ORNL/SPR-2023/3062

NEAMS Workbench MOOSE Integration Update

Robert A. Lefebvre
Brandon R. Langley
Mark L. Baird
Marc-Olivíer G. Delchini

September 2023

Document has been reviewed and approved for public release.

OAK RIDGE National Laboratory

ORNL IS MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

ORNL/SPR-2023/3062

Nuclear Energy and Fuel Cycle Division

NEAMS WORKBENCH MOOSE INTEGRATION UPDATE

Robert A. Lefebvre
Brandon R. Langley
Mark L. Baird
Marco-Olivíer G. Delchini

September 2023

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831
managed by
UT-BATTELLE LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

This work was supported by the Department of Energy – Nuclear Energy Advanced Modeling and Simulation Program (NEAMS) under the Multiphysics Applications Technical Area. Review and feedback from Derek Gaston (INL), Cody Permann (INL), Kyle Gamble (INL), Stephen Novascone (INL), Mauricio Tano-Retamales (INL), Emily Shemon (ANL), Kun Mo (ANL), Nicolas Stauff (ANL), Travis Mui (ANL), Rui Hu (ANL), Matthew Jessee, Eva Davidson, and the many anonymous user-survey respondents were very appreciated.

ABSTRACT

The Multiphysics Object Oriented Simulation Environment (MOOSE) Framework provides a foundation for scientists and engineers to extend and deploy application-specific solvers for multiphysics modeling and simulation (M&S) [1]. These problems can present complex analysis workflows that are difficult for non-developers to approach. The NEAMS Workbench is an activity in the US Department of Energy (DOE) Nuclear Energy Advanced Modeling and Simulation (NEAMS) campaign that aims to reduce these obstacles for users by deploying a common analysis and problem development environment for nuclear energy M&S that is integrated at the MOOSE framework and user levels [2]. This report documents updates to improve access and usability of MOOSE capabilities from within the NEAMS Workbench.

1. INTRODUCTION

The Nuclear Energy Advanced Modeling and Simulation (NEAMS) Workbench–MOOSE integration had two primary goals in fiscal year 2023. The first objective was to provide interactive native MOOSE problem diagnostics in Workbench via incorporation of the Workbench Analysis Sequence Processor's (WASP) [3] Language Server Protocol (LSP). The second objective was to improve MOOSE application feature accessibility within the Workbench graphical user interface (GUI). An ancillary objective was to engage users and address feedback, where feasible, with timely public Workbench releases and installations on Idaho National Laboratory's (INL's) Nuclear Computational Resource Center (NCRC) system.

2. MOOSE LANGUAGE SERVER INTEGRATION

The NEAMS Workbench supports the Microsoft LSP. This support enables applications such as Workbench and other integrated development environments (e.g., Visual Studio Code) to avoid a proliferation of integration logic and to focus on a single protocol, enabling user interactions with application inputs: for example, input component creation via autocompletion, input checking via interactive diagnostic messages, and document navigation via go-to definition and find-references.

The MOOSE Framework has been updated to include an extension of the WASP Language Server. Work has been completed to support MOOSE diagnostics, input autocompletion, and navigation functionality. The details of these improvements are documented in M2MS-23AN0701011[4]. This integration is the foundation on which MOOSE applications will allow application developers to respond to user requests for results visualization metadata and input component widget preferences. It builds upon past integration work that incorporated the WASP Hierarchical Input Text (HIT) language interpreter, including new HIT capabilities such as file includes [5].

3. WORKBENCH INTERFACE IMPROVEMENTS

The Workbench GUI has had seven releases, three of which were major version updates that included planned enhancements. A final major release is scheduled before the end of September 2023.

Primary features added to support MOOSE integration specifically are as follows. Absent from the list is the ability to use graphical widgets to create MOOSE inputs. The most effective way to achieve the incorporation of this widget feature and the ultimate objective of allowing MOOSE application developers to control widget options and layouts is by extending the LSP. This implementation eliminates

the need to reimplement MOOSE logic in the GUI, as has been done in the past. With the MOOSE language server coming online this year, it is expected that the extension to support GUI widgets will be viable in the coming year.

