OXBOW: A MULTI-FIDELITY SUITE OF ANALYSIS TOOLS FOR SFR CORE MECHANICAL DESIGN

Alexander H. James[1], Jason B. Meng[2], Jeffrey S. Baylor[3]

[1] Principal Engineer, Core Mechanical Methods and Analysis, TerraPower LLC, Bellevue, WA, USA (ajames@terrapower.com)
[2] Principal Engineer, Core Mechanical Methods and Analysis, TerraPower LLC, Bellevue, WA, USA (jmeng@terrapower.com)
[3] Principal Engineer, Core Mechanical Methods and Analysis, TerraPower LLC, Bellevue, WA, USA (jbaylor@terrapower.com)

ABSTRACT

TerraPower, LLC. has developed the OXBOW suite of mechanical analysis tools, which is part of the TerraPower core multi-physics software ecosystem, to enable different types of core mechanical analysis at varying levels of fidelity. This paper describes the high-level design of OXBOW and how its capabilities enable continuity between different core mechanical analyses. OXBOW is built with interfaces to the TerraPower-developed Advanced Reactor Modeling Interface (ARMI®) which is an open-source reactor modeling framework (Touran, et al 2017). The ARMI® reactor model contains the reactor state information (geometry, materials, temperatures, flux distributions, etc.) and is the centralized location where TerraPower core multi-physics codes pass state information for coupled analysis of neutronics, thermal hydraulic, and mechanical aspects of a reactor core. Select analysis packages contained in OXBOW are summarized in this paper along with three case studies that demonstrate the flexibility of OXBOW for different mechanical analysis needs.

INTRODUCTION

The mechanical analysis of a sodium fast reactor (SFR) core involves multiple levels of modeling fidelity and the need to transfer results from an analysis at one fidelity to one of a different level of fidelity. For example, modeling of the entire reactor core requires a lower fidelity model for computational efficiency, but the results of that model are required as inputs to a higher fidelity model of a single assembly or component. In a typical reactor development program, those analyses are performed by different groups of people with an analysis specific data handoff between the groups. The two models on either side of the possible interface are constructed only with regard for the analysis they are performing and not for the ease of results transfer between them. For example, the Fast Flux Test Facility (FFTF) used the unrelated analysis codes NUBOW3D (Argonne National Lab 1978 for core restraint system analysis and CRPBOW (Hanford Engineering Development Laboratory 1977) for detailed control assembly inelastic analysis. TerraPower, LLC. has developed the OXBOW suite of mechanical analysis tools, which is part of the TerraPower core multi-physics software ecosystem, to enable and automate different types of core mechanical analysis at varying levels of fidelity.

OXBOW utilizes reactor state information to generate finite element models of different fidelities for various mechanical analyses as described in Baylor, et al 2018. These models range from core motion in a seismic analysis, to detailed stress models for evaluation of deformation in high residence core assemblies, to analysis of core assembly handling loads required to perform assembly shuffling operations. OXBOW is designed so that analysis results can be passed between OXBOW analyses even if they are of a different fidelity. From a software standpoint, OXBOW is intended to be flexible to support any reactor core analysis and is agnostic to finite element solver.

SOFTWARE ECOSYSTEM

OXBOW is built on top of a general finite element analysis (FEA) pre- and post- processing package and has interfaces to the TerraPower-developed Advanced Reactor Modeling Interface (ARMI®) which is an open-source reactor modeling framework (Touran, et al 2017). TerraPower's general FEA package is capable of building an object-oriented version of a finite element model and then interfacing with the chosen finite element solver. OXBOW combines user input information with data from the ARMI® reactor model to build a finite element model with the FEA package.

The ARMI® reactor model contains state information (geometry, materials, temperatures, flux distributions, etc) and is the centralized location where TerraPower core multi-physics codes pass reactor state information for coupled analysis of neutronics, thermal hydraulic, fuel performance, and mechanical aspects of the core. A schematic diagram of the ARMI® ecosystem is shown in Figure 1. The reactor definition is stored in a single set of inputs that is shared between analysis codes allowing for consistency between analyses as well as reducing rework associated with creating multiple input files. While the ARMI® ecosystem supports tight coupling between physics codes, OXBOW is loosely coupled with the other physics codes. OXBOW reads the results of other relevant physics codes, performs mechanical analysis, and then populates the mechanical results back onto the ARMI® data model for downstream analysis use.

