ORNL/TM-2023/2988

VERA Integration in the NEAMS Workbench

Matthew E. Louis
Robert A. Lefebvre
Aaron M. Graham

Approved for public release.
Distribution is unlimited.

August 22, 2023

OAK RIDGE
National Laboratory

ORNL IS MANAGED BY UT–BATTELLE, LLC, FOR THE U.S. DEPARTMENT OF ENERGY

DOCUMENT AVAILABILITY

Reports produced after January 1, 1996, are generally available free via US Department of Energy (DOE) SciTech Connect.

Website osti.gov

Reports produced before January 1, 1996, may be purchased by members of the public from the following source:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone 703-605-6000 (1-800-553-6847)
TDD 703-487-4639
Fax 703-605-6900
E-mail info@ntis.gov
Website classic.ntis.gov

Reports are available to DOE employees, DOE contractors, Energy Technology Data Exchange representatives, and International Nuclear Information System representatives from the following source:

Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831
Telephone 865-576-8401
Fax 865-576-5728
E-mail reports@osti.gov
Website osti.gov

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-2023/2988

Nuclear Energy and Fuel Cycle Division

VERA INTEGRATION IN THE NEAMS WORKBENCH

Matthew E. Louis
Robert A. Lefebvre
Aaron M. Graham

Date Published: August 22, 2023

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-Battelle, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CIPS	crud-induced power shift
DNB	departure from nucleate boiling
DOE	US Department of Energy
HPC	high-performance computing
INL	Idaho National Laboratory
NCRC	Nuclear Computational Resource Center
NEAMS	Nuclear Energy Advanced Modeling and Simulation
NSUF	Nuclear Science User Facility
PCI	pellet–clad interaction
VERA	Virtual Environment for Reactor Applications

ABSTRACT

The Virtual Environment for Reactor Applications (VERA) is a cutting edge simulation code suite used to solve complex multiphysics problems for nuclear reactors. VERA makes use of high-performance computing (HPC) systems to solve large problems that were not historically tractable. Although extensive work has been done to enhance the usability of VERA, some work is still involved to build reactor models, to ensure their correctness, to properly execute the calculations, and to conduct the necessary analysis of the results. This effort is typical when using complex codes such as VERA. The NEAMS Workbench has been developed and maintained as part of the Nuclear Energy Advanced Modeling and Simulation (NEAMS) programto provide a single interface to execute a wide variety of physics codes, ranging from the initial building of input models through analysis of the results. This document serves as a manual, detailing the current integration of VERA with the NEAMS Workbench, and it provides a guide for configuring Workbench to allow remote execution of VERA.

1. INTRODUCTION

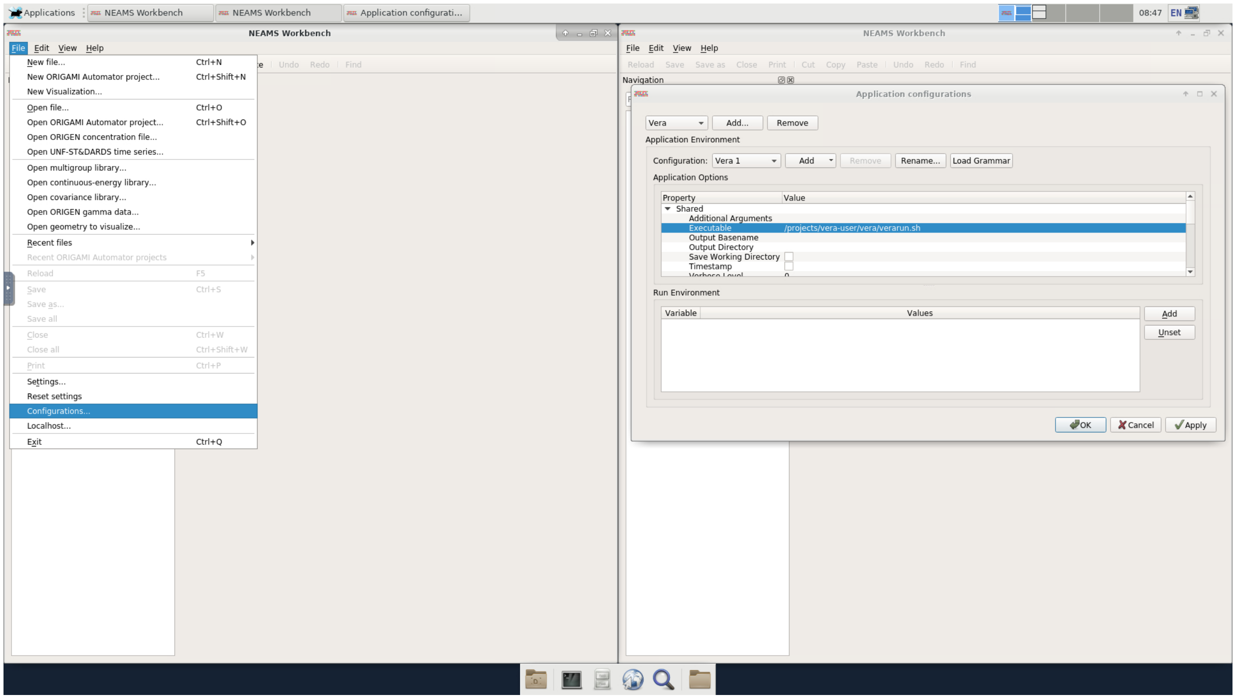
The Virtual Environment for Reactor Applications (VERA) [1] is a state-of-the-art fully coupled, pin-resolved reactor core simulator. In addition to high-fidelity core simulation for typical reactor depletion and transient calculations [2], VERA also enables more advanced analyses such as calculations of vessel fluence [3], departure from nucleate boiling (DNB) [4, 5], fuel performance [6], and various multiphysics phenomena such as pellet–clad Interaction (PCI) [7] and crud-induced power shift (CIPS) [8]. Although significant effort has been made to ensure the usability of VERA, constructing and validating reactor models, significant remaining work includes ensuring successful execution of the reactor models, as well as subsequent post-processing of results. Additionally, VERA's cutting edge resolution requires the use of high-performance computing (HPC) resources which can be prohibitive for some nuclear analysts.

The Nuclear Engineering Advanced Modeling and Simulation (NEAMS) Workbench [9] was developed to improve usability of nuclear codes by providing a unified framework for editing and validating input files, executing them, and visualizing the results. In addition, Workbench is readily available for use via the Idaho National Laboratory's (INL's) Nuclear Computational Resource Center (NCRC) [10], a part of a DOE Nuclear Science User Facility (NSUF) program providing HPC resources for nuclear modeling and simulation. Therefore, to improve the usability of VERA, preliminary work was performed to integrate it into the NEAMS Workbench and to allow remote execution of VERA jobs at the NCRC. At the time of writing, VERA input validation, job execution, and basic plotting and visualization are supported. This document serves as a guide for configuring Workbench to allow remote execution of VERA at the NCRC and demonstrates its input validation, job execution, and plotting features.

2. CONFIGURING VERA IN WORKBENCH

First, the user must have a valid NCRC account. Steps for requesting an NCRC account are given here: https://inl.gov/ncrc/how-ncrc-works/, and access to VERA, including instructions for obtaining individual/commercial licenses, are given here: https://vera.ornl.gov/vera-license/. Additionally, to use VERA on NCRC, the user must be added to the VERA users group, which can be requested by emailing vera-support@ornl.gov. If all of the above have been completed, VERA may be configured in Workbench through the NCRC by following the steps given below.

1. Initialize a Workbench session by following the instructions given in Step 1 here: Workbench Virtual Test Bed [11].
2. Add the path to the verarun executable in the VERA configuration. Click "File" in the top left of the Workbench app, and then click "Configurations" from the drop-down menu. In the top-left corner of the configurations window, select "Vera". Then, under "Application Options", double-click the box next to "Executable", and enter the path to the verarun executable. On the NCRC site, the path to the verarun executable is "/projects/vera-user/vera/verarun.sh".
3. Click "Apply" in the lower right corner to save changes, and then press "OK" to exit.



Configuration: Vera 1

Property	Value
Shared	
Additional Arguments	
Executable	/projects/vera-user/vera/verarun.sh
Output Basename	
Output Directory	
Save Working Directory	□
Timestamp	□
Verbose Level	0

Variable	Values

Figure 1. VERA configuration.

