

Assessment of SFR Wire Wrap Simulation Uncertainties



Approved for public release.
Distribution is unlimited.

Marc-Olivier Delchini

Emilian Popov

David Pointer

Laura Swiler

September 2016

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/SCITECH/](http://WWW.OSTI.GOV/SCITECH/)

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://WWW.NTIS.GOV/HELP/ORDERMETHODS.ASPX](http://WWW.NTIS.GOV/HELP/ORDERMETHODS.ASPX)

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 [HTTP://WWW.OSTI.GOV/CONTACT.HTML](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

Assessment of SFR Wire Wrap Simulation Uncertainties

Marc-Olivier Delchini

Emilian Popov

David Pointer

Laura Swiler

Date Published: September 30, 2016

Distribution:

Justin Thomas, ANL

Tanju Sofu, ANL

Chris Stanek, LANL

Dan Funk, DOE, NE-41

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	iv
LIST OF TABLES	vi
1. INTRODUCTION	1
2. SOFTWARE USED: NEK5000, STARCCM+ AND DAKOTA	1
3. COUPLING INTERFACE DAKOTA TO NEK5000 AND STARCCM+	2
3.1 FORK SIMULATION INTERFACE	2
3.2 COUPLING TO Nek5000	3
3.3 COUPLING TO STARCCM+	4
4. THORS MESH GENERATION	4
4.1 THORS SINGLE WIRE WRAP PIN	5
4.2 THORS 7 PIN BUNDLE	9
5. DAKOTA Nek5000 UQ DEMONSTRATION RUNS	9
5.1 THE MODEL OF INTEREST	10
5.2 THE SETUP AND METHOD	13
5.3 NUMERICAL RESULTS	15
6. STARCCM+ SINGLE PIN THORS CALCULATIONS	24
7. NEK5000 WIRE WRAPPED BUNDLE CALCULATIONS	26
7.1 SINGLE-PIN THORS SIMULATION	27
7.2 7-PIN BUNDLE SIMULATIONS	29
8. CONCLUSIONS	34
9. BIBLIOGRAPHY	35

LIST OF FIGURES

Figure 1. Dakota-Nek5000 workflow overview.	3
Figure 2. Original wire wrap pins arranged in triangular lattice of SFR (left), and extracted fluid domain used for meshing (right).....	5
Figure 3. THORS single pin mesh: overall view (top), details (bottom left), and central pin wire (bottom right).	6
Figure 4. Single pin mesh quality metrics: graphical representation of element distribution per quality parameter (left), mesh colored by Jacobian showing the high distortion elements at the wire pin connection points (right).	8
Figure 5 Cross section (left) and 3-D (right) views of the THORS 7-pin-bundle mesh.....	9
Figure 6. 3-D pipe mesh with 84 radial elements.	11
Figure 7. 3-D pipe mesh with 84 radial elements.	11
Figure 8. 3-D pipe mesh with 184 radial elements.	12
Figure 9. 3-D pipe mesh with 832 radial elements.	12
Figure 10. Numerical (Nek5000), experimental, and interpolated data as a function of the wall distance after postprocessing for mesh with 184 radial elements and $lx1=5$	14
Figure 11 Numerical (Nek5000), experimental, and interpolated data as a function of the wall distance after postprocessing for mesh with 184 radial elements and $lx1=8$	15
Figure 12 Main effects plot for L2 error norm	17
Figure 13. Interval plot for means of L2 error norm for P101.....	18
Figure 14. Interval plot for means of L2 error norm for $Lx1$	19
Figure 15. Interval plot for means of L2 error norm for P103.....	20
Figure 16. Interval plot for means of L2 error norm for mesh density.....	21
Figure 17. Interval plot for means of L2 error norm for mesh type.....	22
Figure 18. Interaction plot between mesh type and mesh density.	23
Figure 19. Summary plot of confidence intervals for the mesh type parameter.....	24

Figure 20. Cross section view of a turbulent flow in THORS single pin.	25
Figure 21 Numerical (STARCCM+), correlation (Todreas and Novendstern) and experimental pressure drop as a function of Reynolds number for a THORS single pin.	26
Figure 22. Velocities at six axial positions of the single pin mesh, symmetry boundaries on the periodic faces.	28
Figure 23. Single pin wire wrap simulation, wall boundaries on the periodic faces	29
Figure 24 Time average velocity magnitude at $z = 0$ (bottom) simulated with $lx1=4$	30
Figure 25 Time average velocity magnitude at $z = 25$ (middle) simulated with $lx1=4$	31
Figure 26 Time average velocity magnitude at $z = 0$ (bottom) simulated with $lx1=4$	32
Figure 27 Time average velocity magnitude at $z = 0$ (bottom) simulated with $lx1=6$	33

LIST OF TABLES

Table 1. MOAB single pin mesh statistics.....	6
Table 2. Single-pin mesh quality metrics: aspect ratio, skew, taper, and volume ratios	7
Table 3. Portion of the complete parameter study over five input parameters	16
Table 4. Summary of run parameters for THORS single-pin mesh	27

1. INTRODUCTION

Predictive modeling and simulation of nuclear reactor performance and fuel are challenging due to the large number of coupled physical phenomena that must be addressed. Models that will be used for design or operational decisions must be analyzed for uncertainty to ascertain impacts to safety or performance. Rigorous, structured uncertainty analyses are performed by characterizing the model's input uncertainties and then propagating the uncertainties through the model to estimate output uncertainty.

This project is part of the ongoing effort to assess modeling uncertainty in Nek5000 simulations of flow configurations relevant to the advanced reactor applications of the Nuclear Energy Advanced Modeling and Simulation (NEAMS) program. Three geometries are under investigation in these preliminary assessments: a 3-D pipe, a 3-D 7-pin bundle, and a single pin from the Thermal-Hydraulic Out-of-Reactor Safety (THORS) facility.

Initial efforts have focused on gaining an understanding of Nek5000 modeling options and integrating Nek5000 with Dakota. These tasks are being accomplished by demonstrating the use of Dakota to assess parametric uncertainties in a simple pipe flow problem. This problem is used to optimize performance of the uncertainty quantification strategy and to estimate computational requirements for assessments of complex geometries. First, Nek5000 will be calibrated using results from the first study. Then the flow in a 7-pin bundle geometry will be simulated with Nek5000, and the THORS geometry will be simulated using STARCCM+ and Nek5000. The objective for both geometries is to perform an uncertainty quantification, along with a sensitivity analysis of the pressure drop. All scripts and input files used in this project are available at the GitLab website.

Section 2 briefly describes the software tools used in this study and provides appropriate references. Section 3 presents the coupling interface between Dakota and a computational fluid dynamic (CFD) code (Nek5000 or STARCCM+), with details on the workflow, the scripts used for setting up the run, and the scripts used for post-processing the output files. In Section 4, the meshing methods used to generate the THORS and 7-pin bundle meshes are explained. Sections 5, 6 and 7 present numerical results for the 3-D pipe, the single pin THORS mesh, and the 7-pin bundle mesh, respectively.

2. SOFTWARE USED: NEK5000, STARCCM+ AND DAKOTA

The CFD code Nek5000 (Fisher, Lottes and Kerkermeier 2008), which was developed at Argonne National Laboratory (ANL), is used to simulate flow behavior in the 3-D pipe. The incompressible Navier-Stokes equations are solved on a computational domain discretized by a spectral element method (SEM). Several temporal discretization methods are available for use in transient simulations. A second-order temporal discretization is used for the results presented in this report. The Nek5000 code is designed to run on massive supercomputers and has been used and tested worldwide for CFD simulations. A Nek5000 session includes multiple files storing the discretization properties (SIZE), parameters related to the solver and the physical properties (*.rea file), a user file that implements the boundary and initial conditions, and source terms.

STAR-CCM+ (CD-adapco 2000) is a commercial computer-aided engineering package developed by CD-adapco that can solve problems involving multiphysics, flow, heat transfer, stress, and complex geometries. STAR-CCM+ has a well-earned reputation for producing high-quality results in a single code with minimum user effort, and it works in various engineering fields. It is a finite-volume-based code that

offers a variety of unsteady and steady-state solvers, turbulent models, and other built-in features. Like Nek5000, it runs on supercomputers and can be used for intensive computer simulations.

The uncertainty quantification (UQ) analysis is performed using the Dakota package (Adams, et al. 2014), which includes methods for parameter studies, optimization, sensitivity analysis, and UQ. Dakota interfaces with external packages and codes such as Nek5000 through a script-driven interface called *fork*. The interface provides maximum flexibility to the users for pre- and post-processing tasks before and after each function evaluation. For this study, a given set of input parameters of known uncertainty is supplied to Nek5000 through the Dakota interface, and then the code corresponding to a function evaluation is run. Outputs of interest—or *response functions* in UQ terms—are returned to Dakota to determine the outputs' uncertainty. More details regarding input and output parameters are provided in (Adams, et al. 2014).

