specimen frame of reference for CIVA.[4] Sets of angles were randomly assigned to each grain, assuring that bordering grains did not share the same angles (refer to the process described in Jacob et al. (2020). The final geometry was then reimported into CIVA.

The resulting DMW specimen model is shown in Figure 3.11. When extruded in the third dimension, it can be used as a general framework for DMW simulations. The model is 33 mm (1.3 in.) thick, which was based on the original size of the EBSD weld model. Note that this model was simplified by excluding the cladding layer, which is typically found on the CS side of such welds. Such a layer may have a small effect on beam simulations, but it is not expected to be significant for flaw response simulations when the flaw is located in the weld. The final model can be scaled up or down to change the specimen size. Euler angle and/or elastic constant assignments can be changed relatively easily to create a "new" model. Changes can be made directly to the CIVA specimen xml file using, for example, Python or MATLAB. Note that downscaling the model may result in small grains that violate CIVA's high-frequency modeling assumptions. In general, regions should be larger than 10% of the wavelength (Chen et al. 2015; Nageswaran et al. 2009).

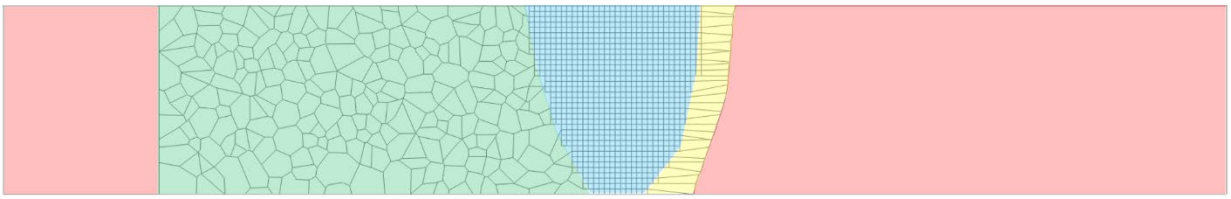

Figure 3.11. DMW model. The regions are color coded as follows: Red—isotropic; green—CASS; blue—weld; yellow—butter. The isotropic region on the left was added to support the probe and allow for longer scan lengths or steeper refraction angles; there was no need for added geometrical complexity in this region, so no grains were defined.

Beam simulations were run on the DMW model using a 2 MHz TRL probe (the same probe used in previous weld models) at 10 mm (0.4 in.) increments across the specimen model. Figure 3.12 shows the simulated beams at the different probe positions from the CASS side (left), and from the CS side (right). Three interesting phenomena were observed from these simulations.

1. The weld scatters the sound more than the CASS material. Almost no beam formation occurs if the beam passes primarily through the weld.

2. The combination of CASS plus weld is more scattering and attenuative than CS plus weld. Flaws that are located such that the sound must propagate through the CASS and the weld will be the most difficult to detect. This result is unsurprising, but it suggests that the responses from such flaws will likely be the most difficult to model accurately because the material properties of the weld and CASS are difficult to define and model.

[4] Converting Euler angles to and from different frames of reference is not trivial, but it is an important aspect of maintaining consistency between reference frames. The EBSD system, CIVA, and OnScale all use different reference frames, so conversions must be done when importing Euler angles or when translating models between software platforms. Appendix B provides details of Euler angle conversions.

3. The sound beam tends to “bunch up” at the interface with the weld near the ID surface, forming a localized region of strong signal intensity. The fusion line and the backwall act as a funnel to channel the sound energy into a small region, so PNNL termed this the “funnel effect.” The peak beam intensity at the CASS/weld interface was 7.4 dB (2.3 times) higher than in the isotropic material and 2.7 dB (1.4 times) higher than in the CASS region. The funneling phenomenon is discussed in depth in Section 4.3.

4. The region on the far side of the weld receives relatively little sound, as the weld essentially creates a “shadow.” This is expected, as the scattering effects of austenitic welds are well known.

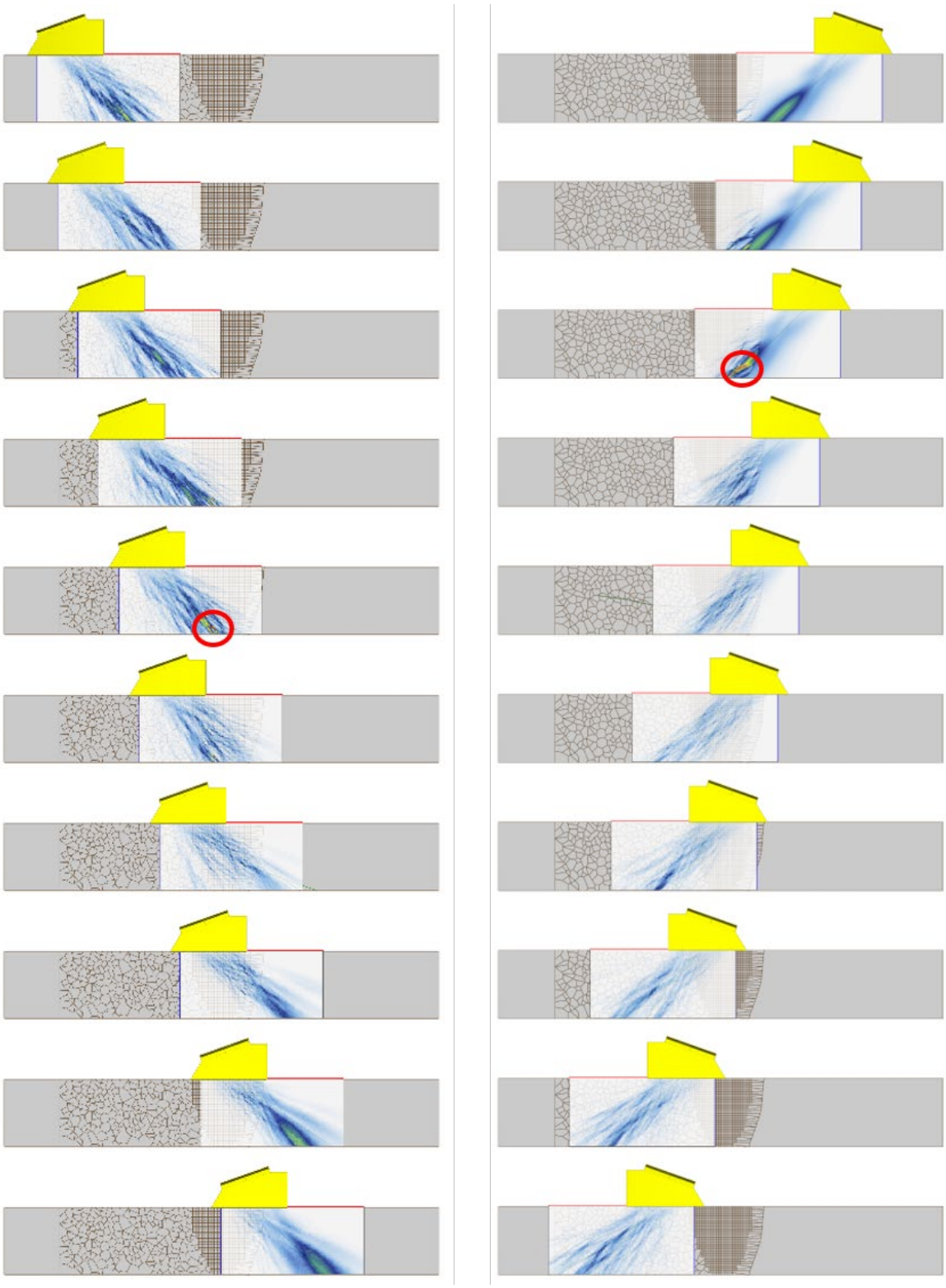

Figure 3.12. Beam simulations with a 2 MHz 45° TRL probe across the DMW model from the CASS side (left) and the CS side (right). The red circles indicate the highest beam intensity due to funneling of the sound by the weld interface and backwall. The box represents the extent of the simulation for each probe position.

Figure 3.12 illustrates the beam at discrete probe positions. During a typical scan, the probe is moved across the pipe, resulting in a continuous and cumulative sound field. To better illustrate the cumulative sound field, the simulated beam maps were combined into one image using Python[5]; see Figure 3.13. This figure more clearly shows the contrast between the scatter in the CASS (green) and weld (blue) regions. It also shows a "weld shadow" region on the far side of the weld (red bracket) that received relatively little sound intensity. The region of highest sound intensity is in the red circle. Note that this figure is a cumulation of beam simulations made at 10 mm (0.4 in.) increments; finer steps would be more realistic, but the 10 mm (0.4 in.) step size illustrates the point without excessive computation time.

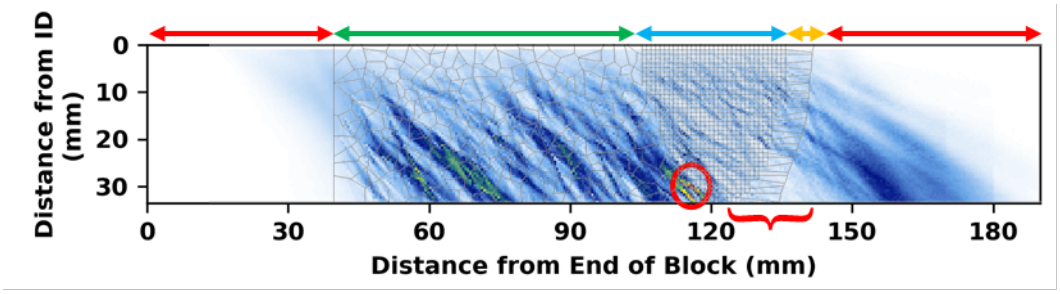

Figure 3.13. Cumulated beam maps. Arrows at the top are color coded by region the same as Figure 3.11. The circle indicates the area of peak beam intensity, and the bracket indicates the region of lowest beam intensity.

PNNL ran some flaw response simulations with the DMW model to see how the funnel and shadow effects affect the flaw response. Several 5 mm (0.2 in.), or 15% through-wall (TW), rectangular notches were placed at 10 mm (0.4 in.) intervals, assuring that one notch was at the base of the weld near the region where the sound field intensity was highest. Another notch was placed far into the isotropic material to serve as a calibration signal. Figure 3.14 shows the results, which were normalized to the highest flaw response. The flaw response at the base of the weld was 7.7 dB (2.4 times) higher than that of the flaw in the isotropic region, similar to the 7.4 dB (2.3 times) difference sound intensity seen in the beam simulations. Results confirm that the strongest echo response came from the flaw closest to the weld on the near side where the highest sound field intensity was observed. The funnel and weld shadow effects are explored in more depth in Section 4.3.

It should be noted that the simulations here were performed at 2 MHz to avoid violating the CIVA high-frequency approximation that requires the wavelength to be approximately the same dimensions or smaller than the feature size (the grain size, in this case). Extensive empirical research has shown that field examinations on this type and thickness of the material should be conducted at 1 MHz or lower to minimize scatter and maximize flaw response amplitudes (Jacob et al. 2019).

[5] CIVA 2021 has a built-in feature that can display a cumulated sound field.

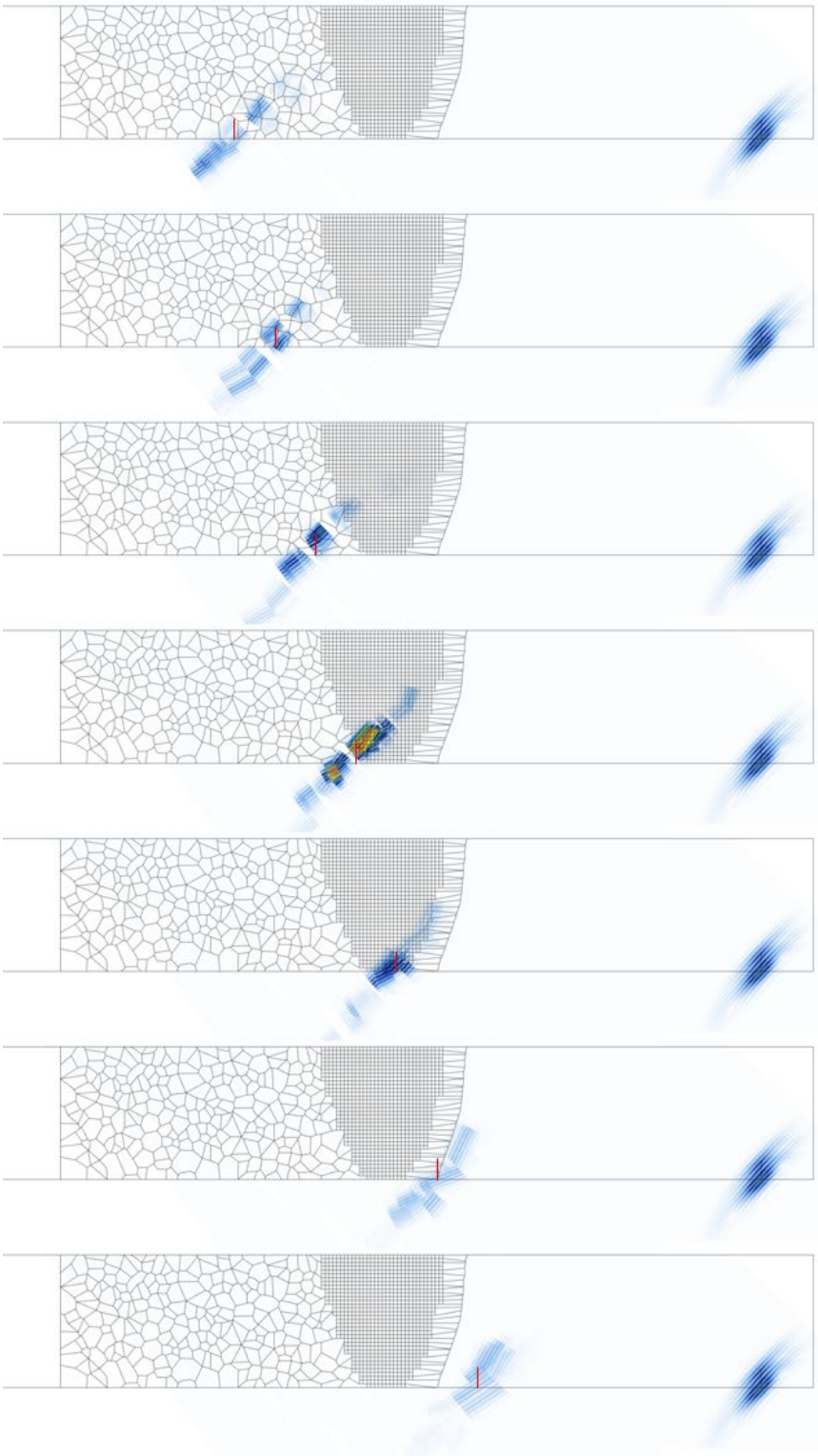

Figure 3.14. Simulated response B-scans from flaws placed at intervals in the DMW model. Red lines were added to highlight the flaw positions. The signal response to the far right is from the calibration flaw. Discontinuities in the signal responses are an artifact of CIVA's reconstruction method.

3.5 Summary

This section was a continuation of coarse-grained equiaxed modeling work described in the previous TLR, where beam simulations on CASS models were shown. In the current report, PNNL presented additional simulations on columnar CASS models. Beam simulations were used to compare a 2D CIVA model created from a section of a physical specimen to a 3D Voronoi model generated by CIVA. As with the equiaxed models, the CIVA accuracy factor was an important parameter for obtaining a complete sound field. The "crossing ratio" metric was used to measure beam segmentation caused by the columnar structure. Results showed that beam simulations using the columnar Voronoi regions with the range of velocities, ΔV, equal to 6% gave results that best agreed with those of empirical beam maps. The specimen-based model gave the worst results. In the previous report, PNNL showed that ΔV = 6% was appropriate for an equiaxed CASS model. The ideal value of ΔV will depend on model parameters and it should not be assumed that one particular value is appropriate for all scenarios.

As discussed in the previous TLR, CASS models using Voronoi regions in CIVA are much easier to produce and execute than realistic specimen-based models; Voronoi regions are inherently 3D, load rapidly, and give realistic results. However, the Voronoi option in CIVA 2020 is only available in planar or cylindrical specimen geometries, none of which include welds or multi-region models such as those drawn in the CAD tool. Models of mixed-grain structures or of materials with a wide variety of grain sizes can be created outside CIVA and imported, but then the advantages of using Voronoi regions are lost. For example, grain boundaries can be defined outside CIVA using Voronoi regions, but the final model will be 2D and extruded in the third dimension.

PNNL created DMW models outside of CIVA by combining a weld region, a CASS region, and a buttering region. The previous EBSD-based weld model was used, and a CASS region was generated outside CIVA using 2D Voronoi regions. Beam simulations showed the different effects that the weld and the CASS regions had on the sound field pattern. Beam scatter and redirection were observed in both regions, but the weld region had a stronger scattering effect. Flaw response simulations were run on the same DMW model with different flaws to illustrate that flaw response was a strong function of flaw position. Results showed that both the specimen model and the flaw location within the model had a significant impact on the predicted flaw response.

Overall, PNNL demonstrated the feasibility of an approach to creating realistic DMW models of coarse-grained materials. Models can be constructed using various methods, and beam simulations and flaw-response simulations can be run in CIVA. It is important to remember that the specimen models shown in this section are illustrative approximations and may not be good representations of every scenario. Models should be developed, changed, and adapted as needed. It is critical to remember that while CIVA appears to have demonstrated efficacy for modeling DMW and CASS under the criteria used in the study, it still is not capable of simulating low frequency UT exam scenarios that would likely be required for these exams as part of ASME Code Case N-824. The role of exam frequency is still a limiting factor.

4.0 Working with Austenitic Weld Models

4.1 Summary of Previous Work

This section is a continuation of work presented in PNNL's previous report on modeling and simulation (Jacob et al. 2020) testing austenitic weld models for simulating empirically-observed sound fields. Previous work showed that simplified weld models, such as the Ogilvy model, can provide satisfactory results in some cases (EPRI 2019). However, simplified models do not show the same degree of sound field scatter or beam redirection observed in empirical sound field maps (Gueudré et al. 2019; Holmer et al. 2017; Jacob et al. 2020; Kim et al. 2021). Another approach is to divide the weld into coarse regions with each assigned different material properties (Chassignole et al. 2000; Mahaut et al. 2007). Although this approach produces more beam scatter than the Ogilvy model, the grains are too coarse to duplicate the degree of scatter observed empirically. PNNL therefore began to develop a more complex weld model based on EBSD data. EBSD is used to measure crystalline boundaries and orientations, or Euler angles, which can be used to define the material properties of a "true-state" model (Maitland and Sitzman 2007). In the previous PNNL report, EBSD data were used to create a simplified weld model while the complete EBSD-based model was being developed. Detailed descriptions were provided of how the model was created and imported into CIVA. Beam simulations with the downsampled model showed beam scatter effects similar to that observed empirically.

In the current report, PNNL continues the work and describes beam simulations performed on the final EBSD model. Simulations were also done on variations of the weld model to determine how dependent the beam simulation results are on the specific material definitions. The first variation used different grain sizes, the second variation used random Euler angles to test the limits of a completely arbitrary model, and the third used Voronoi regions in place of the square-shaped grains to test the effects of grain shape.

4.2 Weld Model Variations

PNNL's previous work used a weld model adapted from EBSD data (Jacob et al. 2020). The models incorporated a simplified approach of binning similar Euler angles of neighboring grains in order to reduce the number of grains in the weld. This made the model more efficient to run during initial testing. One issue with extracting the Euler angles is the reference frame. The EBSD results were in the Bunge reference frame, or the reference frame of the grain, while CIVA works in a fixed frame, or the reference frame of the specimen. (Note: OnScale Solve, another modeling package that PNNL is evaluating, uses yet another reference frame.) Therefore, special care was taken to assure that the proper Euler angles were assigned and in the proper reference frames, or coordinate frames, so that the models were consistent. Appendix B describes in detail the mathematics behind Euler angle coordinate transformations. This appendix is a copy of an internal PNNL technical report describing the mathematical framework of the Euler angle conversion between different reference frames.

Multiple variations on the EBSD model were tested. Three variations used the same geometry but random Euler angle assignments in order to determine the importance of the Euler angle assignments. If the Euler angle assignments can be made arbitrarily, then generating realistic weld models will be relatively simple, since the steps of collecting EBSD data, translating reference frames, and downsampling can be skipped. A fourth variation used the same Euler angles as the EBSD model, but the grain geometries were changed from a square to a Voronoi region to address whether irregular grain sizes would produce different scattering.

4.2.1 Electron Backscatter Diffraction Weld Model

As described in Jacob et al. (2020), EBSD was used to determine grain sizes and crystalline orientations (Euler angles) in an austenitic weld section. Once the Euler angles were extracted from the EBSD data, they were mapped to the weld geometry. The weld was then decomposed, or downsampled, to models with grain sizes of 64, 128, and 256 pixels.[6] The models are shown in Figure 4.1; each color represents a 3D vector defined by the Euler angle rotations.

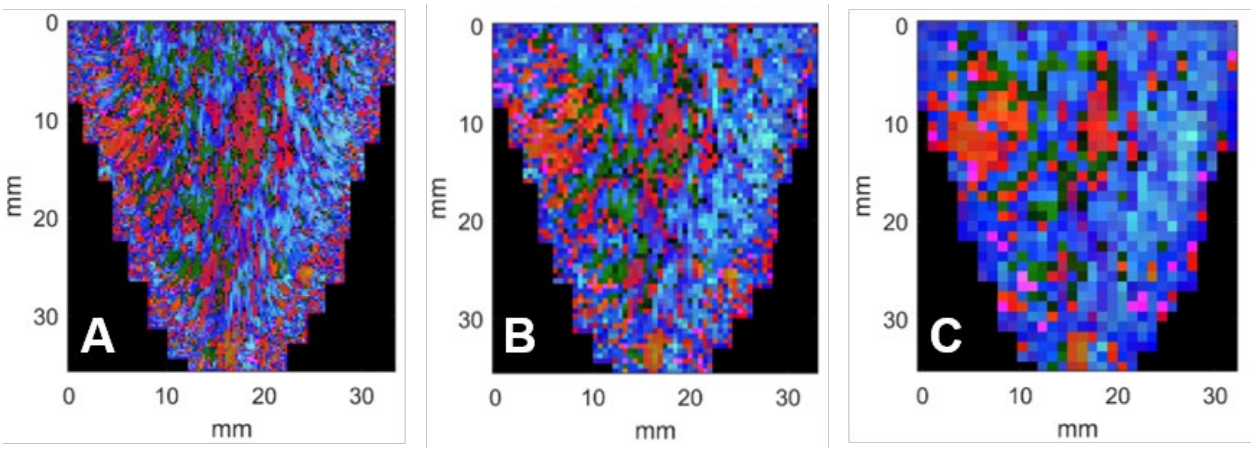

Figure 4.1. Weld models derived from the EBSD data. A: 64-pixel weld model with grains of 256 µm × 256 µm. B: 128-pixel model with 512 µm × 512 µm grains. C: 256-pixel model with 1024 µm × 1024 µm grains. (1000 µm = 1 mm = 0.0394 in.)

The geometries were written into an .xml file format compatible with CIVA 2020. The number of line segments and regions for each decomposition are shown in Table 4.1. For comparison, the numbers of segments and regions used in the 3- and 7-bin models used in Jacob et al. (2020) are also shown. Note that for these binned models the number of line segments was artificially increased to accommodate CIVA 2017's inability to handle quadpoint intersections (this issue was described in Jacob et al. (2020) and fixed with the release of CIVA 2020). As the table indicates, the complexity of the geometry increases rapidly with increasing resolution. Geometric complexity correlates with the time required for CIVA to open and process the .xml file in addition to the overall simulation time. For example, CIVA required several hours just to open and display the 64-pixel geometry.

[6] The original EBSD images were acquired at 4 µm × 4 µm resolution. "64-pixel resolution" means that 64×64 pixels from the original EBSD pixels were combined to form a single grain. The decomposition method is described in detail in Jacob et al. (2020).

Table 4.1. Number of line segments and regions in the different EBSD-based weld geometries

	Line Segments (Regions)*		
	256 pixels	**128 pixels**	**64 pixels**
3-bin	832 (272)	3193 (1064)	
7-bin	1360 (453)	5215 (1783)	
True Euler	1815 (873)	7003 (3433)	27287 (13557)

*Numbers from CIVA 2017 models

A through-weld beam was simulated using the true Euler 64-, 128-, and 256-pixel geometries and a 2 MHz, 45° TRL probe, the same probe used in the previous weld simulations (Jacob et al. 2020). The accuracy factor was set to 16. For significant time savings, a 2D computation zone was used—even so, the 64-pixel simulation took 32 hours to complete a single beam simulation. The results are shown in Figure 4.2. Qualitatively, the patterns of beam scatter in the 256- and 128-pixel scenarios are similar, although the increased scatter of the 128-pixel model reduced the overall sound intensity. There is too much scatter in the 64-pixel scenario to make a comparison. The total sound energy that passed through the weld was measured (in arbitrary units) by integrating the signal intensity along the backwall of the specimen on the far side of the weld. Results were 11, 73, and 303 for the 64-, 128-, and 256-pixel models, respectively. One way to interpret this is that the 64-pixel model effectively scattered 27 times more than the 256-pixel model, and the 128-pixel model scattered 4 times more. Scatter in CIVA is strongly affected by the number of grain interfaces, so the relatively fewer interfaces in the 256-pixel model are less detrimental to the sound field. Based on a qualitative assessment of empirical scans, the 256-pixel model appears to work well at 2 MHz. The grain sizes may need to be adjusted for different probe frequencies, and that will have to be approached by trial-and-error. At 2 MHz, the 64-pixel model is pushing the limits of the high-frequency approximation that CIVA uses, since the grain size is about 10% of the wavelength; however, this model may be appropriate for higher frequencies.

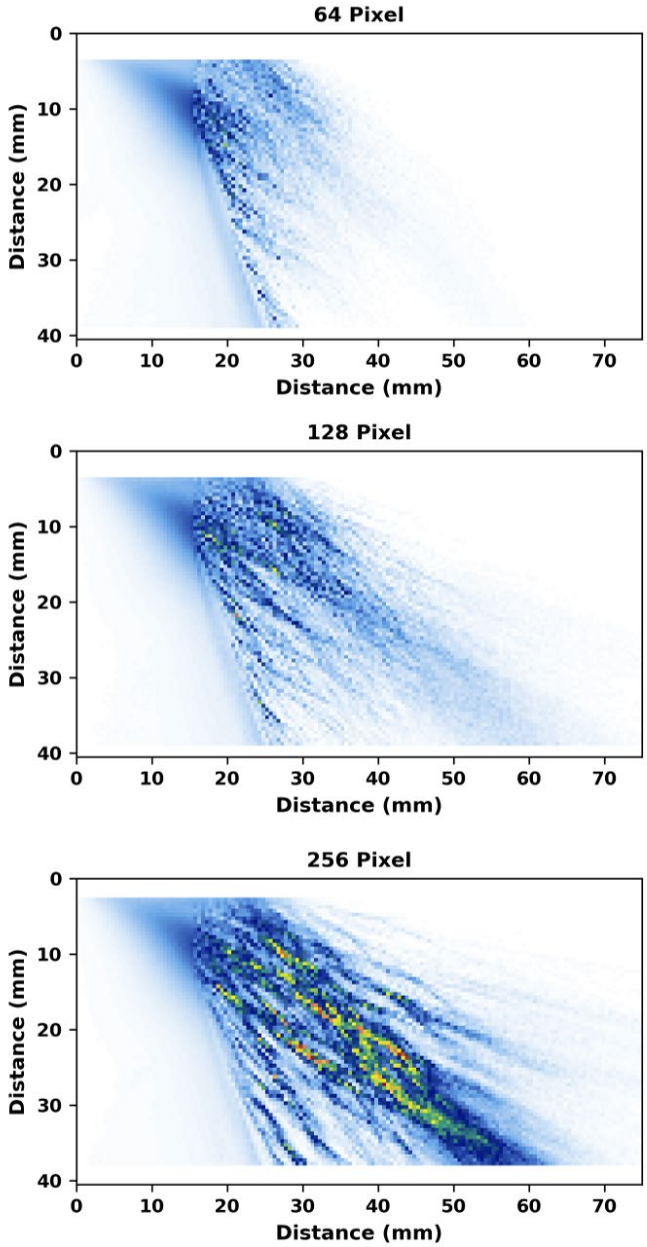

Figure 4.2. Beam simulations with the 64-, 128-, and 256-pixel weld models.

Empirical beam maps were acquired using a laser vibrometer (Castellini et al. 2006). The laser vibrometer, shown in Figure 4.3, can map the sound field along a side of a specimen; the side needs to be smooth but can be unpolished. A TRL probe is placed such that only the transmit side is in contact with the specimen and the plane between the transmit and receive elements aligns with the specimen's face. A weight can be placed atop the probe for stability and to improve coupling. To optimize reflectivity, a surface membrane such as retro-reflective tape can be applied to the specimen face.[7] Beam maps were acquired on two specimens. The first was the same that was used to acquire the EBSD data, Specimen 3C-022-03 in Jacob et al. (2020).

[7] Retro-reflective tape has a prismatic coating that reflects light back to the source regardless of the angle of incidence.

This specimen comprised two 36 mm (1.4 in.) thick 304 WSS piping sections welded together with 308/308L stainless steel (SS) weld material. The second was a 76 mm (3.0 in.) thick 304 SS plate mockup described as Specimen 19C-358-1 in Harrison et al. (2020). The plate comprised two flat pieces welded together using 308/308L SS weld material. The focal depth of the probe was only 35 mm (1.4 in.), so the thicker specimen was chosen in order to show the entire sound field. Two beam maps were acquired on each specimen: one through parent material and one through the weld. The same 2 MHz probe that was modeled for simulations was used for the empirical maps.

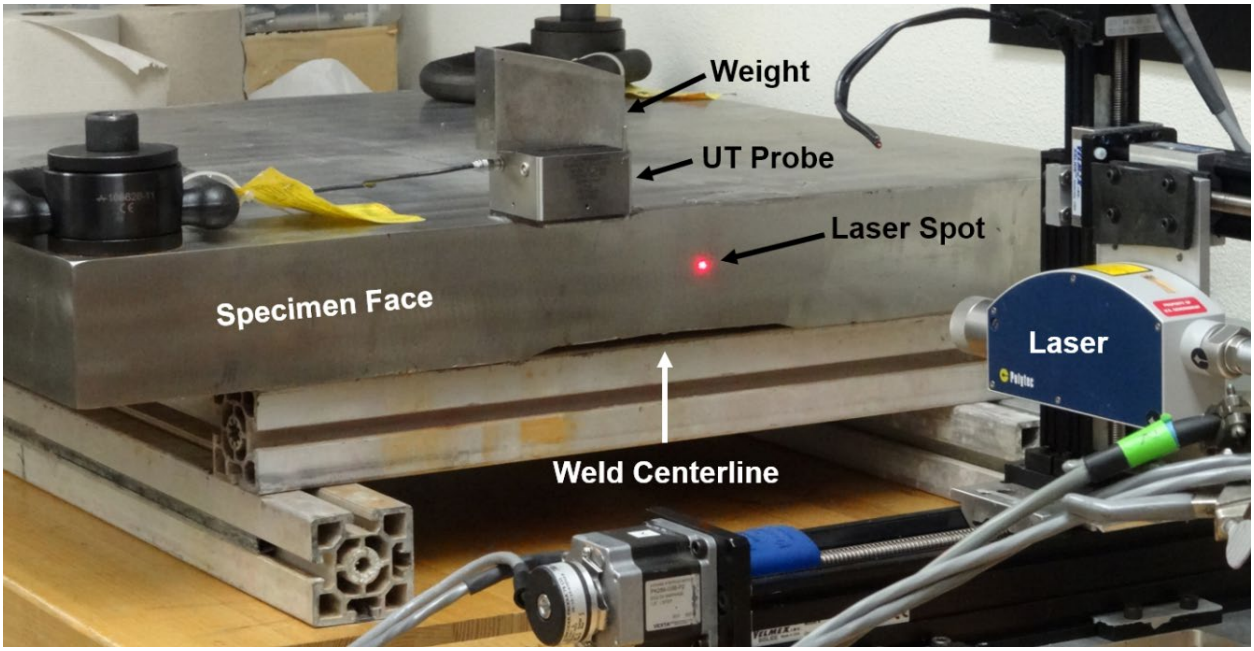

Figure 4.3. The setup used for laser vibrometry measurements.

The empirical beam maps are shown in Figure 4.4. The top row shows beam maps from 3C-022-3 in parent material (left) and through the weld (right). The sound beam is scattered through the weld, but the amount of sound energy incident on the far-side backwall is appreciable. This is more akin to the results of the 256-pixel weld model than those of the other weld models. The bottom row shows the same sound beam through the thicker specimen. In this case, the sound field is almost entirely scattered by the weld with very little sound making it through, and the 256-pixel simulation results do not agree as well with the empirical results. However, the weld and specimen model were significantly different from the laboratory mockup, so it would be more surprising if the results did agree. The beam maps illustrate that different welds can have significantly different effects on the degree of scatter and the importance of using a model appropriate to the scenario.

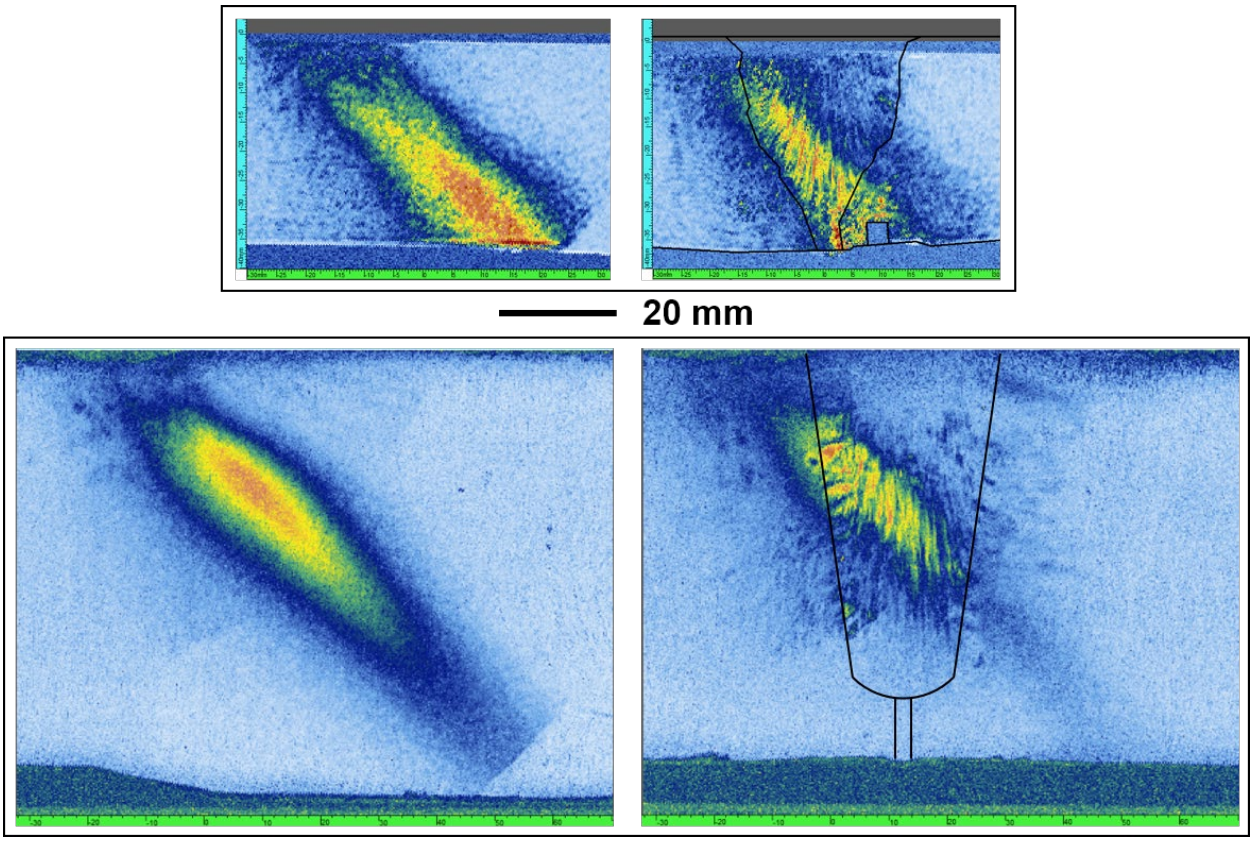

Figure 4.4. Empirical beam maps through austenitic welds. Top: Specimen 3C-022-3. Bottom: Specimen 19C-358-1. The probe has a 35 mm (1.4 in.) focal depth, so minimal backwall insonification is expected in the thick specimen, even without the weld. Left images are beam maps through parent material. Right images are beam maps through the welds. All images are to scale.

4.2.2 Random Euler Angle Welds

Weld models based on EBSD data are destructive, costly, and time-consuming to create. They require a cut and polished section of the weld, access to EBSD equipment, and computational expertise to compile the model .xml file, to translate the Euler angles into the correct coordinate system, and to execute the simulation. The EBSD-based models are useful for research and development, but they are impractical for routine simulation work. It is therefore important to identify what, if any, components of the model can be simplified or eliminated. For example, if the key effect of the weld is beam scatter rather than beam redirection, then random Euler angles may produce simulation results comparable to those of the EBSD-based angles. An arbitrary weld model could then be made without need for specific grain information, eliminating much of the time and expense of generating new models and allowing multiple scenarios to be tested without starting each model from scratch.

PNNL demonstrated in Figure 4.2 that changing the grain size (while keeping the Euler angles the same) has a profound impact on the beam scatter as well as simulation time. Here, the effects are tested of changing the Euler angles but maintaining a constant grain size. PNNL created a weld identical to the 256-pixel EBSD weld but with random Euler angle assignments; see Figure 4.5. Each set of Euler angles is represented by a different color. Three random weld models were generated.

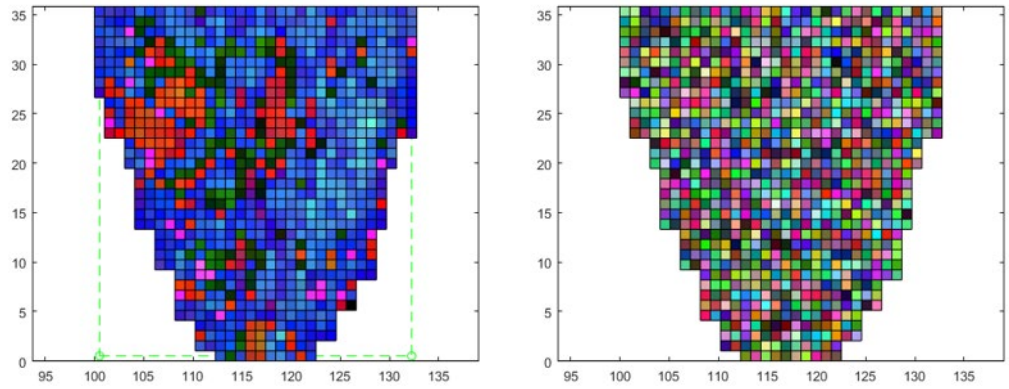

Figure 4.5. Example of a random weld model. Left: EBSD-based 256-pixel weld model. Right: Weld model with random Euler angles.

Figure 4.6 shows CIVA beam simulations with the random weld models using the 2 MHz, 45° TRL probe; the probe was in the same position for each simulation. Based on the qualitative appearance of the scatter, there are noticeable but slight variations between the different weld simulations. Comparing the results of using finer grains versus using random Euler angles, PNNL concludes that the number of interfaces has a stronger effect on scatter than the specific Euler angle assignments (assuming enough variability in Euler angles). Note that PNNL has created only one EBSD-based weld model, so the comparisons here are indicative but not statistically meaningful.