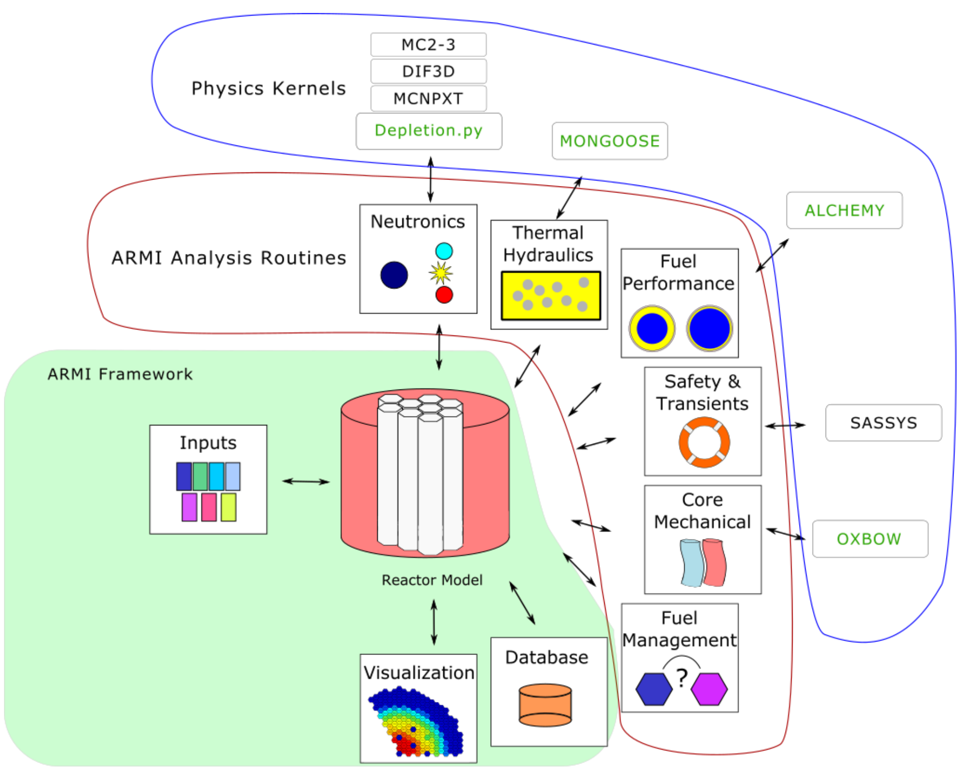

Figure 1. ARMI® Analysis Framework and Multiphysics Interfaces. See https://terrapower.github.io/armi/ for more details.

SOFTWARE DESIGN

The general FEA package that underlies OXBOW builds an object-oriented finite element model with each finite element concept represented by an object. Each object has ownership over a set of properties and can perform actions during pre- and post- processing. The hierarchy of the FEA objects is shown in Figure 2.

At the highest level, a *Model* object has all the information about a specific finite element analysis. Each *Model* object contains *Meshable Objects*, that store mesh and boundary condition information such as nodes, elements, surfaces, loads, and boundary conditions, *Step* objects that store analysis step-related information, and *Interaction* objects that describe how *Meshable Objects* interact with each other (for example through contact). Once a *Model* has been created, an interface can be established between that *Model* and a particular finite element solver. The interface interacts with the external finite element solver and loads the results onto the relevant objects for post processing after the analysis is complete. The same model can be analyzed with different solvers by linking it to different interfaces.