3.1 AUTOMATIC RESULTS MESH VISUALIZATION

Upon successful completion of a MOOSE application job, the resulting mesh is automatically loaded and visualized in Workbench. Previously, the user was required to drag-and-drop the mesh from a file browser onto Workbench or use the *Associated Files…* or *File > Open …* features.

3.2 DEFAULT PARAVIEW RENDERVIEW REPRESENTATION AND COLORING

When Workbench opens a mesh file, the default representation and coloring of the ParaView RenderView has been updated from *Surface* and *Solid* to *Surface with Edges* and *vtkBlockColors*. This reduces required interaction from the user when conducting mesh generation iterations where the mesh and associated blocks are to be visually verified.

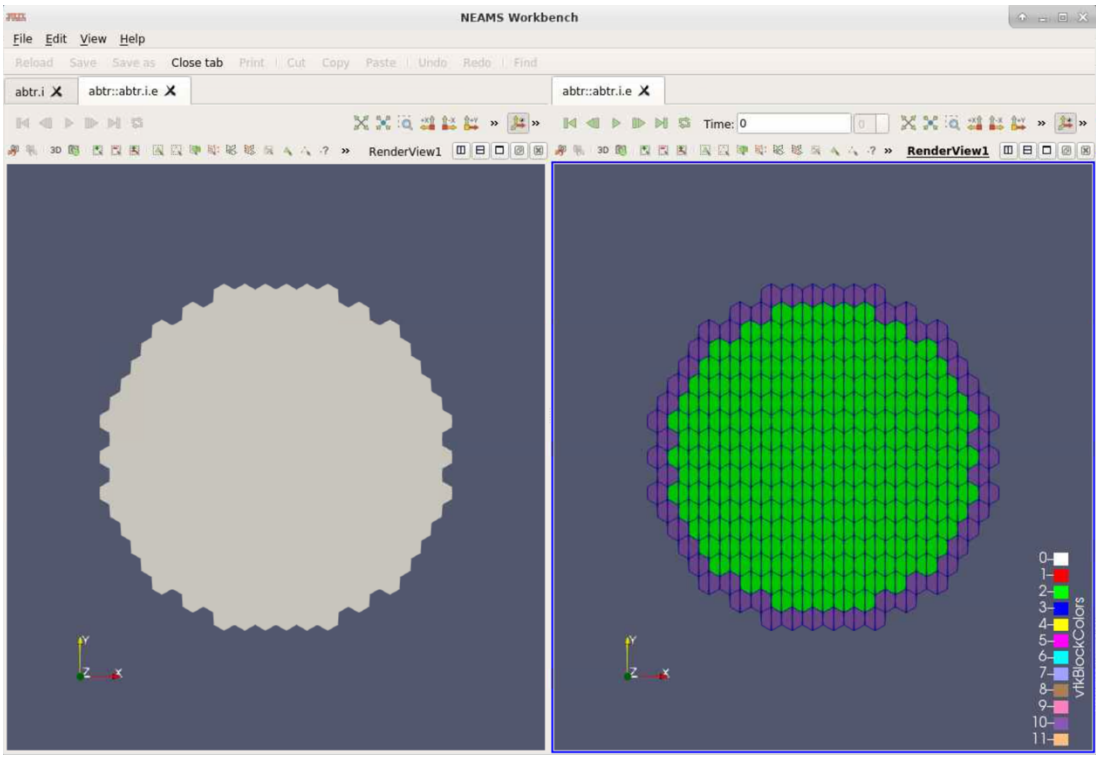

Figure 1. (Left) Prior default mesh RenderView and (right) the new default mesh RenderView.

3.3 MESH COMPONENT SHOW AND HIGHLIGHT

Workbench Text Editor allows a user to right-click a block, nodeset, or sideset ID and show or highlight that block in the active ParaView RenderView. Highlighting a component increases visibility by coloring the edges and vertices of the component a neon purple; however, doing so will not hide any other components that may obstruct the visibility of the component. Highlighted components will be only visible if a clipping or other view operation allows it to be visualized. Showing a component will hide all but that component, a feature previously available only via an *Inspect* widget. These operations assist the user in visual verification of mesh input as well as visual analysis of mesh results.

