ORNL/TM-2019/1208

NEAMS Workbench and BISON Fuel Performance Remote Application Configuration

Kaylee Cunningham
Robert A. Lefebvre
Jeffrey Powers
L. Paul Miller
Mark L. Baird
Brandon R. Langley

09/30/2019

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2019/1208

Reactor Nuclear Systems Division

NEAMS WORKBENCH AND BISON FUEL PERFORMANCE REMOTE APPLICATION CONFIGURATION

Kaylee Cunningham
Robert A. Lefebvre
Jeffrey Powers
L. Paul Miller
Mark L. Baird
Brandon R. Langley

09/30/2019

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

ABSTRACT

The mission of the US Department of Energy (DOE) Nuclear Energy Advanced Modeling and Simulation (NEAMS) Program is to develop, apply, deploy, and support state-of-the-art predictive modeling and simulation tools for the design and analysis of current and future nuclear energy systems. This is accomplished using computing architectures that range from laptops to leadership class facilities. The NEAMS Workbench [1,2,3] initiative will facilitate the transition from conventional computer codes to high-fidelity ones by providing a common analysis environment and interface for model creation, review, execution, output review, and visualization for integrated codes. The BISON fuel performance code [4] is a finite element method (FEM)-based software being developed by multiple organizations led by the Idaho National Laboratory for the US Department of Energy Office of Nuclear Energy. BISON can simulate a variety of fuel and cladding compositions and geometries in varying dimensionalities, depending upon the problem type and complexity, including one-dimensional (1D), 1.5D, 2D axially symmetric (2D-rz), 2D radial-azimuthal (2D-rθ), or 3D models. The NEAMS Workbench allows execution of the BISON fuel performance code on local and remote compute resources. In this report, the detailed instructions of configuring the NEAMS Workbench for remote execution of the BISON fuel performance code on a portable batch system (PBS)-scheduled compute resource, as well the transfer of associated Bison files, is illustrated.

1. NEAMS WORKBENCH AND BISON REMOTE APPLICATION CONFIGURATION

The mission of the US Department of Energy (DOE) Nuclear Energy Advanced Modeling and Simulation (NEAMS) Program is to develop, apply, deploy, and support state-of-the-art predictive modeling and simulation tools for the design and analysis of current and future nuclear energy systems. This is accomplished using computing architectures that range from laptops to leadership class facilities. The NEAMS Workbench [1,2,3] initiative will facilitate the transition from conventional tools to high-fidelity tools by providing a common analysis environment and interface for model creation, review, execution, output review, and visualization for integrated codes. The BISON fuel performance code [4] is a finite element method (FEM)-based software tool being developed by multiple organizations led by the Idaho National Laboratory for the US Department of Energy Office of Nuclear Energy. BISON can simulate a variety of fuel and cladding compositions and geometries in varying dimensionality, depending upon the problem type and complexity, including one-dimensional (1D), 1.5D, 2D axially symmetric (2D-rz), 2D radial-azimuthal (2D-rθ), or 3D models.

The NEAMS Workbench can be used with a local or remote installation of BISON. The BISON remote application job launch has been tested on Windows, Mac, and Linux operating systems. The steps presented below are intended to provide an overview and guide to enable the setup and execution of the NEAMS Workbench and BISON, specifically the remote server configuration.

These instructions assume that the user has installed the NEAMS Workbench on his or her local machine, or the *client*, and on the remote machine, or the *server*. Additionally, it is assumed that a version of BISON that is compatible with NEAMS Workbench is installed on the server. Installation instructions for NEAMS Workbench may be obtained by contacting the NEAMS Workbench team (nwb-help@ornl.gov) or following the instructions listed in the README file, while installation instructions for BISON may be found in the Bison Workshop slides [5].

The NEAMS Workbench software bundle is deployed as a compressed file. Once fully decompressed, the user can double-click or use the command line to invoke the `NEAMS-Workbench/bin/Workbench` executable.

1.1 NEAMS WORKBENCH INTEGRATED APPLICATION REMOTE JOB LAUNCH OVERVIEW

The NEAMS Workbench supports scheduling integrated application inputs on servers that contain PBS-based schedulers (Torque-Maui/MOAB qsub, IBM LSF bsub, etc.) and no scheduler (i.e., single-box machines with available cores). This is done through the NEAMS Workbench runtime environment (RTE). Each application (BISON [4], ARC [6], MOOSE-Apps [7], Dakota [8], SCALE [9], CTFFuel [10], etc.) that is integrated into the NEAMS Workbench receives an RTE to assist the user and the NEAMS Workbench in managing the execution of the application on both local and remote machines. By installing the NEAMS Workbench on both the client (local) and the server (remote) machines, the NEAMS Workbench RTE enables a bridge over the network.

The network bridge must be established for each application-client-server combination. The NEAMS Workbench has a native application—Setup_remote—to enable the configuration of the remote application. Configuration options include the server network address, location of application, the NEAMS Workbench, and the scheduler, as installed on the server and the scheduler options.

The secure shell (SSH) protocol is used for communication across the network and requires that the client be authorized for passwordless connection via the use of an SSH-authorized key. Details are provided in the instructions below.

1. Ensure NEAMS Workbench is installed on the client and server (via README instructions or contacting the NEAMS Workbench team: nwb-help@ornl.gov).
2. Ensure BISON is installed on the server [5].
3. Configure the client and server for passwordless SSH connection [see Section 1.2].
4. Using the client NEAMS Workbench, do the following:
    a. Enable the NEAMS Workbench's Setup Remote application.
    b. Create the application-server Setup_remote input.
    c. Run application-server setup remote input to generate the NEAMS Workbench Bison remote RTE.
    d. Configure the BISON remote RTE.
    e. Open any BISON input and select the BISON remote RTE to activate input validation, autocompletion, syntax highlighting, and job launch capabilities.
    f. Edit input as needed.
    g. Use the BISON remote RTE to launch the BISON input from your client to the server.
    h. Wait for the job to finish.
    i. Open files associated with the input file.

1.2 SSH CONFIGURATION

In order to configure SSH from the client machine to the server, the client's public SSH key must be appended to the user's list of authorized keys residing on the server. This may be done through any terminal emulator that supports SSH. The following statement should be sufficient to do so on Mac and Linux clients.

```
ssh <server> 'cat >> .ssh/authorized_keys && chmod 644 .ssh/authorized_keys' <
~/.ssh/id_rsa.pub
```

Replace `server` with the name of the server. On Windows clients that have OpenSSH available the following statement should be sufficient.

```
type .ssh\id_rsa.pub | ssh <server> "cat >> .ssh/authorized_keys && chmod 644
.ssh/authorized_keys"
```

If OpenSSH is not available a `keygen.py` python utility is deployed as part of the NEAMS Workbench software bundle. The NEAMS Workbench includes a Python environment with the necessary Python packages to enable SSH transactions. The following statement will guide the user through the passwordless SSH setup on Windows when executed from within the NEAMS Workbench installation folder.

```
.\rte\entry.bat .\rte\util\keygen.py
```

The passwordless SSH can be tested by attempting to `ssh` to the server via `ssh <server>` on Linux and Mac or `Putty` on Windows. The user should no longer observe the need to specify a password.

1.3 NEAMS WORKBENCH BISON CONFIGURATION STEPS

From the client, open the NEAMS Workbench (see Figure 1) by double-clicking the "Workbench.exe" file (C://NEAMSWorkbench/bin/Workbench.exe).

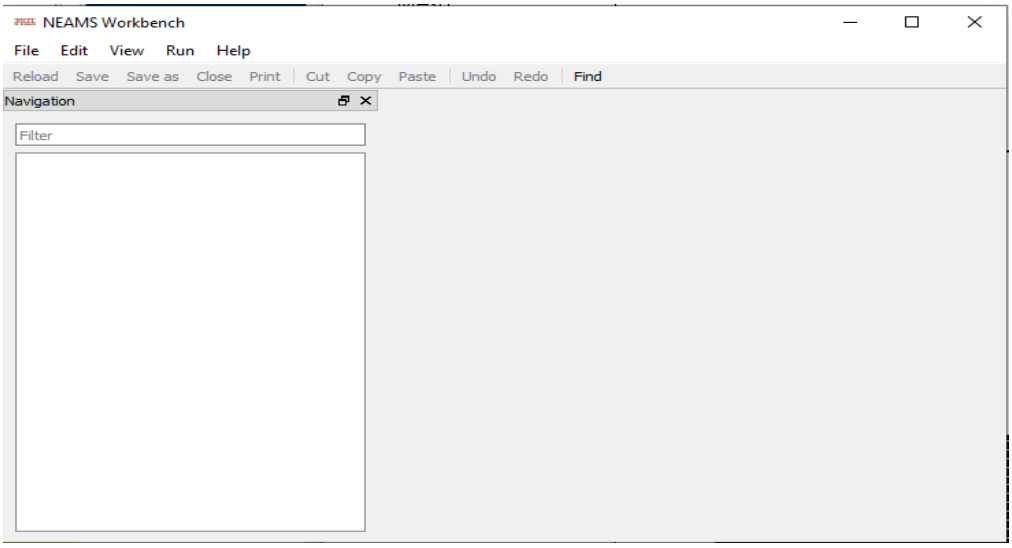

Figure 1. The NEAMS Workbench at Startup on Windows 10.