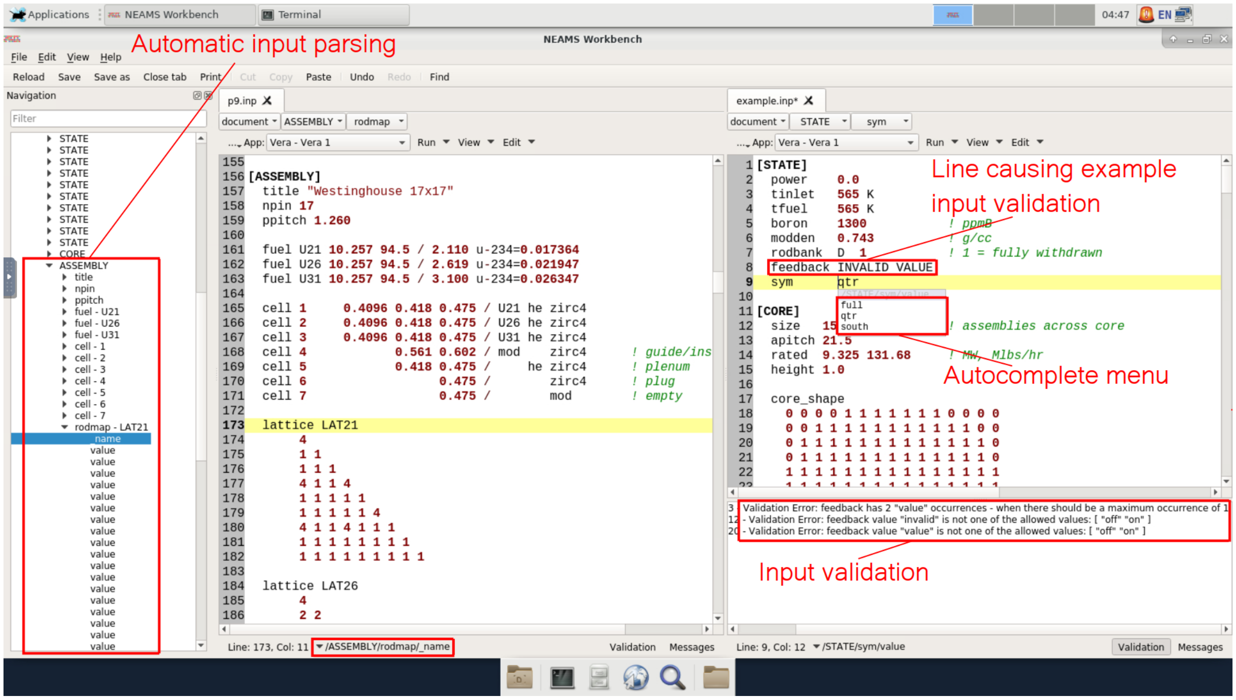
3. INPUT EDITING AND JOB EXECUTION

Now that VERA has been properly configured in Workbench, all of the plotting, execution, and input validation features are available.

3.1 INPUT EDITING

The input validation, syntax highlighting, and autocomplete features are readily seen by opening a VERAIn [12] input file. Examples of syntax highlighting, input validation, and auto complete are shown in Fig. 2. This input file and other VERA progression problem [13] input files are freely available here. The input block structure is shown in the navigation pane to the left and is produced by clicking on the "document" drop-down menu. Similarly, the Workbench-parsed label of any given input element can be seen in the bottom corner to the right of the navigation pane. Validation errors can be seen by clicking the "Validation" tab in the bottom right of the graphical user interface (GUI).

Additionally, autocomplete can be activated by pressing "CTRL + SPACE" anywhere in the input file. When autocomplete is activated in a given input block, all available input cards for that block will be shown in a list, which is narrowed down as the user begins typing; selecting an input card from that list will autocomplete it, along with a default value. Values of input cards can also be autocompleted by placing the cursor in front of the existing/default value and enabling autocomplete with "CTRL + SPACE" to display a list of supported values for the given input card, as shown in the right side of Fig. 2. (Note that autocomplete is not supported for all input cards). More information about autocomplete and input validation features can be found in the Workbench help documentation which can be accessed in Workbench by clicking "Help" in the top left of the Workbench app, and "Help Documentation" from the drop-down menu).



```
[ASSEMBLY]
  title "Westinghouse 17x17"
  npin 17
  ppitch 1.260

  fuel U21 10.257 94.5 / 2.110 u-234=0.017364
  fuel U26 10.257 94.5 / 2.619 u-234=0.021947
  fuel U31 10.257 94.5 / 3.100 u-234=0.026347

  cell 1     0.4096 0.418 0.475 / U21 he zirc4
  cell 2     0.4096 0.418 0.475 / U26 he zirc4
  cell 3     0.4096 0.418 0.475 / U31 he zirc4
  cell 4            0.561 0.602 / mod    zirc4      ! guide/ins
  cell 5            0.418 0.475 /     he zirc4      ! plenum
  cell 6                  0.475 /        zirc4      ! plug
  cell 7                  0.475 /        mod        ! empty

  lattice LAT21
       4
       1 1
       1 1 1
       4 1 1 4
       1 1 1 1 1
       1 1 1 1 1 4
       4 1 1 4 1 1 1
       1 1 1 1 1 1 1 1
       1 1 1 1 1 1 1 1 1

  lattice LAT26
       4
       2 2
```

```
[STATE]
  power    0.0
  tinlet   565 K
  tfuel    565 K
  boron    1300              ! ppmB
  modden   0.743             ! g/cc
  rodbank  D  1              ! 1 = fully withdrawn
  feedback INVALID VALUE
  sym      qtr

[CORE]
  size   15                  ! assemblies across core
  apitch 21.5
  rated  9.325 131.68        ! MW, Mlbs/hr
  height 1.0

  core_shape
    0 0 0 0 1 1 1 1 1 1 1 0 0 0 0
    0 0 1 1 1 1 1 1 1 1 1 1 1 0 0
    0 1 1 1 1 1 1 1 1 1 1 1 1 1 0
    0 1 1 1 1 1 1 1 1 1 1 1 1 1 0
    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
3 Validation Error: feedback has 2 "value" occurrences - when there should be a maximum occurrence of 1
12 - Validation Error: feedback value "invalid" is not one of the allowed values: [ "off" "on" ]
20 - Validation Error: feedback value "value" is not one of the allowed values: [ "off" "on" ]
```

Figure 2. An example of input validation, syntax highlighting, and autocomplete features for VERA input files in Workbench.

3.1.1 KNOWN ISSUES

Some known issues with the current implementation of the VERA input validation and input parsing incorrectly produce validation errors and result in incorrect input parsing in certain cases. These issues will be resolved in future work, but they can be safely disregarded in the meantime.

1. **Validation Error:** "could not find <include-file>"
   Description: When including one of the built-in include files from the VERA installs directory in the VERA project folder (e.g. "include w4loop_8r_qtr.txt", a common include in the VERA progression problem input files [13]), this validation error will appear. By default, the Workbench validation engine checks for include files adjacent to the input file and thus overlooks valid include files that are checked by verarun (the VERA runtime script) when processing the input file.
   Action: If the include file is one of the valid input files in the verarun installs directory, then the include will be successful, and the validation error should be ignored. If the include file can not be found in the verarun installs directory, then an error message will be produced when running the input file. Note that this *only* applies when including built-in include files, and validation errors produced when including user-generated include files should *not* be ignored.

2. **Validation Error:** "<input card> value <start..stopxstep> is not of type <input type>"
   Description: Workbench does not support validation of inputs using automatic list generation. For example, the depletion card: deplete EFPD 0 10 20 30 40 could be shortened to: deplete EFPD <0..40x10>, and would result in the validation error: "deplete value <0..40x10> is not of type Real". Inputs created using the automatic list generation feature are still valid, but are not correctly validated by Workbench.
   Action: Ignore this validation error when using automatic-list generation.

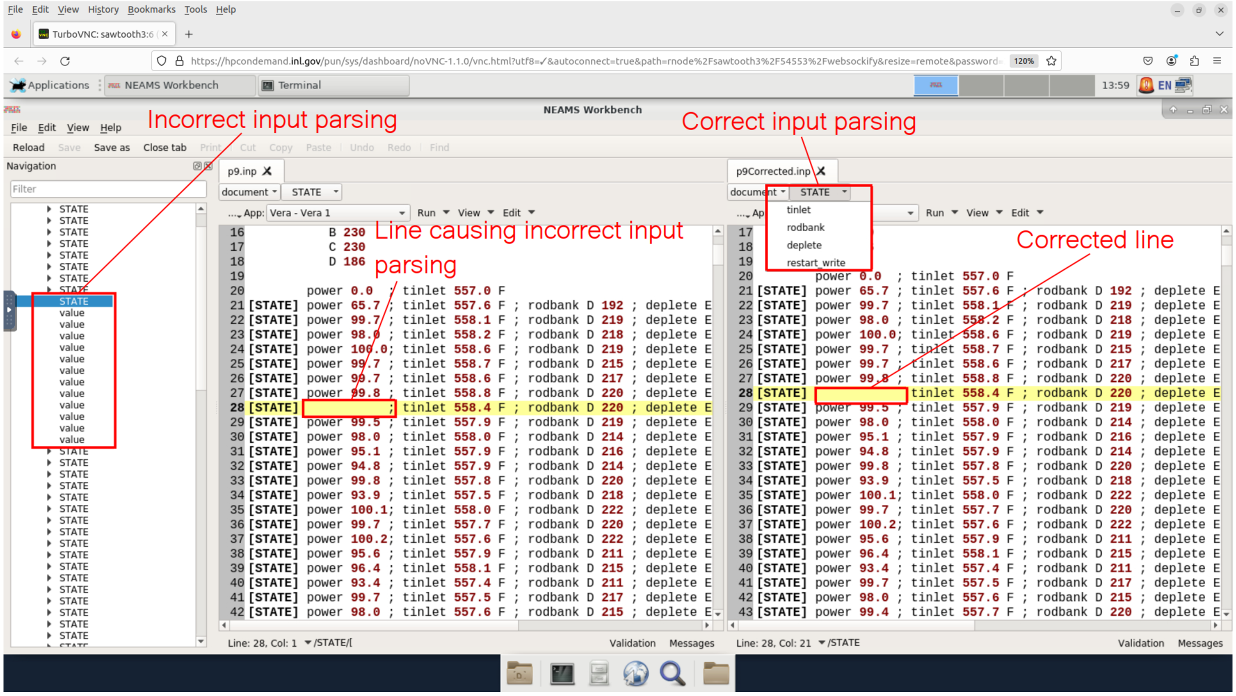
3. **Validation Error:** "/ASSEMBLY/fuel/<dopant> is not a valid piece of input"
   Description: This error is produced when specifying a dopant in a fuel card, for example: "fuel U400Gd 10.412 94.5 / 4.0 / gad=1.25" specifies $4^{w}\%$ enriched UO_2 fuel interspersed with $1.25^{w}\%$ gadolinia, where the dopant is "gad". The Workbench parser does not support validation of "dopant" sections of fuel cards.
   Action: Ignore this validation error.