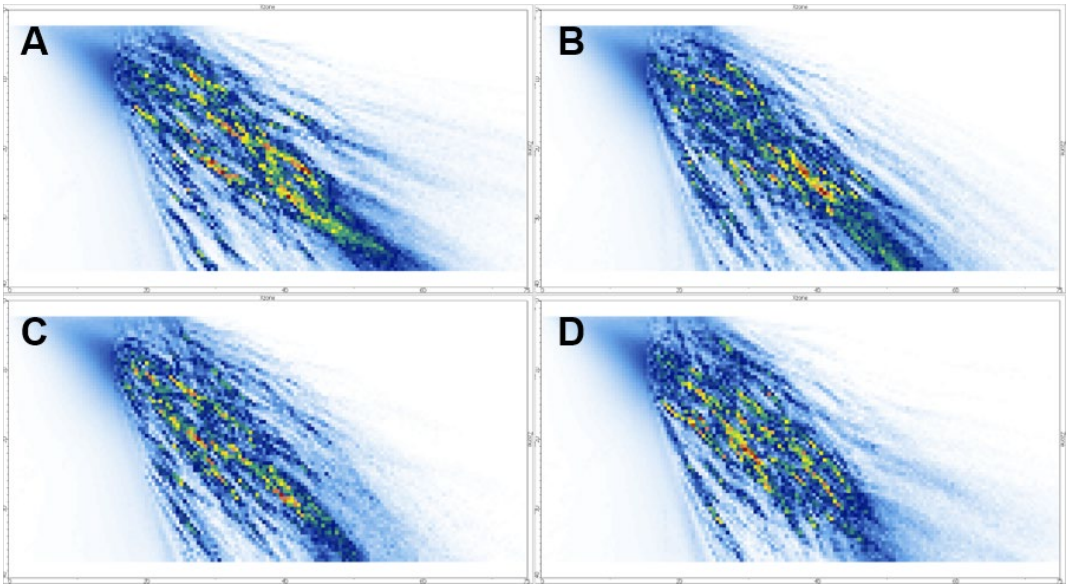

Figure 4.6. Beam simulations through different weld models. A) EBSD weld, B) random weld #1, C) random weld #2, and D) random weld #3.

Additional simulations were run on each weld model in a “dynamic” scenario to visualize the cumulated beam acquired from multiple probe positions. The probe was started to the left of the weld such that the entire beam was in the parent material and was stopped nearly past the weld such that the sound path at the last probe position was initially through the weld for a short distance but was focused in the parent material. The probe increment between simulations was

10 mm (0.4 in.). Figure 4.7 shows the results of adding the beam simulations together in Python (the Voronoi simulation at the bottom is discussed in Section 4.2.3). All simulations were identical to the left of the weld (as expected) and show significant, and essentially random, scatter through the weld. Some differences can be seen to the far right of the weld. In the EBSD weld, the rightmost beam looks largely undisturbed, perhaps with some slight redirection. On the other hand, the random welds appear to scatter more strongly with less sound reaching the right side. It is important to remember that these beam simulations are 2D and do not show what is happening to the beam outside the simulation plane.

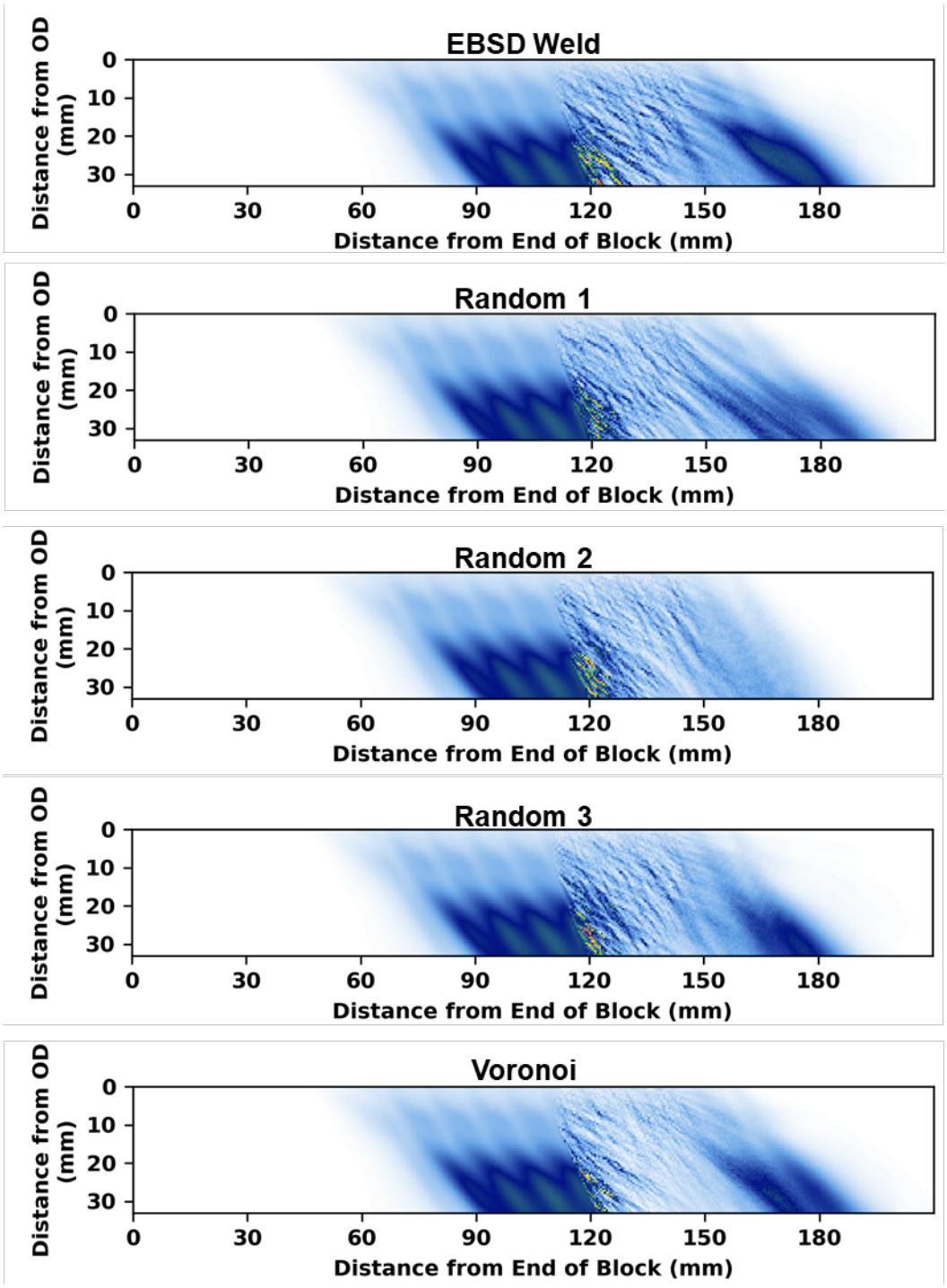

Figure 4.7. Cumulated beam maps using the EBSD weld model and other weld models.

4.2.3 Voronoi Weld Model

A valid criticism of the weld model is that actual grains are not square, and the square grains may induce unrealistic beam scatter. To address this, a Voronoi weld model was created in MATLAB from the 256-pixel EBSD weld model. The original model was used as a template to generate a new model with the same average grain size and the same Euler angle assignments

as the original, but with irregularly shaped grains. Seed points were randomly placed, one point in each square, and Voronoi regions were grown from the seed points. Figure 4.8 shows the original model with randomly-placed points (left) and the resulting Voronoi regions (right). This is similar to the approach used by (Van Pamel et al. 2015) to create a polycrystalline model. The Euler angles were then transferred from the squares to the respective Voronoi regions. The regions outside the weld boundary (shaded area of the figure) were truncated or removed for ease of generating the model, since some Voronoi regions grew into the parent material or outside the specimen boundaries. An important difference between the models is the number of grain interfaces. The original model had 1,815 interfaces and the new Voronoi model required 2,579 interfaces. This is a 42% increase, which increases the time required for CIVA to process the geometry and run the simulation.

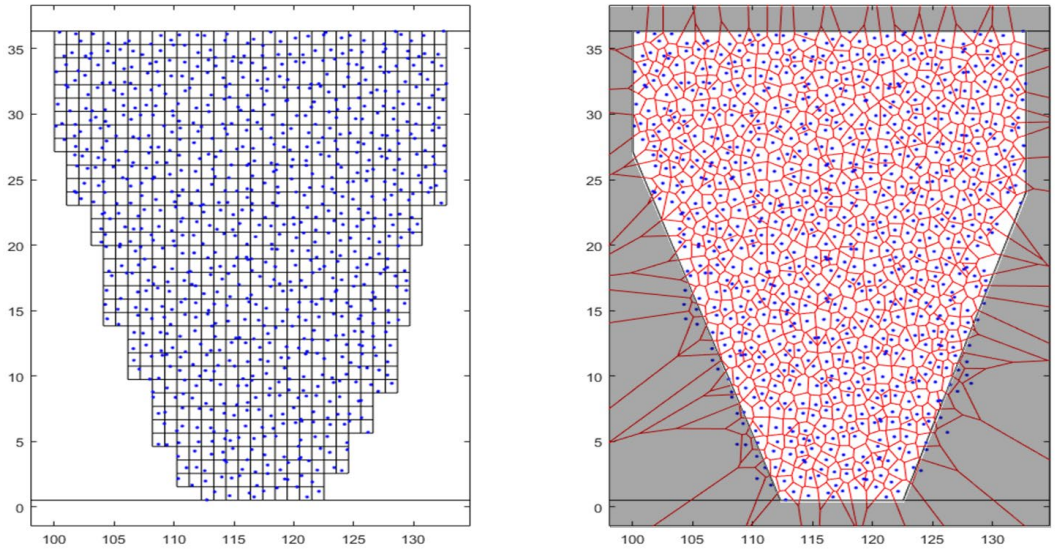

Figure 4.8. A Voronoi weld model was created from the 256-pixel weld model. Left: The 256-pixel model with one seed point placed randomly in each square. Right: A weld model with Voronoi regions grown from the seed points. The regions outside the weld boundary (i.e., in the shaded area) were removed.

To compare the new Voronoi model to the previous model, beam maps were simulated using the same 2 MHz, 45° TRL probe as before. All simulation parameters were identical to those of previous simulations except for the weld geometry. The simulation results are shown in Figure 4.9. The left panel shows the original weld model, and the right shows the Voronoi model. Both panels are normalized to the peak signal intensity of the original model. The Voronoi weld induces much more scatter than the original model, as indicated by the lower overall signal intensity, because of the higher number of interfaces.

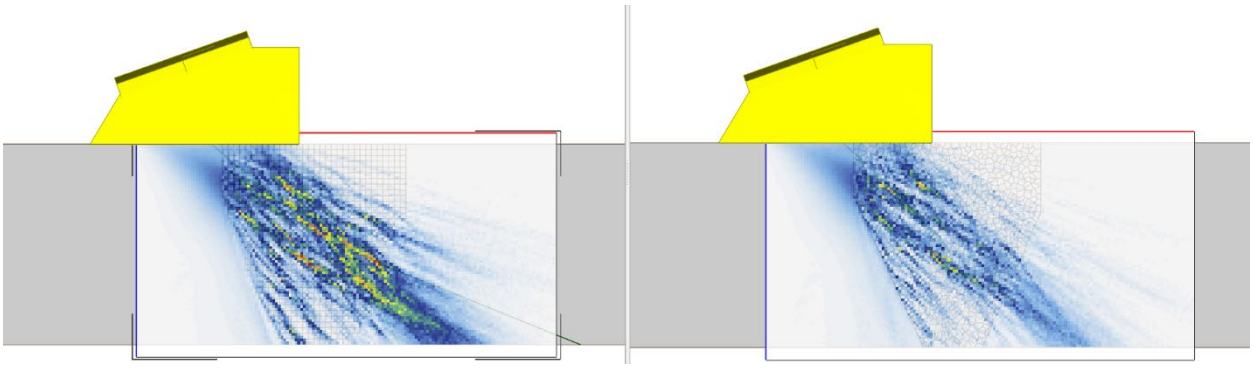

Figure 4.9. Simulation results using the original EBSD weld model and the Voronoi weld model. Left: The original EBSD model that used square grains. Right: The new Voronoi model.

In the second simulation with the Voronoi model, the probe was stepped at 10 mm (0.4 in.) increments across the specimen, and a beam simulation was performed at each position. Figure 4.7 shows the beam simulations for the original weld model (top) and the Voronoi weld model (bottom). Of the models shown in the figure, the Voronoi weld caused the most scatter, and therefore the lowest sound intensity penetrated through the weld. Qualitative comparisons to empirical results suggest that the Voronoi weld model scatters too strongly.

There is little to be gained by trying to make the original weld model more “realistic” by using Voronoi regions. However, this exercise provided some additional insight into how the number of interfaces affects scatter. Scatter occurs as sound reflects off the interfaces and refracts through the grains, so more interfaces means more scatter. It is important to strike a balance between the number/size of grains and the number of grain boundaries, or interfaces. Although square grains are not realistic, they have several advantages over other grain shapes:

- Square grains have a low number of grain boundaries for a given number of grains (only a triangle tessellation would have fewer boundaries), which facilitates faster computation in CIVA;
- Models of higher or lower resolution can be tested because downsampling and upsampling are readily accomplished with square grains;
- Models with square regions are easier to create and modify because the computer code is simple relative to what would be needed for other shapes;
- Correlating material properties from EBSD data to a grid (square grains) simplifies defining and maintaining grid ordering in CIVA.

Whatever the model or grain pattern one wishes to use, PNNL recommends a weld model that uses appropriate grain sizes (i.e., that reflects the empirically-measured grain sizes while not violating modeling software assumptions) and minimizes the number of line segments, or interfaces. For the models tested, such an approach gives a realistic amount of scatter while minimizing the model complexity and simulation time.

4.2.4 Polycrystalline Weld

Another potential method of creating a weld model is to take advantage of the built-in "polycrystalline" material feature of CIVA. Although PNNL has not explored the polycrystalline option in depth, it is designed to model specimens comprising multiple crystals or grains. The user can define an average grain size (but not a range of grain sizes), an elongation factor (for non-equiaxed grains), and an orientation. The polycrystalline approach has the *potential* to provide a good option for simulating austenitic welds. For example, the user could create a weld with, say, 10 regions of dominant grain orientation, such as shown in Kim et al. (2021) or Chassignole et al. (2000), then define different polycrystalline properties (including orientation) of each region. This approach might be useful for creating a realistic weld model that is much faster to simulate than the EBSD-based model and more realistic than the fast but overly simplistic Ogilvy model. In limited testing, PNNL did not achieve satisfactory results with the polycrystalline weld approach, but it is premature to dismiss it as a possibility.

4.3 The Funnel and Shadow Effects

An interesting effect was observed in beam simulations shown in Sections 3.4 and 4.2. Beam simulations showed that the sound field tended to intensify at or near the base of the weld nearest the probe, as if sound was being "funneled" into that region by the backwall and weld fusion line. PNNL refers to this as the "funnel" effect, where the weld interface and specimen backwall act as a sort of funnel for redirecting and concentrating the sound. Figure 4.10 illustrates the effect, where sound energy enters the funnel region, resulting in stronger local intensity. The effect was observed in simulations shown in Figure 3.13 and Figure 4.7.

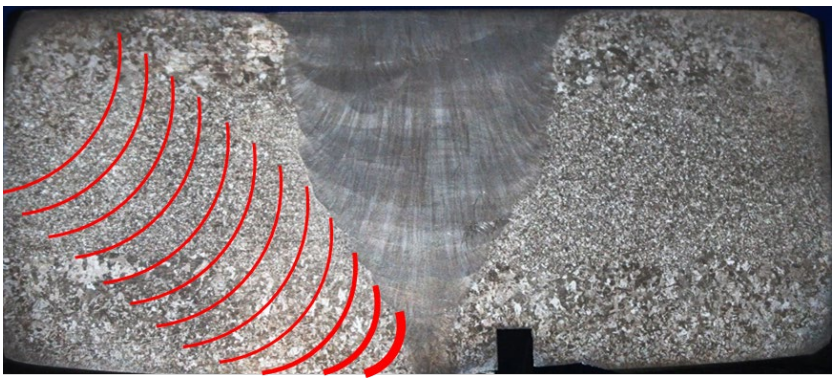

Figure 4.10. Schematic illustrating the funnel effect. Incoming sound energy is funneled by the weld interface and specimen backwall, resulting in an increase in sound energy density near the base of the weld.

A second effect observed in the beam simulations is that scatter from the weld casts a sound field "shadow" on the far side of the weld, resulting in a region of relatively weak sound intensity This "shadow effect" is a well-known phenomenon in austenitic weld inspections (Anderson et al. 2011; Crawford et al. 2009). It was also observed in Figure 4.7 for all cases and was strongest for the Voronoi case, where scatter was also strongest.

Both the funnel and shadow effects were observed (but not commented on) in an empirical and simulated data set in another study of the effects of material properties in UT simulations of austenitic welds (Gueudré et al. 2019). To confirm the effects, PNNL acquired empirical beam maps using the laser vibrometer on the same specimen from which the EBSD weld was

obtained. 2 MHz, 45° and 60° TRL probes were used. The 45° probe is the same that was used for the weld simulations, with a focal depth of 35 mm (1.4 in.). The 60° probe is nearly identical but has a focal depth of 42 mm (1.7 in.). The probes were placed at five intervals of 5 mm (0.2 in.) with the central position chosen such that the sound beam would be directly incident where the weld meets the specimen backwall. Figure 4.11 shows the cut and polished section of the weld and a beam map with the 60° probe. The UltraVision timing gates were opened so that the entire sound field was visible. This makes the longitudinal-to-transverse (L-T) mode converted signal appear prominently to the left of the intended longitudinal-wave beam. It also highlights what happens to sound within the weld region. In effect, sound that reflects or refracts into the weld ends up bouncing around like a pinball, which is interesting because it highlights the weld region distinctly in the scan. The timing gates can be narrowed to help isolate the longitudinal-wave portion of the incident beam.

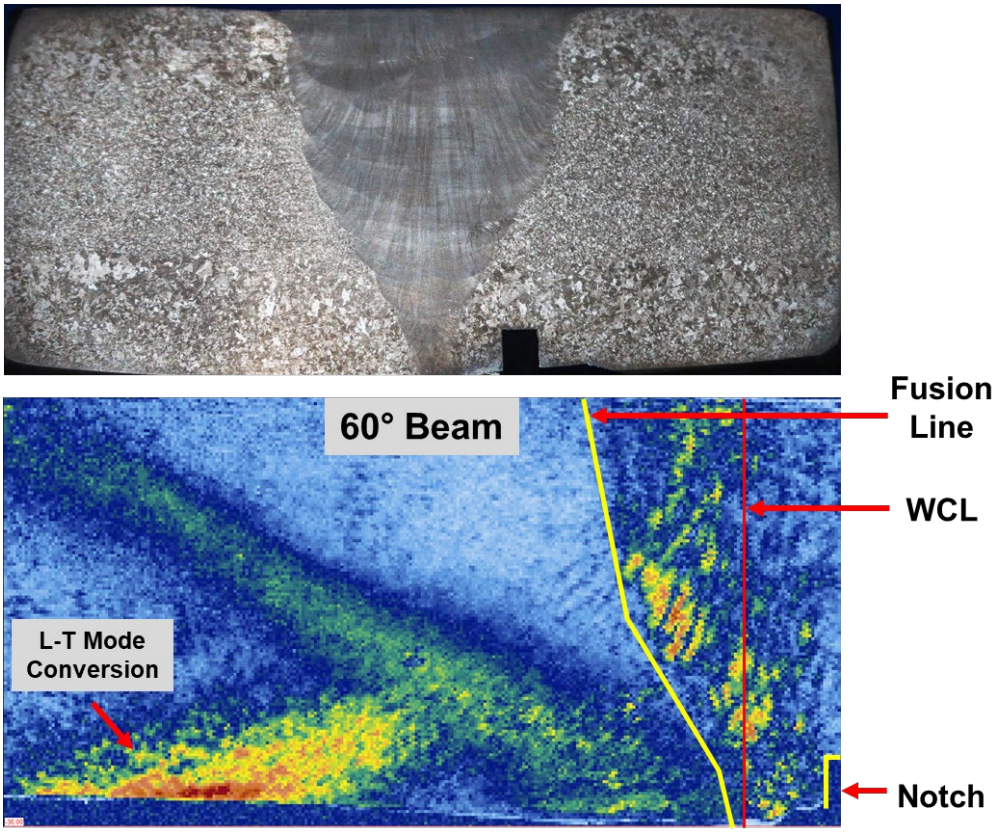

Figure 4.11. Example of an empirical side-beam map. Top: cut and polished section showing the geometry of the weld. Bottom: Beam map with the 60° probe. The weld fusion line (yellow), weld centerline (red), and notch are indicated in the beam map.

Figure 4.12 shows the 45° (left) and 60° (right) side-beam maps. For the 45° maps, the funnel effect is observed in the bottom two panels just to the right of the fusion line where the beam signal intensity is the highest (red circle). The effect appears to be less prominent in the 60° maps. The shadow effect has been shown empirically in multiple beam maps, including Figure 4.4 above and in Dib et al. (2018) and Jacob et al. (2020).

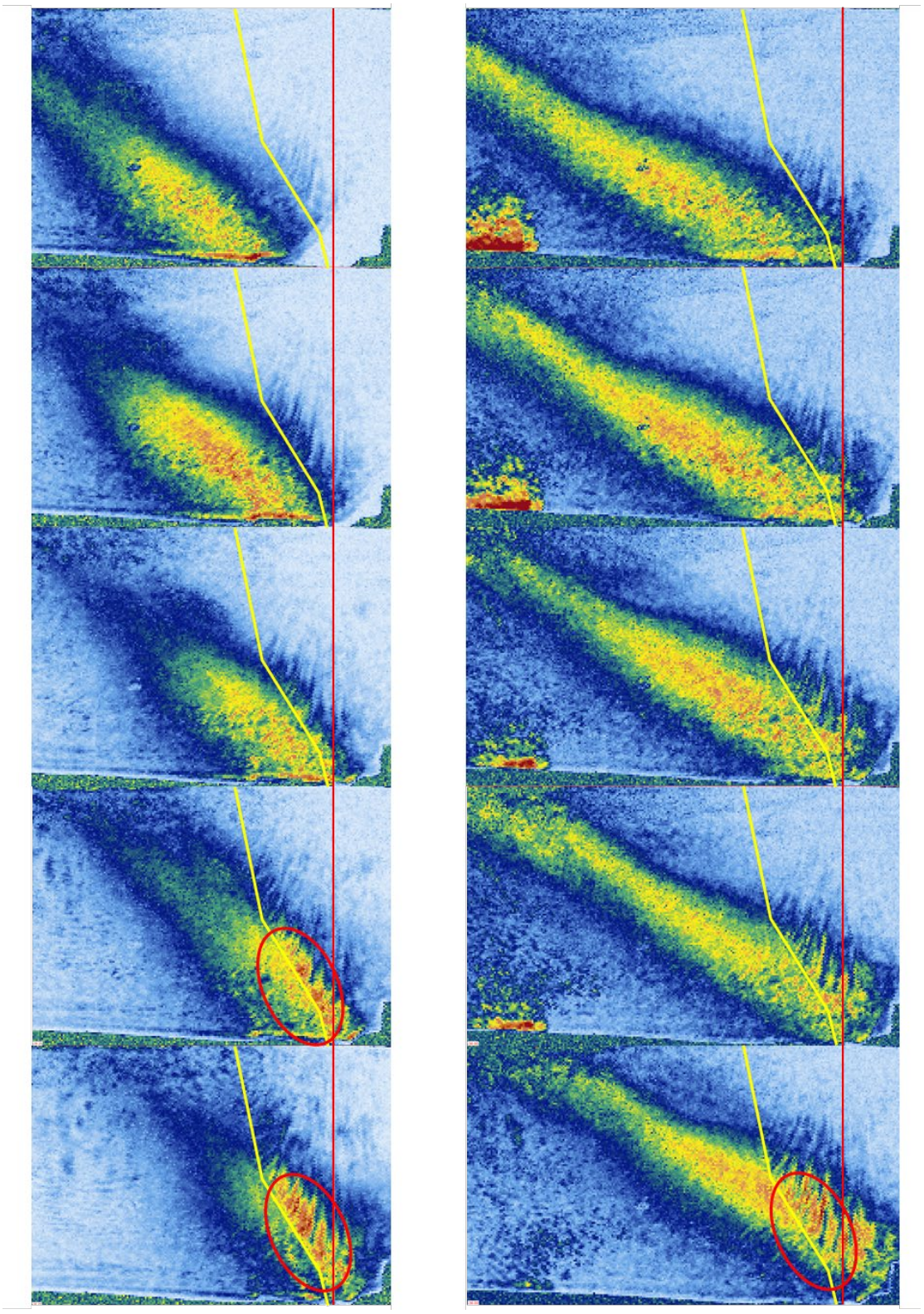

Figure 4.12. Beam maps acquired at 5 mm intervals with a 45° TRL probe (left) and the 60° probe (right). The circled regions show the highest levels of the funnel effect. The vertical red lines indicate the weld centerline.

PNNL considered how the two effects might affect flaw detection. The funnel effect should cause the highest signal response from a near-side flaw located at the base of the weld approximately between the fusion line and weld centerline. The shadow effect should cause flaws on the far side of the weld to have a relatively low signal response; difficulty with far-side detection is well established, but far-side detections will be included in this analysis for completeness. Empirical flaw response data were acquired on notches that were placed in a welded plate. The plate, labeled 19C-358-2 and described in Harrison et al. (2020), was 32 mm (1.25 in.) thick and had a 3 mm (0.13 in.) counterbore. This plate contains four 8.4 mm (0.33 in.) deep, or 23% TW, notches positioned at 0 mm, 4 mm, 8 mm, and 21 mm (0 in., 0.16 in., 0.31 in., and 0.83 in.) from the weld centerline. Scans were acquired with several probes from both sides of the weld, resulting in a full complement of near- and far-side detection data. The probes are described in detail in Section 4 of Harrison et al. (2020). The probes used here were the 2 MHz phased array (PA) at 45° and 60°, GEIT (General Electric Sensing & Inspection Technologies) 2M-45L (2 MHz, 45° TRL), GEIT 2M-60L (2 MHz, 60° TRL), SNI (Sensor Networks, Inc.) 2M-45L (2 MHz, 45° TRL), and SNI 2M-60L (2 MHz, 60° TRL).

For comparative simulations in CIVA, a specimen was created with the EBSD weld model flanked by isotropic steel. Additional similar models were made with the three random welds and the Voronoi weld. Seven planar flaws were placed in the models at the same positions as in the mockup, namely at 0, ±4, ±8, and ±21 mm (0, ±0.16, ±0.31, and ±0.83 in.) from the weld centerline. Flaws were 8.4 mm (0.33 in.) deep, or 23% TW, to match those of the empirical specimen. Figure 4.13 illustrates the plate model as drawn in CIVA. Flaw 1 acted as a control or baseline since it was unaffected by the weld (when scanning from left to right). It is important to note that, although the flaw spacing is precise, *the weld geometry is approximated with respect to that of the actual plate*. Therefore, the relative positions of flaws with respect to the weld fusion line may vary from those in the laboratory mockup. Simulations were run as a CIVA variation, or parametric study, with one flaw activated for each variation. This approach allowed all the simulations to be run at once and avoided flaw response signals from interfering with one another.

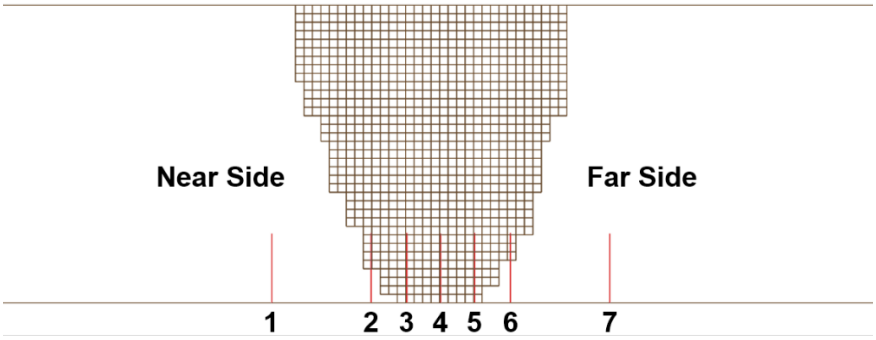

Figure 4.13. Model of the welded plate with seven flaws used for simulations.

Simulations were run using the same six probe definitions that were used for the empirical scans. For the Voronoi weld, only the two SNI probes were simulated due to time constraints. The Half Skip mode was used in 2D with no mode conversions or specimen echoes. Simulation results were exported from CIVA, and the total and peak echo responses were calculated in Python. The peak echo response was measured from the empirical results for comparison. Thus, both the total and peak simulation results were measured to see which simulated response had a better correlation with the peak empirical data. Note that the empirical scans

were 3D, whereas the simulations were 2D, so the empirical and simulated results are not directly analogous.

Studying the funnel effect has led to several observations that have important implications for future weld simulations. These observations are discussed in the subsections below. First, the empirical results from the different probes will be compared, and average flaw responses will be calculated for the 45° and 60° probes. Next, two methods of measuring the echoes in CIVA—the peak echo responses versus the integrated echo responses—will be compared to one another and to the empirical averages. Third, simulation results from different weld models will be examined. Finally, simulation results from the different probes will be compared.

4.3.1 Comparison of Empirical Scans

First, the results of the empirical scans are examined. Figure 4.14 shows the peak flaw response in UltraVision versus flaw position for the three 45° (left) and 60° (right) probes, with the averages indicated by the red lines. Each point represents a flaw, and each dataset was normalized to the baseline signal intensity of the leftmost flaw, i.e., the flaw that was unaffected by the weld. Thus, all of the datasets start with a baseline value of 1. There are a couple of important takeaways from these plots.

First, the signal intensity increases above the baseline signal as the flaw gets closer to the weld. The signal reaches a maximum at the second or third flaw (at the −8 mm or −4 mm position). The echo signal is above the baseline when the flaw is near the base of the weld on the near side, which implies a stronger sound field intensity at that position. This is experimental confirmation of the funnel effect.

Second, the signal diminishes to below the baseline as the flaw moves through the weld to the far side. This is experimental confirmation of the shadow effect, where weaker sound intensities on the far side of the weld produce weaker flaw echoes. Interestingly, the decrease in signal is not steady in the 60° scans, as there is a slight bump or increase in signal for the sixth flaw. The increase was observed for all three probes, and the reasons for it are unclear.

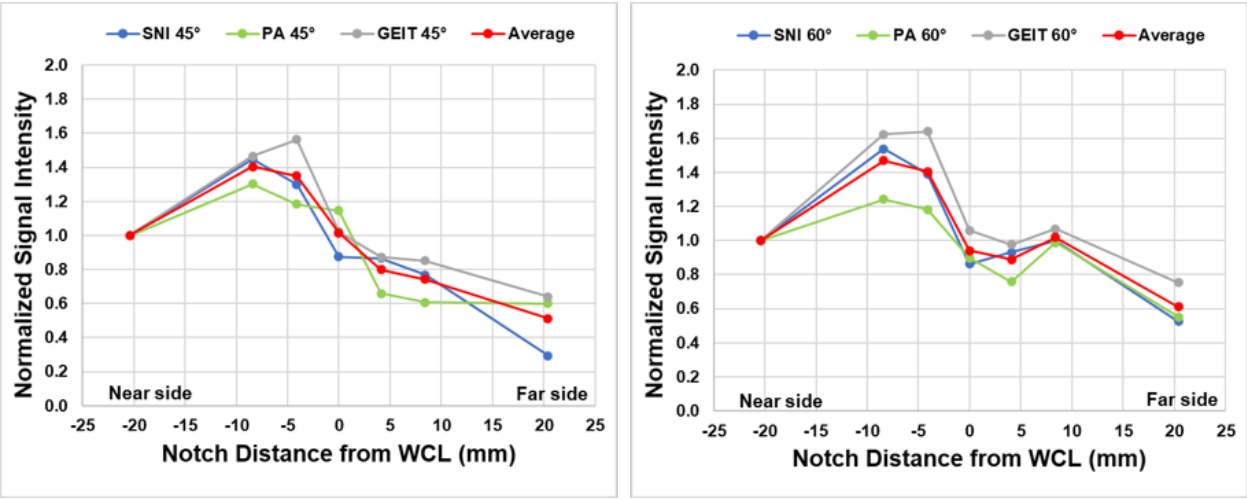

Figure 4.14. Graphs showing the normalized signal intensity versus flaw position for the 45° (left) and 60° (right) empirical scans.

A practical consequence of the funnel effect is the potential of having to recalibrate the hard gain during a scan. For example, the gain is typically set to 80% full-screen-height (FSH) based on a baseline signal, such as a corner-trap echo (analogous to the leftmost flaw). Thus, any flaw signal that peaks at more than 125% of the baseline will be saturated. The data show flaw signals as high as about 160% of the baseline for the flaws near the base of the weld.

4.3.2 Integrated Signal Response vs Peak Signal Response

Simulated data were analyzed two ways: by measuring the total, or integrated, echo response and by measuring the peak echo signal. There were no simulated noise or mode-converted signals, so calculating the integrated signal was a simple matter of summation. Both methods were investigated for all the 45° and 60° simulations and compared to the empirical results. For the empirical data, the conventional method of simply measuring the peak echo response was used. Figure 4.15 shows an example of the results with the GEIT probes. The red lines are the empirical averages shown in Figure 4.14.

The empirical results (red lines) appear to agree better with the integrated simulation results (left column) than they do with the peak simulation results (right column). This was also observed with the SNI TRL probes, but less so for the PA probe. The Pearson's correlation coefficient R was calculated to compare each of the integrated and peak simulation results to the empirical data. A Fisher z-transformation was then applied to obtain an average correlation coefficient R_z.[8] Table 4.2 shows how the empirical data correlated with the simulated data (R_z=1 is perfect correlation). Overall, the integrated data correlate better to the empirical data. On balance, the CIVA peak measurements tend to underestimate actual flaw responses and are not as good at matching the trend of the empirical responses. Therefore, all results shown in the rest of this section will use the integrated CIVA signal. Note, however, that CIVA 2020 does not have an integration function, so the data were exported and the integration was done in Python.

Table 4.2. Correlations between empirical data and integrated or peak simulated data.

	GEIT 45L	GEIT 60L	SNI 45L	SNI 60L	PA45	PA60
R_z Integral	0.901	0.815	0.928	0.719	0.821	0.822
R_z Peak	0.783	0.681	0.857	0.630	0.840	0.770

It should be noted that empirical data acquisition in UltraVision assumes a constant sound speed, and UltraVision reconstructs the data based on straight-line sound paths, since it has no information about grain structures. However, CIVA reconstructs B-scans by using ray paths as opposed to assuming a constant sound speed (see Appendix A.1 for a detailed explanation), and CIVA reconstructions tend to be choppy with signal more spread out, possibly resulting in lower peak measurements. Thus, measuring the peak signal in CIVA with complex geometries has the potential to give inconsistent results.

[8] https://en.wikipedia.org/wiki/Fisher_transformation

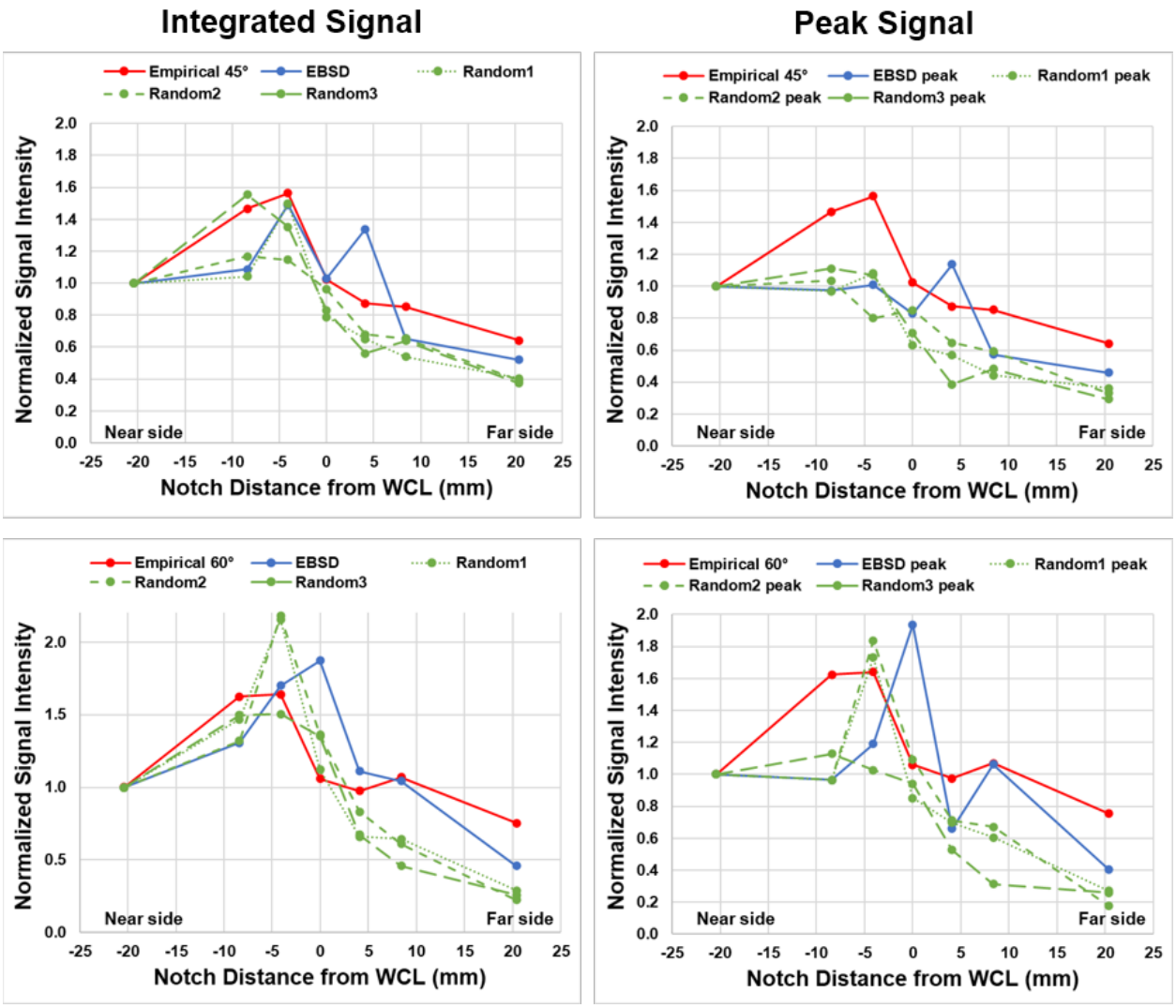

Figure 4.15. Normalized simulated and empirical echo responses versus flaw position. This plot shows an example of simulated data from the GEIT 45° (top row) and 60° (bottom row) probes. The left column is the integrated B-scan signal, and the right row is the peak B-scan signal. The red lines are the empirical data from the GEIT probe.

4.3.3 Comparison of Weld Models

Next, the differences between the weld models are examined. Recall that the original weld model was based on EBSD measurements, and subsequent models used the same geometry but were given random Euler angle assignments. It is important to remember that the EBSD measurements were made from a different welded mockup, so the granular structure is likely different from that of the plate mockup used here. Thus, the EBSD-based weld model is being used as an approximation for a general austenitic weld.

Figure 4.16 shows empirical and integrated simulation results from all six probe scenarios along with the correlations to the empirical results. The left column is the 45° data, and the right column is the 60° data. The red lines show the empirical results. Recall that the Voronoi weld model was run with the SNI probes only.