3. COUPLING INTERFACE DAKOTA TO NEK5000 AND STARCCM+

3.1 FORK SIMULATION INTERFACE

The UQ analysis is performed using the Dakota package (Adams, et al. 2014), which includes methods for parameter studies, optimization, sensitivity analysis, and UQ. Dakota interfaces with external packages/codes such as Nek5000 and STAR-CCM+ using either fork, direct function invocations, or computational grid invocations. Only the fork interface is of interest for this project. The fork interface requires a set of files/scripts to run; these are briefly described below:

- **Dakota input file** - supplies input parameters of known uncertainty, the number of response functions and other information related to the interface capabilities
- **Driver script** - sets up each job or function evaluation; called by Dakota for each new job, and takes two files as arguments:
 - *parameters_file* (default name) - stores values of Dakota input parameters for the given job.
 - *results_file* - used to return output values of interest to Dakota which will be used later to determine the uncertainty of the outputs
- **Template input file** - drives with Dakota through the fork interface (The template is modified with the *dprepro* tool that generates a new input file for each job by substituting the Dakota input parameters into the template input file.)
- **One or more script(s)** - post-process output file(s) and extract the output of interests; also known as *response function* (Each output value is written in the file *results_file*.)

In the case of the fork interface, Dakota treats the simulation as a black box through a driver script that (1) sets up each job based on the Dakota input parameters, (2) runs the jobs, (3) post processes the outputs of interest, and (4) returns the output values to Dakota. More details regarding the different fork options can be found in the Dakota user manual. Figure 1 depicts the information flow between Dakota and a simulation code such as Nek5000:

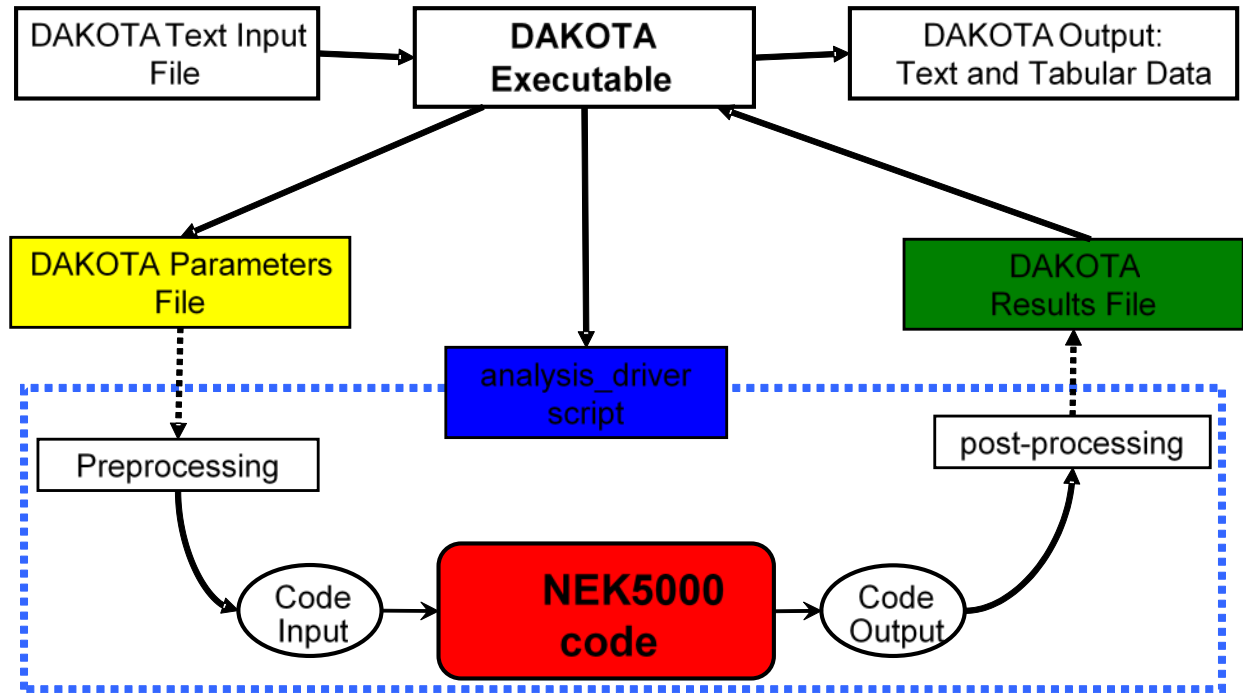


Figure 1. Dakota-Nek5000 workflow overview.

(Adams, et al. 2014).

3.2 COUPLING TO Nek5000

The coupling between Dakota and Nek5000 follows the workflow described above and in (Delchini, Popov and Pointer 2016). All scripts are available on the website and are designed to run either on Romulus or Panacea Oak Ridge National Laboratory (ORNL) clusters. Each script in the workflow is described below:

- **The driver script** consists of three main parts.
 - A local directory named after the job ID is created and used to set up and run the simulation. The input files are generated from the template input files.
 - Then, a queue submission file named *pbs_submission* and a *SESSION.NAME* file are generated. The *SESSION.NAME* file is read by Nek5000 and gives the path to the relevant input files describing the structure and parameters of the simulations, along with the name of the *.rea file to run. The *pbs_submission* supplies information for the parallel run, compiles the Nek5000 executable, and runs the job in parallel. Note that the driver script *nek5000-dakota-driver* is designed to look for the log file of the simulation and pause until it detects it.
 - In the third part of the script, the job has finished, and post-processing is performed. Data are extracted from the output file using Visit, and then they are passed to Python script to compute the L2 error norm between the numerical and experimental data.
- **A Visit script** reads the Nek5000 output files, extracts the time average velocity magnitude from a slice at a given position, and stores the data in a comma-separated value (csv) format file. Note that Visit is called in batch mode.

- **The L2 error norm** between the experimental data and the numerical data is computed with a Python script. The value of the L2 error norm corresponding to the response function is written in a Dakota output file that is later returned to Dakota.

3.3 COUPLING TO STARCCM+

The coupling between Dakota and STARCCM+ also requires a driver to run each job, as well as script(s) to extract and post-process the data. Through its graphical user interface (GUI) and JAVA Scripting – Process Automation, STARCCM+ allows for easy setup of the job and extraction of the data in batch mode.

- The driver script calls java and Python scripts to set up the STARCCM+ input file, run the job, extract the data, and then return the computed L2 error norm to Dakota.
- The STARCCM+ input file is run in batch mode and requires a java script that will modify the input parameters and physical models for each run. It also extracts the data of interest at the end of the run to be stored in a csv format file.
- The L2 error norm is computed with a Python script and then written to the Dakota result file and returned to Dakota.

4. THORS MESH GENERATION

The Fuel Failure Mockup (FFM) facility and the Thermal-Hydraulic Out-of-Reactor Safety (THORS) facility were large thermal hydraulic liquid metal (sodium) loops built at ORNL in 1970, with testing started in 1971. The last tests took place in December of 1984, the program was closed in 1985, and facilities were decommissioned. The purpose of these tests was to support the Liquid Metal Fast Breeder Reactor (LMFBR) Program by acquiring liquid metal thermal hydraulic data.

During the 15 years of FFM and THORS operation, 11 different bundles were designed, 7 of which were built and tested in 14 different configurations. The initial charter of the program was to investigate bundle blockages, but the emphasis shifted to transient tests, sodium boiling, and dryout. All reports and data from the complete FFM and THORS programs are available at ORNL (Carbajo J. 2016). These data can be used to support future liquid metal reactor (LMR) programs and to benchmark liquid metal thermal hydraulic computer models.

Due to data classification (Applied Technology), this report does not include specific sizes of modeled geometries, but it does provide detail on the generated computational meshes. The primary focus of the meshing was to produce a universal grid that allows for import in any of the codes used for this UQ study. The Nek5000 code operates only on hexahedral elements, thus imposing the largest limitation on the meshing technique. To comply with hexahedral elements two viable options are currently available: (1) use of a code developed at ANL that is based on MATLAB, or (2) acquire HEXPRESS, commercial software developed by www.numeca.com. The MATLAB code is not universal, as it can generate meshes only for Nek5000 because no convertors exist to translate the mesh into any common format. However, the generated mesh includes high quality and curvilinear elements suitable for use in the Nek5000 code. Both options have been pursued, and two grids were developed with different geometries: a single-pin mesh generated with HEXPRESS, and a 7-pin bundle generated with the MATLAB code.