The first time the NEAMS Workbench opens, the User Documentation will open in the default internet browser, which is shown in Figure 2.

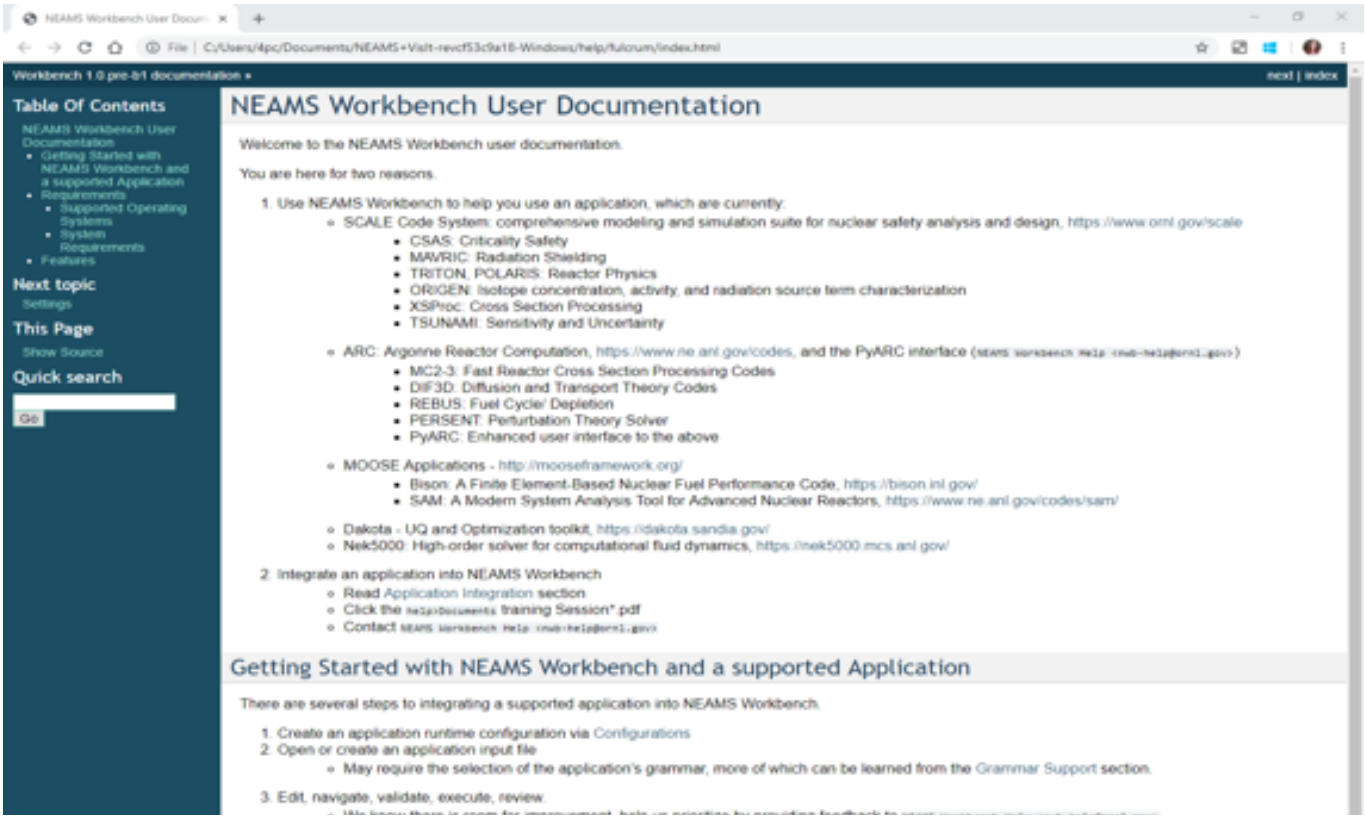

Figure 2. The NEAMS Workbench user documentation at startup.

The User Documentation will not automatically display for future startups of the NEAMS Workbench. It is accessible via the `Help>Help Documentation` menu, as shown in Figure 3.

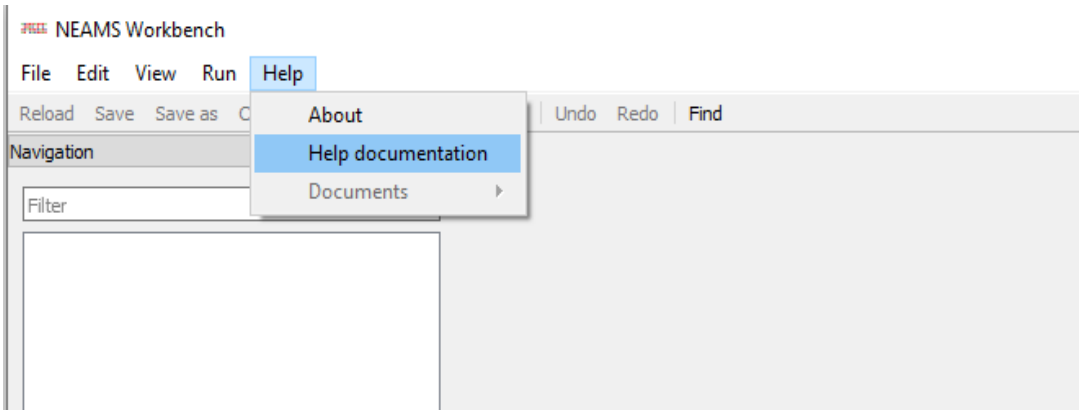

Figure 3. Accessing the NEAMS Workbench user documentation.

Because the NEAMS Workbench contains many integrated applications, no applications are configured to execute by default. The Setup_remote application is needed to configure the application on the remote server. Configuring the Setup_remote application will allow the creation, edit, and execution of Setup_remote inputs.

Click the `File>Configurations` menu item as demonstrated in Figure 4 to access the application configuration panel:

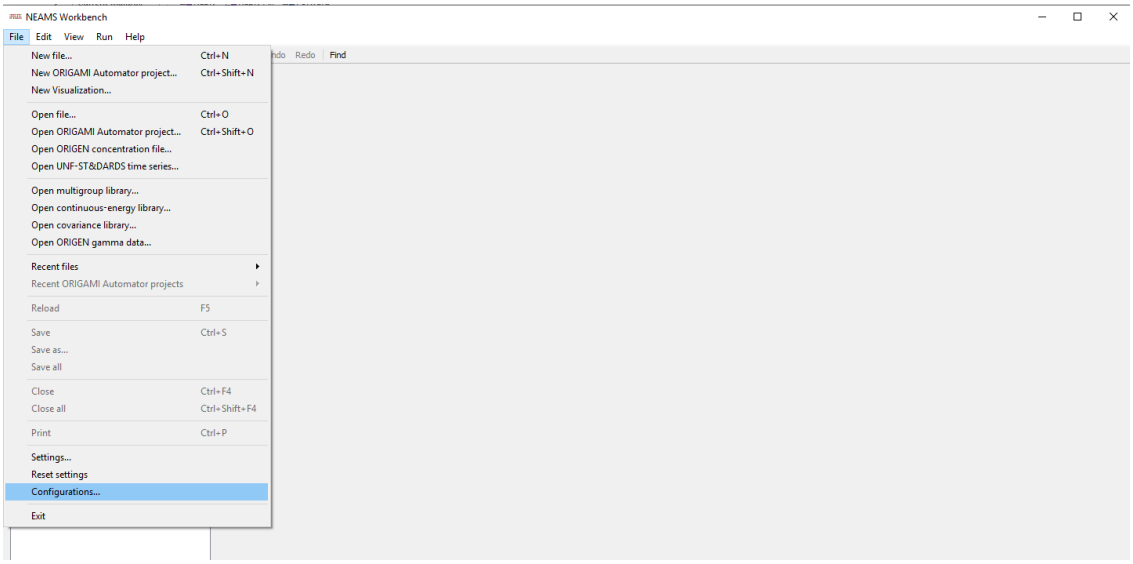

Figure 4. Accessing the application configurations panel.

