A Versatile Simulation Framework for Elastodynamic Modeling of Structural Health Monitoring

Elizabeth D. Gregory
Langley Research Center, NASA
Hampton, VA USA
elizabeth.d.gregory@nasa.gov

Cara Leckey
Langley Research Center, NASA
Hampton, VA USA
cara.ac.leckey@nasa.gov

William C. Schneck III
Langley Research Center, NASA
Hampton, VA USA
william.c.schneck@nasa.gov

Paul Swindell
William J. Hughs Technical Center, FAA
Atlantic City, NJ USA
paul.swindell@faa.gov

ABSTRACT

Structural health monitoring (SHM) has the capacity to reduce failure by detecting damage during service life, by periodic, automated monitoring. Guided Wave (GW) Ultrasound is a common SHM approach for aerospace structures. Modelling the physics of GW SHM systems provides a route for understanding system dependencies, capabilities and limitations as damage evolves during service life. Such a toolset can strengthen the understanding of the connection between GW SHM results and the true material state. The most useful modelling tools are those that provide versatile solutions with respect to the simulated component geometry and computational grid connectivity. This work details a versatile application programming interface (API) for the elastodynamic finite integration technique for modelling GW SHM of metals. The custom code implementation, EFIT-CompCell, allows for the modelling of diverse geometries by automatically balancing the message passing interface parallelization layout. The user provides the basic parameters of the simulation and the software automatically performs an initial balancing based on anticipated computational loads, and establishes the CPU communication patterns for any geometry. This work describes the programming philosophy and code structure used to create EFIT-CompCell and compares its performance and capacity to simulation tools that are more specialized for specific architectures. Results are presented for a simulation of GW SHM of an aluminum fuselage section being tested by the FAA. The simulation consists of 733M voxels which took approximately 70 hours to complete 25000 time steps using 40 Intel Xeon E5-4650v2 Ivy Bridge processor cores.

ABOUT THE AUTHORS

Elizabeth Gregory is Research Engineer at NASA Langley working as part of the High Performance Computing Incubator project.

William Schneck is Research Engineer at NASA Langley working as part of the High Performance Computing Incubator project.

Cara Leckey is a Senior Research Physicist at NASA Langley and lead of the High Performance Computing Incubator Project.

Paul Swindell is Program Manager for Structural Health Monitoring at the FAA Tech Center.

A Versatile Simulation Framework for Elastodynamic Modeling of Structural Health Monitoring

Elizabeth D. Gregory
NASA Langley Research Center,
NASA
Hampton, VA USA
elizabeth.d.gregory@nasa.gov

Cara Leckey
NASA Langley Research Center,
NASA
Hampton, VA USA
cara.ac.leckey@nasa.gov

William C. Schneck III
NASA Langley Research Center,
NASA
Hampton, VA USA
william.c.schneck@nasa.gov

Paul Swindell
William J. Hughs Technical Center,
Federal Aviation Administration
Atlantic City, NJ USA
paul.swindell@faa.gov

INTRODUCTION

Structural health monitoring (SHM) typically entails the use of sensors attached to a structure for the purpose of monitoring the health of the structure. For aerospace structures, SHM systems would, ideally, detect unexpected in-service damage above a critical size (for example, due to bird strike or hail) or manufacturing defects that have grown to reach a critical size due to component cycling during service life. SHM has the potential to reduce failure by detecting and monitoring damage prior to a component's critical state, while enabling more frequent and less labor-intensive automated monitoring. Ultrasonic guided waves (GW) are a commonly used investigative SHM approach for aerospace structures [1]. GW SHM is a method that frequently relies on the use of piezoelectric sensors to generate ultrasound in the structure, and to receive returning ultrasound signals which carry details on structural health. Modelling the physics of GW SHM systems provides a route for understanding system dependencies, capabilities and limitations. For example, it has been shown that for a sparse array GW SHM sensor system, the sensor locations can greatly affect the system's ability to detect and locate damage [2]. In that work by Yu and Leckey, it was also demonstrated that simulation tools can be used to investigate the effect of sensor location on detectability. Additionally, simulation tools can strengthen the understanding between GW SHM results and the true material state.