4.1 THORS SINGLE WIRE WRAP PIN

The geometry for this effort was created using computer-aided design (CAD) software. It consists of a wire-wrapped pin in a triangular matrix with a certain pitch. The wire wrap serves as a spacer for the pins and is a common means for arranging fuel in liquid metal reactors. Only one axial pitch is modeled; this provides sufficient axial distance for the flow to develop. The geometry was non-dimensionalized using the rod diameter as a scaling parameter. Normally, the pins have no preferential position relative to the wire wrap, but to simplify the meshing, the pins have been placed so that the wraps appear to be in the same direction. When meshing a periodic domain, this uniformity helps to obtain identical symmetry boundaries. Using 6 pins, the flow area around the central pin was extracted as shown in Figure 2. The original geometry is depicted on the left.

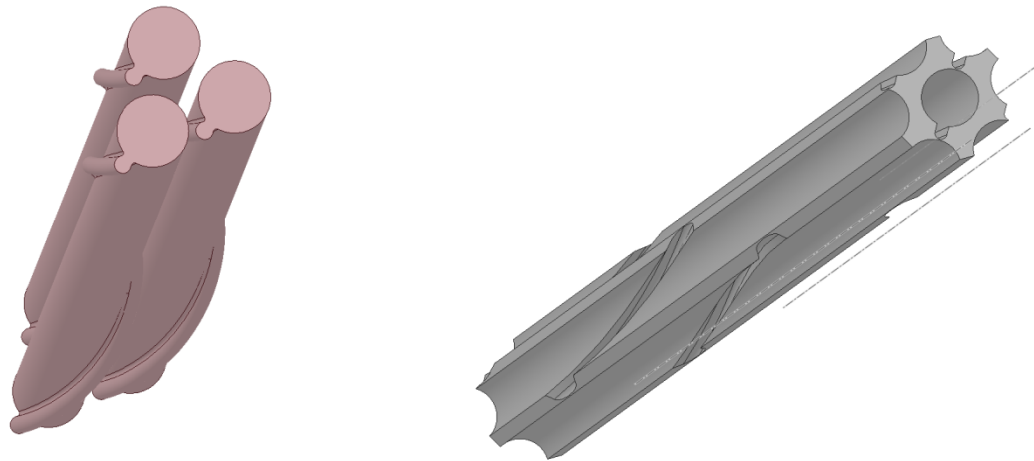


Figure 2. Original wire wrap pins arranged in triangular lattice of SFR (left), and extracted fluid domain used for meshing (right).

Generating a fully hex mesh on the resultant domain with predominantly curved and convoluted surfaces is a substantial task. Using the traditionally known meshers like CUBIT, GAMBIT, ICEM, GRIDGEN, etc., is very difficult and does not guarantee success. The volume-based meshing algorithms as octree decomposition have a greater capability and can be tailored to provide almost automatic meshing. Based on previous experience using HEXPRESS/HYBRID, the NUMECA Inc. meshing products, the decision was made to obtain a commercial license for this tool.

The software has two major features to help minimize the cell number and maximize the quality: tetrahedral (tet) stitching and geometry morphing. Tet elements are not allowed by requirement, but the other built-in option is the cut cell technique that results in hanging nodes. While some CFD codes accept such nodes, Nek5000 does not accept them, so this option was discarded. The only available fully hex option was to use a very small base size. In effect, the meshing was performed by adjusting the cells to the minimum geometry features, resulting in a very dense mesh with high cell count. Standard smoothing techniques such as Laplace and others were used to improve the mesh quality. The generated mesh is shown in Figure 3. Initially it was exported into Exodus format, and some statistics were produced by reading the mesh with the MOAB code package (

Table 1). The mesh has almost 300,000 hex elements and just over 100,000 quadrilateral (quad) sides that belong to the boundary facets.

Table 1. MOAB single pin mesh statistics

type	count	total	minimum	average	rms	maximum	std.dev.
Quad	100560	2.1e+04	0.00248	0.21125	0.227	0.9852	0.085121
Hex	299357	1.5e+04	5.54e-5	0.04934	0.052	0.1493	0.01766
1D	3994524	2.9e+06	0.00301	0.71527	0.975	9.2613	0.66289
Vertex	350407						

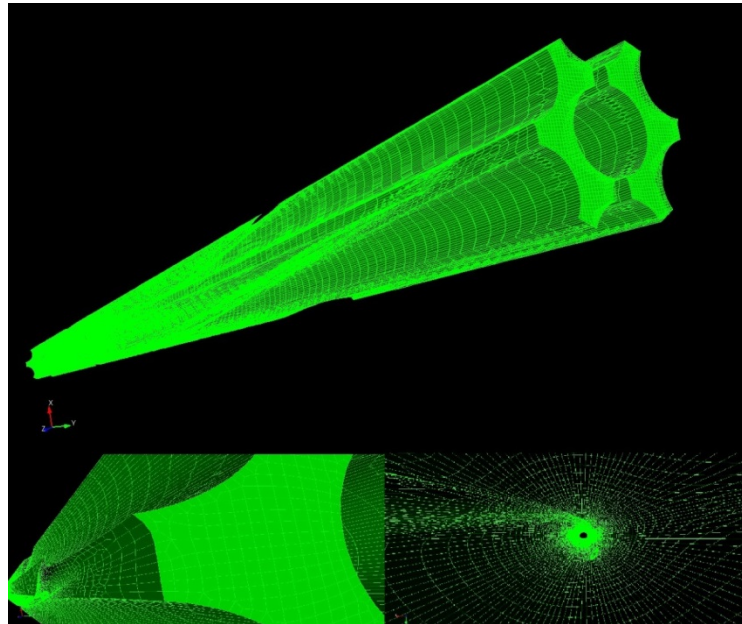


Figure 3. THORS single pin mesh: overall view (top), details (bottom left), and central pin wire (bottom right).

In reality, the wire is welded to the pin, and the two surfaces (pin and wire) have a single point of contact. Meshing such surfaces is impossible due to the singularities that appear, and some approximation of the problem is needed. In the case of the single pin mesh, overlapping surfaces are used, while in the 7-pin mesh, non-contact is allowed. Both of these techniques are acceptable if the mismatch is within 5% of the wire size. Non-contact is not appropriate for octree meshing since the cells have to be squeezed into the gaps, which creates significant distortion. Regardless of what technique is used, the element quality is

compromised. To verify the mesh adequacy for the modeled problem, some mesh diagnostics were run, and the results were compared to acceptable limits.

Some common metrics for mesh quality are *aspect ratio*, *skew*, *taper*, and *element volume*. Their values and the elements in compliance with those values and other mesh parameters are given in Table 2. Figure 4 illustrates the element distributions for each quality metric.

Table 2. Single-pin mesh quality metrics: aspect ratio, skew, taper, and volume ratios

Metrics	Target	Compliant (%)	5% below maximum	Average	Maximum	Standard deviation
Aspect	10	83	1 el.	8	111	2.99
Skew	0.7	67	49,961	0.48	1	0.33
Taper	0.5	92	1 el.	0.21	21.4	0.33
Jacobian	0.8	50	n/a	0.63	1	0.35

The aspect ratio is perhaps the most used parameter to qualify every mesh type (hex or tet). While 83% of the elements comply with the chosen limit of 10, only one element falls within 5% of the upper ratio. The maximum for the mesh is 111, and meshes have been run with more than 200 showing valid results.