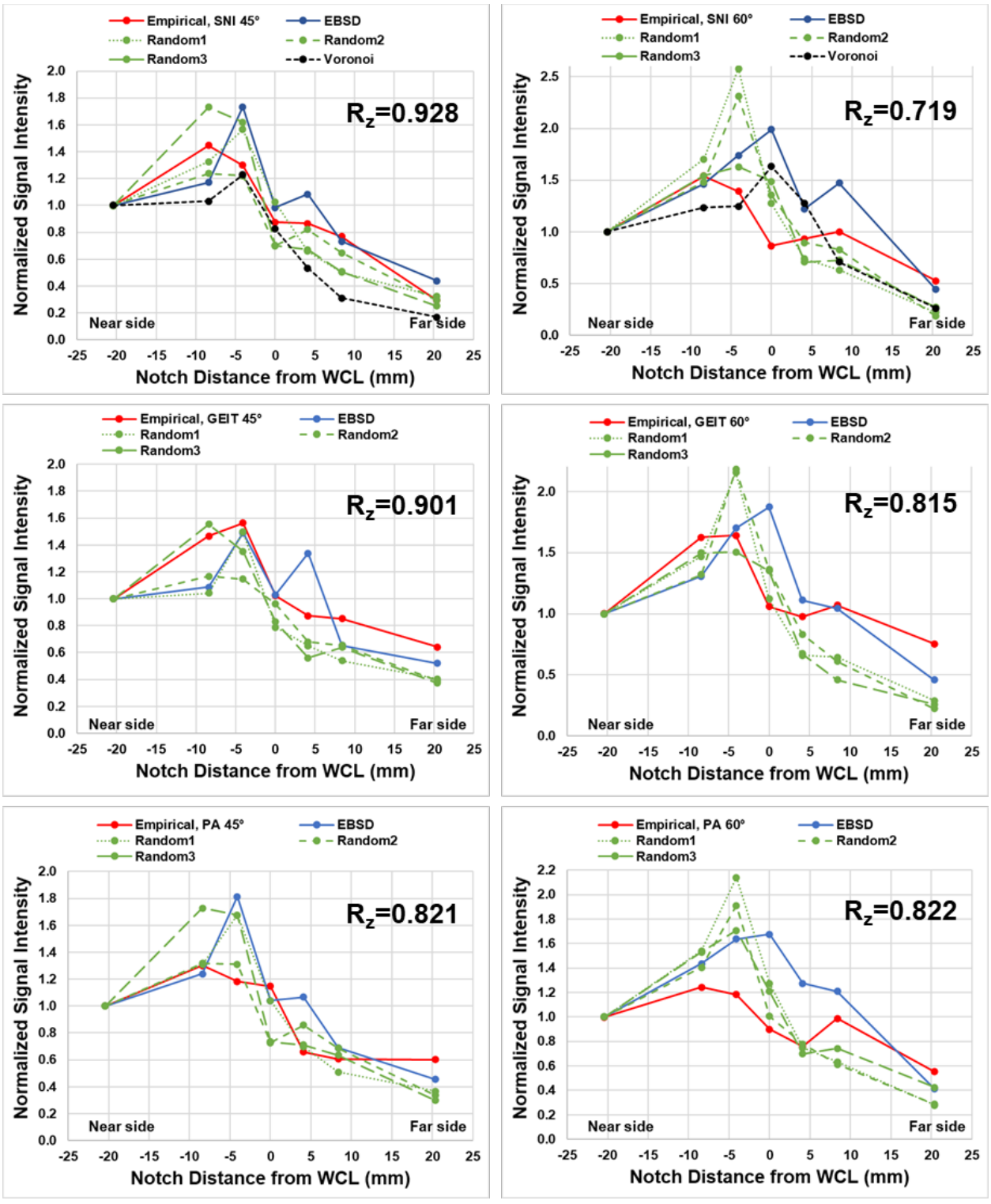

Figure 4.16. Normalized simulated and empirical signal response versus flaw position for the SNI (top row), GEIT (middle row), and PA (bottom row) probes.

There are several points to note here.

- The 45° simulated data tend to agree better with the empirical results than do the 60° simulated data. Overall, the 60° simulations appear to overestimate the empirical flaw signals in the weld and underestimate them on the far side. The reasons for one angle providing more accurate results than another are not clear; additional studies would be needed to elucidate the issue.
- There are some simulation-to-simulation and flaw-to-flaw variations in the data. At 45°, the EBSD weld shows a signal peak at flaw 3, in front of the weld centerline, and a second peak at flaw 5, after the weld centerline. The random welds, on the other hand, only show a peak at flaws 2 or 3 (Random Weld #2 shows a couple exceptions to this). The second peak is not observed in the empirical results at 45°. However, at 60°, the reverse is seen as a second peak is observed in all the empirical results but not in the simulated results (except with the SNI probe and EBSD weld model).
- The random welds show signal peaks at flaws 2 or 3, consistent with the empirical results. The EBSD weld peaked at flaw 3 for the 45° simulations but at flaw 4 for the 60° simulations. Although a small difference, it differentiates the EBSD weld from the random welds.
- The Voronoi model results are not necessarily outliers, but they are also not an improvement on the EBSD or even the random models. Recall that the Voronoi weld model uses the same Euler angles as the EBSD weld model and has the same average grain size, yet the Voronoi and EBSD results are considerably different. In fact, the Voronoi weld results are worse in several ways. 1) The funnel effect is weaker for the Voronoi model at both 45° and 60°. 2) The shadow effect is stronger for the Voronoi model at 45°. 3) The Voronoi results tend to diverge more overall from the empirical results at 60° than the other models do.
- CIVA tends to overemphasize the funnel effect by making the predicted signal response too high. In general, the figure shows that simulated flaw responses tend to peak more sharply than the empirical responses, exhibiting more dramatic changes in signal intensity as a function of flaw position. This issue is more pronounced at 60° than it is at 45°.
- CIVA tends to overestimate the shadow effect by making the predicted signal response too low. In almost all cases, the empirical signal response from the right-most flaw was relatively higher than the corresponding simulated responses, sometimes by as much as a factor of two.
- More weld models and empirical data are needed to develop reliable statistics on expected variances between models. However, results do suggest that the specific weld geometry (and Euler angle definitions) can have a significant effect on simulation results. It is highly unlikely that there is one “ideal” weld model that can be used for all scenarios. The question is, how many weld models would be needed in order to achieve representative average results?

4.3.4 Comparison of Probes

Every probe should be modeled individually, especially if there are significant differences in aperture size and/or frequency. PNNL has shown that nominally identical probes can perform significantly differently (Dib et al. 2018). Probe parameters, such as peak frequency and bandwidth, should be measured and not assumed from manufacturer specifications. Even so, it is interesting to explore the variation in simulated results between similar probes. Figure 4.17 shows the results from two of the weld models—the EBSD weld (top row) and Random Weld #1 (bottom row)—along with the correlation to the empirical data. Table 4.3 shows the R_z values of only the simulated results to one another. The simulated data from the three probes are strongly

correlated for each scenario, with some flaw-to-flaw variations as high as about 20% or so. Of the three variables investigated here—the weld model, the inspection angle, and the probe—the probe appeared to have the least impact and the inspection angle the most. Note that the probes were nominally similar with comparable aperture, center frequency, and bandwidth.

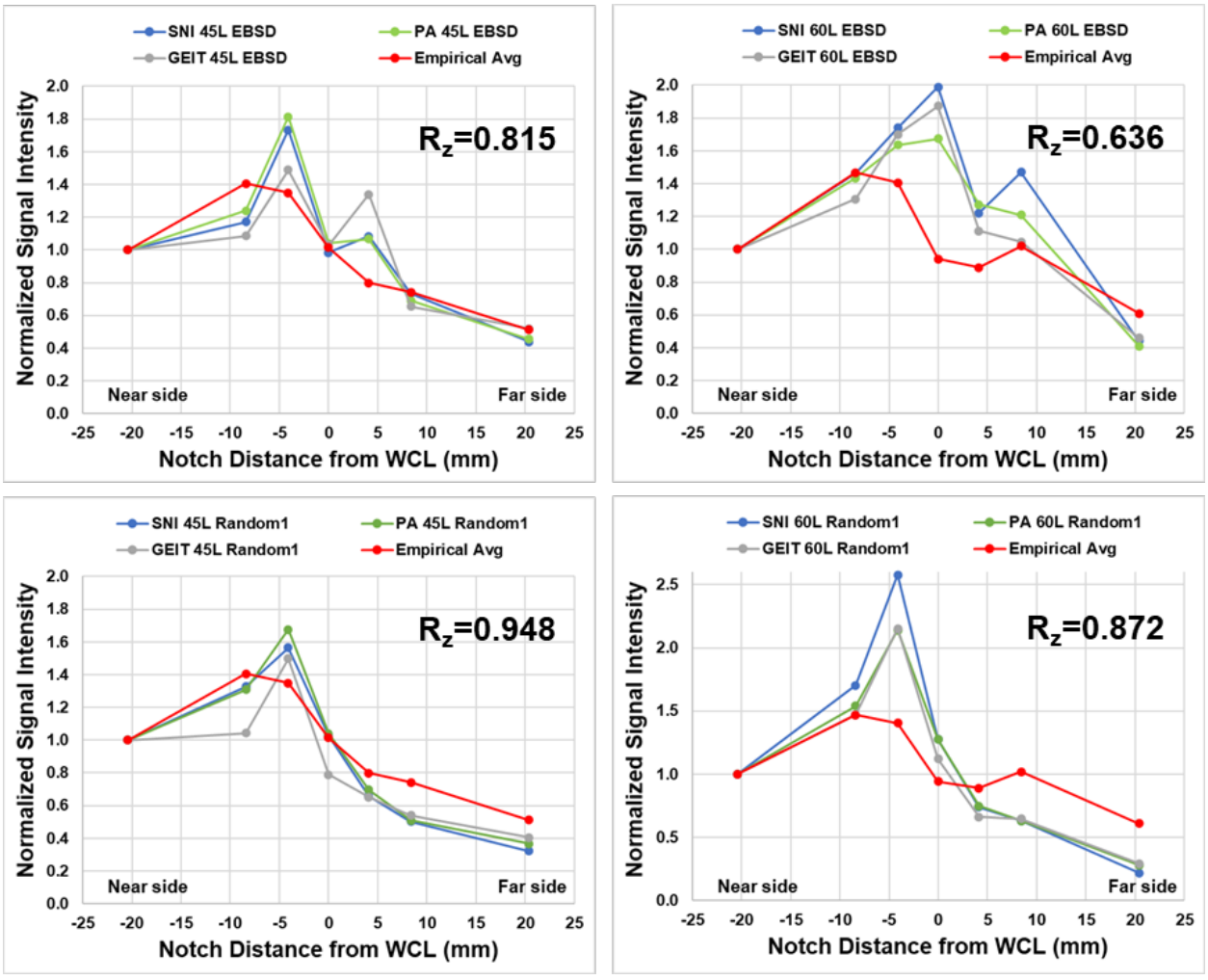

Figure 4.17. Normalized simulated signal response versus flaw position for the EBSD weld model and Random Weld #1 at 45° (left column) and 60° (right column). The average of the empirical measurements is also shown. R_z values show the correlation with the empirical measurements.

Table 4.3. Correlations of the simulated responses to one another.

	45L EBSD	60L EBSD	45L Random1	60L Random1
R_z	0.970	0.961	0.983	0.996

4.4 Summary

PNNL developed seven weld models. The first three were based on EBSD data acquired from an austenitic weld, as described in (Jacob et al. 2020) and shown in Figure 4.1. The other four were variations on the largest-grained EBSD model: three using random Euler angles (Figure 4.5) and one using the EBSD Euler angles but with Voronoi grains (Figure 4.8). Beam simulations were run in CIVA to determine if there were any observable differences between the effects of the weld models on the sound fields. The two smaller-grained EBSD models and the Voronoi model all caused too much scatter and beam attenuation. The other four models were qualitatively similar.

The weld models predicted the funnel effect, in which there is a stronger sound field at the near-side base of the weld. The funnel effect was also observed empirically in beam maps acquired with a laser vibrometer and in flaw responses. Simulated results tend to overemphasize the effect, more so at 60° than at 45°. This implies that near-side flaws may be more difficult to detect than simulations predict.

The weld models also predicted the shadow effect, where beam scatter from the weld resulted in a weaker sound field on the far side of the weld. The shadow effect was also observed empirically in beam maps and flaw responses. Simulated results tend to overemphasize the effect, meaning that far-side flaws may be easier to detect than simulations predict.

Results of the flaw response simulations and empirical flaw responses show that variations in the Euler angles between the weld models affect flaw detectability, although not dramatically in most cases tested. In fact, the results in this section, combined with previous simulations of coarse-grained materials (Jacob et al. 2020), suggest that the number of interfaces and grain sizes have more impact on beam scatter than the specific Euler angle assignments. The ideal minimum grain size is probably on the order of one-half of a wavelength for the CIVA models. Grains that are too large will not give realistic scatter (unless they represent the actual grain structure of the material), while grains that are too small will violate the CIVA high-frequency model assumptions. Note that FEM in other platforms (or CIVA Athena 2D) can be used with arbitrarily small grains, but small grains (especially oddly-shaped grains) will require a finer mesh and much longer simulation times.

In developing a weld model from scratch, it should be noted that quasi-random models—models that incorporate some aspects from available weld data and some random aspects to fill in the data gaps—may be useful. For example, a generic dendritic grain structure common to austenitic welds may be applied to a model geometry and realistic Euler angles assigned to the dendrites. The literature can be used to obtain examples of Euler angles and EBSD-based weld structures (Carpentier et al. 2010; Gardahaut et al. 2014; Nageswaran et al. 2009). There are as many unique weld model possibilities are there are actual welds; the weld models shown in this report should be considered illustrative and not definitive.

5.0 Noise

5.1 Introduction

Noise is a fundamental and inescapable part of UT examinations and has been studied extensively. Although some noise is inherent to the electronic equipment, noise that can substantially affect a UT examination's effectiveness is often a result of structural echoes, such as those from grains, inclusions, or porosities (Jenson et al. 2010). Such noise is referred to by many names including granular noise, structural noise, or backscattered noise. Models of such noise often assume that the noise is the sum of independent scattering centers (Dorval et al. 2013; Margetan et al. 1994). Other noise signals can be a result of surface conditions, such as corrosion or cladding; such surface noise, sometimes referred to as clutter, does not occur throughout the material volume. Surface roughness can mask flaw signals if the coherent scatter signals are high enough, but scrambling or dephasing of the wavefield can also reduce coherent flaw responses (Bilgen et al. 1993; Greenwood 1998).

Flaw response simulations in coarse-grained materials require an understanding of flaw detection in coarse-grained materials in addition to an ability to add realistic granular noise to simulations. PNNL has focused on understanding the effects of grain scatter and noise on flaw detection in CASS materials (Anderson et al. 2007; Crawford et al. 2014; Diaz et al. 2008; Diaz et al. 2009b; Jacob et al. 2019; Ramuhalli et al. 2010). The bulk of this section will concentrate on describing methods for simulating structural noise and noise in coarse-grained models using CIVA.

5.2 Fundamentals of Simulating Noise in CIVA

5.2.1 Structural Noise with Point Reflectors

CIVA can simulate structural or granular noise, which occurs when the sound field encounters a grain boundary and a resulting coherent echo is received by the probe. Every grain boundary is a potential reflector and thus a potential noise source, depending on the geometry and orientation of the boundary facets. To simulate a noise field, one can assume a superposition of echoes from multiple, uniformly-distributed, non-interacting scattering centers (Margetan et al. 1994). CIVA simulates structural noise by randomly adding virtual point-like reflectors to the specimen model and calculating echo responses from those reflectors (the reflectors are not visible in the model). CIVA provides the user two parameters for controlling the noise response (Chatillion et al. 2003; Dorval et al. 2013). The first parameter is the spatial density of the reflectors (ρ), in units of reflectors per cubic millimeter, or reflector/mm^3. The second is the scattering amplitude (*A*), which determines the strength, or reflectivity, of the reflectors; *A* is unitless.[9] It may be more intuitive (although not strictly accurate) to think of noise as being simulated by the addition of pores. ρ controls the spacing, and therefore the total number, of

[9] A couple of technical notes about *A*. *A* is a standard deviation of the reflector amplitude on the basis of a "zero-mean Gaussian distribution" (see the CIVA user's manual). The larger the *A*, the bigger the range of reflectivity values, which allows stronger echoes to be received from some of the reflectors and weaker echoes from other reflectors. Per an email from EXTENDE to PNNL: "Reflectivity may not be the most appropriate word. In CIVA, for structural noise a diffraction coefficient D is assigned to each point. Its diffracted wave is proportional to D." Either way, the parameter *A* controls the overall signal strength from the noise points.

pores, and *A* controls the size or reflectivity of the pores. The CIVA default for each parameter is 1.

Simulations were conducted in the flaw response module to determine the effects of ρ and *A* on the overall noise amplitude and appearance (it should be noted that CIVA 2020 cannot simulate noise in beam simulations, although the noise options are available in the beam simulation module). One immediate finding was that adding noise to a simulation can increase simulation time dramatically. Therefore, to minimize iterations, it is important to understand how the noise parameters interact with one another to give desired results. PNNL's initial tests used a relatively small specimen model (to minimize the absolute number of noise points and therefore minimize simulation time) and simple inspection parameters.

The relationship between ρ and *A* was iteratively examined. ρ was varied over several orders of magnitude, and *A* was adjusted to maintain an approximately constant signal-to-noise ratio (SNR). This was done by simulating a B-scan with a simple 13 mm (0.5 in.) diameter probe at 1 MHz, 2 MHz, and 5 MHz. A 1 mm (0.04 in.) side-drilled hole (SDH) was added to the specimen model. ρ was increased by factors of 10 from 0.01 reflector/mm^3 to 100 reflector/mm^3. It was empirically determined that decreasing *A* by a factor of approximately 3 while simultaneously increasing ρ by a factor of 10 maintained a nearly steady SNR for values of ρ ≥ 0.1 reflector/mm^3. As shown in panel A of Figure 5.1, the peak signal intensity of the side-drilled-hole echo remained constant for all sets of noise parameters (the x-axis shows ρ/*A* combinations used for each simulation). Panel B shows that the average noise level was fairly steady but slowly increased for values of ρ ≥ 0.1 reflector/mm^3. The resulting SNRs are shown in panel C. For ρ < 0.1 reflector/mm^3, the noise level decreased rapidly, resulting in a sharp increase in SNR and indicating that the 10 times increase to 3 times decrease relationship of ρ to *A* fails for low values of ρ.

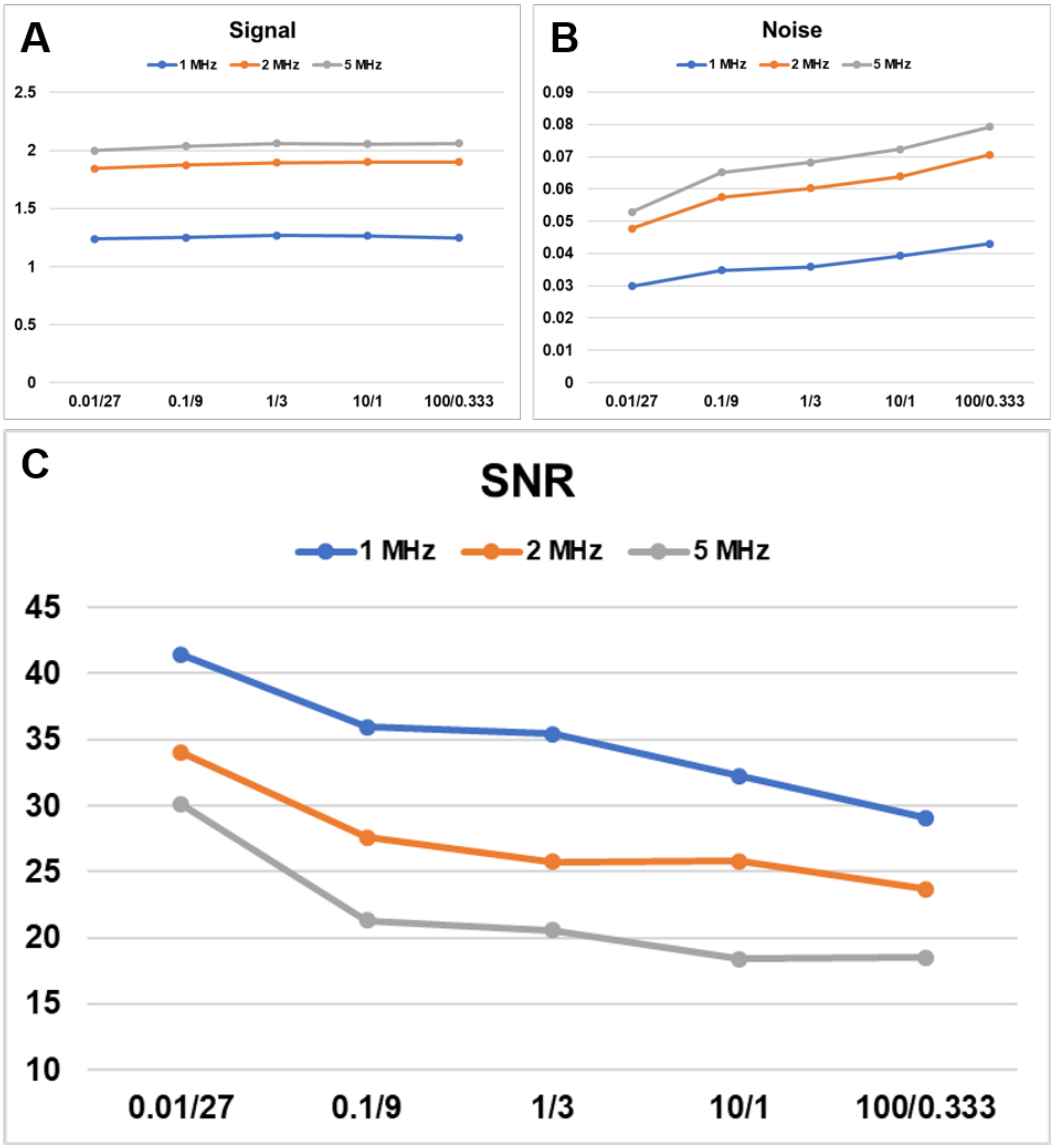

Figure 5.1. Plots showing how the SNR trend is affected by ρ and *A*. The x-axis labels are ρ/*A* for each simulation run. A: Peak signal intensity of the SDH echo (arbitrary units). B: Average noise level (arbitrary units). C: SNR, or the data in A divided by the data in B.

There are three important observations. 1) In an empirical scan (especially in coarse-grained materials), some sound energy is scattered or received as noise, thereby diminishing the total amount of sound energy available to reflect from the flaw. However, as shown in Figure 5.1, the flaw response in CIVA is constant and independent of the noise response. The noise and the flaw response echoes are calculated independently. 2) A desired SNR can be obtained through trial-and-error by different combinations of ρ and *A*. 3) Different noise densities can be achieved with an approximately constant SNR, because the increased noise from a factor of 10 increase in ρ can be approximately compensated for by a factor of three decrease in *A*.

It should be noted that the noise scattering centers are automatically randomized throughout the material for every simulation. Thus, duplicating a noise field in CIVA from simulation to

simulation is not possible. Also, PNNL found that, depending on their location, noise reflectors may produce an echo that appears unusually strong, likely the result of a reflector directly in the simulation plane. As a result, multiple iterations may be needed to produce the desired noise field. Furthermore, a strong noise echo can be coincident with a flaw response, making the flaw response appear stronger than it actually is.

5.2.2 Noise Using Polycrystalline Specimen Models

CIVA 2020 has a “polycrystalline” option for defining a specimen microstructure. As described in Section 4.2.4, this option is designed to model specimens composed of multiple crystals or grains. The user can define an average grain size (but not a range of grain sizes), an elongation factor for non-equiaxed grains, and an orientation. PNNL ran several investigatory simulations that will be discussed in the context of noise.

First, separate simulations were run with different polycrystalline grain sizes, ranging from 10 to 10,000 µm (0.0004 to 0.4 in.). Each simulation also included a 1 mm (0.04 in.) diameter SDH. A single-element, 2 MHz probe was used in CIVA’s Direct Echo mode; that is, only echoes directly from the flaw were calculated, and specimen bounces were ignored. Simulation results are shown in Figure 5.2. This figure illustrates the following: 1) There is no visible noise at the 10 µm (0.0004 in.) grain size. 2) The noise intensity peaks at 1000 µm (0.04 in.) then diminishes for the largest grain size. At the largest grain size, the reduced noise probably results from violating the Born approximation; CIVA uses this approximation to calculate scatter in sound fields due to inclusions (Mahaut et al. 2010). It assumes weak scattering and calculates the interaction of the sound field with each scattering center (or grain) as if no other grains were present then adds all the contributions of each grain. Some important information was provided by EXTENDE: “Because this calculation is based on the Born approximation, it becomes less accurate when grains become larger or more anisotropic. In the diffusion regime, the grains are too large for the Born approximation to be accurate and it fails to yield the correct frequency dependency.” Thus, the results with the 10,000 µm (0.4 in.) grains are not expected to be accurate. (Again, this illustrates the importance of being familiar with the model approximations that CIVA uses.) 3) As with the structural noise, the signal from the SDH remains constant regardless of grain size or noise level.

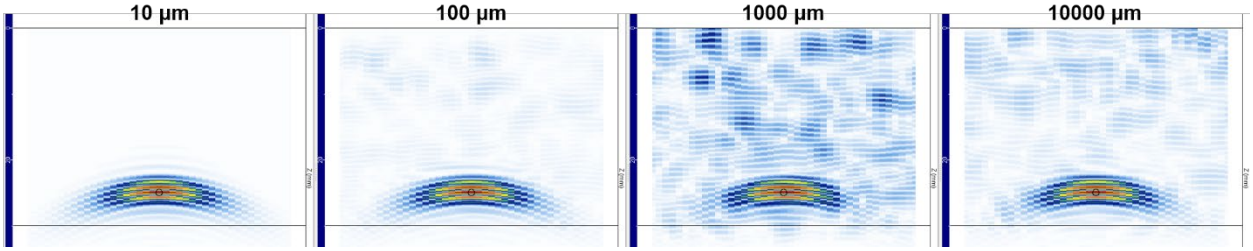

Figure 5.2. Simulations of an SDH echo response using the CIVA polycrystalline model.

A second simulation was run to test how CIVA handles multiple polycrystalline regions in one specimen. Such a setup might be useful for creating a simplified austenitic weld model, as discussed in Section 4.2.4. In such a model, there may be macroscopic regions of preferential grain size and/or orientation that could be conveniently grouped using the polycrystalline option. The result would be a simple model with a few regions, each with different polycrystalline definitions. In the simplified model tested, a single specimen was defined with multiple regions, and each region was assigned a different polycrystalline grain size. An SDH was placed in each region, and the probe was moved across the specimen for a B-scan simulation. Figure 5.3

shows the results of the simulation. In the top row, the Half Skip option was selected, which calculates both direct and incidental flaw echoes. Though echoes from the specimen backwall (BW) were not computed, there was a series of echoes that appeared to be backwall echoes. However, EXTENDE explained to PNNL that “the additional echoes you see at a higher time of flight are not the backwall echoes but a second echo of the SDH after a skip on the backwall.” In other words, these echoes are from the SDH→BW (or BW→SDH) interaction (these interactions are illustrated at the top of the figure). The fact that this echo was stronger than the direct echo from the SDH seemed odd but was not further investigated. There is also a series of weaker half-skip echoes, in which the sound first reflected from the backwall, then interacted with the SDH, and finally reflected from the backwall again back to the probe (i.e., BW→SDH→BW).

In the second row of Figure 5.3, the polycrystalline “material noise” option was selected. It is unclear from the CIVA user’s manual what material noise is, but it did not appear to increase the simulated noise substantially, if at all. In the third row, the Direct option was selected, which calculates flaw echoes from sound directly incident on the SDH and ignores all specimen bounces (i.e., the SDH→BW and BW→SDH→BW echoes were not calculated). In the bottom row, the Half Skip option was again selected, but the flaws were deactivated so that they would be ignored in the simulation. This scenario should result in no flaw or specimen echoes at all—only noise from the polycrystalline structure. However, there persisted a mystery echo apparently from the backwall, but only in the third and fourth regions. The source of the mystery echo was not further investigated.

The following summarizes what PNNL has learned about CIVA’s polycrystalline model option:

- The polycrystalline option can be used to generate simulated noise. However, based on the limited work that PNNL did with this option, it is unclear if there are any advantages to this approach over the structural noise approach.
- There is a frequency-dependent effect with the simulated noise and scatter, which is expected because such effects are observed empirically. Frequency dependence was verified in CIVA for a wide range of frequencies and confirmed by EXTENDE. CIVA uses a modified Born approximation for simulating echoes due to scattering from inclusions. This is a low frequency approximation. That is, the wavelength of the sound should be larger than the dimensions of the grain. For additional information about polycrystalline models and scatter in the different regimes, see Ganjehi et al. (2012). (Note that the diffusion regime in that paper [Section 3.2] is incorrect [i.e., for $\lambda << D$, $\alpha \propto 1/D$]).
- It is unclear how the distribution of the size and orientation of the grains is controlled. The user can select the grain size and orientation, but not a range. In a communication to PNNL, EXTENDE noted that “A homogenization process calculates the parameters of the equivalent medium: the speed of sound and the attenuation coefficient.” Each region is considered homogeneous with respect to grain size, grain orientation, wave velocities, and attenuation (Ribay et al. 2012).

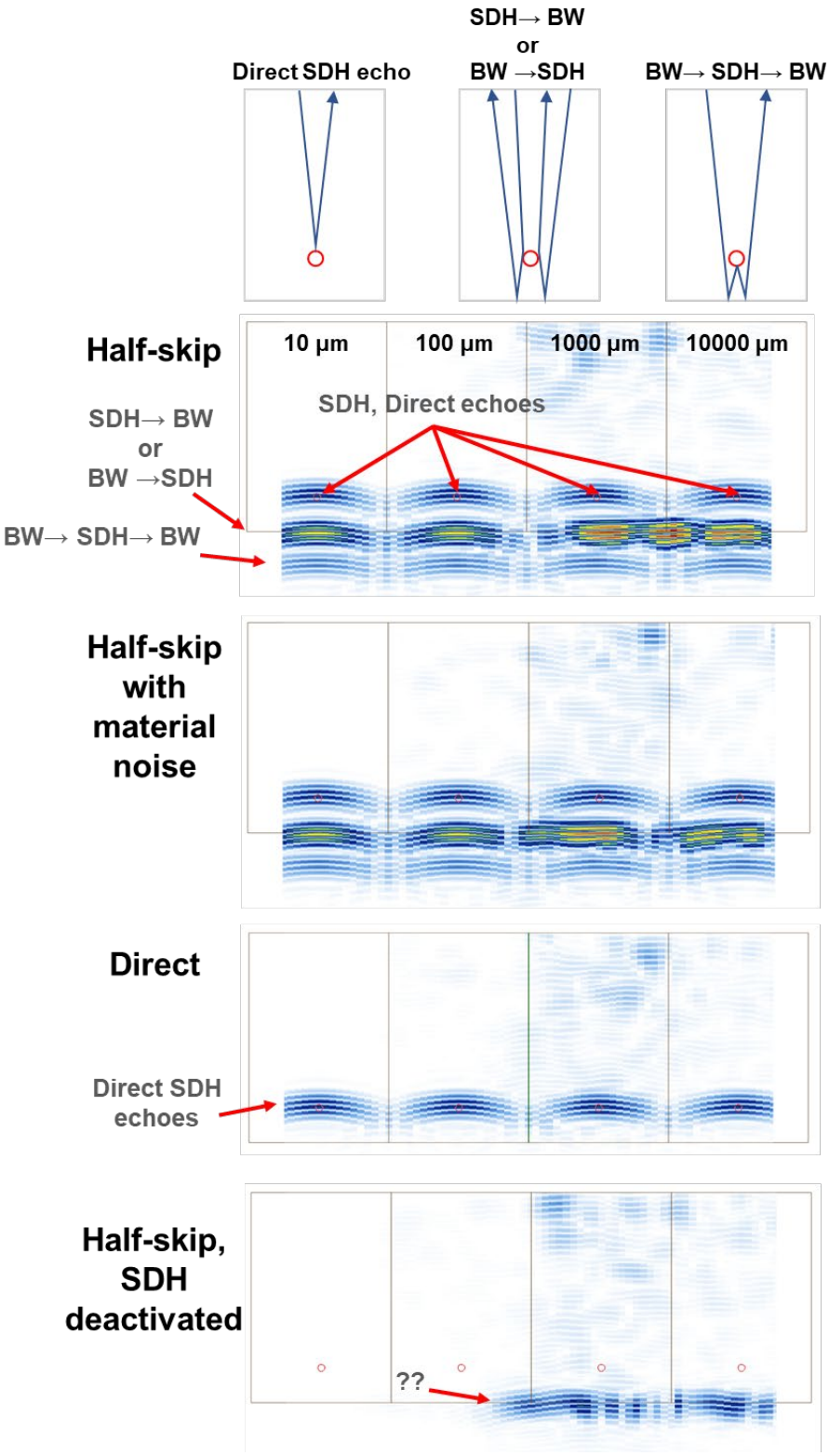

Figure 5.3. Multi-region polycrystalline simulation results.

5.2.3 Surface Noise

In certain instances, noise or “clutter” originating from the ID surface of a specimen can interfere with flaw detection. This is illustrated in Figure 5.4, a laboratory scan of a DMW pressurizer surge nozzle specimen that contains an implanted flaw; this specimen is described in detail (as 8C-901) in Diaz et al. (2009a). The flaw was a hot isostatically pressed EDM notch that was placed at (or near) the weld fusion line on the CS side of the weld. The TW depth is about 18% (7 mm [0.28 in.]). The CS side, including the weld region, is tapered outward at about 7°. According to specimen diagrams, the flaw was implanted at a 0° angle with respect to the horizontal, so the flaw is effectively tilted at 7° with respect to a UT search unit on the CS side. The flaw should have been readily detected in the empirical scan, but it was not due to excessive ID surface noise. The source of the noise is unclear, but it may be from surface roughness or the SS cladding layer. Modeling was undertaken in an attempt to duplicate these results.

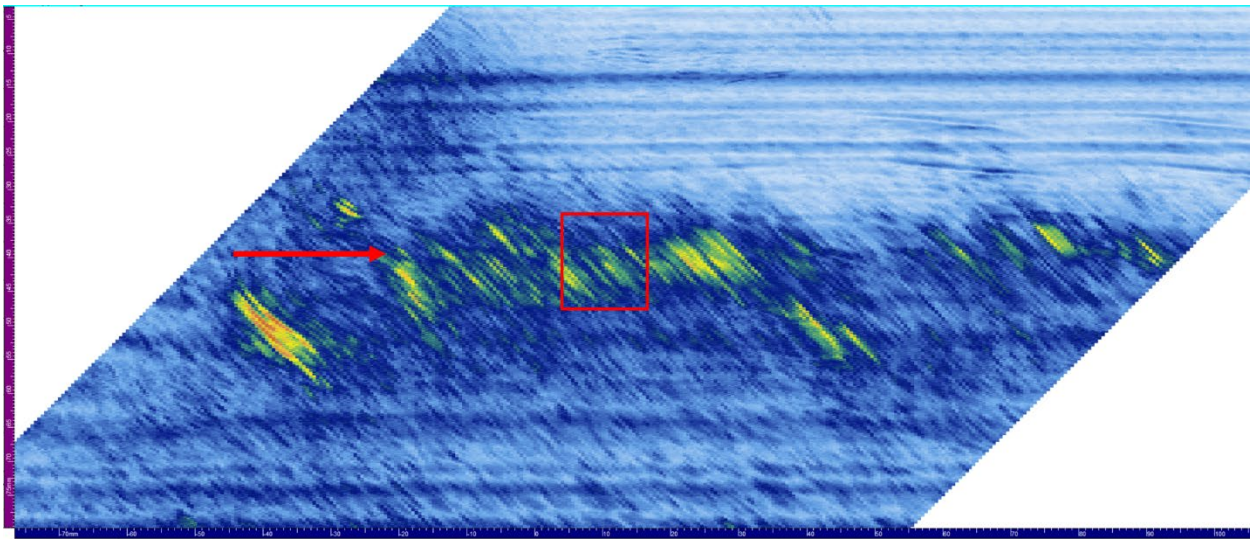

Figure 5.4. Empirical B-scan with surface clutter. The flaw was expected to be found within the red box. The arrow indicates the ID surface.

For simulating the mockup, we used the EBSD weld model and added a 7° tilt to the weld end, as shown in Figure 5.5. Tilting the model altered the vertical alignment of regions and affected CIVA’s weld region sequencing and subsequent Euler angle assignments. As a result, a few of the regions were assigned incorrect Euler angles due to the tilt; this was ignored. Initial simulations were performed with the 2 MHz 45° TRL SNI probe. A reference flaw at 0° with respect to the specimen backwall and 5 mm (0.2 in.) deep was added to the CIVA model.

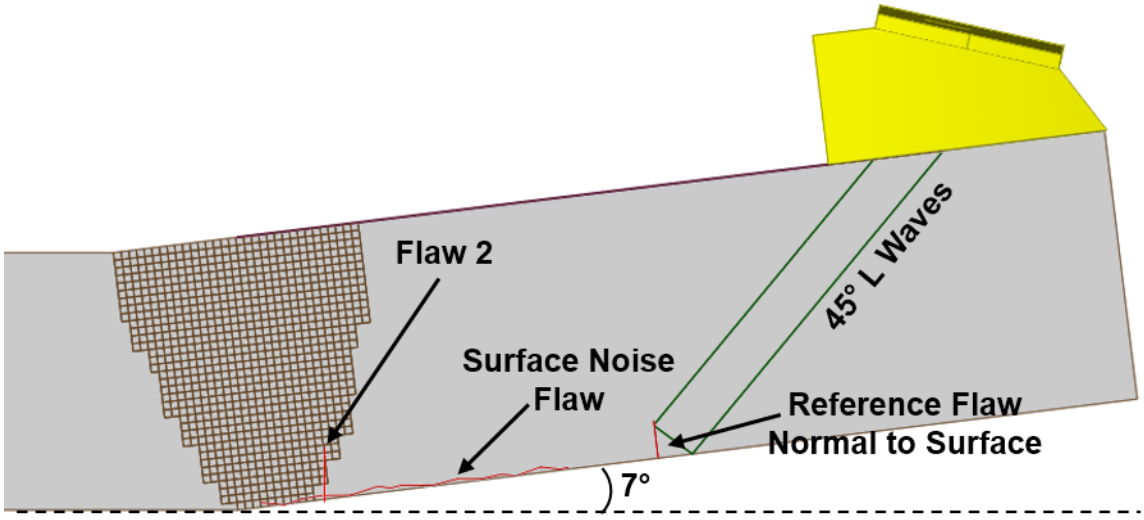

Figure 5.5. The 8C-091 specimen model used in CIVA.

To simulate the noise, PNNL first attempted to use structural noise in a narrow, rectangular simulation zone near the ID surface. However, CIVA 2020 would not produce any noise with this setup. The issue was sent to EXTENDE, and they confirmed that there was a bug (PNNL has not verified if this bug was fixed in subsequent releases). PNNL then devised a workaround comprising a multifaceted flaw that was drawn in CIVA and positioned on the ID surface extending through the weld region. This flaw is labeled “Surface Noise Flaw” in Figure 5.5. Simulation results are shown in Figure 5.6. In the left panel, the surface-noise flaw was omitted, and the echo from flaw 2 was readily visible near the weld. In the right panel, the surface-noise flaw resulted in diminished signal from flaw 2. Additional echoes from the surface-noise flaw facets were indistinguishable from the intended flaw. Simulated results agreed well with the empirical results, illustrating that simulated surface noise can cause an otherwise detectable flaw to go undetected. The workaround was likely better than trying to use the noise simulation because the source of the noise signal originated from the specimen surface—just as in the empirical scan. Also, simulation time was probably less than with a noise zone, depending on the noise parameters.

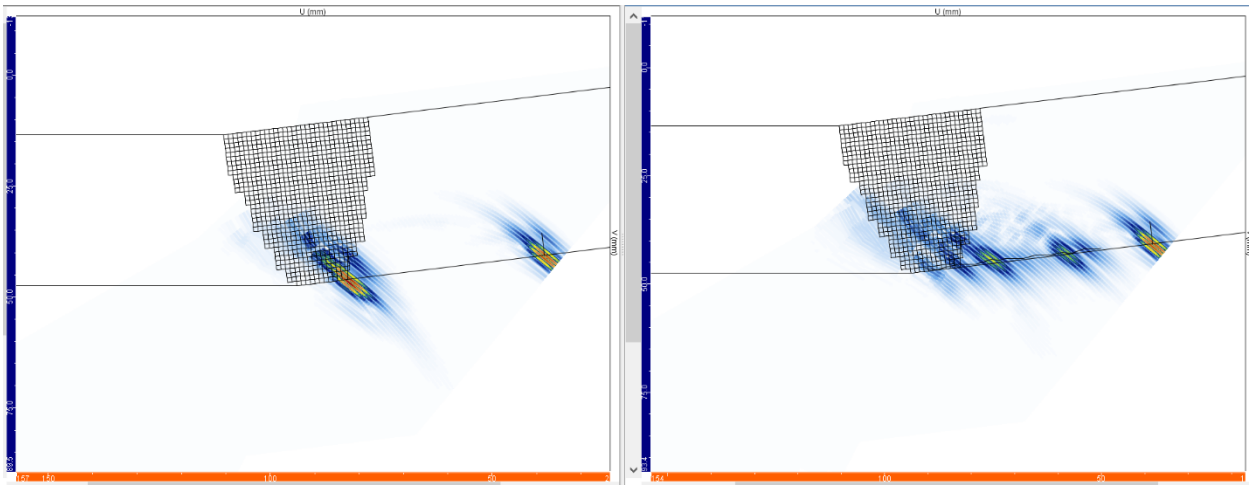

Figure 5.6. Simulation results illustrating an approach to surface noise. Left: the flaw is visible in the absence of surface noise. Right: with surface noise added, the flaw is no longer detected.