A local MOOSE-BISON application is the default enable application. Each configured application can have one or more application environment configurations. This facilitates variations in application execution with respect to application version or environment. To configure the Setup_remote, click the `Add...` button (see Figure 5):

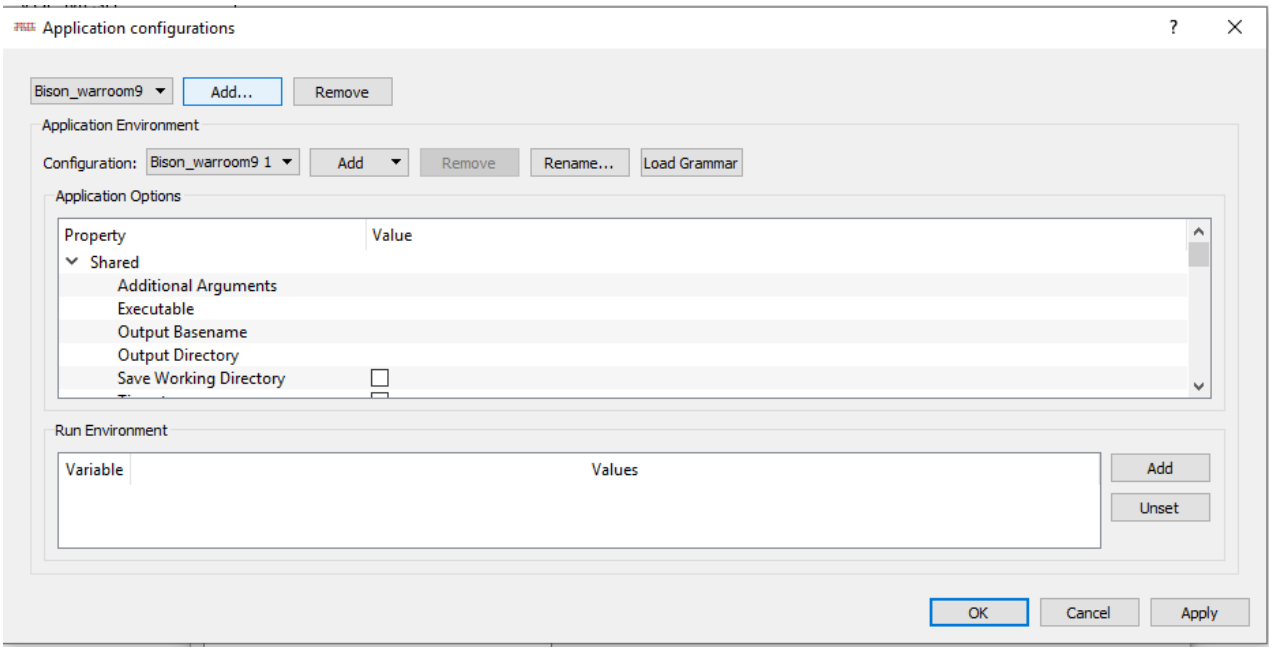

Figure 5. The application configuration add button.

The list of applications that can be configured will be displayed. Select the Setup_remote application and click the OK button, as is shown in Figure 6.

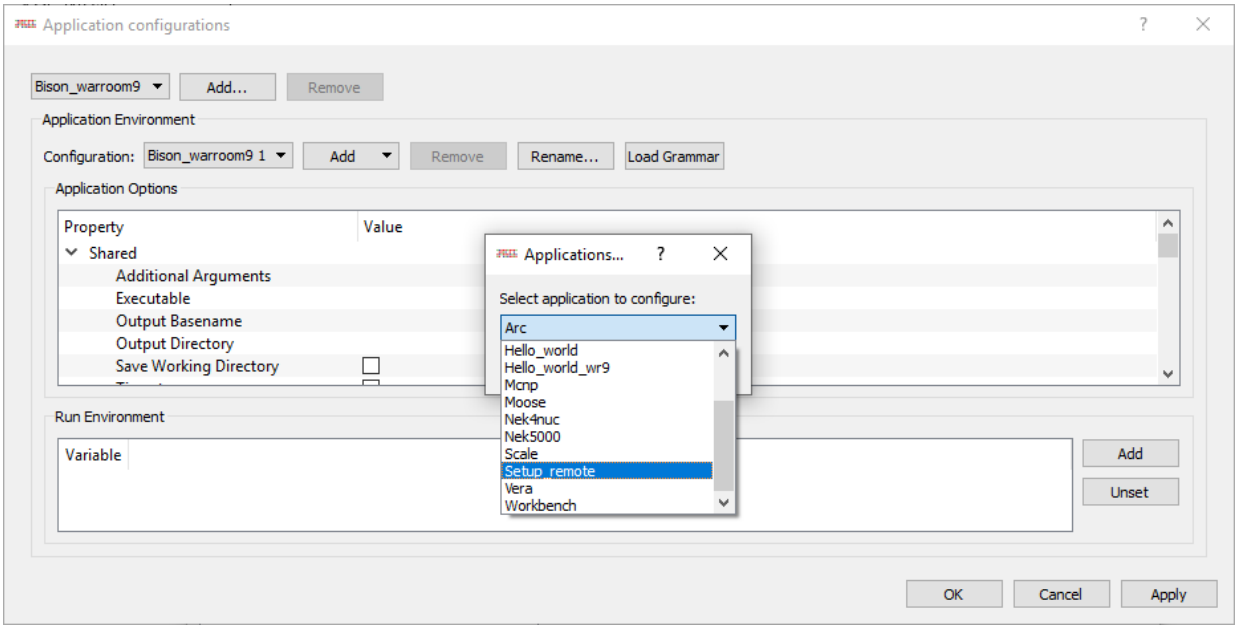

Figure 6. Selecting Setup_remote for configuration.

The default Setup_remote application configuration will be displayed (see Figure 7):

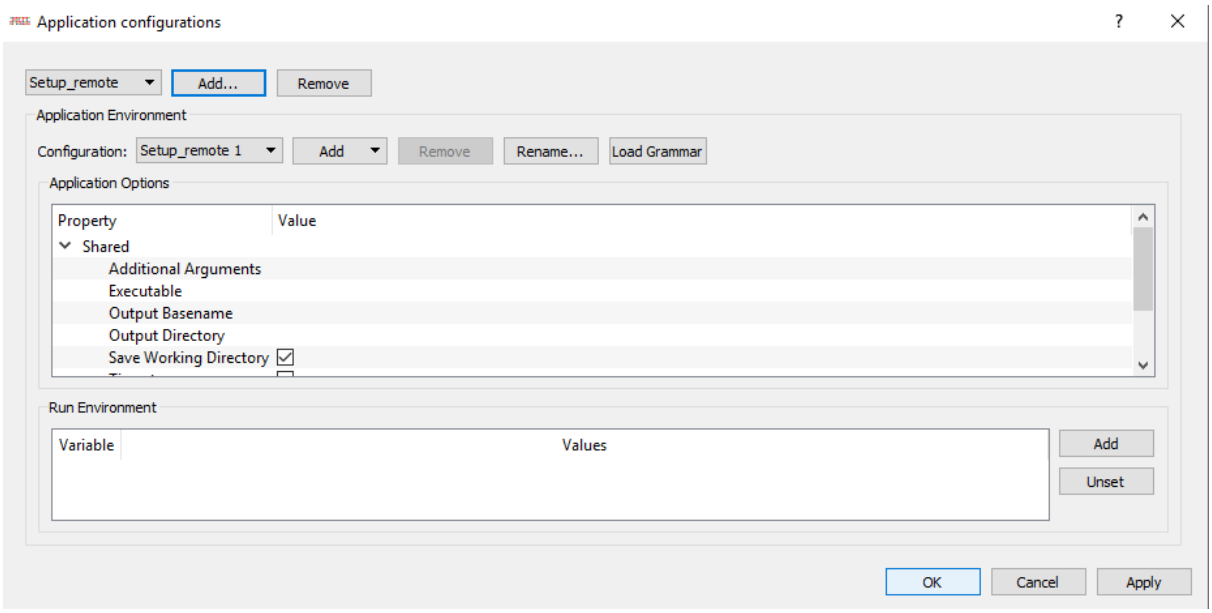

Figure 7. Setup_remote default application configuration.

Click the Setup_remote application's `Load Grammar` button as shown in Figure 8 to have NEAMS Workbench load input autocompletion and validation capabilities for the Setup_remote application, and click `OK`:

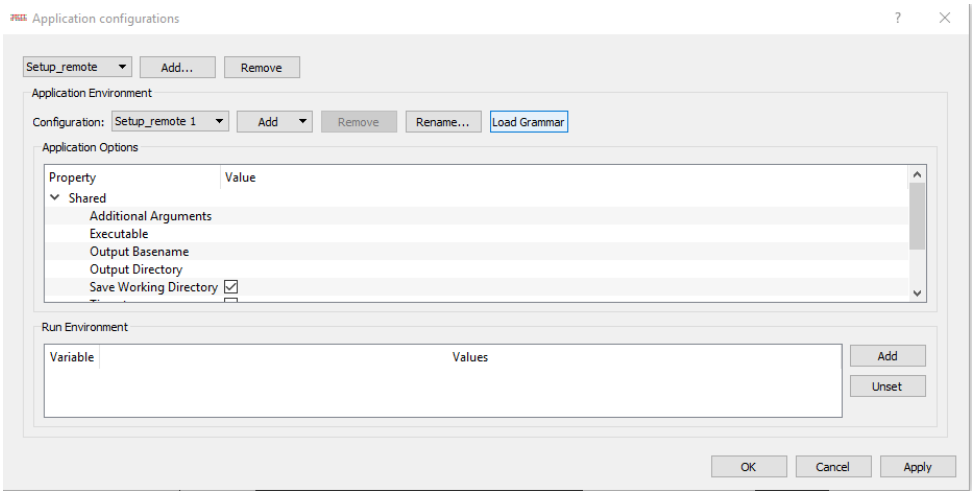

Figure 8. The Setup_remote Application's Load Grammar button.

