Development of a Fast Analysis Solver for Welding Sequence Optimization of Ship Structures

Task Order 29 of Base Task Order Agreement 2018-453

Deliverable for Milestone 30

Final Report

Revision 0

Submitted by Ingalls Shipbuilding, a division of HII on behalf of the Integrated Project Team members:

Austal USA   Oak Ridge National Laboratory

General Motors LLC   Hexagon

January 2026

Distribution Unlimited
Data included in this report is considered Category A

Table of Contents

List of Figures

Executive Summary

Numerical simulations using Finite Element Analysis (FEA) have long been used to analyze and estimate welding distortion. However, for large assemblies such as a ship panel, these analyses can take days or weeks to set up, run, and obtain results and optimization is not feasible. To allow a ship builder to optimize welding sequences, a user-friendly and fast-analysis software for welding-process simulation has been developed to predict distortion and residual stress through the National Shipbuilding Research Program (NSRP) project entitled “Development of a Fast Analysis Solver for Welding Sequence Optimization of Ship Structures.”

The project started with the evaluation of the ORNL solver and Hexagon Simufact Welding software on a representative ship panel to identify the technical gaps for shipbuilding applications and develop software requirements. It was found that the major technical gap is that both the ORNL solver and the Hexagon software did not include shell elements which are mainly used in the design of ship structures.

ORNL added shell elements in the fast mechanical analysis solver (DR-Weld) and established two complementary approaches (transient analysis approach and strain-based approach) for distortion and residual-stress welding simulations. The transient analysis approach explicitly simulates the motion of each weld pass, capturing the detailed thermal–mechanical history of the welding process. It is well suited for weld sequence optimization with improved computational efficiency. The first version of this transient simulation capability has been validated by comparing distortion prediction with Abaqus on one-tee, single-cell, multi-cell and full-panel model. The strain-based approach employs a simplified formulation in which thermally induced strains are used as input to drive the mechanical response, eliminating the need for coupled transient thermal-mechanical simulations.

Hexagon developed FEA and analytical based fast thermal analysis solver to predict temperature using shell elements. The predicted temperature can be input to DR-Weld for mechanical analysis. The solvers were validated against reference thermo-mechanical results from Simufact, with reasonable accuracy.

Hexagon also developed an automated mesh creation tool (Auto-Mesher) in MSC Apex. The created mesh is used in thermal analysis and mechanical analysis. This development significantly reduces the mesh creation time for large ship structures. Shipyard tests show that the meshing time for a typical ship panel is reduced to several hours from several days using commercial meshing software.

A graphical user interface (GUI) and an automated analysis process was developed by integrating Auto-Mesher, thermal-analysis solver, DR-Weld, and post-process tool, which allows automatically running the thermal solver and DR-Weld, and importing and processing of results without user intervention.

Introduction

During the shipbuilding manufacturing process, the material is exposed to significant stresses induced both thermally and mechanically, that alter the intended design and significantly affect the production schedule, labor hours (fitting, welding, rework, etc.), and material structural performance. The type and magnitude of deformation of a given structure depends on many factors such as the material, thickness and quality of components, the process heat input, preheat and inter-pass temperatures, type and size of welds, welding sequence and direction, location, sequence, and degree of fixturing.

Numerical simulations using Finite Element Analysis (FEA) have long been used to analyze and estimate welding distortion. Joint level detailed welding analyses are required to capture the physics driving distortion and are performed by expert analysts with access to pedigreed material properties. For large assemblies however, these analyses can take days or weeks to set up, run, and obtain results and optimization is not feasible. Traditional approaches of performing welding analysis using FEA are not suitable for very large structures and production environments where hundreds of analyses are needed for each ship.

The project team is developing a user-friendly and fast finite element analysis solver for shipbuilding applications in this National Shipbuilding Research Program (NSRP) project entitled "Development of a Fast Analysis Solver for Welding Sequence Optimization of Ship Structures". The development has been conducted by leveraging the work of Oak Ridge National Laboratory (ORNL) and Hexagon for the automotive and nuclear industries.

ORNL Fast Solver (DR-Weld)

To reduce computational cost, the fusion welding process is generally solved in a one-way coupled manner with heat transfer analysis and mechanical analysis performed in sequence. Compared to heat transfer analysis, mechanical analysis requires a significantly longer computation time because of the greater degrees of freedom and the highly nonlinear thermal elastic-plastic responses required to solve. Therefore, the ORNL fast solver focuses on the acceleration of the mechanical analysis portion of welding stress and distortion simulation. The ORNL fast solver takes full advantage of modern *GPU-based HPC hardware* and uses an *explicit finite element method (FEM)* that has excellent scalability on GPU-based numerical solvers.

Explicit FEM with Time Scaling

Currently, most traditional analysis methods for welding simulation are based on implicit FEM that requires the solution of large simultaneous linear equations. A comparative study (Feng 2015) has shown that parallel computing with commercial code ABAQUS beyond 64 cores gains only marginal efficiency gains in welding simulations.

In developing methods to take further advantage of parallel computing potentials in welding simulation, the explicit FEM simulation becomes an attractive candidate to be used on a massive-core computer platform because it does not require assembling and solving the global matrix. Ma and Umetsu (2009 and 2016) initiated the application of explicit FEM in analyzing welding stress and distortion. Using LS-Dyna software and mass scaling techniques, they were able to accelerate the welding analysis by more than seven times as compared with simulation using implicit FEM. Ikushima and Shibahara (2014) developed an idealized explicit FEM by selecting the appropriate density for different elements. An acceleration factor of twelve times was achieved compared to the implicit FEM solver.

In the case of fusion welding, the heating and cooling time ranges from minutes to hours that can result in a great number of time integration cycles. In general, mass scaling and time scaling techniques can be used to accelerate computation for explicit simulations. The scaling factor is usually predefined based on a user's experience or a sensitivity study. Huang (2020) calibrated the optimum scaling factor SF by the inherent strain concept to significantly reduce the time integrated over heating processes. To account for a longer time span of cooling, a variable time scaling factor in a range of 1~100 times SF is employed as illustrated in Figure 1. The computational time is greatly reduced since the simulation event becomes much shorter (less than 1/SF of original time span) in an explicit simulation.

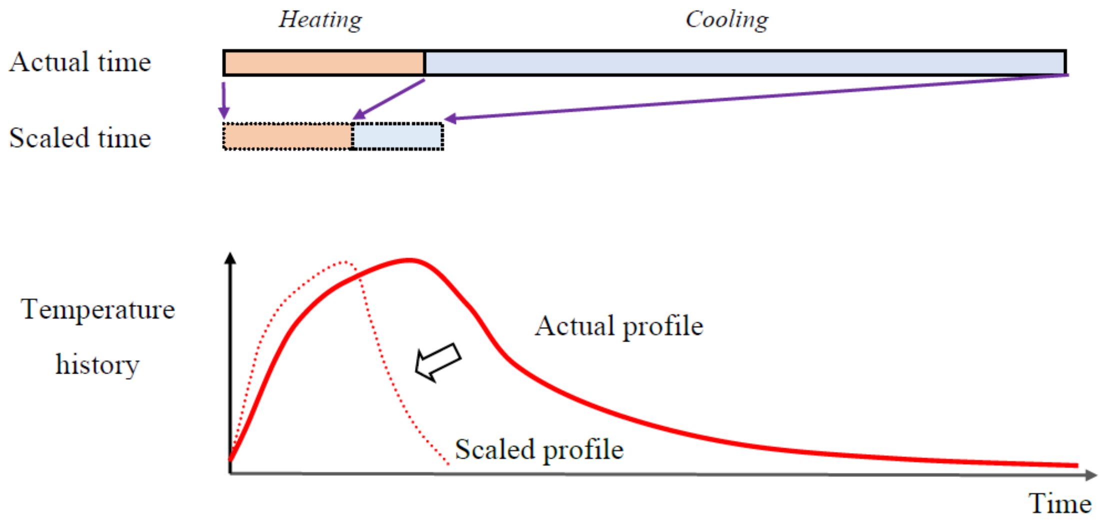


Figure 1. Schematic Drawing of Time Scaling for a Welding Simulation (Huang 2021)

GPU Implementation to Accelerate Time Integration

GPU was initially designed to support rapid manipulation of image data, and later it was extended to perform general computing traditionally handled by CPUs. The advent of programming frameworks such as CUDA, OpenACC and OpenCL has greatly advanced the scientific computing based on GPU/CPUs, leading to fast growth of computing power in modern supercomputers.

In recent years, the increasing double-precision capabilities in GPU devices such as NVIDA products Tesla and Pascal series have attracted significant interest from the FEM community. Essentially, one GPU contains thousands of cores, which are parallel and thus can process large blocks of data very efficiently. Figure 2 is a schematic of GPU architecture and memory access. Thread blocks can be executed in parallel or in series since they are independent. A kernel is called by CPU and executed on GPU multiprocessors in the form of grids. A grid is composed of many thread blocks. Threads within a block can cooperate by sharing data through shared memory and by synchronizing their execution to coordinate memory accesses.

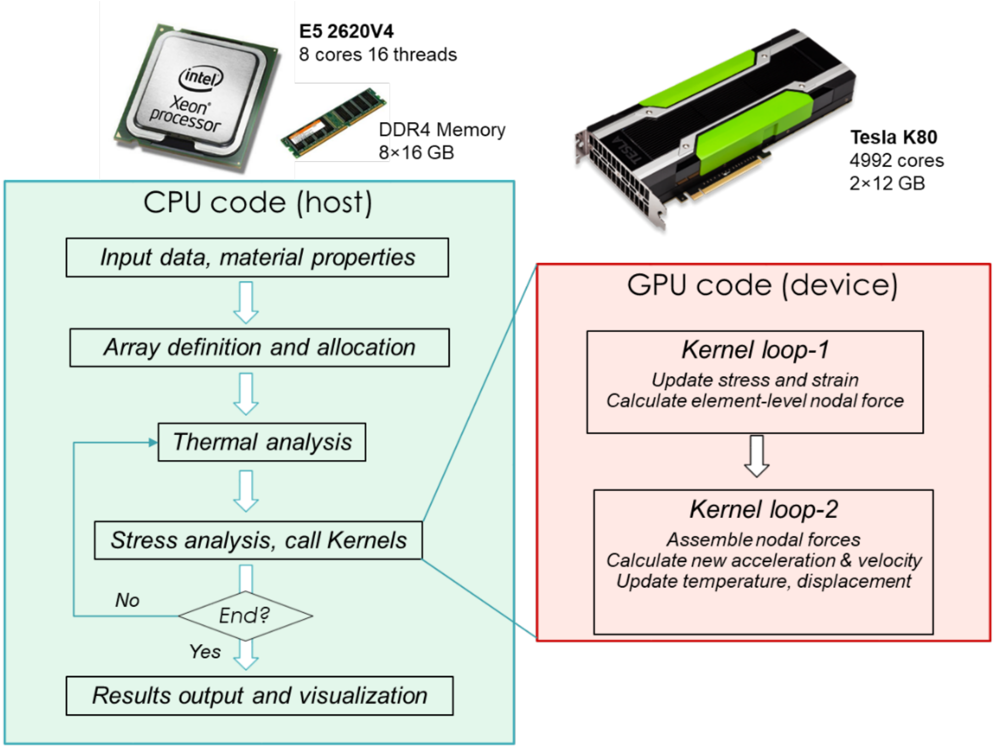


Figure 2. Code Architecture of ORNL Fast Solver

Hexagon Software (Simufact Welding)

Hexagon Simufact Welding is used to model and optimize a variety of thermal joining processes, considering the welding sequence and fixture. One of the biggest challenges in welding is the thermal deformations or residual stresses of the entire assembly that occur after a non-optimized welding and clamping process. As a result, subsequent assembly processes cannot be easily performed because the geometry of the sub-assembly is out of tolerance. With Simufact Welding, the welding process and parameters can be optimized to achieve the desired product quality. Compared with other commercial software, the advantages of Simufact Welding are:

- Simplicity, designed to fit the needs of process engineers rather than CAE specialists. The excellent usability enables a significant reduction of modeling times.
- Semi-automated meshing, simply set the element size and create refinement boxes for particularly important areas if needed, click "Create Mesh" and the meshing will be performed automatically.
- Process oriented, all-in-one solution. Not only easy to use due to its process-oriented and intuitive user interface but also includes comprehensive functionality for all simulation steps from modeling to calculation and the evaluation of results.

Objective

The objectives of this project are to develop a user-friendly and fast-analysis software for welding-process simulation to predict distortion on ship structures including panel structures and complex unit structures. The software includes a graphical user interface (GUI) and a fast-analysis analysis solver. The GUI will be able to directly access geometric, material, and weld information from ship design software, ShipConstructor, automatically create finite element mesh, launch analysis, and post-process the results. The solver will be able to analyze large ship structures meshed with solid elements, shell elements or combined solid-shell elements at a fast speed by taking advantage of modern GPU-based HPC hardware and using an explicit FE method. Utilizing the total software package developed in this project will significantly decrease the modeling time required, drastically increase the number of models that can be analyzed and implemented for subsequent production cost savings, and allow less technical users to perform these analyses.

Fast Mechanical Analysis Solver (DR-Weld)

The project started with the evaluation of the ORNL solver, Dr-Weld, on a representative ship panel to identify the technical gaps for shipbuilding applications and develop software requirements. Then, ORNL developed a new version of solver for shipbuilding applications. The developed solver was tested on increasing complex structures (one Tee, one cell, multiple cells, and panel). All the tested models were created manually. To automate the modeling process, Hexagon developed a preliminary version of automatic mesh generation, and a capability of transferring designed solid models from Shipbuilding CAD software, ShipConstructor, to Hexagon's software package.

Evaluation of the ORNL Solver

Figure 3 shows a finite element model for a simple ship panel which includes both longitudinal and transverse stiffeners welded to a plate. The model consisted of 1.6 million nodes and 1.2 million 8-node solid brick elements. The model was created with Hyperworks and converted into the inputs with Python scripts required by the ORNL solver. The ORNL solver was able to run on a laptop equipped with Intel(R) Core (TM) i7-4700MQ CPU @ 2.40GHz and 24 GB memory.

However, the ORNL solver initially had many limitations compared with commercially available finite element software such as Abaqus. The solver did not include 6-node solid elements, which has to be used in meshing the complex geometries. In addition, shell elements are generally used for thin large ship structures, but the ORNL solver did not have the shell element formation.

Furthermore, the ORNL solver did not have the pre- and post-processing capability to prepare the inputs and process the analysis results. Since it takes time to develop a graphical user interface, Python scripts were developed for pre-processing and an open-source code (Paraview) was used for post processing of analysis results.

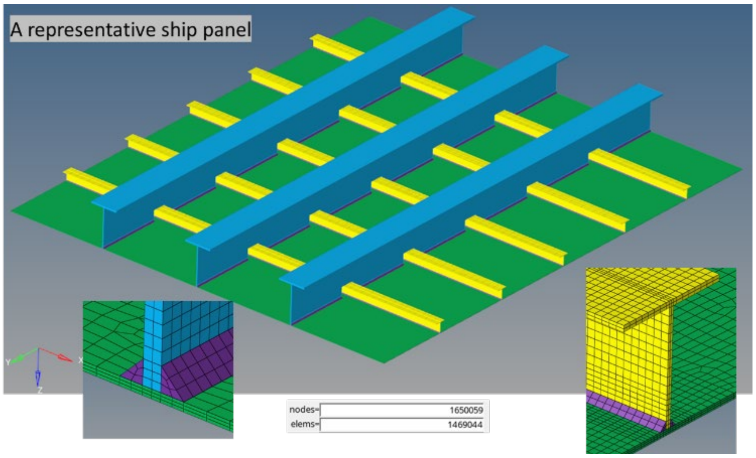


Figure 3. A Representative ship panel to evaluate the ORNL Fast Solver

Identification of Technical Gaps

Based on the evaluation outlined above, the technical gaps were summarized as follows. A fast-solver development plan was generated to overcome the current limitations for applications specific to this project.

- The solver could only run eight-node solid element. Six-node (triangular) solid elements are needed to simulate complex ship structures.

- The solver did not include shell elements which are mainly used in the design of ship structures.
- The solver did not have a user-friendly graphical user interface for pre- and post-processing.
- The solver did not have a thermal solver to predict temperature.

Selection of Shell Elements for Modeling Ship Structures

Thin structures are increasingly used in shipbuilding to reduce the weight of a ship and increase its mobility to allow for more advanced weapon capabilities and increasingly complex (and heavy) mechanical systems. Shell elements are best suited for FEA of structural design evaluation and most efficient option for welding simulation of ship structures.

However, shell elements will not be able to model the detailed geometry in the welded joints. Significant developments were made to develop the shell-model techniques to predict distortion using shell elements in Abaqus (Yang 2000). Later, other commercial finite element software such as Marc (Olsen 2001), and LS-Dyna (Schill 2021) have been used for welding simulation.

To gain confidence using shell elements for welding simulation in ship structures, the project team conducted a study using both a solid-element model and a shell-element model to analyze the most used welded joint (double-sided fillet welds) in shipbuilding to compare the results.

Figure 4a shows a double-sided fillet weld which is typically used to join stiffeners to a plate to form a ship panel and a bulkhead. The exact weld shape can be built in a solid-element model (Figure 4b) while a much-simplified representation is built in a shell-element model (Figure 4c). Transient thermal elastic-plastic analyses were conducted with the Goldak's double ellipsoidal heat source model (Goldak 1984) to predict distortion on both the solid model and the shell model using Abaqus.

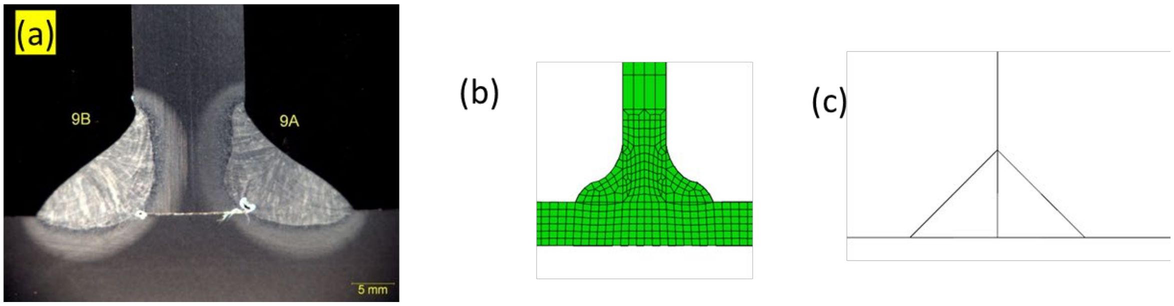


Figure 4. A fillet weld (a) weld cross section (b) Solid-element mesh (c) shell-element mesh

Figure 5 shows the predicted distortion on the solid and shell model on the Tee joint in which the plate dimensions have a width of 24 inches (609.6mm), length (along the welding direction) 12 inches

(304.8mm) and thickness 8mm and the stiffener was WT205x23. A similar distortion shape was predicted on both the solid model and the shell model. The average angular distortion was 8.4mm for the solid model, which was slightly higher than the experimental measurement 7.6mm. The average angular distortion on the shell model was 6mm which is smaller than the experimental measurement. The solid model is more accurate than the shell model. However, the solid model took much longer to run than the shell model (about 70 minutes for the solid model and 13 minutes for shell model).

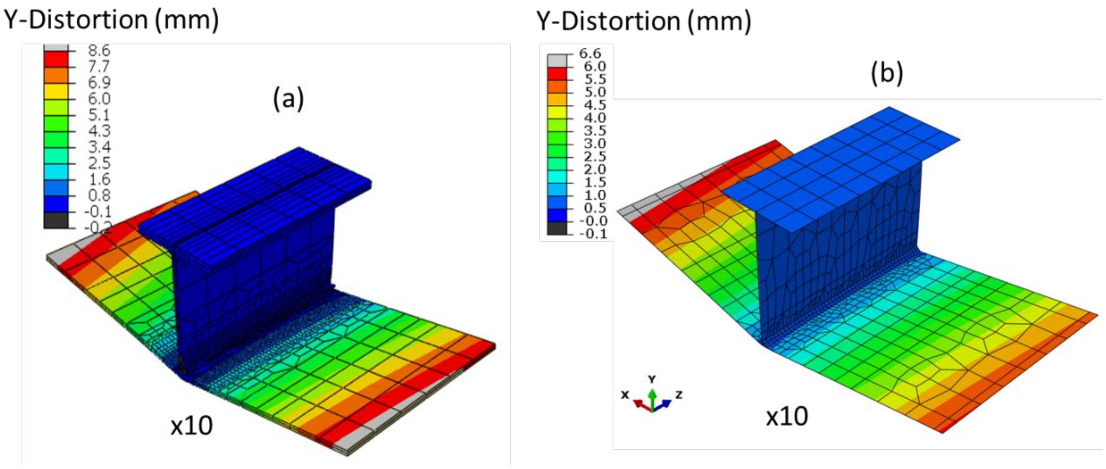


Figure 5. Predicted distortion (a) solid model (b) shell model

In deliberation about which direction to pursue with the fast-analysis solver development, the shell-element model was selected by carefully considering the trade-off of accuracy and analysis speed. Due to modeling applications currently prioritized by shipbuilding and the inexact nature of ship construction relative to other industries, very precise distortion prediction does not change the effectiveness of the modeling results to improve quality and reduce production labor cost. Additionally, the accuracy improvements solid element analysis provide are minimal over the size and scale of ship panels and full unit assemblies. The shell model is much easier to generate, and the computational time is much shorter than the solid model. In addition, the shell element mesh for structural design can be modified for welding simulation to further reduce the model setup time.

