

Integration of the Nek5000 Computational Fluid Dynamics Code with the NEAMS Workbench



**Approved for public release.
Distribution is unlimited.**

Marc-Olivier G. Delchini
Robert A. Lefebvre
William D. Pointer

September 30, 2019

DOCUMENT AVAILABILITY

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

Website: <http://www.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: <http://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: report@osti.gov
Website: <http://www.osti.gov/contact.html>

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.

Reactor and Nuclear Systems Division

**Integration of the Nek5000 Computational Fluid Dynamics Code with the NEAMS
Workbench**

Author(s)
Marc-Olivier G. Delchini
Robert A. Lefebvre
William D. Pointer

Date Published: September 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	v
LIST OF TABLES	vii
1. INTRODUCTION	1
2. POST-PROCESSING CAPABILITIES	1
3. LINE-PLOT FOR VELOCITY-BASED VARIABLES	3
4. REGULAR EXPRESSIONS	5
5. CONCLUSIONS AND FUTURE WORK	7

LIST OF FIGURES

1	Post-processing (PPS) block in the Nek4Nuc input file.	2
2	Template file (left) called by HALITE to generate FORTRAN 77 code (right) in the *.usr file.	3
3	Velocity magnitude in a 2D rectangle after 100 time steps.	4
4	Vorticity magnitude in a 2D rectangle after 100 time steps.	4
5	Input file for the <i>2d-rectangle-line-plot</i> example illustrating the syntax for the line plot sub-blocks.	5
6	Line plot of the time-average velocity extracted after 100 time steps.	6
7	Velocity and temperature blocks with regular expressions.	6
8	FORTRAN 77 code generated by HALITE in the *.usr file to implement temperature-dependent density and viscosity coefficients.	7
9	Velocity (top) and temperature (bottom) fields for the low-Mach test using the method of manufactured solution.	8

LIST OF TABLES

1	Input parameters for the PPS block when using Nek4Nuc input file	2
---	--	---

1. INTRODUCTION

The mission of the US Department of Energy’s 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 using computing architectures ranging from laptops to leadership-class facilities. NEAMS is organized into three product lines: the Fuels Product Line (FPL), the Reactors Product Line (RPL), and the Integration Product Line (IPL). The FPL and RPL focus on the development of high-fidelity tools such as the MOOSE-based applications BISON and MARMOT [3, 9] at Idaho National Laboratory, and the PROTEUS neutronics code and the NEK5000 computational fluid dynamics code from the Simulation-based High-efficiency Advanced Reactor Prototyping (SHARP) framework [10]. The NEAMS IPL will enable users to apply high-fidelity simulations to inform lower order models for the design, analysis, and licensing of advanced nuclear systems.

The NEAMS Workbench [6, 7, 5] initiative is part of the NEAMS IPL and is designed to facilitate the transition from conventional tools to high-fidelity tools by providing a common user interface for model creation, review, execution, and visualization for integrated codes. It was first used to support MOOSE applications in June 2016 and has since been broadly applied to other high-fidelity tools.

The overall objective of this project is to facilitate the pre- and post-processing steps when running a model with Nek5000 by leveraging the NEAMS Workbench user-friendly interface. The long-term objective is to provide the user with the same workflow that is proposed in most of the commercial and open-source CFD packages. Previous work focused on developing Python modules and scripts to integrate a typical Nek5000 workflow within the NEAMS Workbench [1]. Recently, a more advanced tool, referred to as *Nek4Nuc*, was developed on the top of the current Nek5000 workflow by leveraging the Hierarchical Input Template Expansion Engine (HALITE) [4] to generate some of the required files directly from the input file. One of the main achievements of this new tool is its ability to generate FORTRAN 77 code to implement boundary conditions, initial conditions, and flow parameters—typically cumbersome tasks for a user.

This report presents the most recent updates on the effort to integrate the Argonne National Laboratory (ANL) tool Nek5000 [2] thermal hydraulic analysis code into the NEAMS Workbench. The most recent efforts have focused on adding post-processing capabilities (Section 2.), implementing line plot capabilities (Section 3.), and supporting regular expressions for initial and boundary conditions, as well as source terms (Section 4.). Each of the following sections is illustrated with an example.