1.3.1 Create Setup_remote input

With the Setup_remote application configured it is time to create the Setup_remote application input that will create the client-server-application remote RTE. Click the `File>New File` menu item (shown in Figure 9):

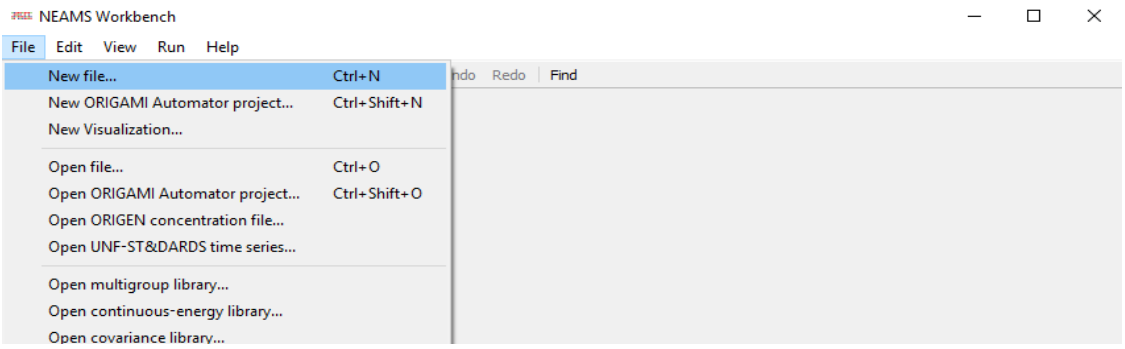

Figure 9. New File creation menu item.

Select the Setup_remote file type, specify the directory where the new file will be created, provide the file name (e.g., "application_server" is common) and click the `Save` button as in Figure 10:

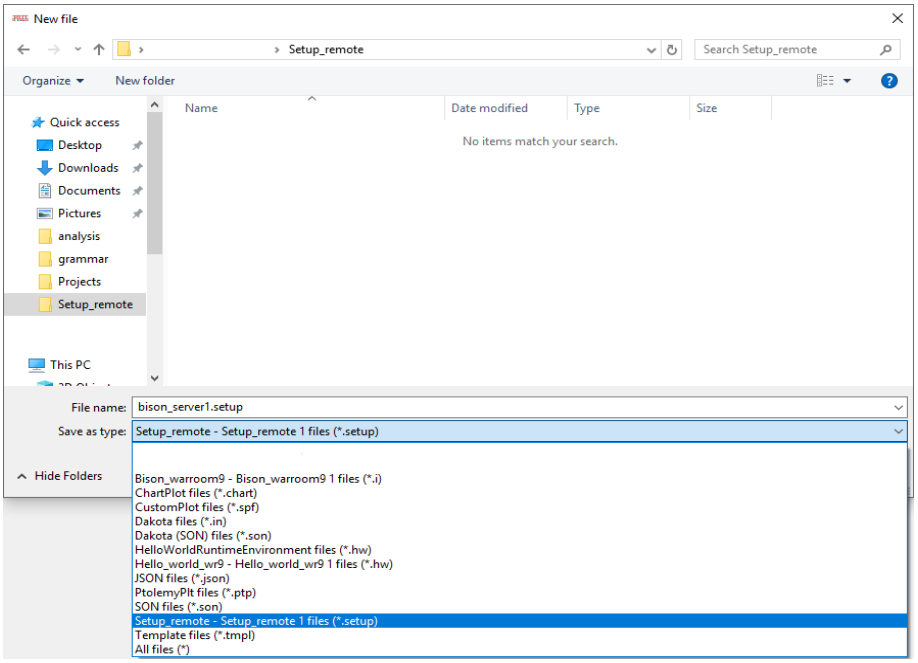

Figure 10. Creation of a new Setup_remote file with extension type and intuitive application-server file name.

The new file will be displayed. The unambiguous file extension (*.setup) allows the NEAMS Workbench to automatically select the language grammar to use for input processing. For ambiguous file extensions (*.inp), the language grammar may need to be selected manually. Confirm the top dropdown menus list "Setup_remote – Setup_remote 1" "document" and "Setup_remote – Setup_remote 1," as demonstrated in Figure 11.

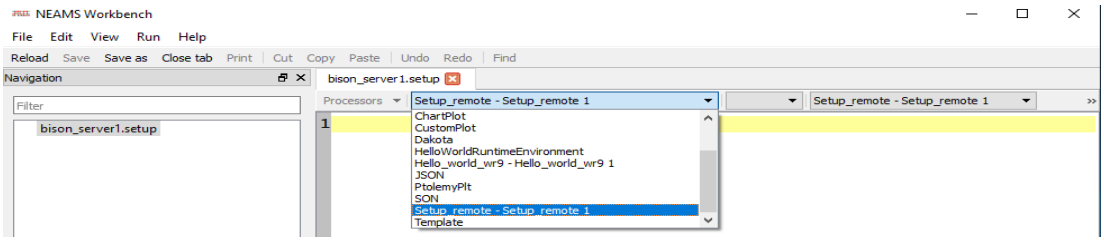

Figure 11. The NEAMS Workbench Language Grammar selection widget.

Ensure focus is in the document text editor by clicking in the text editor. Using the NEAMS Workbench input autocomplete key combination, CTRL+SPACE will display the available input autocompletion options (see Figure 12). Select the `setup_properties - bison` option:

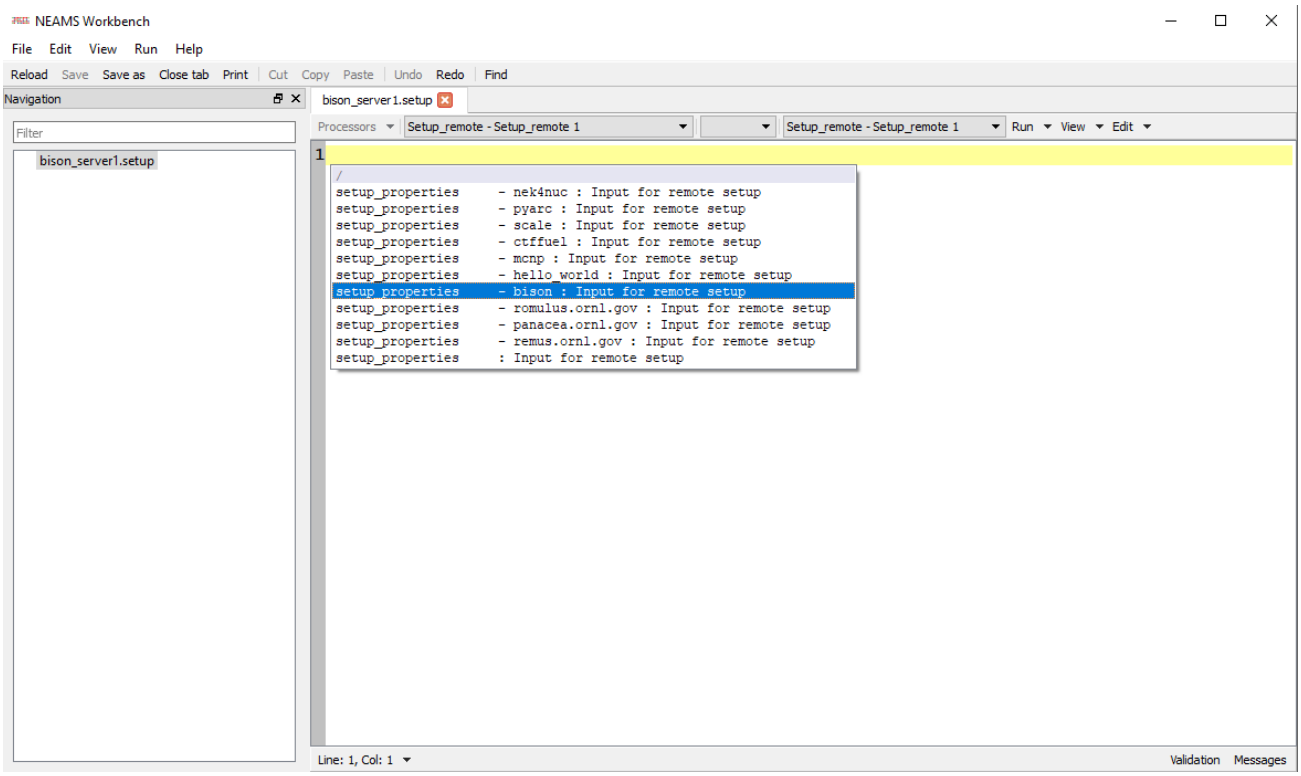

Figure 12. Input Autocompletion of the Setup_remote Bison input variant.

This will display a boilerplate input configured for remote BISON application setup, but this will require the update of the specific username, server name, NEAMS Workbench server, and server scheduler location (see Figure 13).