Develop a Shell-Element Based Solver

Th development of the shell elements was based on Reissner-Mindlin Plate Theory [Reissner, 1945; Mindlin, 1951]. The formula includes the combination of the following three components.

- In-plane tress deformation,

$$\begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{bmatrix}=\begin{bmatrix}\frac{\partial N}{\partial x} & 0\\ 0 & \frac{\partial N}{\partial x}\\ \frac{\partial N}{\partial y} & \frac{\partial N}{\partial x}\end{bmatrix}\begin{bmatrix}u\\ v\end{bmatrix}$$

- bending

$$\begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{bmatrix}=\begin{bmatrix}0 & 0 & z*\frac{\partial N}{\partial x}\\ 0 & -z*\frac{\partial N}{\partial y} & 0\\ 0 & -z*\frac{\partial N}{\partial x} & z*\frac{\partial N}{\partial y}\end{bmatrix}\begin{bmatrix}w\\ \theta_x\\ \theta_y\end{bmatrix}$$

- transverse shear deformation.

$$\begin{bmatrix}\gamma_{yz}\\ \gamma_{zx}\end{bmatrix}=\begin{bmatrix}\frac{\partial N}{\partial x} & -N & 0\\ \frac{\partial N}{\partial y} & 0 & -N\end{bmatrix}\begin{bmatrix}w\\ \theta_x\\ \theta_y\end{bmatrix}$$

Where {*x, y, z*} represents the 3D coordinate with *z* normal to the plate surface. { $\varepsilon_x, \varepsilon_y, \gamma_{xy}$} are in-plane strain components and { γ_{yz} , γ_{zx}} are transverse shear strain components. {*u, v, w*} are translational displacement along {*x, y, z*} direction and {θ_x, θ_y} are rotational degree of freedoms along {*x, y*}. N represents the shape function.

To allow the discretization of complex panel structures, 4-node and 3-node shell elements with full-integration and reduced integration were formulated. In addition, multiple through-thickness integration points (with Simpson's rule) can be defined, as shown in Fig. 6.

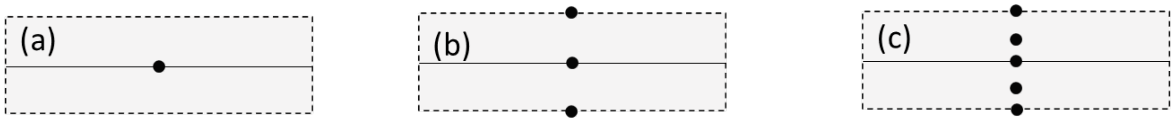


Figure 6. Through thickness integration points (a) one (b) three (c) five

A flexible computational framework was designed by ORNL to accelerate welding simulations while maintaining predictive fidelity. Dr. Weld combines implicit and explicit mechanical solvers to significantly reduce computational time in the mechanical portion of the analysis. Within this framework, ORNL has established two complementary approaches for distortion and residual-stress welding simulations:

1. **Transient Analysis Approach**: This approach explicitly simulates the motion of each weld pass, capturing the detailed thermal–mechanical history of the welding process. It is well suited for transient analyses and has been applied to weld sequence optimization, enabling the evaluation of multiple welding paths with improved computational efficiency. In the transient thermo-mechanical approach, a precomputed temperature history is applied within a sequential thermal–mechanical analysis that is enhanced by a patented acceleration algorithm and massively parallel CPU/GPU computing. This combination enables efficient time-marching of the mechanical response for large-scale welded structures, overcoming the computational bottlenecks typically associated with transient simulations and allowing practical application of high-fidelity distortion and residual stress prediction at industrial scale.
2. **Strain-Based Approach**: In contrast, the strain-based approach employs a simplified formulation in which thermally induced strains are used as input to drive the mechanical response, eliminating the need for coupled transient thermal-mechanical simulations. This method allows for rapid estimation of weld-induced distortions, making it particularly attractive for large-scale components or design screening studies. In the thermal-strain-based approach, the welding process is represented by a compact equivalent thermal strain formulation that is directly imposed on the structural finite element model. The approach is designed for seamless integration with Ingall's existing thermal-strain-based software framework, implemented as a dynamically linked library (DLL). This modular design supports straightforward adoption within established workflows while preserving computational efficiency, making it well suited for rapid distortion assessment of large ship and offshore structures.

The shell version of DR-Weld code was written by Visual Studio C++ with OpenMP to enable parallel computing with multiple CPU threads and massive parallelization with GPU to gain additional speed-up.

Solver Implementation and Execution

The solver is implemented for the Windows operating system and supports multiple execution configurations depending on the analysis approach. The transient thermo-mechanical solver supports execution on single-CPU, multi-CPU, and GPU platforms, enabling scalable performance for large-scale simulations. The thermal-strain-based solver supports single-CPU and multi-CPU execution. High-performance linear algebra operations are accelerated using the Intel oneAPI Math Kernel Library (MKL), a free and open-source library that must be installed and accessible in the runtime environment.

Transient thermo-mechanical approach

The executable of the transient thermo-mechanical approach is “DR-Weld-transient.exe”. It requires a mesh file as input and optionally accepts parameters to control CPU and GPU usage. The required argument mesh=<mesh_file> specifies the path to the mesh data file to be processed. Users can optionally specify the number of CPU threads by including cpus=<num_cpus>; if omitted, the program will use a default value of 1 . Additionally, the gpu flag can be included to enable GPU acceleration. Arguments enclosed in square brackets [ ] are optional, while required arguments must always be provided. The general usage pattern is:

```
DR-Weld-transient.exe mesh=<mesh_file> [cpus=<num_cpus>] [gpu]
```

Thermal-strain-based approach

The thermal-strain-based approach can be used in two different ways. The first is similar to the transient thermo-mechanical approach and runs as an executable named “DR-Weld-strain.exe”. The executable of the strain-based approach also processes a mesh file and optionally allows control over the number of CPU threads. The required argument mesh=<mesh_file> specifies the path to the mesh data file to be processed. Users can optionally provide cpus=<num_cpus> to set the number of CPU threads. This executable does not include a GPU option.

```
DR-Weld-strain.exe mesh=<mesh_file> [cpus=<num_cpus>]
```

In addition, a library version of the thermal-strain-based solver is also available, which can be linked and integrated into Ingall’s existing software framework.

Validation Testing

Transient thermo-mechanical approach

The transient thermos-mechanical approach was tested in 4 steps (Figure 7) to confirm its prediction accuracy and efficiency. Step 1 analyzed a simple Tee joint with double-sized fillet-weld model with the newly developed shell-element code. The predicted distortion was compared with the measurement data and Abaqus shell-model prediction for accuracy verification. Step 2 analyzed one robotic-cell model (including 4 fillet welds). Step 3 analyzed a multiple-cell model. The predicted distortion of the ORNL solver was compared with Abaqus since no experiment results were available and Abaqus has been extensively used with validated accuracy in shipbuilding specific applications. In Step 4, a full ship panel which included about 150 welding cells was analyzed using the ORNL shell-element code. All analyses in the four steps were conducted with a transient thermal elastic-plastic solution.

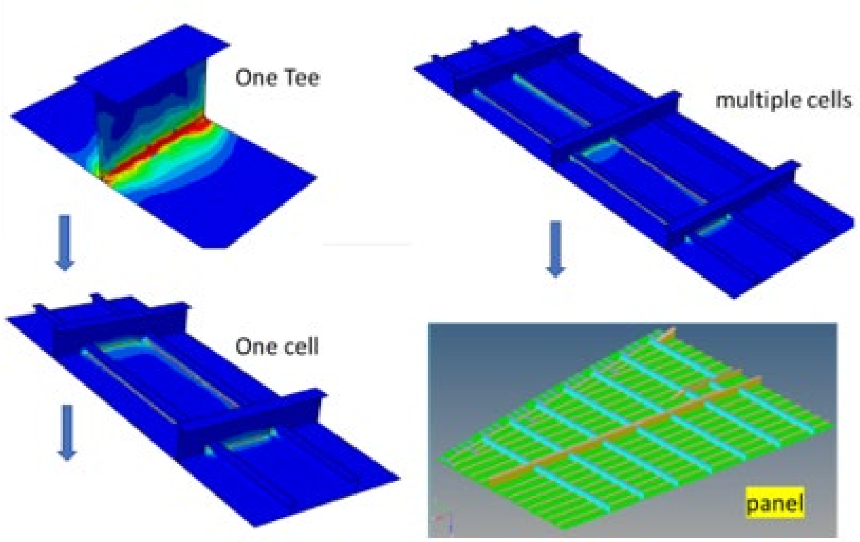


Figure 7. Technical approach to develop the fast-analysis solver with shell-element model

One-Tee Model

Beginning with the simplest test case (one-tee joint with two fillet welds), discrepancies between the ORNL solver and the Abaqus results were found. Abaqus simulated the welding process using implicit static analysis. The ORNL solver could not accurately predict the angular distortion due to the dynamic effect induced by the time scaling in explicit analysis, as shown in Fig. 8. This dynamic effect might be negligible when modeling solid, or bulk structures in the nuclear industry or even for small thin-panel structure in automotive industry because the welded structures are small and rigid. On the other hand, a ship structure is large and more flexible so the dynamic effects become more profound and can’t be ignored in accelerated simulations.

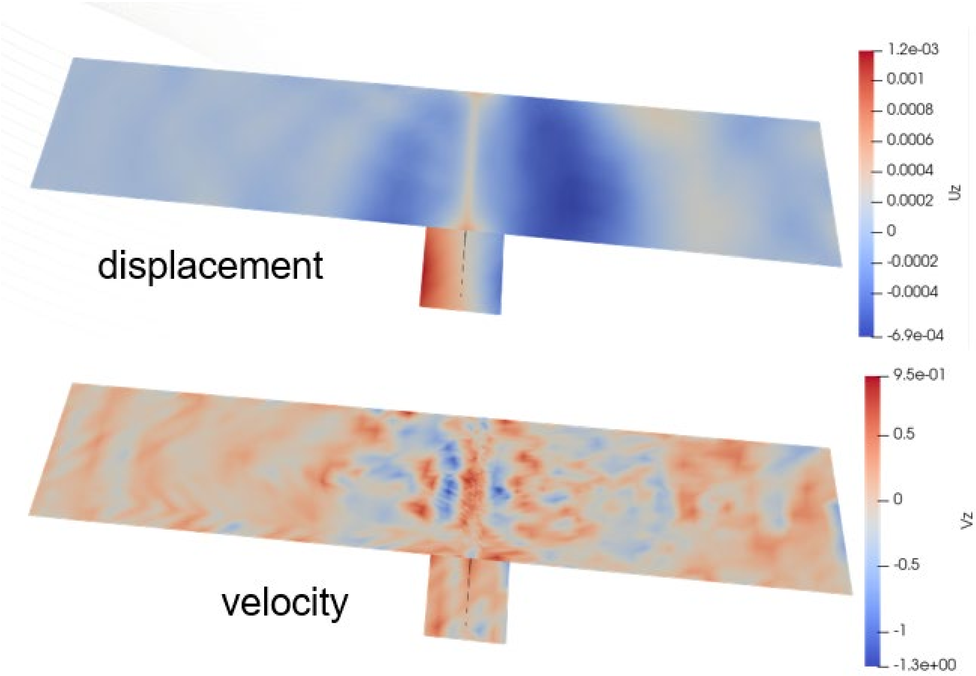


Figure 8. Dynamic effect in the explicit analysis of ORNL solver

Welding can generally be considered as mechanically static problem where the inertial effect is very small. Modeling the process with an explicit analysis and a large time scaling factor will result in the dynamic structure reaction. Thus, the angular distortion after cooling cannot be predicted accurately. To address this issue, ORNL developed a hybrid explicit and implicit analysis method in which the heating is mainly simulated with an explicit analysis and the cooling is modeled with the implicit analysis. This hybrid method makes sense since welding time is much shorter than the cooling time. The time scaling factor in the explicit analysis to simulate welding does not need to be very large, which can also reduce the dynamic effect. An implicit solver is better suited for distortion prediction during cooling to account for the slow cooling rate of the plate material.

The hybrid explicit-implicit shell-element solver developed by ORNL is capable of predicting the angular distortion, as shown in Figure 9. The predicted distortion shape is similar to the Abaqus prediction (see Fig. 5b). The distortion magnitude is 1mm smaller than the Abaqus prediction. Since the developed fast solver will mainly be used to compare the relative difference in welding sequence analysis or fixture/tooling evaluation, the project team decided to move on with testing the one-cell model. Nevertheless, fine-tuning the hybrid approach has been on-going to improve its accuracy, although this is deemed to be a low priority in Phase I of the project.

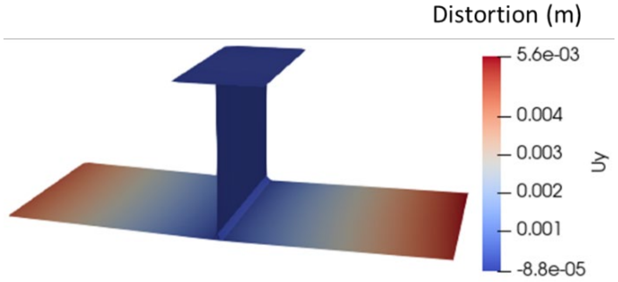


Figure 9. Predicted distortion on one-Tee model using the modified ORNL solver

One-Cell and Multi-Cell Model

The name “cell” refers to the robotic welding process on a ship panel, as shown in Fig. 10. To save production time, a robot welds the four tees to deck connections enclosed by stiffeners and then moves to the next cell to continue welding. Currently, the welding sequence used considers only robotic path movement to minimize welding time. The welding sequence cannot be optimized to minimize distortion because of the current modeling capability. After successful completion of this project, the welding sequence would be optimized using the developed fast-analysis solver, considering both distortion and weld processing speed.

Figures 11 and 12 show the predicted distortion comparison between the new ORNL hybrid shell-element solver and Abaqus. The results predicted with the ORNL solver were plotted using an open-source code called Paraview. Although the color is different between the two predictions, the predicted distortion distributions are essentially the same. The ORNL solver predicted distortion magnitude is slightly smaller than that predicted by Abaqus. The difference on the multiple-cell model shows closer comparison to the Abaqus model than the one-cell model because the multiple-cell model has more resistance to the dynamic effect as discussed in the one-Tee model.

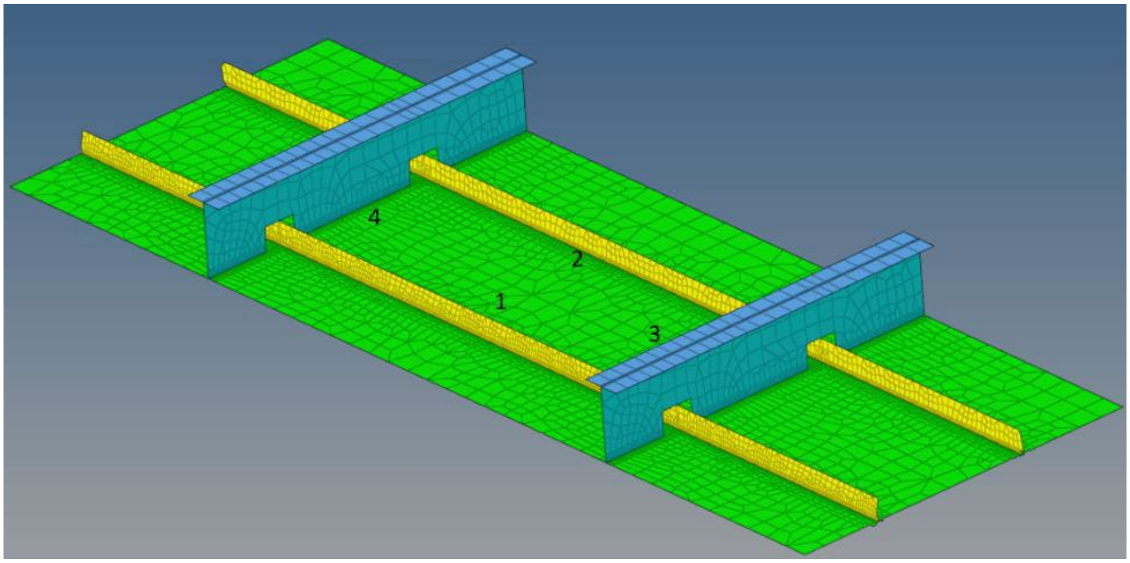


Figure 10. Typical Welding Cell Mesh Configuration

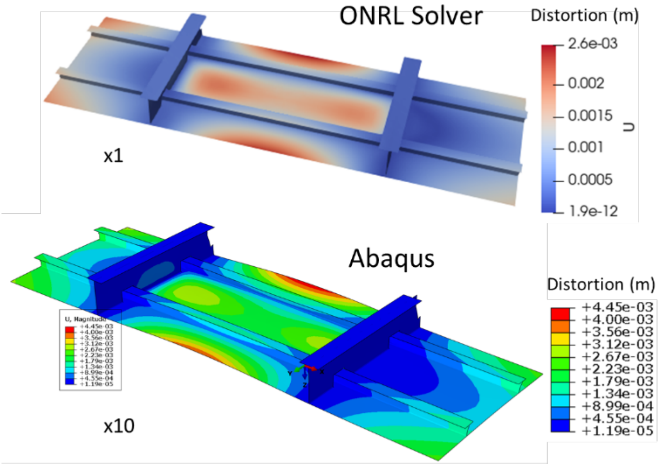


Figure 11. One-cell model: distortion comparison between the ORNL solver and Abaqus

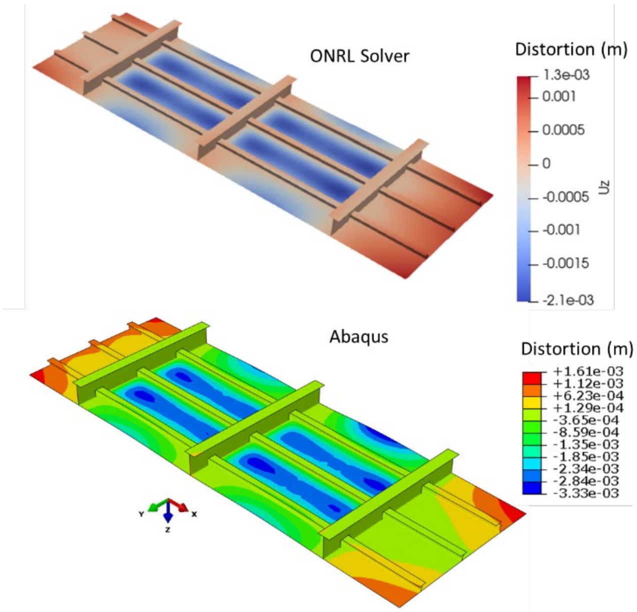
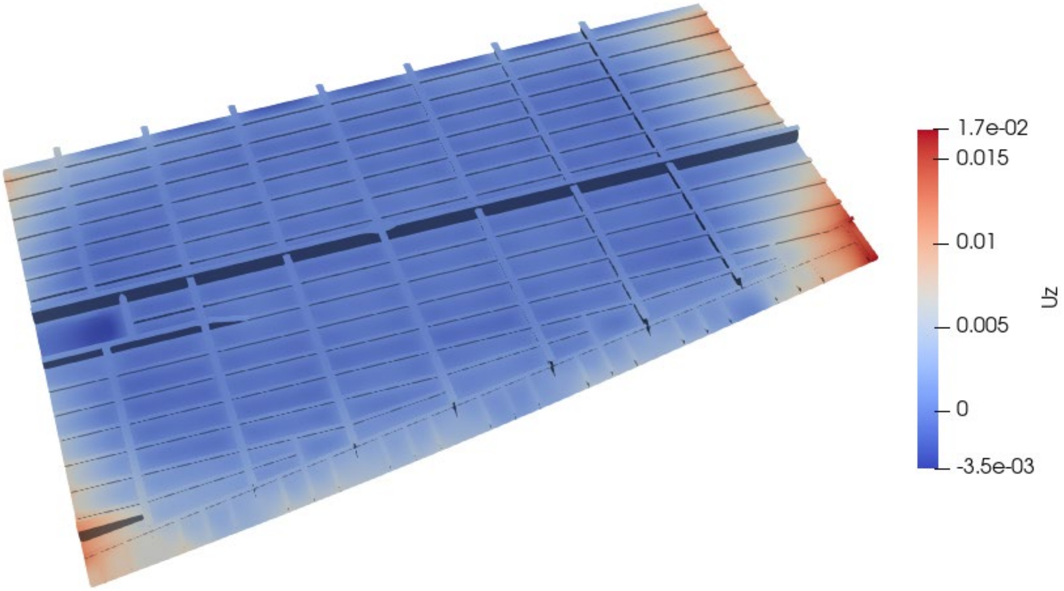