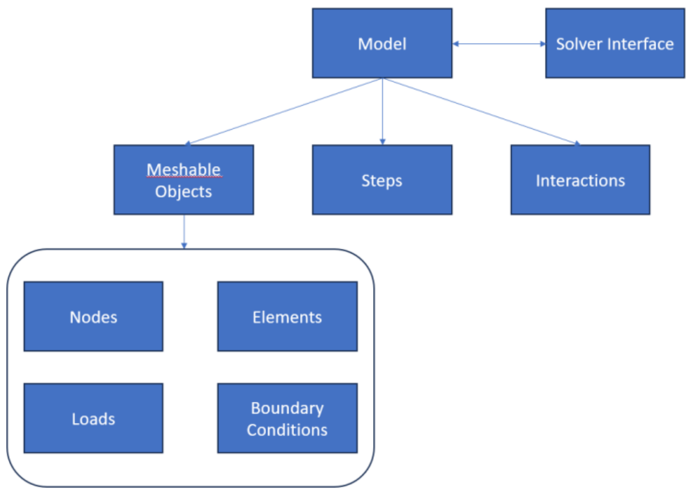

Figure 2. FEA package object hierarchy.

The power in this object-oriented approach is that the user can be shielded from the implementation details of each solver, data access and ownership is intuitive, and that developers can build new functionality without needing to be aware of all the implementation details in the rest of the code base. A code example of how a model can be built with the modeling package is shown in Figure 3. In this model, two nodes are created and connected with a beam element. The model has cantilever boundary conditions applied to one end and a transverse point load on the other end. Finally, the model is sent to Abaqus for analysis.

OXBOW builds upon this modeling package by creating reactor-related objects that inherit from the base modeling package objects and expand on their functionality. The object hierarchy in OXBOW is related to the object hierarchy in ARMI®. Each reactor core contains multiple assemblies that are then discretized axially into blocks, see Figure 4. The axial discretization is chosen so that there is a block boundary at every change in assembly cross section. A finer discretization may be chosen based on mesh refinement needs for the relevant physical analyses. While ARMI® discretizes each block into components, OXBOW discretization does not go beyond the block level as generally internal block components are not structurally significant. In cases where internal components are important, the OXBOW block handles the relevant meshing operations for all components in the block. In relation to the FEA modeling package, the reactor core model inherits from the *Model* object and each assembly is a *Meshable Object*. The core support structure is also a *Meshable Object*. While blocks do not have a corollary to a modeling package object, they can be thought of as meshing aids for the *Meshable Object* assemblies. This choice of relationship between ARMI® objects and modeling package objects is intuitive because it groups each physically connected object into *Meshable Objects* that can then be interacted at the core model level.

```python
class ExampleBeam(MeshableObject):
    """This is a class that makes a cantilever beam"""

    def __init__(self):
        self.fixNode = None # Placeholder for the cantilevered end of the beam
        self.loadedNode = None # Placeholder for the loaded end of the beam

    def mesh(self):
        self.fixNode = self.makeNode(x=0, y=0, z=0)  # Create a node at (0, 0, 0)
        self.loadedNode = self.makeNode(x=1, y=0, z=0)  # Create a node at (1, 0, 0)

        material = MaterialModel("exampleMaterial")  # Create a material.
        material.setIsotropicElasticity(
            youngs=10, poisson=0.3
        )  # Set the material elastic constants
        section = BeamElementSection(
            "exampleSection", material
        )  # Create a beam section from that material
        section.setHexCrossSection(
            radius=5, thickness=1
        )  # Set the cross section as a hex.

        # create a beam element that connects the two nodes and assign it the created cross section.
        self.makeElement([self.fixNode.nID, self.loadedNode.nID], elemType="B31", section=section)

    def loadBeam(self):
        self.makeBC(
            self.fixNode, 1, 6
        )  # Constrain the first node to be fixed in all degrees of freedom
        # Create a transverse concentrated load at the second node in the first analysis step
        self.addLoad(
            ConcentratedLoad(load=1, degreeOfFreedom=2, node=self.loadedNode), "Step 1"
        )


model = Model()  # Create a model object
meshableObject = ExampleBeam()  # Create a meshable object that is our example beam
model.addMeshableObject(meshableObject)  # Add the meshable object to the model

meshableObject.mesh()  # Mesh the beam

model.addStaticStep()  # Add a static load step to the analysis

meshableObject.loadBeam() # Apply loads and boundary conditions

abqInterface = AbaqusInterface(model)  # Create an interface to Abaqus for the model

# Now abqInterface can be used to share the model with the solver and get back results.
```

Figure 3. Cantilever beam example code.