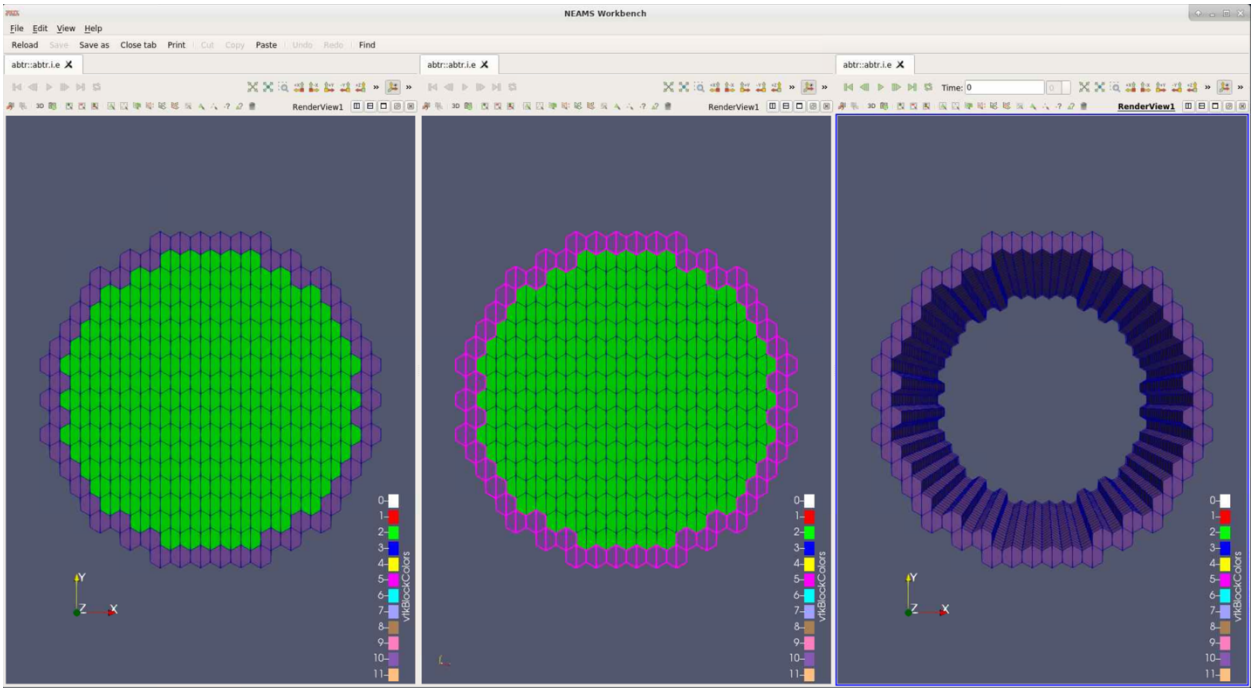

Figure 2. Workbench embedded ParaView RenderView illustration. (Left) default, (center) after block highlight, and (right) after block show.

3.4 APPLICATION CHOOSER POPUP

The Workbench supports editing multiple application inputs simultaneously. This capability supports complex workflows such as is the case with MOOSE MultiApp jobs. However, the applications input's extension is often ambiguous, which requires the user to select which application to use for interpreting the input and launching the file. Workbench now automatically displays a popup when an ambiguous input extension is detected, which allows the user to quickly choose the preferred application logic to apply for editing and launching the input file. Previously, the Workbench selected the first active application with the given extension and required the user to change the selection if it was not the desired application. The up, down, and enter keys can be used to quickly select the desired application interpreter when opening an input file.

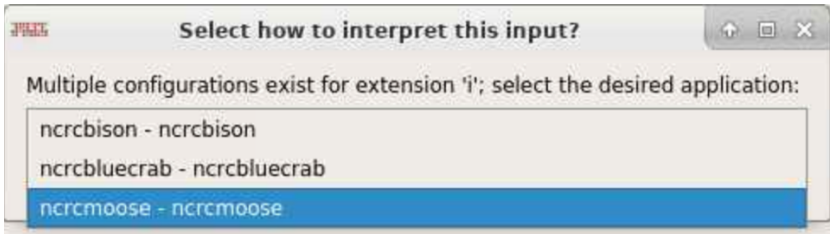

Figure 3. Application interpreter selection, as seen on NCRC with multiple MOOSE application configurations active.