4. **Validation Error:** "<input card> value <value> does not exist in set: <input set>"
   Description: Produced when specifying input values using the asterisk notation to repeat values (e.g. "det_map 192*2".) Used to add detector 2 to all 192 assemblies in the detector map
   Action: Avoid using asterisk notation, or disregard validation errors when using asterisk notation.

5. **Incorrect Validation:** Input cards are incorrectly parsed when a blank line is specified with a semicolon. See Fig. 3.
   Description: The Workbench parser does not support termination of an empty record
   Action: Remove the empty input record by removing the empty input record's terminating semicolon.

3.2 JOB EXECUTION

Once an input file has been written and validated, it can be run by clicking the "Run" button in the top right corner of the GUI. This will create a new tab in the "Messages" pane with the date and time of execution as the label, and subsequent runs will appear as additional tabs; this is shown in Fig. 4a. If the job is successfully submitted to the scheduler, then "[verarun] submitted: <job-id>" will be printed to the messages pane. The job will sit in the queue for a variable time, depending on the computational resources requested by the job (e.g., CPU cores and memory), after which it will be executed, and status messages, which are also contained in the .log file produced upon job completion, with time stamps will be printed to the Messages pane. Example status messages from a multi-state calculation are shown in Fig. 4b.

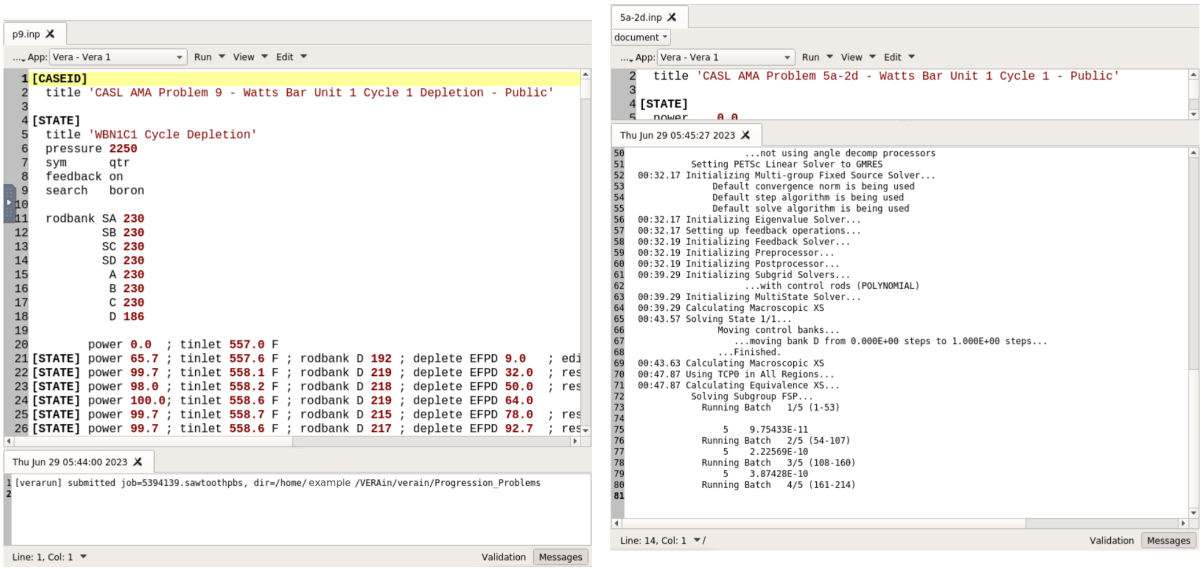

File Edit View History Bookmarks Tools Help
TurboVNC: sawtooth3:6
https://hpcondemand.inl.gov/pun/sys/dashboard/noVNC-1.1.0/vnc.html?utf8=✓&autoconnect=true&path=rnode%2Fsawtooth3%2F54553%2Fwebsockify&resize=remote&password=
120%
Applications
NEAMS Workbench
Terminal
13:59
EN
NEAMS Workbench
File Edit View Help
Reload Save Save as Close tab Print Cut Copy Paste Undo Redo Find
Navigation
Filter
STATE
value
Incorrect input parsing
Correct input parsing
Line causing incorrect input parsing
Corrected line
p9.inp
p9Corrected.inp
document
STATE
tinlet
rodbank
deplete
restart_write
App: Vera - Vera 1
Run
View
Edit
Line: 28, Col: 1 /STATE/[
Line: 28, Col: 21 /STATE
Validation
Messages

```
                B 230
                C 230
                D 186

           power 0.0  ; tinlet 557.0 F
[STATE] power 65.7 ; tinlet 557.6 F ; rodbank D 192 ; deplete E
[STATE] power 99.7 ; tinlet 558.1 F ; rodbank D 219 ; deplete E
[STATE] power 98.0 ; tinlet 558.2 F ; rodbank D 218 ; deplete E
[STATE] power 100.0; tinlet 558.6 F ; rodbank D 219 ; deplete E
[STATE] power 99.7 ; tinlet 558.7 F ; rodbank D 215 ; deplete E
[STATE] power 99.7 ; tinlet 558.6 F ; rodbank D 217 ; deplete E
[STATE] power 99.8 ; tinlet 558.8 F ; rodbank D 220 ; deplete E
[STATE]            ; tinlet 558.4 F ; rodbank D 220 ; deplete E
[STATE] power 99.5 ; tinlet 557.9 F ; rodbank D 219 ; deplete E
[STATE] power 98.0 ; tinlet 558.0 F ; rodbank D 214 ; deplete E
[STATE] power 95.1 ; tinlet 557.9 F ; rodbank D 216 ; deplete E
[STATE] power 94.8 ; tinlet 557.9 F ; rodbank D 214 ; deplete E
[STATE] power 99.8 ; tinlet 557.8 F ; rodbank D 220 ; deplete E
[STATE] power 93.9 ; tinlet 557.5 F ; rodbank D 218 ; deplete E
[STATE] power 100.1; tinlet 558.0 F ; rodbank D 222 ; deplete E
[STATE] power 99.7 ; tinlet 557.7 F ; rodbank D 220 ; deplete E
[STATE] power 100.2; tinlet 557.6 F ; rodbank D 222 ; deplete E
[STATE] power 95.6 ; tinlet 557.9 F ; rodbank D 211 ; deplete E
[STATE] power 96.4 ; tinlet 558.1 F ; rodbank D 215 ; deplete E
[STATE] power 93.4 ; tinlet 557.4 F ; rodbank D 211 ; deplete E
[STATE] power 99.7 ; tinlet 557.5 F ; rodbank D 217 ; deplete E
[STATE] power 98.0 ; tinlet 557.6 F ; rodbank D 215 ; deplete E
```

```
           power 0.0  ; tinlet 557.0 F
[STATE] power 65.7 ; tinlet 557.6 F ; rodbank D 192 ; deplete E
[STATE] power 99.7 ; tinlet 558.1 F ; rodbank D 219 ; deplete E
[STATE] power 98.0 ; tinlet 558.2 F ; rodbank D 218 ; deplete E
[STATE] power 100.0; tinlet 558.6 F ; rodbank D 219 ; deplete E
[STATE] power 99.7 ; tinlet 558.7 F ; rodbank D 215 ; deplete E
[STATE] power 99.7 ; tinlet 558.6 F ; rodbank D 217 ; deplete E
[STATE] power 99.8 ; tinlet 558.8 F ; rodbank D 220 ; deplete E
[STATE]              tinlet 558.4 F ; rodbank D 220 ; deplete E
[STATE] power 99.5 ; tinlet 557.9 F ; rodbank D 219 ; deplete E
[STATE] power 98.0 ; tinlet 558.0 F ; rodbank D 214 ; deplete E
[STATE] power 95.1 ; tinlet 557.9 F ; rodbank D 216 ; deplete E
[STATE] power 94.8 ; tinlet 557.9 F ; rodbank D 214 ; deplete E
[STATE] power 99.8 ; tinlet 557.8 F ; rodbank D 220 ; deplete E
[STATE] power 93.9 ; tinlet 557.5 F ; rodbank D 218 ; deplete E
[STATE] power 100.1; tinlet 558.0 F ; rodbank D 222 ; deplete E
[STATE] power 99.7 ; tinlet 557.7 F ; rodbank D 220 ; deplete E
[STATE] power 100.2; tinlet 557.6 F ; rodbank D 222 ; deplete E
[STATE] power 95.6 ; tinlet 557.9 F ; rodbank D 211 ; deplete E
[STATE] power 96.4 ; tinlet 558.1 F ; rodbank D 215 ; deplete E
[STATE] power 93.4 ; tinlet 557.4 F ; rodbank D 211 ; deplete E
[STATE] power 99.7 ; tinlet 557.5 F ; rodbank D 217 ; deplete E
[STATE] power 98.0 ; tinlet 557.6 F ; rodbank D 215 ; deplete E
[STATE] power 99.4 ; tinlet 557.7 F ; rodbank D 220 ; deplete E
```