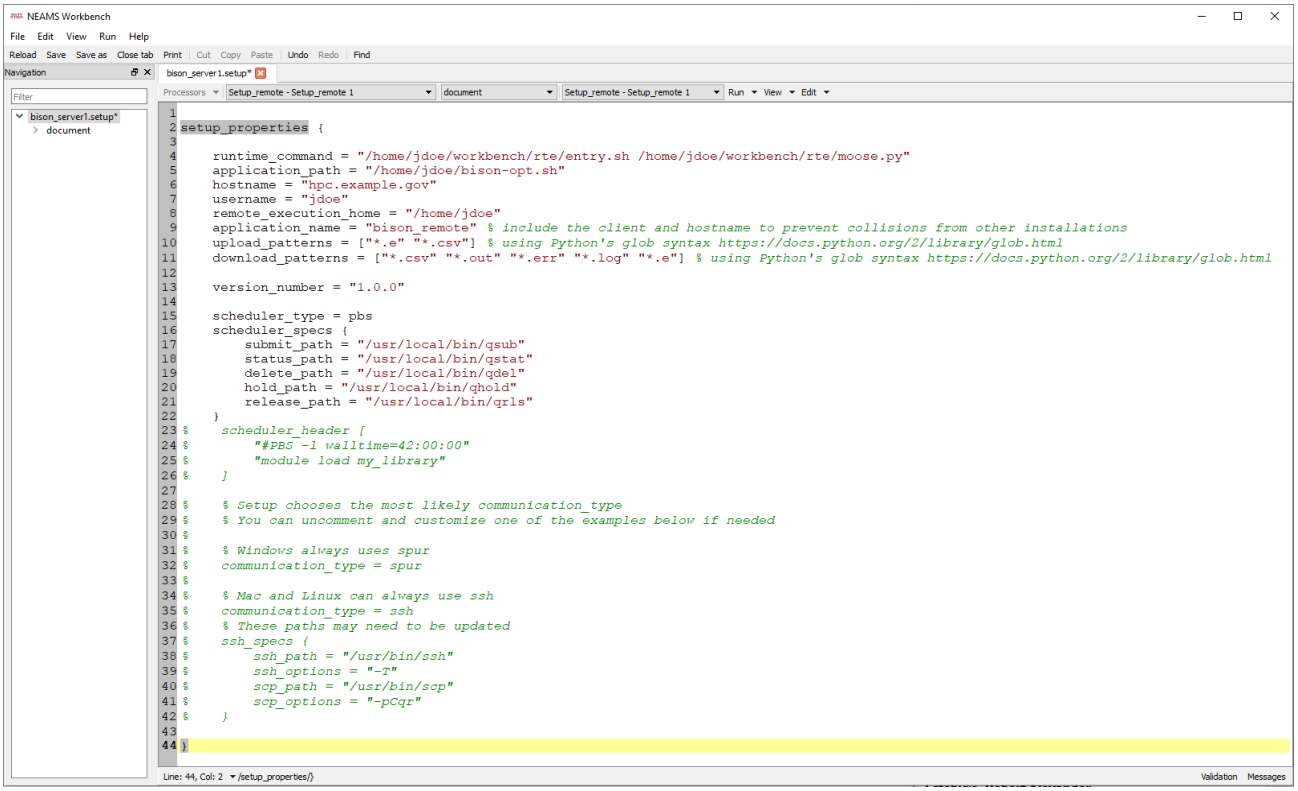

Figure 13. Setup_remote application BISON input variant.

The default is configured for a PBS-based scheduler (lines 15–21). The following steps are required to complete the Setup_remote input.

1. Update the runtime_command to use the correct server location for the NEAMS Workbench
    a. Replace '/home/jdoe/workbench' with <LOCATION-TO-SERVER-Workbench> Bison
2. Update **username** and **remote_execution_home** from 'jdoe' to your username
3. Optionally, update the application_name to be more specific. For example, replace generic 'remote' with server name bison_server1

Note the `application_path` is a shell script located in the home directory on the server. This script should look like the following:

```
#!/bin/bash

for i in $@; do if [ ${i##*.} == i ]; then dos2unix $i; fi ; done

/projects/Bison/bison-opt "$@"
```

Update `/projects/Bison/bison-opt` to be the path to the BISON executable on the server. This script scrubs Windows carriage returns from the input file using the dos2unix utility. This is necessary, as at the time of this writing the BISON application fails to process inputs containing Windows carriage returns.

1.3.2 Running the Setup_remote application input to create remote application runtime

Running the Setup_remote application file via the `Run` button at the top right of the input editor will configure the bison_server1 remote application runtime and create the named-application runtime (where

the name is specified as `application_name`). Select the `Messages` panel button at the bottom right of the input editor to display Setup_remote execution messages.

In Figure 14, the items important to notice are line 26, indicating that server credentials are valid, line 34, indicating that the new remote runtime class has been written, and line 39, indicating that the remote runtime's command line options have been successfully downloaded. Once these are confirmed, the new remote runtime application can be configured. The line numbers may vary with NEAMS Workbench release.

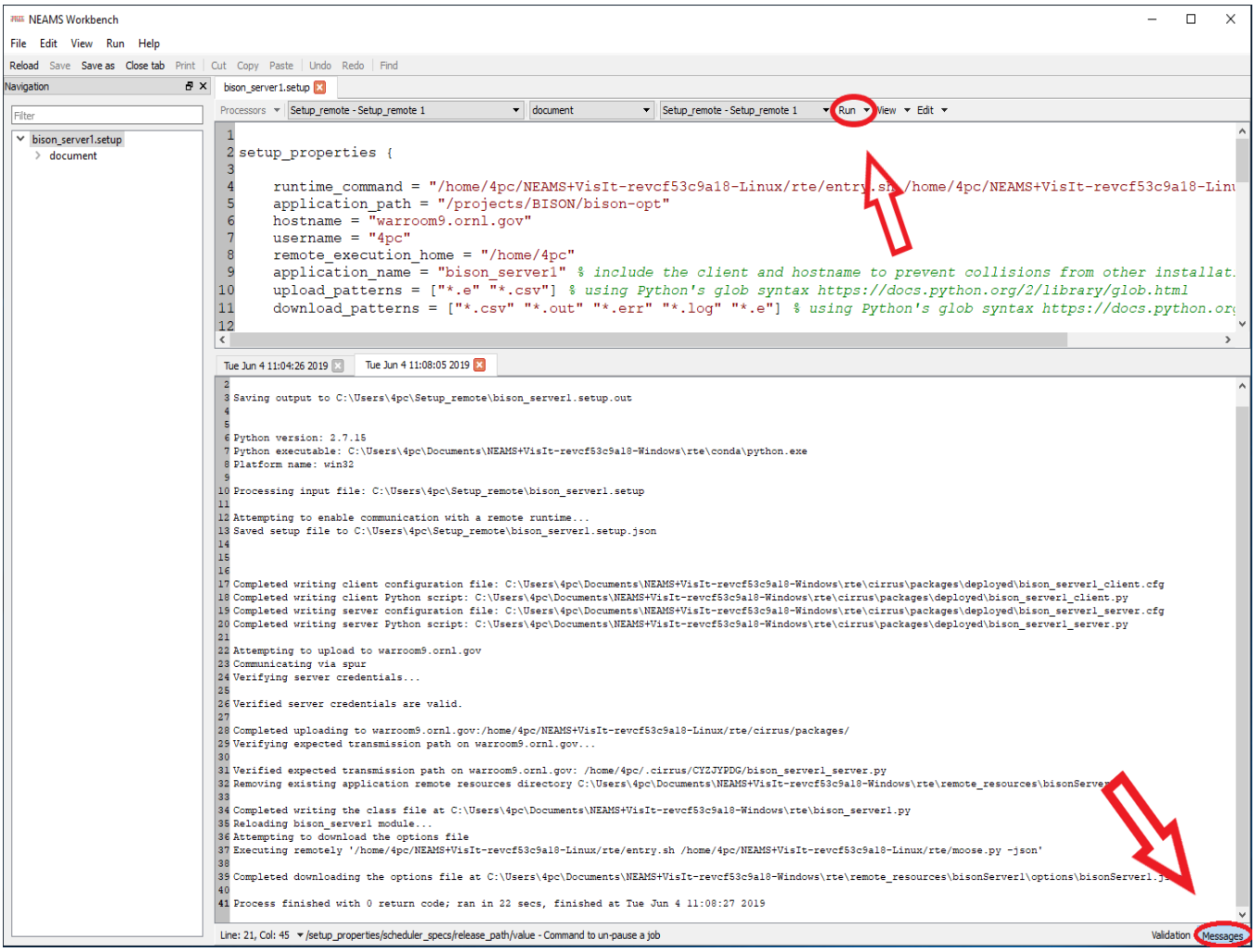

Figure 14. Example execution of Setup_remote highlighting the run button and execution messages panel button.

1.4 CONFIGURING THE SETUP_REMOTE GENERATED REMOTE APPLICATION RUNTIME

The execution of the Setup_remote application generates an application server–specific NEAMS Workbench remote runtime, but it does not activate this run configuration. To activate this new remote configuration, open the NEAMS Workbench configuration dialog via the `File>Configurations…` menu item.