The reader is referred to ORNL/TM-2016/215 [1] for more details regarding integration of Nek5000 into the NEAMS Workbench.

2. POST-PROCESSING CAPABILITIES

Nek5000 already supports post-processing capabilities such as time-average data and history file to collect field values at given locations and for each time step. These capabilities require a user to modify the **.usr* file written in FORTRAN 77 that serves as the interface between the user and the Nek5000 program.

These post-processing capabilities are now available to a user through the Nek4Nuc input file, an extension of the Nek5000 input. The syntax is simple and consists of a single block denoted by *[PPS]* in the input file, as shown in Fig. 1. Setting *time_average* to *yes* collects time-average data for all fields by adding the

```

45 [PPS]
46 time_average = yes
47 history_points {
48     file = "txt"
49     number_points = 3
50     step_begin = 1
51     step_end = 10
52 }
53 vorticity {
54     output = 10
55 }
56 # insert PPS components above this line

```

Figure 1. Post-processing (PPS) block in the Nek4Nuc input file.

line *call avg_all()* in the *.usr file. Input parameters for setting the history points is achieved with a sub-block that consists of the file format (*file*) containing the coordinates of the points (*txt*, *csv* and *his* formats), the number of points (*number_points*), and specification of when to start/stop collecting data (*step_begin* and *step_end*). The vorticity is enabled with the sub-block *vorticity* that takes a single variable *output* to control the frequency at which the vorticity data are output. Binary output files storing the vorticity are readable by Workbench via the integrated VisIt visualization tool. A brief description of the keywords supported by the PPS block is provided in Table 1.

Table 1. Input parameters for the PPS block when using Nek4Nuc input file

Input argument	Description	Type
time_average	Collect time-average data for all fields	Single value
history_points	Set history points for all fields	Sub-block
vorticity	Compute and dump vorticity	Sub-block

The Workbench’s Nek4Nuc runtime environment uses the open source Workbench Analysis Sequence Processor (WASP) Standard Object Notation (SON) interpreter to parse the document and the Hierarchical Input Validation Engine (HIVE) to verify that the document is correct according to the Nek4Nuc input schematic [4]. Nek4Nuc subsequently writes the data in JSON format for use by HALITE and invokes HALITE with the Nek5000 templates. The FORTRAN 77 code is then automatically generated by HALITE by substituting the appropriate data values into the Nek5000 templates, as illustrated in Fig. 2. The HALITE syntax is denoted by the hash-tag symbol # followed by keywords surrounded with <>, which refers to variables from the input file converted to JSON format. Nek4Nuc compiles the FORTRAN 77 code generated by HALITE by using the Nek5000-provided compiler script, *makenek*. The PPS capabilities detailed above are illustrated in an example called *2d-rectangle-pps* that models an inlet/outlet flow in a two-dimensional rectangle. Numerical results of the instantaneous velocity field and the vorticity are shown in Figures 3 and 4, respectively. These figures were obtained using Workbench-integrated VisIt.