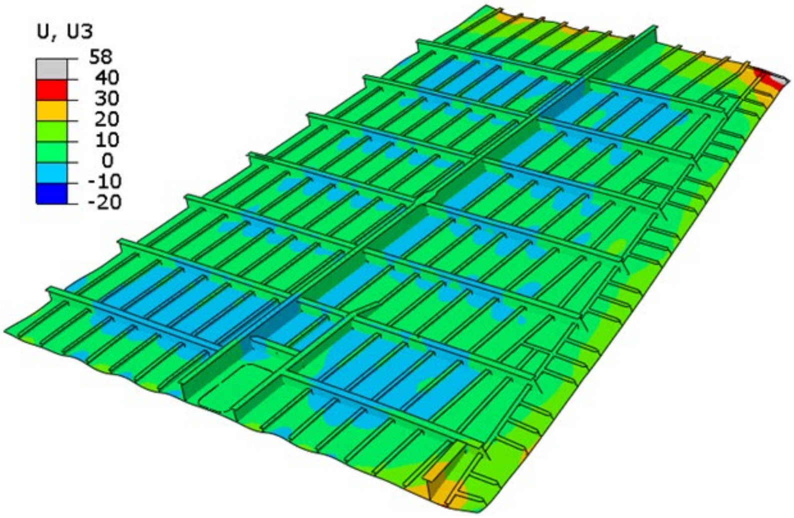
Figure 12. Multiple-cell model: distortion comparison between the ORNL solver and Abaqus

Panel Model

Figure 13 shows the distortion prediction of a full panel consisting of 154 cells, comprising approximately 1,000,000 elements. The panel involved several hundred weld passes, with individual welds ranging from a few meters to tens of meters in length. In actual production, welding this panel structure required several days to complete. Due to the size and complexity of the superstructure, it was not feasible to use commercial simulation tools to optimize welding sequences for distortion control. In contrast, DR-Weld completed the full thermal-elastic-plastic simulation (see Figure 13a) in less than two days on a desktop workstation, demonstrating both exceptional scalability and strong industrial applicability. For comparison purposes, Abaqus inherent strain-based analysis, an approximated distortion prediction method and not good for welding sequence optimization, was conducted to predict distortion (see Figure 13b). DR-Weld prediction (Figure 13a) and Abaqus prediction (Figure 13b) show similar distortion distributions. The predicted distortion magnitudes are different between DR-Weld and Abaqus. The possible reason is that Abaqus used an approximate solution.



(a) DR-Weld Prediction (meter)



(b) Abaqus inherent strain-based solution (approximated prediction, mm)

Figure 13. A comparison of distortion prediction between DR-Weld and Abaqus

Thermal-Strain-Based Approach

The strain-based solver was tested on a curved plate in which a line heating was applied in the middle of the plate, as shown in Figure 14. The plate had a dimension of 10 feet by 10 feet by 5/8’’. The curve radius was about 200’’. DR-Weld input is thermal strain and strain orientation. Dr-Weld output is deformation. Figures 15-17 present the comparison of the displacement fields along the X, Y and Z directions between DR-Weld and Abaqus. The results are reasonably comparable in all directions, indicating that DR-Weld’s strain-based solver can capture the overall thermal-structural response in 3D geometries with curvature and localized heating. This case demonstrates the capability of DR-Weld to predict weld-induced distortions in more practical engineering components.

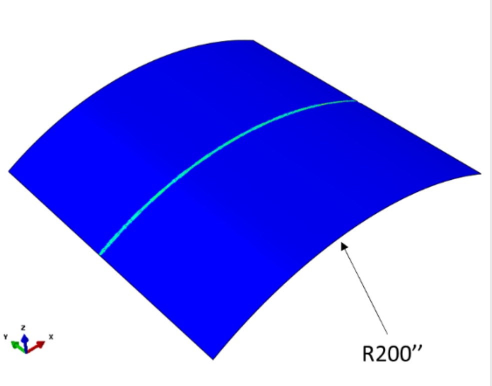


Figure 14. Heating a line in the middle of a curved plate

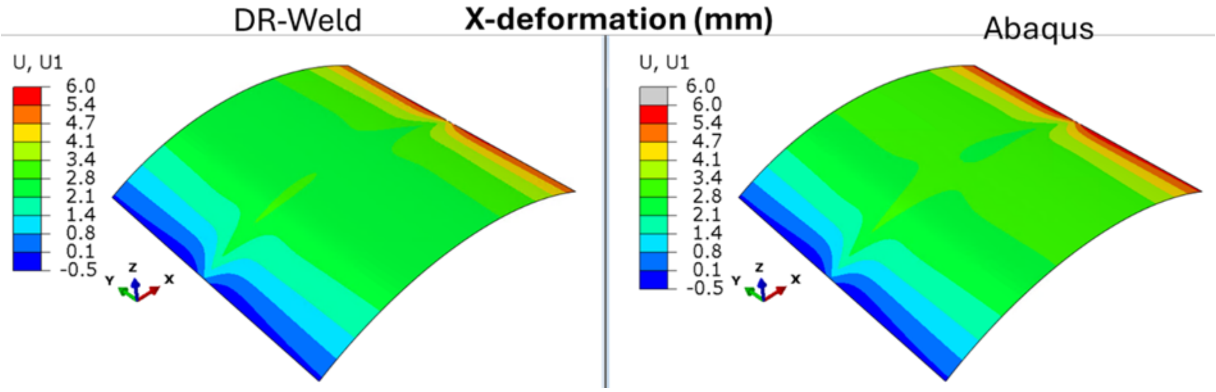


Figure 15. A comparison of X-deformation prediction between DR-Weld and Abaqus

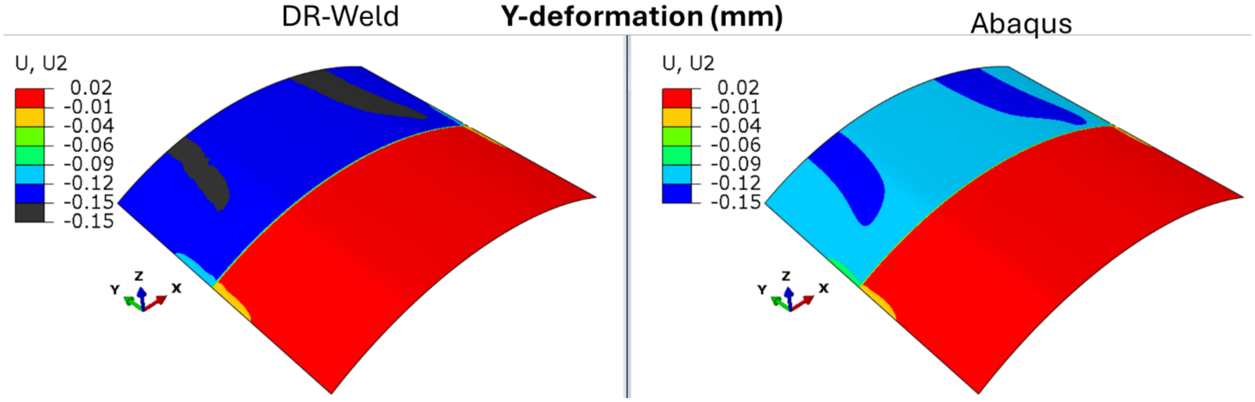


Figure 16. A comparison of Y-deformation prediction between DR-Weld and Abaqus

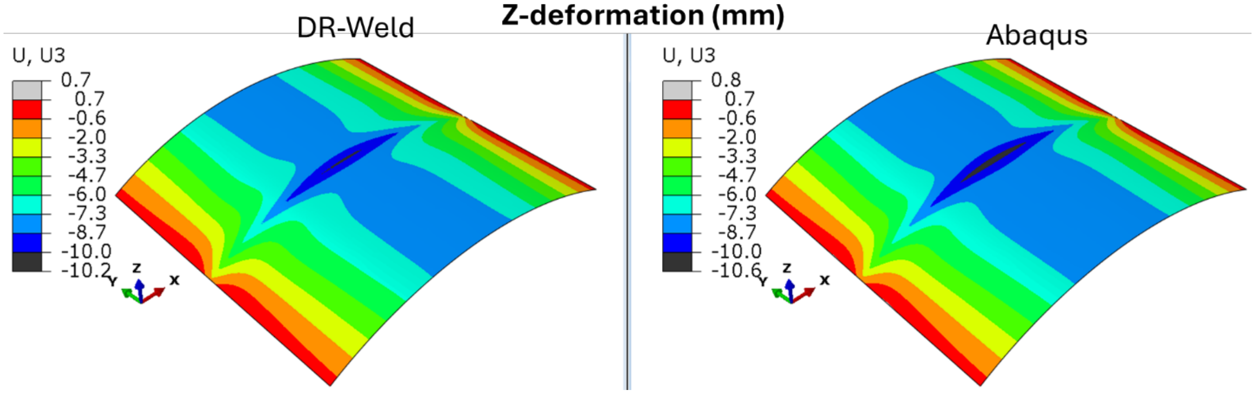


Figure 17. A comparison of Z-deformation prediction between DR-Weld and Abaqus

Fast Thermal-Analysis Solver

One of the technical gaps identified during evaluation of the ORNL solver is that the ORNL solver does not include a thermal-analysis solver. Hexagon Simufact Welding also does not have a thermal-analysis solver for shell elements. In this task, Hexagon has developed the thermal-analysis solver for shell elements to predict temperatures. The predicted temperature will directly input to the ORNL solver Dr-Weld for mechanical analysis to predict distortion.

Finite-Element Based Approach

The shell-element model was not supported in the Simufact Welding environment. In the project Phase 1, a prototype of thermal-analysis solver was implemented in MSC Marc Mentat, which is compatible with Simufact Welding. The thermal analysis solver was developed on the one-Tee model, as shown in

Figure 18. Both 3D solid model (see Figure 18c) and shell-element model (see Figure 18d) were analyzed to valid the developed thermal-analysis solver.

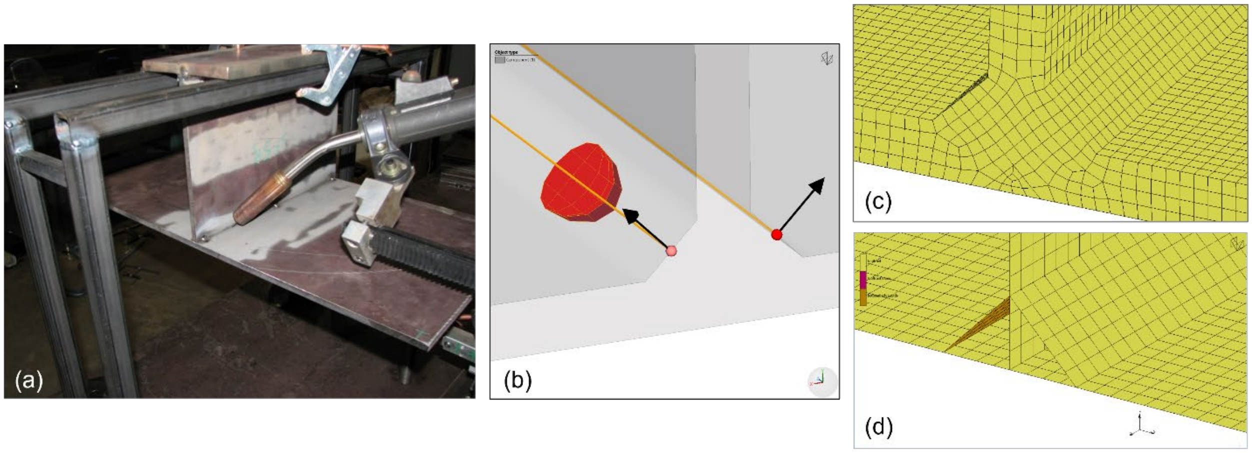


Figure 18. (a) Image of the welding process. (b) Definition of weld lines in Simufact Welding. (c) Representation of the weld joint via 3D solid mesh. (d) Representation of the weld joint via the shell mesh.

Validation of 3D Solid Models

In this task, 3D solid models were validated using both Simufact Welding and MSC Marc. Figure 19 shows the y displacement of the structure after final cooling. Both models come very close to the experimental results with an underestimation of the displacement on the side of the first weld bead and an overestimation on the side of the second weld bead.

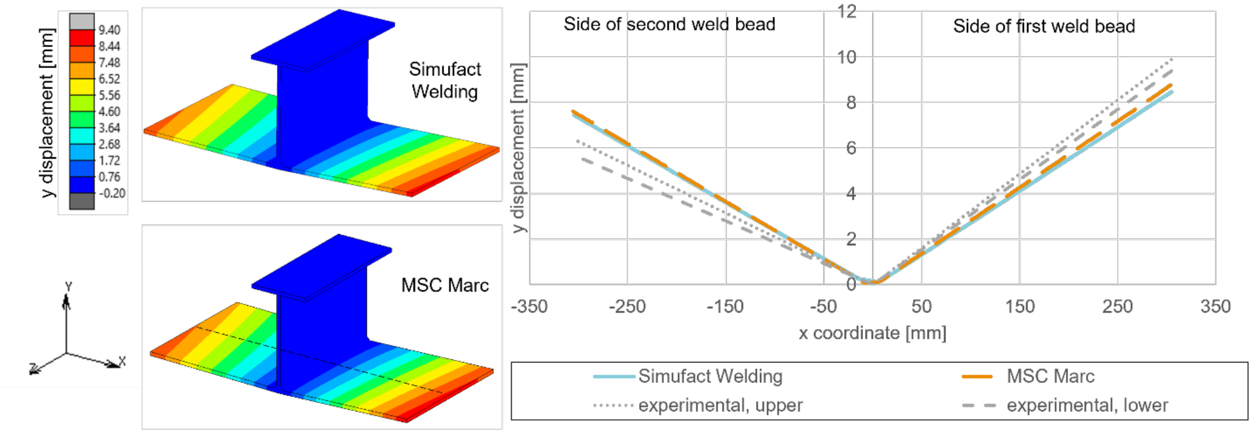


Figure 19. Validation of the volume models for Simufact Welding and MSC Marc via the y deformation after final cooling. The diagram shows the y deformation at the bottom of the base plate as indicated by the dashed line.

Shell Models in MSC Marc and Simufact Welding

Shell models were set up with the mesh illustrated in Figure 18d and all boundary conditions were directly transferred from the corresponding volume reference. As the diagonal weld bead elements are likely to affect the thermal profile, the MSC Marc shell model were validated against the volume reference via a short weld segment on the top plate. Figure 20 shows that both shell model and volume reference give very similar temperature profiles at the top and bottom of the plate which qualifies the applied approach.

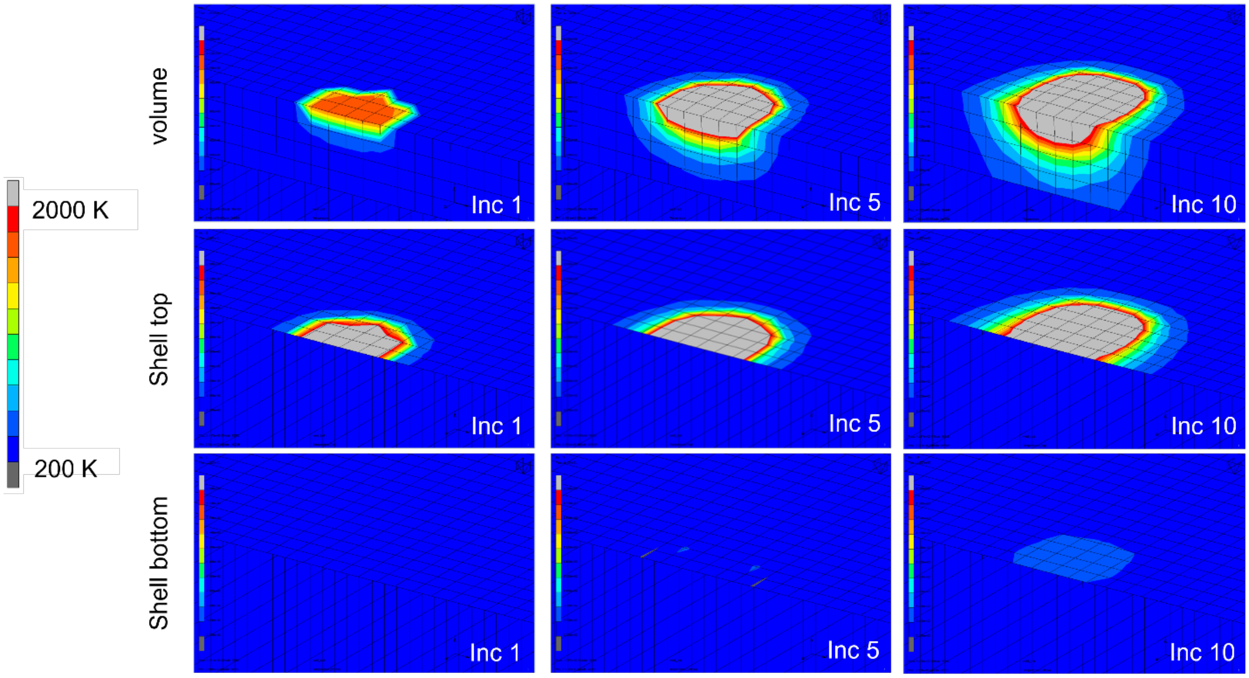


Figure 20. Temperature profile for volume and shell model for a short weld line at the top of the plate

Likewise, Figure 21 illustrates that the shell model of the complete weld line results in a very similar peak temperature as the volume (3D solid) reference and that melting temperature is achieved along the entire weld line.

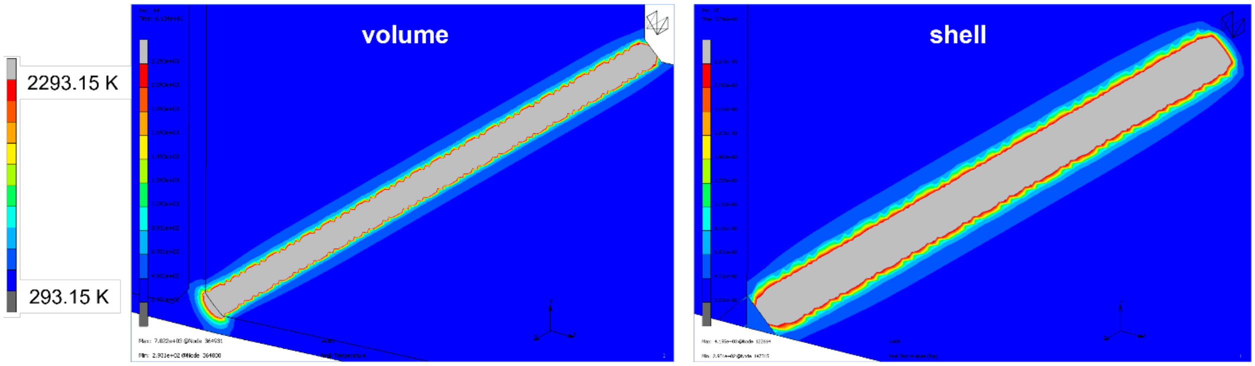


Figure 21. Peak temperature after completion of the first weld bead both for volume (3D solid) reference and MSC Marc shell model

Finally, it was demonstrated that shell models can be run within the framework of the Simufact Welding solver input file. To this end, manual modification of the input deck was done to implement the shell mesh and make necessary changes to the boundary conditions, e.g. change the node IDs in set definitions. As illustrated in Figure 22, a visualization of the simulation results of the shell top side is possible in Simufact Welding while the shell bottom remains blank. A full visualization of these results is still accessible via MSC Marc Mentat.