This work illustrates that surface noise can be simulated by a multifaceted flaw on the surface of the specimen model. The noise can be toggled on or off by activating or deactivating the flaw in the Simulation Settings tab. The multifaceted flaw's geometry may have to be iterated until it produces desired results, but flaws can be easily edited, resized, repositioned, and exchanged in the model. A couple items to note are: 1) the facets should not be so steeply angled that they resemble pitting or corrosion (unless those are the desired targets of the simulation), and 2) the facets should not be so small that they violate CIVA's high-frequency approximations (i.e., the reflector size should be on the order of, or larger than, the wavelength).

It should be noted that CIVA has a "surface roughness" option in the Geometry tab of the Specimen tab. This option only affects the wedge/specimen interface and is modeled as a modification of the transmission coefficient, which is a function of the probe frequency. This option will not generate surface noise but will act as a source of probe attenuation. Surface roughness is explored in Section 6.3.

5.3 Noise Simulations in Coarse-grained Material

5.3.1 CIVA Structural Noise

PNNL investigated methods to effectively add noise to simulations of coarse-grained materials, such as CASS. B-scans were simulated for values of ρ ranging from 0.001 reflector/mm^3 to 100 reflector/mm^3 and are shown in Figure 5.7. Each value of ρ was approximately matched to a laboratory specimen as a real-world example of a granular structure with a comparable scatterer, or grain, density. Photographs of the polished and etched specimens are also shown in the figure. Four points gained from this figure will be discussed.

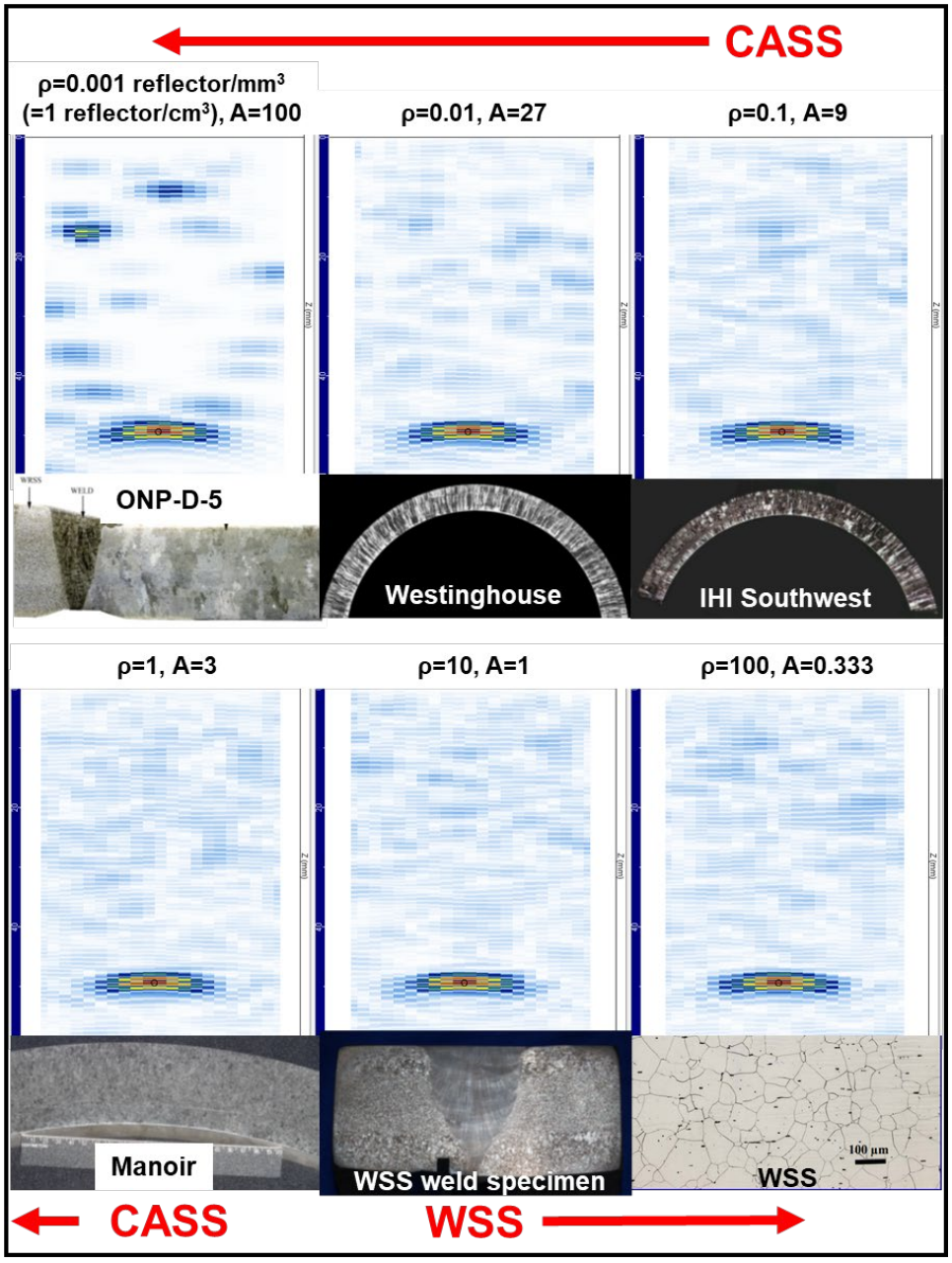

Figure 5.7. Flaw response simulations with different noise parameters. A and ρ pairs were selected such that approximately the same SNR was maintained for all simulations (except for ρ = 0.001). Examples are shown of stainless steel materials that exhibit a grain density approximately comparable to ρ.

First, the pattern of noise appears relatively consistent for values of ρ ≥ 1 reflector/mm^3 (bottom row of the figure), i.e., high scatterer densities had very little effect on the appearance of the noise. On the other hand, decreasing ρ below 1 reflector/mm^3 eventually leads to discrete noise centers since the noise reflectors are far enough apart that their echoes do not substantially overlap. For example, see the top left panel of the figure where ρ=0.001 reflector/mm^3. The level of ρ where discrete noise echoes are obtained is a function of probe frequency because lower frequencies produce larger noise echoes.

Second, for comparisons with CASS materials, these dimensions can be roughly translated to a linear grain size d. Imagine a cubic grain with the dimension d of each edge and a volume of d^3. For example, one noise reflector per cm^3 can be thought of as corresponding to hypothetical grains that are 1 cm × 1 cm × 1 cm, or 1 cm^3 (0.4 in. × 0.4 in. × 0.4 in., or 0.06 $in.^3$). Ten noise reflectors per cm^3 would correspond to grains that are 0.46 cm × 0.46 cm × 0.46 cm, or 0.1 cm^3 (0.18 in. × 0.18 in. × 0.18 in., or 0.006 $in.^3$).[10] Note that ρ=0.001 reflector/mm^3 corresponds to 1 reflector/cm^3 (16 reflectors per $in.^3$), ρ=0.01 reflectors/mm^3 corresponds to 10 reflectors/cm^3 (160 reflectors per $in.^3$), and so forth. Table 5.1 summarizes the noise reflector density and amplitude parameters used, the corresponding grain dimensions, and the simulation time for the B-scans shown in Figure 5.7.

Table 5.1. Noise reflector density and simulation time.

Noise Reflector Density (per mm^3)	Noise Reflector Density (per cm^3)	Noise Reflector Density (per $in.^3$)	Grain Dimension d (mm)	Noise Reflector Amplitude A	Approximate Simulation Time
0.001	1	16	10	100	4 seconds
0.01	10	164	4.6	27	5 seconds
0.1	100	1,640	2.2	9	12 seconds
1	1,000	16,400	1	3	75 seconds
10	10,000	164,000	0.46	1	12 minutes
100	100,000	1,640,000	0.22	0.333	15 hours

The third point regarding Figure 5.7 is that empirical scans from specimens with known grain structures are important for benchmarking and assessing simulation results. The figure shows examples of SS material that approximately represent the grain sizes for each ρ. The CASS materials were laboratory specimens described in (Jacob et al. 2019) for end-of-block signal dropout studies. The figure illustrates how the range of CASS and WSS grains in different laboratory specimens can be modeled with different values of ρ. Empirical B-scans from four of the laboratory specimens are shown in Figure 5.8. These scans were acquired with a 1 MHz probe but are illustrative in this context. Qualitative comparisons can be made of the appearance of the noise in this figure to the simulations of the corresponding specimens in Figure 5.7. Overall, the empirical scans show discrete noise echoes from individual scattering centers (presumably grains), similar to the patterns that were simulated with the smallest values of ρ. However, noise in the empirical scans is also characterized by a more uniform and lower intensity background, similar to that obtained using higher values of ρ. Even coarse-grained CASS has a large range of grain sizes, so the distribution, size, and orientation of “noise reflectors” is not uniform. Smaller grains provide a weaker noise echo that, in ensemble, give a uniform-looking noise background, and larger grains may provide stronger, more discrete, and more coherent echo responses that are visible above the noise background.

[10] The dimensions of the hypothetical grain are found by taking the inverse cube root of the number of reflectors per cm^3; i.e.; $d = \rho^{(-1/3)}$, or $0.46 = 10^{(-1/3)}$.

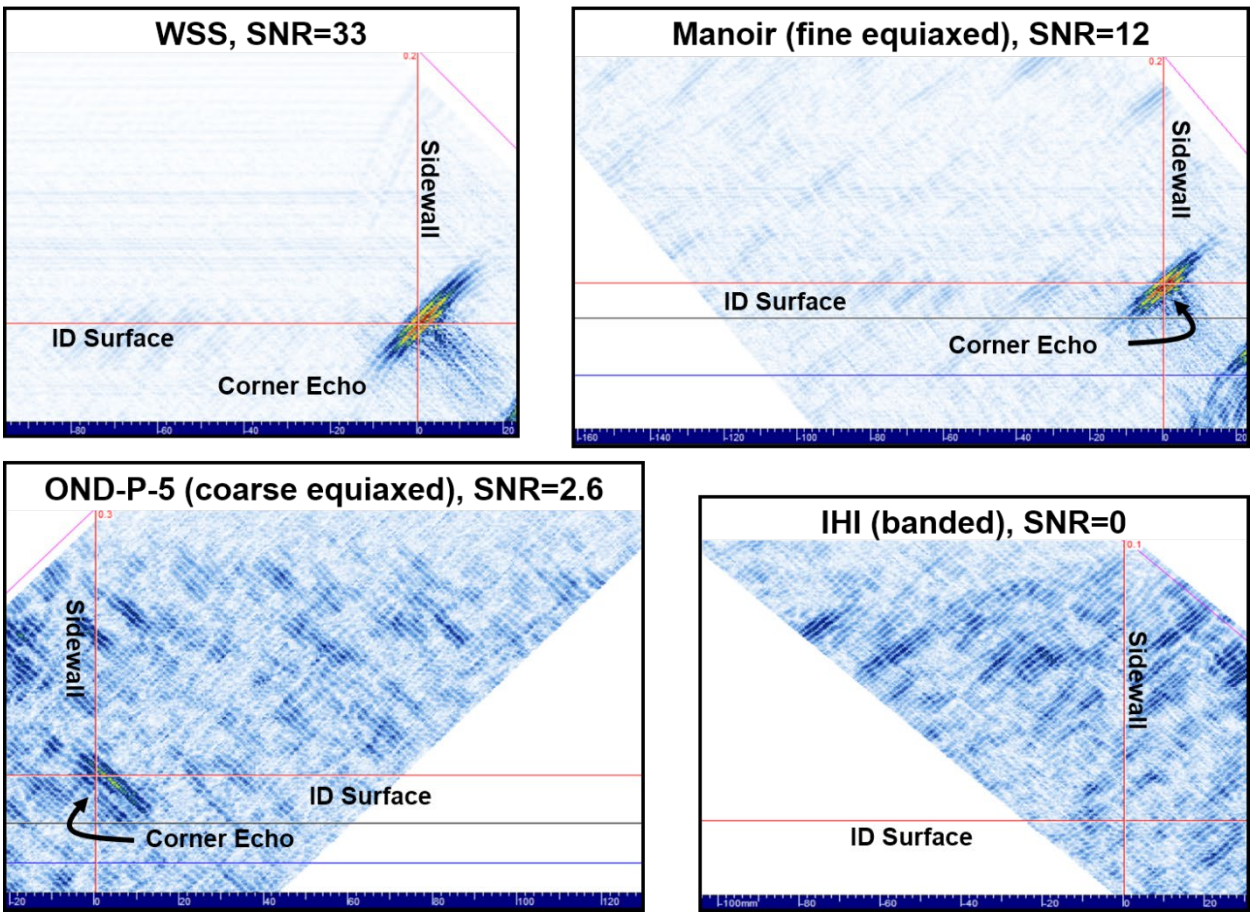

Figure 5.8. Empirical scans through various CASS materials illustrating noise patterns and corner echo intensities.

Fourth, Table 5.1 shows that simulation time is a strong function of reflector density. Figure 5.9 illustrates the relationship between simulation time and ρ on a log-log scale. A straight line on a log-log plot indicates an exponential relationship, with the slope of the line defining the exponent. However, in this case, the increasing slope of the line indicates that simulation time is a very strong function of ρ, stronger than an exponential. Using higher values of ρ would result in impractically long simulation times. Indeed, extrapolating the data in Figure 5.9 suggests that a simulation with ρ = 1000 reflector/mm^3 would take approximately one year (≈3×10^7 sec) to complete. In PNNL's testing, values of ρ ≥ 1 reflector/mm^3 resulted in only small differences in the appearance of the noise, so using high values of ρ is both impractical and unnecessary.

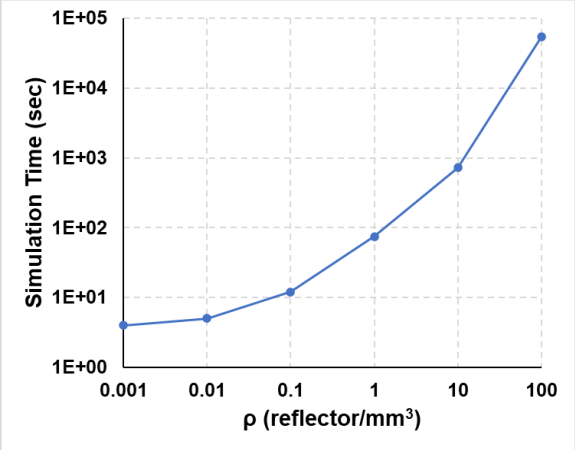

Figure 5.9. Simulation time as a function of noise reflector density. Note that the plot is on a log-log scale.

The information learned about noise simulations in CIVA was applied to an example of a planar flaw response. The flaw response was simulated with three noise scenarios: no noise, a low noise amplitude (relative to the flaw response), and a strong noise amplitude. The specimen was 85 mm (3.3 in.) thick with a 10 mm (0.4 in.) rectangular flaw (12% TW). The probe was 1 MHz TRL with a focal depth of 79 mm (3.1 in.) at a 45° refraction angle. ρ was set to 0.001 reflector/mm^3 to imitate a coarse-grained material, and *A* was varied. Figure 5.10 shows the results; *A*=0 is in the top panel, *A*=100 is in the middle panel, and *A*=1,000 is in the bottom panel. In all three situations, the flaw response is prominent. In the top panel where there is no noise, the tip response is clearly visible. In the middle panel, the tip is still visible, and some background noise is apparent. The bottom panel shows prominent noise, and the tip echo is no longer clearly distinguishable. The red circles indicate regions where the noise signal is stronger than the tip signal. Note that in the bottom panel, a noise signal is incidentally near the location of the tip, but it is not a tip signal; the flaw response amplitudes are not affected by the simulated noise, so the true tip signal is weaker, as seen in the top and middle panels. The SNR of the tip signal in the bottom panel is < 1. These results demonstrate how noise simulations can be used to reduce SNR and make it more difficult to characterize the flaw depth. Higher noise amplitudes could be used to obscure corner echoes; however, the noise amplitude would likely have to be unrealistically high. A better method of simulating signal loss in coarse-grained materials is to control the sound field scatter (Section 5.3.2) and/or attenuation (Section 6.0).

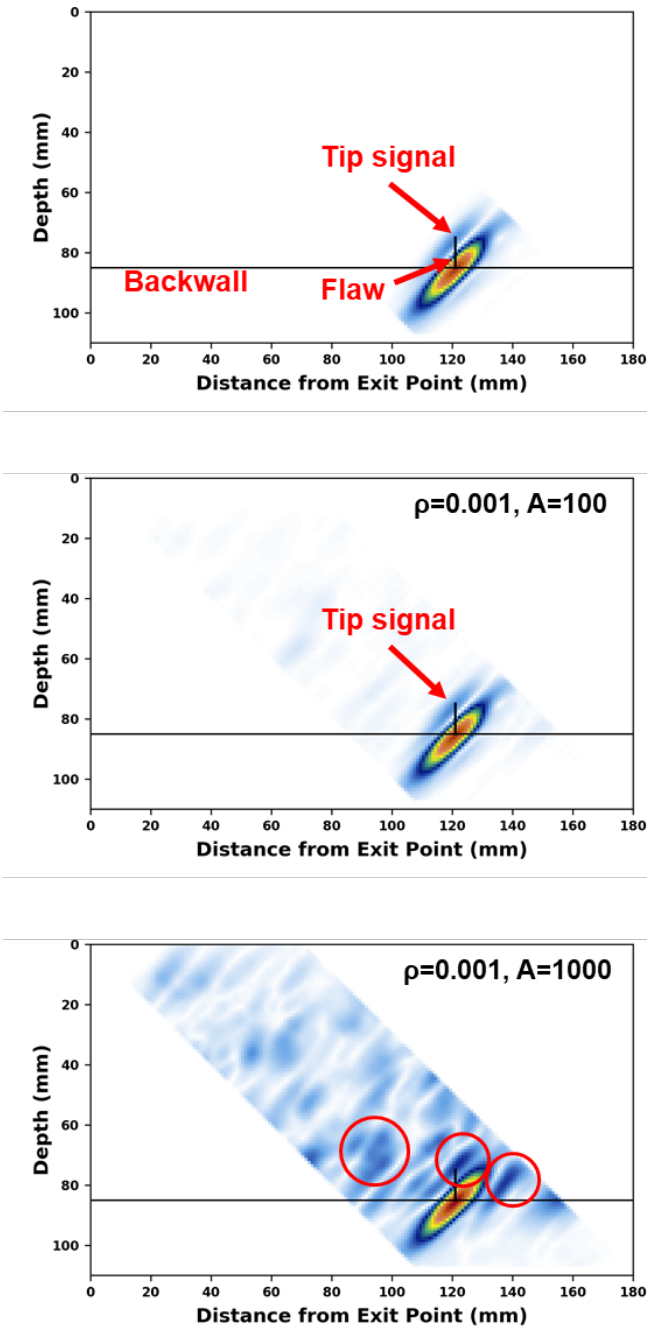

Figure 5.10. Flaw response simulations with noise. Top: Flaw response simulations with no noise (top), low noise (middle), and high noise (bottom). There are several regions in the high-noise figure where the noise signal is greater than the tip signal (red circles).

5.3.2 Simulating Noise with a CIVA Voronoi Model

Note in Figure 5.10 that the flaw response is unaffected by the presence of noise. To generate a more realistic response, a noise model and a coarse-grained model must be combined. There are two options in CIVA 2020 for simulating a flaw response in a coarse-grained specimen: create a coarse-grained model (such as PNNL has done previously with polished and etched

CASS specimens) or use CIVA's Voronoi feature. PNNL previously showed that the Voronoi grains resulted in beam simulations that were similar to empirical beam maps and were much easier to create and faster to run (Jacob et al. 2020); also see Section 4.0. The Voronoi approach is the obvious choice. However, there are two important limitations of using Voronoi regions. First, the Voronoi option is not available with CAD-generated specimen models or weld models. Any custom geometry, such as a dissimilar metal weld or a geometry based on a real specimen, would not support Voronoi grains in CIVA 2020. Thus, simulating a dissimilar metal weld with CASS as one of the base materials would require a manually-created specimen. Second, noise cannot be simulated with Voronoi regions in CIVA 2020. When the "Coarse Grained" material type is selected (which activates the Voronoi region option), the "Attenuation/Structural Noise" option disappears. Voronoi regions cause scatter but do not by themselves result in noisy simulations; the region boundaries are apparently treated differently from other interfaces, lacking backscatter-associated noise. However, PNNL devised a workaround that may be suitable for simulating a flaw response with noise in coarse-grained Voronoi materials.

The workaround includes performing a first simulation with the Voronoi regions activated, then performing a second simulation with the same model parameters but through an isotropic material with noise activated. The two results can then be combined outside of CIVA. An example of this approach is shown in Figure 5.11. The top two figures show the flaw response and the noise response. In the noise simulation, the flaw must be deactivated so that no flaw response is calculated—only noise is simulated in that case. The bottom figure shows the sum of the two simulations. The simulations were reconstructed outside of CIVA using Python code that assumes straight-line beam paths (see Appendix A.1). The image summation was also performed using Python. More sophisticated approaches might be devised to combine the images (for example, to normalize signal intensities or to blend noise and flaw response signals), but simply adding them together was adequate for this illustration. This method of simulating a flaw response with noise is relatively fast, as long as the individual simulations are not too complicated. Using an excessive number of Voronoi regions and/or a high ρ can make this approach impractical due to long simulation times. There are a couple of drawbacks to this workaround. One is that off-line processing is needed to correct the ray paths and to combine the images. The second is that considerable trial-and-error may be necessary to find the desired noise level and appearance. These shortcomings can be overcome with a parametric study to change noise properties and an automated post-processing approach.

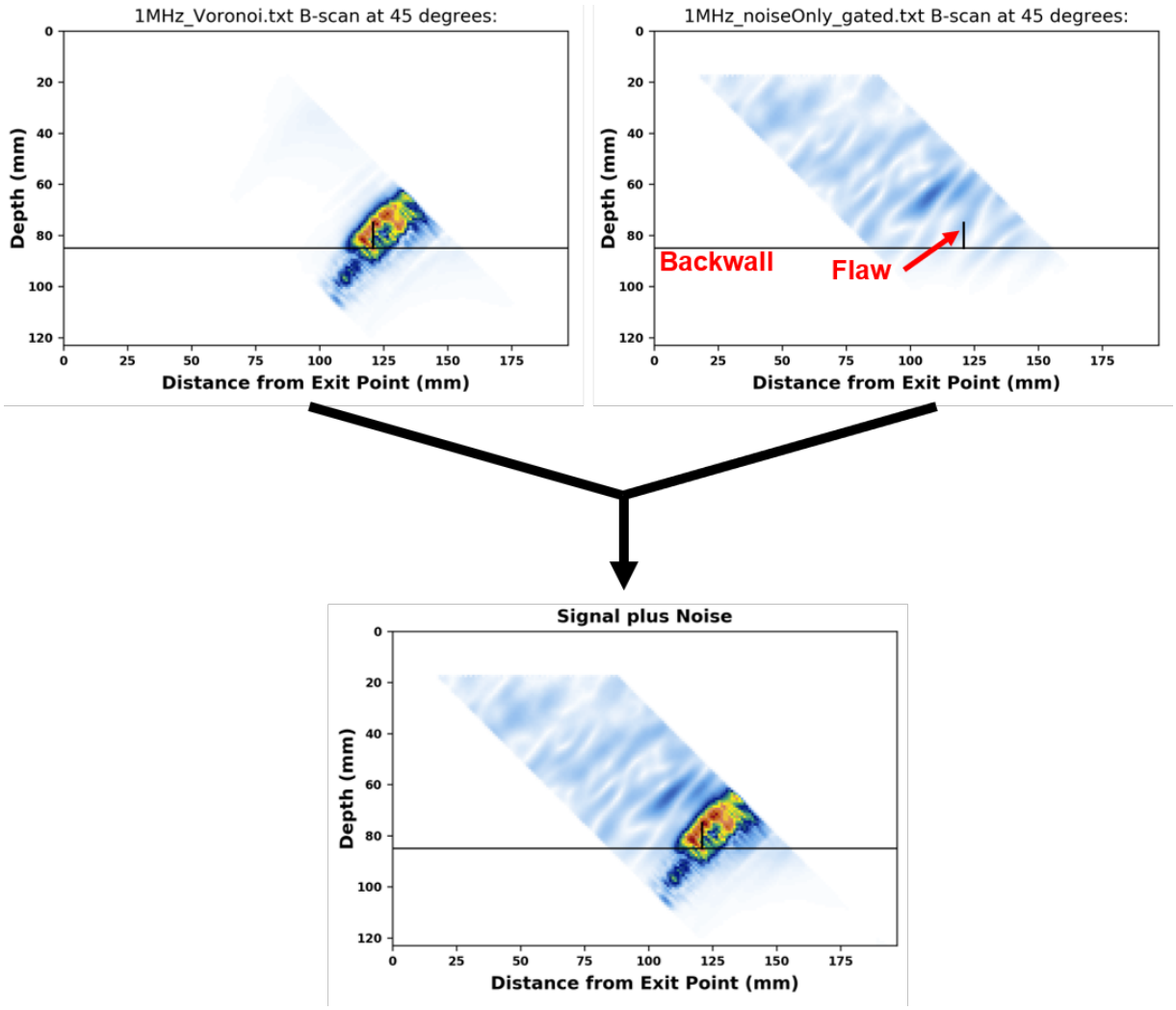

Figure 5.11. Simulation results with Voronoi regions (top left), noise only (top right), and their sum.

Next, an example is provided to illustrate the effect of combining noise in an isotropic specimen with a flaw response in a Voronoi specimen over a range of Voronoi parameters. A simple (non-weld) specimen model was used with 10,000 Voronoi regions to imitate coarse-grained CASS. The specimen was 160 mm × 80 mm × 50 mm (6.3 in. × 3.1 in. × 2.0 in.), resulting in a typical grain volume of 64 mm^3 (0.0039 $in.^3$), equating to an average grain size *d* of 4 mm (0.16 in.). From Table 5.1, this grain size can be represented by a ρ of about 0.01 reflector/mm^3. A 1 MHz TRL probe was modeled with a focal depth of 45 mm (1.8 in.) at 45°. The speed of sound range ΔV of the Voronoi regions was increased from 0% to 6% in 1% intervals. A 100% TW flaw was added to imitate an end-of-block so that comparisons could be made to the empirical end-of-block scans in Jacob et al. (2019).

Figure 5.12 shows the simulation results (results were postprocessed in Python). All panels were normalized to the isotropic case (upper left). The simulations show a steady decrease in echo signal intensity as ΔV is increased until the echo signal is essentially gone at ΔV=5%; this relationship is expected, since grain scatter causes attenuation (this will be explored further in Section 6.5). A separate noise-only simulation was performed and then added to the flaw response simulations. Figure 5.13 shows the results with ρ = 0.01 reflector/mm^3 and *A* = 400.

The lower right panel is the noise without any flaw response simulation. The echo signal is effectively overwhelmed by the noise at the ΔV = 3–4% range. Note in Figure 5.12 that spurious signals exist in the region above the echo response, especially for ΔV=1% and greater. The origin of the artifact is not currently known. It appears to be present at very low levels in the isotropic case, so the Voronoi regions are not causing it, but they are somehow intensifying it.

Accurate flaw response modeling in CASS models will require parametric studies to determine ideal values of *d*, ΔV, ρ, and *A* for a given specimen, particularly if a polished and etched face of the specimen is unavailable.

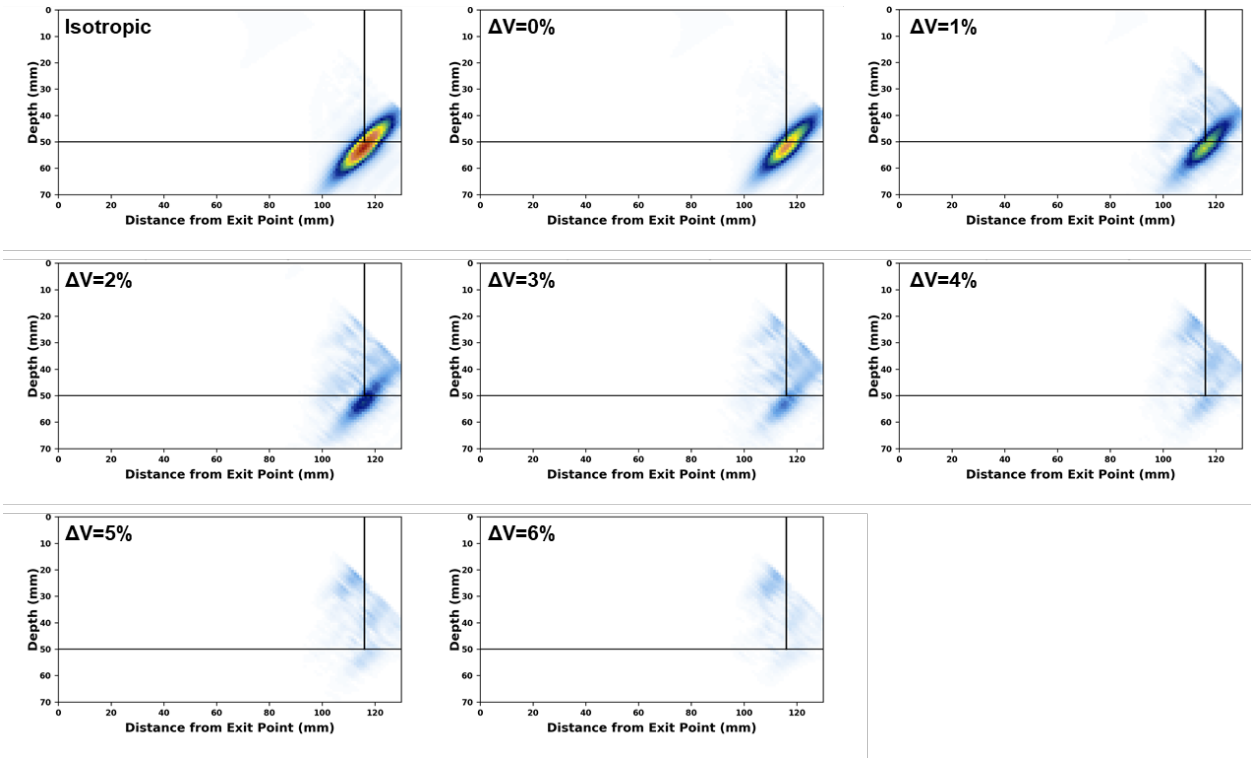

Figure 5.12. Flaw response simulation results through isotropic material (top left) and through the same Voronoi geometry with different values of ΔV.

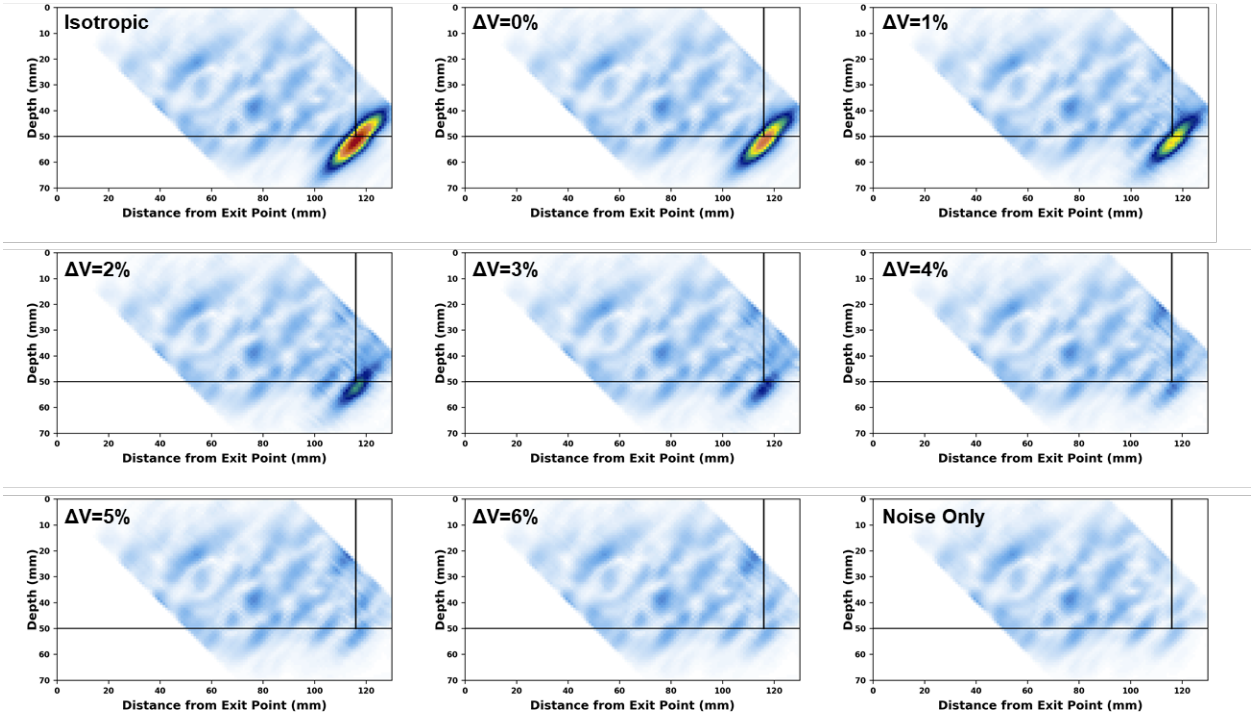

Figure 5.13. Flaw response simulation results through isotropic material (top left) and through the same Voronoi geometry with different values of ΔV with noise (bottom right) added to each result.

5.3.3 Background and Foreground Noise

PNNL noticed that noise in empirical scans of CASS materials appears to be composed of a consistent, low-level background interspersed with brighter noise echoes. The empirical noise can therefore be thought of as two independent, superimposed noise fields: a relatively low-level and spatially smooth "background" noise and a "foreground" noise with stronger, discrete echoes. The background noise likely originates from the cumulative scatter, while the foreground noise originates from a few individual grains that are serendipitously shaped or oriented to form a coherent echo. PNNL observed that CIVA can simulate the background noise by using a large ρ or foreground noise by using a small ρ. However, efforts to simulate both types of noise at once were unsatisfactory. A better reproduction of simulated CASS noise was generated by simulating background and foreground noise fields separately then combining the results. For the background noise, ρ=1 reflector/mm^3 and A=1 were used (values of ρ higher than 1 take too long to simulate). For the foreground noise, ρ=0.001 reflector/mm^3 and A=200 were used. In addition, a 100% TW (i.e., end-of-block) flaw response was simulated while using a Voronoi model of CASS. Other settings were: 1 MHz TRL probe, 2 mm (0.08 in.) step size (i.e., 2 mm [0.08 in.] between A-scans), ΔV=3%, 160×80×50 mm^3 (6.3×3.1×2.0 in.^{3}) specimen block, and 10,000 equiaxed Voronoi cells (average region size 64 mm^3 [0.004 in.^{3}]), the equivalent volume of 4×4×4 mm^3 (0.16×0.16×0.16 in.^{3} cubes). A single B-scan was simulated. Data were exported and processed in Python.

The flaw response was then combined with the two different noise simulations. Figure 5.14 shows the flaw response with the background noise, foreground noise, both combined, and an empirical scan. The empirical data shows an example B-scan (single slice) from the end-of-block scan of OND-P-5 (see Figure 5.7). The mean grain size of OND-P-5 was measured to be

about 4 mm (0.16 in.), approximately matching the *d* of the model specimen, which had ≈64 mm^3 (0.004 $in.^3$) Voronoi grains. In the combined image, the foreground noise was multiplied by two and the background noise by 0.5 to better match signal intensities of the empirical scan. Note that the noise texture in the empirical scan appears to be “grainier” with smaller discrete echoes, much different than the blurred appearance of the noise peaks in the simulations.

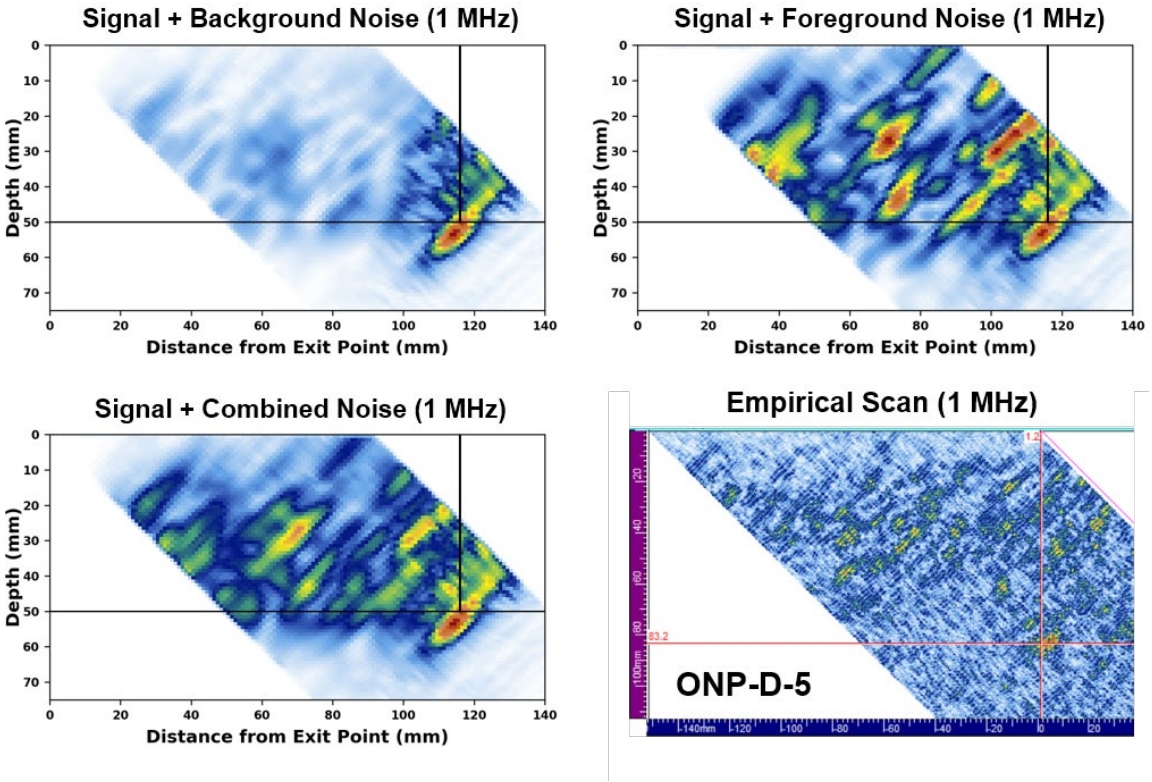

Figure 5.14. Top row: Flaw response with background noise, and flaw response with foreground noise. Bottom row: Combined noise and flaw response, and an empirical scan.

The foreground noise simulation was repeated with higher spatial resolution to try to match the grainy texture of the empirical noise. The same scan region was covered with 0.2 mm (0.008 in.) step sizes to better match the empirical scan resolution to see if a grainier noise appearance could be achieved with increased resolution. However, results did not show any difference in simulated noise texture.

A second attempt to match the texture of the empirical noise was undertaken by increasing the probe frequency in the simulation to 2 MHz. Figure 5.15 shows the results with the background noise, foreground noise, and combined noise (the background noise was multiplied by three in this case). Results show that increasing the probe frequency indeed reduces the size of the noise echoes in simulations. There are several drawbacks to using a higher frequency for simulating noise than for the flaw response simulation. 1) Changing the probe frequency adds another level of complexity because the probe frequency becomes a free parameter. 2) Higher frequencies increase simulation time. 3) An additional simulation setup is required with the higher frequency probe.