The most useful modelling tools are those that provide versatile solutions with respect to the simulated component geometry and computational grid connectivity. This work details a versatile and modular application programming interface (API) for the elastodynamic finite integration technique (EFIT) for modelling GW SHM of metallic aerospace components. EFIT is a staggered grid finite difference approach that was applied to elastodynamics in the 1990s [3,4]. Numerous mathematical approaches have been developed to model ultrasound behavior in solid media, including analytical and numerical methods [5]. Numerical methods are appropriate to use in cases with complex geometries, irregular defects, and/or complicated ultrasonic wave behavior as is often the case with GW ultrasound. Finite element analysis (FEA) is the most commonly used numerical approach for ultrasound modeling, with several packages that are commercially available. However, methods such as EFIT allow for easily parallelizable code that can be readily adapted to the latest computational architectures [6].

The custom code implementation described in this work, EFIT-CompCell, allows for diverse structural geometries with no changes to the computational parallelization scheme. EFIT-CompCell utilizes message passing interface (MPI) for parallelization. The user provides the basic parameters of the simulation and the software automatically performs an initial computational volume balancing based on anticipated computational loads, and establishes the CPU communication patterns for any geometry. This work describes the programming philosophy and code structure used to create EFIT-CompCell and compares its performance and capacity to simulation tools that are more specialized for specific architectures. The EFIT-CompCell code described here has been developed with the target application of GW SHM simulations for aerospace structures. In particular, NASA Langley is partnering with the FAA to study capabilities and limitations of GW SHM for aircraft fuselage components. Results are presented for EFIT-CompCell simulation of GW SHM of an aluminum fuselage section being tested by the FAA.

KERNEL DESCRIPTION

The EFIT custom code, previously developed at NASA Langley Research Center by the Computation Nondestructive Evaluation (CNDE) team, was intended provide a relatively fast, customizable, and versatile ultrasound simulation tool. The code was written in C++ with MPI for parallelization across processors. The previously developed code was not optimized and did not efficiently simulate complex geometries.

EFIT becomes unstable when spatial and time step sizes are not properly selected, therefore implementing EFIT for realistically sized aerospace components requires that the technique be parallelized. The code must also be able to accurately simulate wave interaction with various boundary conditions and behavior in complex geometry components. The previously developed code for isotropic EFIT was implemented in parallel and was validated against both experiment and theory [2,7-8].

In the Spring of 2017 under the High Performance Computing Incubator (HPCI) Project at NASA Langley, the CNDE team began work to create an optimized EFIT code. Significant changes were made to modularize the previous implementation and move to an API structure. The new structure uses a C++ objects to specialize the computational cell class that gives EFIT-CompCell its name. I/O was changed from a collect/write format to a MPI I/O format and input was changed to xml, human readable inputs. The optimized EFIT-CompCell tool structure is outlined below.

Tool Structure

The EFIT-CompCell simulation tool can be thought of as containing two stages. The first stage consists of all the pre-simulation steps required to set up the code. The second stage consists of the computational loop that runs the elastodynamic calculations used in ultrasonic wave propagation. The pre-simulation stage contains many more case-specific calculations but makes fewer total calculations compared to the simulation phase.