Figure 3. Incorrect identification of input cards in the "[STATE]" block of the p9 progression problem input.



```
[CASEID]
  title 'CASL AMA Problem 9 - Watts Bar Unit 1 Cycle 1 Depletion - Public'

[STATE]
  title 'WBN1C1 Cycle Depletion'
  pressure 2250
  sym      qtr
  feedback on
  search   boron

  rodbank SA 230
          SB 230
          SC 230
          SD 230
           A 230
           B 230
           C 230
           D 186

        power 0.0  ; tinlet 557.0 F
[STATE] power 65.7 ; tinlet 557.6 F ; rodbank D 192 ; deplete EFPD 9.0   ; edi
[STATE] power 99.7 ; tinlet 558.1 F ; rodbank D 219 ; deplete EFPD 32.0  ; res
[STATE] power 98.0 ; tinlet 558.2 F ; rodbank D 218 ; deplete EFPD 50.0  ; res
[STATE] power 100.0; tinlet 558.6 F ; rodbank D 219 ; deplete EFPD 64.0
[STATE] power 99.7 ; tinlet 558.7 F ; rodbank D 215 ; deplete EFPD 78.0  ; res
[STATE] power 99.7 ; tinlet 558.6 F ; rodbank D 217 ; deplete EFPD 92.7  ; res
```

```
[verarun] submitted job=5394139.sawtoothpbs, dir=/home/ example /VERAin/verain/Progression_Problems
```

```
  title 'CASL AMA Problem 5a-2d - Watts Bar Unit 1 Cycle 1 - Public'

[STATE]
  power     0.0
```

```
                      ...not using angle decomp processors
             Setting PETSc Linear Solver to GMRES
  00:32.17 Initializing Multi-group Fixed Source Solver...
                 Default convergence norm is being used
                 Default step algorithm is being used
                 Default solve algorithm is being used
  00:32.17 Initializing Eigenvalue Solver...
  00:32.17 Setting up feedback operations...
  00:32.19 Initializing Feedback Solver...
  00:32.19 Initializing Preprocessor...
  00:32.19 Initializing Postprocessor...
  00:39.29 Initializing Subgrid Solvers...
                      ...with control rods (POLYNOMIAL)
  00:39.29 Initializing MultiState Solver...
  00:39.29 Calculating Macroscopic XS
  00:43.57 Solving State 1/1...
                  Moving control banks...
                     ...moving bank D from 0.000E+00 steps to 1.000E+00 steps...
                  ...Finished.
  00:43.63 Calculating Macroscopic XS
  00:47.87 Using TCP0 in All Regions...
  00:47.87 Calculating Equivalence XS...
             Solving Subgroup FSP...
               Running Batch   1/5 (1-53)

                   5    9.75433E-11
               Running Batch   2/5 (54-107)
                   5    2.22569E-10
               Running Batch   3/5 (108-160)
                   5    3.87428E-10
               Running Batch   4/5 (161-214)
```

(a) Job submission message

(b) Job execution messages (from the 5a-2d progression problem)

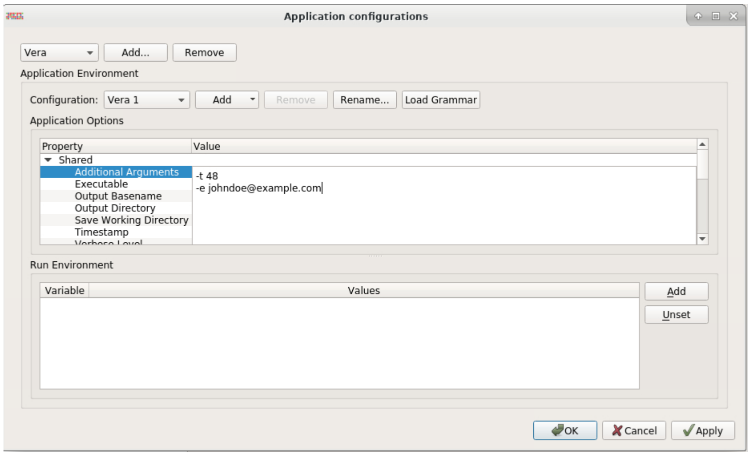
Figure 4. Example message pane from executing an input file.

3.3 SETTING RUNTIME OPTIONS

The VERA runtime script (verarun) generates the .PBS script using data specified in the input file (e.g., the number of cores), as well as a few defaults. For example, the wall-time defaults to 24 hours, which is a reasonable default for a cycle depletion, but it could easily be far too long for simpler calculations or far too short for highly complex calculations. To update runtime options such as wall-time, open the VERA configuration menu as described in step 3 of Section 2, and double-click in the box next to “Additional Argument”. supply additional arguments that will be applied to verarun at runtime, and click “Apply” and “OK” to save the configuration. For example, to set the walltime to 48 hours and set the email address to which the job completion email will be sent to “johndoe@example.com”, the user would supply the additional argument:

“-t 48
-e johndoe@example.com”

as in Fig. 5. Note that argument-value pairs *must* be listed on different lines in the “Additional Arguments” box in order for them to be parsed correctly by Workbench. Note also that when specifying processors per node, and memory per process, the user must set the “cpu_per_node” and “memory_per_process” properties in the configuration menu rather than supplying the verarun additional arguments “--ppn” and “--pmem”, as Workbench will not properly process these verarun arguments. For a list of verarun arguments and descriptions, see the current version of the VERAIn User’s Manual [14].



Configuration: Vera 1

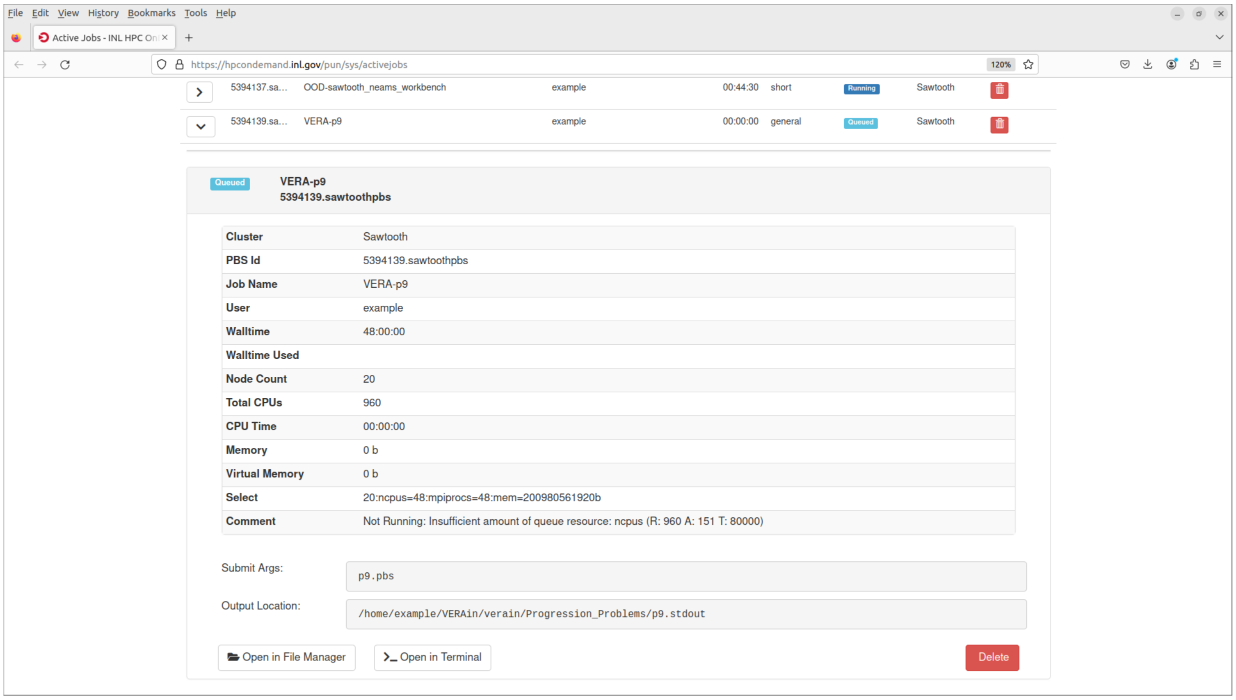
Property	Value
Shared	
Additional Arguments	-t 48 -e johndoe@example.com
Executable	
Output Basename	
Output Directory	
Save Working Directory	
Timestamp	
Verbose Level	

Figure 5. Example of modifying runtime options.

3.3.1 JOB TERMINATION AND RUNTIME ERRORS

A submitted job can be terminated by simply right clicking the Messages pane and clicking “Terminate Process” from the drop-down menu. Upon successful job deletion, “[verarun] Job aborted: Deleting job” will be printed to the Messages pane. Note that if “Kill Process” is selected from the drop-down instead of “Terminate Process”, the VERA runtime script (verarun) will not be able to delete the job before it is killed, so the job must manually be deleted from the HPC OnDemand job manager, the “Active Jobs” page, or qsub/scancel must be used from the terminal. Likewise, a job will continue running if Workbench is closed, which can be desirable for longer running jobs, but if not, it can manually be deleted as mentioned above.