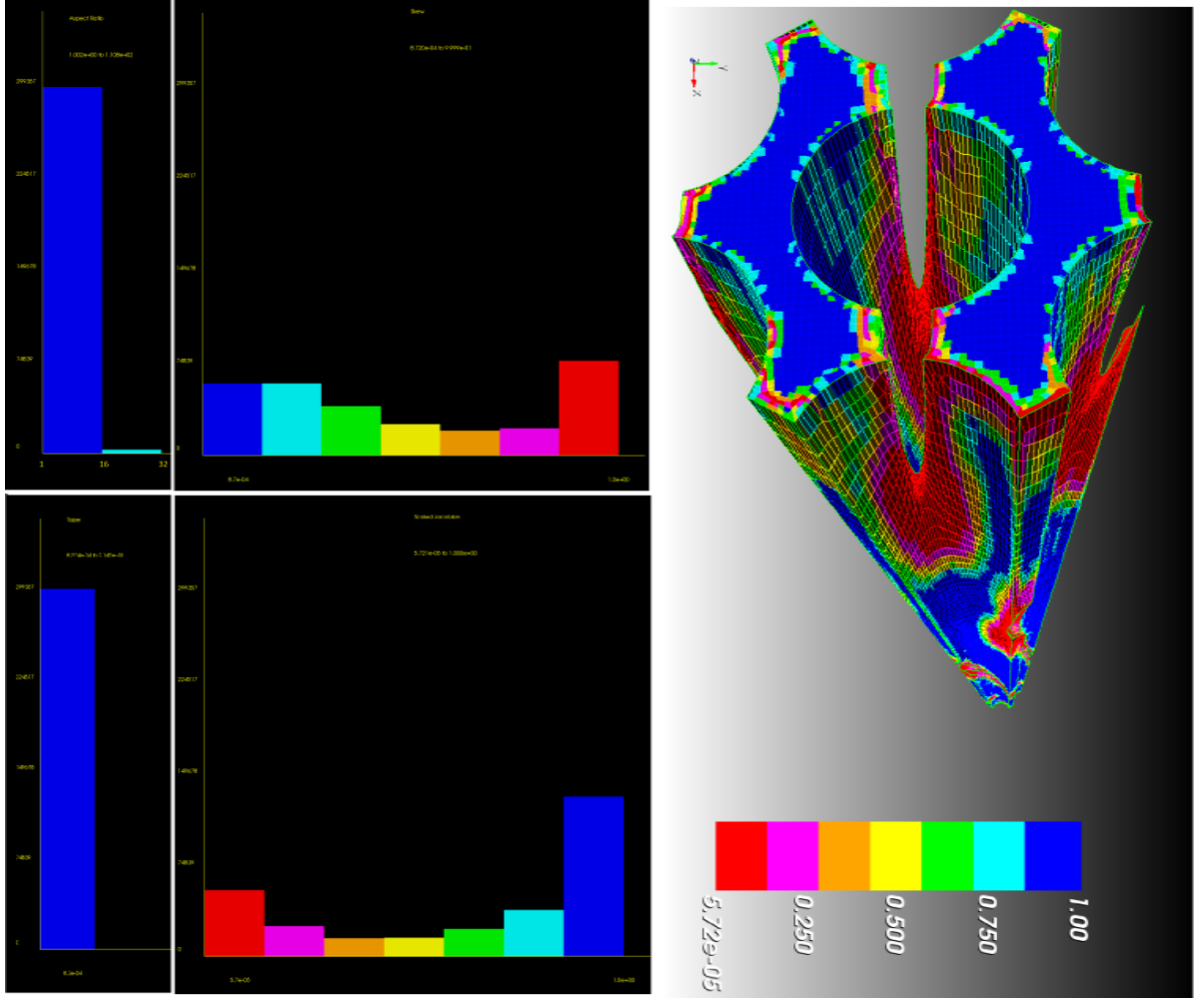


Figure 4. Single pin mesh quality metrics: graphical representation of element distribution per quality parameter (left), mesh colored by Jacobian showing the high distortion elements at the wire pin connection points (right).

The taper has even better values than the aspect ratio. This is likely inherently typical for volume-based meshing techniques, where the domain is usually first voxelized, and the produced initial volumes are quite symmetric.

The cells around the connection wire-pin remain the most critical. Varying the surface overlapping has little effect, and the variation margins are not large. Both Jacobian and Skewness capture those cells, which are highlighted in Figure 4 (right). The distorted cells follow the wire. However, the histogram of the Jacobean distribution shows that the mesh has a good number of qualifying cells, which should help resolve the bulk flow correctly.

HEXPRESS code can export meshes in Exodus or Fluent (.msh) formats. For such meshes to be used in Nek5000, they must be either run/read with MOAB or translated to Nek5000 native format. Although it was possible to read and diagnose the mesh with MOAB (see

Table 1), porting it into Nek5000 was not successful. Some of the major difficulties relate to the parallel configuration of MOAB, mesh partitioning in parallel, and the METIS and HDF5 libraries' performance.

A separate convertor had to be developed based on the PYTHON language. A prototypic file was used as a starting point and is available in the Nek5000 repository. Some of the boundary condition routines were updated and improved, and a Nek5000-specific mesh was generated. Two versions were set up, with walls on all external boundaries and symmetries at the pin sides. Periodic conditions were not set because (1) the sides are not conformal, and the PYTHON code cannot handle elements that are not identical, and (2) no boundary periodicity can be set in Nek5000 if the surfaces are not aligned with the coordinates. Therefore, the calculations performed with Nek5000 using the single-pin grid either were with symmetry or wall boundaries. The other code used to perform calculations on this mesh was the STARCCM code, which has no restriction on the type of boundary condition set (periodic or symmetry).

4.2 THORS 7 PIN BUNDLE

The Nek5000-compatible THORS 7-pin bundle mesh is generated with a MATLAB script developed at ANL (the resulting mesh can only be read by Nek5000). The script requires a certain number of input parameters such as pin diameter and pitch, wire diameter and pitch, and the number of pins. The resulting mesh is curvilinear, is nondimensional with respect to the pin diameter, is generated with inlet/outlet or periodic boundary conditions, and is directly written in a format compatible with Nek5000.

For this study, a THORS 7-pin bundle mesh with a nondimensional length of 51.5 with periodic boundary conditions was generated with the MATLAB script. The 3-D mesh and a cross section of the mesh are shown in Figure 5.

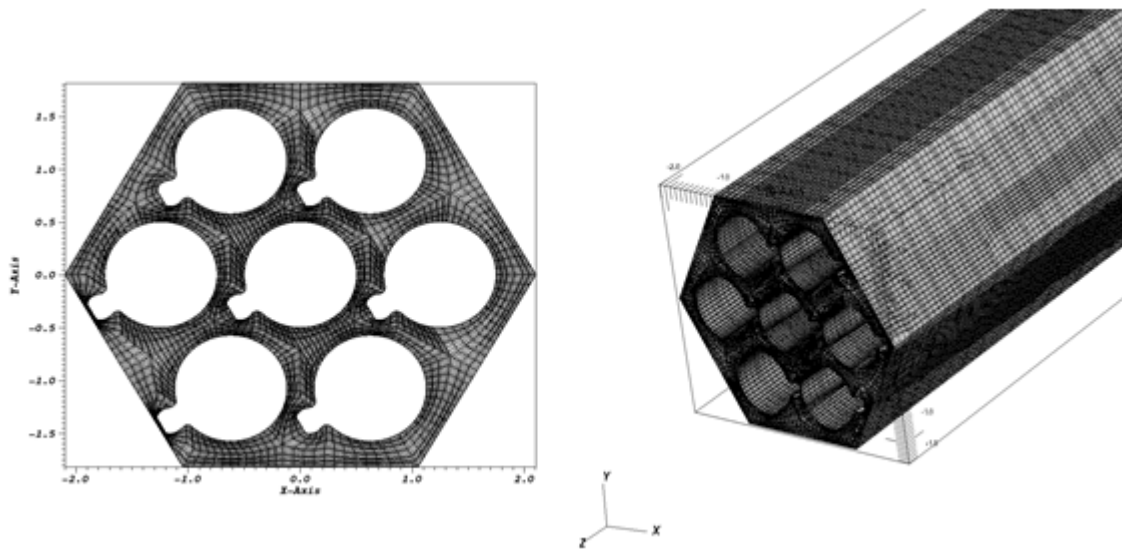


Figure 5 Cross section (left) and 3-D (right) views of the THORS 7-pin-bundle mesh.

5. DAKOTA Nek5000 UQ DEMONSTRATION RUNS

This section details the model, method, and setup used in the Dakota Nek5000 UQ demonstration runs performed for a turbulent flow in an infinite 3-D pipe.

5.1 THE MODEL OF INTEREST

Flow of Reynolds number $Re=31,000$ in a 3-D pipe with a nondimensional length of 13 diameter in the z-direction is investigated using the code Nek5000; the time-dependent incompressible Navier-Stokes equations are solved on an unstructured mesh of hexagonal elements using SEM. 3-D meshes are investigated with 32 (Figure 6), 84 (Figure 7), 184 (Figure 8), and 832 (Figure 9) radial elements. Curvilinear and linear versions are generated for each mesh. Thus, eight meshes are investigated in this study.

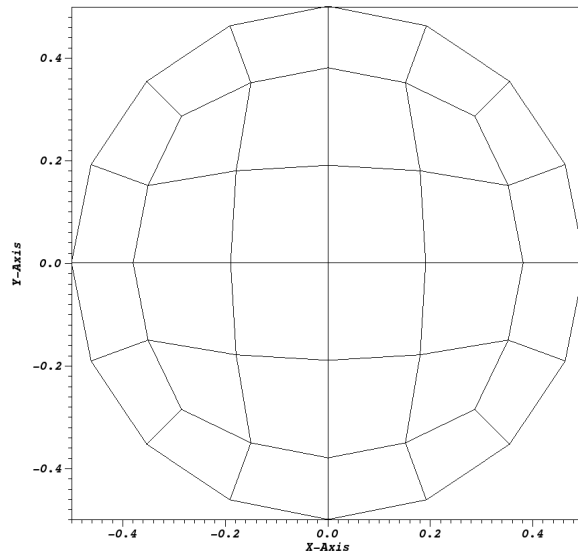


Figure 6. 3-D pipe mesh with 84 radial elements.