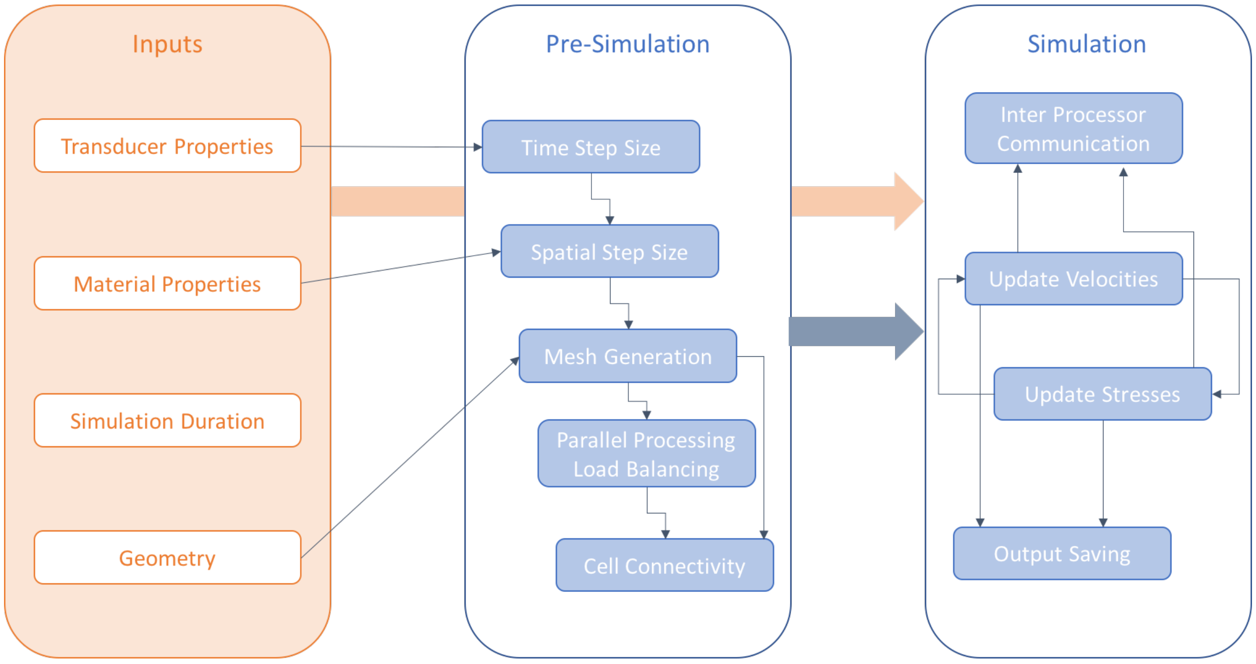


Figure 1 - EFIT-CompCell Functional Flow Block Diagram.

Figure 1 shows the functional flow block diagram for the two stages. By taking a modular approach to the entire application, changes can be made in relative isolation without affecting the other code elements (i.e., mini-applications). Likewise, specialized elements can be substituted for the general elements in specific cases without a wholesale redesign of the entire application. Computational kernels can even be swapped allowing other elastodynamic kernels to use the same preprocessing and simulation inputs.

The pre-simulation stage is an example of the code modularization philosophy mentioned earlier. The user provides transducer frequency, material properties, and component geometry (as an STL file), as well as simulation specific details such as simulation duration and transducer placement. A python mini-application generates the ultrasound excitation function and determines the ideal spatial and temporal step size.

A second mini-application takes the STL file and the spatial step size and creates the geometry mesh input necessary for the simulation. This meshing application uses ray intersection tools in the Visualization Tool Kit (VTK) library and MPI to mesh any volume.

The EFIT mathematical kernel is written in C++ with use of MPI. The EFIT equations for velocity are:

$$\Delta V_x = \frac{\Delta t}{\Delta n \frac{(\rho+\rho[x+1])}{2}}\left(S_x[x+1] - S_x + T_{xy} - T_{xy}[x-1] + T_{xz} - T_{xz}[x-1]\right) + F_x$$

$$\Delta V_y = \frac{\Delta t}{\Delta n \frac{(\rho+\rho[y+1])}{2}}\left(S_y[y+1] - S_y + T_{yz} - T_{yz}[y-1] + T_{xy} - T_{xy}[y-1]\right) + F_y, \qquad \textbf{Equation 1}$$

$$\Delta V_z = \frac{\Delta t}{\Delta n \frac{(\rho+\rho[z+1])}{2}}\left(S_z[z+1] - S_z + T_{xz} - T_{xz}[z-1] + T_{yz} - T_{yz}[z-1]\right) + F_z$$

S refers to normal stress and *T* refers to shear stress. ρ is density. *Δn* refers to spatial step size and *Δt* is the time step size. Finally, *F* refers to any external input, in this case the ultrasound excitation function converted into velocities. The square brackets refer to the parameter at its *x, y, or z* neighbor.

The EFIT equations for normal and shear stress are:

$$\Delta S_x = \frac{\Delta t(\lambda+2\mu)}{\Delta n}(V_x - V_x[x-1]) + \frac{\Delta t\lambda}{\Delta n}\left(V_y - V_y[y-1]\right) + \frac{\Delta t\lambda}{\Delta n}(V_z - V_z[z-1]),$$

$$\Delta S_y = \frac{\Delta t(\lambda+2\mu)}{\Delta n}\left(V_y - V_y[y-1]\right) + \frac{\Delta t\lambda}{\Delta n}(V_x - V_x[x-1]) + \frac{\Delta t\lambda}{\Delta n}(V_z - V_z[z-1]), \qquad \textbf{Equation 2}$$

$$\Delta S_z = \frac{\Delta t(\lambda+2\mu)}{\Delta n}(V_z - V_z[z-1]) + \frac{\Delta t\lambda}{\Delta n}(V_x - V_x[x-1]) + \frac{\Delta t\lambda}{\Delta n}\left(V_y - V_y[y-1]\right),$$