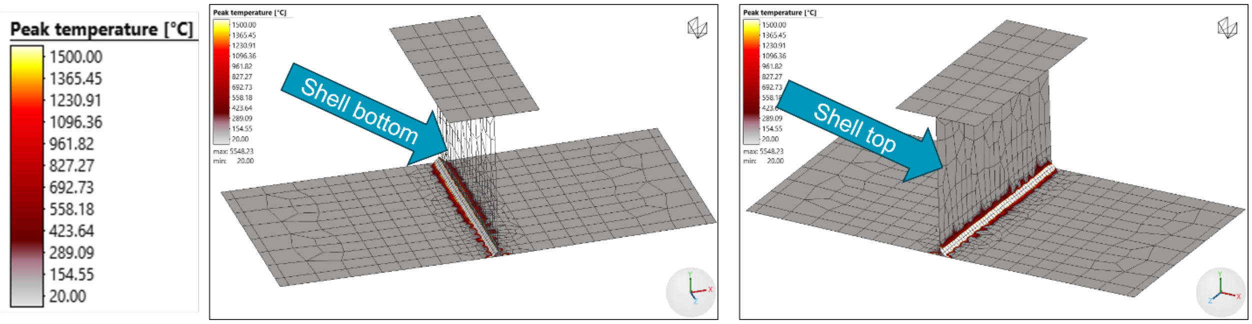


Figure 22. Visualization of shell welding results within Simufact Welding

Distance Based Approach

The finite element based thermal analysis solver is still taking hours to run. To further reduce the computational time of thermal analysis, a distance-based approach was developed by Hexagon by leveraging simplified mathematical equations, akin to the well-known Rosenthal's equation, to efficiently model the temperature decay and heat distribution in and around the weld pool. This allows for a rapid yet accurate approximation of thermal history, paving the way for significantly faster and more accessible welding simulations. This approach does have limitations but should be appropriate for the geometries in this project. Additional description of limitations can be provided on request.

This approach assumes three main regions during the weld phase:

1. Regions away from weld: room temperature is assumed.
2. Regions within weld bead and fusion zone: temperature is defined as an input.
3. Regions between items 1 and 2 (“heat affected zone”): an exponential decay is defined.

The distance considered for item 1 is open to the user. In this example, it was defined as 100 mm. For the fusion zone, the bead size is assumed as 5 mm, but this can again be changed by the user and is directly related to the size of the weld bead itself. For the exponential decay, a simple law was defined according to the equation below:

$$T_{local} = T_{min} + (T_{max} - T_{min}) \cdot \exp\left(a_1 \cdot (d - l)\right)$$

T_{local} is the local temperature, T_{min} is the minimum (room) temperature, T_{max} is the maximum temperature, a_1 is the exponential decay parameter, d is the Euclidian distance from a given node, and l is the weld bead size (leg).

The decay amount has been manually adjusted in the validation phase of the equation to make sure the displacement field produced by this approach was similar to one produced by the full-transient thermal-mechanical analysis from Simufact (the validation results are shown later in the report). However, this value can be changed according to user preference.

The distance in the equation is calculated per node, in relation to the current heat source location. The distance is calculated simply as the absolute distance between the nodal coordinate and the current trajectory position. The current trajectory position can be found using the current time, and the weld start and end times:

$$\alpha = (t - t_0)/(t_f - t_0)$$

$$P = L_0 + \alpha \cdot (L_f - L_0)$$

In which t is the current time, t_0 is the trajectory start time, t_f is the trajectory end time, P is the trajectory current coordinate, L_0 is the trajectory start coordinate, and L_f is the trajectory end coordinate.

By calculating the distances for each node, per increment, it is possible to calculate the temperatures for each node, for each time step.

Temperature Validation Through Subroutine Implementation

To evaluate and validate the approach, the analytical thermal approach was implemented on a Fortran subroutine and ran with the standard thermal-mechanical solver from Simufact. The temperature and displacement fields were then compared to the standard heat transfer based, Simufact/Marc thermal-mechanical solver. The subroutine used is the FORCDT in Marc, that combines a “FIXED TEMPERATURE” model definition with the subroutine temperature input.

Briefly, the subroutine follows the steps below:

1. Define weld trajectories start and end times;
2. Define weld trajectories start and end coordinates (XYZ);
3. For each time step and for each node:
    a. If time is between a given trajectory start and end time, that is the active trajectory;

b. Calculate the “alpha” value that represents the ratio between start and end of the trajectory for the current time step;
c. Calculate current position of the heat source based on the alpha value;
d. Calculate the distance between the node and the current heat source position:
    i. If the distance is smaller than the defined bead size, set as maximum temperature;
    ii. If the distance is larger than the considered room temperature “cutoff” region, define as room temperature;
    iii. Else, define temperature according to exponential decay.

This is a simple but effective way of representing the transient behavior of the heat source, while still maintaining good accuracy in terms of the temperature fields, as it is shown in the validation cases.

Feature Updates and Improvements

In the latter part of Phase-II the solver was improved in some ways, with changes done to solid-based models, while a new script was derived from the original one to support shell-based models. The two approaches (solid and shell-based solvers) are intrinsically different and thus require two different scripts. Some notable changes from the previous versions were:

- Added capacity to specify temperature calculation “mode” where different logics can be added.
- A Rosenthal approach was added as an optional thermal calculation. Results are promising although limited testing has been done.
- An algorithm for dealing with curved weld trajectories was added.
- Fixed issue when using cooling/pause time between welds.
- Other minor bug fixes.

More importantly, most of the work was directed into integrating the fast thermal solver with DRW, which is explored in a later section (Simufact to Dr. Weld Connection).

Validation Examples

Two models were selected for validation. Both models were created in Simufact Welding Version 2025.1. The first model is a simple “T” joint with two weld beads (see Figure 23a), and the second model (see Figure 23b) is a subsection model with 4 weld beads. The “T” model is fixed on the four corners of the base plate. The subsection model is fixed in the three edges shown in the image. For all validation examples, the distance-based approach was used. The peak temperature and total displacement plots for the full transient, heat transfer based, thermal-mechanical model are shown in Figures 24 and 25 for both models.

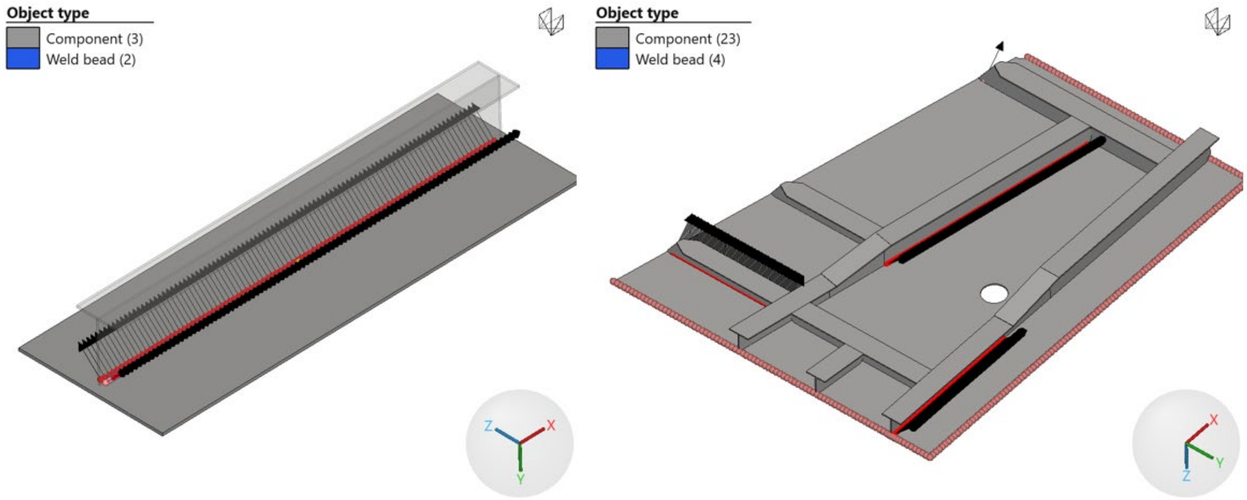


Figure 23. Simufact Welding model view of (a) simple T-joint, (b) representative subsection model

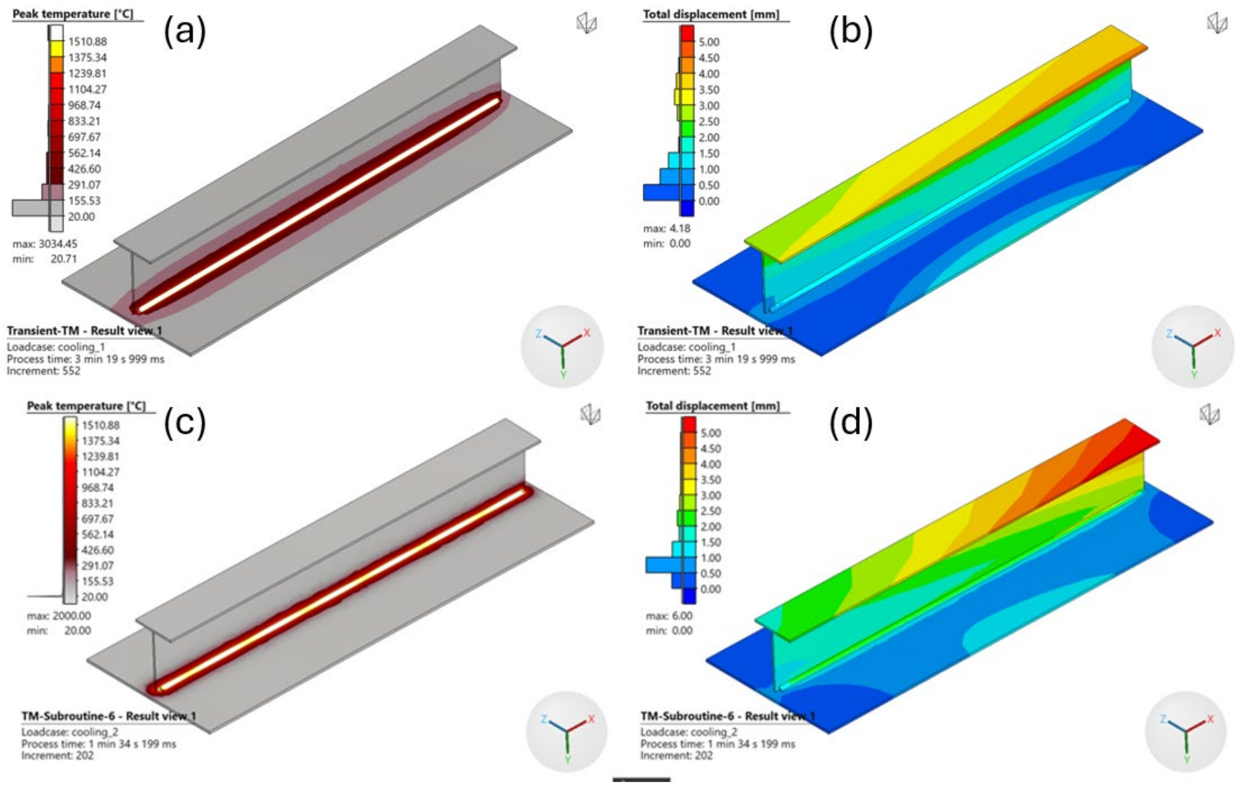


Figure 24. Results for the T-joint model with (a) peak temperature, and (b) deformation, calculated with standard heat transfer models, and (c) peak temperature, and (d) deformation, calculated with new simplified approach

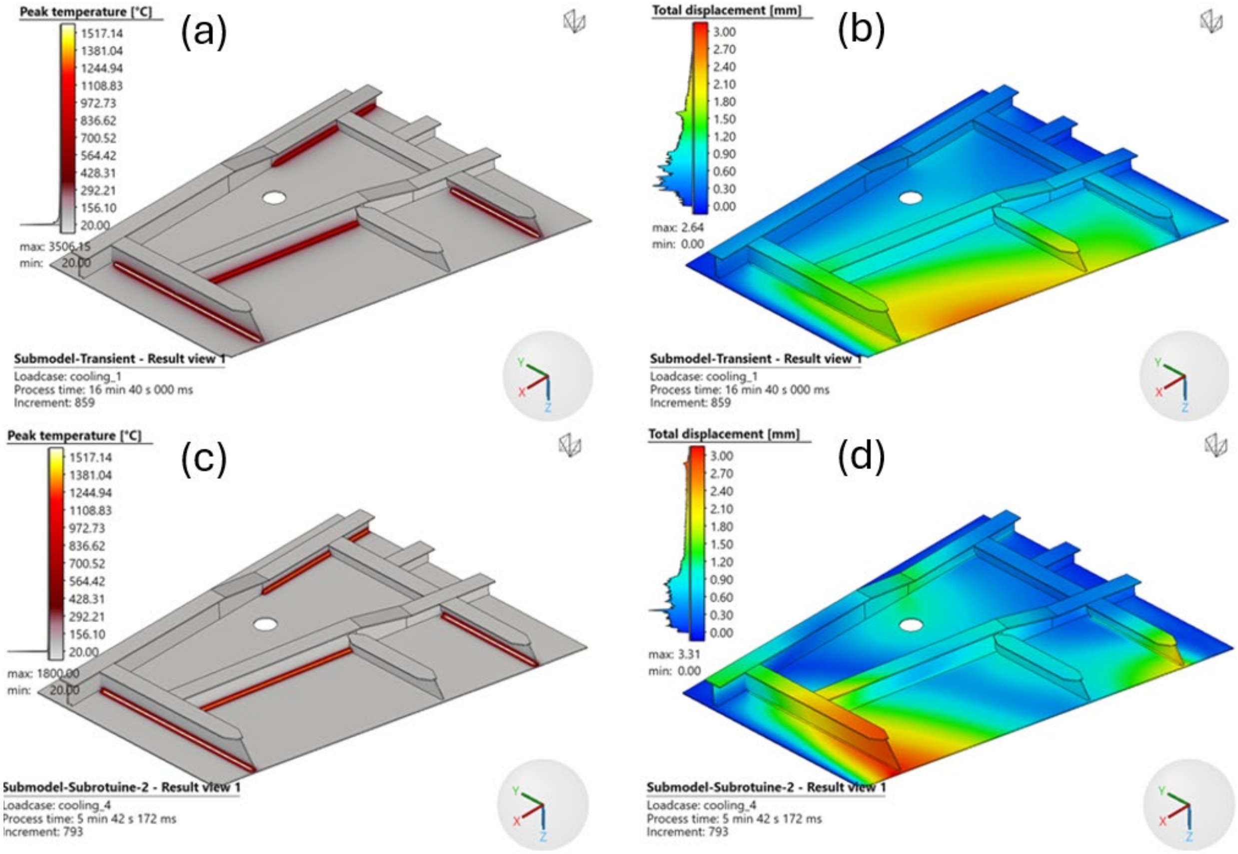


Figure 25. Results for the Subsection model with (a) peak temperature, and (b) deformation, calculated with standard heat transfer models, and (c) peak temperature, and (d) deformation, calculated with new simplified approach.

The general trends and the absolute displacement values are very comparable between the models, although some differences in trends can be observed for the subsection model. This difference is believed to be due to over constraining the model resulting in local buckling, which might drive the distortion to different locations even with slight changes to the model inputs.

Next, the same algorithm is implemented into a Pyhton script, to automate other tasks related to processing the model, generating DR. Weld input files, and executing Dr. Weld automatically.

Python Implementation

The Python implementation follows the same logic as described in the sections above, with a few differences that leverage smarter scripting approaches compared to Fortran. Namely:

- Trajectories are defined as a class with the appropriate attributes
- Simufact project and process are read by the script and trajectory/sequence information are parsed automatically
- Node coordinates are parsed automatically
- Result (temperature) file writing is done automatically
- Plotting is done optionally through the matplotlib library

The code was run for both validation models, and the results were compared to the subroutine prediction for the "T" joint model. Figure 26 and 27 show the Python script results and comparison to the subroutine comparison.

The execution time for the Python script for the "T" joint was 92 seconds, with a time step of 0.1 second. This represents a significant improvement over the reference and subroutine models.

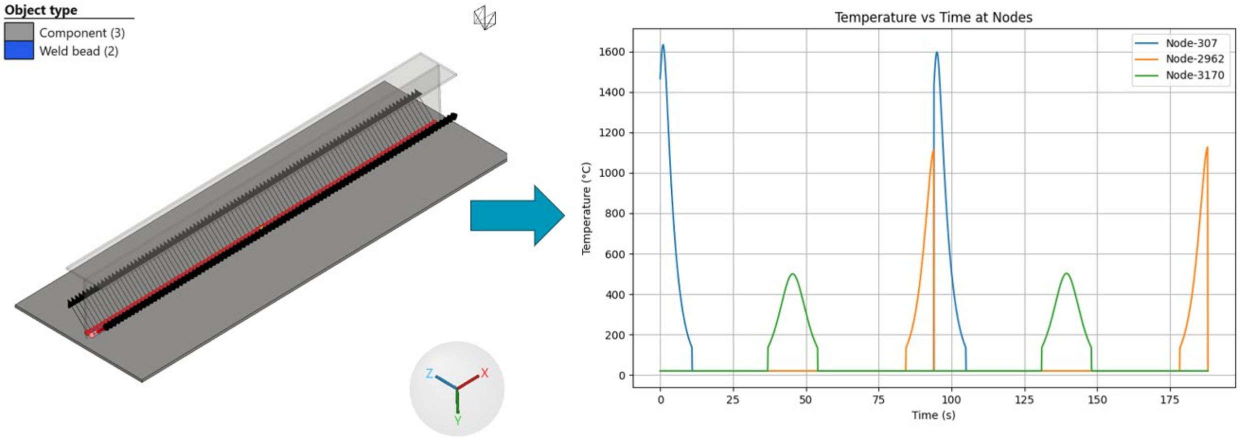


Figure 26. Subroutine results for the T-joint for temperature calculation with the code embedded in the solver

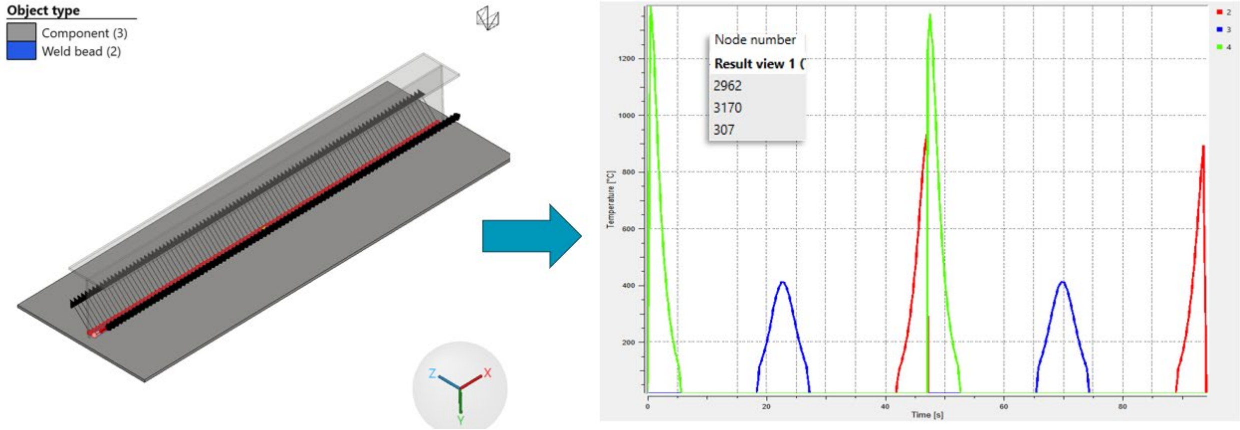


Figure 27. Python results for the T-joint for temperature calculation with the python code

Automated Mesh Creation (Auto-Mesher)

Introduction

The creation of weld surfaces in Global-FEM models has been a challenging aspect of modeling in large shipbuilding structures. It is usual to represent the weld beads as weld surfaces and connect the weld mesh to the adjacent panel mesh, both being 2D elements. Figure 28a shows a representative model and (b) the detail of modeled surface welds.