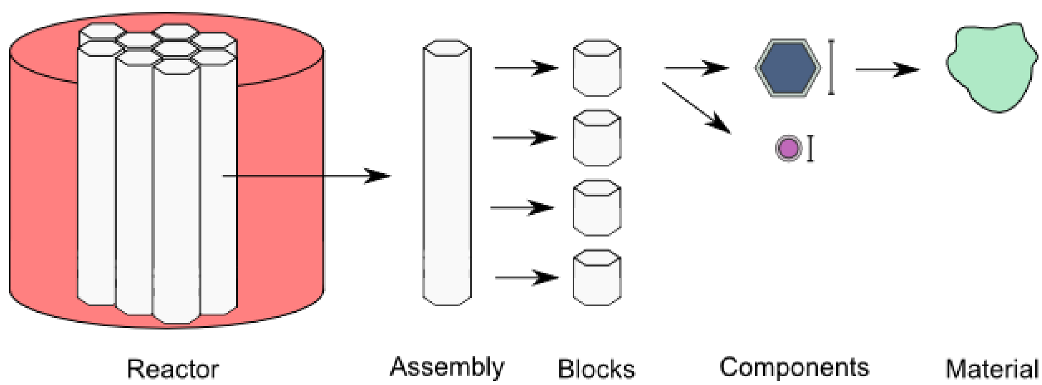

Figure 4. ARMI® data model. See https://terrapower.github.io/armi/developer/guide.html#the-reactor-model.

Currently, the FEA modeling package supports interfaces to Abaqus and Ansys for choice of finite element solver. This flexibility allows the analyst to change finite element solver based on project constraints without having to develop a new model. For example, if an analysis is updated to use features that are outside the scope of the commercial grade dedication of Abaqus, the analysis could be switched to Ansys. In addition to Abaqus and Ansys, MCNP has been identified as a future development project. The similarity between MCNP and Abaqus for mesh specification along with the ability of the modeling package to generate meshes for parametrized shapes of interest makes MCNP an ideal candidate for future capability expansion. Other possible future solver interfaces include Code_Aster, an open-source FEA solver developed by Électricité de France, and MOOSE, developed by Idaho National Laboratory.

The post processing capabilities in OXBOW are generally based around calculating data to populate on the ARMI® model. Basic Python plotting is also supported to visualize results. An area of future development is to interface with ParaView for a solver independent visualization tool.

OXBOW CAPABILITIES

OXBOW's flexibility allows for building models with different element topologies such as beam, continuum shell, and solid elements to support reactor analysis. The different implementations in OXBOW share code and capabilities which allows for commonality between models. For example, all solid models share the same meshing algorithms so structured meshes can be created for known geometries, regardless of analysis specifics. Another example is that all OXBOW models process the same temperature input data off the ARMI® reactor model and the method of applying those temperatures to the model is standardized for beam and solid models respectively. The utility of these common implementations is most apparent in the third case study discussed in this paper. The following sections describe various analysis implementations of OXBOW and identify linkages between them.

OXBOW.CRS – Core Restraint System Analysis

At operating conditions, SFR core assemblies are subjected to thermal and irradiation loads which result in elastic and inelastic deformation of core assemblies. OXBOW.CRS models the whole reactor core to determine assembly deflections, inelastic residual deformations, and contact loads between assemblies and between assemblies and other core components.

The OXBOW.CRS model can be created with each assembly represented by a collection of beam elements, or with each assembly fully meshed with solid elements. Temperature and dose fields are automatically read off the ARMI® reactor model and applied to the model. Multi-cycle analysis that includes inelastic deformations is possible via a set of automated core assembly shuffling boundary conditions. The multi-cycle capabilities allow for investigation of how the load path develops at different points in the reactor life. Typically, the core restraint model is the starting point of all OXBOW analyses and provides assembly deflections for downstream analyses to use as boundary conditions. The results can also be used directly to calculate reactivity due to radial feedback, or reaction loads on core support structures.

OXBOW.SEISMIC – Core Seismic Analysis

SFR reactors are sensitive to seismic excitations because vibrational displacement of core assemblies leads to reactivity swings. Additionally, displacement of control assemblies can lead to an increase in control rod insertion times. OXBOW.SEISMIC simulates the whole reactor core with a dynamic analysis to model the assembly deflections when subjected to seismic accelerations.