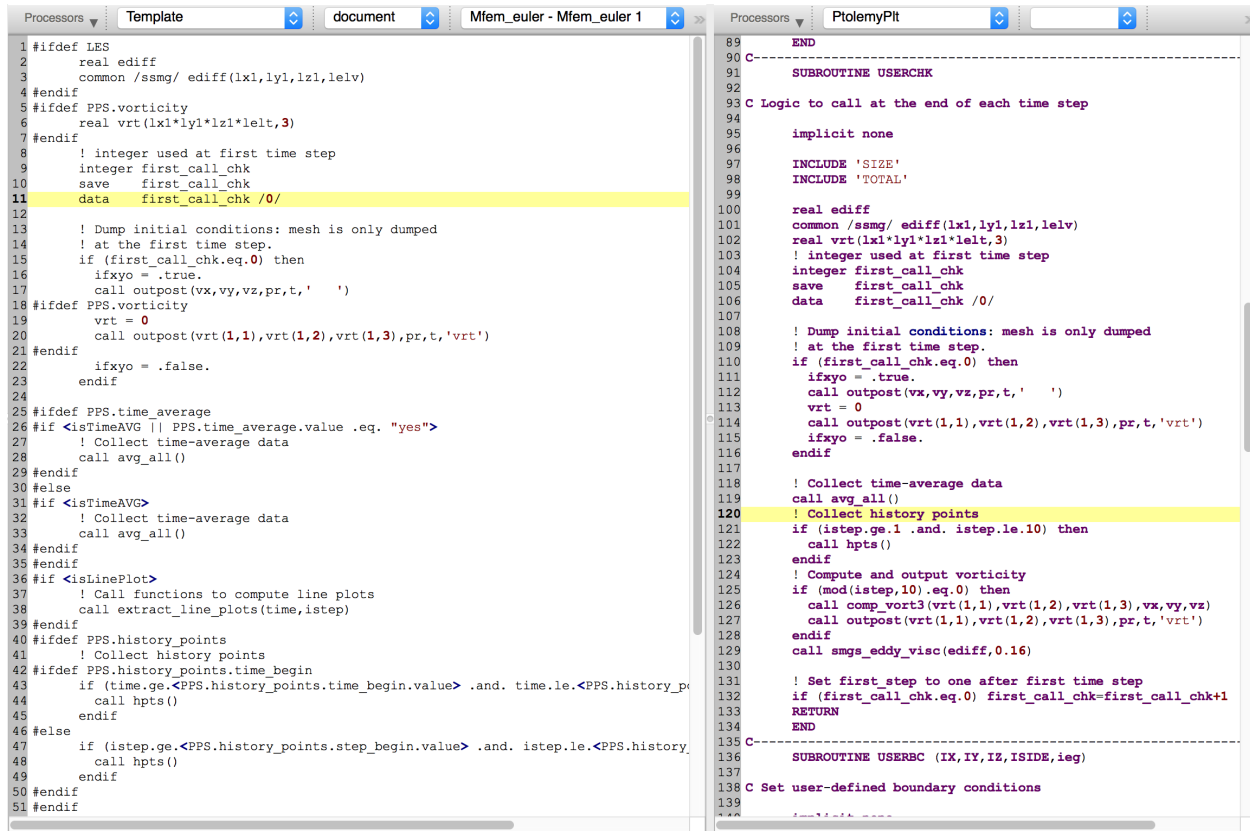


Figure 2. Template file (left) called by HALITE to generate FORTRAN 77 code (right) in the *.usr file.

3. LINE-PLOT FOR VELOCITY-BASED VARIABLES

The line plot capability allows a user to extract velocity-based variables along a line within the geometry. The data are saved in a text file and can be plotted using Workbench's post-processing capabilities. The line plot capabilities are currently only available for the following velocity-based variables:

- Instantaneous velocity field (*type_var* = itt)
- Time-average velocity field (*type_var* = avg)
- Kinetic energy (*type_var* = knrg)
- Turbulent kinetic energy (*type_var* = tknrg)
- Reynolds stresses (*type_var* = rm2)

Line plot parameters are specified in a sub-block inside the *VELOCITY* block. A user can define multiple line plots within the same input file (one sub-block per line plot). Each line plot is defined by its two end points—which are referred to as *pt1* and *pt2*—and a number of samples denoted by *nb_points*. The data to be extracted are specified with the variable *type_var*. Data are extracted at every *output* time step and are written in the text file *name*. The line plot syntax is illustrated in the example *2d-rectangle-line-plot*, and