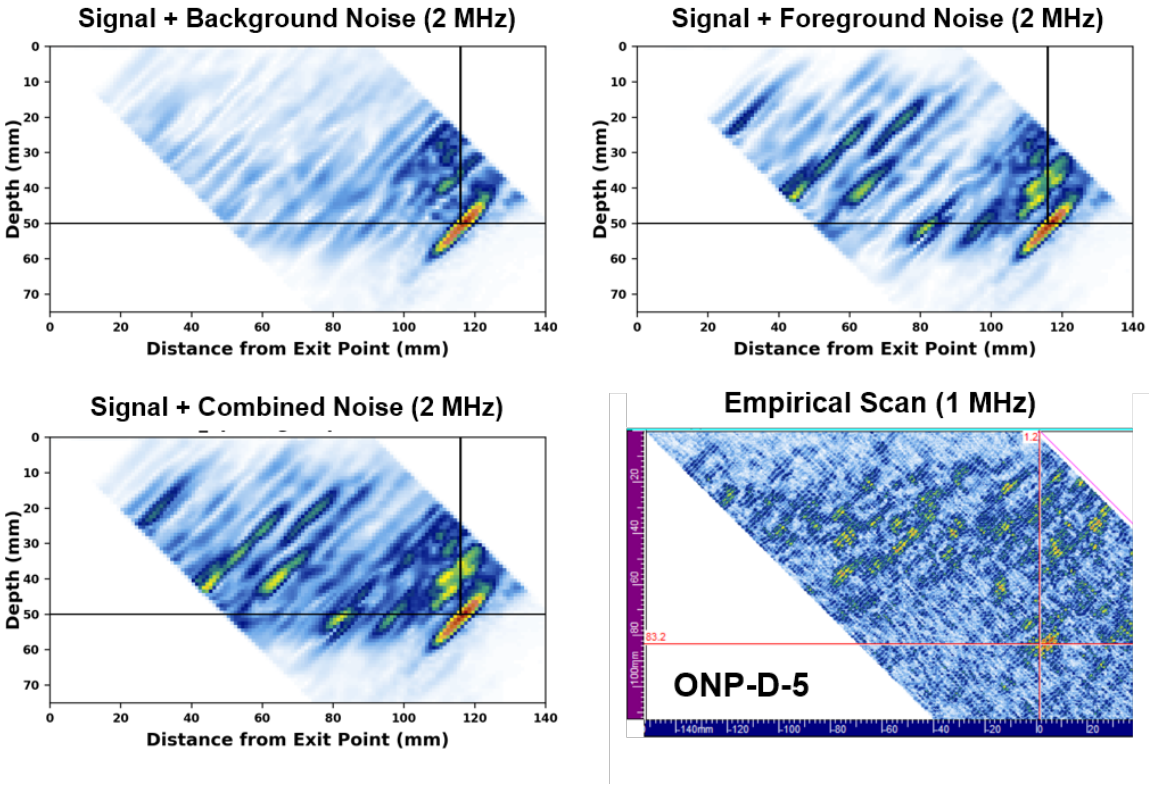

Figure 5.15. Top row: Flaw response with background noise, and flaw response with foreground noise. Bottom row: Combined noise and flaw response, and an empirical scan.

Figures 5.14 and 5.15 illustrate a method of simulating coarse-grained noise. Neither figure shows noise fields that are comparable to or representative of those observed empirically. In PNNL's testing, no methods of satisfactory coarse-grained noise field simulations were found.

5.3.4 Iterative Approach to Simulating Backscattered Noise

The CIVA user's manual notes that "realistic description of scattered noise in terms of the two parameters density and amplitude *requires use of experimental calibration*" (emphasis added). Segmentation and statistical analysis tools (which PNNL has not yet explored) are suggested and may be useful for streamlining the process. This section explains the process step-by-step. The approach is essentially trial-and-error and is the basis of the method described in this and preceding sections. Individual users may find a subset or combination of procedures to be helpful.

A description of how to simulate noise in coarse-grained specimens is provided in (Chatillion et al. 2003). This paper is provided as a reference for structural noise modeling in the CIVA 2020 user's manual and outlines the following steps to simulate empirical noise:

1. Create a noise field in CIVA and segment the noise peaks. Segmentation is a method in CIVA of identifying the location and extent of noise echoes that exceed a threshold value.
2. In CIVA, create a histogram of the segmented peaks based on amplitude.

3. In the empirical data, fill the active window with the noise field (i.e., gate out a region that contains only noise).
4. In the empirical data, adjust the soft gain as necessary so that the maximum noise is at 100%.
5. Create a histogram of the empirical noise field.
6. Compare the simulated and empirical histograms. Qualitatively observe the characteristics of the histogram peak.
7. In CIVA, tune the scatterer distribution (the parameter *A*) until the histograms are similarly shaped. This will require running multiple simulations.
8. In CIVA, adjust the density (ρ) to achieve a simulated noise level similar to the empirical one. This will require running multiple simulations.

PNNL attempted to implement these steps but kept getting different simulation results in CIVA from the exact same model setup. That is because CIVA automatically randomizes the locations of the noise scatterers before each simulation, so no two noise simulations are alike. One way to avoid this is to increase ρ to a high enough value so that the distribution of scatterers does not substantially vary from simulation to simulations, but this causes simulation time to increase dramatically. Also, the outlined approach seems to only work if the simulated and experimental data can somehow be normalized or calibrated to each other. PNNL found matching noise characteristics in simulated to real data to be challenging, time-consuming, and frustrating. The process might be simplified if the empirical data were being analyzed in CIVA. (PNNL was using UltraVision because CIVA 2020 could not read UltraVision data; however, CIVA 2021 can read UltraVision data exported in .txt format.) At any rate, Chatillion et al. (2003) points out that the noise characteristics in CIVA are a property of the specimen and not the probe. Thus, once noise properties are established, they can be kept constant for simulating different probe parameters. PNNL did not test this.

5.4 Summary

PNNL evaluated three types of simulated noise in CIVA: structural, polycrystalline, and surface. Structural noise can be added to flaw-response simulations in CIVA by using the built-in noise feature, and the noise response can be changed by adjusting the noise-scatter density ρ and the amplitude *A*. Note that some model scenarios, such as Voronoi regions, do not support structural noise (in CIVA 2020). Initial tests of the polycrystalline method showed that it can generate noise, but this method lacks advantages over the structural approach in the scenarios tested. Finally, a multifaceted flaw placed at the surface of the specimen can effectively simulate surface noise or clutter by providing echo responses that are similar in appearance to empirical surface noise. The shape of the surface flaw should be iterated until the desired noise signature is achieved.

The structural noise option in CIVA is the most straightforward method of simulating noise. There are some important things to remember:

- Structural noise reflectors are always the same size and shape. The spatial size of the echo is determined by the wavelength, so lower frequencies produce spatially larger noise echoes than higher frequencies.
- Adding noise to a model does not diminish the flaw response amplitude. However, if a noise reflector is coincident with the flaw, then the flaw signal will appear to increase. When

simulating noise, the SNR is affected only by a change in noise and not by a change of signal. The user should be careful not to misinterpret results with regard to flaw detectability.

- CIVA noise reflectors are randomly redistributed for each simulation. Rerunning a simulation with identical parameters gives a different noise pattern. A controlled test of flaw response with variable flaw parameters but a constant noise distribution is not possible in CIVA 2020. This can be a problem if, for example, a noise reflector happens to lie on the flaw in some simulations but not in others, because the noise added to the flaw response makes the flaw response appear stronger.
- Noise reflectors are on occasion randomly assigned to the simulation plane. Occasionally, a reflector will end up in the simulation plane, resulting in a very strong and anomalous-looking noise response. It may be best to rerun such instances.
- Simulation time is a very strong function of ρ, so minimizing ρ is advantageous. In PNNL's tests, simulation times were increased by well over a factor of 10,000 from a simulation with no noise to one with ρ = 100 reflector/mm^3. Low reflector densities, such as ρ ≈ 0.001 to 0.01 reflector/mm^3, have little impact on simulation time in the scenarios that PNNL tested. Again, the appropriate value of ρ should be determined iteratively.

Flaw response simulations in CASS materials can be simulated using Voronoi regions as models for grains, but structural noise cannot be added to simulations with Voronoi regions. That is, one can simulate noise or a flaw response in a coarse-grained specimen, but not both simultaneously. Adding granular noise must be done in stages by combining separate noise and flaw response simulations. Accurate flaw-response modeling in Voronoi CASS models will require parametric studies to determine ideal values of d, ΔV, ρ, and *A* for a given specimen, particularly if a polished and etched face of the specimen to reveal the grain size and structure is unavailable. Achieving the desired noise level and appearance is an iterative approach, and empirical noise can be used to guide the selection of CIVA parameters.

There are pros and cons to simulating noise. The pros are: 1) simulated noise can give a more accurate picture of the sound field; 2) simulated noise can help predict SNR for flaw detectability, especially in coarse-grained materials. On the other hand, the downsides of simulating noise are: 1) simulating noise can significantly extend computation time; 2) simulated noise and the resulting SNR depend strongly on correctly determining multiple parameters through a time-consuming iterative process that requires access to empirical data. Errors in this process can cause an unrealistic or incorrect noise field. If the required empirical data are not available, then simulated noise should not be considered reliable.

PNNL anticipates that adding noise to simulations of coarse-grained materials will be helpful to predicting flaw detectability, and it should be done when feasible. Even if the simulated noise field does not exactly resemble that of empirical scans, there are benefits to being able to predict flaw detectability in the presence of structural noise. As with other simulation scenarios, running multiple simulations with a range of parameters can help establish nominal results and potential outlier cases.

6.0 Attenuation

6.1 Introduction

Attenuation is the reduction of the sound-field amplitude as it propagates through a medium. Acoustic attenuation can be observed in every material and is typically a strong function of frequency, with higher frequencies more strongly attenuated than lower frequencies. Applying attenuation to simulated scenarios is usually not necessary unless relatively high frequencies are used (i.e., ≳ 5 MHz) or strongly attenuating materials are being modeled, such as many types of polymers or aggregates. Based on PNNL's experience to date, adding attenuation to CIVA models is straightforward and results in virtually no increase to simulation time. This section describes methods for adding attenuation to simple and coarse-grained CIVA models.

6.2 Sound Field Attenuation in CIVA

Attenuation in CIVA is defined in units of dB/mm, or decibels of signal loss per millimeter of sound path, and it can be characterized as a frequency-dependent function using either an exponential or a polynomial. The type of frequency dependence must be determined empirically or theoretically, since it is an inherent property of the material. For materials that exhibit exponential attenuation, the user enters three parameters: a wave frequency f_0 (in MHz), the attenuation at that frequency α_0 (in dB/mm), and the exponent, or power of the attenuation rate p. These values are unique to each material and should be determined empirically or found in the literature. CIVA calculates the attenuation α (in dB/mm) for any frequency f using the following equation:

$$\alpha = \alpha_0 \left(\frac{f}{f_0}\right)^p$$

For the polynomial attenuation, the user enters relevant coefficients and exponents, depending on the frequency dependent function. For example, from Ono (2020), the attenuation of longitudinal waves in 304 stainless steel is of the form:

$$\alpha = C_d f + C_R f^4$$

where α is the attenuation in dB/mm, C_d is the coefficient of attenuation due to damping of longitudinal waves in dB/mm/MHz, and C_R is the coefficient of attenuation due to Rayleigh scattering in dB/mm/MHz^4. Rayleigh scattering occurs when the distance between scattering centers is much smaller than the wavelength and exhibits attenuation dependent on the fourth power of the frequency. Rayleigh scattering typically becomes significant at about 5–10 MHz (Ono 2020). (By comparison, stochastic scattering, where the spacing between scattering centers is about the same as the wavelength, results in attenuation proportional to the square of the frequency and is significant for most metals at frequencies ≳ 20 MHz.) In this example, the user would enter the value of C_d with 1 for the corresponding exponent and the value of C_R with 4 for the exponent. It should be noted that CIVA provides default sound speed and density values for many materials, but coefficients or exponents for attenuation are not included, since measured values can vary considerably from specimen to specimen, even if they comprise nominally the same material. Values for sound attenuation coefficients and exponents for many common materials can be found in published literature (c.f., Ono (2020)).

Figure 6.1 shows four examples of CIVA-generated attenuation versus frequency graphs defined in an isotropic material. The bottom example is a polynomial with C_d = 0.0212 dB/mm/MHz and $C_R = 7.78\times10^{-6}$ dB/mm/MHz^4. These values were taken from non-magnetic 304 stainless steel (test F86 in Table 2 of Ono (2020)). The black line shows the attenuation as a function of frequency; at 5 MHz, the attenuation is 0.111 dB/mm. The attenuation was then defined for the other cases to also be 0.111 dB/mm at f_0=5 MHz. The first graph shows a constant attenuation, which was defined using the power law option with an exponent of 0. Here the black line is horizontal (i.e., attenuation is independent of frequency). The second and third examples show a power law attenuation with an exponent of 2 and 4, respectively. Note that the vertical axes have different scales, but the attenuation at 5 MHz—at the red cursor—is always 0.111 dB/mm.

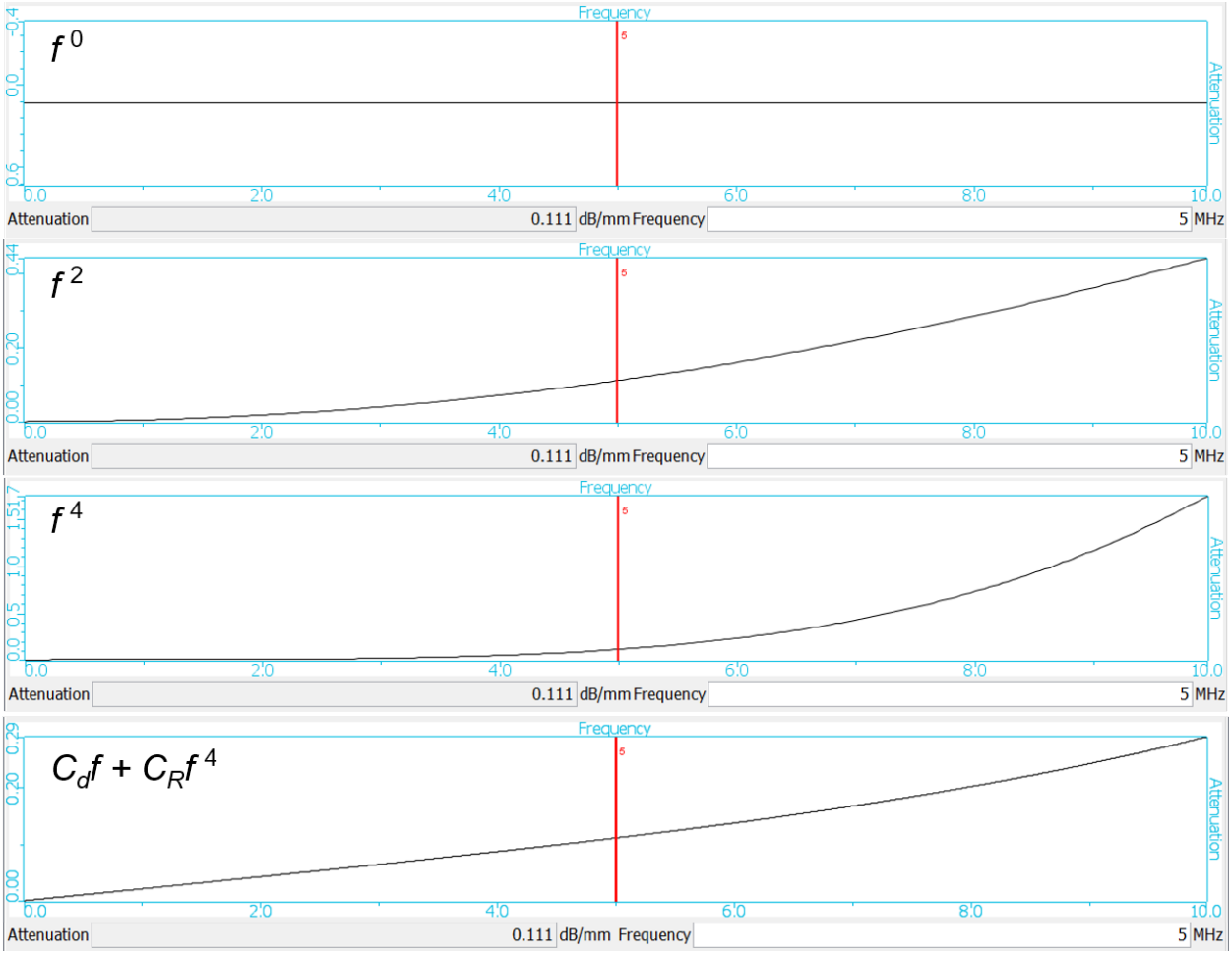

Figure 6.1. Graphs showing attenuation vs frequency for four different scenarios.

From the attenuation curves given in CIVA (shown in Figure 6.1), the user can determine the attenuation at other relevant frequencies. For example, the modeled 5 MHz probe was assigned a −6 dB bandwidth of 80%. Thus, the −6 dB drop-off of the transmitted signal occurred at 3 MHz and 7 MHz. Using the attenuation curves and the red vertical cursor, the user can obtain attenuation values at those frequencies, as shown in Table 6.1.

Table 6.1. Attenuation values at different frequencies for different attenuation laws

Attenuation Law	Atten. at 3 MHz (dB/mm)	Atten. at 5 MHz (dB/mm)	Atten at 7 MHz (dB/mm)
Constant	0.111	0.111	0.111
Power, $p = 2$	0.040	0.111	0.218
Power, $p = 4$	0.014	0.111	0.426
Polynomial	0.064	0.111	0.167

Figure 6.2 shows the simulated sound field for the scenarios shown in Figure 6.1 for a 12.7 mm (0.5 in.) diameter 5 MHz probe with an 80% bandwidth. Isosonic curves, or contour lines of constant sound intensity, were added at the −3 dB, −6 dB, and −12 dB levels to help the eye differentiate between the sound fields. The unattenuated scenario shows the sound field propagating through the full 200 mm of simulated material. It is not realistic to select this probe for inspections of materials of that thickness; these simulations are illustrative. The attenuated scenarios show significant truncation of the overall sound field, but the depths of the near field/far field transitions, indicated by the red dashed lines, are not affected much. The sound field is not dramatically affected at depths of up to about 50 mm (2 in.) in any of the cases.

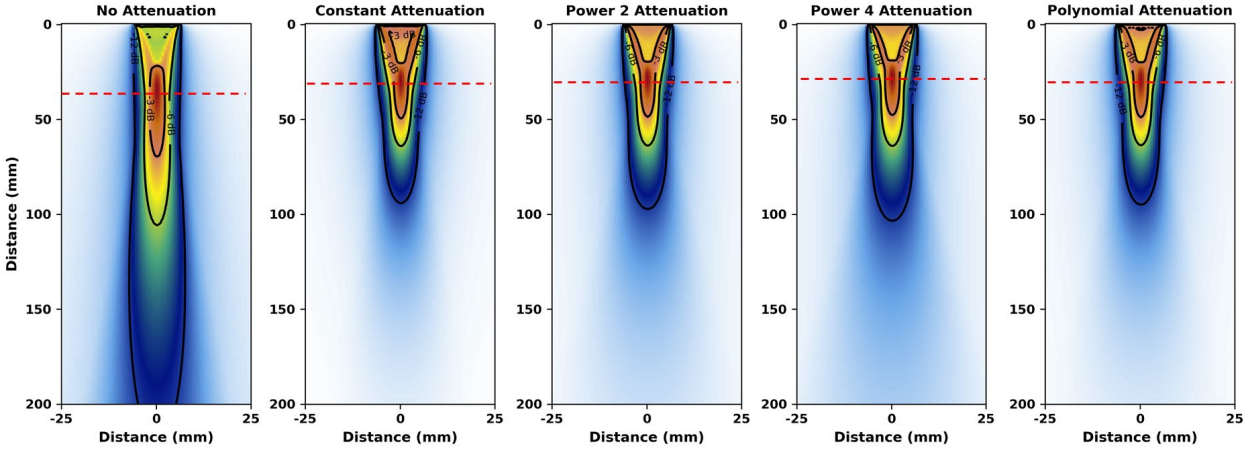

Figure 6.2. Simulated sound fields at 5 MHz for different attenuation laws. The dashed lines indicate the position of the near field/far field transition.

Figure 6.3 shows simulated sound fields for a 25 mm (1 in.) diameter 2 MHz probe with an 80% bandwidth using the same attenuation scenarios. A different probe diameter was simulated in order to generate similar unattenuated sound fields at the lower frequency to facilitate visual comparison to the 5 MHz results. This figure illustrates that the lower frequency probe is not strongly affected by the frequency-dependent scenarios (i.e., the power law and the polynomial attenuation laws), since the attenuation decreases as the frequency decreases. Note that both probes are similarly affected by the constant attenuation scenario; this is an expected outcome since the attenuation is the same at both frequencies. As with the 5 MHz scenario, the location of the near field/far field transition is not strongly affected by attenuation at 2 MHz.

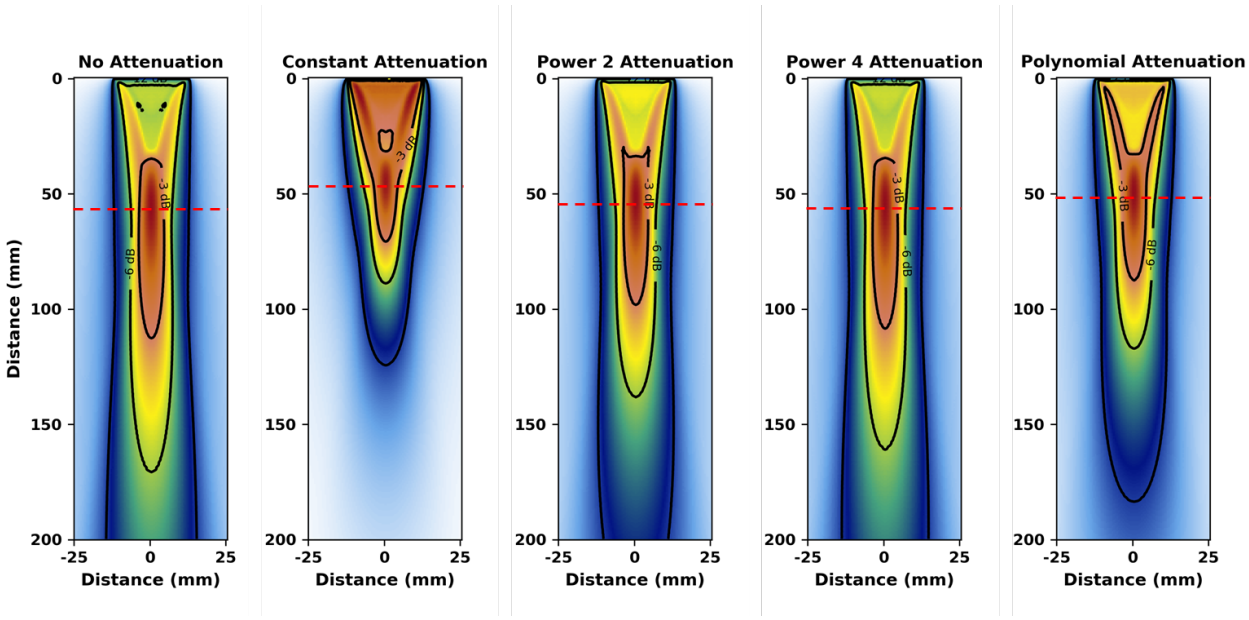

Figure 6.3. Simulated sound fields at 2 MHz for different attenuation laws. The dashed lines indicate the positions of the near field/far field transitions.

Results show that it is probably unnecessary to add attenuation to simulations in steel at frequencies ≤ 5 MHz, since the sound field at relevant depths is not strongly affected. Ultimately, the decision to add attenuation will depend on the material type and actual sound path in addition to the probe frequency. For example, field examinations of coarse-grained material would be conducted at relatively low frequencies of ≤ 1 MHz and examinations of austenitic welds would be at frequencies up to about 2 MHz. Attenuation in these materials may be best simulated by adjusting the material scattering properties, such as in Figure 5.12.

For flaw response simulations, attenuation would affect the echo amplitude; however, it is not useful to simulate attenuation through homogeneous materials unless the probe and/or signal response is well calibrated to a standard signal. Applying attenuation through a homogeneous material will simply result in a reduction of the relative echo strength, but the absolute echo signal will still be arbitrary. In other words, without a calibrated probe or signal response in the simulation, all sound field intensities and signal responses are arbitrary. CIVA does not model the probe electronics or piezoelectric elements; thus, it is essential to use a calibration signal when absolute measurements are needed. Indeed, the CIVA user's manual emphasizes the need for a calibration signal:

> *It is compulsory to normalize [a] simulated amplitude by the simulated echo amplitude from the calibration flaw. … The value of this ratio is consequently experimentally measurable and can lead to comparisons between simulation and experience. The use of this amplitude normalization by a calibration technique allows [the user] to overcome the complex modelling of the electro-acoustic transduction (not modelled in CIVA).*

For example, a weak signal response in an empirical scan could result from a variety of sources, such as a poorly-performing probe, a weak transmit pulse, poor coupling, or attenuation in the material. Without a simulation calibration signal, simulated attenuation will only allow the user to observe relative changes to sound fields and echo responses but will not

provide any direct information on flaw detectability. All direct, quantitative comparisons to empirical results require some type of calibration, such as a side-drilled hole or a corner echo.

When simulating attenuation in CIVA, it is critical to remember to 1) define the appropriate attenuation in the Specimen tab, and 2) activate the attenuation in the Simulation tab by checking the Attenuation box. Also, CIVA allows the user to define directionally-dependent attenuation in anisotropic materials; PNNL has not investigated this feature.

6.3 Attenuation due to Surface Roughness

Another option for adding attenuation to a simulation is by using the Roughness option in the Geometry tab of the Specimen tab. This option adds virtual surface roughness to the specimen model (the roughness is not shown in the model diagram). The user enters this as a roughness Ra value in units of μm.[11] According to the CIVA user's manual, a roughness factor is added to the specimen/probe interface (i.e., the surface that interfaces with the beam after it exits the probe), and "the effect is modelled as a modification of the transmission coefficient, [and is] a function of the frequency." That is, the transmission coefficient of the sound into the component is reduced with increased surface roughness and increasing probe frequency (Rodriguez et al. 2003). Figure 6.4 shows an example of beam maps simulated at 2 MHz with different values of roughness, ranging from 0 μm to 200 μm. Note that the shape of the beam is constant with increasing roughness, only the intensity changes. This is because changes to the transmission coefficient only occur at the surface. The overall effect can be likened to that of a poorly-coupled probe or a probe with reduced output intensity. In other words, reduced probe output can be modeled by using the surface roughness option as a surrogate. For comparison to empirical results, a calibration flaw response signal is required.

[11] https://www.sciencedirect.com/topics/chemistry/surface-roughness

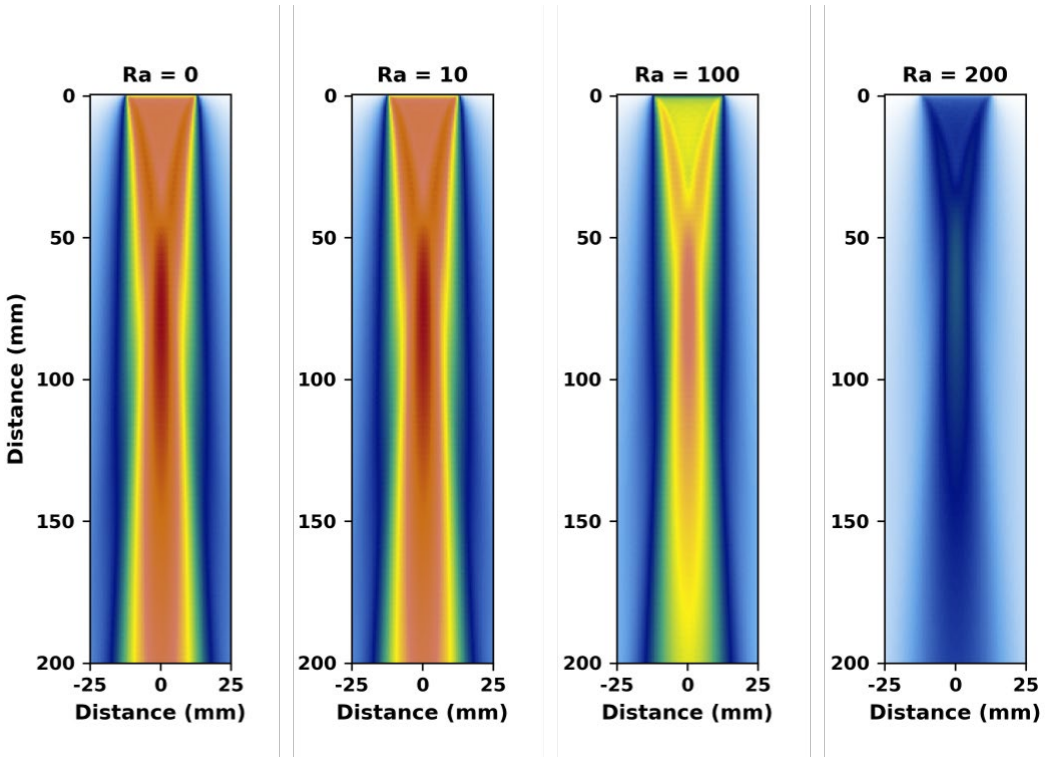

Figure 6.4. Simulated beam maps for different values of surface roughness (Ra). Ra is in units of microns.

The surface roughness affects the transmission coefficient, so attenuation with this method is fundamentally different from the sound field attenuation described in Section 6.2. Figure 6.5 is a comparative illustration of simulations done with surface roughness and with specimen attenuation. Attenuation values were set such that the simulations in each row gave the same peak beam intensity. Attenuation in the bottom row was more aggressive than in the top row. With strong specimen attenuation, the sound beam barely penetrates, whereas high surface roughness diminishes the overall beam intensity. Results show that the simulated sound fields with the different types of attenuation are fundamentally distinct. The attenuation from surface roughness diminished the entire sound field uniformly, whereas the specimen attenuation reduced the sound field intensity with increasing depth. Before adding attenuation to a model, the user should understand the difference between the attenuation types and the effects each will have on the simulation.

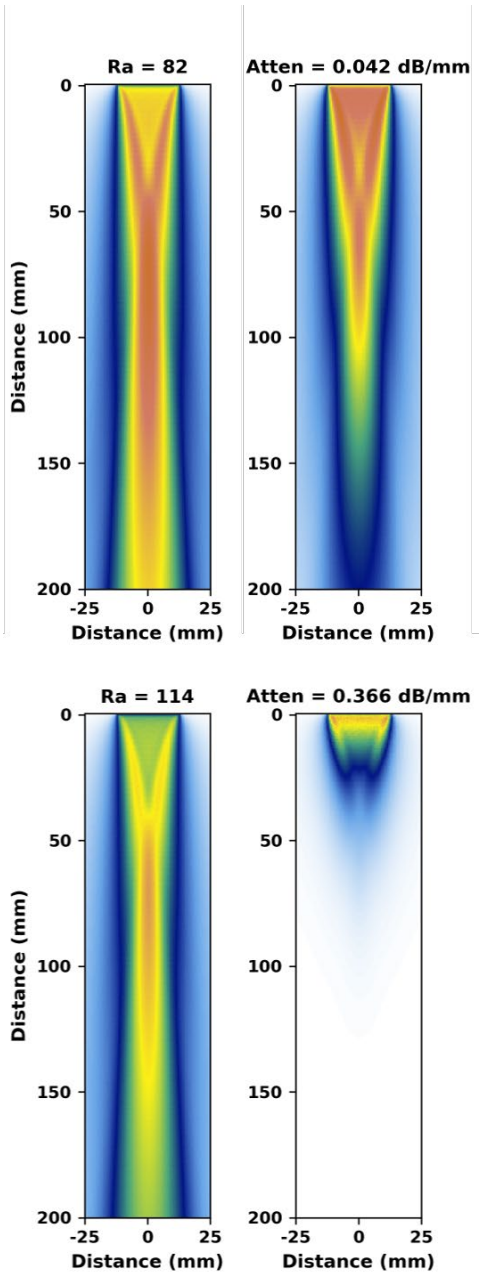

Figure 6.5. Examples comparing surface roughness (or transmission attenuation, left) and specimen attenuation (right). Attenuation values were chosen such that each row has the same maximum signal intensity. Surface roughness diminishes the entire sound field uniformly, whereas specimen attenuation causes a reduction in sound intensity with increasing depth.

6.4 Attenuation in a Weld Model

To show an example of the application of specimen attenuation in a beam simulation, a simulation was computed using the 256-pixel EBSD weld model. This simulation used a 2.0 MHz PA probe with a 45° focal law and a focal depth of 24 mm. A constant attenuation of 0.1 dB/mm was added. This level of attenuation is similar to the empirically-measured

attenuation in the specimen used to develop the model. Note that attenuation must be added to *each region* of the specimen model. The results of the beam simulations are shown in Figure 6.6. The left panel shows the beam simulation without attenuation, and the right panel includes attenuation. The beam maps were normalized to the peak signal of the non-attenuation result. Results illustrate that attenuation and scatter can be applied in the same simulation to affect the sound intensity. Again, it is important to note that the absolute signal intensity is arbitrary, and quantitative results from flaw response simulations can only be achieved by using calibration signals. Adding a flaw to the far side of the weld model would result in a stronger flaw response in the unattenuated scenario than in the attenuated one. However, there is no way to know which flaw response more accurately predicts reality without a calibration signal—preferably from an SDH in both the simulation and empirical scans—that can be used to normalize the simulated and empirical results to one another. It is important to note that the example shown here is one specific scenario and that attenuation should be customized for every scenario independently.

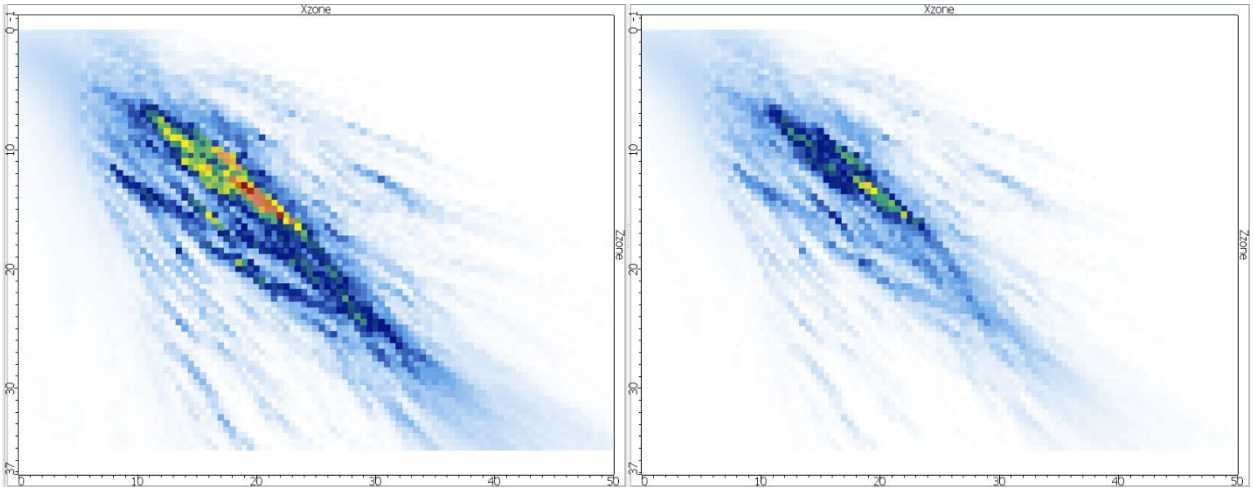

Figure 6.6. Through-weld beam simulations without (left) and with (right) attenuation.

Adding attenuation to a CIVA 2020 specimen model that comprises multiple regions requires defining the attenuation in each region independently. If the same attenuation law is to be assigned to all the regions, then they can all be changed *en masse* in CIVA by right-clicking in the Material tab of the Specimen tab. However, if different regions require different attenuation laws, then the attenuation can be assigned manually region-by-region. If there are many regions, the specimen .xml file can be exported and edited outside of CIVA using a program such as MATLAB or the Find/Replace function of a text editor. PNNL has successfully tested all of these approaches.

6.5 Simulated Signal-to-noise Ratio and Attenuation in CASS

Attenuation in empirical scans of CASS material is caused largely by sound reflection and refraction (manifest as scatter and redirection) as the sound passes through grains of different size and orientation, typically in the stochastic or geometric scattering regimes (Ryzy et al. 2018). As discussed above, CIVA can simulate attenuation directly by allowing the user to enter attenuation parameters, such as the attenuation rate (in dB/mm) and the frequency dependence (in the form of an exponential or power relationship). It should be noted, however, that CIVA does not give the option for adding attenuation when using the Voronoi model for coarse-

grained material. Therefore, PNNL has investigated two alternative approaches for simulating attenuation in Voronoi-based CASS materials.

Empirical scans of CASS tend to have high structural noise. The first method simulates an "apparent" attenuation by substituting the simulation of noise for attenuation. Noise has the effect of reducing the *relative* signal intensity of an echo, or the signal intensity of an echo relative to structural noise. The echo signal is not reduced in an absolute sense, but as the noise level increases, the SNR level decreases and gives the effect of an attenuated echo response. This apparent attenuation can be tuned by increasing or decreasing the intensity of the simulated noise. Once the noise level overwhelms the echo signal, then the SNR becomes ≤1 and the beam is essentially fully attenuated. This method is not ideal because it will always result in an echo signal (whereas a strongly attenuated beam will not always produce an echo signal), even though the signal may be weaker than the average noise level. This approach may require very high noise amplitudes.

The second method is to leverage the Voronoi parameters to imitate attenuation. It is well known that scattering and attenuation are closely related, as beam scatter leads to signal loss (Dorval et al. 2013; Feuilly et al. 2009; Ryzy et al. 2018). Coarse-grained materials are strongly attenuative due to the high degree of scatter that occurs. CIVA 2020 does not model noise or scattered signals with Voronoi regions, but the regions do deteriorate the echo response via beam scatter, such as shown in Figure 5.12. Two variables can be used to affect attenuation with Voronoi regions: the total number of regions and ΔV, which is the range of velocities assigned to the Voronoi regions. Here, PNNL explored the effect of ΔV on beam attenuation and SNR in a coarse-grain Voronoi model. The SNR ratios of the images shown in Figure 5.13 were measured. Each image represents stronger scatter due to increased ΔV but has the same noise background. Table 6.2 shows the measured echo signal intensity for each ΔV and the calculated SNR, and Figure 6.7 shows a graph of the SNR as a function of ΔV. For reference, the dashed line in the figure represents the 3:1 SNR level. Results show that, for the given noise field and the given grain size, a ΔV of 3% results in an SNR of about 3 and probably represents a typical coarse-grained CASS exam. Values of ΔV that are higher than this represent worse (i.e., lower SNR) scenarios, and values below this are more favorable scenarios. Again, it is important to remember that *adjusting ΔV does not affect the modeled grain size, but it does affect the ability of the ensemble of grains to scatter the sound*. Finding the optimal value of ΔV for a given grain size is important for accurately simulating beam attenuation in coarse-grained materials.

Table 6.2. Signal, SNR, and attenuation for a given simulation scenario.

ΔV (%)	Echo Signal (arb. units)	SNR*	Apparent Atten.** (dB/mm)
0	83.1	12.2	0.017
1	62.0	9.1	0.030
2	38.3	5.6	0.052
3	22.5	3.3	0.076
4	11.9	1.8	0.106
5	6.6	1.0	0.132
6	3.7	0.5	0.159

* mean noise = 6.8
** The reference signal through an isotropic specimen was 120 (arb. units) and the metal path was 190 mm.

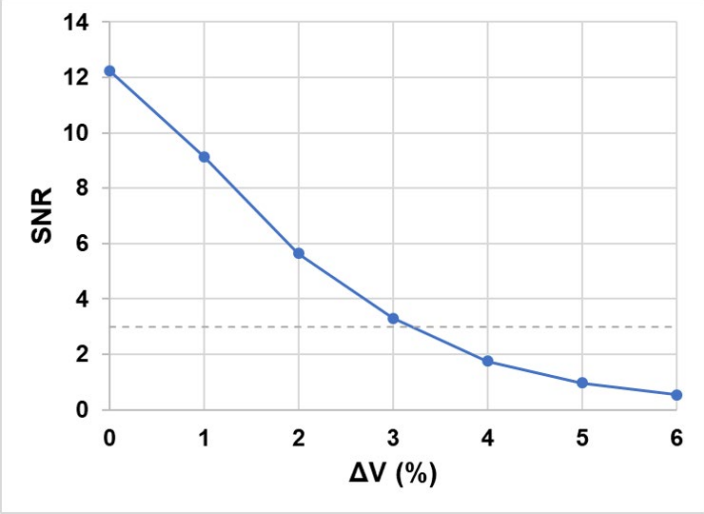

Figure 6.7. Relationship between SNR and ΔV in Voronoi simulations. The dashed line indicates the 3:1 SNR level.