$$\Delta T_{xy} = \frac{4\Delta t}{\Delta n\left(\frac{1}{\mu}+\frac{1}{\mu[x+1]}+\frac{1}{\mu[y+1]}+\frac{1}{\mu[z+1]}\right)}\left(V_y - V_y[x-1] + V_x - V_x[y-1]\right),$$

$$\Delta T_{xz} = \frac{4\Delta t}{\Delta n\left(\frac{1}{\mu}+\frac{1}{\mu[x+1]}+\frac{1}{\mu[y+1]}+\frac{1}{\mu[z+1]}\right)}(V_z - V_z[x-1] + V_x - V_x[z-1]), \qquad \textbf{Equation 3}$$

$$\Delta T_{yz} = \frac{4\Delta t}{\Delta n\left(\frac{1}{\mu}+\frac{1}{\mu[x+1]}+\frac{1}{\mu[y+1]}+\frac{1}{\mu[z+1]}\right)}\left(V_z - V_z[y-1] + V_y - V_y[z-1]\right),$$

where λ and μ are the first and second Lamé constants. Equations are based on the method described in [3,4].

The EFIT code uses MPI to pass updated stress and velocity components between neighboring processors, as required. Therefore, the last two steps of the preprocessing phase are to ensure that each MPI process has approximately the same number of computational cells on which to operate (for computational load balancing) and that all cells are properly connected to their neighbors.

The input geometry mesh created is a rectangular volume that encompasses the entire geometry and is saved in a file as unsigned 8-bit integer values that describe the contents of each cell, 0 for a void cell (i.e., vacuum) and a non-zero number designated to correspond to a specific material. An instance of the EFIT-CompCell class array is created with one element for every non-void cell (i.e., void cells do not add to the computational time). The geometry file is used to define the boundary conditions and which cells are neighbors (based on the mathematical stencil which requires information from all neighboring cells). The simulation also uses a single array to hold all the velocities and stresses for the computational volume, which is computationally more efficient than constructing a 3D matrix for the computational volume. Instead of a standard stride for neighboring cells, which would only work well for rectilinear grids, EFIT-CompCell keeps track of its neighbors using pointers to the neighboring stresses and velocities. Once set, these pointers do not change during the simulation. Data is written to pointer locations by an update step or an inter-process communication (elements on a processor boundary) and is read by its neighbors for calculation.

CompCells have one member function update the velocities (Equation 1) and one that update the Stresses (Equations 2 and 3) at each step. The IF-THEN statements are handled implicitly for boundary conditions because each CompCell uses an unsigned 16-bit integer where each bit flag describes one boundary. This boundary integer is then used to set the pointers described above based on the bitwise boundary definitions of Table 1. If a neighbor does not exist then it points to a zero value, nullifying that component of the calculation. If it has a transducer on one of its faces then it points to the drive function.

Table 1 - Boundary condition bitwise definition.

bit	Use	bit	Use
0	+X neighbor exists	8	+YZ neighbor exists
1	-X neighbor exists	7	Transducer is on +X face
2	+Y neighbor exists	10	Transducer is on -X face
3	-Y neighbor exists	11	Transducer is on +Y face
4	+Z neighbor exists	12	Transducer is on -Y face
5	-Z neighbor exists	13	Transducer is on +Z face
6	+XY neighbor exists	14	Transducer is on -Z face
7	+XZ neighbor exists	15	unused

Validation

Validation was performed by comparing results to the Lamb Wave group velocity on a uniform plate of the same thickness. The EFIT-CompCell code results were also compared to results from the initial EFIT code version which has been validated against experiment and theory [2,7-9]. The simulation group velocity was computed by tracking the movement of the maximum of the Hilbert envelope of the larger-magnitude part of solution. This result was then compared to the expected group velocity from the dispersion curves at the appropriate frequency-thickness [10]. The simulation group velocity was found to be 2653 m/s for a case of 200 kHz excitation frequency and 1.6 mm plate thickness. This result had an error of 0.822% compared to the theoretical group velocity, 2675 m/s calculated for this geometry and material.

Code Performance Studies

Strong scaling studies were conducted on the algorithm itself for several flat plate cases. Three cases were investigated, a 17 million cell case (1024 x 1024 x 16 cells), a 67 million cell case (2048 x 2048 x 16), and a 270 million cell case (4096 x 4096 x 16). These cases were run on the Pleiades Haswell system (NASA Ames Research Center's Petascale supercomputer) for one to five compute nodes, with 24 cores per node (two sockets, 12 cores per socket). Strong scaling demonstrates, for a given computational domain size, whether more processors working the calculation will yield a faster time to solution. Strong scaling is important where the required simulation domain is fixed(i.e., necessary to perform some useful task), but improvements in runtime to solution are desired or required.