The OXBOW.SEISMIC models each assembly with a mesh of beam elements. The seismic excitation is applied via boundary conditions to the grid plate, core former ring, and core barrel. Fluid structure interaction is accounted for between assemblies and the core barrel. Assembly to assembly interaction is modeled with springs and dashpots. A core restraint system analysis is used to provide an initial deformed state. The results of an OXBOW.SEISMIC model can be used in a downstream OXBOW.CASS analysis or they can be used directly to calculate reactivity change in the core as a function of time due to seismic excitation, or to calculate impact loads on assembly load pads and core support structure.

OXBOW.CADA – Core Assembly Distortion Analysis

When core assemblies are subjected to stresses, irradiation, and high temperatures there will be inelastic deformations that develop. While a core restraint system analysis can calculate those deformations from a core-wide perspective, it does not capture more detailed assembly deformations. For example, a beam model will exhibit bending related inelastic deformations, but it will not capture cross section deformations. OXBOW.CADA is designed to provide single assembly deformations as it is a high-fidelity solid element sub-model that is driven by an OXBOW.CRS model.

A single core assembly is modeled with solid elements to include all geometric features of its structural components and meshed with sufficient resolution to accurately capture the stress distribution. The assembly-assembly contact load history from a core restraint system analysis is then applied to the model along with consistent boundary conditions at the grid plate. The temperature and dose fields consistent with the core restraint model are applied as well. Internal fluid pressure loads from coolant flow are applied based on the ARMI® model data. The results of an OXBOW.CADA model are used as an input to downstream analyses that require an inelastically deformed solid mesh like OXBOW.CASS and OXBOW.WI.

OXBOW.CASS – Control Assembly SCRAM under Seismic

The control rod bundle inside control assemblies is rapidly inserted during a SCRAM event. During a seismic event, the outer duct deformations lead to increased friction between the control rod bundle and the outer duct during the bundle’s insertion. If the deformations are significant enough, it can lead to the control rod bundle binding due to greatly increased friction loads. OXBOW.CASS is a high-fidelity solid element dynamic model that simulates the control rod bundle motion to predict the time that it takes the bundle to reach full insertion during a seismic event.

The OXBOW.CASS model includes the outer duct, internal structural components, and the control rod bundle. Contact is modeled between the control rod bundle and the inner surface of the outer duct. Displacement boundary conditions are applied to the outer duct based on an OXBOW.SEISMIC analysis. The fluid interaction effects between the control rod assembly and the internal coolant flow are included via body loads or damping. The model can be used to provide the control rod bundle insertion time profile (displacement vs. time) as well as inform the impact loads at the end of the insertion.

OXBOW.WI – Core Assembly Withdrawal and Insertion

To support core operations and core performance, the assemblies in an SFR core are removed from the core and shuffled to a new location in the core or replaced by a fresh assembly during re-fueling outages. The assemblies can experience significant inelastic deformations through their lifetime making it challenging to remove them due to increased friction loads. Residual bowing and swelling results in contact between assemblies even at isothermal refueling temperatures. OXBOW.WI predicts the handling loads that are expected to withdraw or insert a core assembly into the core. It is a mixed solid element and beam element model that simulates the contact loads between adjacent assemblies when they are being inserted or withdrawn.

The assembly that is being handled and all its neighbor assemblies are modeled with solid elements. Other assemblies in the core are modeled with beam elements to improve computational efficiency. The reactor state from a core restraint analysis is used as the starting point of the model. The assembly is then withdrawn or inserted with solid surface contact enforced between it and neighboring assemblies. The results of OXBOW.WI can be used to inform handling load requirements and also used in conjunction with OXBOW.CRS to design the core restraint system.

CASE STUDY 1 – ABAQUS COMMERCIAL GRADE DEDICATION

A good demonstration of the general utility of the FEA package, which is developed under TerraPower's NQA-1 program, is from TerraPower's commercial grade dedication of the Abaqus finite element solver. As part of that dedication, many test cases are run to verify that Abaqus's calculations match expected solutions. These test cases consist of a finite element model input file that is run in Abaqus and a reference value for comparing to the output results. Generally, the test cases come from the Abaqus Verification Guide which provides simple models to verify specific features. To automate the pipeline of running and checking test cases, the FEA package is used to manage the process.