Next, the apparent attenuation was calculated by comparing the signal loss of each ΔV case to the signal of the isotropic case for the given metal path; this was done independently of any noise. Table 6.2 shows the calculated attenuation values, and Figure 6.8 shows a graph of the relationship between ΔV and attenuation. The relationship appears to be nearly linear for ΔV≥1%.[12] Sakamoto et al. (2012) suggest that attenuation in equiaxed CASS is about 0.75 dB/mm. This agrees very well with the attenuation value obtained here with ΔV=3%. Therefore, both the SNR and attenuation results of this example suggest that ΔV=3% provides good approximations of both SNR and attenuation levels (for the simulation parameters chosen)

[12] It is unclear what happens in CIVA when ΔV is defined as 0%. Results were expected with ΔV=0% to match those with an isotropic specimen, since there is no velocity variation from grain to grain and the specimen should be effectively isotropic. However, the echo response in the isotropic model was about 1.4 times higher than that of the ΔV=0% Voronoi model.

that have been empirically observed in CASS. *It is important to remember that these results are only illustrative and represent one simulated noise background and one average size of Voronoi regions.* For example, if we had used a higher noise amplitude *A*, then the noise level would have been higher and the SNR results would have been lower. Such a change, however, would not have affected the attenuation values. To optimize ΔV for a given scenario, or other parameters that may affect signal intensity, it is imperative to conduct parametric studies by systematically varying the noise parameters and Voronoi parameters. It is essential to have access to empirical data to guide the parametric study.

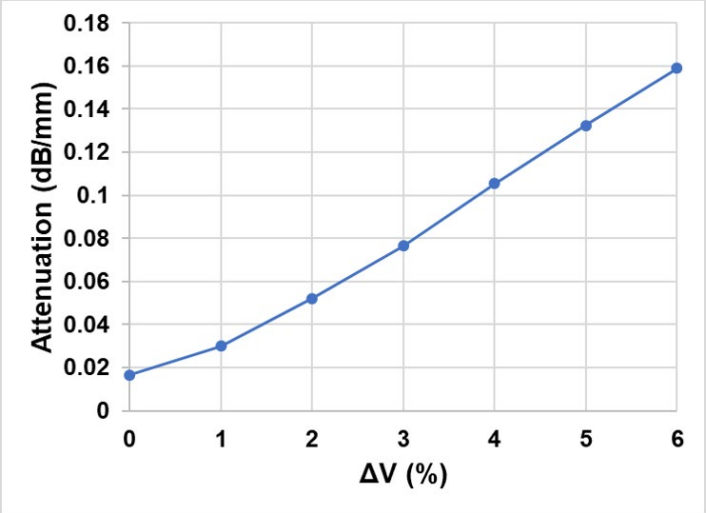

Figure 6.8. The relationship between calculated attenuation and ΔV in Voronoi simulations.

It can be an onerous task to simultaneously and iteratively optimize ρ, *A*, ΔV, and the Voronoi region size *d*. Below is some step-by-step guidance to help with the process.

1. Determine a typical grain size and shape. This may require referring to empirical data, a body of literature, engineering judgment, and/or polished and etched specimens similar to the one that is being modeled. It is ideal to repeat the process for a range of grain sizes in order to explore a range of possible scenarios.
2. Using the typical grain size, calculate the number of Voronoi regions needed for the modeled specimen. For example, if the typical grain is determined to be 5 mm (0.2 in.) across, then a typical grain volume is approximately $(5\ \text{mm})^3$, or 125 mm^3 (0.008 in.^3). Calculate the specimen volume from the dimensions entered in CIVA, then divide the specimen volume by the grain volume. For example, if the specimen is 100 mm × 100 mm × 40 mm (3.9 in. × 3.9 in. × 1.6 in.), then the volume is 400,000 mm^3 (24.4 in.^3), and the number of Voronoi regions needed is 400,000/125 = 3,200. Note that simulation times increase with higher numbers of Voronoi regions. It is best to minimize the volume of the modeled specimen.
3. Determine ΔV based on attenuation. If an attenuation value is known from empirical data, then ΔV can be determined from a parametric study using a backwall echo, a rectangular flaw, or other reflector. It is important to remember to use the full metal path *x* in the calculation (i.e., *x* is twice the distance from the probe to the backwall or flaw). The attenuation in dB/mm is calculated by:

$$Attenuation\left(^{db}/_{mm}\right) = \frac{1}{x} 20 \log_{10}\left(\frac{S}{S_0}\right)$$

where S is the attenuated signal and S_0 is the unattenuated signal, such as through isotropic material. Alternatively, an echo train can be simulated and the attenuation calculated by measuring the decay rate of successive echoes.

4. Simulate a desired noise field. This is largely qualitative and depends on the noise patterns or characteristics observed empirically. For example, a coarse noise field with many strong echoes, such as in coarse-grained CASS, will require sparse noise scattering centers. Select a fixed value of A, and iterate ρ until the desired noise field is found. Note that it is best to use as small of a ρ as possible to reduce simulation time. Also, combining sparse and fine noise fields (i.e., “foreground” and “background” noise), as shown in Section 5.3.3, may be desired.
5. Iteratively determine the value of A that provides the desired SNR for the values ρ and ΔV that have already been determined.
6. Because CIVA does not simulate noise with Voronoi regions, separate simulations will need to be run to generate a noise field and a flaw response. Run the simulations with the parameters determined above. If changes are needed, repeat any necessary steps. Use care to only change one variable at a time.

6.6 Summary

Applying specimen attenuation to CIVA models is straightforward but does require some knowledge of attenuation coefficients and frequency dependence. Attenuation is generally only needed for simulation scenarios where the frequency is high, i.e., at frequencies $\gtrsim$ 5 MHz, although the importance of attenuation will have to be evaluated on a case-by-case basis. Attenuation of the transmitted sound can be applied in CIVA by using the surface roughness parameter. Calibration signals are critical for quantitative evaluation of flaw response simulations. Otherwise, the amount of attenuation added to a simulation is arbitrary.

Attenuation can be simulated in Voronoi-based coarse-grained materials by selecting appropriate grain sizes and sound field scattering parameters. The addition of noise can also be used to increase apparent attenuation by reducing the SNR. Step-by-step procedures were described for iteratively optimizing the parameters.

7.0 Summary

This report is the fourth in a series of modeling and simulation studies that have taken place at PNNL over the past several years (Dib et al. 2017; Dib et al. 2018; Jacob et al. 2020). The project has systematically addressed issues of ultrasonic modeling and simulation that are relevant to nuclear NDE. Most of the work to date has focused on CIVA because of its widespread use in the global nuclear industry. These reports have covered issues including:

- Quantitative metrics for measuring simulation outcomes and comparing results to empirical data;
- Understanding uncertainties in simulation results and the role of verification and validation;
- Understanding how variability in parameter selection affects variability in simulation outcomes;
- Comparing beam simulations generated by UltraVision and CIVA;
- Exploring the relationship between beam simulations and flaw response models and whether the former can be used as a surrogate for the latter;
- Developing and testing models of austenitic welds, coarse-grained materials, and dissimilar metal welds;
- Investigating the effects of adding noise to simulations;
- Investigating models of attenuation.

Results from this report and Jacob et al. (2020) show that CASS models can be reliably produced in CIVA using Voronoi regions. These models are inherently 3D, load rapidly, and generate realistic results. However, the Voronoi option in CIVA 2020 is limited to planar or cylindrical specimen geometries, none of which include welds or multi-region models. Models of mixed grain-structures, like combinations of columnar and equiaxed CASS, or of a wide variety of grain sizes can be created outside CIVA and imported, but the final model will be 2D and extruded in the third dimension. Such models tend to be time consuming to create and do not produce as good of results as the 3D Voronoi models.

PNNL tested the ability of CIVA to simulate the sound field and flaw responses in a DMW model. Because of the limitations with the Voronoi option in CIVA, PNNL created DMW models outside of CIVA by combining regions of weld, coarse-grained (i.e., CASS), buttering, and isotropic materials. Sound field distortion and changes to the flaw response echo amplitudes were observed in the simulations, as expected. Results showed that the specimen model, including the flaw location within the model, has a strong impact on the simulated flaw response.

PNNL observed two effects in the beam simulations. First, a “funnel effect,” or a stronger sound field at the near-side base of the weld was observed. This effect was due to sound being funneled by the backwall and fusion plane, and all of the weld simulations predicted the funnel effect. Empirical sound field maps verified the presence of the funnel effect. Simulated flaw response results tended to overemphasize the effect, more so at 60° refraction angles than at

45°, meaning that the simulated flaw response may be overestimated for near-side flaws. Second, PNNL observed a “shadow effect,” where beam scatter through a weld resulted in a weak sound field on the far side of the weld. The shadow effect was also observed empirically. Simulated flaw response results tend to overemphasize the effect, meaning that simulated far-side flaw responses may be underestimated.

The specific Euler angles of the weld models affect flaw detectability, although not dramatically in most cases tested. Results of this work and the previous work (Jacob et al. 2020) suggest that the impact of the specific Euler angle assignments is smaller than that of the number of interfaces and grain sizes. Grain size is important because grains that are too large will not give realistic scatter, while grains that are too small will violate the CIVA high-frequency model assumptions. The ideal weld grain size is probably about one-half of a wavelength. Note that FEM in other platforms (or CIVA Athena 2D) can be used with arbitrarily small grains, but small grains (especially oddly-shaped grains) require a fine mesh and much longer simulation times.

CIVA is well suited for simulating noise, although PNNL found it challenging or impossible to reproduce CASS noise fields. Structural noise is straightforward to simulate in CIVA by entering a couple parameters, but homing in on the parameters that will generate the desired noise field is an iterative process and may require some additional data processing. Simulation times can increase dramatically when noise is added. ID surface noise from corrosion or cladding can be simulated with a multifaceted flaw placed along the surface of the specimen. Again, the desired noise response may require iterative changes to the flaw shape.

Structural noise is not an available option in CIVA 2020 in simulations with Voronoi regions. PNNL showed that separate noise and flaw response simulations can be run and results combined outside of CIVA. Accurate flaw-response modeling with appropriate noise backgrounds in Voronoi CASS models will require parametric studies to determine ideal simulation parameters that agree with empirical scans, particularly if a polished and etched face of the specimen is unavailable. Indeed, for realistically simulating noise, it is particularly important to have empirical data for comparison.

CIVA has multiple options for defining sound attenuation within a specimen model. The user should have some knowledge of relevant attenuation coefficients and frequency dependence. PNNL observed that attenuation is typically only needed for steel simulation scenarios when the frequency is $\gtrsim$ 5 MHz, although the importance of attenuation should be evaluated on a case-by-case basis. For CASS models, attenuation can be simulated in Voronoi materials by selecting appropriate grain sizes and sound field scattering parameters. Noise can also be used to increase apparent attenuation by reducing the SNR. Step-by-step procedures were described for iteratively optimizing the Voronoi and noise parameters. CIVA does not allow the user to directly control the probe transmit power, but the strength of the transmitted sound can be reduced by using the surface roughness parameter. In all cases of simulations with attenuation, the amount of attenuation added to a simulation is arbitrary unless calibration signals are used.

8.0 References

Anderson, M T, S L Crawford, S E Cumblidge, K M Denslow, A Diaz, and S R Doctor. 2007. NUREG/CR-6933; PNNL-16292. *Assessment of Crack Detection in Heavy-Walled Cast Stainless Steel Piping Welds Using Advanced Low-Frequency Ultrasonic Methods* Pacific Northwest National Laboratory. Richland, WA.

Anderson, M T, A Diaz, A D Cinson, S L Crawford, S E Cumblidge, S R Doctor, K M Denslow, and S Ahmed. 2011. NUREG/CR-7113, PNNL-19353 *An Assessment of Ultrasonic Techniques for Far-Side Examinations of Austenitic Stainless Steel Piping Welds.*

Bates, DJ, S R Doctor, PG Heasler, and E Burck. 1987. NUREG/CR-4970, PNL-6266, PISC III Report No. 3. *Stainless Steel Round Robin Test: Centrifugally Cast Stainless Steel Screening Phase.* U.S. Nuclear Regulatory Commission. Washington, D.C.

Bilgen, M, J H Rose, and P B Nagy. 1993. "Ultrasonic inspection, material noise and surface roughness." In *Review of Progress in Quantitative Nondestructive Evaluation*, 1767-1774. Boston, MA: Springer.

Carpentier, C., C. Nageswaran, and Y. Y. Tse. 2010. "Evaluation of a new approach for the inspection of austenitic dissimilar welds using ultrasonic phased array techniques." Proc 10th ECNDT conference, Moscow. ECNDT.

Castellini, P., M. Martarelli, and E. P. Tomasini. 2006. "Laser Doppler Vibrometry: Development of advanced solutions answering to technology's needs." *Mechanical Systems and Signal Processing* 20 (6): 1265-1285. https://doi.org/10.1016/j.ymssp.2005.11.015.

Chassignole, Bertrand, D Villard, and M Dubuget. 2000. "Characterization of austenitic stainless steel welds for ultrasonic NDT." AIP Conference Proceedings. American Institute of Physics. Vol. 509. No. 1 10.1063/1.1307835.

Chatillion, S, C Poidevin, N Gengembre, and A Lhémery. 2003. "Simplified modeling of backscattered noise and attenuation phenomena for quantitative performance demonstration of UT methods." AIP Conference Proceedings. IOP INSTITUTE OF PHYSICS PUBLISHING LTD. no. A 93-100.

Chen, Y., Z. Luo, Q. Zhou, L. Zou, and L. Lin. 2015. "Modeling of ultrasonic propagation in heavy-walled centrifugally cast austenitic stainless steel based on EBSD analysis." *Ultrasonics* 59: 31-39. https://doi.org/10.1016/j.ultras.2015.01.009. https://www.ncbi.nlm.nih.gov/pubmed/25670411.

Crawford, S L, A D Cinson, M T Anderson, A Diaz, and S E Cumblidge. 2009. PNNL-18334. *Ultrasonic Flaw Detection of Intergranular Stress Corrosion Cracks as Observed In Austenitic Stainless Steel Piping Welds.*

Crawford, S L, M S Prowant, A D Cinson, M R Larche, A A Diaz, and M T Anderson. 2014. PNNL-23393. *Phased Array Ultrasonic Sound Field Mapping in Cast Austenitic Stainless Steel.* Pacific Northwest National Laboratory. Richland, Washington: ML14155A165.

Diaz, A, A D Cinson, S L Crawford, and M T Anderson. 2009a. PNNL-19018. *An Evaluation of Ultrasonic Phased Array Testing for Reactor Piping System Components Containing Dissimilar Metal Welds.* Pacific Northwest National Laboratory. Richland, WA.

Diaz, A, A D Cinson, S L Crawford, S E Cumblidge, K M Denslow, M Morra, M Prowant, and M T Anderson. 2008. PNNL-17698. *Assessment of Ultrasonic Phased Array Testing for Cast Austenitic Stainless Steel Pressurizer Surge Line Piping Welds and Thick Section Primary System Cast Piping Welds.* Pacific Northwest National Laboratory. Richland, WA: Published in ADAMS July 2011.

Diaz, A, A D Cinson, S L Crawford, R Mathews, T L Moran, and M T Anderson. 2009b. PNNL-18596. *Assessment of Ultrasonic Phased Array Inspection Method for Welds in Cast Austenitic Stainless Steel Pressurizer Surge Line Piping* Pacific Northwest National Laboratory. Richland, WA.

Dib, G, S L Crawford, R. E. Jacob, M Prowant, and A Diaz. 2017. PNNL-26336. *Validation of Ultrasonic Nondestructive Examination (NDE) Computational Models - Phase 1.* Pacific Northwest National Laboratory. Richland, WA.

Dib, G, M R Larche, M Prowant, R. E. Jacob, P Ramuhalli, and A Diaz. 2018. PNNL-28362. *Ultrasound Modeling and Simulation: Status Update.* Pacific Northwest National Laboratory. Richland, WA.

Dorval, V, L Ducousso-Ganjehi, S Chatillion, and F Jensen. 2013. "Modeling ultrasonic structural noise based on the microstructural properties of metals in the context of non destructive evaluation." 13th International Symposium on Nondestructive Characterization of Materials (NDCM-XIII).

EPRI. 2019. *Evaluation of Ultrasonic Simulation Software: Summary of Recent Results—Ultrasonic Weld Simulations.* 3002015980. Electric Power Research Institute. Palo Alto, CA.

Feuilly, N, O Dupond, B Chassignole, J Moysan, and G Comeloup. 2009. "Relation between ultrasonic backscattering and microstructure for polycrystalline materials." AIP Conference Proceedings. American Institute of Physics. Vol. 1096. No.1 10.1063/1.3114093.

Ganjehi, L, V Dorval, and F Jensen. 2012. "Modelling of the ultrasonic propagation in polycrystalline materials." *Acoustics 2012*.

Gardahaut, A, K Jezzine, and D Cassereau. 2012. "Modelling tools for ultrasonic inspection of bimetallic welds." *Acoustics 2012*.

Gardahaut, A, H Lourme, F Jenson, S Lin, and M Nagai. 2014. "Ultrasonic Wave Propagation in Dissimilar Metal Welds - Application of a Ray-Based Model and Comparison with Experimental Results." Proceedings of the 11th European Conference on Non-destructive Testing, Prague, Czech Republic.

Greenwood, M.S. 1998. NUREG/CR-6589; PNNL-11751. *The Effects of Surface Condition on an Ultrasonic Inspection: Engineering Studies Using Validated Computer Model.* Pacific Northwest National Laboratory. Richland, WA.

Gueudré, Cécile, Jean Mailhé, Marie-Aude Ploix, Gilles Corneloup, and Bertrand Chassignole. 2019. "Influence of the uncertainty of elastic constants on the modelling of ultrasound propagation through multi-pass austenitic welds. Impact on non-destructive testing." *International Journal of Pressure Vessels and Piping* 171: 125-136. https://doi.org/10.1016/j.ijpvp.2019.02.011.

Harrison, J M, R. E. Jacob, M Prowant, A Holmes, C Hutchinson, and A Diaz. 2020. *Evaluating Flaw Detectability Under Limited-Coverage Conditions.* PNNL-30238.

Heasler, PG, TT Taylor, and S R Doctor. 1993. *Statistically Based Reevaluation of PISC-II Round Robin Test Data.* NUREG/CR-5401, PNL-8577. U.S. Nuclear Regulatory Commission. Washington, D.C.

Holmer, G, W Daniels, and T Zettervall. 2017. 2017:29. *Evaluation of the simulation software CIVA for qualification purpose.* Swedish Radiation Safety Authority. Stockholm, Sweden.

Jacob, R. E., S L Crawford, T L Moran, M R Larche, M Prowant, A A Diaz, and C A Nove. 2019. NUREG/CR-7263; PNNL-28840. *NDE Reliability Issues for the Examination of CASS Components.* Washington, D.C.: U.S. Nuclear Regulatory Commission. ADAMS Accession No. ML19255J814.

Jacob, R. E., M Prowant, C Hutchinson, N Deshmukh, and A Diaz. 2020. PNNL-29899. *Modeling and Simulation of Austenitic Welds and Coarse-grained Specimens.* ML20122A252.

Jenson, F, L Ganjehi, C Poidevin, G Cattiaux, and T Sollier. 2010. "Modelling of Ultrasonic Propagation in Cast Stainless Steels with Coarse Grained Structures." 7th International Conference on NDE in Relation to Structural Integrity for Nuclear and Pressurized

Components, Yokohama, Japan, 12-15 May, 2009. https://www.ndt.net/article/jrc-nde2009/papers/42.pdf.
Kim, Hun-Hee, Hak-Joon Kim, Sung-Jin Song, Kyung-Cho Kim, and Yong-Buem Kim. 2016. "Simulation Based Investigation of Focusing Phased Array Ultrasound in Dissimilar Metal Welds." *Nuclear Engineering and Technology* 48 (1): 228-235. https://doi.org/10.1016/j.net.2015.10.011.
Kim, Y., S. Cho, and I. K. Park. 2021. "Analysis of Flaw Detection Sensitivity of Phased Array Ultrasonics in Austenitic Steel Welds According to Inspection Conditions." *Sensors (Basel)* 21 (1): 242. https://doi.org/10.3390/s21010242. https://www.ncbi.nlm.nih.gov/pubmed/33401492.
Kumar, S, M Menaka, and B Venkatraman. 2021. "Performance comparison of phased array transducers for inspection of dissimilar welds on nuclear reactor components." *Annals of Nuclear Energy* 162: 108482.
Mahaut, S, S Chatillion, N Leymarie, F Jensen, and P Calmon. 2007. "Simulation tools for predicting non destructive testing of heterogeneous and anisotropic structures." The International Congress of Ultrasonics.
Mahaut, S., S. Chatillon, M. Darmon, N. Leymarie, R. Raillon, and P. Calmon. 2010. "An overview of ultrasonic beam propagation and flaw scattering models in the CIVA software." AIP Conference Proceedings. American Institute of Physics. 1211 (1): 2133–2140.
Maitland, T, and S Sitzman. 2007. "Electron Backscatter Diffraction (EBSD) Technique and Materials Characterization Examples." In *Scanning Microscopy for Nanotechnology: Techniques and Applications*, edited by Weilie Zhou and Zhong Lin Wang, 41-75. New York: Spengler-Verlag.
Margetan, F. J., R. B. Thompson, and I. Yalda-Mooshabad. 1994. "Backscattered microstructural noise in ultrasonic toneburst inspections." *Journal of Nondestructive Evaluation* 13 (3): 111-136. https://doi.org/10.1007/bf00728250.
Nageswaran, C., C. Carpentier, and Y. Y. Tse. 2009. "Microstructural quantification, modelling and array ultrasonics to improve the inspection of austenitic welds." *Insight - Non-Destructive Testing and Condition Monitoring* 51 (12): 660-666. https://doi.org/10.1784/insi.2009.51.12.660.
Ono, Kanji. 2020. "A Comprehensive Report on Ultrasonic Attenuation of Engineering Materials, Including Metals, Ceramics, Polymers, Fiber-Reinforced Composites, Wood, and Rocks." *Applied Sciences* 10 (7). https://doi.org/10.3390/app10072230.
Ramuhalli, P, LJ Bond, R Mathews, KC Roberts, RV Harris, A Diaz, M T Anderson, and CO Ruud. 2010. PNNL-19325. *In-situ Characterization of Cast Austenitic Stainless Steel Microstructure: An Interim Study.* Pacific Northwest National Laboratory. Richland, WA.
Ribay, G, S Mahaut, G Cattiaux, and T Sollier. 2012. "Assessment of Reliability of Phased Array NDT of Coarse Grain Component Based on Simulation." 7th European-American Workshop on Reliability of NDE.
Rodriguez, E, F Fraudita, and L Amorer. 2003. "Effect Of Surface Roughness In Ultrasonic Testing(Pulse-Echo By Direct Contact) In AISI/SAE 4340 Steel Samples." *Journal of Nondestructive Testing* 8.12: 1-8.
Ryzy, M., T. Grabec, P. Sedlak, and I. A. Veres. 2018. "Influence of grain morphology on ultrasonic wave attenuation in polycrystalline media with statistically equiaxed grains." *J Acoust Soc Am* 143 (1): 219. https://doi.org/10.1121/1.5020785. https://www.ncbi.nlm.nih.gov/pubmed/29390780.
Sakamoto, K, F Takashi, K Ichiro, K Toshinori, and M Tsuyoshi. 2012. "Study on the Ultrasound Propagation in Cast Austenitic Stainless Steel." *E-J Adv. Maint.* 4 (1).

Schuster, GJ, FA Simonen, and S R Doctor. 2008. NUREG/CR-6945, PNNL-16726. *Fabrication Flaw Density and Distribution In Repairs to Reactor Pressure Vessel and Piping Welds.* U.S. Nuclear Regulatory Commission. Washington, D.C.

Szávai, Szabolcs, Zoltán Bézi, Judit Dudra, and István Méhész. 2016. "Modelling of Phased Array Ultrasonic Inspection of a Steam Generator Dissimilar Metal Weld." *Procedia Structural Integrity* 2: 1015-1022. https://doi.org/10.1016/j.prostr.2016.06.130.

Van Pamel, A., C. R. Brett, P. Huthwaite, and M. J. Lowe. 2015. "Finite element modelling of elastic wave scattering within a polycrystalline material in two and three dimensions." *J Acoust Soc Am* 138 (4): 2326-2336. https://doi.org/10.1121/1.4931445. https://www.ncbi.nlm.nih.gov/pubmed/26520313.

Wan, T., T. Naoe, T. Wakui, M. Futakawa, H. Obayashi, and T. Sasa. 2017. "Effects of Grain Size on Ultrasonic Attenuation in Type 316L Stainless Steel." *Materials (Basel)* 10 (7). https://doi.org/10.3390/ma10070753. https://www.ncbi.nlm.nih.gov/pubmed/28773115.

Appendix A – CIVA Tips

A.1 Reconstruction of B-scans

During field UT examinations, echoes are measured in time and projected to a position in the specimen-based on the refracted angle, the speed of sound in the specimen, and a straight-line sound path. Bending of the beam is likely to occur through non-isotropic materials, but the sound path is always assumed to be straight, because the degree of bending and the actual sound path are unknown. As a consequence, the received echoes may be projected to the wrong location in the specimen. Although this is typically not a dramatic effect, a flaw may appear to be in one location when it is actually in a slightly different location. CIVA, however, appears to reconstruct the data based on simulated beam paths, so any bending of the beam is accounted for. This causes simulation results to show the flaw at the *actual* location as opposed to the *apparent* location that one would see in a field exam. Figure A.1 illustrates the difference. A specimen was modeled with two regions and a single interface; a small, embedded flaw was added. The regions were given different material properties so that the sound path would bend as the beam passed across the interface. The solid green line indicates the actual sound path from the probe to the flaw; this path was used by CIVA to calculate the echo response at the actual flaw location. The dashed line shows the straight-line path along which an examiner (or software such as UltraVision) would project the flaw location. The flaw would appear in a field exam to be several millimeters away from the actual flaw location.

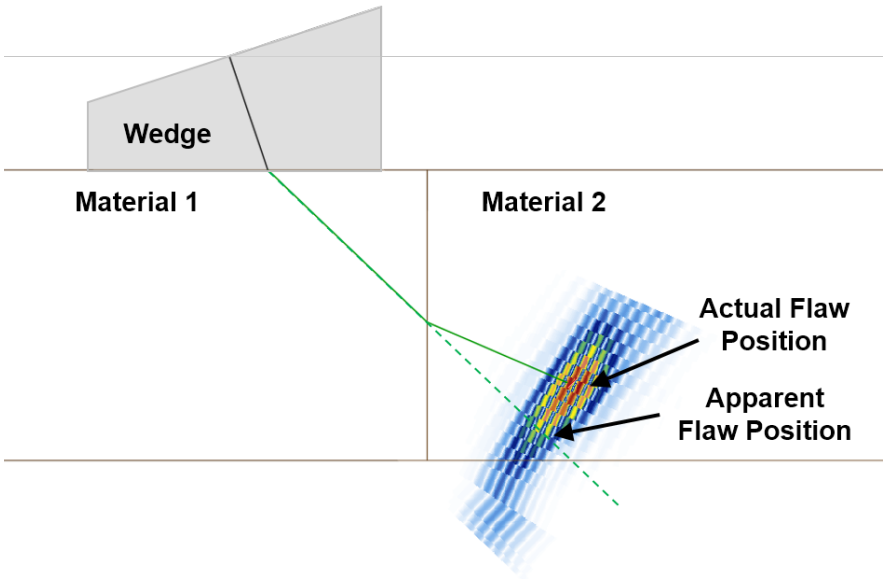

Figure A.1. CIVA flaw response simulation with the CIVA-calculated beam path (solid green line) and the straight-line beam path (dashed line). In an empirical exam, the flaw position would be projected along the dashed line.

A more complex example of the beam path issue is shown in Figure A.2. The top panel shows the ray tracing diagram of a phased-array probe over an isotropic material with the rays incident on a flaw tilted at 19°. The bottom panel shows the same PA probe over an austenitic weld model. In an actual examination, such as with UltraVision, all received signals in the bottom panel would be mapped to the lower portion of the flaw, based on the straight-line ray paths in the top panel and the sound time-of-flight. However, CIVA maps the received signals based on

the actual beam paths. If the user wants to compare flaw response simulations to empirical data (for simulations with non-isotropic materials), the user will have to reconstruct the B-scans outside of CIVA and assume straight-line beam paths. PNNL did this by exporting the “BScan (scanning/time)” data (not the “True BScan (scanning/depth)” data) and performing reconstructions in Python.

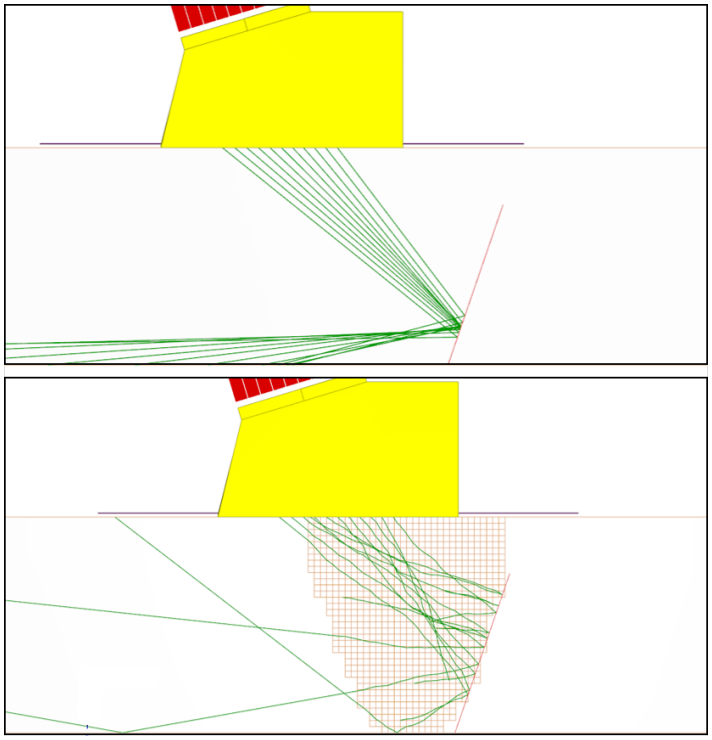

Figure A.2. Ray paths from a PA probe through isotropic material (top) and the weld model (bottom).

The Python B-scan reconstruction algorithm projects the simulated A-scans along the refraction angle. Some image interpolation and anti-aliasing is required to remove artifacts. An example of the reconstruction results is presented in Figure A.3. Here, a flaw response simulation was run through the EBSD weld model with a flat flaw of about 80% through-wall depth and 19° tilt. The top panel shows a screenshot of the CIVA-reconstructed B-scan. Discontinuities and gaps are visible in the reconstruction. The bottom panel is the “True BScan (scanning/depth)” data reconstructed in Python assuming straight-line beam paths. The Python reconstruction is smooth and contiguous as would be seen in a field exam.

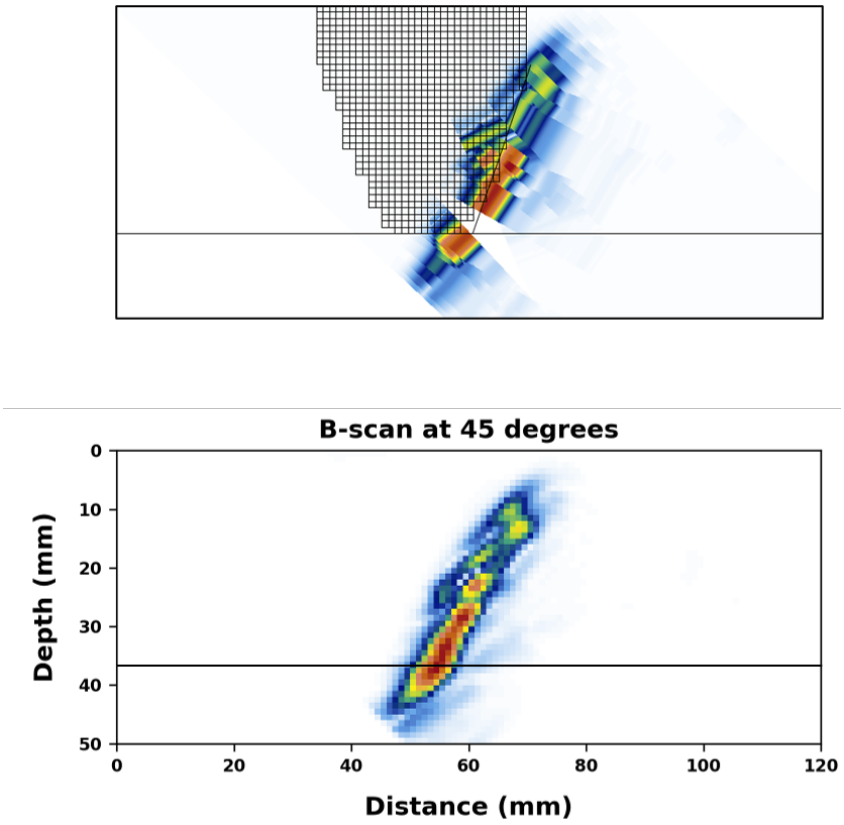

Figure A.3. Top: Screenshot of the CIVA-reconstructed B-scan. Bottom: B-Scan reconstructed using Python and straight-line beam paths.

As an aside, it is interesting to note the degree of distortion that the weld model causes in the flaw response; Figure A.4 shows the response from the same flaw but with isotropic material as reconstructed in CIVA. The distinct corner and tip responses are absent when the weld model is used.

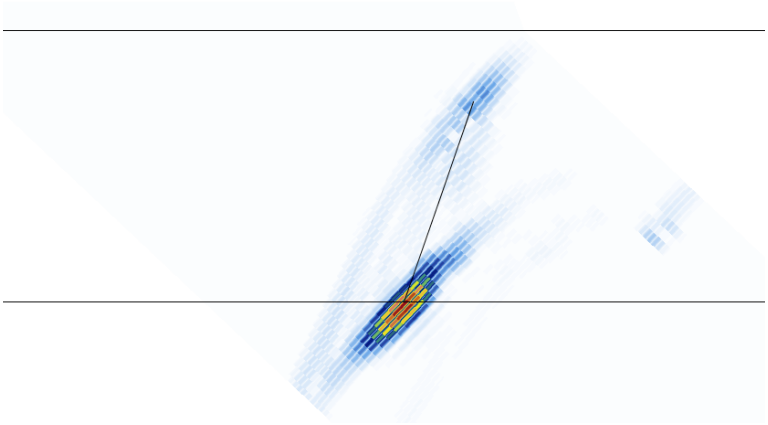

Figure A.4. Flaw response through isotropic material.

A.2 Correcting for Wedge Delay

When reconstructing B-scans outside of CIVA, it is important to understand that the starting timepoint of the simulation does not necessarily occur at the surface of the specimen. Instead, the initial timepoint of the simulation often occurs somewhere within the wedge; this can be thought of as a "wedge delay," or a calculation of some portion of the sound path through the wedge. PNNL found that the duration of the wedge delay, or the number of data points calculated in the wedge, can vary from simulation to simulation, even with the same wedge and specimen parameters. Inconsistent delays can cause problems with the offline reconstruction. However, the delay can be set manually with a gate before execution of the simulation, but without *a priori* knowledge of where the simulation will begin, setting the gate is initially arbitrary. The delay duration is not recorded in the output files, so it must be determined manually. Therefore, when exporting the data to Python for reconstruction, there is no way to tell from the information in the data file where the outer surface of the specimen is in relation to the start of the data file. However, after the simulation is complete, CIVA gates can be used in postprocessing to eliminate the wedge delay and force the first data point to be at the surface of the specimen. The postprocessed data can then be exported for reconstruction.

A.3 Region sequencing

CIVA 2020 had a significant update to the method of region sequencing. The previous method, described in (Jacob et al. 2020), was slow to load geometries in CIVA and cumbersome to create geometry .xml files outside of CIVA. The new method is a major improvement that is more intuitive, much faster, and easier to implement in MATLAB. The new method employs "a lexicographic order on the bounding boxes of the faces."[1] The bounding box is the smallest possible square or rectangle drawn in the Cartesian frame that fully encompasses the shape. An example of a bounding box around an arbitrary shape is shown in Figure A.5. The minimum and maximum X and Y coordinates of the bounding box vertices are labeled. After all regions in the geometry are assigned bounding boxes, the regions are then sequenced using the vertices of the bounding boxes based on the following hierarchy:

1. Minimum X value
2. Maximum X value
3. Minimum Y value
4. Maximum Y value.

[1] E-mail received from EXTENDE, January 2020

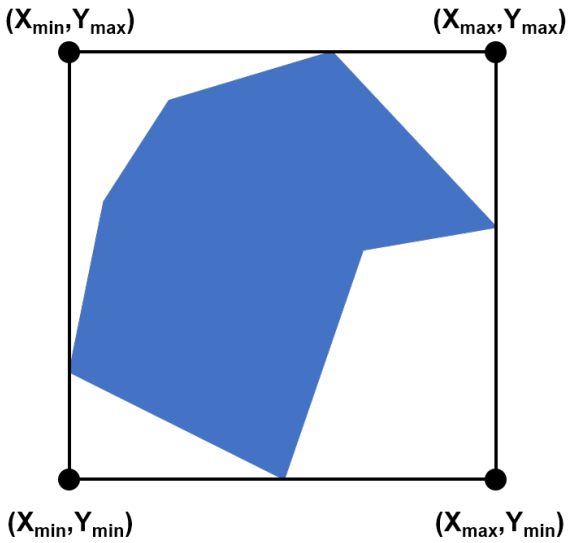

Figure A.5. Example of a bounding box around an arbitrary shape.

If two regions share identical bounding boxes, then the algorithm uses the iso-barycenter[1]:

1. Minimum X value of iso-barycenter
2. Minimum Y value of iso-barycenter.

An example of when the iso-barycenters may have to be invoked is shown in Figure A.6. In this case, two right triangles that share a hypotenuse (i.e., a rectangle with a diagonal line through it) have the same bounding box but different iso-barycenters.

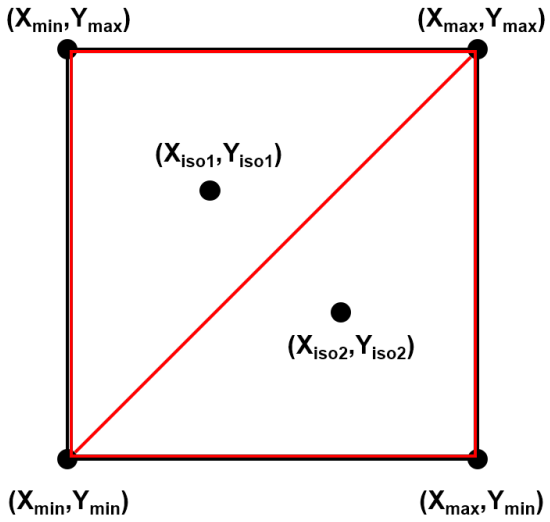

Figure A.6. Example of when the iso-barycenter is used to distinguish regions with identical bounding boxes.

[1] The iso-barycenter is the centroid, or "center of gravity," of a shape based on the endpoints of the line segments, with each endpoint given equal weight.

A.4 Euler Angles

CIVA 2020 has new options for entering Euler angles. The user can now select “CIVA,” “ROE,” or “BUNGE” conventions. However, no matter what convention is entered, CIVA immediately calculates values in the “CIVA” convention and saves only those values. This is typically a non-issue, since the calculation is instantaneous and behind-the-scenes. However, when the user is generating a custom geometry with an .xml file outside of CIVA, then the angles in that .xml file must be of the “CIVA” convention. Bunge angles still need to be converted to the specimen reference frame when generating a custom .xml file prior to opening the file in CIVA.

When working with Euler angles, it is critical to understand the frame of reference. The default CIVA reference frame is the reference frame of the specimen, and the CIVA Euler angles are calculated with respect to that frame. Bunge angles, on the other hand, are calculated with respect to the crystal orientation. Prior to editing Euler angles in custom .xml files, users must account for the different reference frames and make any necessary conversions. Furthermore, OnScale Solve uses a different reference frame, and the raw EBSD data that PNNL used to generate the weld model were in yet a different reference frame. Any operations that require interpreting Euler angles from one frame to another must use the correct Euler angle transformation. For example, verifying a model that was run in one software package by running it in another software package will require careful scrutiny of the Euler angle conversions to assure that the same Euler angles are actually being used in both cases. Appendix B is a technical description of relevant Euler angle conversions with a Python implementation.

A.5 Custom Cracks

It is possible to create custom flaws in CIVA using the CIVA CAD tool. PNNL has explored the “Branched” and “Multifaceted defect” options. A multifaceted flaw was created for the surface noise tests in Section 5.2.3. Users should understand how these flaw types are handled differently by CIVA 2020. A multifaceted flaw cannot be changed into a branched flaw simply by adding a branching line segment, since the flaw types are defined differently in CIVA. Branched flaws are defined by line segment lengths and branch angles, whereas multifaceted flaws are defined by segment endpoint coordinates. Multifaceted flaws are easily scaled in the CAD editor, but branched flaws are not. Rather, branched flaws must be scaled segment by segment. Some of the scaling issues came up for PNNL when they were testing flaw .xml files with computer-generated branched cracks that were defined by endpoints.