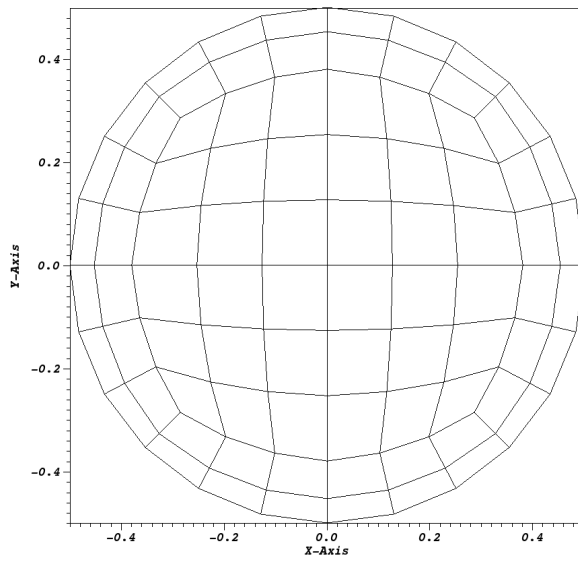


Figure 7. 3-D pipe mesh with 84 radial elements.

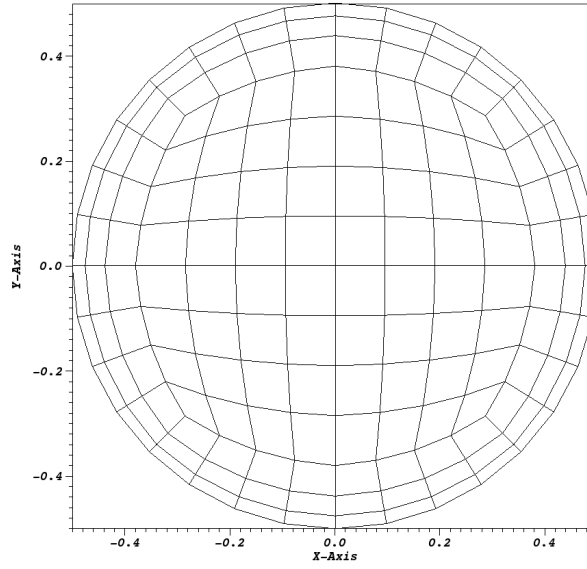


Figure 8. 3-D pipe mesh with 184 radial elements.

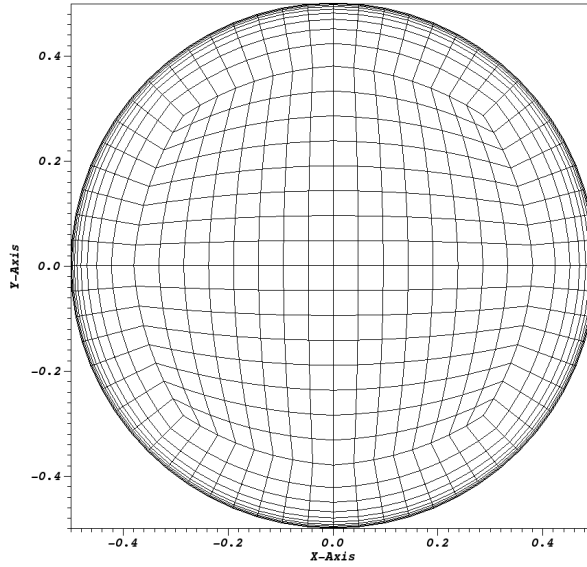


Figure 9. 3-D pipe mesh with 832 radial elements.

The curvilinear meshes were generated with the Nek5000 built-in mesh generator Pretex. First, a 2-D circle is meshed with the required number of radial elements and then extruded in a 3-D pipe. The script allows the user choose between inlet/outlet and periodic boundary conditions. The linear meshes are obtained from the curvilinear meshes after deleting the CURVED SIDE DATA block in each mesh file. All numerical results presented in this section were obtained with periodic boundary conditions while maintaining the spatial average velocity constant and equal to one at the inlet by setting the p054

parameter to 3 and the p055 parameter to -1 in the rea file. The code is run until $t = 100$ seconds, with a constant time step of 0.002 seconds using the Pn-Pn formulation.

5.2 THE SETUP AND METHOD

The UQ study is performed with three input parameters. The first parameter is the polynomial order $lx1$ that is set in the SIZE file and controls the number of degree of freedom in the mesh. The two other input parameters consist of filtering parameters p101 and p103 that allow to run Nek5000 with a LES-like model. These are set in the rea input file and applied to the velocity field. The parameter p101 controls the number of modes filtered, whereas p103 controls the amplitude of the filter. The overall effect of the filter is to suppress high wavenumber instabilities while preserving exponential convergence of the SEM.

Using a Dakota input file, the three input parameters are sampled and passed to the Nek5000 input files to run the session. For each session, the time average velocity is dumped every 10 seconds. At the end of the run, the L2 error norm between the numerical and experimental data describing the fully developed radial velocity profile is monitored. The L2 norm is computed using the following four-step process (all lengths are in nondimensional units):

- The time-averaged z-direction velocity is extracted from the Nek5000 output file from a slice at $z=13$ using Visit. This information is collected from 0–5 s and is stored in a text file, along with node coordinates.
- For each node of the slice S , the distance to the outside wall is computed using the following relation:

$$dist_{to\ wall} = D/2 - \sqrt{x^2 + y^2},$$
 where $D=1$ is the diameter of the pipe in nondimensional units.
- Once the variable $dist_{to\ wall}$ is obtained, the nodes are gathered into pools using the following rule: a node belongs to a pool if the relative difference between the value $dist_{to\ wall}(node)$ and the pool radius r_{pool} is less than a user-supplied tolerance: tol . When a node meets this criterion, the corresponding velocity magnitude $V(node)$ is added to the pool velocity V_{pool} . A script representing the above is:

```

for loop over nodes of slice S
    ratio = |rpool - distto wall(node)|/rpool
    if ratio > tol
        node does not belong to pool rpool
    else
        node belongs to pool rpool
        Vpool += V(node)
        Nnodes += 1
    endif
endfor
Vpool *= 1./Nnodes

```

Note that the number of nodes, N_{nodes} added to each pool must be tracked to compute the pool's average velocity. The pool radius r_{pool} can be set to equal the coordinates of the experimental data, but it is preferred to have a sufficient number of pools so that all nodes from numerical data fall inside a pool. The number of nodes stored in each pool depends on the value of the tolerance tol .

- The experimental and numerical values of the velocity magnitude are now available at the same coordinates and are used to compute the L2 norm of the error, as follows:

$$L_{error}^2 = \sqrt{\sum_i^{N_{pool}} |V_{pool}(i) - V_{exp}(i)|^2},$$

where N_{pool} is the number of pool. The L_{error}^2 norm value is returned to Dakota.

The process described above is driven by a Python script and thus does not require any user intervention besides launching the Dakota input file. Once Dakota has collected all L2 norms from the Nek5000 runs, a UQ or statistics assessment is performed.

To illustrate the above steps involved in the post-process of the numerical simulations, plots of the numerical data (pool data), interpolated data, and experimental data are given for the velocity magnitude in Figure 10 ($lx1=5$) and Figure 11 ($lx1=8$). L2 error norm measures the difference between the experimental and numerical (Nek5000) data.

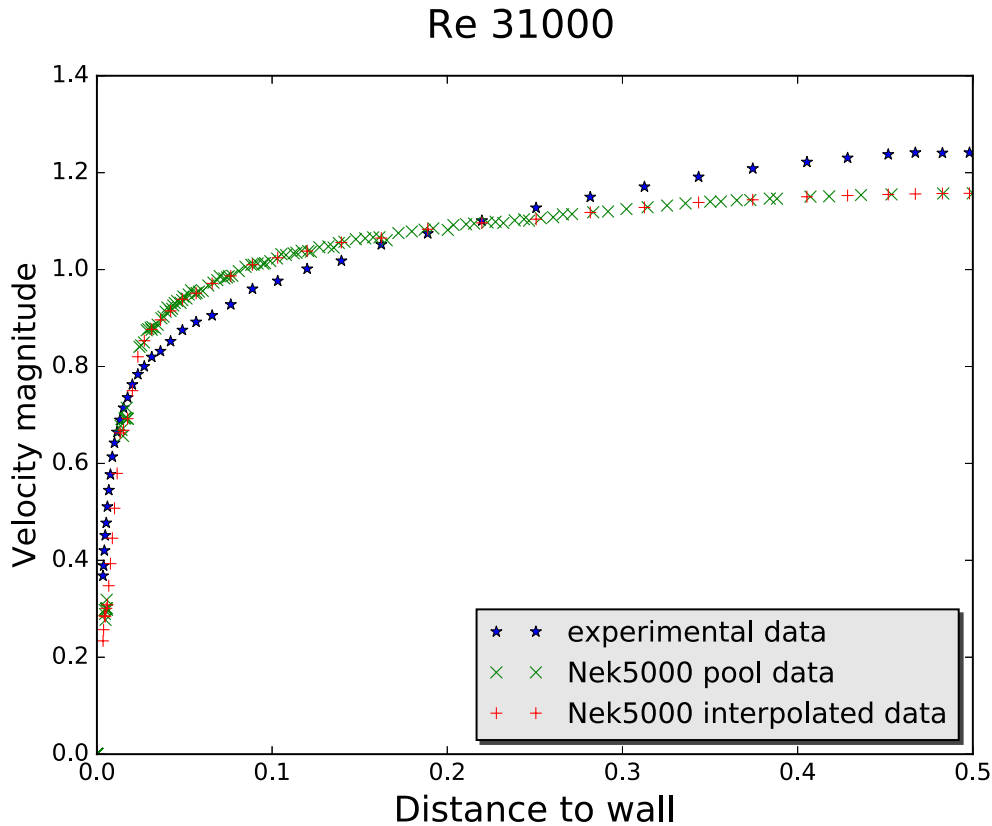


Figure 10. Numerical (Nek5000), experimental, and interpolated data as a function of the wall distance after postprocessing for mesh with 184 radial elements and $lx1=5$.