Once a job has been submitted to the scheduler, it can be viewed in the HPC OnDemand job manager, from which the job status, runtime, cluster, and other items, can be viewed. Additionally, a detailed description of job attributes and computational resources can be seen by clicking on the drop-down menu to the left (shown in Fig. 6). If the job is terminated by the scheduler because the job reaches wall-time or too many computational resources are requested, for example, or if the execution of VERA fails (exits with a nonzero return code) then an error message will be printed to the Messages pane. Details on error messages and their potential causes are given in Table 1. If the execution of VERA fails, then additional error information can be found in the .log and/or the .stdout file that is created in the working directory adjacent to the input file, an example of which is shown in Fig. 7. If there is any unexpected behavior, the user should contact vera-support@ornl.gov.


File Edit View History Bookmarks Tools Help
Active Jobs - INL HPC On
https://hpcondemand.inl.gov/pun/sys/activejobs
120%

5394137.sa...	OOD-sawtooth_neams_workbench	example	00:44:30	short	Running	Sawtooth
5394139.sa...	VERA-p9	example	00:00:00	general	Queued	Sawtooth

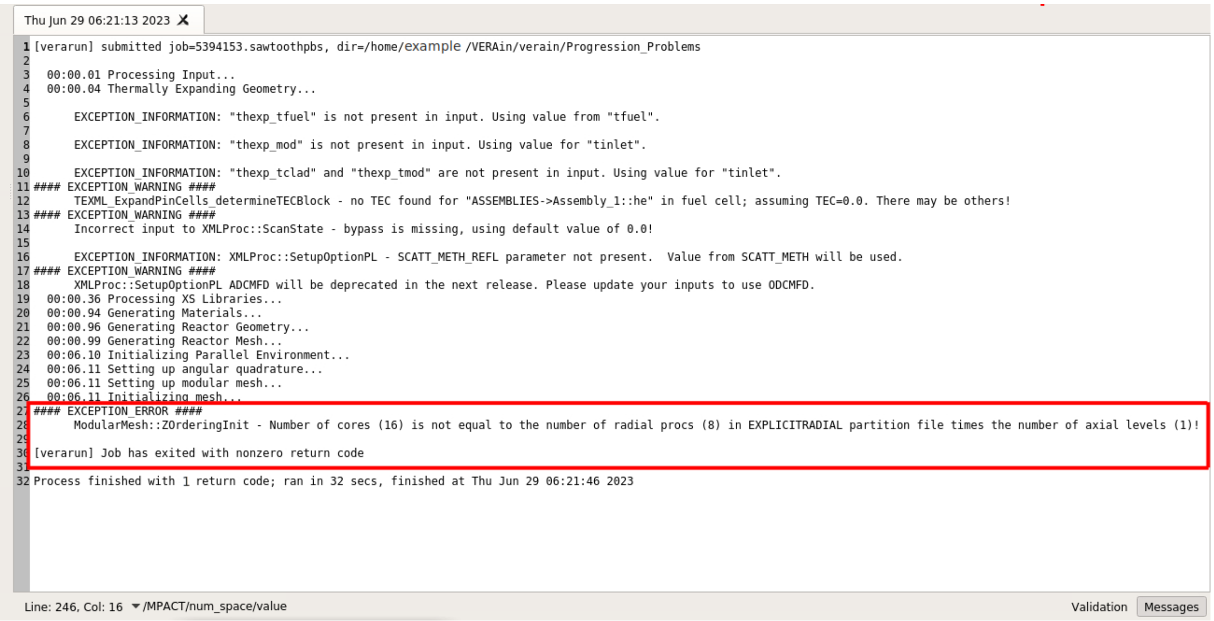
Queued **VERA-p9**
5394139.sawtoothpbs

Cluster	Sawtooth
PBS Id	5394139.sawtoothpbs
Job Name	VERA-p9
User	example
Walltime	48:00:00
Walltime Used	
Node Count	20
Total CPUs	960
CPU Time	00:00:00
Memory	0 b
Virtual Memory	0 b
Select	20:ncpus=48:mpiprocs=48:mem=200980561920b
Comment	Not Running: Insufficient amount of queue resource: ncpus (R: 960 A: 151 T: 80000)

Submit Args: `p9.pbs`

Output Location: `/home/example/VERAin/verain/Progression_Problems/p9.stdout`

Figure 6. A queued VERA job on the "Active Jobs" page of HPC OnDemand.



```
[verarun] submitted job=5394153.sawtoothpbs, dir=/home/example /VERAin/verain/Progression_Problems

  00:00.01 Processing Input...
  00:00.04 Thermally Expanding Geometry...

      EXCEPTION_INFORMATION: "thexp_tfuel" is not present in input. Using value from "tfuel".

      EXCEPTION_INFORMATION: "thexp_mod" is not present in input. Using value for "tinlet".

      EXCEPTION_INFORMATION: "thexp_tclad" and "thexp_tmod" are not present in input. Using value for "tinlet".
#### EXCEPTION_WARNING ####
      TEXML_ExpandPinCells_determineTECBlock - no TEC found for "ASSEMBLIES->Assembly_1::he" in fuel cell; assuming TEC=0.0. There may be others!
#### EXCEPTION_WARNING ####
      Incorrect input to XMLProc::ScanState - bypass is missing, using default value of 0.0!

      EXCEPTION_INFORMATION: XMLProc::SetupOptionPL - SCATT_METH_REFL parameter not present.  Value from SCATT_METH will be used.
#### EXCEPTION_WARNING ####
      XMLProc::SetupOptionPL ADCMFD will be deprecated in the next release. Please update your inputs to use ODCMFD.
  00:00.36 Processing XS Libraries...
  00:00.94 Generating Materials...
  00:00.96 Generating Reactor Geometry...
  00:00.99 Generating Reactor Mesh...
  00:06.10 Initializing Parallel Environment...
  00:06.11 Setting up angular quadrature...
  00:06.11 Setting up modular mesh...
  00:06.11 Initializing mesh...
#### EXCEPTION_ERROR ####
      ModularMesh::ZOrderingInit - Number of cores (16) is not equal to the number of radial procs (8) in EXPLICITRADIAL partition file times the number of axial levels (1)!

[verarun] Job has exited with nonzero return code

Process finished with 1 return code; ran in 32 secs, finished at Thu Jun 29 06:21:46 2023
```

Figure 7. Example of a VERA execution error caused by specifying an incorrect number of processors for an explicit radial partition.

Table 1. Runtime error messages and their potential causes

Error message	Cause
[verarun] Job has been aborted by scheduler	Scheduler sends a SIGINT, SIGABRT, SIGQUIT, SIGTERM signal. This can result from the user cancelling a job, insufficient computational resources, or a job reaching walltime or running out of memory.[1]
[verarun] Job has exited with nonzero return code	The execution of VERA has exited with a nonzero return code. This could be caused by an input error, or a bug in VERA. When this message is received, more information can be found in the .log file.
[verarun] Job aborted: Deleting job	verarun receives a SIGTERM signal from its parent process, Workbench. This is caused by right clicking the Messages pane and clicking "Terminate Process".

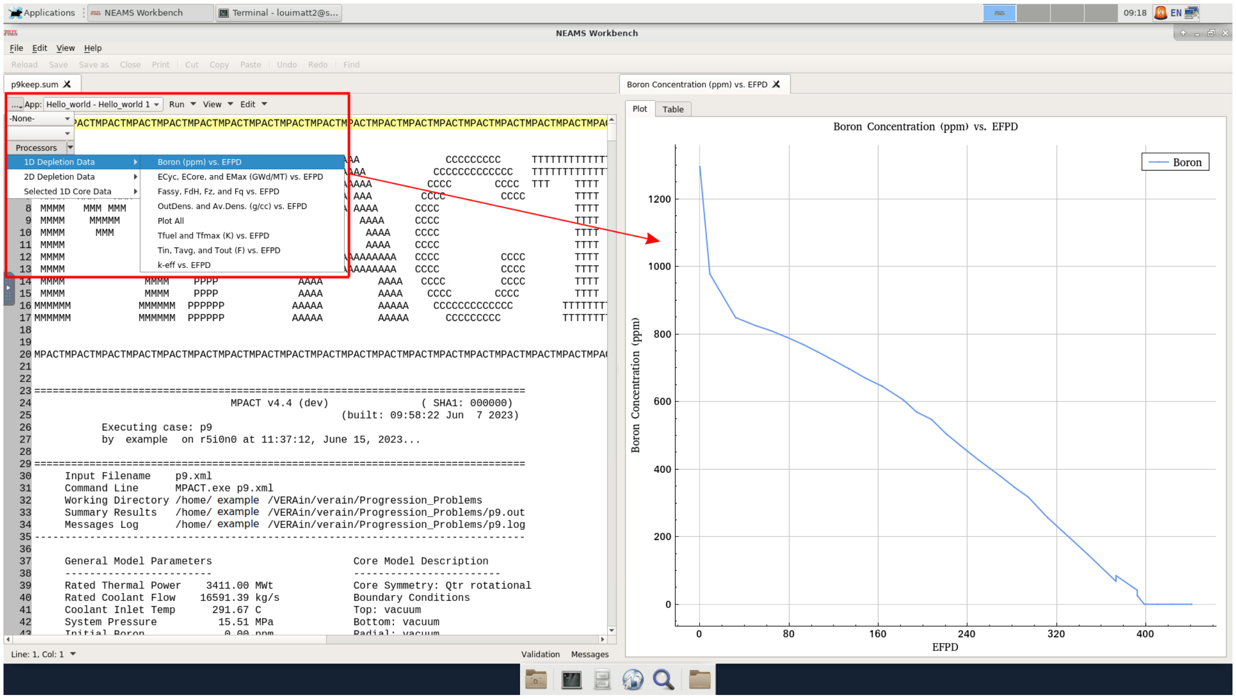
[1] By default, the scheduler sends SIGTERM and waits a specified time before sending SIGKILL. Note that the delay between the SIGTERM and SIGKILL signals is zero by default when a user manually executes qdel <Job ID> or deletes the job via the HPC OnDemand job manager, and the PBS script is terminated before signaling to verarun to print the "Job aborted" message and terminate; in this case, the Workbench process must be terminated manually.