For this broad span of problem sizes, linear strong scaling was observed for EFIT-CompCell, as demonstrated in Figure 2 As fractions of single node performance, the execution time trends as 1/node count. The 270 million cell problem did not fit within the memory of a single node, and was therefore only run on two to five nodes. This case is therefore scaled by twice the two node execution time to a facilitate a direct comparison among all cases. These results demonstrate effective strong scaling behaviors over a broad suite of problem sizes, which is desired for good computational performance across a wide range of problems of scientific interest.

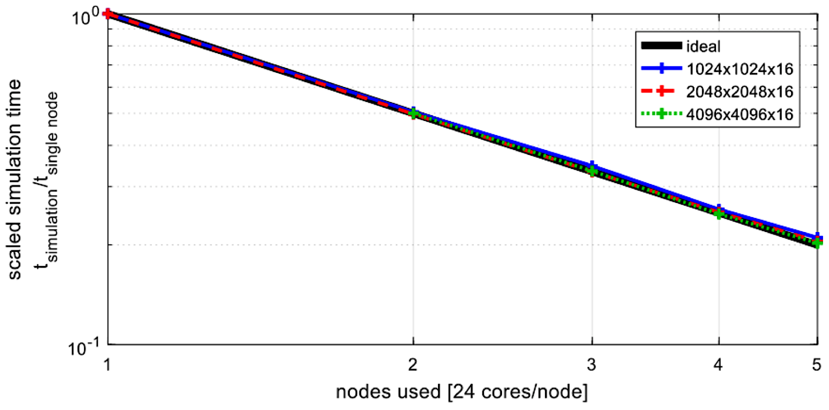
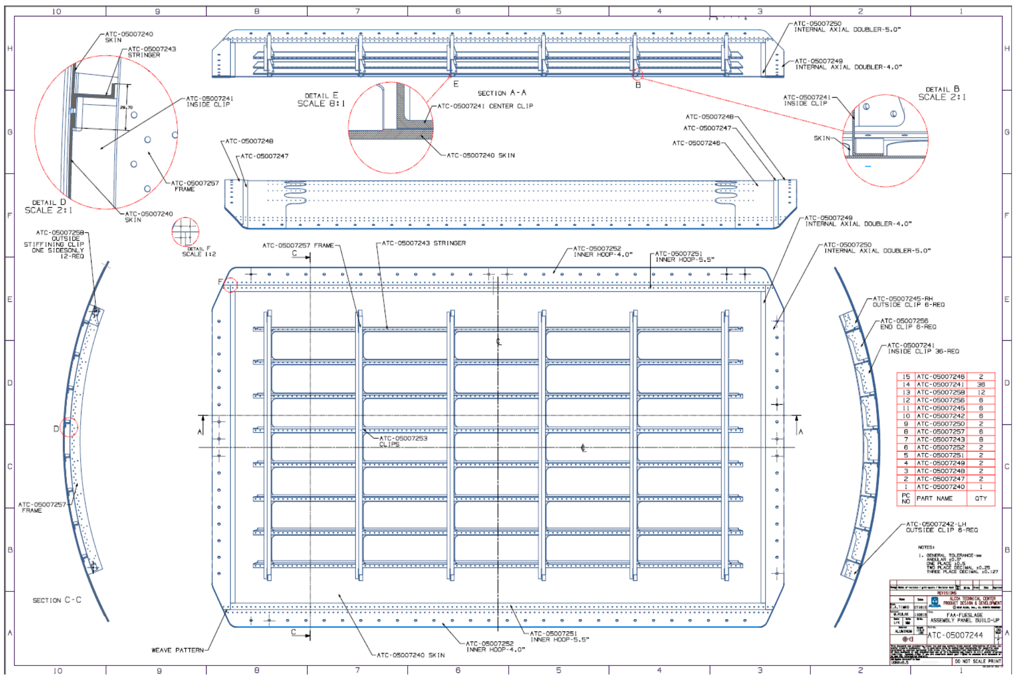


Figure 2 - Strong scaling results.

EXAMPLE SHM SIMULATION CASES

The FAA is currently collaborating with a commercial SHM company to perform GW SHM system testing on a representative aircraft fuselage panel. The CNDE team will run simulations of the test setup to enable further exploration of the SHM system capabilities/limitations than is feasible through only experiment. This simulation-assisted approach for testing GW SHM has previously been demonstrated in [2].