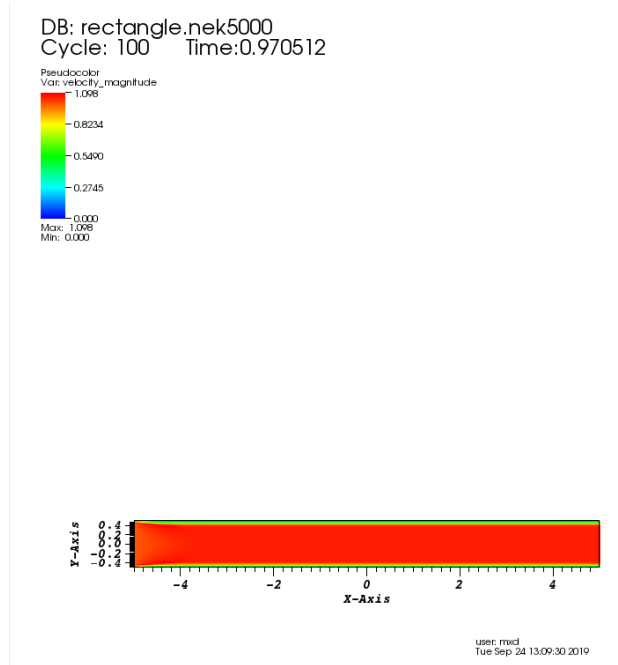


Figure 3. Velocity magnitude in a 2D rectangle after 100 time steps.

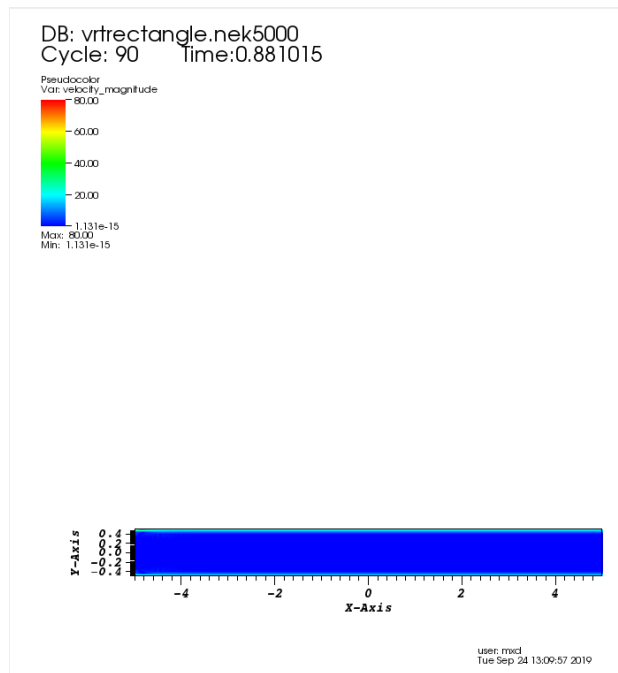


Figure 4. Vorticity magnitude in a 2D rectangle after 100 time steps.

the associated input file is shown in Fig. 5. Four line plots are set at different locations of the mesh to collect the velocity-based variables. Note that each line plot sub-block requires up to 350 lines of FORTRAN 77 code to extract, compute, and write the data. Implementation of the logic in the *.usr file is