4. OUTPUT VISUALIZATION

Nominally, upon job completion, .sum and .h5 files are produced adjacent to the input file, both of which can be used to visualize the results of the calculation. Other output data can also be produced by supplying the edits card in the input file as described in Section 4.3.

4.1 WORKBENCH SUM-FILE PLOTTING

To visualize the results in the .sum file, open the .sum file in Workbench and click “File”, “Open file” (or CTRL+O), select “All files (*)” as the file type, and click on the desired .sum file. Once the file is open, click the “...” drop-down menu in the top left of the file pane (shown in Fig. 8), and then click on the “Processors” drop-down menu to see the available processors for the given file. This drop-down menu will be grayed out if the given file contains no plottable data. Boron concentration vs. effective full power day (EFPD) as in the example in Fig. 8) and click the file to create a plot of the data. A list of supported output processors is given in Table 2. Additionally, data can be viewed in tabular format and easily exported to Excel or other applications by clicking on the “Table” tab.



```
=============================================================================
                              MPACT v4.4 (dev)              ( SHA1: 000000)
                                                (built: 09:58:22 Jun  7 2023)
          Executing case: p9
          by  example   on r5i0n0 at 11:37:12, June 15, 2023...

=============================================================================
     Input Filename    p9.xml
     Command Line      MPACT.exe p9.xml
     Working Directory /home/ example /VERAin/verain/Progression_Problems
     Summary Results   /home/ example /VERAin/verain/Progression_Problems/p9.out
     Messages Log      /home/ example /VERAin/verain/Progression_Problems/p9.log
-----------------------------------------------------------------------------

     General Model Parameters                      Core Model Description
     ------------------------                      ------------------------
     Rated Thermal Power    3411.00 MWt            Core Symmetry: Qtr rotational
     Rated Coolant Flow    16591.39 kg/s           Boundary Conditions
     Coolant Inlet Temp      291.67 C              Top: vacuum
     System Pressure          15.51 MPa            Bottom: vacuum
```

Figure 8. An example .sum file (p9.sum) and the available Workbench processors.

4.1.1 MULTI-STATE DATA

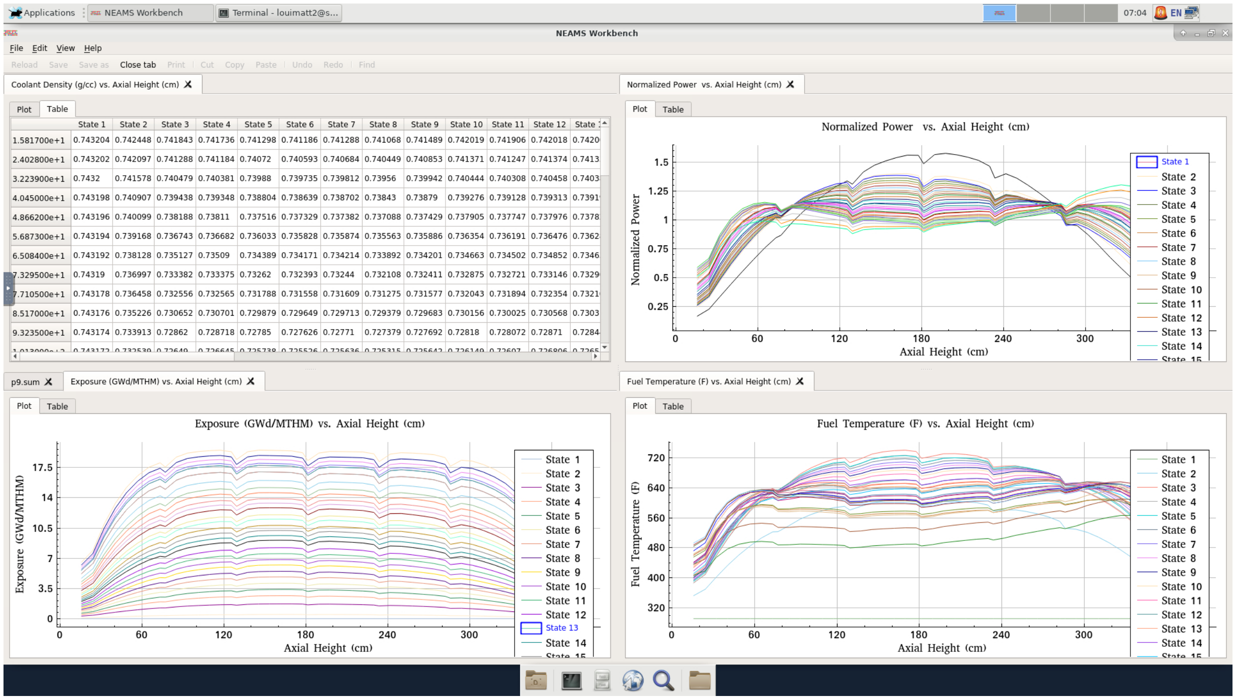
When the user is viewing multistate data, processors automatically plot the given data series over all state points (labelled as in the “input file”). When lists of state points become large, then the plot legend containing the state series can be navigated by dragging it up to scroll down or dragging it down to scroll up.

Development of the plotting function for 2D multi-state data is a work in progress, but it is fully functional with some workarounds. When plotting multi-state 2D data, only the last state will be shown in the plot window, but other states can be viewed by selecting the state from the legend, right-clicking the plot window,

Table 2. List of sumfile processors

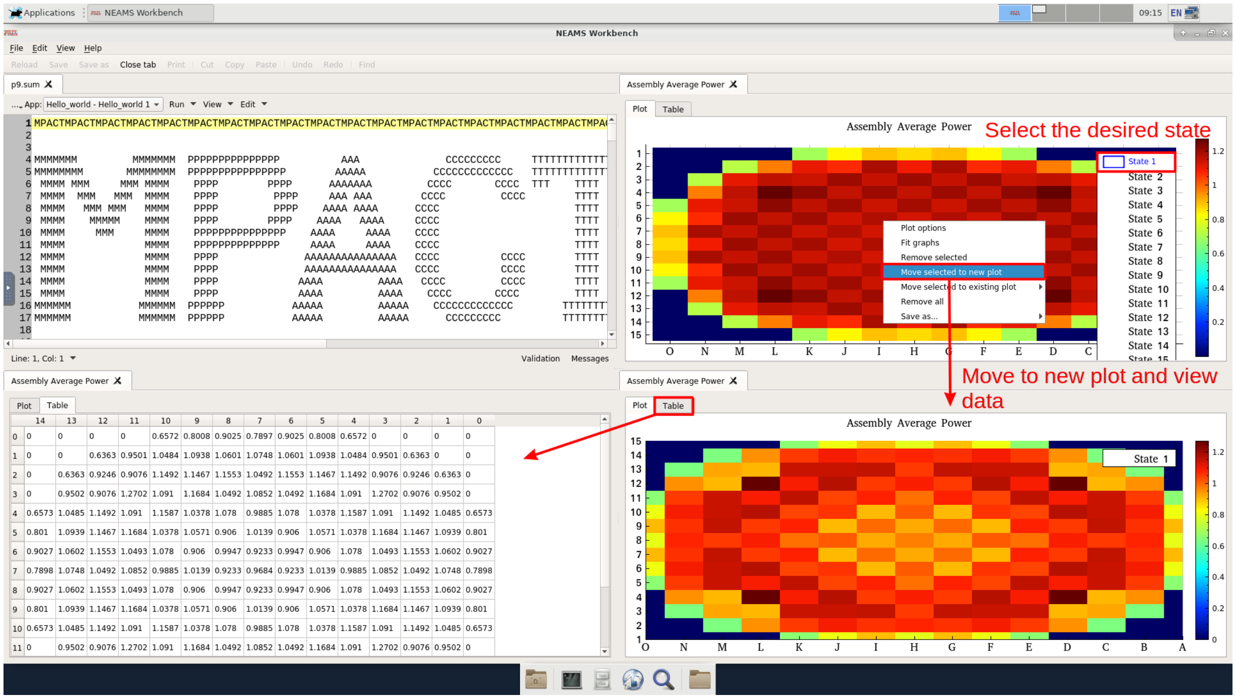
Data block	Status
Assembly average power	Supported
Maximum 3D pin power	Supported
Maximum 2D pin power	Supported
Lattice at max pin power	Supported
Assembly average exposure	Supported
Maximum 2D pin exposure	Supported
Assembly Exit density	Supported
Assembly Exit temperature	Supported
Selected 1D core data	Supported
Case summary	Supported
Assembly initial mass	Not supported

and selecting “Move selected to new plot” from the drop-down menu. Additionally, in the composite plot, which includes multiple states on the same figure, the table view is blank, but tabular data can still be viewed by isolating individual states in a single plot, as previously described.