Click the configuration panel's `Add…` button, and the new application runtime will be listed for selection. Select the application, click `OK>Load Grammar>Apply` and `OK` to close the configuration dialog, as is illustrated in Figure 15:

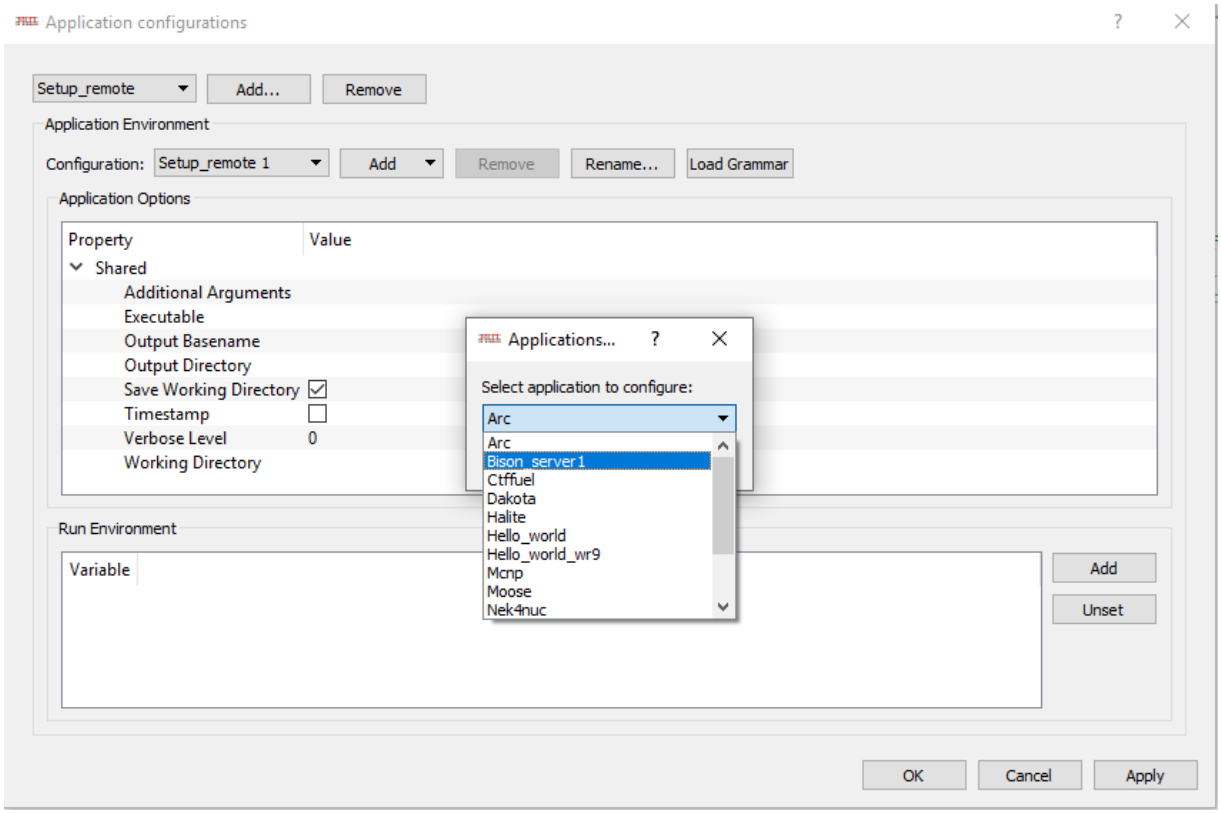

Figure 15. Configuring the new remote application runtime (bison_server1).

At this point the remote application runtime has been configured and is available for use by the NEAMS Workbench.

Select an application input via the `File>Open file…` menu item and click the `Open` button (see Figure 16):

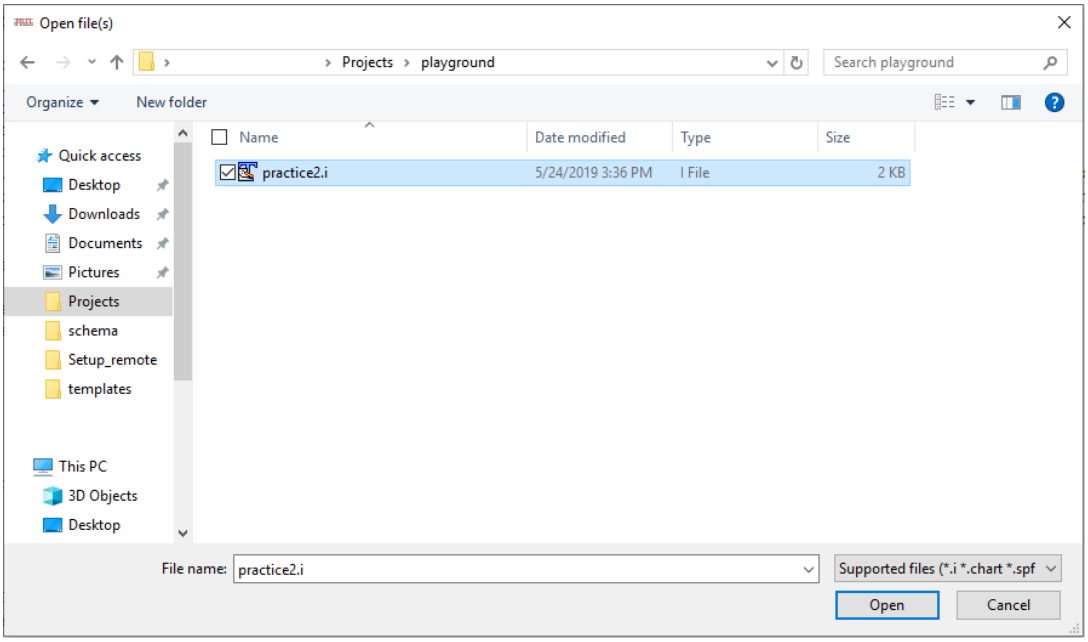

Figure 16. Example open file dialog with *.i application input.

The practice2.i file contents are listed in Appendix A. Upon opening the application input, the NEAMS Workbench will attempt to automatically select the application runtime and grammar. However, if the application input extension is ambiguous with other configured and enabled applications, the user may need to select the appropriate runtime (which automatically selects the associated grammar) via the runtime widget, as illustrated in Figure 17:

Figure 17. The NEAMS Workbench application runtime selection.

Note that once the Bison grammar is loaded, the file can detect input validation errors, visible when the "Validation" tab is selected. For instance, in Figure 18, "type" is misspelled "typ," which results in three validation errors relating specifically to that line.

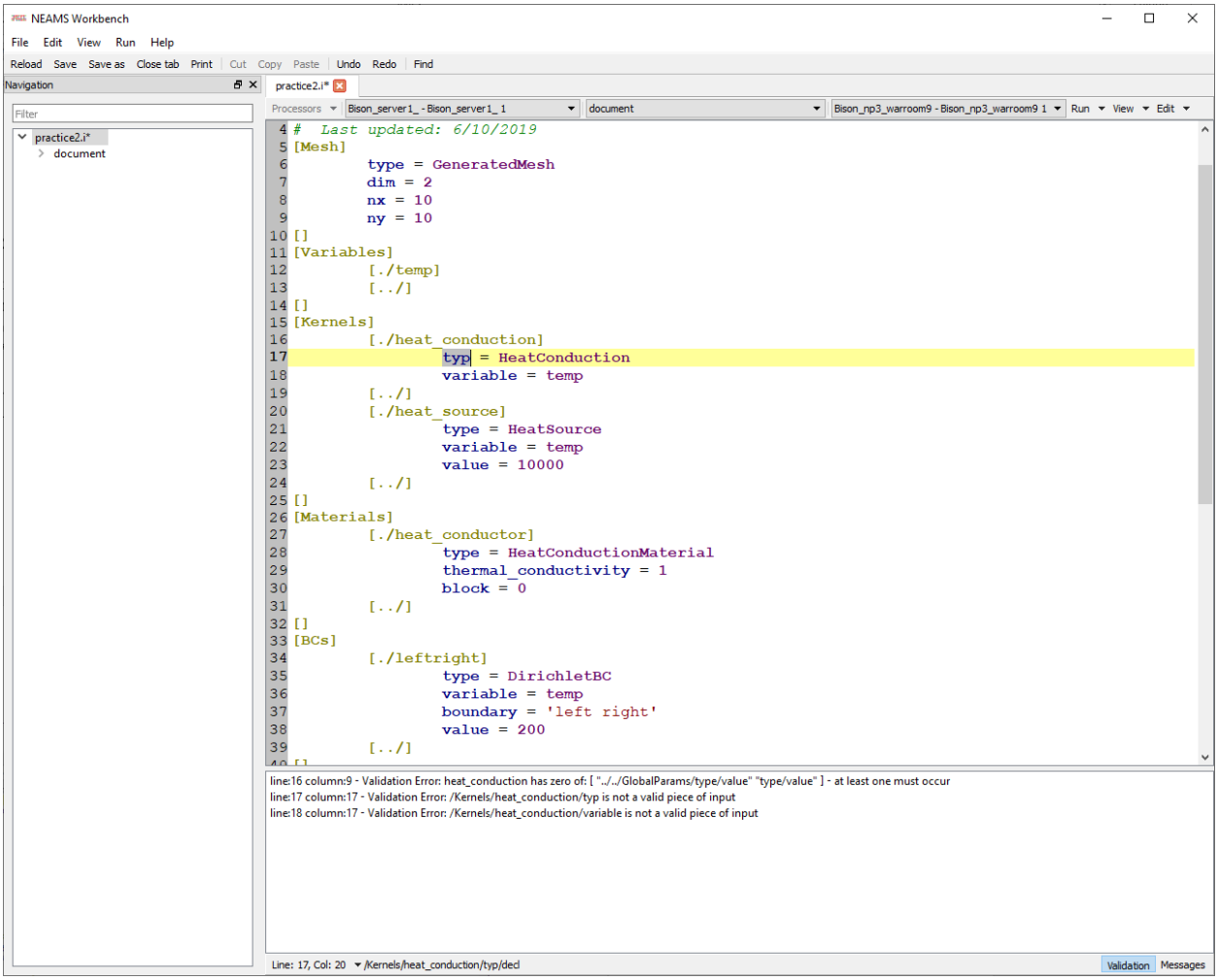

Figure 18. Visualization of the NEAMS Workbench input validation feature

Likewise, the autocomplete function can now be used, as is demonstrated in Figure 19, by pressing `CTRL + space` on the keyboard.