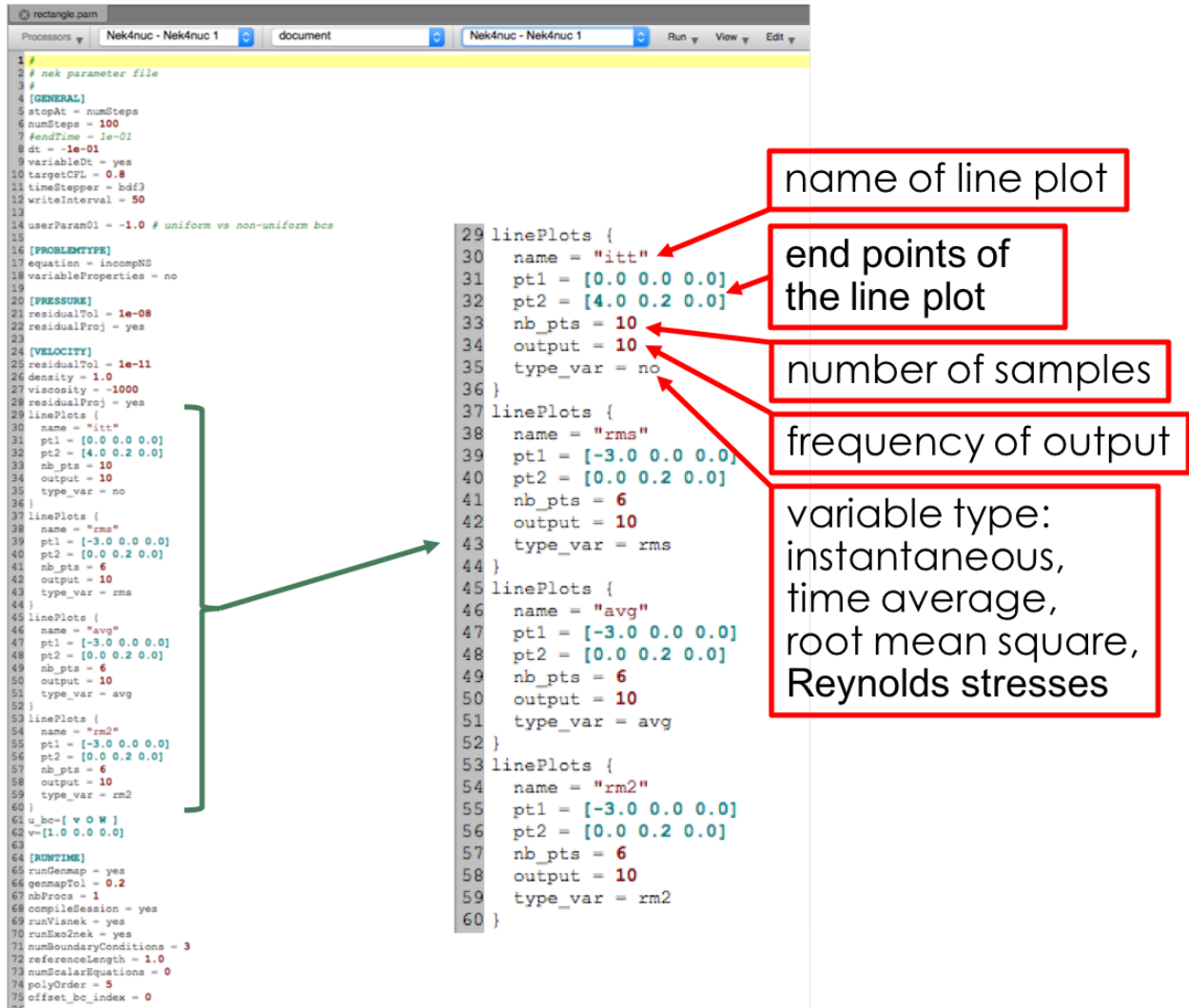


Figure 5. Input file for the *2d-rectangle-line-plot* example illustrating the syntax for the line plot sub-blocks.

handled by HALITE and does not require any programming from the user. Once the data is extracted and stored in a text file, the values can be plotted using the post-processing capabilities available in Workbench, as shown in Fig. 6.

4. REGULAR EXPRESSIONS

Regular expressions are now supported by the Nek4Nuc input file, allowing the user to set non-uniform boundary conditions, initial conditions, volumetric source terms, forces, and flow properties. This new capability was tested on an example referred to as *low_mach_test*, which is part of the Nek5000 example suite [8]. In [8], the method of manufactured solution (MMS) is used to assess the convergence order of the spectral element method. The same Nek5000 session was generated from the Nek4Nuc input file shown in