Figure 3 shows the geometry of panel being tested by the FAA and Figure 4 shows the geometry used in the EFIT-CompCell simulation. The panel is 3.175 m (125 in) long by 1.85 m (73 in) wide. The skin is made of Aluminum 2024-T3, the stringers are 7150-T7511, and the frames are 7075. Preliminary test simulations determined that little influence was exerted by structure beyond the bays (the space between two frames and two stringers) where the piezoelectric excitation source was located. Therefore, the simulation volume only contains two of the bays.


ATC-05007240 SKIN
ATC-05007243 STRINGER
ATC-05007241 INSIDE CLIP
DETAIL E SCALE 8:1
SECTION A-A
ATC-05007241 CENTER CLIP
ATC-05007240 SKIN
ATC-05007250 INTERNAL AXIAL DOUBLER-5.0"
ATC-05007249 INTERNAL AXIAL DOUBLER-4.0"
DETAIL B SCALE 2:1
ATC-05007241 INSIDE CLIP
SKIN
ATC-05007248
ATC-05007247
ATC-05007246
ATC-05007257 FRAME
DETAIL D SCALE 2:1
ATC-05007240 SKIN
ATC-05007258 OUTSIDE STIFFINING CLIP ONE SIDESONLY 12-REQ
DETAIL F SCALE 1:2
ATC-05007257 FRAME
ATC-05007243 STRINGER
ATC-05007252 INNER HOOP-4.0"
ATC-05007251 INNER HOOP-5.5"
ATC-05007245-RH OUTSIDE CLIP 6-REQ
ATC-05007256 END CLIP 6-REQ
ATC-05007241 INSIDE CLIP 36-REQ
ATC-05007253 CLIPS
ATC-05007242-LH OUTSIDE CLIP 6-REQ
ATC-05007257 FRAME
SECTION C-C
WEAVE PATTERN
ATC-05007240 SKIN
ATC-05007252 INNER HOOP-4.0"
ATC-05007251 INNER HOOP-5.5"
ALCOA TECHNICAL CENTER PRODUCT DESIGN & DEVELOPMENT
FAA-FUSELAGE ASSEMBLY PANEL BUILD-UP
ATC-05007244
DO NOT SCALE PRINT

15	ATC-05007246	2
14	ATC-05007241	36
13	ATC-05007258	12
12	ATC-05007256	6
11	ATC-05007245	6
10	ATC-05007242	6
9	ATC-05007250	2
8	ATC-05007257	6
7	ATC-05007243	8
6	ATC-05007252	2
5	ATC-05007251	2
4	ATC-05007249	2
3	ATC-05007248	2
2	ATC-05007247	2
1	ATC-05007240	1
PC NO	PART NAME	QTY

NOTES:

1. GENERAL TOLERANCE-mm
   ANGULAR ±0.5°
   ONE PLACE ±0.5
   TWO PLACE DECIMAL ±0.25
   THREE PLACE DECIMAL ±0.127

Figure 3 – Drawing of FAA fuselage test panel.

The simulation, using a 200 kHz excitation signal, was conducted both on the pristine geometry shown in Figure 3 and a second case that included a vertical cut through the middle of the center stringer. The cut is only 0.1 mm wide and 60 mm long. The nominal skin thickness is 1.6 mm. The transducer is located on the inner wall of the skin, near the cut. Table 2 contains the material properties for Aluminum 2024 used in the simulation and Table 3 contains the simulation parameters.

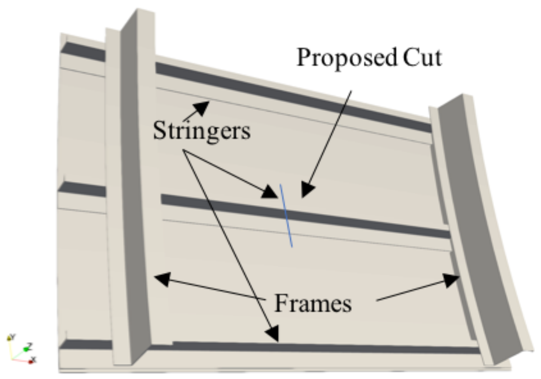


Figure 4 – Simulation geometry for 2 of the fuselage bays, including 2 frame sections, 3 stringer sections, and 4 shear ties.

Table 2 – Aluminum 2024 Properties.