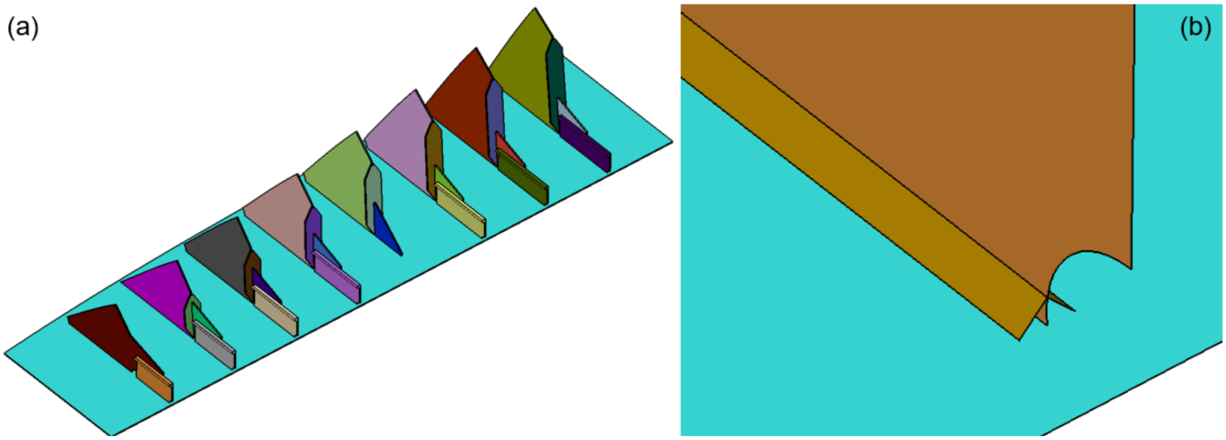


Figure 28. (a) Representative sub-assembly and (b) detail of modeled surface weld

Traditionally, the surface weld bead creation required the careful attention of the engineer to create the surfaces correctly, assign properties, create the mesh connection, and verify the quality of the mesh.

Overall, the whole modeling process for large, welded shipbuilding structures has been a tedious, long, error prone process. A Python script for MSC Apex has been created to automate the major steps in this type of modeling. Namely, the mid-surface extraction, property assignment, weld bead creation, meshing, and mesh connection, are all automated through the script. The scripts can read the weld locations from an XML file coming from the ShipConstructor software, or if that is not available the surface edges can be manually selected.

Objective

The script's main objective is to reduce the workload for engineers working with large shipbuilding structures welding simulation. It enables the process of meshing that would have taken many man hours to be completed nearly 100% automatically, with only some initial inputs. Script execution still takes time, especially for very large models, but it can be left running with no user intervention. Some minor adjustment may be required after the execution is complete to achieve a run-ready model, but the time savings from the script should greatly supersede the time required for such adjustments. Additionally, along with a fully automated script, a set of similar, smaller scripts are available to realize individual steps of the process. For example, a script dedicated to only extracting the mid-surfaces and assigning the thickness property.

Implementation

The script is written in the Python programming language and uses MSC Apex's application programming interfaces (API's) to execute commands from within the script. The script's execution can be divided into a few major sections:

1. CAD Import
2. Mid-surface extraction and thickness assignment
3. Surface extension (closing gaps)
4. Weld Surface Creation
5. Meshing
6. Mesh connection

The fully automated script requires two inputs: (1) an XML file with definition of weld paths and coordinates of the vertexes, and (2) CAD files of the structure. The XML file can be obtained from the ShipConstructor software, containing information about each weld path, such as the locations, and other important information, in a standardized format.

For items 1, 2, 3, 5, and 6, standard functions from MSC Apex are used. For item 4, a detailed logic had to be developed to create the surface welds based on the weld locations. In summary, the weld creation is done following the steps below:

1. For each weld path in the XML file:
    a. Re-construct a spline from vertexes in the XML file
    b. Search the 2 closest surfaces to the spline
    c. Create a cylinder along the weld path
        i. The radius is equal to the selected weld bead leg
    d. Split the 2 near surfaces with the created cylinder
    e. From the split locations, create a surface loft between the new edges (these are the weld surfaces)
    f. Apply the smallest thickness of the two closes surfaces to the weld surface
    g. Optionally, create a transition zone around the weld region
2. Mesh the weld surfaces with selected meshing parameters.
    a. The mesh size (in plane) can be independent of the element length (along weld axis) for additional mesh control.
3. Mesh the panels with selected global mesh size
4. After all welds are created, run a command to make all meshes congruent at the interfaces.
    a. This will connect panels to welds, but also panels to other panels (away from welds).
    b. This command will automatically create a transition zone from the weld mesh size to the global mesh size.
5. Optionally, export mesh files, trajectories, and load them into a Simufact Welding project.

Full descriptions and technical details regarding the script implementation are out of the scope of this document.

Example Results

A few test models are presented below to demonstrate the capability of the script. Figure 29 shows a simple “T” model with two corner welds. This model required a total of two minutes to complete starting from CAD. Meshign and weld creation script took five seconds through the script. No adjustments were needed for a run ready model. The element count with default settings was approximately 3000 elements.

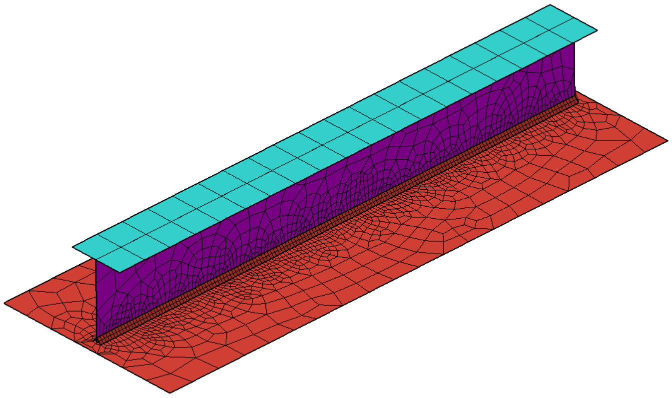


Figure 29. Example mesh on "T" model

Figure 30 shows (a) a representative model with several stiffeners connected to a panel and (b-c) the details for the welded regions and the created meshes. This model required four minutes of modeling time starting from CAD. Meshing and weld creation took 10 minutes through the script. Five minutes were spent with adjustments before achieving a run ready model. The element count with default settings was approximately 50,000 elements.

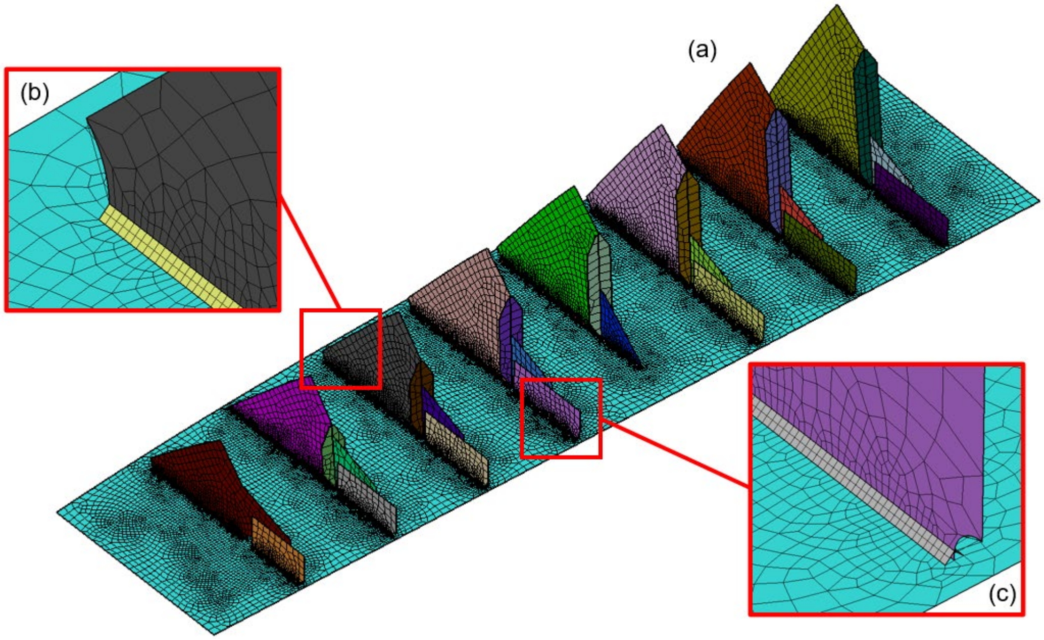


Figure 30. (a) Example mesh on representative model and (b-c) detail of created welds and meshes

Figure 31 shows (a) a model of a subsection of a deck, and (b-d) the detail of created welds and meshes. This model required four minutes of modeling time starting from CAD. Meshing and weld creation took five hours through the script. 30 minutes were spent with adjustments before achieving a run ready model. The element count with default settings was approximately 770,000 elements.

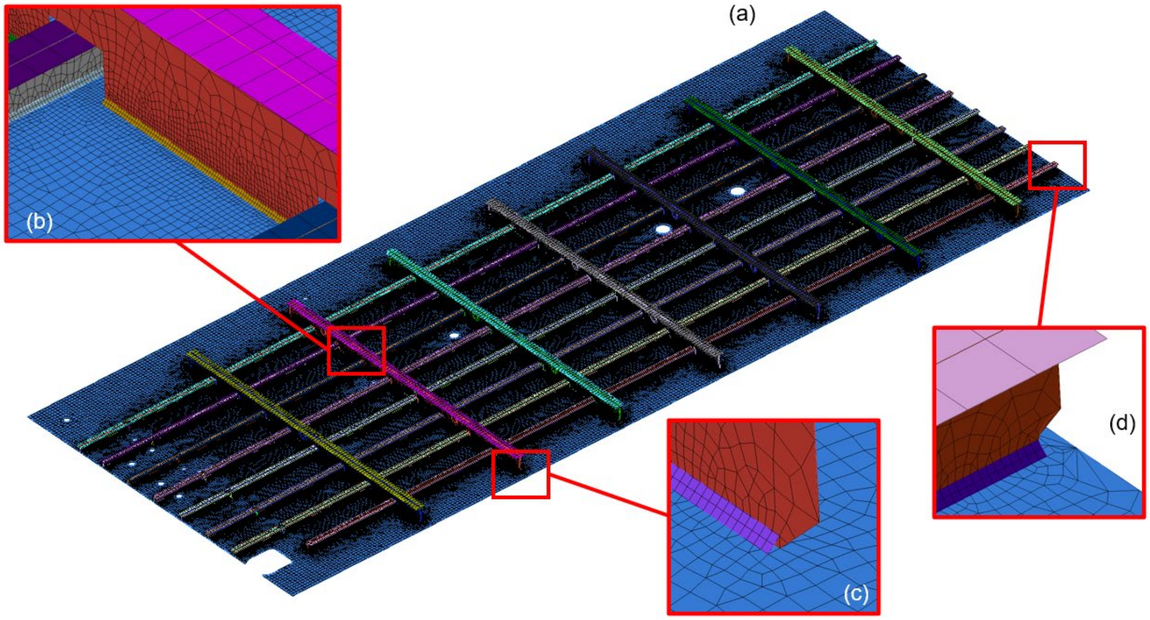


Figure 31. (a) Example mesh on subsection model and (b-d) detail of created welds and meshes.

Figure 32 shows a modal analysis result (a-c) for the 3 models, showing the connection between the welds and the rest of the structure was effective. Supporting PowerPoint presentations contain more details about each model and the execution of the script and results. It is important to note that element counts are very dependent on selected meshing settings.

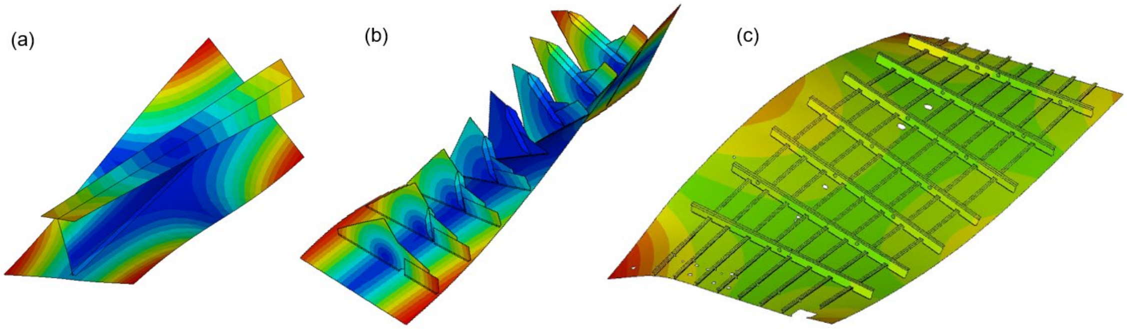


Figure 32. Modal analysis results for "T" model, (b) representative model, and (c) subsection model.

Feature Updates and Script Improvements

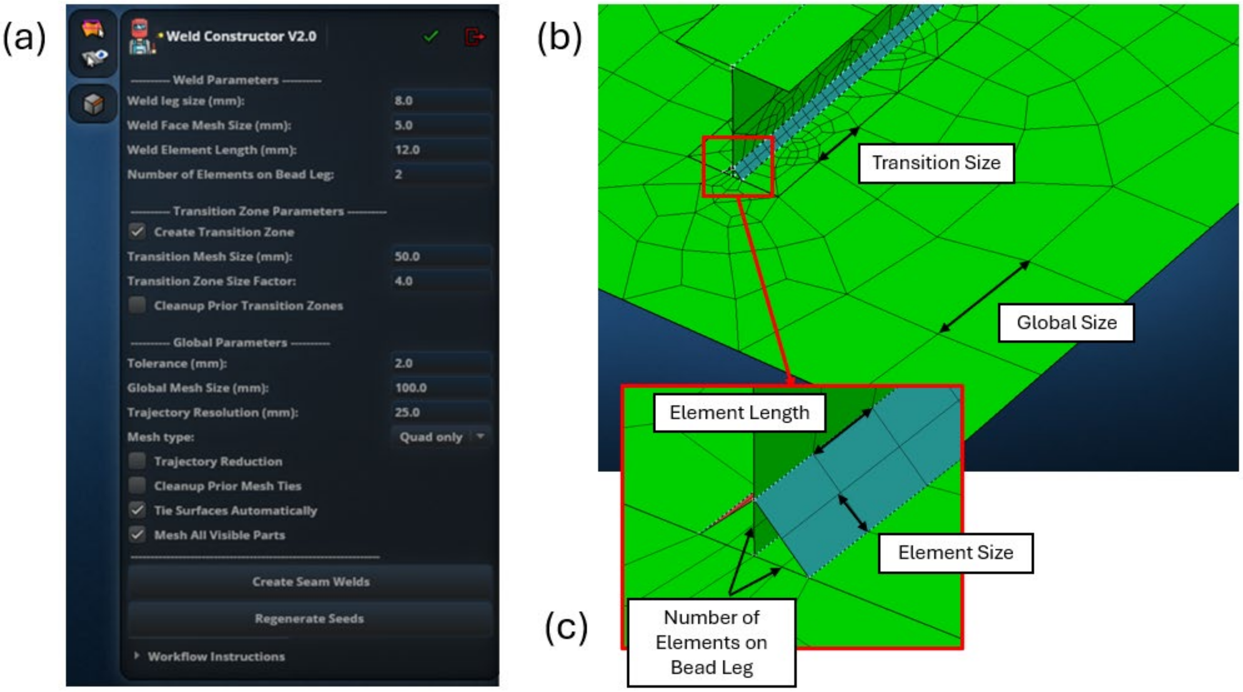
Several improvements and bug fixes were made to the meshing automation. For simplicity, only the major bugs and improvements are listed below:

- **Added**: Transition zone mesh size input – size of mesh on the edges of transition zone, which is Important to better control mesh quality.
- **Added**: Number of elements on weld leg input – controls how many element edges on weld bead leg are created, which is important for better fusion zone behavior.
- **Added**: Transition zone size ratio input – controls how large the transition zone is in relation to the bead size. This proves mesh quality.
- **Added**: Meshing toggle – activates/deactivates meshing, which is useful when creating welds sequentially, so meshing is only done at the end.
- **Added**: Button to regenerate seeds – useful for when geometry changes are made and seeding is lost.
- **Fixed**: Bad mesh quality when using transition zones.
- **Fixed**: Meshing would fail on welds connected side by side.
- **Improved:** Seeding is persistent over multiple executions.
- **Improved:** Script execution script was greatly improved after code profiling and the removal of several bottlenecks.
- **Improved:** Better mesh transitioning.
- **Improved:** Better defaults.

Figure 33 shows (a) the GUI of the Auto-Mesher, (b) the detail of important mesh sizes on the global view and (c) the detail of important mesh sizes on a detail view. The list below describes each option in Figure 33:

1. Weld leg size (mm): size of the weld bead leg.

2. Weld face mesh size (mm): Mesh size perpendicular to the weld trajectory. Controls how many elements are across the width of the weld face.
3. Weld element length (mm): Element length in the direction of the weld trajectory.
4. Number of elements on weld leg: Controls how many elements are on each weld leg.
5. Create transition zone: Toggle to create a region of controlled mesh around weld beads.
6. Transition zone mesh size (mm): Controls the mesh size on the boundary of the transition zone.
7. Transition zone size factor: Size of the transition zone as a function of the bead size (i.e. 4.0 means the transition zone width will be 4 times the weld leg size).
8. Cleanup prior transition zones: Deletes prior edges related to transition zones. It is important to be used if prior transition zones exist in the weld region.
9. Tolerance (mm): Internal tolerance, only important in the case of curved edges. In case an edge is missing a weld, try increasing this value.
10. Global mesh size (mm): Mesh size away from weld regions.
11. Trajectory resolution (mm): Internal resolution of auxiliary edges. This will be the amount of reduction that is applied if the "Trajectory reduction" option is selected.
12. Mesh type: Quad only or Mixed. In quad only, some triangles might be generated. A pop-up will appear in such cases and the user can choose to accept or discard the mesh.
13. Trajectory reduction: Toggle to remove some material at the beginning and end of welds. This is important in regions where two edges would clash.
14. Cleanup prior mesh ties: Activate to delete all mesh ties in the models. Mesh ties are the objects that control the mesh congruence.
15. Tie surfaces automatically: Executes the code that ensures mesh congruence.
16. Mesh surfaces automatically: Toggle to activate or deactivate meshing after the weld surfaces are created.
17. Create seam welds: Button to execute the weld creation script.
18. Reseed edges: Will re-seed the model edges, to result in the correct meshing controls defined by the user. This is necessary when some geometry changes are made to the model and seeds are lost.



--------- Weld Parameters ---------
Weld leg size (mm): 8.0
Weld Face Mesh Size (mm): 5.0
Weld Element Length (mm): 12.0
Number of Elements on Bead Leg: 2

--------- Transition Zone Parameters ---------
☑ Create Transition Zone
Transition Mesh Size (mm): 50.0
Transition Zone Size Factor: 4.0
☐ Cleanup Prior Transition Zones

--------- Global Parameters ---------
Tolerance (mm): 2.0
Global Mesh Size (mm): 100.0
Trajectory Resolution (mm): 25.0
Mesh type: Quad only
☐ Trajectory Reduction
☐ Cleanup Prior Mesh Ties
☑ Tie Surfaces Automatically
☑ Mesh All Visible Parts

Figure 33. (a) Updated script and user interface, (b) example of mesh sizes on global view, and (c) example of mesh sizes on detail view

The scripts capabilities were validated on a subsection model with 3 small stiffeners, 2 large stiffness, and a total of 38 welds. Figure 34 shows (a) the global view of mesh generated by the script with (b) the detail of the welds at stiffener overlap region, (c) mesh at the edge of a baseplate, (d) mesh at connection of two baseplates and two welds, and (e) mesh at the end of a stiffener.