3.5 SCHEDULER JOB SUBMISSION PREVIEW AND EDITOR

For scheduled jobs, Workbench now provides a scheduler submission preview upon clicking *Run*. The user can confirm or update the scheduler options (e.g., number of CPUs and amount of memory) prior to submitting the job for execution. Previously, updating this information required navigating to the Run Configuration options dialog and editing the *Remote Scheduler Header* field.

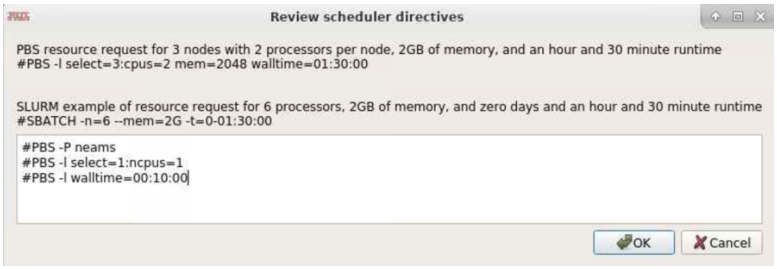

Figure 4. Job scheduler directive review and edit.

3.6 AUTOMATIC MESH GENERATION AND VISUALIZATION

The Workbench Text Editor's *View* button now displays the problem's input mesh in a new ParaView RenderView tab. If a mesh generator is used, then the MOOSE mesh-only option will be invoked, and the Messages panel will display the mesh generation log. Then, upon successful generation, a ParaView RenderView tab will display the generated mesh for further inspection. If a mesh file is referenced, then Workbench will open this file in a ParaView RenderView tab.

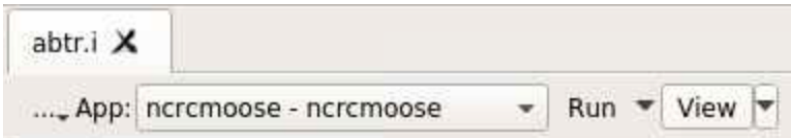

Figure 5. Workbench TextEditor View button automatically displays input mesh, generating the mesh if needed.

3.7 CONSOLIDATED EDITOR OPTIONS

The Workbench Text Editor's grammar, quick navigation, and processor widgets have been consolidated into an ellipses button (…) that opens a dropdown menu to reduce the busyness of the toolbar.

Figure 6. Prior unconsolidated TextEditor toolbar with Processor, Grammar, and document navigation widgets.

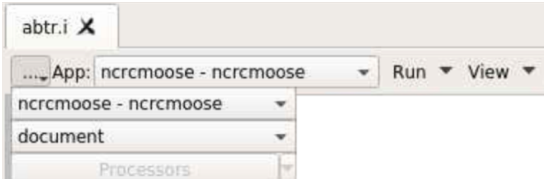

Figure 7. Updated TextEditor toolbar with Grammar, document navigation, and Processor accessible via ellipses button (shown expanded).

3.8 MOOSE CSV SCALAR RESULTS PLOTTING

The Workbench MOOSE CSV plotting has been updated with increased line weight and scatter size defaults that improve overall presentation. The MOOSE CSV processor was renamed from the generic 'MOOSE CSV Results' to a more specific 'MOOSE Timeseries' processor. Additionally, the intermediate scale plot format (SPF) file is no longer created. Lastly, the ability to move data series to new or alternate graphs enables data sets to be separated for analysis and presentation.