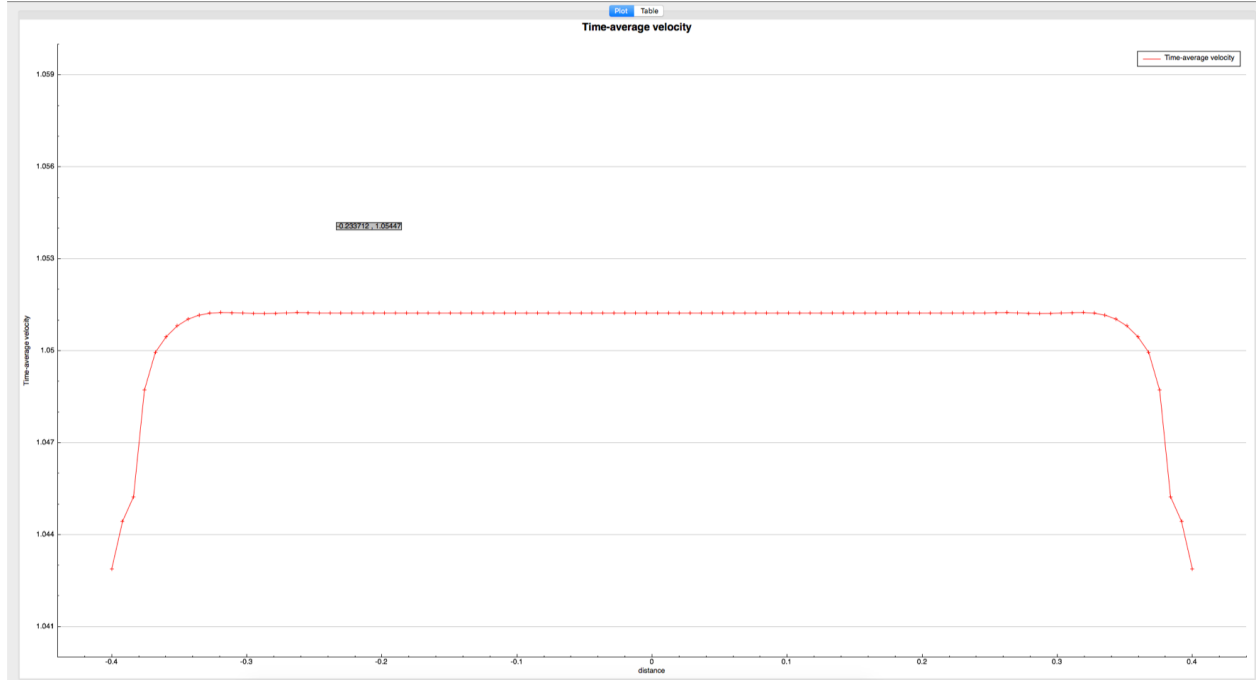


Figure 6. Line plot of the time-average velocity extracted after 100 time steps.

Fig. 7. The source terms derived from the MMS are functions of the local coordinates X and Y, and the

```
[VELOCITY]
residualTol = 1e-08
u_bc=[ v ]
u_ic = [ "0.5*( 3.0 + tanh(X/0.2) )" 0.0 0.0 ]
v = [ "0.5*( 3.0 + tanh(X/0.2) )" 0.0 0.0 ]
viscosity = "1.0"
density = "1.0 /TEMP"

[TEMPERATURE]
residualTol = 1e-12
ps_bc=[ t ]
ps_ic = "0.5*( 3.0 + tanh(X/0.2) )"
t = "0.5*( 3.0 + tanh(X/0.2) )"
qv = "1.0/cosh(X/0.2)*1.0/cosh(X/0.2)/0.2*(0.5+tanh(X/0.2)/0.2)"
conductivity = "1.0"
rhoCp = "1.0/TEMP"
```

Figure 7. Velocity and temperature blocks with regular expressions.

flow properties (i.e., density and viscosity) are functions of the temperature, which is denoted by TEMP. Regular expressions are also used to set the initial conditions (see u_{ic} and ps_{ic} variables) and the boundary conditions (see u_{bc} and ps_{bc} variables) for the velocity field and temperature field, respectively. A sample of the *usr file generated by HALITE using the regular expressions provided by the

user in the Nek4Nuc input file is shown in Fig. 8. Numerical results of the velocity and temperature fields

```

C-----
      SUBROUTINE USERVP (IX,IY,IZ,ieg)
C Set user variable properties

      implicit none

      integer ix,iy,iz,ieg

      INCLUDE 'SIZE'
      INCLUDE 'TOTAL'
      INCLUDE 'NEKUSE'

      ! velocity mesh
      real utransScalars(ldimt)
      real udiffScalars(ldimt)

      if (ifield.eq.1) then ! velocity field
         utrans=1.0 /TEMP
         udiff=1.0
      elseif (ifield.eq.2) then ! temperature field
         utrans=1.0/TEMP
         udiff=1.0
      else
         write(6,*) 'ifield index is larger than the number of '
         write(6,*) 'fields set to 2 '
         call exitt()
      endif

      RETURN
      END
C-----