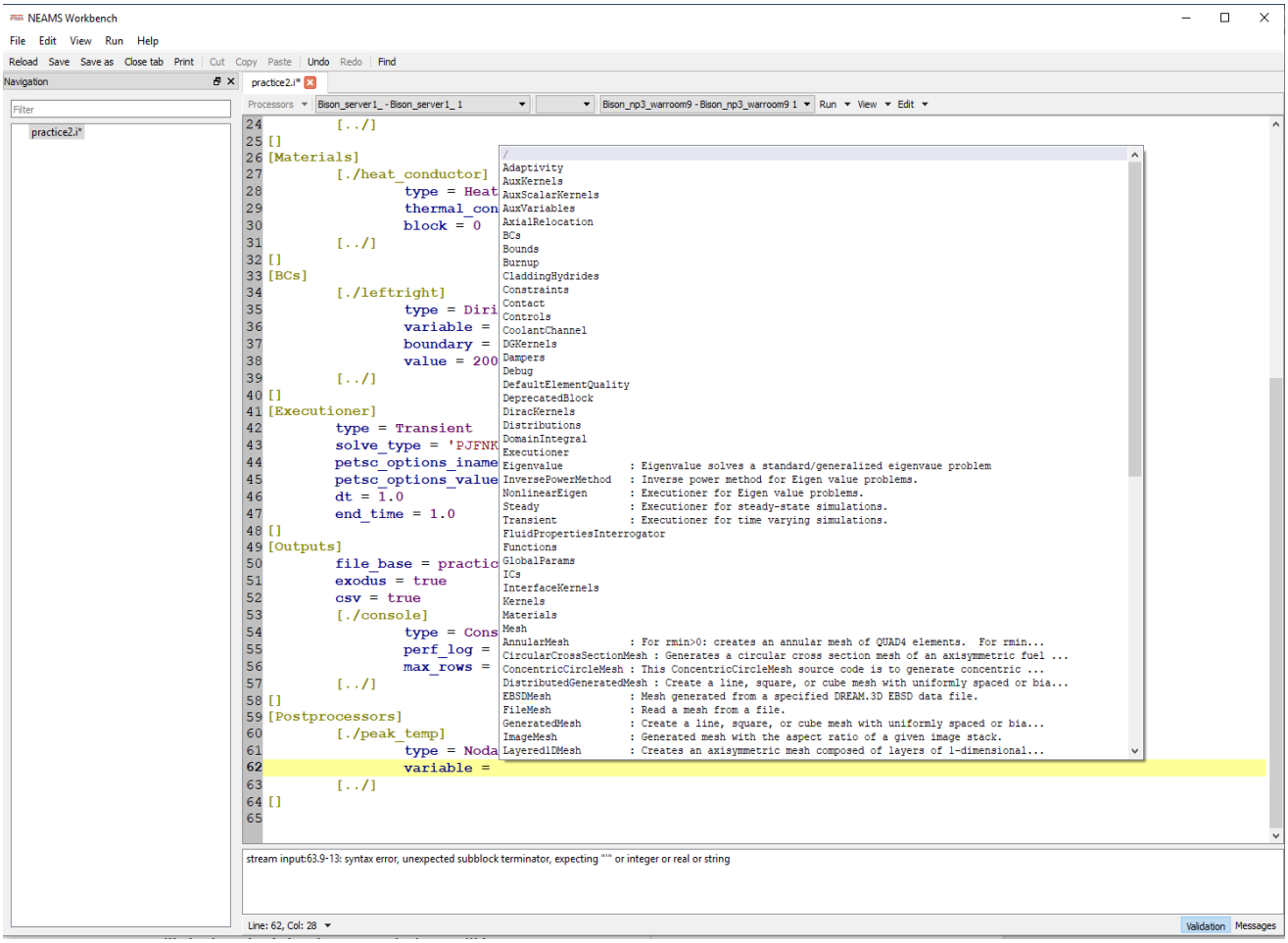

Figure 19. Demonstration of the autocomplete function available in NEAMS Workbench

Clicking the `Run` button will execute the remote application runtime, which will upload the application input to the server. In addition to the input file, any associated files matching the remote_application.setup's `upload_patterns` will also be uploaded to the server. The input will be submitted to the scheduler queue for execution. The NEAMS Workbench will status the job and print an appropriate activity message when the job starts and finishes, as demonstrated in Figure 20. Upon completion, the application's remote runtime environment will download files matching the remote_application.setup's `download_patterns`.

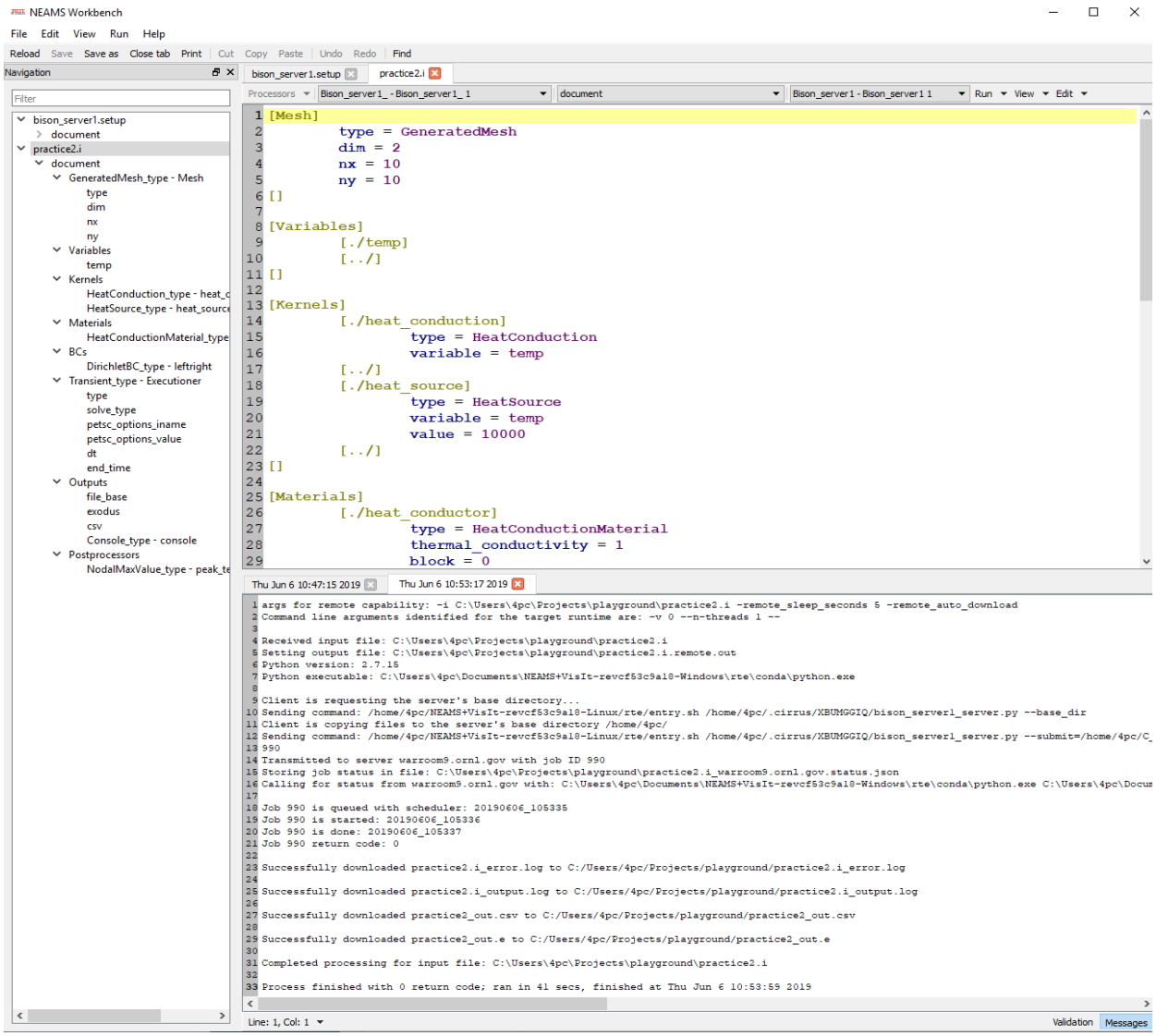

Figure 20. NEAMS Workbench remote application execution.