The Abaqus interface in the FEA package first fetches the input file using Abaqus fetch commands. Then, it reads the input file and builds a Python representation of that finite element model. The input file is submitted to the Abaqus solver. Finally, the results are read by parsing the Abaqus output files for data. These results are stored in the relevant FEA Python objects and compared to expected solutions.

The flexibility of the FEA package is demonstrated in this case study with the wide range of Abaqus features that are encountered in these tests. At TerraPower, Abaqus is used for mechanical and fuel performance analyses, thus static mechanical, dynamic mechanical, heat transfer, and coupled mechanical-heat transfer analyses are covered by the dedication. The range of features supported allows the FEA package to support general one-off finite element analyses within the scope of use of the Abaqus commercial grade dedication.

CASE STUDY 2 – IWGFR BENCHMARK WITH ABAQUS AND ANSYS

The International Atomic Energy Agency's (IAEA) International Working Group on Fast Reactors (IWGFR), now the Technical Working Group on Fast Reactors (TWG-FR), facilitated a series of benchmarks for reactor core mechanical analysis codes from multiple countries. The benchmarks have varying complexity including single assembly, free-flowering core restraint, and limited free bow core restraint cases.

For this case study, IWGFR benchmark problem 4 (IWGFR 4) is selected which is a limited free-bow core with an applied temperature gradient (IAEA 1990). Using the problem definition in the report, an OXBOW.CRS model is created. The same OXBOW model is then solved by Abaqus and Ansys via their respective interfaces. Results of the two models are consistent and agree well with the benchmark. An image of the Abaqus model, the Ansys model, and the benchmark comparison are shown in Figure 5.