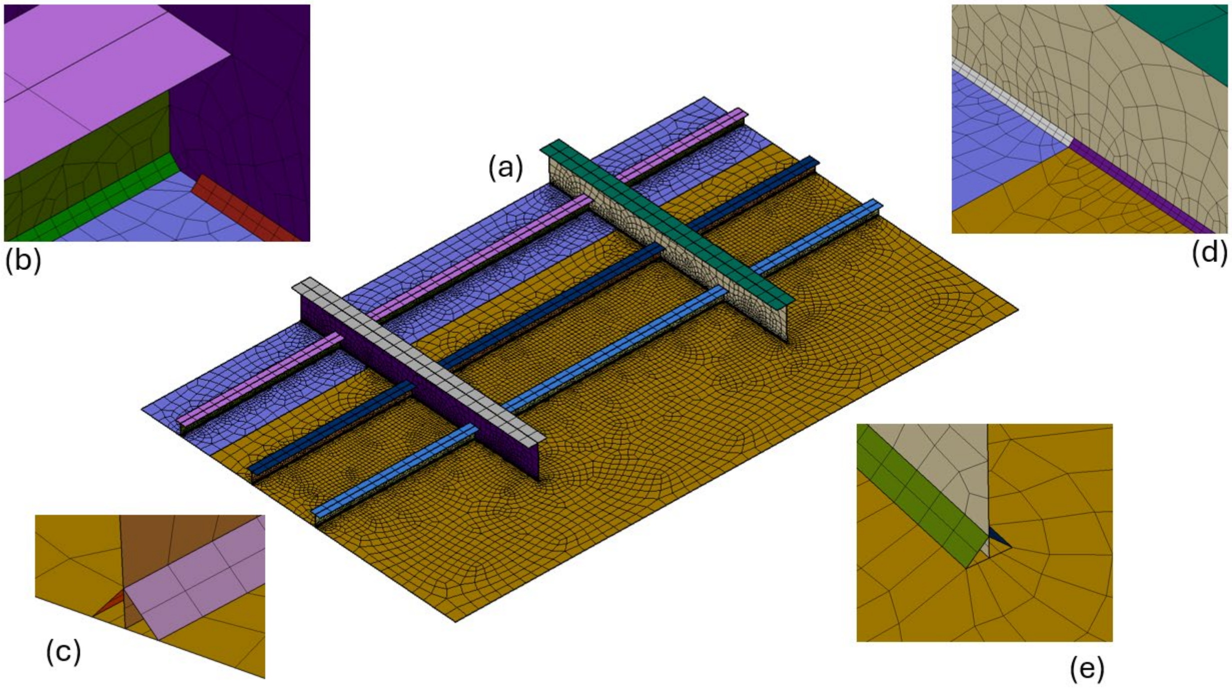


Figure 34. Example of mesh created by automation script

Some manual work is still required in the model preparation stage to make sure that the midsurface connection between the plates is adequate. This can be done with several standard tools in MSC Apex. The version of MSC Apex used in this example was version 2025.1.

With the latest script version, performance bottlenecks were substantially reduced and now even a large quantity of welds can be created in minutes. For the validation model, weld creation took only a few minutes over a few iterative steps. The detailed procedure and training videos are provided separately from this report. Based on user feedback, a model that could take 2 weeks of modeling time, now takes 2 hours, resulting in an improvement of approximately 40 times.

Additional Tools

Along with the Auto-Mesher, a package of tools is provided for other useful operations. Figure 35 shows the “toolkit” and below a list of tools with descriptions is shared.

Figure 35. Toolkit in MSC Apex

1. **Import Beads**: Reads XML file from ShipConstructor and creates reference weld edges in those locations.
2. **Get Trajectory**: Creates a CSV file for the selected edges. This CSV file can be imported in Simufact as a weld trajectory.
3. **Heat Source**: Creates an XML file based on input weld parameters and can be imported in Simufact to be used as a heat source.
4. **Export BDF**: Exports selected parts into BDF format that can be imported in Simufact, along with CSV files with trajectories for any welds in the model.
5. **Extract Midsurface**: Extracts the midsurface geometry from a solid and annotates the thickness to the part name.
6. **Weld Constructor**: Main script that creates the weld geometry based on selected edges.
7. **Mesh Stats**: Provides statistics about the model such as total element/node count and per part element/node count.
8. **Split edge**: In some cases, it is desired to break one side of the edge with an additional vertex. This tool can be used for that purpose.

With this automation script, it is estimated that 80-95% of model setup time can be saved. The assumptions for the estimate are the following, based on the validation model presented (Table 1):

Table 1 - Comparison of modeling time for weld creation (manual vs automated)

Operation	Manual Total Time	Automated Total Time
Geometry split	~30 minutes	<2 minutes
Weld surface creation	~30 minutes	<2 minutes
Geometry cleanup	~1 hour	<5 minutes
Seeding	~30 minutes	<1 minute
Meshing	~10 minutes	<1 minute
Mesh congruence	~10 minutes	<5 minutes
TOTAL	>2 hours 50 minutes	<16 minutes

Those numbers are estimates but represent a fair approximation of the effort required for the model shown. Additionally, the big benefit of automation is that the operations are executed without user intervention.

Process to Create a Mesh

The process to create a mesh is shown in Figure 36. After importing a CAD model to MSC Apex, a user will clean the model by removing small holes and non-necessary parts, creating a middle surface by clicking the icon of "Extract Midsurface" in Figure 35, and partitioning the stiffeners by clicking the icon of "Split Surface" in Apex. The following steps are to split fillet-weld edges near seam welds and outside welds, create weld surface, and create meshes. Finally, the mesh quality will be improved by fixing some bad elements. Re-meshing may be done to coarse the mesh in the area away from the welds.

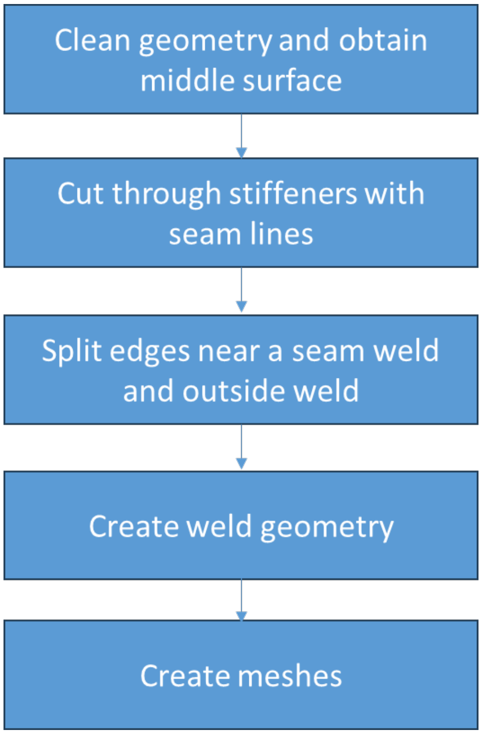


Figure 36. Process of creating a mesh using Auto-Mesher

Shipyard Testing

Auto-Mesher was tested to create a mesh for a typical ship panel that includes a plate, longitudinal stiffeners, and transverse stiffeners. Figure 37 shows the middle surface and the partitioned transverse stiffeners with seam weld lines. Then the panel was flipped over to the smooth side and split edges near seam welds and outside welds, as shown in Figure 38. This step is manually defined as the transverse fillet weld stops. The next step is to create weld surfaces (Figure 39) for longitudinal fillet welds and transverse fillet weld near the seam welds and at the end of stiffeners. Figure 40 shows the created meshed for the panel. Note that the fine mesh is between stiffeners. Those fine mesh is not necessary and will increase the computational time during welding analysis. The final step is to create coarse mesh in those areas by partitioning.

The Auto-Mesher significantly reduces the staff time to create a weld mesh on a production panel. Typically, creating a weld mesh for a production panel takes several days of staff time. With Auto-Mesher, it will take several hours to create a similar mesh.

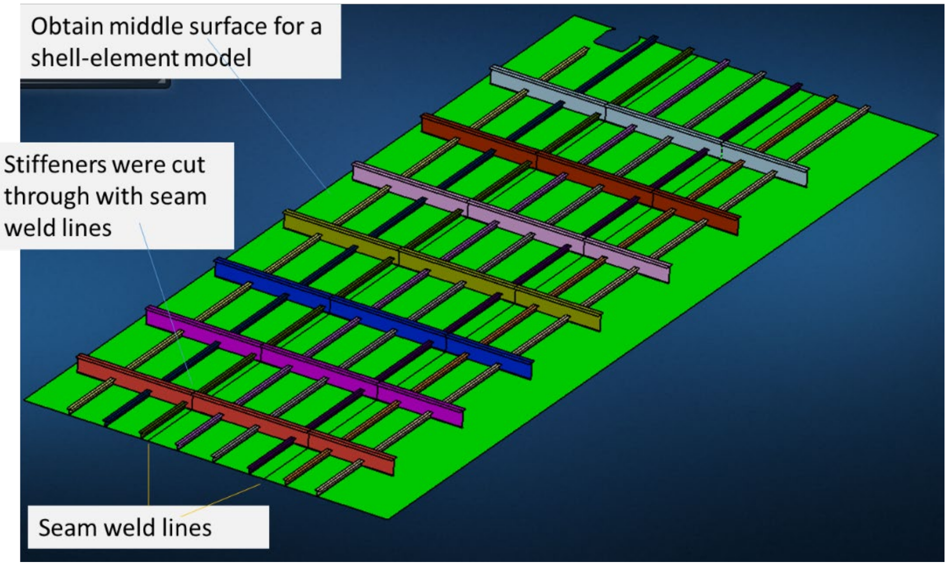


Figure 37. Middle surface and stiffeners partitioning of a typical ship panel

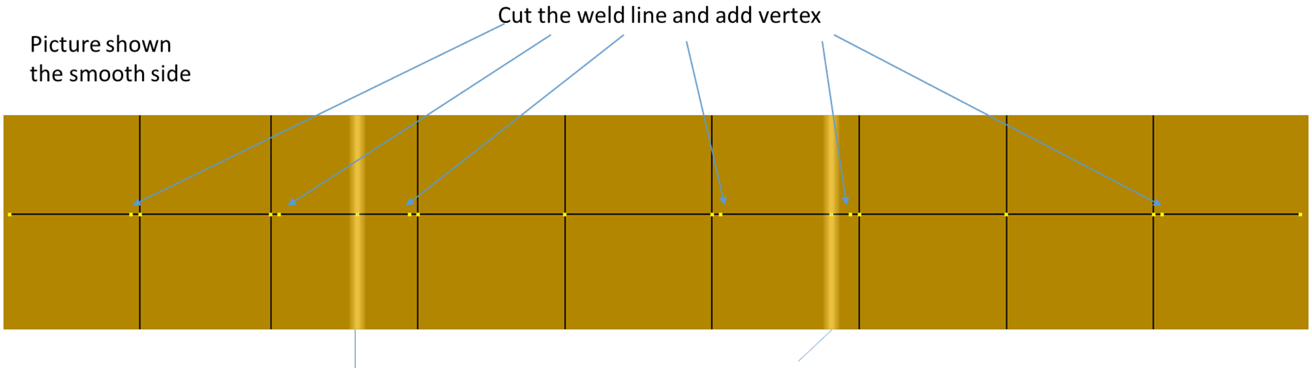


Figure 38. Splitting Edges

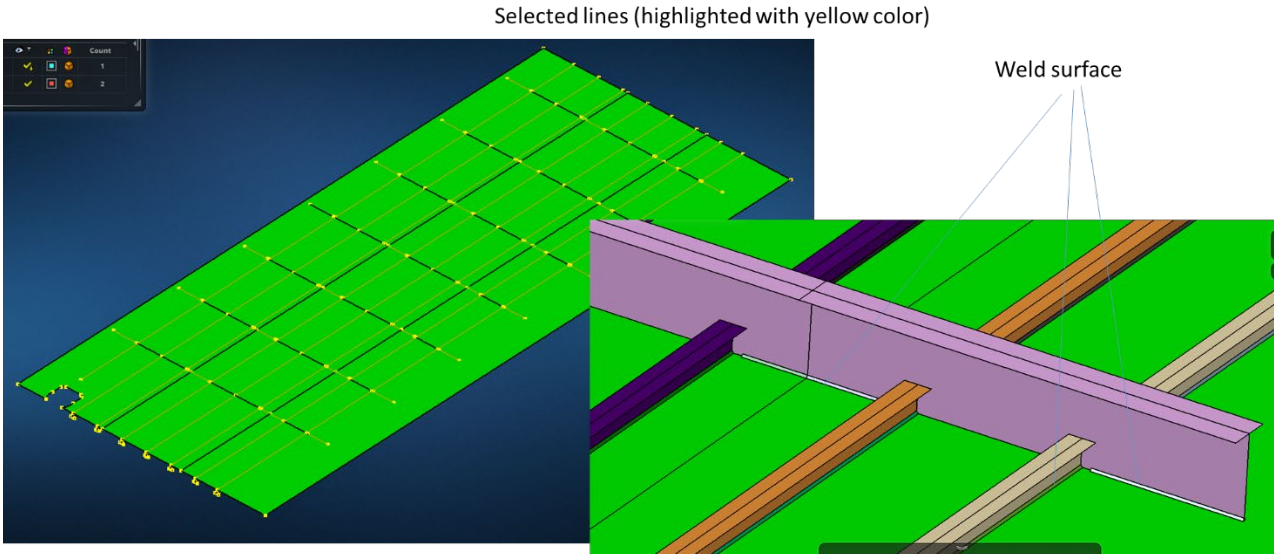


Figure 39. Selecting weld lines and creating weld surfaces

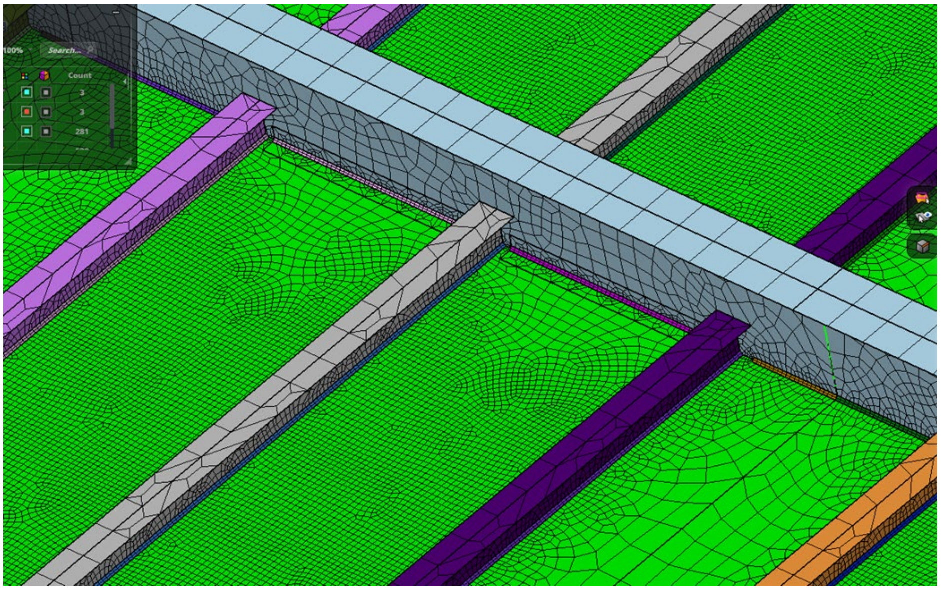


Figure 40. Creating meshes

Software Integration

The integration between Simufact and Dr. Weld (DRW) allows for the automatic execution of Dr. Weld's solver from within the Simufact Welding interface with automated results import, creating a seamless experience for the user executing the analysis. Different approaches are available:

- Standard Thermal + DRW (solid-based)
- Fast Thermal + DRW (solid-based)
- Fast Thermal + DRW (shell-based)

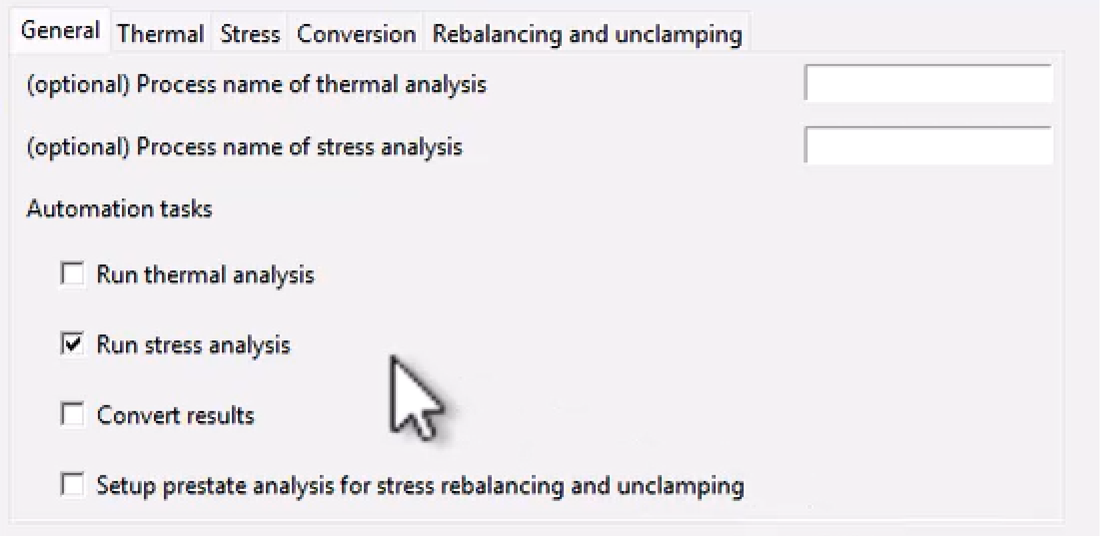
The shell-based approach is the focus of this report since it is the mostly used method for finite element analysis of welding in naval structures. For all approaches, mesh congruency is a requirement. A node merge step is conducted before execution, and the default node merge tolerance is 1E-6 m.

Standard Thermal Solver + Dr. Weld Integration (Solid-based)

This is a legacy method and not recommended, but it is still available. In this method, a Simufact thermal model must be run, and only then will DRW read the results and execute the mechanical part. The advantage of this method is that the temperature prediction will be very accurate, but the disadvantage is that the thermal model might take too long to run.

The execution GUI contains:

1. General settings



(optional) Process name of thermal analysis

(optional) Process name of stress analysis

Automation tasks

☐ Run thermal analysis

☑ Run stress analysis

☐ Convert results

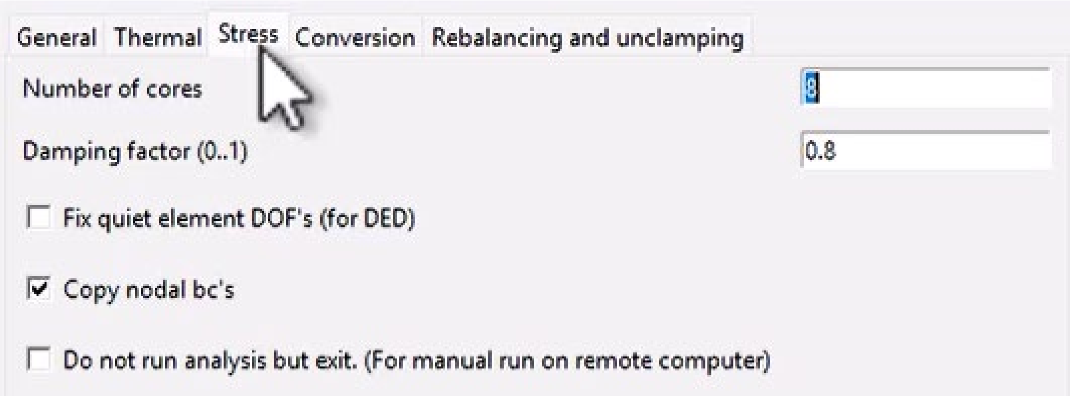
☐ Setup prestate analysis for stress rebalancing and unclamping

2. Thermal settings

☑ Merge nodes

Merge tolerance in m: 1e-7

3. Stress settings



Number of cores: 8

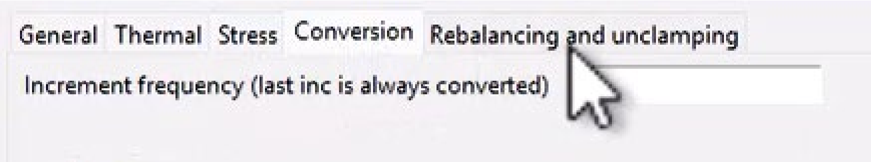
Damping factor (0..1): 0.8

☐ Fix quiet element DOF's (for DED)

☑ Copy nodal bc's

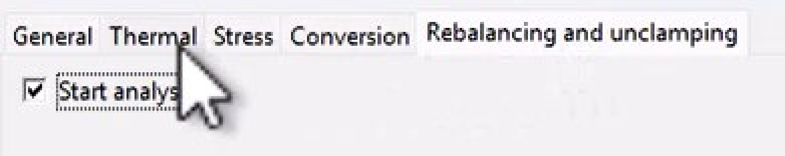
☐ Do not run analysis but exit. (For manual run on remote computer)