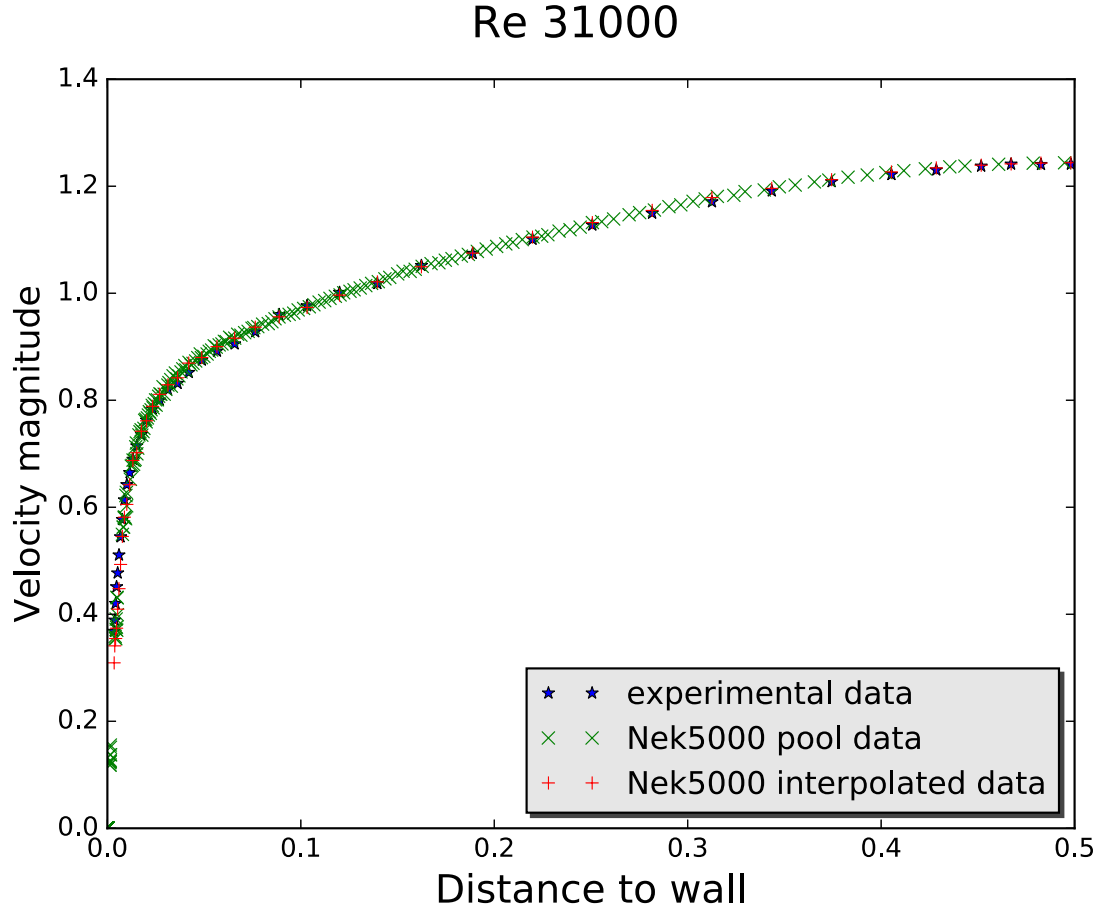


Figure 11 Numerical (Nek5000), experimental, and interpolated data as a function of the wall distance after postprocessing for mesh with 184 radial elements and $lx1=8$.

Using the coupling between Dakota and Nek5000, a calibration study can be performed to match the numerical solution to the experimental data by varying the polynomial order $lx1$ and the filtering parameters $p101$ and $p103$.

5.3 NUMERICAL RESULTS

As previously mentioned, three input parameters are of interest for this study: the polynomial order $lx1$ and the filtering parameters $p101$ and $p103$. These parameters are varied in a certain range using the *discrete_uncertainty_set* method from the Dakota package. In addition to these three parameters, influence of the initial mesh density (number of radial elements) and quality (curvilinear vs linear) are also investigated. Since four different mesh densities are used, each of which is generated with curvilinear and linear options, a total of eight meshes are considered. The objective of this study is to perform a mesh sensitivity and calibration study by using the experimental data as reference, with a focus on the influence of the mesh quality: (1) curvilinear vs. linear, (2) the degree of freedom that is a function of the mesh density and the polynomial order, and (3) the filtering parameters.

Multiple Dakota studies were run with a fixed mesh (using one of the densities and either the curvilinear or linear option), with each study varying the three parameters $lx1$, $p101$, and $p103$. Results of these

studies were then concatenated into an Excel spreadsheet with 5 columns. Part of the spreadsheet is shown in Table 3 for illustrative purposes. Mesh type = 0 corresponds to a linear mesh, while mesh type = 1 corresponds to a curvilinear mesh. Note that many simulations were not able to converge and give a proper norm for the L2 error. In those cases, the L2 error norm was returned as a “nan.” These data points were ignored in the statistical analysis. For the full analysis, there were 314 rows of data with 78 rows having “nan” for the L2 error norm (about 25% of the data set). This resulted in 236 good samples used in the statistical analysis.

Table 3. Portion of the complete parameter study over five input parameters

p101	lx1	p103	Mesh Density	Mesh Type	L2 ErrorNorm
0	3	0.05	84	0	1.703620523
1	3	0.05	84	0	nan
0	4	0.05	84	0	1.120704629
1	4	0.05	84	0	nan
0	5	0.05	84	0	1.003631826
0	5	0.05	84	1	0.759151201
1	5	0.05	84	1	1.065148484
0	6	0.05	84	1	0.604024074
1	6	0.05	84	1	0.781776655
0	3	0.1	84	1	1.645992541
1	3	0.1	84	1	nan
0	4	0.1	84	1	1.005396474
1	4	0.1	84	1	nan
0	5	0.1	84	1	0.723992347
1	5	0.1	84	1	1.098424575

The data in Table 3 can be analyzed with a statistical method called *main effects analysis*. This falls under a class of methods called ANOVA (analysis of variance). The analysis determines whether the mean response (in this case, the mean L2ErrorNorm) is the same across the different levels of one input parameter. An example question is, “does the mean L2ErrorNorm change as p101 goes from 0 to 1 to 2?”

Ideally, there would be the same number of samples per level of a variable, so that the number of simulations with p101=0, p101=1, and p101=2 would be equal. However, in this dataset, each level does not have the same number of samples due to the concatenation of data across multiple Dakota studies and the many nans. However, the analysis does take into account the varying number of samples per level.

Figure 12 shows a main effects plot. This plot shows the mean L2ErrorNorm for each level of each variable. For example, for p101, the mean L2ErrorNorm across all samples that have p101=0 is 0.97. The mean L2ErrorNorm dips to 0.86 at p101=1, but it increases at p101=2 to 0.90. The overall mean of the L2ErrorNorm averaged over all samples is shown in the dotted horizontal line with a value of 0.924. Figure 12 shows that lx1 has the most significant main effects: the mean L2ErrorNorm varies quite broadly as lx1 goes from 3 to 10: from 1.64 to 0.68. The mesh density and the mesh type both exhibit significant effects. The differences in the means from p101 and p103 are not significant, as explained in further detail below. The figures and statistical analysis presented in this section were produced using a statistical analysis package called Minitab. However, there are other statistics packages, such as R and the Matlab statistics toolbox, which can produce similar plots.