Figure A.7 shows an example of a branched flaw definition. Each segment is defined by a length, an angle, and a parent segment. The flaw can be drawn in the CAD editor, as shown in the bottom of the figure. Note that every branched or multifaceted flaw must be accompanied by an extrusion segment, shown by the magenta line in the figure. The extrusion segment is used to define how the flaw is extruded in the third dimension. The flaws can also be rotated arbitrarily about the x, y, or z axes and placed anywhere within the specimen. For example, the flaw shown was drawn at an angle, but it could be placed vertically in a specimen model if desired.

Geometry | Positioning

Geometry: Branched

Comment: | Edit

2D profile

Name	Parent	Length	Angle	Absolute angle
Element 1	0	1.201	2.386	2.386
Element 2	1	0.472	55.609	57.995
Element 3	2	0.2	-57.995	0
Element 4	3	0.559	10.305	10.305
Element 5	4	1.079	66.303	76.608
Element 6	5	0.894	-50.042	26.565
Element 7	6	1.167	16.699	43.264
Element 8	7	1.408	-26.76	16.504
Element 9	8	1.484	40.876	57.381
Element 10	8	1.408	-10.389	6.116
Element 11	10	1.681	24.264	30.379
Element 12	11	1.681	29.242	59.621
Element 13	11	0.269	37.819	68.199
Element 14	11	1.209	29.876	60.255
Element 15	14	1.544	0.69	60.945
Element 16	15	0.652	-3.417	57.529
Element 17	15	0.901	-41.505	19.44
Element 18	17	1.504	2.008	21.448
Element 19	18	1.518	51.311	72.759
Element 20	19	1.329	-2.557	70.201
Element 21	20	1.423	-51.766	18.435
Element 22	21	0.962	9.462	27.897

2D extrusion

Name	Parent	Length	Angle	Absolute angle
Element 1	0	50	0	0

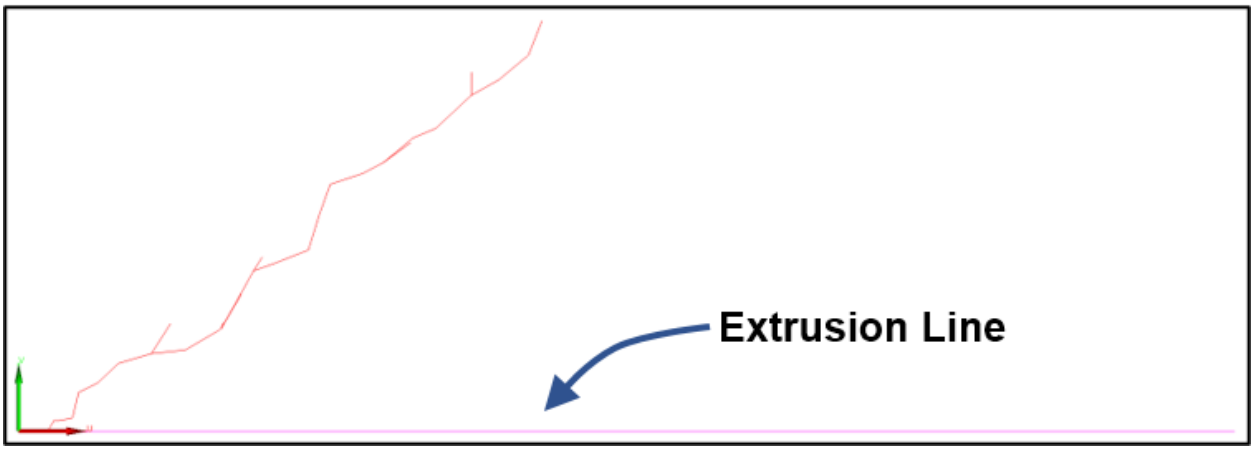

Figure A.7. Example of the definition of a branched flaw.

A.6 Low frequency tests in CIVA

PNNL briefly investigated the relationship between flaw size and probe frequency in CIVA simulations. As explained in the CIVA user's manual, CIVA uses a high-frequency approximation for crack or notch simulations; the approximation is valid when the defect size is approximately greater than the wavelength. Results confirmed that flaw response simulation results are not reliable for cases where the flaw size is smaller than the wavelength. This limitation is important and should be remembered for all simulated scenarios. For example, simulations of flaw responses in CASS materials using low-frequency probes (e.g., 500 kHz) are not recommended in CIVA.

Appendix B – Euler Angle Coordinate Transformations

This appendix is a reproduction of an internal PNNL report authored by M. Hughes. It describes in detail the mathematical foundation of Euler angle transformations. It also contains descriptions of an implementation in Python.

Relationships Between Different Euler Angle Systems Defined in Different Coordinate Systems

Pacific Northwest National Laboratory

Contents

Abstract

Given Euler angles, α_i, β_i, and γ_i, in an initial coordinate system, i, (e.g., Bunge, CIVA, OnScale), f, we with wish to find the geometrically equivalent the Euler angles, α_f, β_f, and γ_f in the final coordinate frame.

There are at least twelve different conventions for defining the Euler angles. The conventions used for α_i, β_i, γ_i may be different from those used for α_i, β_i, and γ_i. This must be accounted for also[1]. Consequently, we must account for two different transformations when computing the final Euler angles from the initial:

A transformation T from the initial to final coordinate system.

A transition from the initial Euler angle convention to the final convention.

This is accomplished in **three** steps.

First construct the Euler transformation matrix, E_i using the initial Euler angle conventions. This is just a rotation matrix. The content of representation of rotations by Euler angles is that any rotation in $\mathbb{R}^3$ can be factored into the product of three rotations each of which is defined in terms of one of the Euler angles.

Second we use use the (invertible) matrices for transformations T (denoted now bt $T_{i,f}$ to be more specific) $T_{i,f} : \mathbb{R}^3 \to \mathbb{R}^3$, between initial coordinate system (*e.g.*, Bunge, CIVA, OnScale), and the final coordinate system to transform to transform E_i from the initial coordinate system, i to the final coordinate system f with the similarity transformation:

$$E_f = T_{i,f} E_i T_{i,f}^{-1}. \tag{1}$$

This transformation is derived in Fig. (1).

Third, we factor E_f into the product of three rotations, α_f, β_f, and γ_f corresponding the final Euler angle convention. This factorization is not possible if the new configuration exhibits "gimbal lock" as described below.

Execution of the third step to find α_f, β_f, and γ_f requires use of inverse trigonometric functions. As a result the "factoring" step is ill-posed for two reasons:

The functions $\cos^{-1}(\theta)$ and $\sin^{-1}(\theta)$ lose accuracy as $\theta \to \pm 1$, and

These "functions" $\cos^{-1}(\theta)$ and $\sin^{-1}(\theta)$ are multi-valued.

The first problem is solved by replacing expressions based on $\cos^{-1}(\theta)$ and $\sin^{-1}(\theta)$ with $\tan^{-1}(\theta)$. The second is solved by recognizing the values of α_f, β_f, and γ_f provided by these numerically stable expressions actually only equal to within some unknown multiple $\pm\pi$ of the true answers. The ultimate source of this ambiguity is the multi-valued nature of the inverse trigonometric functions combined with the potential inversion of axes in different xyz coordinate systems. Only two different values for α_f, β_f, and γ_f need be considered, depending on the lower bound of the range of allowed valued (this choice is a matter of convention and different software packages take different choices). Numerical studies indicate that of the 8 different combinations only one is consistent with the Eq. (1) so the ambiguity may be resolved.

All three steps have been implemented in a python module written to use the same variable names that appear in this document. Additionally, the code is commented with references the equations appearing below.

Program desriptions and test results are included at the conclusion of this report.

$$\vec{x}_1 = A\vec{x}, \quad \left.\begin{array}{l} \vec{y}_1 = T\vec{x}_1 \implies T^{-1}\vec{y}_1 = \vec{x}_1 \\ \vec{y} = T\vec{x} \implies T^{-1}\vec{y} = \vec{x} \end{array}\right\} \implies T^{-1}\vec{y}_1 = AT^{-1}\vec{y} \implies \vec{y}_1 = TAT^{-1}\vec{y}$$

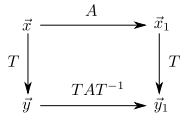

Figure 1: Transformation of a transformation under a invertible linear map $T : \mathbb{R}^3 \to \mathbb{R}^3$.

[1] There are actually even more conventions to account for see,[1] pg 477.

1 Python Implemention

1.1 Function Definitions

The following functions are implemented in the Python module Euler_to_Euler.py:

MAIN CALLER ROUTINE:

Euler_convert(*input_filename, output_filename, conversion_mode, verbose*=False)
Parameters: *input_filename* contains lines containing: $(x, y, \alpha_i, \beta_i, \gamma_i)$, while *output_filename* contain lines $(x, y, \alpha_f, \beta_f, \gamma_f)$, *conversion_mode* specifies the system conventions (Bunge, CIVA, OnScale) used for α_i, β_i, γ_i, and the system conventions desired for α_f, β_f, γ_f. It must be one of six strings: *'Bunge_to_OnScale', 'OnScale_to_Bunge', 'Bunge_to_CIVA', 'CIVA_to_Bunge', 'CIVA_to_OnScale', 'OnScale_to_CIVA'*. If the *verbose* parameter is set to True, the parameter is passed through to the conversion functions described below to trigger generation of debugging output.
Outputs: The outfile specified by output_filename containing lines lines $(x, y, \alpha_f, \beta_f, \gamma_f)$.

DOUBLE CHECKING ROUTINES:

Check_Bunge_EA_against_OnScale_EA (*alpha_B, beta_B, gamma_B,alpha_O, beta_O, gamma_O*)
Parameters: *alpha_B, beta_B, gamma_B* are Euler angles specified in the **Bunge** system, while *alpha_O, beta_O, gamma_O* are (supposed to be) the geometrically equivalent Euler angles specified in the **OnScale** system. This function uses *alpha_B, beta_B, gamma_B* to compute the rotation matrix, *E_B* defined in Eq. (23). Subsequently, Eq. (26) is used to compute a rotation matrix: E_O_fromXformed_E_B: the transformation of *E_B* from the **Bunge** system to the **OnScale** system. This matrix is then compared with the rotation matrix E_O, computed from *alpha_O, beta_O, gamma_O* via Eq. (24). They should be equal.
Outputs: The matrices *E_B,E_O,E_O_fromXformed_E_B,status*, where *status* is the boolean **True** if *E_O=E_O_fromXformed_E_B* and is the boolean **False** otherwise.

Check_OnScale_EA_against_Bunge_EA (*alpha_O, beta_O, gamma_O,alpha_B, beta_B, gamma_B*)
Parameters: *alpha_O, beta_O, gamma_O* are Euler angles specified in the **OnScale** system, while *alpha_B, beta_B, gamma_B* are (supposed to be) the geometrically equivalent Euler angles specified in the **Bunuge** system. This function uses *alpha_O, beta_O, gamma_O* to compute the rotation matrix, *E_O* defined in Eq. (24). Subsequently, Eq. (27) is used to compute a rotation matrix: E_B_fromXformed_E_O: the transformation of *E_O* from the **OnScacle** system to the **Bunge** system. This matrix is then compared with the rotation matrix E_B, computed from *alpha_B, beta_B, gamma_B* via Eq. (23). They should be equal.
Outputs: The matrices *E_O,E_B,E_B_fromXformed_E_O,status*, where *status* is the boolean **True** if *E_B=E_B_fromXformed_E_O* and is the boolean **False** otherwise.

Check_Bunge_EA_against_CIVA_EA (*alpha_B, beta_B, gamma_B,alpha_C, beta_C, gamma_C*)
Parameters: *alpha_B, beta_B, gamma_B* are Euler angles specified in the **Bunge** system, while *alpha_C, beta_C, gamma_C* are (supposed to be) the geometrically equivalent Euler angles specified in the **CIVA** system. This function uses *alpha_B, beta_B, gamma_B* to compute the rotation matrix, *E_B* defined in Eq. (23). Subsequently, Eq. (28) is used to compute a rotation matrix: E_C_fromXformed_E_B: the transformation of *E_B* from the **Bunge** system to the **CIVA** system. This matrix is then compared with the rotation matrix E_C, computed from *alpha_C, beta_C, gamma_C* via Eq. (25). They should be equal.
Outputs: The matrices *E_B,E_C,E_C_fromXformed_E_B,status*, where *status* is the boolean **True** if *E_C=E_C_fromXformed_E_B* and is the boolean **False** otherwise.

Check_CIVA_EA_against_Bunge_EA (*alpha_C, beta_C, gamma_C,alpha_B, beta_B, gamma_B*)
Parameters: *alpha_C, beta_C, gamma_C* are Euler angles specified in the **CIVA** system, while *alpha_B, beta_B, gamma_B* are (supposed to be) the geometrically equivalent Euler angles specified in the **Bunge** system. This function uses *alpha_C, beta_C, gamma_C* to compute the rotation matrix, *E_C* defined in Eq. (25). Subsequently, Eq. (29) is used to compute a rotation matrix: E_B_fromXformed_E_C: the transformation of *E_C* from the **CIVA** system to the **Bunge** system. This matrix is then compared with the rotation matrix E_B, computed from *alpha_B, beta_B, gamma_B* via Eq. (23). They should be equal.
Outputs: The matrices *E_C,E_B,E_B_fromXformed_E_C,status*, where *status* is the boolean **True** if *E_B=E_B_fromXformed_E_C* and is the boolean **False** otherwise.

Check_CIVA_EA_against_OnScale_EA (*alpha_C, beta_C, gamma_C,alpha_O, beta_O, gamma_O*)
Parameters: *alpha_C, beta_C, gamma_C* are Euler angles specified in the **CIVA** system, while *alpha_O, beta_O, gamma_O* are (supposed to be) the geometrically equivalent Euler angles specified in the **OnScale** system. This function uses *alpha_C, beta_C, gamma_C* to compute the rotation matrix, *E_C* defined in Eq. (25). Subsequently, Eq. (30) is used to compute a rotation matrix: E_O_fromXformed_E_C: the transformation of *E_C* from the **CIVA** system to the **OnScale** system. This matrix is then compared with the rotation matrix E_O, computed from *alpha_O, beta_O, gamma_O* via Eq. (24). They should be equal.
Outputs: The matrices *E_C,E_O,E_O_fromXformed_E_C,status*, where *status* is the boolean **True** if *E_O=E_O_fromXformed_E_C* and is the boolean **False** otherwise.

Check_OnScale_EA_against_CIVA_EA (*alpha_O, beta_O, gamma_O,alpha_C, beta_C, gamma_C*)
Parameters: *alpha_O, beta_O, gamma_O* are Euler angles specified in the **OnScale** system, while *alpha_C, beta_C, gamma_C* are (supposed to be) the geometrically equivalent Euler angles specified in the **CIVA** system. This function uses *alpha_O, beta_O, gamma_O* to compute the rotation matrix, *E_O* defined in Eq. (24). Subsequently, Eq. (31) is used to compute a rotation matrix: E_C_fromXformed_E_O: the transformation of *E_O* from the **OnScale** system to the **CIVA** system. This matrix is then compared with the rotation matrix E_C, computed from *alpha_C, beta_C, gamma_C* via Eq. (25). They should be equal.
Outputs: The matrices *E_O,E_C,E_C_fromXformed_E_O,status*, where *status* is the boolean **True** if *E_C=E_C_fromXformed_E_O* and is the boolean **False** otherwise.

EULER ANGLE CONVERSION ROUTINES:

Bunge_to_OnScale(*alpha_in_deg, beta_in_deg, gamma_in_deg, verbose*=False)
Based on Eqs. (36), (37), (38), and (34), and regularized by Eqs. (32) and (34).
Parameters: *alpha_in_deg, beta_in_deg, gamma_in_deg* are the Euler angles in the Bunge System
verbose, if set to True, **Bunge_to_OnScale** prints debugging output
Outputs: *alpha_out_deg, beta_out_deg, gamma_out_deg* are the geometrically equivalent Euler angles in the OnScale system.

OnScale_to_Bunge(*alpha_in_deg, beta_in_deg, gamma_in_deg, verbose*=False)
Based on Eqs. (46), (47), (48), and (44), and regularized by Eqs. (43) and (44).
Parameters: *alpha_in_deg, beta_in_deg, gamma_in_deg* are the Euler angles in the OnScale System
verbose, if set to True, **OnScale_to_Bunge** prints debugging output
Outputs: *alpha_out_deg, beta_out_deg, gamma_out_deg* are the geometrically equivalent Euler angles in the Bunge system.

Bunge_to_CIVA(*alpha_in_deg, beta_in_deg, gamma_in_deg, verbose*=False)
Based on Eqs. (56), (57), (58), and (54), and regularized by Eqs. (53) and (54).
Parameters: *alpha_in_deg, beta_in_deg, gamma_in_deg* are the Euler angles in the Bunge System
verbose, if set to True, **Bunge_to_CIVA** prints debugging output
Outputs: *alpha_out_deg, beta_out_deg, gamma_out_deg* are the geometrically equivalent Euler angles in the CIVA system.

CIVA_to_Bunge(*alpha_in_deg, beta_in_deg, gamma_in_deg, verbose*=False)
Based on Eqs. (66), (67), (68), and (64), and regularized by Eqs. (63) and (64).
Parameters: *alpha_in_deg, beta_in_deg, gamma_in_deg* are the Euler angles in the CIVA System
verbose, if set to True, **CIVA_to_Bunge** prints debugging output
Outputs: *alpha_out_deg, beta_out_deg, gamma_out_deg* are the geometrically equivalent Euler angles in the Bunge system.

CIVA_to_OnScale(*alpha_in_deg, beta_in_deg, gamma_in_deg, verbose*=False)
Based on Eqs. (76), (77), (78), and (74), and regularized by Eqs. (73) and (74).
Parameters: *alpha_in_deg, beta_in_deg, gamma_in_deg* are the Euler angles in the CIVA System
verbose, if set to True, **CIVA_to_OnScale** prints debugging output
Outputs: *alpha_out_deg, beta_out_deg, gamma_out_deg* are the geometrically equivalent Euler angles in the OnScale system.

OnScale_to_CIVA(*alpha_in_deg, beta_in_deg, gamma_in_deg, verbose*=False)
Based on Eqs. (86), (87), (88), and (84), and regularized by Eqs. (83) and (84).
Parameters: *alpha_in_deg, beta_in_deg, gamma_in_deg* are the Euler angles in the InScale System
verbose, if set to True, **OnScale_to_CIVA** prints debugging output

Outputs: *alpha_out_deg, beta_out_deg, gamma_out_deg* are the geometrically equivalent Euler angles in the CIVA system.

"Regularize" means setting the values of k, m, n) appearing in the equations cited above.

If "gimbal lock" does occur, *i.e.*, conversion is not possible, then these functions return a numpy array [0, 0, 0]. Otherwise the a numpy array containing the Euler angles in the new system is returned: [alpha_f, beta_f, gamma_f]. All angles are measured in degrees.

1.2 Software Testing

As mentioned above, conversion of Euler angles α_i, β_i, and γ_i from one system followed by a conversion of the resulting angles, α_f, β_f, and γ_f, back to the original system should produce Euler angles α'_i, β'_i, and γ'_i that produce the same rotation matrix as do the angles α_i, β_i, and γ_i.

We have tested all of the python functions written above over more than one million randomly chosen sets of Euler angles, to a precision of individual marix elements of 10^{-8} to detect unaccounted for edge cases. No deviations were observed in any of the tests

2 Introduction

Euler angles α, β and γ may be used to define rotations in three-dimensional space. For example, the coordinate axes may be rotated to successively produce the sequence of new axes: $xyz \to x'y'z' \to x'', y'', z''$ where

α represents a rotation around the z axis,

β represents a rotation around the x' axis,

γ represents a rotation around the z'' axis.

The range of values taken by α, β and γ in different software packages may vary, where $\alpha, \gamma \in [A, A + 2\pi]$, and $\beta \in [B, B + \pi]$, with A, B being positive or negative constants.

The *range* conventions applicable to the **Bunge**, **CIVA**, and **OnScale** systems are not clearly documented. We have written the python Euler angle transformation Python code to return angles between $[-\pi, \pi]$. Ambiguities introduced by the unavoidable use of inverse trigonometric are sesolved using internal mathematical consitency requirements specified by Eqs. (32), (43), (53), (63), (73), and (83) as described below.

There are many different conventions concerning the application of Euler angles. Consequently, any discussion must be proceeded by a complete specification of the choices that have been made. To indicate the range of possibilties we begin with a description of Euler angles taken from the Wikipedia page (URL: https://en.wikipedia.org/wiki/Euler_angles) since this seems to be one of the clearest available (however, see also[1] and §4.4.1 "The active interpretation and the active transformation" on p. 74 in "Robots and screw theory: applications of kinematics and statics to robotics"[2]):

> Euler angles can be defined by elemental geometry or by composition of rotations. The geometrical definition demonstrates that three composed elemental rotations (rotations about the axes of a coordinate system) are always sufficient to reach any target frame.
>
> The three elemental rotations may be extrinsic (rotations about the axes xyz of the original coordinate system, which is assumed to remain motionless), or intrinsic (rotations about the axes of the rotating coordinate system XYZ, solidary with the moving body, which changes its orientation after each elemental rotation).
>
> Euler angles are typically denoted as α, β, γ, or ψ, θ, ϕ. Different authors may use different sets of rotation axes to define Euler angles, or different names for the same angles. Therefore, any discussion employing Euler angles should always be preceded by their definition.
>
> Without considering the possibility of using two different conventions for the definition of the rotation axes (intrinsic or extrinsic), there exists twelve possible sequences of rotation axes, divided in two groups:
>
> > Proper Euler angles $(z-x-z, x-y-x, y-z-y, z-y-z, x-z-x, y-x-y)$
> >
> > Tait–Bryan angles $(x-y-z, y-z-x, z-x-y, x-z-y, z-y-x, y-x-z)$.
>
> Tait–Bryan angles are also called Cardan angles; nautical angles; heading, elevation, and bank; or yaw, pitch, and roll. Sometimes, both kinds of sequences are called "Euler angles". In that case, the sequences of the first group are called proper or classic Euler angles.

We see that **extrinsic** transformations are the result of transforming the vectors in the vector space $\mathbb{R}^3$, whereas **intrinsic** transformations are accomplished by transforming the basis vectors of $\mathbb{R}^3$. The terms "extrinsic" and "intrinsic" are thus examples of "active" and "passive" mathematical transformations respectively. The "extrinsic" *vs.* "intrinsic" terminilogy adds additional physical context to the mathematical terms and we will use it in the naming of variables and functions defined to compute the Euler angle transformations. However, to write these functions we must actually understand the mathematical terms and we next give their explicit statement.

A linear transformation, $A_a : \mathbb{R}^2 \to \mathbb{R}^3$ is called **active** (hence the subscript "a") if it entails the calculation of $(x'_1, x'_2, x'_3) = \vec{x}'$ such that

$$\vec{x}' = A_a \vec{x}. \tag{2}$$

A linear transformation, $A_p : \mathbb{R}^2 \to \mathbb{R}^3$, is called **passive** (hence the subscript "p") if it entails calculation of $(X_1', X_2', X_3') = \vec{X}'$ such that

$$\begin{aligned}\vec{X} =& X_1' A_p \vec{e}_1 + X_2' A_p \vec{e}_2 + X_3' A_p \vec{e}_3 && (3)\\ =& A_p \left(X_1' \vec{e}_1 + X_2' \vec{e}_2 + X_3' \vec{e}_3 \right) && (4)\\ =& A_p \vec{X}', && (5)\end{aligned}$$

or

$$\vec{X}' = A_p^{-1} \vec{X}. \tag{6}$$

Returning to software documentation, we will follow the conventions set forth in the Wikipedia Euler angle page: **extrinsic** transformations (="**active**") are executed in coordinate frames denoted using small letters x, y, z. On the other hand, **intrinsic** (="**passive**") transformations are executed in coordinate frames denoted using capitol letters X,Y,Z

Calculation of $(X_1', X_2', X_3') = \vec{X}'$ is equivalent to solving

$$A_i^{-1} \vec{X} = \vec{X}'. \tag{7}$$

From these definitions we see that extrinsic rotations are executed using active transformations and intrinsic rotations are executed using passive transformations.

Finally, Euler angles, α, β and γ, suffer from the well-known limitation of "gimbal locking", To quote the Wikipedia page one last time:

> The Euler angles α, β and γ are uniquely determined **except** for the singular case that the xy and the XY planes are identical, *i.e.* when the z-axis and the Z-axis have the same or opposite directions. Indeed, if the z-axis and the Z-axis are the same, $\beta = 0$ and only the $(\alpha + \gamma)$ is uniquely defined (not the individual values), and, similarly, if the z-axis and the Z-axis are opposite, $\beta = \pi$ and only $(\alpha - \gamma)$ is uniquely defined (not the individual values). These ambiguities are known as gimbal lock in applications.

3 Preliminary Transforms

3.1 Coordinate Transforms

We are interested in transformations of Euler angles between three different systems: **Bunge**, **CIVA**, and **OnScale**. The term "system" encompasses *both the choice of coordinate axes and the sequence of rotations employed*. These transformations are accomplished using, $T_{i,f} : \mathbb{R}^3 \to \mathbb{R}^3$ described above. For the three coordinate systems that we wish to transform "from" an "to", $T_{i,f}$ is one of:

$$\begin{aligned}T_{B,C} &: \text{Bunge} \longleftrightarrow \text{CIVA}, && (8)\\ T_{B,O} &: \text{Bunge} \longleftrightarrow \text{OnScale}, && (9)\\ T_{C,O} &: \text{CIVA} \longleftrightarrow \text{OnScale}. && (10)\end{aligned}$$

These have matrix representations,

$$T_{B,C} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \tag{11}$$

$$T_{B,O} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{12}$$

$$T_{C,O} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \tag{13}$$

Interestingly

$$T_{B,C} = T_{B,C}^{-1} = T_{C,B}, \tag{14}$$

$$T_{B,O} = T_{B,O}^{-1} = T_{O,B}, \tag{15}$$

$$T_{C,O} = T_{C,O}^{-1} = T_{O,C}, \tag{16}$$

so these $T_{B,C}, T_{B,O}, T_{C,O}$ are the only matrices we need to facililtate transformations between the coordinate systems that interest us.

3.2 Euler Matrices

Next we specify the rotation matices that embody the Euler angle conventions adopted by each of the three systems: **Bunge**, **CIVA**, and **OnScale**. These are constructed by composing the following matrices (which represent rotations about the single axis indicated by the subscript) for active transformations

$$R_{x,a}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}, \tag{17}$$

$$R_{y,a}(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \tag{18}$$

$$R_{z,a}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{19}$$

For passive transformations we use the inverses of these matrices as indicated by Eq.(6). It turns out that for each of the matrices above the inverse is the transpose. Here is the list

$$R_{x,p}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix}, \tag{20}$$

$$R_{y,p}(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \tag{21}$$

$$R_{z,p}(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{22}$$

Now we may list the relevant Euler rotation matrices for each of the three systems we must consider. They are

$$R_{z,p}(\alpha)R_{x,p}(\beta)R_{z,p}(\gamma) = E_{\text{Bunge}}(\alpha,\beta,\gamma) = \begin{bmatrix} -\cos(\beta)\sin(\alpha)\sin(\gamma) + \cos(\alpha)\cos(\gamma) & \cos(\beta)\cos(\gamma)\sin(\alpha) + \cos(\alpha)\sin(\gamma) & \sin(\alpha)\sin(\beta) \\ -\cos(\alpha)\cos(\beta)\sin(\gamma) - \cos(\gamma)\sin(\alpha) & \cos(\alpha)\cos(\beta)\cos(\gamma) - \sin(\alpha)\sin(\gamma) & \cos(\alpha)\sin(\beta) \\ \sin(\beta)\sin(\gamma) & -\cos(\gamma)\sin(\beta) & \cos(\beta) \end{bmatrix}, \tag{23}$$

$$R_{x,a}(\alpha)R_{y,a}(\beta)R_{z,a}(\gamma) = E_{\text{OnScale}}(\alpha,\beta,\gamma) = \begin{bmatrix} \cos(\beta)\cos(\gamma) & -\cos(\beta)\sin(\gamma) & \sin(\beta) \\ \cos(\gamma)\sin(\alpha)\sin(\beta) + \cos(\alpha)\sin(\gamma) & -\sin(\alpha)\sin(\beta)\sin(\gamma) + \cos(\alpha)\cos(\gamma) & -\cos(\beta)\sin(\alpha) \\ -\cos(\alpha)\cos(\gamma)\sin(\beta) + \sin(\alpha)\sin(\gamma) & \cos(\alpha)\sin(\beta)\sin(\gamma) + \cos(\gamma)\sin(\alpha) & \cos(\alpha)\cos(\beta) \end{bmatrix} \tag{24}$$

$$R_{z,a}(\alpha)R_{y,a}(\beta)R_{x,a}(\gamma) = E_{\mathrm{CIVA}}(\alpha,\beta,\gamma) = \begin{bmatrix} \cos(\alpha)\cos(\beta) & \cos(\alpha)\sin(\beta)\sin(\gamma)+\cos(\gamma)\sin(\alpha) & -\cos(\alpha)\cos(\gamma)\sin(\beta)+\sin(\alpha)\sin(\gamma) \\ -\cos(\beta)\sin(\alpha) & -\sin(\alpha)\sin(\beta)\sin(\gamma)+\cos(\alpha)\cos(\gamma) & \cos(\gamma)\sin(\alpha)\sin(\beta)+\cos(\alpha)\sin(\gamma) \\ \sin(\beta) & -\cos(\beta)\sin(\gamma) & \cos(\beta)\cos(\gamma) \end{bmatrix}, \tag{25}$$

4 Specific Transforms

From these we may enumerate the set of equations we must solve to transform from one system (*i.e.*,coordinate system and Euler angle conventions) to another. They are

$$E_{\mathrm{OnScale}}(\alpha_f,\beta_f,\gamma_f) = T_{B,O}E_{\mathrm{Bunge}}(\alpha_i,\beta_i,\gamma_i)T_{B,O}^{-1}, \tag{26}$$
$$E_{\mathrm{Bunge}}(\alpha_f,\beta_f,\gamma_f) = T_{O,B}E_{\mathrm{OnScale}}(\alpha_i,\beta_i,\gamma_i)T_{O,B}^{-1}, \tag{27}$$
$$E_{\mathrm{Civa}}(\alpha_f,\beta_f,\gamma_f) = T_{B,C}E_{\mathrm{Bunge}}(\alpha_i,\beta_i,\gamma_i)T_{B,C}^{-1}, \tag{28}$$
$$E_{\mathrm{Bunge}}(\alpha_f,\beta_f,\gamma_f) = T_{C,B}E_{\mathrm{Civa}}(\alpha_i,\beta_i,\gamma_i)T_{C,B}^{-1}, \tag{29}$$
$$E_{\mathrm{OnScale}}(\alpha_f,\beta_f,\gamma_f) = T_{C,O}E_{\mathrm{CIVA}}(\alpha_i,\beta_i,\gamma_i)T_{C,O}^{-1}, \tag{30}$$
$$E_{\mathrm{CIVA}}(\alpha_f,\beta_f,\gamma_f) = T_{O,C}E_{\mathrm{OnScale}}(\alpha_i,\beta_i,\gamma_i)T_{O,C}^{-1}, \tag{31}$$

4.1 Bunge to OnScale Conversion

We rewrite Eq. (26) as an explicit matrix equation to obtain (using SageMath)

$$T_{B,O}E_{\mathrm{Bunge}}(\alpha_i,\beta_i,\gamma_i)T_{B,O}^{-1} = \begin{bmatrix} -\cos(\beta_i)\sin(\alpha_i)\sin(\gamma_i)+\cos(\alpha_i)\cos(\gamma_i) & \cos(\beta_i)\cos(\gamma_i)\sin(\alpha_i)+\cos(\alpha_i)\sin(\gamma_i) & -\sin(\alpha_i)\sin(\beta_i) \\ -\cos(\alpha_i)\cos(\beta_i)\sin(\gamma_i)-\cos(\gamma_i)\sin(\alpha_i) & \cos(\alpha_i)\cos(\beta_i)\cos(\gamma_i)-\sin(\alpha_i)\sin(\gamma_i) & -\cos(\alpha_i)\sin(\beta_i) \\ -\sin(\beta_i)\sin(\gamma_i) & \cos(\gamma_i)\sin(\beta_i) & \cos(\beta_i) \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\beta_f)\cos(\gamma_f) & -\cos(\beta_f)\sin(\gamma_f) & \sin(\beta_f) \\ \cos(\gamma_f)\sin(\alpha_f)\sin(\beta_f)+\cos(\alpha_f)\sin(\gamma_f) & -\sin(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\alpha_f)\cos(\gamma_f) & -\cos(\beta_f)\sin(\alpha_f) \\ -\cos(\alpha_f)\cos(\gamma_f)\sin(\beta_f)+\sin(\alpha_f)\sin(\gamma_f) & \cos(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\gamma_f)\sin(\alpha_f) & \cos(\alpha_f)\cos(\beta_f) \end{bmatrix}. \tag{32}$$

To simplify subsequent discussion, and to expedite accurate software implementation, we rewrite the first matrix in this equation in the more compact form (these variables appear in the Python implementation)

$$F \equiv \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}, \tag{33}$$

so that Eq. (32) becomes

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} =$$

$$\begin{bmatrix} \cos(\beta_f)\cos(\gamma_f) & -\cos(\beta_f)\sin(\gamma_f) & \sin(\beta_f) \\ \cos(\gamma_f)\sin(\alpha_f)\sin(\beta_f)+\cos(\alpha_f)\sin(\gamma_f) & -\sin(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\alpha_f)\cos(\gamma_f) & -\cos(\beta_f)\sin(\alpha_f) \\ -\cos(\alpha_f)\cos(\gamma_f)\sin(\beta_f)+\sin(\alpha_f)\sin(\gamma_f) & \cos(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\gamma_f)\sin(\alpha_f) & \cos(\alpha_f)\cos(\beta_f) \end{bmatrix}. \tag{34}$$

Equation (34) enables us to "factor" the final matrix into the product of three rotations and thus recover the final Euler angles, α_f, β_f, and γ_f. In practice, this amounts to no more than picking off suitable

components of the matrix and solving for the final angles using that fact that α_i, β_i, and γ_i are known. Here are the equations displayed in the order in which they should be solved. They were obtained by equating the $(1,3)$, $(2,3)$ and $(1,2)$ elements of the matrices in the preceeding equation,

$$\begin{aligned}\beta_f &= \sin^{-1}\left(a_{1,3}\right),\\ \alpha_f &= \sin^{-1}\left(\frac{a_{2,3}}{\cos(\beta_f)}\right),\\ \gamma_f &= \sin^{-1}\left(\frac{a_{1,2}}{-\cos(\beta_f)}\right),\end{aligned} \tag{35}$$

where the angles α_i, β_i, γ_i are specified in the Bunge system and α_f, β_f, γ_f are the geometrically equivalent Euler angles for the OnScale system[2].

While these are definitely obvious equations, they are numerically unstable since they require evaluation of $\sin^{-1}$. The use of this function for computation of α_f, β_f, γ_f should be avoided as it is not even possible to perform a "back-and-forth" test [3] without accuracy loss approaching 20% in some cases. This is due to the fact that the derivative of $\sin^{-1}$ is infinite near ± 1. We will however, write similar equations for all of the system conversions we derive, since they illustrate the limits imposed by "gimbal lock".

Alterntive expressions based on $\tan^{-1}$ are more accurate, as we might expect, since its derivative is finite everywhere. Taking that approach we divide the $(2,3)$ by the $(3,3)$ elements appearing in Eq ((32)) we obtain

$$\alpha_f = -\tan^{-1}\left(\frac{a_{2,3}}{a_{3,3}}\right) \pm k\pi, \qquad k = 0, 1, \ldots. \tag{36}$$

If we divide the $(1,2)$ element by by the $(1,1)$ element we obtain

$$\gamma_f = -\tan^{-1}\left(\frac{a_{1,2}}{a_{1,1}}\right) \pm n\pi, \qquad n = 0, 1, \ldots. \tag{37}$$

Finally, we observe that $\pm\sqrt{a_{1,1}^2 + a_{1,2}^2} = \cos(\beta_f)$ so that

$$\beta_f = \pm\tan^{-1}\left(\frac{a_{1,3}}{\sqrt{a_{1,1}^2 + a_{1,2}^2}}.\right) \pm m\pi, \qquad m = 0, 1, \ldots. \tag{38}$$

where the $+$ or $-$ sign is chosen so that

$$\begin{cases}\beta_f > 0 & \text{if } a_{1,3} > 0,\\ \beta_f < 0 & \text{if } a_{1,3} < 0,\end{cases} \tag{39}$$

We need consider only two choices each of k, n, m due to the periodicity of the function $\tan(\theta)$. Our Python implementation considers $k, n, m = \pm 1$, other conventons are also possible and the software could be generalized to account for this. Determination of the correct valules of k, n, m is determined in the Python implementation of Eqs.(36) through (37) by enumerating all 2^3 posibilities and testing which satisfy Eq. (34)). At most twwo solutions may be valild although extensive testing indicates that only one valid solution is the typical outcome. If two valid solutions are found, the conversion software stores the first in the converted Euler angles file produced by the Python implementaion. The second solution is, however, equally valid since it produces the same rotation matrix, which is the only way in which Euler angles enter physical modelling calculations.

If $\beta_f = n\pi/2$ where n is an integer we cannot employ Eqs (35). However, in this case Eqs (34) simplifies

[2] by geometric equivalence we mean that any vector is rotated to the same vector regardless of which system is used to perform the calculations.

[3] *e.g.*, conversion of test coordinates α_i, β_i, and γ_i, from **CIVA** to **OnScale** and the back to **CIVA** should result in the same **rotation matrix**. However, the angles need not return back to the same values. This depends on the conventions chosem to define the each system system

to

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} =$$
$$\begin{bmatrix} 0 & 0 & (-1)^n \\ (-1)^n \cos(\gamma_f)\sin(\alpha_f) + \cos(\alpha_f)\sin(\gamma_f) & (-1)^{n+1}\sin(\alpha_f)\sin(\gamma_f) + \cos(\alpha_f)\cos(\gamma_f) & 0 \\ (-1)^{n+1}\cos(\alpha_f)\cos(\gamma_f) + \sin(\alpha_f)\sin(\gamma_f) & (-1)^n \cos(\alpha_f)\sin(\gamma_f) + \cos(\gamma_f)\sin(\alpha_f) & 0 \end{bmatrix}. \quad (40)$$

If n is even, the last matrix in Eq. (40) becomes

$$\begin{aligned} L &\equiv \begin{bmatrix} 0 & 0 & 1 \\ \cos(\gamma_f)\sin(\alpha_f) + \cos(\alpha_f)\sin(\gamma_f) & -\sin(\alpha_f)\sin(\gamma_f) + \cos(\alpha_f)\cos(\gamma_f) & 0 \\ -\cos(\alpha_f)\cos(\gamma_f) + \sin(\alpha_f)\sin(\gamma_f) & \cos(\alpha_f)\sin(\gamma_f) + \cos(\gamma_f)\sin(\alpha_f) & 0 \end{bmatrix}, \\ &= \begin{bmatrix} 0 & 0 & 1 \\ \sin(\gamma_f + \alpha_f) & \cos(\gamma_f + \alpha_f) & 0 \\ -\cos(\gamma_f + \alpha_f) & \sin(\gamma_f + \alpha_f) & 0 \end{bmatrix}. \end{aligned} \quad (41)$$

Equations (40) and (41) may be combined to obtain

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \sin(\gamma_f + \alpha_f) & \cos(\gamma_f + \alpha_f) & 0 \\ -\cos(\gamma_f + \alpha_f) & \sin(\gamma_f + \alpha_f) & 0 \end{bmatrix}, \quad (42)$$

from which we may extract only equations that enable computation of $\alpha_f + \gamma_f$, which we recognize as a manifestation of "gimbal lock". Consequently, we cannot perform the conversion in this case.

Similarly if n is odd, inspection of Eq. (40) shows that $L \to -L$ and, in this case also, we may only solve for $\alpha_f + \gamma_f$. Consequently, we cannot perform the conversion in this case either.