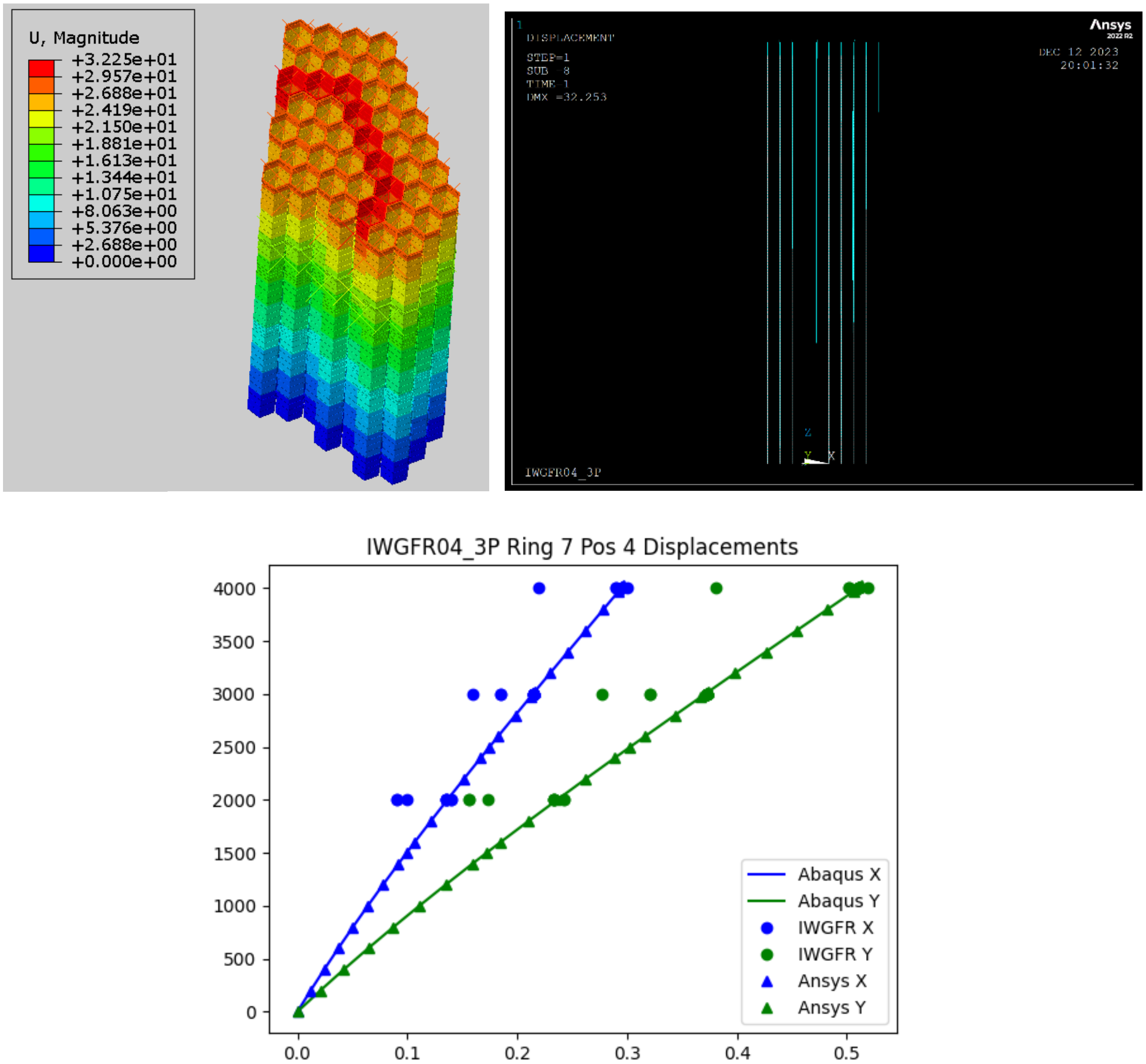

Figure 5. Abaqus IWGFR 4 (top left), Ansys IWGFR 4 (top right), and benchmark comparison for assembly at ring 7, position 4 (bottom).

Because Oxbow analyses are solver agnostic, a solver can be chosen based on analyst comfort or company restrictions. In this case, the difference between the two models is simply that a different interface was used to solve the model. This ability also allows for different analysts to look at the same model in different programs, helping to bridge the gap if they are only knowledgeable in the usage of one finite element program.

CASE STUDY 3 – CONTROL ASSEMBLY SCRAM UNDER SEISMIC ANALYSIS SEQUENCE

The final case study is to show the analysis flow that leads to an OXBOW.CASS analysis. The focus of this case study is on how analysis results are transferred between models and not on how each analysis is performed in detail. A control assembly SCRAM under seismic analysis has four pieces: an OXBOW.CRS analysis, an OXBOW.CADA analysis, an OXBOW.SEISMIC analysis, and finally an OXBOW.CASS analysis. See Figure 6 for a flowchart of the analysis flow. The core restraint results are used as initial state conditions for the core seismic analysis and are used as boundary conditions for the core assembly distortion analysis. The core assembly distortion analysis results are used as the initial conditions for the control rod insertion time analysis and the core seismic analysis provides the boundary conditions during the insertion.

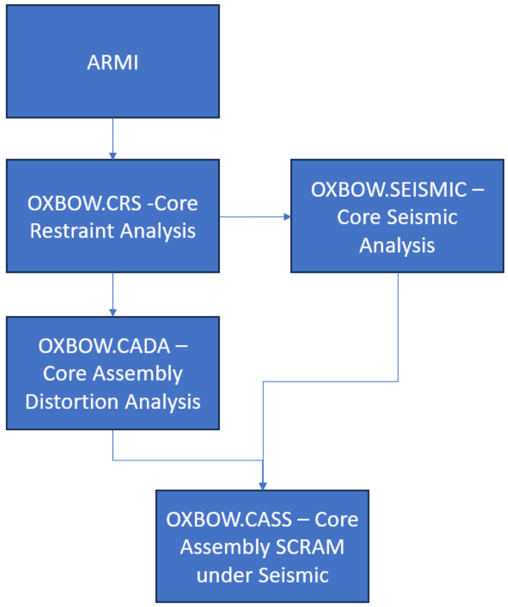

Figure 6. Control assembly SCRAM under seismic analysis sequence.

The first step in the analysis chain is to use the core restraint results for downstream analyses. There are connector elements in the core restraint model that enforce contact between neighboring assemblies. A time history of the load in these connector elements can be extracted and used as boundary conditions in the OXBOW.CADA model. At the end of the core restraint analysis, the connector elements can be deleted allowing the assemblies to deflect with residual creep deformation to their natural state with no external loads. Those deflected centerlines can be transferred to the OXBOW.SEISMIC analysis and imported as the initial shape of the assemblies prior to loading.