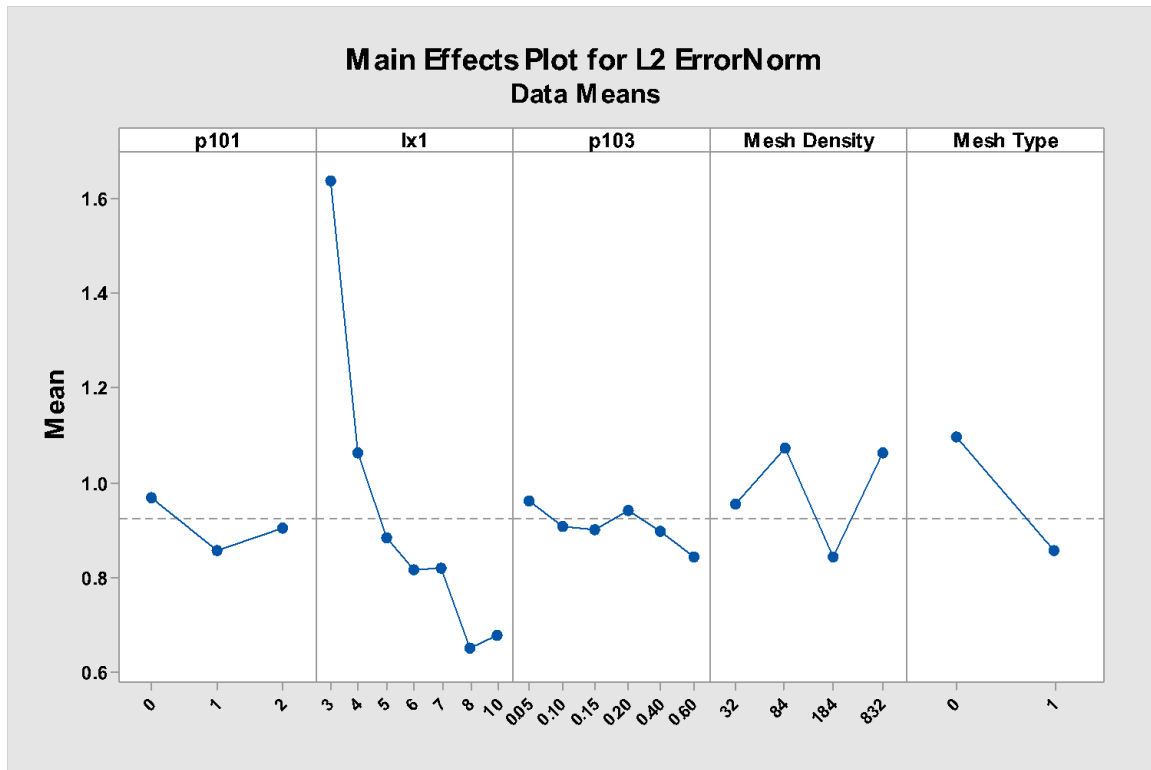


Figure 12 Main effects plot for L2 error norm

For a main effects analysis, there is a null hypothesis stating that all the means at different levels of a particular variable are equal. The alternate hypothesis states that at least one mean is different. There are various tests for comparing means. Often these tests involve confidence intervals on the difference in the means. If the confidence interval on the difference of two means contains zero, the means are not considered statistically significantly different.

First, p101 is investigated, showing an interval plot as in Figure 13:

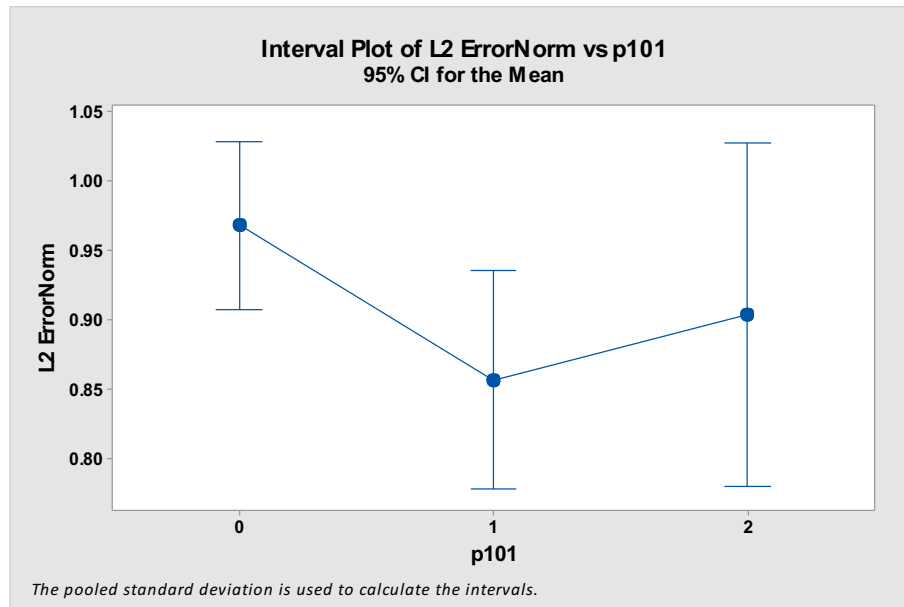


Figure 13. Interval plot for means of L2 error norm for P101.

This plot shows the 95% confidence intervals on the mean values of the L2ErrorNorm as p101 go from 0 to 1 to 2. Although the means are different, there is significant overlap between these intervals. This leads to the conclusion that the null hypothesis that the means are the same cannot be rejected. In addition to the intervals around the means, Minitab generates summary information about the test, as shown below. The null hypothesis is that all means are equal, while the alternate hypothesis is that at least one mean is different. Out of the 314 data points, 78 rows were unused because of the nan response discussed earlier. The analysis of variance section shows a p-value of 0.087. If the p-value is zero, the null hypothesis can be rejected. Since the p-value is greater than the significant level of 0.05, the null hypothesis for p101 cannot be rejected. The means of the L2ErrorNorm at the three levels of p101 are shown along, with their standard deviations and confidence intervals. Finally, a grouping analysis is performed which indicates that all three of these means could be said to come from the same group.

One-way ANOVA: L2 ErrorNorm versus p101

Null hypothesis All means are equal
 Alternative hypothesis At least one mean is different
 Significance level $\alpha = 0.05$
 Rows unused 78

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
p101	2	0.6031	0.3016	2.46	0.087
Error	233	28.5113	0.1224		
Total	235	29.1144			

Means

p101	N	Mean	StDev	95% CI
0	129	0.9682	0.4112	(0.9075, 1.0289)
1	76	0.8572	0.2272	(0.7782, 0.9363)
2	31	0.9041	0.3160	(0.7804, 1.0279)

Grouping Information Using the Tukey Method and 95% Confidence

p101	N	Mean	Grouping
0	129	0.9682	A
2	31	0.9041	A
1	76	0.8572	A

Means that do not share a letter are significantly different.

By comparison, Figure 14 shows the results for the lx1 variable. Here the means and their confidence intervals do not overlap, especially the mean for lx1=3, which has a mean L2ErrorNorm of 1.64. The p-value is zero, indicating that the alternate hypothesis should be accepted. The grouping analysis indicates that some of these levels might share groups. For example, lx1=7 could be in the same population as lx1=6 (group C) or lx1=8 and 10 (group E). Overall, however, it appears there are five distinct groups with different means from the lx1 study.

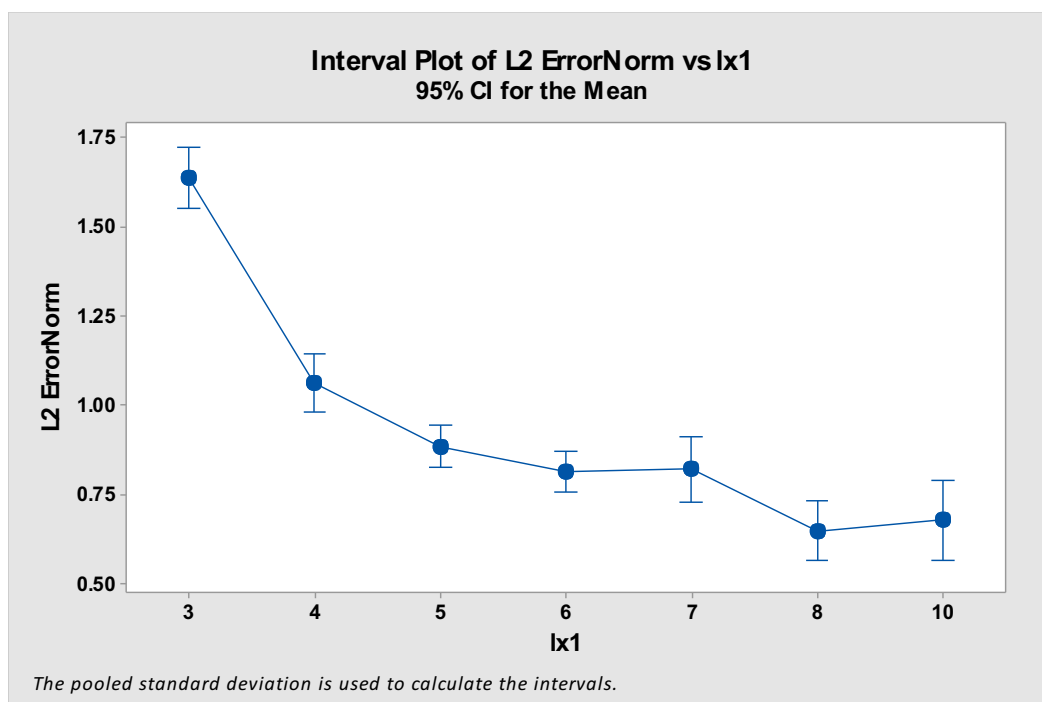


Figure 14. Interval plot for means of L2 error norm for Lx1.