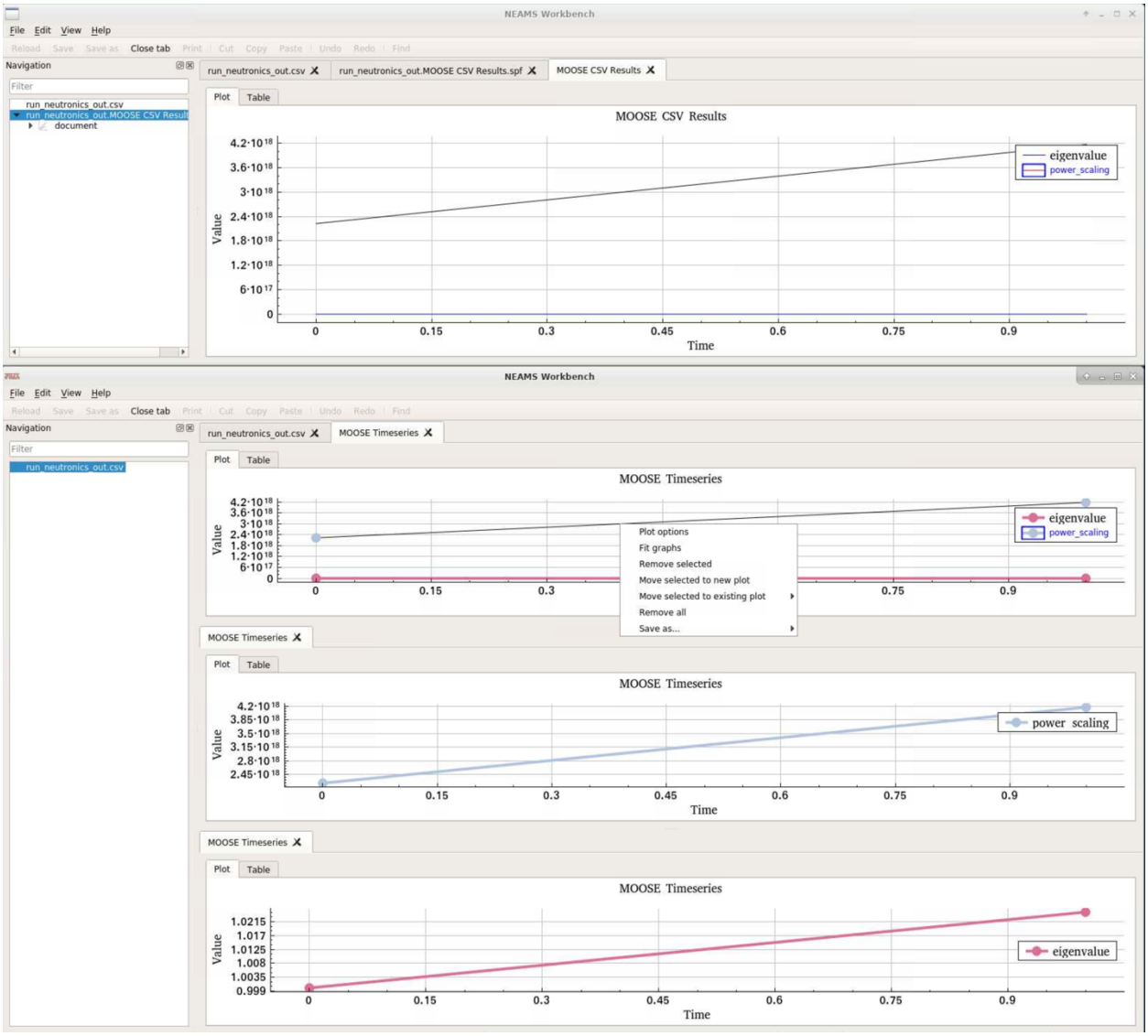

Figure 8. Example eigenvalue and power scaling CSV plot. (Top) prior default behavior. (Bottom) new behavior with no intermediate SPF, improved presentation, and the ability to move selected plot data.

3.9 SUPPORT ALTERNATE APPLICATION RUNTIME INVOCATION

The Workbench runtime layer now allows integrated applications to communicate alternate invocations of the application's choosing. This allows the Workbench's MOOSE runtime to provide the *Mesh-only* alternate execution. In addition to the default Run action, users now have the alternate *Run > mesh-only* option. This supplements the new Text Editor's *View* action, as communicated in Section 3.6. Previously,

the user was required to clone their MOOSE application's Run Configuration, toggle on the *--mesh-only* option, and provide a meaningful name (e.g., mesh-only). Then they could select the *MOOSE --mesh-only* run configuration and use the *Run* button to execute the MOOSE application with the additional *--mesh-only* command line option.

3.10 REUSE OF CURRENT INTERACTIVE SCHEDULER SESSION, REMOVING QUEUE WAIT TIMES FOR FAST INPUT ITERATIONS

The Workbench was updated to support the ability to submit jobs in the same interactive session as was provisioned to Workbench by the Scheduler. This avoids a time-consuming scheduler queue, greatly accelerating trial jobs as is needed for the MOOSE Reactor Module mesh generator. For example, the mesh generation job executes on the same node as Workbench without a new request for scheduler resources. Figure 9 illustrates the additional *Run* button dropdown entry of *Using current session*.