```

Figure 8. FORTRAN 77 code generated by HALITE in the *.usr file to implement temperature-dependent density and viscosity coefficients.

are presented in Fig. 9. The manufactured solution corresponds to a steady-state solution that should be preserved for each new time step, assuming that the numerical method is correct.

5. CONCLUSIONS AND FUTURE WORK

This report summarizes the most recent additions to the Nek4Nuc capability that drives Nek5000 within the NEAMS Workbench. Post-processing capabilities can now be set by a user through a Nek4Nuc input file using a simple syntax. The line plot capability offers easy access to the numerical solution for analysis of relevant turbulent flow parameters. Extracted data can be plotted using the NEAMS Workbench post-processing capabilities, avoiding the use of a third-party package while streamlining the workflow and the production of relevant data for analysis. Each new capability is illustrated with an example, all of which are available with the latest Workbench release and can serve as templates or tutorials.

In this report, it was also demonstrated that the combination of Nek4Nuc and HALITE can be used to generate complex FORTRAN 77 codes to be compiled by the Nek5000 tool. The ability to generate

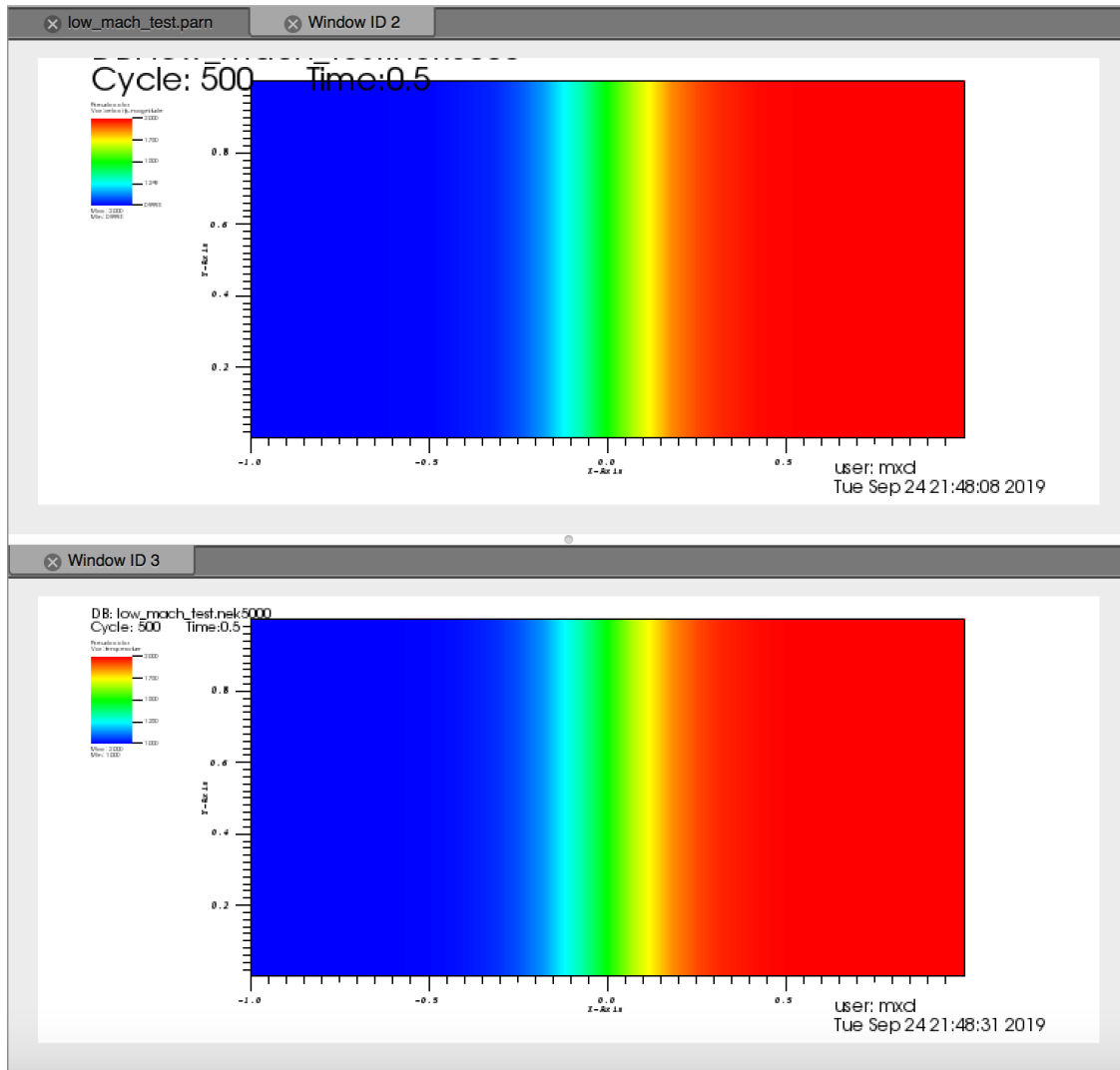


Figure 9. Velocity (top) and temperature (bottom) fields for the low-Mach test using the method of manufactured solution.

FORTRAN 77 code in an automated manner will provide additional quality assurance to the analyst's work. Since Nek5000 must be compiled before run-time, the Nek4Nuc capability of consistently generating the *.usr file and compiling the Nek5000 executable will improve Nek5000 performance in an NQA-1 environment.

Future work will include support for the latest Nek5000 versions through updates to the input schematic and addition of new post-processing capabilities while supporting Nek5000 end-users collaborating with ORNL.