The second step in the analysis chain is to use the core seismic and core assembly distortion results in the OXBOW.CASS model. The OXBOW.CADA results can be directly moved to the OXBOW.CASS model because the two implementations share the same solid meshing scheme. The same mesh can be used for the core assembly distortion analysis as the control rod insertion time analysis. This allows the deformed shape to be imported from one analysis to the other in the solver. The core seismic results are the centerline displacements of the beams that represent each assembly versus time. These centerline displacements are the boundary conditions of the duct during the control rod insertion. To transfer results from the beam model to the solid model, the nodes at each elevation in the duct in the solid model are constrained with constraint equations to a central node. The seismic displacements are then applied as boundary conditions to that central node. See Figure 7.

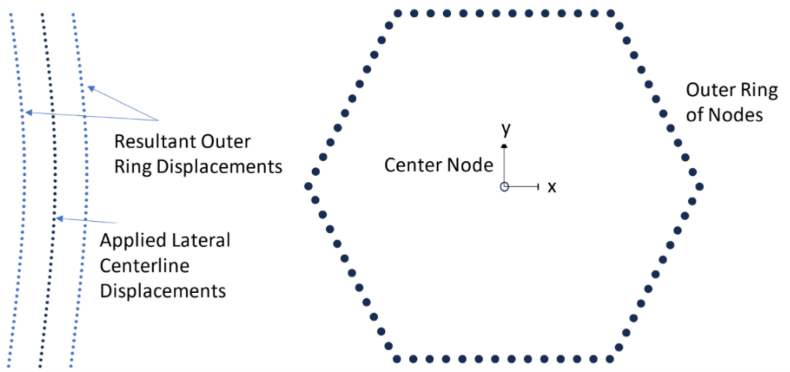

Figure 7. OXBOW.CASS constraint scheme

Because all the models in the OXBOW analysis suite are intended to function together, it is easy to transfer results from one model to another. There are standardized concepts for how to implement certain types of transference. For example, the ring of nodes constraint used to transfer results between OXBOW.SEISMIC and OXBOW.CASS, is also used to interact beam and solid assembles in OXBOW.WI. The harmony between models in the analysis suite allows for quicker analysis times as handoffs are standardized and reduces the chance for misinterpretation of the results after the results are handed off.

CONCLUSION

This paper describes the design of TerraPower, LLC. developed OXBOW suite of mechanical analysis tools, which is part of the TerraPower core multi-physics software ecosystem. OXBOW is built to enable different types of core mechanical analysis at varying levels of fidelity. It interfaces with the TerraPower-developed Advanced Reactor Modeling Interface (ARMI®) for reactor state information (geometry, materials, temperatures, flux distributions, etc.). Select analysis packages contained in OXBOW are summarized in this paper along with three case studies that demonstrate the flexibility of OXBOW for different mechanical analysis needs.

ACKNOWLEDGEMENTS

Subject to DOE cooperative agreement NO. DE-NE0009054.

REFERENCES

Argonne National Laboratory. (1978). *NUBOW3D (Inelastic): A FORTRAN Program for the Static Three-Dimensional Analysis of Bowed Reactor Cores Including the Effects of Irradiation Creep and Swelling*, ANL-CT-78-19.

Baylor, J., Choi, J., Touran, N., Werner, M., Cohen, M. (2018). “Seismic Analysis of a Full 3D Reactor Core Using Multi-Physics Modeling Methodology”, Top Fuel Reactor Fuel Performance Conference, Prague, CZ.

Hanford Engineering Development Laboratory. (1977). *CRBOW – A Program for Calculating Absorber Assembly Bowing*, HEDL-SA-888.

IAEA. (1990). “Verification and Validation of LMFBR Static Core Mechanics Codes”, Vienna, IWGFR/75

Touran, N. W., Gilleland, J., Malmgren, G. T., Whitmer, C., Gates III, W. H. (2017). “Computational Tools for the Integrated Design of Advanced Nuclear Reactors”, Engineering, Chinese Academy of Engineering, 3(4), 518-526.