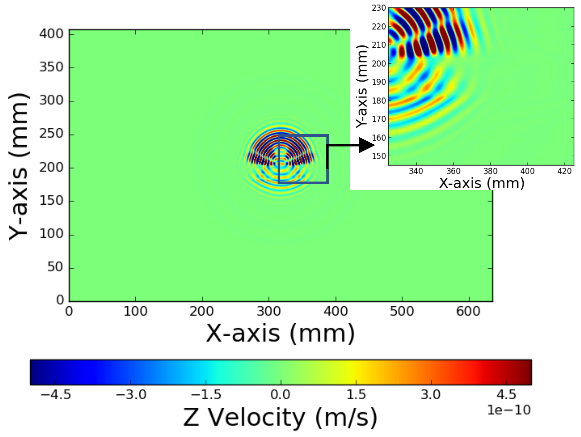
Aluminum 2024	
Longitudinal Velocity, c_L	6148 m/s
Transverse Velocity, c_T	3097 m/s
Density, ρ	2700 kg /m^3
Lamé first parameter, λ	51.8 GPa
Lamé second parameter, μ	26.7 GPa

Table 3 – Simulation Parameters.

Time step size	7.9 ns
Spatial step size	0.1 mm
Save frequency	50 time steps
Duration	25000 time steps

Simulation Results

The simulations were completed with 40 CPU processors and took 70 hours and 12 minutes to complete 25000 time steps. The simulation consisted of more than 75 million computational cells.



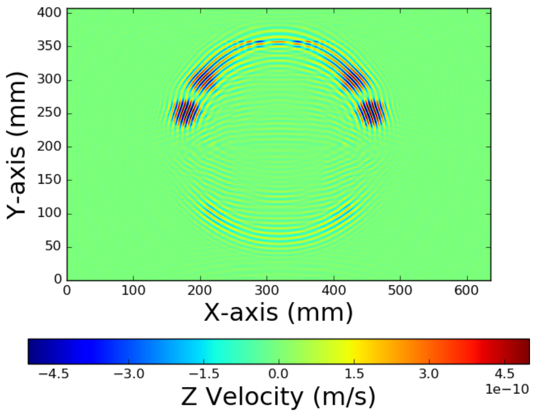


Figure 5 – Z-component of Velocity for the smooth, outer surface of the panel at 25.28 μs and 71.1 μs for a pristine panel.

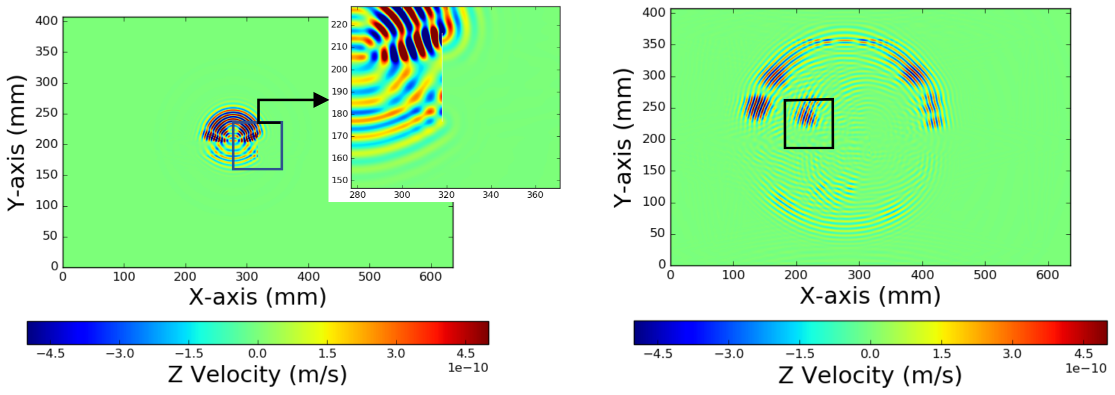


Figure 6 – Z-component of Velocity for the smooth, outer surface of the panel at time steps 25.28 µs and 71.1 µs for a Panel with a cut in the Y axis through the center stinger and skin.