References

- [1] M. Delchini, B. R. Langley, R. A. Lefebvre, W. D. Pointer, and B. T. Rearden. Integration of the Nek5000 Computational Fluid Dynamics Code with the NEAMS Workbench. Technical Report ORNL/TM-2016/215, Oak Ridge National Laboratory, 2018.
- [2] P. Fisher, J. W. Lottes, and S. G. Kerkermeier. Nek5000 web page, 2008.
- [3] D Gaston, C Newman, G Hansen, and D Lebrun-Grandie. Moose: A parallel computational framework for coupled systems of nonlinear equations. *Nucl. Eng. Design*, 239:1768–1778, 2009.
- [4] R. A. Lefebvre, B. R. Langley, and J. P. Lefebvre. Workbench Analysis Sequence Processor. Technical Report ORNL/TM-2017/698, Oak Ridge National Laboratory, 2017.
- [5] R. A. Lefebvre, B. R. Langley, P. Miller, M. Delchini, M. L. Baird, and J. P. Lefebvre. NEAMS Workbench Status and Capabilities. Technical Report ORNL/TM-2019/1314, Oak Ridge National Laboratory, 2019.
- [6] R. A. Lefebvre, B. R. Langley, and A. B. Thompson. Report on NEAMS Workbench Support for MOOSE Applications. Technical Report M3MS-16OR0401086, Oak Ridge National Laboratory, 2016.
- [7] B. R. Rearden, R. A. Lefebvre, and Langley B. R. Thompson, A. B. Introduction to the Nuclear Energy Advanced Modeling and Simulation Workbench. In *M&C - 2017 International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*, Jeju, Korea, 2017.
- [8] A. G. Tomboulides and S. A. Orzag. A Quasi-Two-Dimensional Benchmark Problem for Low Mach Number Compressible Codes. *Journal of Computational Physics*, 146:691–706, 1998.
- [9] 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. *Nucl. Eng. Design*, 423:149–163, 2012.
- [10] Y. Q. Yu, E. R. Shemon, J. W. Thomas, V. S. Mahadevan, R. O. Rahaman, and J. Solberg. SHARP User Manual. Technical Report ANL-NE-16/6, Argonne National Laboratory, 2016.