The files associated with the input (matching base-name as the input) are easily accessed from the `Open associated files` menu item available via right clicking on the input file in the `Navigation` panel (see Figure 21):

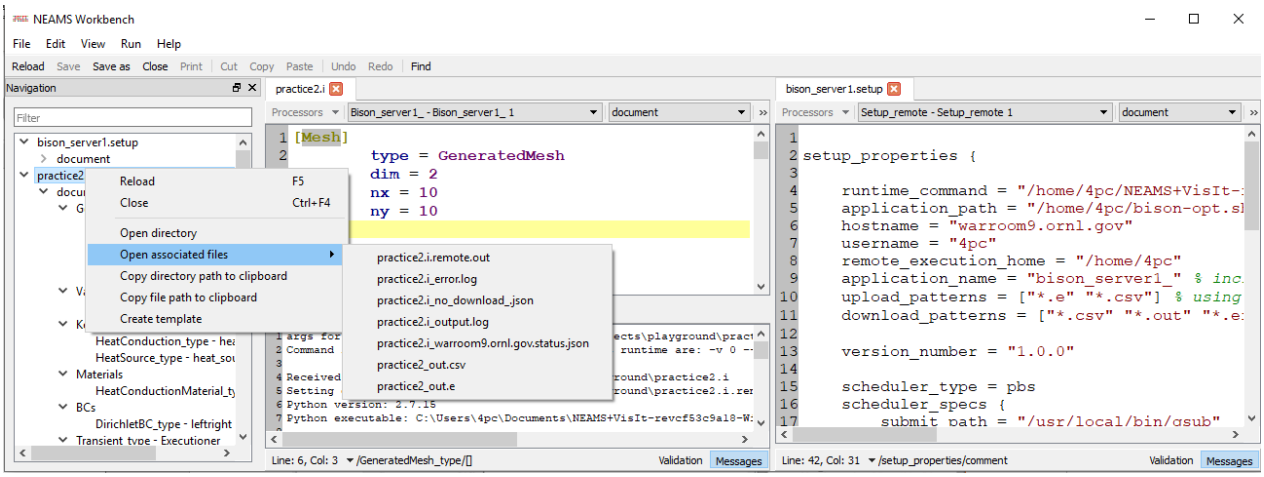

Figure 21. Opening file associated with the application input via right clicking and selecting Open associated files.

This demonstrates the remote execution of a BISON fuel performance input from a Windows 10 client onto a Linux cluster and concludes the instructions.

Quick-reference instructions:

1. Open NEAMS Workbench
2. If needed, open the help documentation
    a. Click Help> Help documentation in the top left corner
3. Click File> Configurations in the top left corner
4. In the Application configurations window, click Add… at the top left
5. Choose Setup_remote from the dropdown menu
6. Click OK
7. In the Applications configurations window, click Load Grammar at the top right
8. Click OK
9. Click File> New File in the top left corner
10. From the Files of type dropdown menu, choose Setup_remote
11. Confirm the top dropdown menus list ‘Setup_remote – Setup_remote 1’ ‘document’ and ‘Setup_remote – Setup_remote 1’
12. Click inside the text editor
13. Type CTRL+SPACE and select setup_properties – bison
14. In line 4, replace ‘/home/jdoe/workbench’ with <LOCATION-TO-SERVER-Workbench>
15. In line 5, replace ‘/home/jdoe/bison-opt.sh’ with <LOCATION-TO-SERVER-Bison>
16. In line 7, update username from ‘jdoe’ to your username
17. In line 8, update ‘jdoe’ to your username
18. In line 9, update the application name by replacing ‘remote’ with your server name
19. Click Run at the top right
20. Click Messages in the bottom right
21. Confirm line 26 in messages says “Verified server credentials are valid”
22. Confirm line 34 in messages says “Completed writing the class file at /…”
23. Confirm line 39 in messages says “Completed downloading the options file at /…”
24. Click File> Configurations in the top left corner
25. In the Application configurations window, click Add… at the top left
26. Choose the new application from the dropdown menu
27. Click OK

28. Click Load Grammar
29. Click OK
30. Click File> Open file…
31. Select an application input from files
32. Click Open
33. Select the appropriate runtime from the top dropdown menu between the document dropdown menu and the run button
34. Click Run
35. Right click the input file
36. Click Open associated files
37. Open any associated output or error message files

In summary, the user should have successfully completed the following:

1. Ensure NEAMS Workbench is installed on the client and server.
2. Ensure BISON is installed on the server.
3. Configure the client and server for passwordless connection.
4. Using the client NEAMS Workbench do the following:
    a. Enable the NEAMS Workbench's Setup Remote application.
    b. Create the application-server Setup_Remote input.
    c. Run application-server setup remote input to generate the NEAMS Workbench BISON remote RTE.
    d. Configure the BISON remote RTE,
    e. Open any BISON input and select the BISON remote RTE to activate input validation, autocompletion, syntax highlighting, and job launch capabilities.
    f. Edit input as needed.
    g. Use the BISON remote RTE to launch the Bison input from the client to the server.
    h. Wait for the job to finish.
    i. Open files associated with the input file.

REFERENCES

1. R. A. Lefebvre, B. R. Langley, Paul Miller, Marco Delchini, Mark L. Baird, Jordan P. Lefebvre, "NEAMS Workbench Status and Capabilities," ORNL/TM-2019/1314, UT-Battelle, LLC, Oak Ridge National Laboratory
2. B. T. Rearden, R. A. Lefebvre, "Objectives of the NEAMS Workbench," ANS Summer meeting, Philadelphia, PA, USA, June 17-21, (2018).
3. Robert A. LEFEBVRE, Brandon R. LANGLEY, and Jordan P. LEFEBVRE, "Workbench Analysis Sequence Processor", ORNL/TM-2017/619, UT-Battelle, LLC, Oak Ridge National Laboratory (2017).
4. R.L. Williamson, J.D. Hales, S.R. Novascone, M.R. Tonks, D.R. Gaston, C.J. Permann, D. Anders, and R.C. Martineau, "Multidimensional multiphysics simulation of nuclear fuel behavior," J. Nucl. Mater, 423 149–163 (2012).
5. Idaho National Laboratory, "Bison Workshop: Implicit, parallel, fully-coupled nuclear fuel performance analysis," presentation slides. Last Accessed June 12, 2019. https://bison.inl.gov/SiteAssets/BISON_Workshop.pdf
6. Nicolas E. Stauff, Taek K. Kim, Robert A. Lefebvre, Brandon R. Langley, Bradley T. Rearden, "Integration of the Argonne Reactor Computation codes into the NEAMS Workbench," ANS Summer meeting, Philadelphia, PA, USA, June 17-21, (2018).
7. D. Gaston, C. Newman, G. Hansen, And D. Lebrun - Grandie´, "MOOSE: A parallel computational framework for coupled systems of nonlinear equations, Nucl. Eng. Des., 239, 1768–1778 (2009).
8. Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.7 User's Manual.
9. Rearden, Bradley T, Dunn, Michael E, Wiarda, Dorothea, Celik, Cihangir, Bekar, Kursat B, Williams, Mark L, Peplow, Douglas E., Perfetti, Christopher M, Gauld, Ian C, Wieselquist, William A, Lefebvre, Jordan P, Lefebvre, Robert A, Havluj, Frantisek, Skutnik, Steven, and Dugan, Kevin. *OVERVIEW OF SCALE 6.2*. United States: N. p., 2013. Web.
10. Aysenur Toptan, Robert K. Salko, Maria N. Avramova, Kevin Clarno, David J. Kropaczek, A new fuel modeling capability, CTFFuel, with a case study on the fuel thermal conductivity degradation, Nuclear Engineering and Design, Volume 341, 2019, Pages 248-258

APPENDIX A. BISON WORKSHOP PRACTICE PROBLEM

```
#  This input file was obtained from the Idaho National Laboratory Bison training workshop,
#  publicly available at https://bison.inl.gov/SiteAssets/Bison_Workshop.pdf#page=44.
#  It has been modified to produce a .csv output file (see line 58), for training purposes.
#  Last updated: 6/10/2019
[Mesh]
        type = GeneratedMesh
        dim = 2
        nx = 10
        ny = 10
[]
[Variables]
        [./temp]
        [../]
[]
[Kernels]
        [./heat_conduction]
                type = HeatConduction
                variable = temp
        [../]
        [./heat_source]
                type = HeatSource
                variable = temp
                value = 10000
        [../]
[]
[Materials]
        [./heat_conductor]
                type = HeatConductionMaterial
                thermal_conductivity = 1
                block = 0
        [../]
[]
[BCs]
        [./leftright]
                type = DirichletBC
                variable = temp
                boundary = 'left right'
                value = 200
        [../]
[]
[Executioner]
        type = Transient
        solve_type = 'PJFNK'
        petsc_options_iname = '-pc_type -pc_factor_mat_solver_package'
        petsc_options_value = 'lu superlu_dist'
        dt = 1.0
        end_time = 1.0
[]
[Outputs]
        file_base = practice2_out
        exodus = true
        csv = true
        [./console]
                type = Console
                perf_log = true
                max_rows = 25
        [../]
[]
[Postprocessors]
        [./peak_temp]
                type = NodalMaxValue
                variable = temp
        [../]
[]
```