Applications
NEAMS Workbench
Terminal - louimatt2@s...
07:04 EN
NEAMS Workbench
File Edit View Help
Reload Save Save as Close tab Print Cut Copy Paste Undo Redo Find
Coolant Density (g/cc) vs. Axial Height (cm)
Plot Table

	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9	State 10	State 11	State 12	State 1
1.581700e+1	0.743204	0.742448	0.741843	0.741736	0.741298	0.741186	0.741288	0.741068	0.741489	0.742019	0.741906	0.742018	0.7420
2.402800e+1	0.743202	0.742097	0.741288	0.741184	0.74072	0.740593	0.740684	0.740449	0.740853	0.741371	0.741247	0.741374	0.7413
3.223900e+1	0.7432	0.741578	0.740479	0.740381	0.73988	0.739735	0.739812	0.73956	0.739942	0.740444	0.740308	0.740458	0.7403
4.045000e+1	0.743198	0.740907	0.739438	0.739348	0.738804	0.738639	0.738702	0.73843	0.73879	0.739276	0.739128	0.739313	0.7391
4.866200e+1	0.743196	0.740099	0.738188	0.73811	0.737516	0.737329	0.737382	0.737088	0.737429	0.737905	0.737747	0.737976	0.7378
5.687300e+1	0.743194	0.739168	0.736743	0.736682	0.736033	0.735828	0.735874	0.735563	0.735886	0.736354	0.736191	0.736476	0.7362
6.508400e+1	0.743192	0.738128	0.735127	0.73509	0.734389	0.734171	0.734214	0.733892	0.734201	0.734663	0.734502	0.734852	0.7346
7.329500e+1	0.74319	0.736997	0.733382	0.733375	0.73262	0.732393	0.73244	0.732108	0.732411	0.732875	0.732721	0.733146	0.7329
7.710500e+1	0.743178	0.736458	0.732556	0.732565	0.731788	0.731558	0.731609	0.731275	0.731577	0.732043	0.731894	0.732354	0.7321
8.517000e+1	0.743176	0.735226	0.730652	0.730701	0.729879	0.729649	0.729713	0.729379	0.729683	0.730156	0.730025	0.730568	0.7303
9.323500e+1	0.743174	0.733913	0.72862	0.728718	0.72785	0.727626	0.72771	0.727379	0.727692	0.72818	0.728072	0.72871	0.7284
1.013000e+2	0.743172	0.732539	0.72649	0.726645	0.725738	0.725526	0.725636	0.725315	0.725642	0.726149	0.72607	0.726806	0.7265

Figure 9. Example of multistate 1D core summary data plots from the VERA progression problem: p9.



Select the desired state

Move to new plot and view data

```
MPACTMPACTMPACTMPACTMPACTMPACTMPACTMPACTMPACTMPACTMPACTMPACTMPACTMPACTMPACTMPACTMPACTMPACTMPA


MMMMMMM         MMMMMMM  PPPPPPPPPPPPPPP          AAA              CCCCCCCCC     TTTTTTTTTTTT
MMMMMMMM       MMMMMMMM  PPPPPPPPPPPPPPPP        AAAAA           CCCCCCCCCCCCC   TTTTTTTTTTTT
 MMMM MMM     MMM MMMM    PPPP        PPPP      AAAAAAA         CCCC       CCCC  TTT    TTTT
 MMMM  MMM   MMM  MMMM    PPPP         PPPP     AAA AAA        CCCC         CCCC        TTTT
 MMMM   MMM MMM   MMMM    PPPP         PPPP    AAAA AAAA      CCCC                      TTTT
 MMMM    MMMMM    MMMM    PPPP        PPPP    AAAA   AAAA     CCCC                      TTTT
 MMMM     MMM     MMMM    PPPPPPPPPPPPPPP    AAAA     AAAA    CCCC                      TTTT
 MMMM             MMMM    PPPPPPPPPPPPPP     AAAA     AAAA    CCCC                      TTTT
 MMMM             MMMM    PPPP              AAAAAAAAAAAAAAA   CCCC          CCCC        TTTT
 MMMM             MMMM    PPPP              AAAAAAAAAAAAAAA   CCCC          CCCC        TTTT
 MMMM             MMMM    PPPP             AAAA         AAAA   CCCC         CCCC        TTTT
 MMMM             MMMM    PPPP             AAAA         AAAA    CCCC       CCCC         TTTT
MMMMMM           MMMMMM  PPPPPP           AAAAA         AAAAA    CCCCCCCCCCCCC        TTTTTTT
MMMMMM           MMMMMM  PPPPPP           AAAAA         AAAAA      CCCCCCCCC          TTTTTTT

```

	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0.6572	0.8008	0.9025	0.7897	0.9025	0.8008	0.6572	0	0	0	0
1	0	0	0.6363	0.9501	1.0484	1.0938	1.0601	1.0748	1.0601	1.0938	1.0484	0.9501	0.6363	0	0
2	0	0.6363	0.9246	0.9076	1.1492	1.1467	1.1553	1.0492	1.1553	1.1467	1.1492	0.9076	0.9246	0.6363	0
3	0	0.9502	0.9076	1.2702	1.091	1.1684	1.0492	1.0852	1.0492	1.1684	1.091	1.2702	0.9076	0.9502	0
4	0.6573	1.0485	1.1492	1.091	1.1587	1.0378	1.078	0.9885	1.078	1.0378	1.1587	1.091	1.1492	1.0485	0.6573
5	0.801	1.0939	1.1467	1.1684	1.0378	1.0571	0.906	1.0139	0.906	1.0571	1.0378	1.1684	1.1467	1.0939	0.801
6	0.9027	1.0602	1.1553	1.0493	1.078	0.906	0.9947	0.9233	0.9947	0.906	1.078	1.0493	1.1553	1.0602	0.9027
7	0.7898	1.0748	1.0492	1.0852	0.9885	1.0139	0.9233	0.9684	0.9233	1.0139	0.9885	1.0852	1.0492	1.0748	0.7898
8	0.9027	1.0602	1.1553	1.0493	1.078	0.906	0.9947	0.9233	0.9947	0.906	1.078	1.0493	1.1553	1.0602	0.9027
9	0.801	1.0939	1.1467	1.1684	1.0378	1.0571	0.906	1.0139	0.906	1.0571	1.0378	1.1684	1.1467	1.0939	0.801
10	0.6573	1.0485	1.1492	1.091	1.1587	1.0378	1.078	0.9885	1.078	1.0378	1.1587	1.091	1.1492	1.0485	0.6573
11	0	0.9502	0.9076	1.2702	1.091	1.1684	1.0492	1.0852	1.0492	1.1684	1.091	1.2702	0.9076	0.9502	0

Figure 10. Manipulating 2D multi-state data.

4.2 H5 PLOTTING

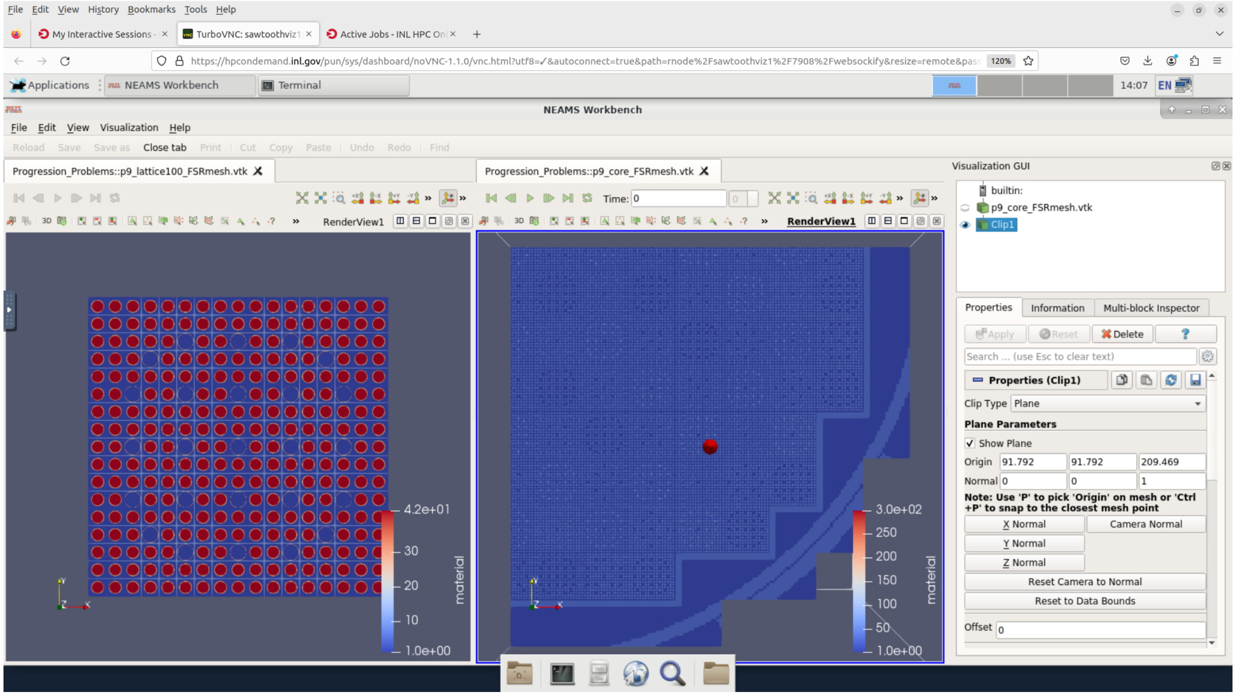
At the time of writing, .h5 processing and visualization are not supported in Workbench, but .h5 files can be visualized using VERAView, an open-source plotting application [15].