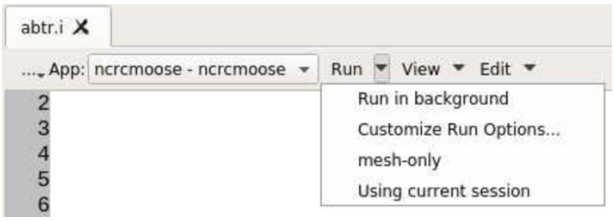

Figure 9. Run configuration dropdown illustrating mesh-only alternate invocation for ncrcmoose.

3.11 MULTIAPP INPUT QUICK NAVIGATION

The MOOSE MultiApp system requires inputs for multiple applications that are referenced by a parent or main application. The ability to quickly navigate to these referenced inputs has been requested, and Workbench now supports CTRL+Left-Click (CMD+Left-Click on MacOs) to open any referenced file in the Workbench Text Editor. The file path identified as able to be opened will be underlined, indicating that it can be clicked. Previously, a user was required to use a File browser and drag-and-drop the MultiApp input onto Workbench's *Navigation Panel* or use the *File > Open …* widget.

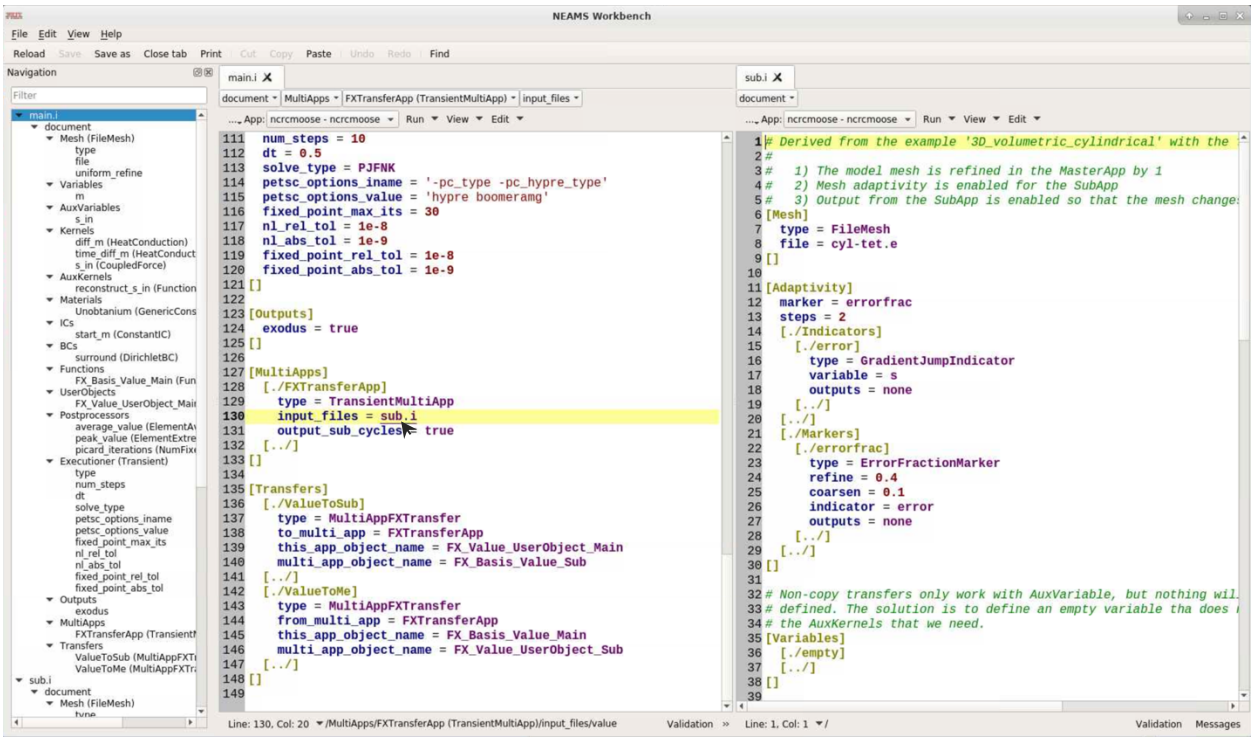

Figure 10. MultiApp example illustrating CTRL and mouse over underlining *sub.i* in *main.i* on line 130; left-clicking opens *sub.i*.