4. Conversion settings



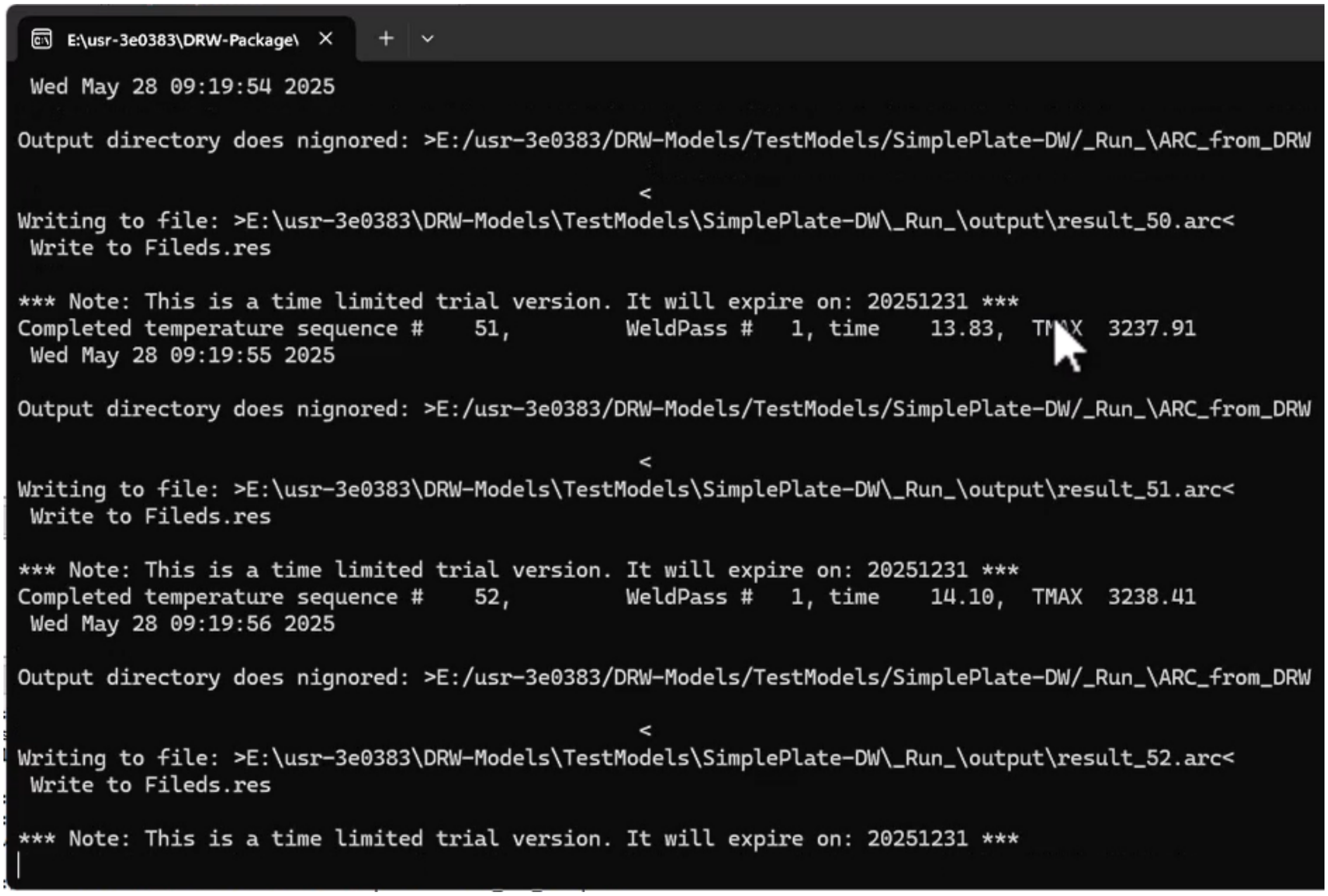
Increment frequency (last inc is always converted)

5. Rebalancing and unclamping settings



☑ Start analys

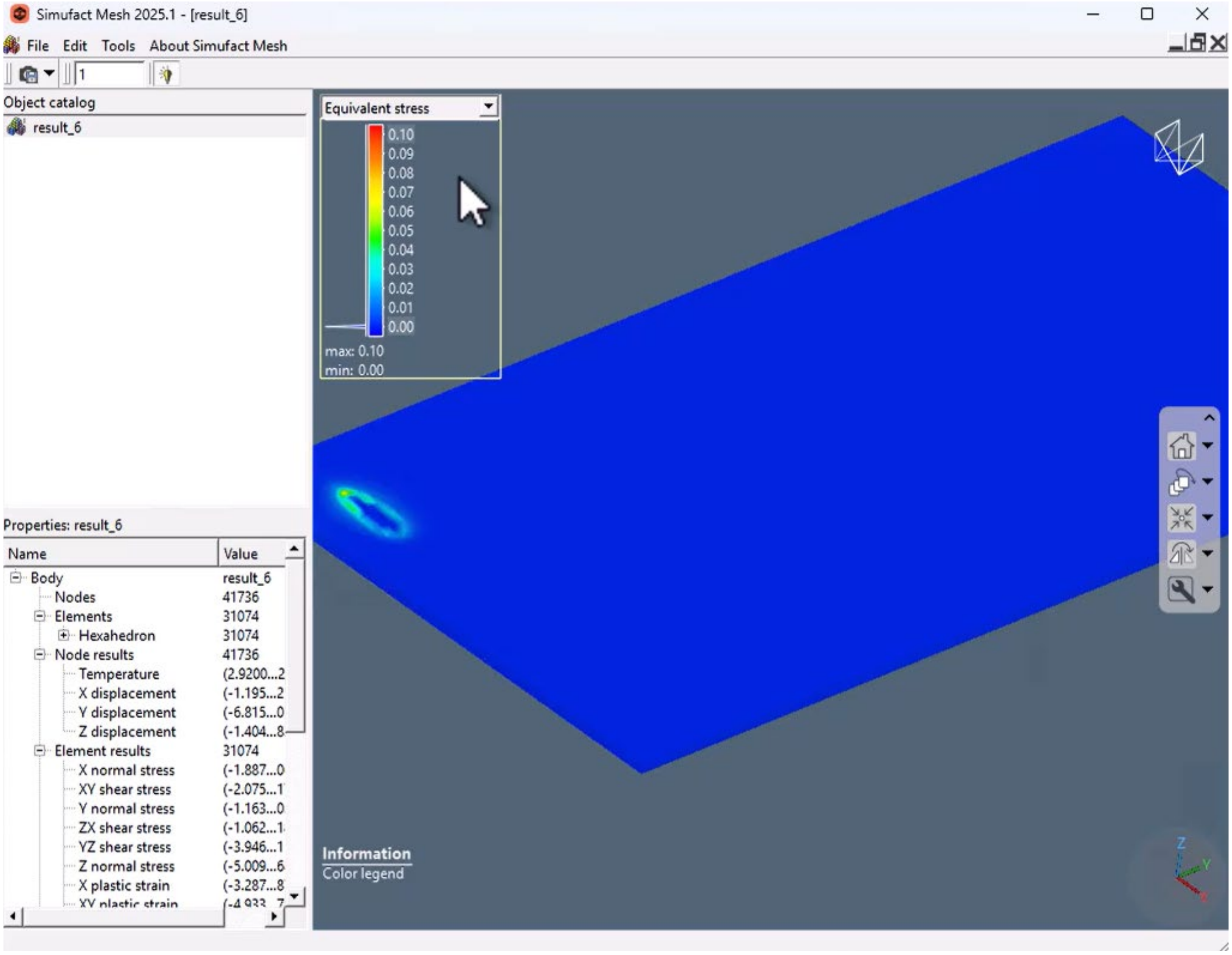
After setting the options, the stress model is ready to be executed as shown in Figure 41.



```
 Wed May 28 09:19:54 2025

Output directory does nignored: >E:/usr-3e0383/DRW-Models/TestModels/SimplePlate-DW/_Run_\ARC_from_DRW
                                                                  <
Writing to file: >E:\usr-3e0383\DRW-Models\TestModels\SimplePlate-DW\_Run_\output\result_50.arc<
 Write to Fileds.res

*** Note: This is a time limited trial version. It will expire on: 20251231 ***
Completed temperature sequence #    51,         WeldPass #   1, time    13.83,  TMAX  3237.91
 Wed May 28 09:19:55 2025

Output directory does nignored: >E:/usr-3e0383/DRW-Models/TestModels/SimplePlate-DW/_Run_\ARC_from_DRW
                                                                  <
Writing to file: >E:\usr-3e0383\DRW-Models\TestModels\SimplePlate-DW\_Run_\output\result_51.arc<
 Write to Fileds.res

*** Note: This is a time limited trial version. It will expire on: 20251231 ***
Completed temperature sequence #    52,         WeldPass #   1, time    14.10,  TMAX  3238.41
 Wed May 28 09:19:56 2025

Output directory does nignored: >E:/usr-3e0383/DRW-Models/TestModels/SimplePlate-DW/_Run_\ARC_from_DRW
                                                                  <
Writing to file: >E:\usr-3e0383\DRW-Models\TestModels\SimplePlate-DW\_Run_\output\result_52.arc<
 Write to Fileds.res

*** Note: This is a time limited trial version. It will expire on: 20251231 ***
```

Figure 41. Command prompt showing the output of Dr Weld messages during a solution calculation.

The example of a Simufact ARC file with Dr. Weld stress results is shown in Figure 42 in the Simufact Mesh tool, that can be used for post processing:



Properties: result_6

Name	Value
Body	result_6
Nodes	41736
Elements	31074
Hexahedron	31074
Node results	41736
Temperature	(2.9200...2
X displacement	(-1.195...2
Y displacement	(-6.815...0
Z displacement	(-1.404...8
Element results	31074
X normal stress	(-1.887...0
XY shear stress	(-2.075...1
Y normal stress	(-1.163...0
ZX shear stress	(-1.062...1
YZ shear stress	(-3.946...1
Z normal stress	(-5.009...6
X plastic strain	(-3.287...8
XY plastic strain	(-4.933...7

Figure 42. Visualization of Dr Weld output in Simufact Mesh using the ARC file format.

For this approach, the integrated post processing is not available.

Fast Thermal Solver + Dr Weld Integration (Solid-Based)

With the fast thermal solver implemented in a Python script, in this approach the user can simply select the process that he wants to analyze, and the complete workflow will be run automatically. The workflow is described in Figure 43. User starts with Simufact model, runs a simple script to select which model to run, and the Fast Thermal Solver and DRW are executed automatically, followed by automatic results import, allowing the user to post process DRW results in the Simufact GUI.

The disadvantage of this method is that the thermal field is not as accurate as the standard thermal solver, but the benefit is that it is much faster than the Simufact standard thermal solver.

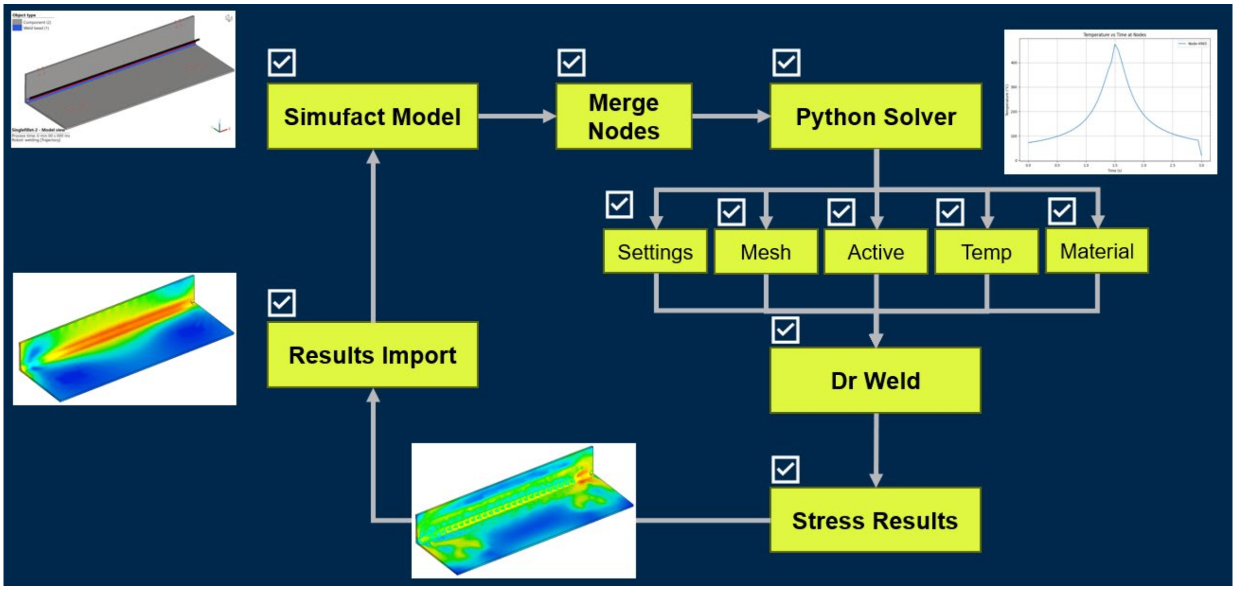
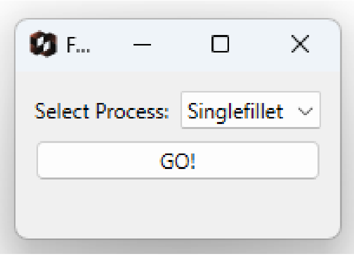


Figure 43. Flowchart for solid-based Simufact + DRW integration

The implementation logic is described in more detail below:

1. Model information is read (mesh, trajectory, sequence, material, etc.)
2. Node merging algorithm is performed
3. Weld sequence and model information are used to compute the temperature for every node in the model for that time step
4. Temperature information is appended to DRW input "Temperature" file
5. After all time steps are completed, all necessary DRW input files are written: "Mesh", "Materials", "Temperature", "Active", "Settings"
6. DRW is called and executed using input files generated by previous step
7. ARC result files are generated by DRW
8. ARC files are automatically imported to Simufact

Figure 44 shows the interface that allows selecting the model to be analyzed. Additional script options are available through the source code but aren't available in the GUI.



Select Process: Singlefillet

Figure 44. Solid-based Fast Thermal Solver + DRW integration GUI.

A simple corner joint welding model was used for testing. The model contains 11 770 nodes, 8 412 hexahedral elements, and 1.88 m weld length. Figure 45 shows the side-by-side comparison of results for the model. For the temperature results, a intermediate result is shown simply as an example. The other results are related to the final increment of the simulation. The model was run using Simufact Welding version 2025.2.

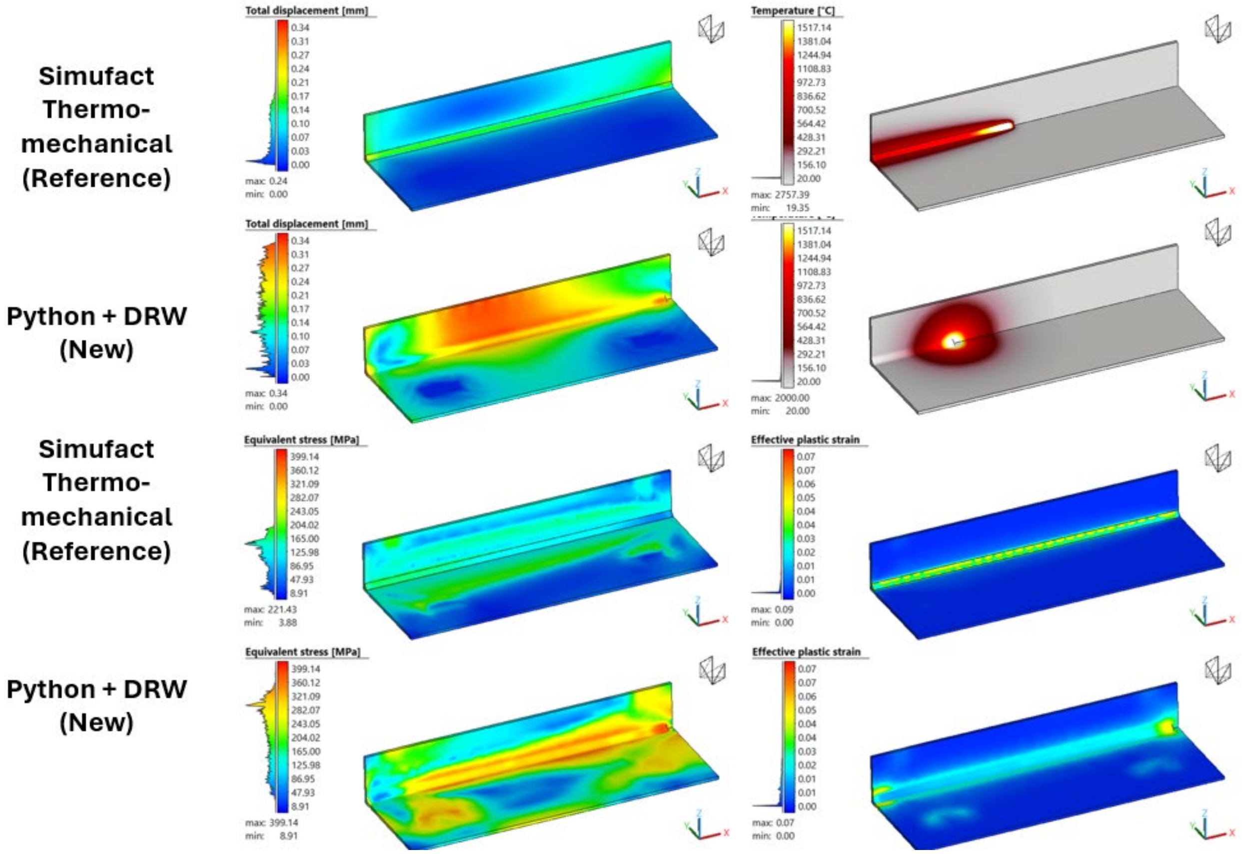


Figure 6. Comparison of results for solid-based approach

Although comparable, the results of the new approach show some significant difference, albeit on the same order of magnitude. This difference is natural due to the level of simplification applied to the thermal solution, plus the use of the explicit method by DRW (in contrast to the implicit method used by standard Simufact). The results could likely be improved by improving thermal behavior in the new approach, since it had a relatively large heat affected zone size. However, the development of the shell-based approach took precedence over the improvement of solid-based approach due to the limited time availability for the project. The next section will detail the shell-based approach and its results.

Fast Thermal Solver + Dr Weld Integration (Shell-Based)

Next, the shell-based integration was developed. In this workflow, the model creation section was added, since it is a crucial step and it can be done using the automated mesher presented in previous sections. In the first portion of the workflow, the user will load the CAD geometry into MSC Apex, execute some pre-processing steps, use the automesher to automatically create welds and mesh the parts, and then export mesh and weld information. Next, an automated script, described later in this section, will automatically set up the Simufact Welding model, with the user being required only to select the boundary conditions. After that is done, the execution GUI can be used to define key simulation parameters, and the simulation will run, first with the Fast Thermal Solver for the thermal part, then automatically calling the DRW solver for the mechanical part. The results are then automatically imported and available for post processing in the standard Simufact Welding GUI. Figure 46 shows the flowchart for the shell-based integration.