4.3 VISUALIZING OTHER OUTPUT FILES

Additional "edits" (output visualization files) can be specified in the input file by including the "vis_edits" card in the "[EDITS]" block, and (optionally) including the "edits" card in any given "[STATE]" block. More information is given in the VERA Input User's Manual [14]. The vis_edits card is used to specify how many visualization files are produced.

- none: generates no visualization files.
- core: only prints core-level edits. This produces a single .pvtu file that can be opened in Workbench.
- fsr: prints all available edits on a flat source region basis. This produces .vtk files that can be opened in Workbench (for reference, this option produces a number of .vtk files on the order of the number assemblies, which can become quite cumbersome for larger problems).

The "edits" card in the "[STATE]" block specifies which output quantities are to be included in the visualization (.vtk or .pvtu) files. For example, including "edits fsr_flux_2g pin_exposures" in a given "[STATE]" block will append both two-group fluxes and pin exposures to the output visualization file for that state; an example is given in Fig. 12. If none are specified, then the visualization files only display geometry/material information as in Fig. 11. To view these files, simply open them in Workbench, and a ParaView visualization will automatically be initiated. Examples are shown in Fig. 11 and Fig. 12). To change which output quantity is being viewed, simply right-click the displayed model, select "Color By", and select the desired output quantity from the drop-down menu. Note that the mesh edges can also be viewed by right-clicking



Clip Type Plane

Plane Parameters

☑ Show Plane

Origin 91.792 91.792 209.469

Normal 0 0 1

Note: Use 'P' to pick 'Origin' on mesh or 'Ctrl +P' to snap to the closest mesh point

Offset 0

Figure 11. Visualizing a core fsr mesh .vtk file in ParaView.

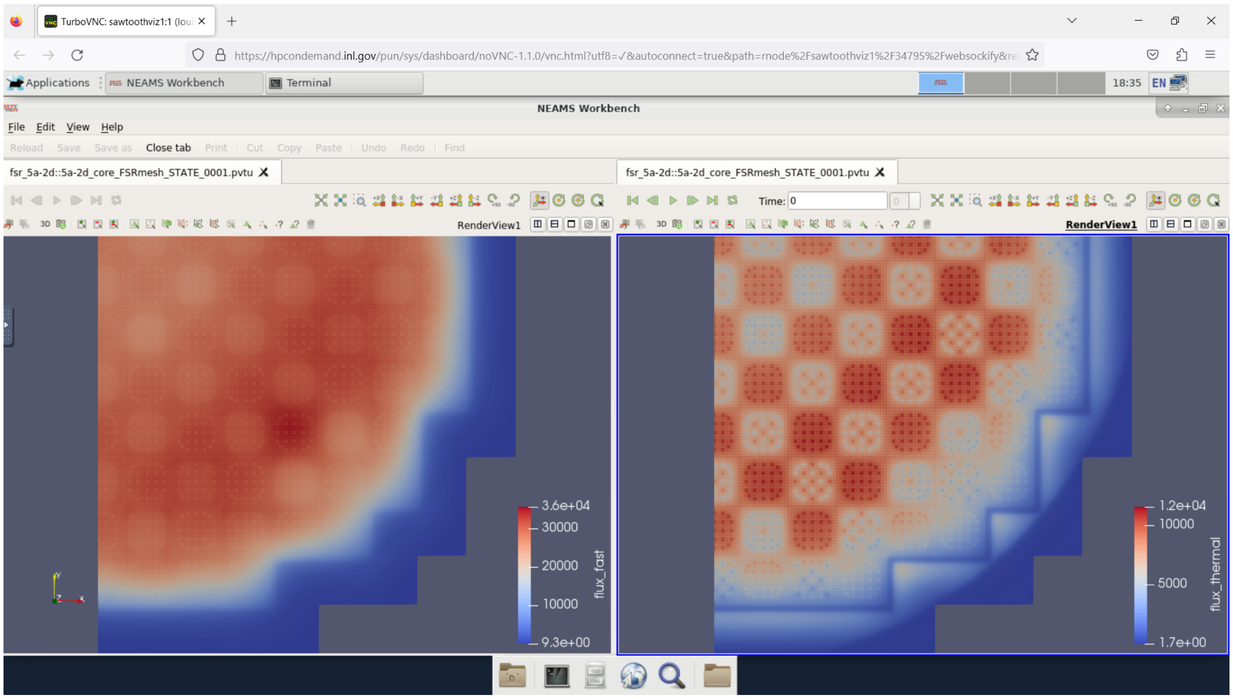


Figure 12. Visualizing a core fsr flux map .pvtu file.

the model, selecting "Representation", and checking "Surface with Edges". More information on ParaView features can be found in the Workbench help documentation and the ParaView User's Guide.

For more complex problems like the p9 progression problem, core-level .vtk files can become extremely large and will load quite slowly in ParaView. In general, .pvtu files load more quickly in ParaView and are preferred for larger visualizations. When viewing large visualization files, running a Workbench session on a visualization node provides greatly improved performance. A visualization node can be requested by opening the “Interactive Apps” bar on the HPC OnDemand Desktop and clicking “Linux Desktop with Visualization”.

5. SUMMARY AND FUTURE IMPROVEMENTS

This manual describes the configuration of VERA in Workbench, provides a demonstration of syntax highlighting and input validation, presents instructions for running input files and details on execution options and job termination, and provides instructions for visualizing the results via integrated Workbench processors and ParaView. In the future, a full integration of the capabilities of VERAView into Workbench—specifically .h5 processing and 3D core data visualization—will provide users the ultimate streamlined experience and will allow integration of the user's entire workflow for crud, vessel fluence, thermal hydraulic, and fuel performance calculations.

6. ACKNOWLEDGMENTS

This work was supported in part by the US Department of Energy Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships (SULI) program.

This research made use of the resources of the High Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the US Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

REFERENCES

[1] J. A. Turner, K. Clarno, M. Sieger, R. Bartlett, B. Collins, R. Pawlowski, R. Schmidt, and R. Summers. The Virtual Environment for Reactor Applications (VERA): Design and Architecture. *Journal of Computational Physics*, 326:544–568, 2016.

[2] B. Kochunas et al. VERA Core Simulator Methodology for PWR Cycle Depletion. *Nuclear Science and Engineering*, 185(1):217–231, 2017.

[3] Eva E Davidson, Andrew T Godfrey, Tara Pandya, and Katherine Royston. Demonstration of comprehensive ex-vessel fluence capability. Technical report, Oak Ridge National Laboratory, 2019.

[4] *VERA-CS Modeling and Simulation of PWR Main Steam Line Break Core Response to DNB*, volume Volume 4: Computational Fluid Dynamics (CFD) and Coupled Codes; Decontamination and Decommissioning, Radiation Protection, Shielding, and Waste Management; Workforce Development, Nuclear Education and Public Acceptance; Mitigation Strategies for Beyond Design Basis Events; Risk Management of *International Conference on Nuclear Engineering*, 06 2016. V004T10A026.

[5] V. N. Kucukboyaci, B. Kochunas, T. J. Downar, A. Wysocki, and R. K. Salko. Evaluation fo VERA-CS Transient Capability for Analyzing the AP1000 Reactor Control Rod Ejection Accident. In *Proceedings of PHYSOR-2018*, 2018.

[6] Ryan Sweet, Ian Greenquist, Aaron Graham, Rick Trotta, Clay Lietwiler, and Faisal Odeh. Coupled neutronics, thermal hydraulics, and fuel performance simulations of a natural circulation based smr. Technical report, Oak Ridge National Laboratory, 2023.

[7] Shane Stimpson, Jeffrey Powers, Kevin Clarno, Roger Pawlowski, Russell Gardner, Stephen Novascone, Kyle Gamble, and Richard Williamson. Pellet-clad mechanical interaction screening using vera applied to watts bar unit 1, cycles 1–3. *Nuclear Engineering and Design*, 327:172–186, 2018.

[8] Benjamin S Collins, Jack Galloway, Robert Salko Jr, Kevin Clarno, Aaron Wysocki, Brett Okhuysen, and Anders David Andersson. Whole core crud-induced power shift simulations using vera. Technical report, Oak Ridge National Laboratory, 2018.

[9] R. A. Lefebvre, B. R. Langley, P. Miller, M. Delchini, M. L. Baird, and J. P. Lefebvre. *NEAMS Workbench Status and Capabilities*. Oak Ridge National Laboratory, 2019.

[10] Idaho National Laboratory. Nuclear Computational Resrouce Center. https://inl.gov/ncrc.

[11] Marco Delchini and Robert Alexander Lefebvre. Documentation on how to run the neams workbench gui on sawtooth. Technical report, Oak Ridge National Laboratory, 2022.

[12] Srdjan Simunovic. Verain. Technical report, Oak Ridge National Laboratory, 2015.

[13] Andrew Godfrey. VERA Core Physics Benchmark Progression Problem Specifications. Technical Report CASL-U-2012-0131-004, Oak Ridge National Laboratory, 2014.

[14] Scott Palmtag, Andrew Godfrey, Mark Baird, and Erik Walker. VERAIn User's Manual. Technical Report ORNL/SPR-2022/2509, Oak Ridge National Laboratory, 2022.

[15] R. W. Lee, B. S. Collins, and A. T. Godfrey. VERAView. Computer Software. https://osti.gov//servlets/purl/1354890.