3.12 CONFIGURABLE APPLICATION-WIDE FONT SIZE

A feature requested by trainers was the ability to increase the Workbench application font size to improve visibility. Workbench has been updated to allow the user to change the application font via the *File* > *Settings* > *Environment* > *Application Font* option. Previously, users were required to toggle their screen resolution lower to increase font size. This new control does not change the Workbench Text Editor's font, which is controlled via the *File* > *Settings* > *Text Editor* > *Font* option.

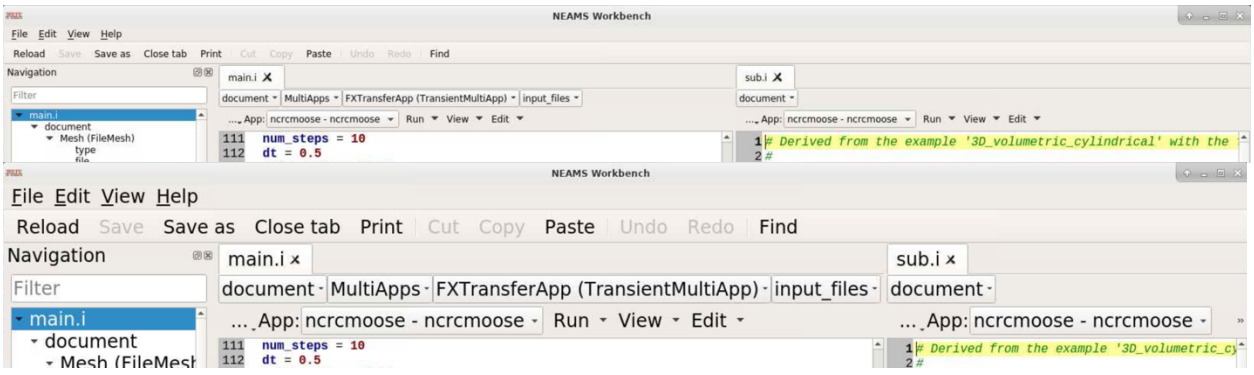

Figure 11. Illustration of default Workbench font size (top) and doubled font size (bottom)

3.13 EXPANDED NAVIGATION TREE

User feedback indicated the desire to keep an application input navigation tree open while the document is being edited. The Workbench now defaults the edited input's navigation tree as expanded. This

improves accessibility of quick navigation functionality. Previous performance concerns have not been observed.

3.14 PARAVIEW SCREENSHOT CAPABILITY

User feedback highlighted that the original ParaView integration into Workbench lacked the ParaView screen capture feature. This feature was requested to enable screen captures involving transparent backgrounds for use in reports and papers. Workbench now includes the entire ParaView File menu (*Visualization > File >...)*, which includes the screen capture and other functionalities.

4. APPLICATION INTEGRATION

As part of FY22 deliverables, the NEAMS Workbench was added to the HPC OnDemand website (NCRC) to be used as a GUI for some Moose apps to run on Sawtooth hosted at INL. This work was demonstrated with examples taken from the Virtual Test Bed reactor website and is documented in reports and tutorials [6]. In FY23, the Workbench team has begun integrating Cardinal into the NEAMS Workbench to run on Sawtooth as part of the same effort. During this preliminary work, Cardinal was compiled and tested on Sawtooth following the Cardinal documentation [7]. Then, a localhost Python script [6] was developed to allow users to submit jobs remotely on Sawtooth using the NEAMS Workbench GUI available through the HPC OnDemand website [6]. Testing was performed using various test cases taken from the regression test suite, the tutorials, and the virtual test bed reactor website. These preliminary steps allowed us to identify the following points of improvement for future work.

- Cardinal should be made available as a module to spare users the lengthy task of compiling it.
- Some Cardinal test cases require users to pass two input files to the executable, which is not currently available with the NEAMS Workbench. This will be made available soon.
- Cardinal relies on the third-party package NekRS, which requires additional pre- and post-processing steps to convert the mesh to a Nek-compatible format and to visualize the numerical solution with ParaView or VisIt. As part of an effort to integrate Nek5000 to the NEAMS Workbench [8,9], the same pre- and post-processing steps were supported. Further integrating Cardinal to the NEAMS Workbench will require leveraging this work to streamline the workflow.
- Cases using OpenMC have additional XML input files that are generated by a Python script. This step currently requires users to call the Python script at the command line.