4.2 OnScale to Bunge Conversion

We next rewrite Eq. (27) as an explicit matrix equation to obtain (using SageMath)

$$T_{O,B} E_{\text{OnScale}}(\alpha_i, \beta_i, \gamma_i) T_{O,B}^{-1} =$$
$$\begin{bmatrix} \cos(\beta_i)\cos(\gamma_i) & -\cos(\beta_i)\sin(\gamma_i) & -\sin(\beta_i) \\ \cos(\gamma_i)\sin(\alpha_i)\sin(\beta_i) + \cos(\alpha_i)\sin(\gamma_i) & -\sin(\alpha_i)\sin(\beta_i)\sin(\gamma_i) + \cos(\alpha_i)\cos(\gamma_i) & \cos(\beta_i)\sin(\alpha_i) \\ \cos(\alpha_i)\cos(\gamma_i)\sin(\beta_i) - \sin(\alpha_i)\sin(\gamma_i) & -\cos(\alpha_i)\sin(\beta_i)\sin(\gamma_i) - \cos(\gamma_i)\sin(\alpha_i) & \cos(\alpha_i)\cos(\beta_i) \end{bmatrix}$$
$$=$$
$$\begin{bmatrix} -\cos(\beta_f)\sin(\alpha_f)\sin(\gamma_f) + \cos(\alpha_f)\cos(\gamma_f) & \cos(\beta_f)\cos(\gamma_f)\sin(\alpha_f) + \cos(\alpha_f)\sin(\gamma_f) & \sin(\alpha_f)\sin(\beta_f) \\ -\cos(\alpha_f)\cos(\beta_f)\sin(\gamma_f) - \cos(\gamma_f)\sin(\alpha_f) & \cos(\alpha_f)\cos(\beta_f)\cos(\gamma_f) - \sin(\alpha_f)\sin(\gamma_f) & \cos(\alpha_f)\sin(\beta_f) \\ \sin(\beta_f)\sin(\gamma_f) & -\cos(\gamma_f)\sin(\beta_f) & \cos(\beta_f) \end{bmatrix}. \quad (43)$$

As was done in Eq. (33), we replace the first matrix with the compact form representing its elements by $a_{i,j}$, $i, j = 1, 2, 3$ in order to simplify subsequent discussion, and to expedite accurate software implementation, we rewrite the first matrix in this equation in the more compact form

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} =$$

$$\begin{bmatrix} -\cos(\beta_f)\sin(\alpha_f)\sin(\gamma_f) + \cos(\alpha_f)\cos(\gamma_f) & \cos(\beta_f)\cos(\gamma_f)\sin(\alpha_f) + \cos(\alpha_f)\sin(\gamma_f) & \sin(\alpha_f)\sin(\beta_f) \\ -\cos(\alpha_f)\cos(\beta_f)\sin(\gamma_f) - \cos(\gamma_f)\sin(\alpha_f) & \cos(\alpha_f)\cos(\beta_f)\cos(\gamma_f) - \sin(\alpha_f)\sin(\gamma_f) & \cos(\alpha_f)\sin(\beta_f) \\ \sin(\beta_f)\sin(\gamma_f) & -\cos(\gamma_f)\sin(\beta_f) & \cos(\beta_f) \end{bmatrix}. \quad (44)$$

Equation (43) enables us to "factor" the final matrix into the product of three rotations and thus recover the final Euler angles, α_f, β_f, and γ_f. In practice, this amounts to no more than picking off suitable components of the matrix and solving for the final angles using that fact that α_i, β_i, and γ_i are known. The following equations (displayed in the order in which they should be solved) were obtained by equating the $(3,3)$, $(2,3)$ and $(3,2)$ elements of the matrices in the preceeding equation,

$$\begin{aligned}
\beta_f &= \cos^{-1}\left(\cos(\alpha_i)\cos(\beta_i)\right), \\
\alpha_f &= \cos^{-1}\left(\frac{\sin(\alpha_i)\cos(\beta_i)}{\sin(\beta_f)}\right), \\
\gamma_f &= \cos^{-1}\left(\frac{\sin\left(\beta_i\right)\sin\left(\gamma_i\right)\cos\left(\alpha_i\right)+\sin\left(\alpha_i\right)\cos\left(\gamma_i\right)}{\sin(\beta_f)}\right),
\end{aligned} \tag{45}$$

where the angles α_i, β_i, γ_i are specified in the Bunge system and α_f, β_f, γ_f are the geometrically equivalent Euler angles for the OnScale system. As discussed in §4.1 these equations are numerically inaccurate but they facilitate clear discussions of restrictions on intersystem conversion that are imposed by "gimbal lock".

A numerically stable equation for α_f is obtained by dividing the $(1,3)$ element by the $(2,3)$, while to obtain γ_f we divide the $(3,1)$ element by the $(3,2)$ element to obtain

$$\alpha_f = \tan^{-1}\left(\frac{a_{1,3}}{a_{2,3}}\right) \pm k\pi, \qquad k = 0, 1, \ldots. \tag{46}$$

$$\gamma_f = -\tan^{-1}\left(\frac{a_{3,1}}{a_{3,2}}\right) \pm n\pi, \qquad n = 0, 1, \ldots. \tag{47}$$

Finally, we observe that $\pm\sqrt{a_{3,1}^2 + a_{3,2}^2} = \sin(\beta_f)$ so that

$$\beta_f = \pm\tan^{-1}\left(\frac{\sqrt{a_{3,1}^2 + a_{3,2}^2}}{a_{3,3}}\right) \pm m\pi, \qquad m = 0, 1, \ldots, \tag{48}$$

where the $+$ or $-$ sign is chosen so that

$$\begin{cases} \beta_f > 0 & \text{if } a_{3,1}/\sin(\gamma_f) > 0, \\ \beta_f < 0 & \text{if } a_{3,1}/\sin(\gamma_f) < 0. \end{cases} \tag{49}$$

Since $\operatorname{sgn}(a_{3,1})/\sin(\gamma_f)) = \operatorname{sgn}(a_{3,1}\sin(\gamma_f))$ we use the latter in Python implementations of Eq. (89) to avoid difficulties that might arise if $\sin(\gamma_f) = 0$. Finally, as discussed in §4.1 need consider only two choices for each of the k, n, m due to the periodicity of the function $\tan(\theta)$. Our Python implementation considers $k, n, m = \pm 1$, other conventons are also possible and the software could be generalized to account for this. Determination of the correct valules of k, n, m is determined in the Python implementation of Eqs.(46) through (47) by enumerating all 2^3 posibilities and testing which satisfy Eq. (44). The functions are written to return the last valid enumeration along with the total number of enumerations satisfying Eq. (44). Extensive numerical testing indicates that there is always only on valid enumeration. We have, therefore, not attempted to prove uniqueness.

Similar to the preceeding case, if $\beta_f = n\pi$ where n is an integer we cannot employ Eqs (45). However, in this case Eqs (44) simplifies to

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \\ \begin{bmatrix} (-1)^{n+1}\sin\left(\alpha_f\right)\sin\left(\gamma_f\right)+\cos\left(\alpha_f\right)\cos\left(\gamma_f\right) & (-1)^{n}\cos\left(\gamma_f\right)\sin\left(\alpha_f\right)+\cos\left(\alpha_f\right)\sin\left(\gamma_f\right) & 0 \\ (-1)^{n+1}\cos\left(\alpha_f\right)\sin\left(\gamma_f\right)-\cos\left(\gamma_f\right)\sin\left(\alpha_f\right) & (-1)^{n}\cos\left(\alpha_f\right)\cos\left(\gamma_f\right)-\sin\left(\alpha_f\right)\sin\left(\gamma_f\right) & 0 \\ 0 & 0 & (-1)^{n} \end{bmatrix} \tag{50}$$

If n is even, the last matrix in Eq. (50) becomes

$$L \equiv \begin{bmatrix} -\sin(\alpha_f)\sin(\gamma_f)+\cos(\alpha_f)\cos(\gamma_f) & \cos(\gamma_f)\sin(\alpha_f)+\cos(\alpha_f)\sin(\gamma_f) & 0 \\ -\cos(\alpha_f)\sin(\gamma_f)-\cos(\gamma_f)\sin(\alpha_f) & \cos(\alpha_f)\cos(\gamma_f)-\sin(\alpha_f)\sin(\gamma_f) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$
$$= \begin{bmatrix} \cos(\alpha_f+\gamma_f) & \sin(\alpha_f+\gamma_f) & 0 \\ -\sin(\alpha_f+\gamma_f) & -\cos(\alpha_f+\gamma_f) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{51}$$

So that (50) becomes

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} \cos(\alpha_f+\gamma_f) & \sin(\alpha_f+\gamma_f) & 0 \\ -\sin(\alpha_f+\gamma_f) & -\cos(\alpha_f+\gamma_f) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{52}$$

from which we may extract only equations that enable computation of $\alpha_f+\gamma_f$, which we recognize as a manifestation of "gimbal lock". Consequently, we cannot perform the conversion in this case.

Similarly if n is odd, inspection of Eq. (50) shows that $L \to -L$ and, in this case also, we may only solve for $\alpha_f+\gamma_f$. Consequently, we cannot perform the conversion in this case either.

4.3 Bunge to CIVA Conversion

We next rewrite Eq. (28) as an explicit matrix equation to obtain (using SageMath)

$$T_{B,C}E_{\text{Bunge}}(\alpha_i,\beta_i,\gamma_i)T_{B,C}^{-1} =$$
$$\begin{bmatrix} -\cos(\beta_i)\sin(\alpha_i)\sin(\gamma_i)+\cos(\alpha_i)\cos(\gamma_i) & \sin(\alpha_i)\sin(\beta_i) & -\cos(\beta_i)\cos(\gamma_i)\sin(\alpha_i)-\cos(\alpha_i)\sin(\gamma_i) \\ \sin(\beta_i)\sin(\gamma_i) & \cos(\beta_i) & \cos(\gamma_i)\sin(\beta_i) \\ \cos(\alpha_i)\cos(\beta_i)\sin(\gamma_i)+\cos(\gamma_i)\sin(\alpha_i) & -\cos(\alpha_i)\sin(\beta_i) & \cos(\alpha_i)\cos(\beta_i)\cos(\gamma_i)-\sin(\alpha_i)\sin(\gamma_i) \end{bmatrix}$$
$$=$$
$$\begin{bmatrix} \cos(\alpha_f)\cos(\beta_f) & \cos(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\gamma_f)\sin(\alpha_f) & -\cos(\alpha_f)\cos(\gamma_f)\sin(\beta_f)+\sin(\alpha_f)\sin(\gamma_f) \\ -\cos(\beta_f)\sin(\alpha_f) & -\sin(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\alpha_f)\cos(\gamma_f) & \cos(\gamma_f)\sin(\alpha_f)\sin(\beta_f)+\cos(\alpha_f)\sin(\gamma_f) \\ \sin(\beta_f) & -\cos(\beta_f)\sin(\gamma_f) & \cos(\beta_f)\cos(\gamma_f) \end{bmatrix}. \tag{53}$$

As was done in Eq. (33), we replace the first matrix with the compact form representing its elements by $a_{i,j}$, $i,j=1,2,3$ in order to simplify subsequent discussion, and to expedite accurate software implementation, we rewrite the first matrix in this equation in the more compact form

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} =$$
$$\begin{bmatrix} \cos(\alpha_f)\cos(\beta_f) & \cos(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\gamma_f)\sin(\alpha_f) & -\cos(\alpha_f)\cos(\gamma_f)\sin(\beta_f)+\sin(\alpha_f)\sin(\gamma_f) \\ -\cos(\beta_f)\sin(\alpha_f) & -\sin(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\alpha_f)\cos(\gamma_f) & \cos(\gamma_f)\sin(\alpha_f)\sin(\beta_f)+\cos(\alpha_f)\sin(\gamma_f) \\ \sin(\beta_f) & -\cos(\beta_f)\sin(\gamma_f) & \cos(\beta_f)\cos(\gamma_f) \end{bmatrix}. \tag{54}$$

Equation (53) enables us to "factor" the final matrix into the product of three rotations and thus recover the final Euler angles, α_f, β_f, and γ_f. In practice, this amounts to no more than picking off suitable components of the matrix and solving for the final angles using that fact that α_i, β_i, and γ_i are known. Here are one set of possible equations displayed in the order in which they should be solved. They were obtained by equating the $(3,1)$, $(3,2)$ and $(2,1)$ elements of the matrices in the preceeding equation,

$$\begin{aligned} \beta_f &= \sin^{-1}\left(\cos(\alpha_i)\cos(\beta_i)\sin(\gamma_i)+\sin(\alpha_i)\cos(\gamma_i)\right), \\ \gamma_f &= \sin^{-1}\left(\frac{\cos(\alpha_i)\sin(\beta_i)}{\cos(\beta_f)}\right), \\ \alpha_f &= \sin^{-1}\left(\frac{-\sin(\beta_i)\sin(\gamma_i)}{\cos(\beta_f)}\right), \end{aligned} \tag{55}$$

where the angles α_i, β_i, γ_i are specified in the Bunge system and α_f, β_f, γ_f are the geometrically equivalent Euler angles for the OnScale system.

As mentioned in the two preceeding sections these equations are relatively inaccurate; we state them to facilitae clear discussion of the limitations of intersystem conversion imposed by "gimbal lock".

More accurate expressions based on $\tan^{-1}$ may be obtained using other elements of the arrays appearing in Eq. (54). For example, if we divide the $(2,1)$ by the $(1,1)$ elements we obtain

$$\alpha_f = -\tan^{-1}\left(\frac{a_{2,1}}{a_{1,1}}\right) \pm k\pi, \qquad k = 0, 1, \ldots. \tag{56}$$

Similarly dividing the $(3,2)$ by the $(3,3)$ elements leads to

$$\gamma_f = -\tan^{-1}\left(\frac{a_{3,2}}{a_{3,3}}\right) \pm m\pi, \qquad m = 0, 1, \ldots. \tag{57}$$

Finally, since $\sqrt{a_{3,2}^2 + a_{3,3}^2} = \cos(\beta_f)$

$$\beta_f = \pm\tan^{-1}\left(\frac{a_{3,1}}{\sqrt{a_{3,2}^2 + a_{3,3}^2}}\right) \pm m\pi, \qquad m = 0, 1, \ldots \tag{58}$$

where the $+$ or $-$ sign is chosen so that

$$\begin{cases} \beta_f > 0 & \text{if } a_{3,1} > 0, \\ \beta_f < 0 & \text{if } a_{3,1} < 0. \end{cases} \tag{59}$$

Similar to the other cases, if $\beta_f = n\pi/2$ where n is an integer we cannot employ Eqs (55). However, in this case Eq. (54) simplies to

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} 0 & (-1)^n \cos(\alpha_f)\sin(\gamma_f) + \cos(\gamma_f)\sin(\alpha_f) & (-1)^{n+1}\cos(\alpha_f)\cos(\gamma_f) + \sin(\alpha_f)\sin(\gamma_f) \\ 0 & (-1)^{n+1}\sin(\alpha_f)\sin(\gamma_f) + \cos(\alpha_f)\cos(\gamma_f) & (-1)^n\cos(\gamma_f)\sin(\alpha_f) + \cos(\alpha_f)\sin(\gamma_f) \\ (-1)^n & 0 & 0 \end{bmatrix}. \tag{60}$$

If n is even, the last matrix in Eq. (60) may be simplified to

$$\begin{aligned} L &\equiv \begin{bmatrix} 0 & \cos(\alpha_f)\sin(\gamma_f) + \cos(\gamma_f)\sin(\alpha_f) & -\cos(\alpha_f)\cos(\gamma_f) + \sin(\alpha_f)\sin(\gamma_f) \\ 0 & -\sin(\alpha_f)\sin(\gamma_f) + \cos(\alpha_f)\cos(\gamma_f) & \cos(\gamma_f)\sin(\alpha_f) + \cos(\alpha_f)\sin(\gamma_f) \\ 1 & 0 & 0 \end{bmatrix}, \\ &= \begin{bmatrix} 0 & \sin(\alpha_f + \gamma_f) & -\cos(\alpha_f + \gamma_f) \\ 0 & \cos(\alpha_f + \gamma_f) & \sin(\alpha_f + \gamma_f) \\ 1 & 0 & 0 \end{bmatrix}, \end{aligned} \tag{61}$$

so that Eq. (61) becomes

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} 0 & \sin(\alpha_f + \gamma_f) & -\cos(\alpha_f + \gamma_f) \\ 0 & \cos(\alpha_f + \gamma_f) & \sin(\alpha_f + \gamma_f) \\ 1 & 0 & 0 \end{bmatrix}, \tag{62}$$

from which we may extract only equations that enable computation of $\alpha_f + \gamma_f$, which we recognize as a manifestation of "gimbal lock". Consequently, we cannot perform the conversion in this case.

Similarly if n is odd, inspection of Eq. (60) shows that $L \to -L$ and, in this case also, we may only solve for $\alpha_f + \gamma_f$. Consequently, we cannot perform the conversion in this case either.

4.4 CIVA to Bunge Conversion

We next rewrite Eq. (29) as an explicit matrix equation to obtain (using SageMath)

$$T_{C,B}E_{\mathrm{CIVA}}(\alpha_i,\beta_i,\gamma_i)T_{C,B}^{-1} =$$
$$\begin{bmatrix} \cos(\alpha_i)\cos(\beta_i) & \cos(\alpha_i)\cos(\gamma_i)\sin(\beta_i) - \sin(\alpha_i)\sin(\gamma_i) & \cos(\alpha_i)\sin(\beta_i)\sin(\gamma_i) + \cos(\gamma_i)\sin(\alpha_i) \\ -\sin(\beta_i) & \cos(\beta_i)\cos(\gamma_i) & \cos(\beta_i)\sin(\gamma_i) \\ -\cos(\beta_i)\sin(\alpha_i) & -\cos(\gamma_i)\sin(\alpha_i)\sin(\beta_i) - \cos(\alpha_i)\sin(\gamma_i) & -\sin(\alpha_i)\sin(\beta_i)\sin(\gamma_i) + \cos(\alpha_i)\cos(\gamma_i) \end{bmatrix}$$
$$=$$
$$\begin{bmatrix} -\cos(\beta_f)\sin(\alpha_f)\sin(\gamma_f) + \cos(\alpha_f)\cos(\gamma_f) & \cos(\beta_f)\cos(\gamma_f)\sin(\alpha_f) + \cos(\alpha_f)\sin(\gamma_f) & \sin(\alpha_f)\sin(\beta_f) \\ -\cos(\alpha_f)\cos(\beta_f)\sin(\gamma_f) - \cos(\gamma_f)\sin(\alpha_f) & \cos(\alpha_f)\cos(\beta_f)\cos(\gamma_f) - \sin(\alpha_f)\sin(\gamma_f) & \cos(\alpha_f)\sin(\beta_f) \\ \sin(\beta_f)\sin(\gamma_f) & -\cos(\gamma_f)\sin(\beta_f) & \cos(\beta_f) \end{bmatrix}. \tag{63}$$

As was done in Eq. (33), we replace the first matrix with the compact form representing its elements by $a_{i,j}$, $i,j = 1,2,3$ in order to simplify subsequent discussion, and to expedite accurate software implementation, we rewrite the first matrix in this equation in the more compact form

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} =$$
$$\begin{bmatrix} -\cos(\beta_f)\sin(\alpha_f)\sin(\gamma_f) + \cos(\alpha_f)\cos(\gamma_f) & \cos(\beta_f)\cos(\gamma_f)\sin(\alpha_f) + \cos(\alpha_f)\sin(\gamma_f) & \sin(\alpha_f)\sin(\beta_f) \\ -\cos(\alpha_f)\cos(\beta_f)\sin(\gamma_f) - \cos(\gamma_f)\sin(\alpha_f) & \cos(\alpha_f)\cos(\beta_f)\cos(\gamma_f) - \sin(\alpha_f)\sin(\gamma_f) & \cos(\alpha_f)\sin(\beta_f) \\ \sin(\beta_f)\sin(\gamma_f) & -\cos(\gamma_f)\sin(\beta_f) & \cos(\beta_f) \end{bmatrix}. \tag{64}$$

Equation (63) enables us to "factor" the final matrix into the product of three rotations and thus recover the final Euler angles, α_f, β_f, and γ_f. In practice, this amounts to no more than picking off suitable components of the matrix and solving for the final angles using that fact that α_i, β_i, and γ_i are known. Here are the equations displayed in the order in which they should be solved. They were obtained by equating the $(3,3)$, $(3,2)$ and $(2,3)$ elements of the matrices in the preceeding equation,

$$\begin{aligned} \beta_f &= \cos^{-1}\left(-\sin(\alpha_i)\sin(\beta_i)\sin(\gamma_i) + \cos(\alpha_i)\cos(\gamma_i)\right), \\ \gamma_f &= \cos^{-1}\left(\frac{\cos(\gamma_i)\sin(\alpha_i)\sin(\beta_i) + \cos(\alpha_i)\sin(\gamma_i)}{\sin(\beta_f)}\right), \\ \alpha_f &= \cos^{-1}\left(\frac{\cos(\beta_i)\sin(\gamma_i)}{\sin(\beta_f)}\right), \end{aligned} \tag{65}$$

where the angles α_i, β_i, γ_i are specified in the Bunge system and α_f, β_f, γ_f are the geometrically equivalent Euler angles for the OnScale system.

As mentioned in the two preceeding sections these equations are relatively inaccurate; we state them to facilitae clear discussion of the limitations of intersystem conversion imposed by "gimbal lock".

More accurate expressions based on $\tan^{-1}$ may be obtained using other elements of the arrays appearing in Eq. (64). For example, if we divide the $(1,3)$ by the $(2,3)$ elements we obtain

$$\alpha_f = \tan^{-1}\left(\frac{a_{1,3}}{a_{2,3}}\right) \pm k\pi, \qquad k = 0,1,\ldots. \tag{66}$$

Similarly, if we divide the $(3,1)$ by the $(3,2)$ elements we obtain

$$\gamma_f = -\tan^{-1}\left(\frac{a_{3,1}}{a_{3,2}}\right) \pm n\pi, \qquad n = 0,1,\ldots. \tag{67}$$

Finally, since $\sqrt{a_{3,1}^2 + a_{3,2}^2} = \sin(\beta_f)$

$$\beta_f = \tan^{-1}\left(\frac{\sqrt{a_{3,1}^2 + a_{3,2}^2}}{a_{3,3}}\right) \pm m\pi, \qquad m = 0,1,\ldots. \tag{68}$$

where the + or − sign is chosen so that

$$\begin{cases} \beta_f > 0 & \text{if } a_{3,1}/\sin(\gamma_f) > 0, \\ \beta_f < 0 & \text{if } a_{3,1}/\sin(\gamma_f) < 0. \end{cases} \tag{69}$$

Since $\mathrm{sgn}(a_{3,1}/\sin(\gamma_f)) = \mathrm{sgn}(a_{3,1}\sin(\gamma_f))$ we use the latter in Python implementations of Eq. (89) to avoid difficulties that might arise if $\sin(\gamma_f) = 0$.

Similar to the OnScale-to-Bunge case, if $\beta_f = n\pi$ where n is an integer we cannot employ Eqs (65). However, in this case Eq. (64) simplies to

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$
$$=$$
$$\begin{bmatrix} (-1)^{n+1}\sin(\alpha_f)\sin(\gamma_f) + \cos(\alpha_f)\cos(\gamma_f) & (-1)^n\cos(\gamma_f)\sin(\alpha_f) + \cos(\alpha_f)\sin(\gamma_f) & 0 \\ (-1)^{n+1}\cos(\alpha_f)\sin(\gamma_f) - \cos(\gamma_f)\sin(\alpha_f) & (-1)^n\cos(\alpha_f)\cos(\gamma_f) - \sin(\alpha_f)\sin(\gamma_f) & 0 \\ 0 & 0 & (-1)^n \end{bmatrix}. \tag{70}$$

If n is even, the last matrix in Eq. (70) becomes

$$\begin{aligned} L &\equiv \begin{bmatrix} -\sin(\alpha_f)\sin(\gamma_f) + \cos(\alpha_f)\cos(\gamma_f) & -\cos(\gamma_f)\sin(\alpha_f) + \cos(\alpha_f)\sin(\gamma_f) & 0 \\ -\cos(\alpha_f)\sin(\gamma_f) - \cos(\gamma_f)\sin(\alpha_f) & -\cos(\alpha_f)\cos(\gamma_f) - \sin(\alpha_f)\sin(\gamma_f) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ &= \begin{bmatrix} \cos(\alpha_f+\gamma_f) & \sin(\alpha_f+\gamma_f) & 0 \\ \sin(\alpha_f+\gamma_f) & \cos(\alpha_f+\gamma_f) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{71}$$

Equations (71) may then be rewritten as

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} \cos(\alpha_f+\gamma_f) & \sin(\alpha_f+\gamma_f) & 0 \\ \sin(\alpha_f+\gamma_f) & \cos(\alpha_f+\gamma_f) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{72}$$

from which we may extract only equations that enable computation of $\alpha_f + \gamma_f$, which we recognize as a manifestation of "gimbal lock". Consequently, we cannot perform the conversion in this case.

Similarly if n is odd, inspection of Eq. (70) shows that $L \to -L$ and, in this case also, we may only solve for $\alpha_f + \gamma_f$. Consequently, we cannot perform the conversion in this case either.

4.5 CIVA to OnScale Conversion

We next rewrite Eq. (30) as an explicit matrix equation to obtain (using SageMath)

$$T_{C,O}E_{\mathrm{CIVA}}(\alpha_i,\beta_i,\gamma_i)T_{C,O}^{-1} =$$
$$\begin{bmatrix} \cos(\alpha_i)\cos(\beta_i) & \cos(\alpha_i)\cos(\gamma_i)\sin(\beta_i) - \sin(\alpha_i)\sin(\gamma_i) & -\cos(\alpha_i)\sin(\beta_i)\sin(\gamma_i) - \cos(\gamma_i)\sin(\alpha_i) \\ -\sin(\beta_i) & \cos(\beta_i)\cos(\gamma_i) & -\cos(\beta_i)\sin(\gamma_i) \\ \cos(\beta_i)\sin(\alpha_i) & \cos(\gamma_i)\sin(\alpha_i)\sin(\beta_i) + \cos(\alpha_i)\sin(\gamma_i) & -\sin(\alpha_i)\sin(\beta_i)\sin(\gamma_i) + \cos(\alpha_i)\cos(\gamma_i) \end{bmatrix}$$
$$=$$
$$\begin{bmatrix} \cos(\beta_f)\cos(\gamma_f) & -\cos(\beta_f)\sin(\gamma_f) & \sin(\beta_f) \\ \cos(\gamma_f)\sin(\alpha_f)\sin(\beta_f) + \cos(\alpha_f)\sin(\gamma_f) & -\sin(\alpha_f)\sin(\beta_f)\sin(\gamma_f) + \cos(\alpha_f)\cos(\gamma_f) & -\cos(\beta_f)\sin(\alpha_f) \\ -\cos(\alpha_f)\cos(\gamma_f)\sin(\beta_f) + \sin(\alpha_f)\sin(\gamma_f) & \cos(\alpha_f)\sin(\beta_f)\sin(\gamma_f) + \cos(\gamma_f)\sin(\alpha_f) & \cos(\alpha_f)\cos(\beta_f) \end{bmatrix}. \tag{73}$$

As was done in Eq. (33), we replace the first matrix with the compact form representing its elements by $a_{i,j}$, $i,j = 1,2,3$ in order to simplify subsequent discussion, and to expedite accurate software implementation,

we rewrite the first matrix in this equation in the more compact form

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} \cos(\beta_f)\cos(\gamma_f) & -\cos(\beta_f)\sin(\gamma_f) & \sin(\beta_f) \\ \cos(\gamma_f)\sin(\alpha_f)\sin(\beta_f)+\cos(\alpha_f)\sin(\gamma_f) & -\sin(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\alpha_f)\cos(\gamma_f) & -\cos(\beta_f)\sin(\alpha_f) \\ -\cos(\alpha_f)\cos(\gamma_f)\sin(\beta_f)+\sin(\alpha_f)\sin(\gamma_f) & \cos(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\gamma_f)\sin(\alpha_f) & \cos(\alpha_f)\cos(\beta_f) \end{bmatrix}. \tag{74}$$

Equation (73) enables us to "factor" the final matrix into the product of three rotations and thus recover the final Euler angles, α_f, β_f, and γ_f. In practice, this amounts to no more than picking off suitable components of the matrix and solving for the final angles using that fact that α_i, β_i, and γ_i are known. Here are onse set of possible equations (displayed in the order in which they should be solved). They were obtained by equating the $(1,3)$, $(1,2)$ and $(2,3)$ elements of the matrices in the preceeding equation,

$$\begin{aligned} \beta_f &= \sin^{-1}(-\cos(\alpha_i)\sin(\beta_i)\sin(\gamma_i) - \cos(\gamma_i)\sin(\alpha_i)), \\ \gamma_f &= \sin^{-1}\left(\frac{\cos(\alpha_i)\cos(\gamma_i)\sin(\beta_i) - \sin(\alpha_i)\sin(\gamma_i)}{-\cos(\beta_f)}\right), \\ \alpha_f &= \sin^{-1}\left(\frac{\cos(\beta_i)\sin(\gamma_i)}{\cos(\beta_f)}\right), \end{aligned} \tag{75}$$

where the angles α_i, β_i, γ_i are specified in the Bunge system and α_f, β_f, γ_f are the geometrically equivalent Euler angles for the OnScale system.

As mentioned in the three preceeding sections these equations are relatively inaccurate; we state them to facilitae clear discussion of the limitations of intersystem conversion imposed by "gimbal lock".

More accurate expressions based on $\tan^{-1}$ may be obtained using other elements of the arrays appearing in Eq. (74). For example, if we divide the $(2,3)$ by the $(3,3)$ elements we obtain

$$\alpha_f = -\tan^{-1}\left(\frac{a_{2,3}}{a_{3,3}}\right) \pm k\pi, \qquad k = 0, 1, \ldots. \tag{76}$$

while if we divide the $(1,2)$ by the $(1,1)$ elements we obtain

$$\gamma_f = -\tan^{-1}\left(\frac{a_{1,2}}{a_{1,1}}\right) \pm n\pi, \qquad n = 0, 1, \ldots. \tag{77}$$

Finally, since $\sqrt{a_{1,1}^2 + a_{1,2}^2} = \cos(\beta_f)$ we obtain

$$\beta_f = \tan^{-1}\left(\frac{a_{1,3}}{\sqrt{a_{1,1}^2 + a_{1,2}^2}}\right) \pm m\pi, \qquad m = 0, 1, \ldots. \tag{78}$$

where the $+$ or $-$ sign is chosen so that

$$\begin{cases} \beta_f > 0 & \text{if } a_{1,3} > 0, \\ \beta_f < 0 & \text{if } a_{1,3} < 0. \end{cases} \tag{79}$$

Similar to the Bunge-to-OnScale case, if $\beta_f = n\pi/2$ where n is an integer we cannot employ Eqs (75). However, in this case Eq. (73) simplies to

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & (-1)^n \\ (-1)^n\cos(\gamma_f)\sin(\alpha_f)+\cos(\alpha_f)\sin(\gamma_f) & (-1)^{n+1}\sin(\alpha_f)\sin(\gamma_f)+\cos(\alpha_f)\cos(\gamma_f) & 0 \\ (-1)^{n+1}\cos(\alpha_f)\cos(\gamma_f)+\sin(\alpha_f)\sin(\gamma_f) & (-1)^n\cos(\alpha_f)\sin(\gamma_f)+\cos(\gamma_f)\sin(\alpha_f) & 0 \end{bmatrix}. \tag{80}$$

If n is even, the last matrix in Eq. (80) becomes

$$
\begin{aligned}
L &\equiv \begin{bmatrix} 0 & 0 & 1 \\ \cos(\gamma_f)\sin(\alpha_f)+\cos(\alpha_f)\sin(\gamma_f) & -\sin(\alpha_f)\sin(\gamma_f)+\cos(\alpha_f)\cos(\gamma_f) & 0 \\ -\cos(\alpha_f)\cos(\gamma_f)+\sin(\alpha_f)\sin(\gamma_f) & \cos(\alpha_f)\sin(\gamma_f)+\cos(\gamma_f)\sin(\alpha_f) & 0 \end{bmatrix}, \\
&= \begin{bmatrix} 0 & 0 & 1 \\ \sin(\alpha_f+\gamma_f) & \cos(\alpha_f+\gamma_f) & 0 \\ -\cos(\alpha_f+\gamma_f) & \sin(\alpha_f+\gamma_f) & 0 \end{bmatrix},
\end{aligned} \tag{81}
$$

Equation (80) then becomes

$$
\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \sin(\alpha_f+\gamma_f) & \cos(\alpha_f+\gamma_f) & 0 \\ -\cos(\alpha_f+\gamma_f) & \sin(\alpha_f+\gamma_f) & 0 \end{bmatrix}, \tag{82}
$$

from which we may extract only equations that enable computation of $\alpha_f+\gamma_f$, which we recognize as a manifestation of "gimbal lock". Consequently, we cannot perform the conversion in this case.

Similarly if n is odd, inspection of Eq. (80) shows that $L \to -L$ and, in this case also, we may only solve for $\alpha_f+\gamma_f$. Consequently, we cannot perform the conversion in this case either.

4.6 OnScale to CIVA Conversion

We finally rewrite Eq. (31) as an explicit matrix equation to obtain (using SageMath)

$$
\begin{aligned}
&T_{O,C}E_{\text{OnScale}}(\alpha_i,\beta_i,\gamma_i)T_{O,C}^{-1} = \\
&\begin{bmatrix} \cos(\beta_i)\cos(\gamma_i) & -\sin(\beta_i) & \cos(\beta_i)\sin(\gamma_i) \\ \cos(\alpha_i)\cos(\gamma_i)\sin(\beta_i)-\sin(\alpha_i)\sin(\gamma_i) & \cos(\alpha_i)\cos(\beta_i) & \cos(\alpha_i)\sin(\beta_i)\sin(\gamma_i)+\cos(\gamma_i)\sin(\alpha_i) \\ -\cos(\gamma_i)\sin(\alpha_i)\sin(\beta_i)-\cos(\alpha_i)\sin(\gamma_i) & -\cos(\beta_i)\sin(\alpha_i) & -\sin(\alpha_i)\sin(\beta_i)\sin(\gamma_i)+\cos(\alpha_i)\cos(\gamma_i) \end{bmatrix} \\
&= \\
&\begin{bmatrix} \cos(\alpha_f)\cos(\beta_f) & \cos(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\gamma_f)\sin(\alpha_f) & -\cos(\alpha_f)\cos(\gamma_f)\sin(\beta_f)+\sin(\alpha_f)\sin(\gamma_f) \\ -\cos(\beta_f)\sin(\alpha_f) & -\sin(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\alpha_f)\cos(\gamma_f) & \cos(\gamma_f)\sin(\alpha_f)\sin(\beta_f)+\cos(\alpha_f)\sin(\gamma_f) \\ \sin(\beta_f) & -\cos(\beta_f)\sin(\gamma_f) & \cos(\beta_f)\cos(\gamma_f) \end{bmatrix}.
\end{aligned} \tag{83}
$$

As was done in Eq. (33), we replace the first matrix with the compact form representing its elements by $a_{i,j}$, $i,j=1,2,3$ in order to simplify subsequent discussion, and to expedite accurate software implementation, we rewrite the first matrix in this equation in the more compact form

$$
\begin{aligned}
&\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \\
&\begin{bmatrix} \cos(\alpha_f)\cos(\beta_f) & \cos(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\gamma_f)\sin(\alpha_f) & -\cos(\alpha_f)\cos(\gamma_f)\sin(\beta_f)+\sin(\alpha_f)\sin(\gamma_f) \\ -\cos(\beta_f)\sin(\alpha_f) & -\sin(\alpha_f)\sin(\beta_f)\sin(\gamma_f)+\cos(\alpha_f)\cos(\gamma_f) & \cos(\gamma_f)\sin(\alpha_f)\sin(\beta_f)+\cos(\alpha_f)\sin(\gamma_f) \\ \sin(\beta_f) & -\cos(\beta_f)\sin(\gamma_f) & \cos(\beta_f)\cos(\gamma_f) \end{bmatrix}.
\end{aligned} \tag{84}
$$

Equation (83) enables us to "factor" the final matrix into the product of three rotations and thus recover the final Euler angles, α_f, β_f, and γ_f. In practice, this amounts to no more than picking off suitable components of the matrix and solving for the final angles using that fact that α_i, β_i, and γ_i are known. Here are the equations displayed in the order in which they should be solved. They were obtained by equating the $(3,1)$, $(3,2)$ and $(2,1)$ elements of the matrices in the preceeding equation,

$$
\begin{aligned}
\beta_f &= \sin^{-1}\left(-\cos(\gamma_i)\sin(\alpha_i)\sin(\beta_i)-\cos(\alpha_i)\sin(\gamma_i)\right), \\
\gamma_f &= \sin^{-1}\left(\frac{\cos(\beta_i)\sin(\alpha_i)}{\cos(\beta_f)}\right), \\
\alpha_f &= \sin^{-1}\left(\frac{\cos(\alpha_i)\cos(\gamma_i)\sin(\beta_i)-\sin(\alpha_i)\sin(\gamma_i)}{-\cos(\beta_f)}\right),
\end{aligned} \tag{85}
$$

where the angles α_i, β_i, γ_i are specified in the Bunge system and α_f, β_f, γ_f are the geometrically equivalent Euler angles for the OnScale system.

As mentioned in the five preceeding sections these equations are relatively inaccurate; we state them to facilitae clear discussion of the limitations of intersystem conversion imposed by "gimbal lock".

More accurate expressions based on $\tan^{-1}$ may be obtained using other elements of the arrays appearing in Eq. (74). For example, if we divide the $(3,2)$ by the $(2,2)$ elements we obtain

$$\alpha_f = -\tan^{-1}\left(\frac{a_{3,2}}{a_{2,2}}\right) \pm k\pi, \qquad k = 0, 1, \ldots. \tag{86}$$

while dividing the $(1,3)$ elements by the $(1,1)$ elements leads to

$$\gamma_f = -\tan^{-1}\left(\frac{a_{1,3}}{a_{1,1}}\right) \pm n\pi, \qquad n = 0, 1, \ldots. \tag{87}$$

Finally, since $\sqrt{a_{3,2}^2 + a_{3,3}^2} = \cos(\beta_f)$, we obtain

$$\beta_f = \tan^{-1}\left(\frac{a_{3,1}}{\sqrt{a_{3,2}^2 + a_{3,3}^2}}\right) \pm m\pi, \qquad m = 0, 1, \ldots. \tag{88}$$

where the $+$ or $-$ sign is chosen so that

$$\begin{cases} \beta_f > 0 & \text{if } a_{3,1} > 0, \\ \beta_f < 0 & \text{if } a_{3,1} < 0. \end{cases} \tag{89}$$

Similar to the other cases, if $\beta_f = n\pi/2$ where n is an integer we cannot employ Eqs. (85). However, in this case Eq. (83) simplies to

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} 0 & (-1)^n \cos(\alpha_f)\sin(\gamma_f) + \cos(\gamma_f)\sin(\alpha_f) & (-1)^{n+1}\cos(\alpha_f)\cos(\gamma_f) + \sin(\alpha_f)\sin(\gamma_f) \\ 0 & (-1)^{n+1}\sin(\alpha_f)\sin(\gamma_f) + \cos(\alpha_f)\cos(\gamma_f) & (-1)^n \cos(\gamma_f)\sin(\alpha_f) + \cos(\alpha_f)\sin(\gamma_f) \\ (-1)^n & 0 & 0 \end{bmatrix}. \tag{90}$$

If n is even, the last matrix in Eq. (90) becomes

$$\begin{bmatrix} 0 & \sin(\alpha_f + \gamma_f) & -\cos(\alpha_f + \gamma_f) \\ 0 & \cos(\alpha_f + \gamma_f) & \sin(\alpha_f + \gamma_f) \\ 1 & 0 & 0 \end{bmatrix}, \tag{91}$$

so that Eq. (90) simplifes finally to

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} 0 & \sin(\alpha_f + \gamma_f) & -\cos(\alpha_f + \gamma_f) \\ 0 & \cos(\alpha_f + \gamma_f) & \sin(\alpha_f + \gamma_f) \\ 1 & 0 & 0 \end{bmatrix}, \tag{92}$$

from which we may extract only equations that enable computation of $\alpha_f + \gamma_f$, which we recognize as a manifestation of "gimbal lock". Consequently, we cannot perform the conversion in this case.

Similarly if n is odd, inspection of Eq. (90) shows that $L \to -L$ and, in this case also, we may only solve for $\alpha_f + \gamma_f$. Consequently, we cannot perform the conversion in this case either.

References

[1] Granino A. Korn and Theresa M. Korn. *Mathematical Handbook for Scientists and Engineers*. Dover, Mineola, NY, USA, 1968.

[2] Joseph K. Davidson and Kenneth Henderson Hunt. *Robots and screw theory: applications of kinematics and statics to robotics*. Oxford University Press, Oxford, U.K., 2004.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354
1-888-375-PNNL (7665)

www.pnnl.gov* | *www.nrc.gov