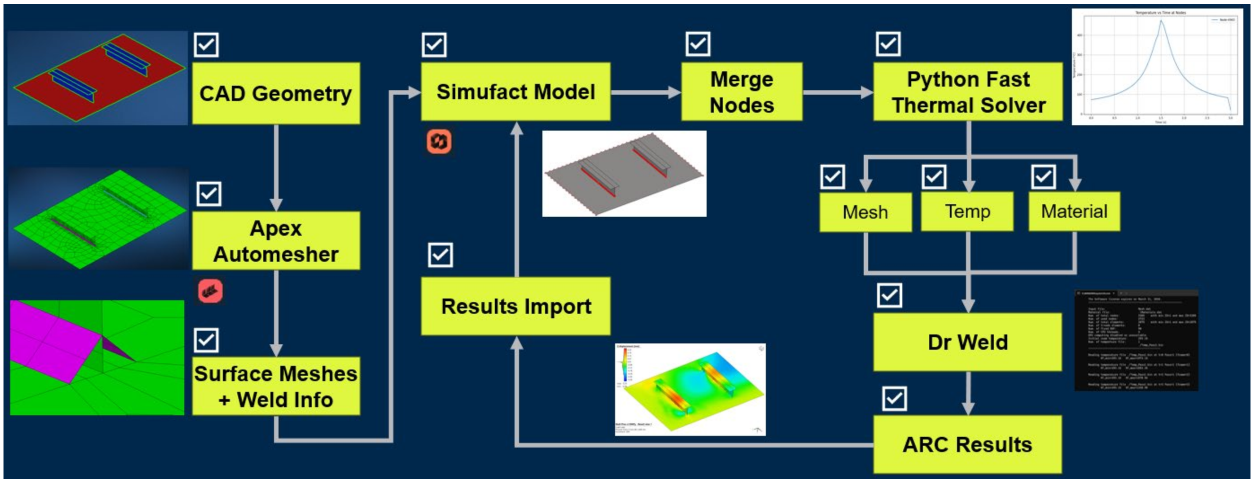
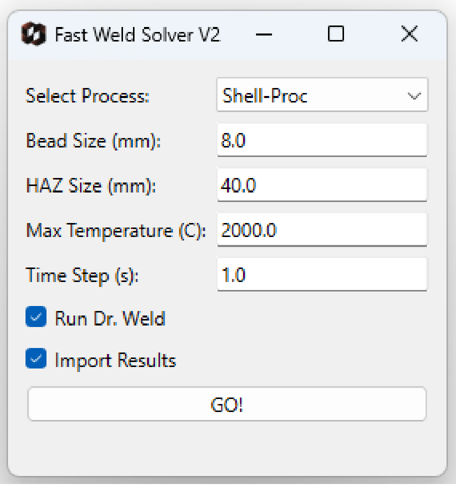


Figure 46. Flowchart for shell-based Simufact + DRW integration

Figure 47 shows the GUI for the shell-based Fast Thermal Solver + DRW integration. Bead size will be the region around the weld trajectory defined as the fusion zone. HAZ size will be the region past the fusion zone where a temperature transition will happen, from the maximum temperature to the room temperature. Max temperature is the temperature for any node within the fusion zone. The time step size will control how many total increments will be in the execution. More increments lead to more precision but also more simulation time. Other parameters, such as the room temperature, and the exponential decay, can only be controlled directly through the source code. However, since the script is written using the Python programming language and standard software API's, it is possible for users to customize and update the script as needed.



Select Process: Shell-Proc

Bead Size (mm): 8.0

HAZ Size (mm): 40.0

Max Temperature (C): 2000.0

Time Step (s): 1.0

☑ Run Dr. Weld

☑ Import Results

Figure 7. GUI for the shell-based Fast Thermal Solver + DRW integration.

A simple "Double Tee" joint welding model was used for testing. The model contains 6 412 nodes, 2 875 quadrilateral elements, and 2.16 m weld length, with 5 seconds cooling time between each of the 4 welds. Figure 48 shows the comparison of the Double Tee model between Simufact's standard solver and the Fast Thermal Solver + DRW approach. Both models shown in this section were run using Simufact Welding version 2025.3.

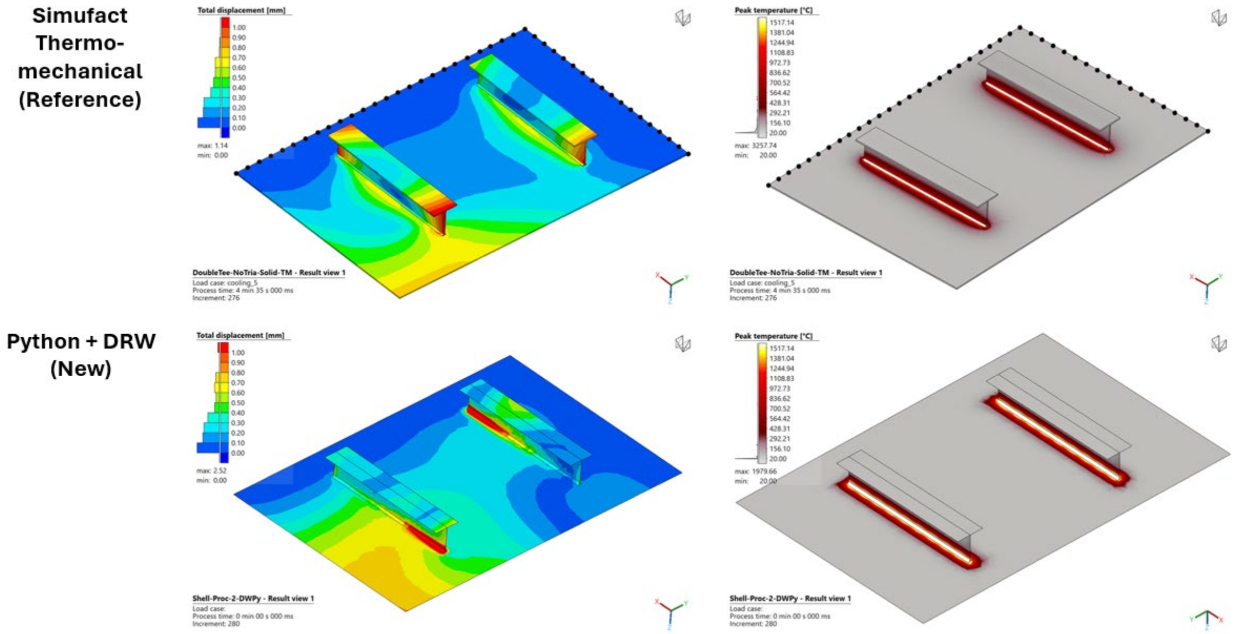


Figure 48. Comparison of results for shell-based double tee model

Both the displacement field and peak temperature results show very similar trends and magnitudes. The DRW model shows some hotspots at the end of the welds, which is currently being investigated.

Next, as a validation case, the same model used for the validation of the automesher was used with the new simulation approach and compared to the standard Simufact solver. In this case, the model contains 78 914 nodes, 34 919 quadrilateral elements, and 37.5 meters of weld length, for a total of 38 welds. Figure 49 shows the comparison of the validation model between Simufact's standard solver and the Fast Thermal Solver + DRW approach.

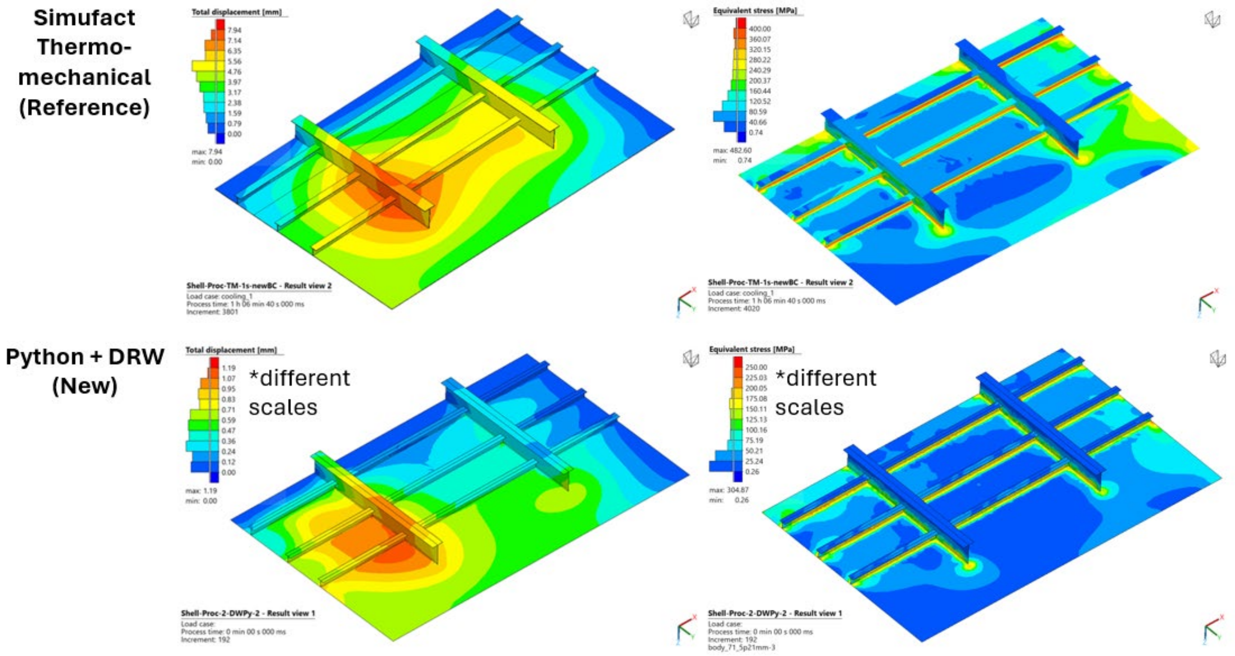


Figure 49. Comparison of results for shell-based validation model (different scales).

The displacement results are shown using two different scales, since there is significant difference between the results magnitudes. It is possible to observe that the distortion trends are captured correctly. This demonstrates that the model can be used to test different model variations, such as different weld sequences, clamping strategies and others, even if the magnitudes are not precisely captured.

The Simufact thermo-mechanical model has a much larger displacement possibly due to higher maximum temperatures occurring in the Simufact heat transfer, fully coupled model. The stress field also has comparable results with hotspots in the same locations, but the DRW results have a lower magnitude. A more in-depth investigation must be conducted as future work to evaluate the root cause of this issue. At this point, this comparison will serve to highlight the time-saving benefits of the new approach, but additional validation must be done in the future.

It is also interesting to note the total folder sizes for both models. With the standard Simufact approach, the total folder size is 15.5 Gb, while the folder for the model using the new approach has 1.7 Gb. Both models have 1 result output for every 10 solver increments. This highlights another benefit which is a reduction in results size, thus requiring less storage.

With the new approach highlights being discussed in this section, the next section will aim in an automated model creation script, that can greatly reduce modeling effort in Simufact.

Automated Simufact Model Creation

Additional to executing the simulation, a script was developed to automatically set up the Simufact model. For that, welding information is exported from MSC Apex and organized into a set of folders. Materials and welding parameters are also added to this folder. Finally, a CSV file with the definition of material per part must be provided. After the folders are organized, the script automatically creates the Simufact model once executed. The only required steps for the user are defining boundary condition locations and defining the total simulation time.

With this automation script, it is estimated that up to 95% of model setup time can be saved. The assumptions for the estimate are presented on Table 2, based on the validation model presented.

Table 2 - Comparison of Simufact model setup times (manual vs automated)

Operation	Manual Total Time	Automated Total Time
Component setup	~10 minutes	<1 minute
Trajectory/Weld bead setup	~30 minutes	<1 minute
Contact table setup	~10 minutes	<1 minute
General process setup	~5 minutes	<1 minute
TOTAL	>55 minutes	<4 minutes

Those numbers are estimates but represent a fair approximation of the effort required for the model shown. Additionally, the big benefit of automation is that the operations are executed without user intervention.

Summary

Table 3 presents the thermal-mechanical run time versus the run time for the models that used the Fast Thermal Solver + DRW, along with the obtained speedup.

Table 3 - Run time comparison for standard Simufact vs Fast Thermal Solver + DR-Weld

Model	Standard Run Time	New Run Time	Speedup
Single Fillet (Solid)	430 s	38 s	11.3X
Double Tee (Shell)	814 s	94 s	8.7X

Subsection (Shell)	84 243 s	4 867 s	17.3X

Table 4 breaks down each of the models run time into the thermal and thermal-mechanical (TM) portions, to illustrate what the contribution of each pass is. For the fully coupled Simufact models, the thermal run time represents a thermal only-model, and the total TM is for the coupled model. For the new approach, the total time is presented, along with the individual thermal and mechanical run times, since those are sequentially coupled and can be independently measured.

Table 4 – Breakdown of thermal and thermal-mechanical run times for each model

Model	# of elements	# of Increments	Standard Total/TM Run Time	Standard Thermal Run Time	New Total Run Time	New Thermal Run Time	New Mechanical Run Time
Single Fillet (Solid)	8 412	70	430 s	33 s	38 s	2 s	32 s
Double Tee (Shell)	2 875	600	814 s	102 s	94 s	54 s	38 s
Subsection (Shell)	34 919	3 800	84 243 s	12 414 s	4 867 s	3 118 s	1 613 s

Note that for the new approach the total time is not the simple sum of thermal and mechanical passes, since other operations in the script will also take some time. For the larger model (Subsection), the thermal analysis takes about double the time as the mechanical analysis. This behavior could arise for low performance on some operation in the thermal prediction code, such as nested loops, which are set to be improved in future work.

To summarize, even with some necessary improvements still required, the Fast Thermal Solver + DRW workflow still represents a significant improvement over the traditional approach and will significantly reduce the development cycle for welded naval assemblies.

Software Implementation Plan and Technology Transfer

The primary use-case for the development of this solver will be weld-sequencing simulation of stiffened panel and unit assembly construction. The complex nature of weld sequencing for distortion control will allow the team to understand the current solver time and computational requirements, address existing limitations, train engineers on efficient and accurate use of the tool, and utilize the GUI to tackle the optimization of other manufacturing processes in shipbuilding industry. The solver will be capable of replacing current analysis tools utilized to simulate welding processes to predict welding-induced stress distributions, which can be used to improve ship structural design, predict fatigue life and cracking in high stress areas, and minimize movement of alignment critical structures where hot work is required in the surrounding areas. Other potential applications include the optimization of the thermal straightening process used to correct distortion, prediction of heating paths for thermally forming curved shell plates, optimization of the hybrid laser arc welding (HLAW) process to improve toughness, and dynamic simulation of mechanical systems.

Presentations have been given in the following meetings/conferences for technology transfer:

- 2023 NSRP all panel meeting, Charleston, SC
- 2023 SNAME conference, San Diego, CA
- 2023 FabTech conference, Chicago, IL
- 2024 NSRP Joint Panel Meeting, April 30th through May 2nd, Suffolk, VA
- 2024 NSRP Business Technologies and Ship Design & Material Technologies Panels, Vancouver, Canada

Business Case and ROI

The business case for implementing the results of this project are based on both engineering labor savings (reduced time per analysis) as well as the reduction in hull labor hours resulting from each analysis ability to optimize design and process variables for improved quality.

Engineering Labor Savings

Utilizing the speed-up rates from the model generation and run time acceleration in tables 2-4 above, and extrapolating to an average full-scale unit, an overall factor of 25x was used to calculate man-hour savings. Table 5 below shows the baseline and new process analysis requirements and the resultant 5-year man-hour expenditures for each.

Table 5 - Engineering Labor Savings

Hrs./Analysis	Total Hours (per year)	Total Hours (5-year)
128 (baseline)	3,200	16,000
13 (new)	325	1,625

Hull Labor Savings

In addition to the Engineering labor savings from the accelerated analysis, the implementation of the process optimization and quality improvement results in significant hull manufacturing labor savings at various workstations and activities throughout construction. Utilizing past unit analyses and average labor savings associated across multiple units, a percentage reduction in hours at various process steps is shown in Table 6 below.

Table 6 - Hull Production Labor Savings

CSA (Fit)	CSA (Weld)	Stacking (Fit)	Flame Straightening	Avg. Hours Savings/Unit	Total Hours Savings (5-year)
20%	10%	25%	30%	640	80,000

Project ROI

Combining the Engineering and Hull labor savings, and converting to labor rates for each, the overall project ROI is shown below:

$$ROI = \frac{Return \; - (Investment + Implementation \; Cost)}{(Investment + Implementation \; Cost)}$$

$$\text{5yr } ROI = \frac{\$8{,}956{,}250 - (\$1{,}146{,}510 + \$200{,}000)}{(\$1{,}146{,}510 + \$200{,}000)}$$

$$\text{5yr } ROI = 5.7$$

Figure 8 - ROI Calculations

Conclusions

Fast Mechanical Analysis Solver (DR-Weld)

- A fast mechanical analysis solver for welding simulation was developed using both transient thermo-mechanical and thermal-strain-based approaches. The solver included shell elements with full and reduced integration. The through-thickness integration points provide the capability to simulate the temperature gradients along with thickness during welding.
- A simple to complex design development approach was taken to test and refine the solver. It was found that the solver is able to predict reasonable distortion trend and magnitude as predicted by Abaqus.
- The solver offers a significant speed advantage over existing commercial software, making it practical for rapid design iterations and large-scale engineering analyses. It can be used to optimize welding sequence for large ship structures.

Fast Thermal Analysis Solver

- A FE-based and an analytical-based solver thermal-analysis solver was developed to predict temperature which will directly input to Dr-Weld to predict distortion. The solvers were validated against reference thermo-mechanical results from Simufact, with reasonable accuracy.

Automated Mesh Creation (Auto-Mesher)

- An automated mesh creation tool was developed in MSC Apex. This development significantly reduces the mesh creation time for large ship structures.

Automated Analysis Process and GUI

- Automated analysis process flow and GUI were developed to allow:
    - Automatically run the Fast Thermal Solver and DRW, without user intervention
    - Automatically import results, without user intervention
    - Post process results on the standard Simufact GUI

Opportunities for Future Development and Applications

Some of the current limitations are listed below, that can be developed for future efforts or for expanding the applications of the software and solver.

- There is still the need for correct geometry preparation and some cleanup after the weld geometries are generated by the Auto-Mesher
- The temperature behavior at the “tail” of the fusion zone can be improved
- Currently only a single robot is supported (only single heat source active at a given time)
- Rosenthal’s method is available as a demonstrator in the source code, but hasn’t been fully validated

- Large models can still take a significant amount of time to run
- The time step size is still a user decision and must be tested on smaller models to find the best compromise between accuracy and speed

Some of the main areas for future work match the known limitations, and are listed below:

- Improvements to thermal field predictions and cooling behavior
- Additional performance improvements
- Coupling thermal solve to Dr-Weld code directly (removing the need for intermediate files)
- Support multiple simultaneous trajectories
- More validation cases with experimental data for correlation

The new approach, although limited in some respects, showed very promising results and the potential for very substantial reductions in running times when coupled to Dr. Weld's mechanical solver. The objective of having a fully working prototype for the shell-based approach was achieved.

Recommendations

Due to the project time constraint, limited shipyard testing was conducted on selected geometries. More tests should be done to identify the software bugs and improve the software user ability.

References

Feng, Z., Chen, G., Chen, J. "High-Performance Computing for Weld Residual Stress Modeling: A Feasibility Evaluation." *Program on Technology Innovation,* 2015, EPRI, Palo Alto, CA.

Goldak, J., Chakravarti, A., and Bibby, M. "A new finite element model for welding heat sources." *Metallurgical Transaction B*, 15B:2 (1984): 299-305.

Huang, H., Ma, N., Chen, J., Feng, Z., Murakawa, H. "Toward large-scale simulation of residual stress and distortion in wire and arc additive manufacturing." Additive Manufacturing 34 (2020): 101248.

Huang, H., Chen, J. Feng, Z. Wang, H.-P., Cai, W. CARLSON, B. E. "Large-scale welding process simulation by GPU parallelized computing." Welding Journal, 100:11 (2021): 359s-370s

Huang, H., Ma, N., Murakawa, H., Feng, Z. "A dual-mesh method for efficient thermal stress analysis of large-scale welded structures. Int. J. Adv. Manuf. Tech. 103 (2019): 769-780.

Ma, N., Umezu, Y. “Application of explicit FEM to welding deformation Analysis.” Welding International 23:1 (2009): 1-8.

Ma, N. An accelerated explicit method with GPU parallel computing for thermal stress and welding deformation of large structure models.” The International Journal of Advanced Manufacturing Technology 87:5-8 (2016): 2195-2211.

Mindlin, R. D., “Influence of rotatory inertia and shear on flexural motions of isotropic, elastic plates.” ASME Journal of Applied Mechanics, Vol. 18 pp. 31–38, 1951.

Olsen, T. M., Runnemalm, H., Berglund, D. “Simulation of welding using MSC.Marc.” https://www.academia.edu/7211140, last access on 7/18/2024.

Reissner, E. “The effect of transverse shear deformation on the bending of elastic plates.” ASME Journal of Applied Mechanics, Vol. 12, pp. A68–77, 1945.

Schill, M., Odenberger, E. L., “Simulation of residual deformation from a forming and welding process using LS-Dyna.” 13th international LS-Dyna users conference, October 5-7, 2021.

Yang, Y. P., Brust, F. W., Dong, P., Zhang, J., and Cao, Z. “Numerical Prediction of Welding-Induced Buckling Distortion and Buckling Mechanism Studies.” International Conference on Computer Engineering and Science, 21-25 August 2000, Los Angeles, CA, USA.

Yang Y P, Athreya B P. “An Improved Plasticity-Based Distortion Analysis Method for Large Welded Structures.” Journal of Materials Engineering and Performance 22: 5 (2013): 1233-1241.

Yang, Y. P., “Recent Advances in Prediction of Weld Residual Stress and Distortion – Part 1.” Welding Journal, 100:5 (2021): 151s-170s.