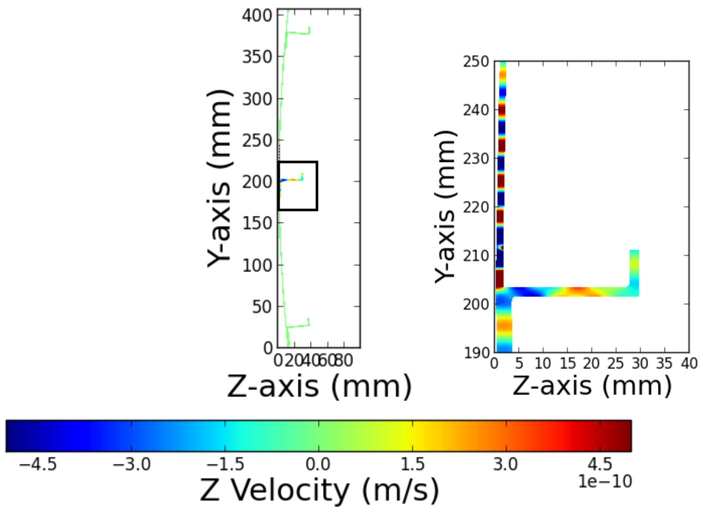


Figure 5 shows the *z* velocity component results visualized on the back surface of the computational volume for the pristine case. By comparison, the results in Figure 6 are for the case with the center vertical stringer cut. GW scattering due to the vertical cut can be observed in Figure 6. The cut is visible when the wave hits it at time step 25.28 µs and an "echo" is seen at time step 71.1 µs in the black box. Figure 7 shows an x-axis slice of the pristine case at 19.75 µs. The sub-frame shows how the wave moves through the stringer.

Figure 7 - Center x-axis slice at 19.75 µs.

CONCLUSIONS

The simulation framework presented in this paper provides a simple, versatile, modular method for ultrasound simulation in complex geometry parts. The modular nature of the framework allows for changes and improvements to be made to the pre-simulation stage and even provides the opportunity for substituting different mathematical solvers. Future work will entail further optimization of the code and further application of the code to GW SHM cases. The example cases presented here, simulations of FAA test panel, provide evidence that the framework is conducive to simulation of ultrasound in real-world sized complex geometry components. In the near future, simulations will be implemented with sensor locations matching those used in the FAA SHM system test setup. Simulations results will be used to gain knowledge of GW SHM system capabilities and limitations. By augmenting SHM with modeling tools it is possible to build a more detailed image of damage and how it evolved that merely that damage does exist. Modeling provides engineers with the ability to connect specific SHM signals with damage geometry.

ACKNOWLEDGEMENTS

We would like to thank the HPCI interns, by Anthony Williams of then ODU now at LaRC, Thomas Nabelek of U. Missouri, and Frank Schoeneman of U. Buffalo who worked on the scaling study. We also thank Yongzhe Tian and John Bakuckas from the FAA for help getting correct details for the fuselage test component geometry.

REFERENCES

1. Raghavan, Ajay, and Carlos ES Cesnik. "Review of guided-wave structural health monitoring." *Shock and Vibration Digest* 39.2 (2007): 91-116.
2. Yu, Lingyu, and Cara AC Leckey. "Lamb wave–based quantitative crack detection using a focusing array algorithm." *Journal of Intelligent Material Systems and Structures* 24.9 (2013): 1138-1152.
3. Fellinger, F. and Langenberg, K.J. "Numerical techniques for elastic wave propagation and scattering," *Elastic Waves and Ultrasonic Nondestructive Evaluation* (1990): 81-86.
4. Marklein, R. "The Finite Integration Technique as a General Tool to Compute Acoustic, Electromagnetic, Elastodynamic, and Coupled Wave Fields." *Review of Radio Science* (1999).
5. Schmerr, Lester W. "Fundamentals of Ultrasonic Nondestructive Evaluation: A modeling approach", New York: Plenum Press. (1998)
6. Leckey, Cara AC, Jeffrey P. Seebo, and Peter Juarez. "Challenges of NDE simulation tool validation, optimization, and utilization for composites." *AIP Conference Proceedings*. 1706.1 (2016).
7. Yu, Lingyu, Zhenhua Tian, and Cara AC Leckey. "Crack imaging and quantification in aluminum plates with guided wave wavenumber analysis methods." *Ultrasonics* 62 (2015): 203-212
8. Leckey, Cara AC, et al. "Multiple-mode Lamb wave scattering simulations using 3D elastodynamic finite integration technique." *Ultrasonics* 52.2 (2012): 193-207.
9. Tian, Zhenhua, Lingyu Yu, and Cara Leckey. "Delamination detection and quantification on laminated composite structures with Lamb waves and wavenumber analysis." *Journal of Intelligent Material Systems and Structures* 26.13 (2015): 1723-1738.
10. B. Pavlakovic and M. Lowe, "Disperse: A general purpose program for creating dispersion curves", in Review of Progress in Quantitative Nondestructive Evaluation, 16A (1997): 185-192.