Discussions will be held soon with the Cardinal lead and developers to see how some of the above points could be addressed.

5. USER SUPPORT AND FEEDBACK

The Workbench team provided user support and received feedback that indicated Workbench is successfully assisting users of the MOOSE framework and applications. Through MOOSE application and module trainings, feedback reinforced existing goals and contributed new, impactful improvements. The support of Workbench on the NCRC has significantly lowered the barrier for users to exercise Workbench and MOOSE. Documentation illustrating the Workbench and MOOSE application for the Virtual Test Bed model repository serves to help users get started [6]. A survey of attendees of the MOOSE Reactor Mesh module training yielded an average rating of 4/5 for Workbench user experience. Users exercised Workbench across all supported platforms (11% Windows, 27% MacOS, 29% Linux, and 33% NCRC), with ~60% of reported issues existing on Linux involving ParaView integration and

OpenGL incompatibility. This primary issue indicates a need to upgrade the version of ParaView integrated into Workbench in a future release. The user survey highlighted users' desire to receive specific training on Workbench features as well as enhancements to aid in their MOOSE workflows.

6. CONCLUSION

The NEAMS Workbench GUI received important updates for user efficiency, including more than 14 enhancements across eight software releases improving MOOSE integration. Additional changes and bug fixes improved user experience on Windows, MacOS, Linux, and INL's NCRC, receiving a 4/5 user experience rating. Integration of the LSP now provides native input diagnostics to the users interactively while editing inputs. Moreover, the implementation of LSP positions the MOOSE Framework integration in Workbench to achieve the desired objectives of deeper integration in support of MultiApp input validation and protocol extensions to effectively communicate results visualization metadata and input widget presentation.

7. REFERENCES

1. Alexander D. Lindsay, et al. "2.0 - MOOSE: Enabling massively parallel multiphysics simulation." *SoftwareX* 20, (2022) 101202, ISSN 2352-7110. https://doi.org/10.1016/j.softx.2022.101202.
2. Lefebvre, Robert Alexander, Langley, Brandon R., Miller, L Paul, Baird, Mark, and Collins, Benjamin S. NEAMS Workbench MOOSE Integration Update. United States: N. p. (2021). Web. doi:10.2172/1820792.
3. Lefebvre, Robert A., Langley, Brandon R., Lefebvre, Jordan P., and USDOE. Workbench Analysis Sequence Processor. Computer software (2017). https://www.osti.gov//servlets/purl/1419488. USDOE. Web. doi:10.11578/dc.20201125.2.
4. Emily Shemon, et al. Meshing, Language Server Protocol, and other User-Oriented MOOSE Framework Improvements to Enhance Reactor Analysis Capabilities and Workflows, ANL/NEAMS-23/3 (2023).
5. Lindsay A.D., Harbour L., Giudicelli G.L., Icenhour C., Kong F., Stogner R., et al. *User-oriented improvements in the MOOSE framework in support of multiphysics simulation*, Idaho National Laboratory. INL/RPT-22-67144 (2022).
6. Marc-Olivier G. Delchini, et al. "Enabling Remote Multiphysics Simulation of Virtual Test Bed Examples on Idaho National Laboratory High Performance Computing Resources through NEAMS Workbench." *Proceedings of ANS Winter* (2022).
7. Argonne National Laboratory, "Cardinal Documentation." Available online at: https://cardinal.cels.anl.gov/index.html (2020).
8. M.-O. G. Delchini, L. Swiler and R. Lefebvre, "Extension of the NEAMS workbench to parallel sensitivity and uncertainty analysis of thermal hydraulic parameters using Dakota and Nek5000," Nuclear Engineering and Technology, vol. 53, no. 10, pp. 3449–3459 (2021).
9. R. Lefebvre, D. Pointer, M.-O. Delchini and B. Rearden, "Integration of the Nek5000 Computational Fluid Dynamics Code with the NEAMS Workbench," in *American Nuclear Society* (2018).