One-way ANOVA: L2 ErrorNorm versus lx1

Null hypothesis	All means are equal
Alternative hypothesis	At least one mean is different
Significance level	$\alpha = 0.05$
Rows unused	78

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
lx1	6	17.84	2.97333	60.39	0.000
Error	229	11.27	0.04923		
Total	235	29.11			

Means

lxl	N	Mean	StDev	95% CI
3	26	1.63667	0.03767	(1.55093, 1.72242)
4	29	1.0632	0.1981	(0.9820, 1.1444)
5	53	0.8852	0.2309	(0.8252, 0.9453)
6	62	0.8147	0.2167	(0.7592, 0.8702)
7	23	0.8212	0.3472	(0.7300, 0.9123)
8	28	0.6496	0.2148	(0.5670, 0.7322)
10	15	0.6790	0.2078	(0.5661, 0.7918)

Grouping Information Using the Tukey Method and 95% Confidence

lxl	N	Mean	Grouping
3	26	1.63667	A
4	29	1.0632	B
5	53	0.8852	C
7	23	0.8212	C D E
6	62	0.8147	C D
10	15	0.6790	D E
8	28	0.6496	E

Means that do not share a letter are significantly different.

The p103 variable, shown in Figure 15, is the least significant: the confidence intervals for the means across levels are almost completely overlapping, and the p-value is 0.87, indicating a strong acceptance of the null hypothesis:

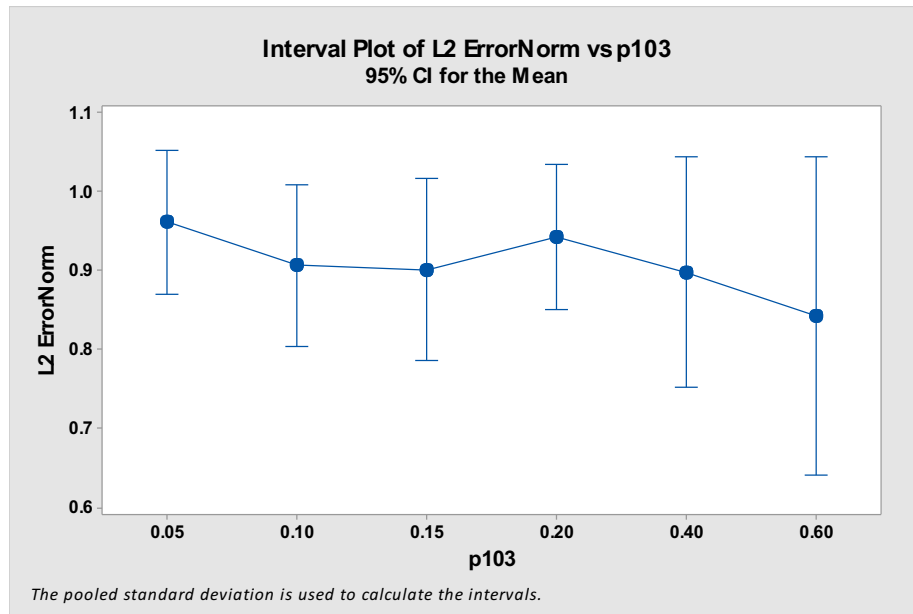


Figure 15. Interval plot for means of L2 error norm for P103.

Finally, the mesh density and the mesh type are examined. For the mesh density shown in Figure 16, the p-value is zero, and the null hypothesis is rejected. An interesting factor regarding mesh density is that there appear to be two groups: the mean at mesh density = 32 is similar to the mean at 184, and the mean at 84 is similar to the mean at 832. At this time, a good explanation for this has not been determined except to that there are only 14 data points for the 832 data set, and the standard deviation of the

L2ErrorNorm at this level is 0.46, which is larger than at the other levels (shown by the larger confidence interval), as shown in the statistical summary below.

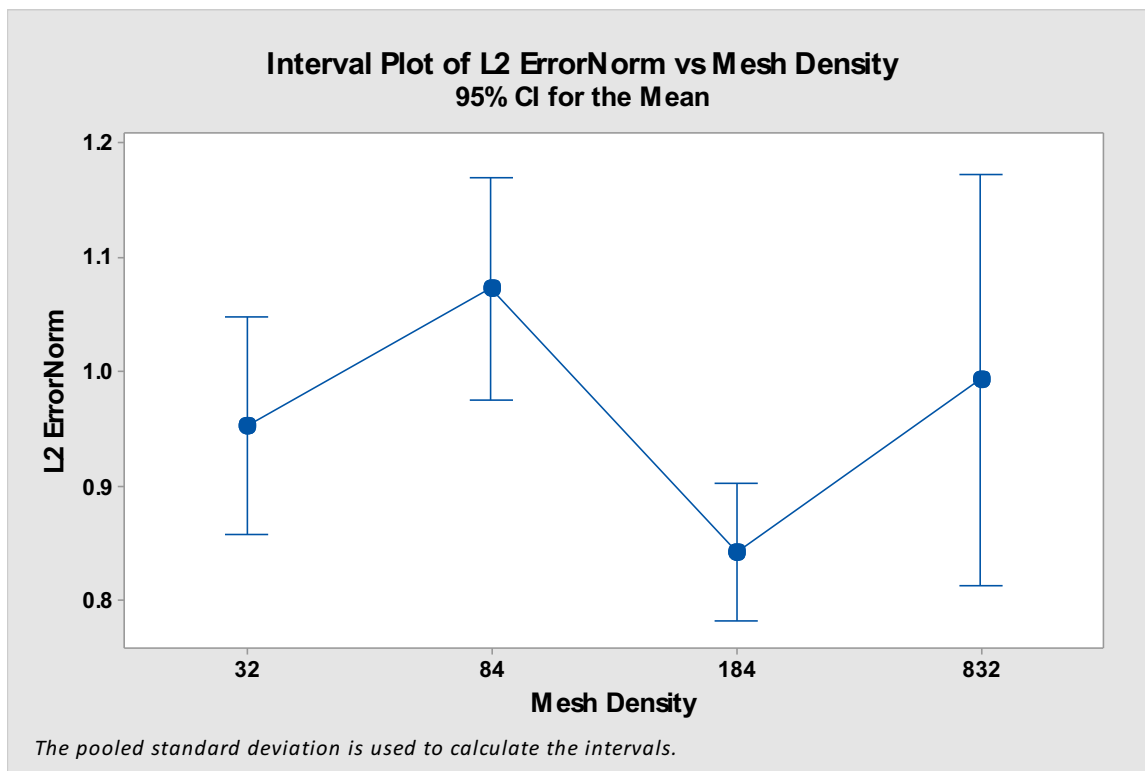


Figure 16. Interval plot for means of L2 error norm for mesh density.

One-way ANOVA: L2 ErrorNorm versus Mesh Density

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Mesh Density	3	2.018	0.6726	5.76	0.001
Error	234	27.331	0.1168		
Total	237	29.349			

Mesh Density	N	Mean	StDev	95% CI
32	50	0.9536	0.3081	(0.8584, 1.0488)
84	48	1.0732	0.3163	(0.9760, 1.1704)
184	126	0.8424	0.3490	(0.7824, 0.9023)
832	14	0.993	0.460	(0.813, 1.173)

Grouping Information Using the Tukey Method and 95% Confidence

Mesh Density	N	Mean	Grouping
84	48	1.0732	A
832	14	0.993	A B
32	50	0.9536	A B

184 126 0.8424 B

Means that do not share a letter are significantly different.

For the mesh type, the results in Figure 17 show that the mean L2ErrorNorm is different between level 0 (the linear mesh) and level 1 (the curvilinear mesh). It is lower for the curvilinear mesh, which is to be expected. The p-value for this test was 0.0, indicating that the null hypothesis would be rejected, and the alternate hypothesis of different means would be accepted. The full statistical results for mesh type are not shown here to save space, but the mesh type analysis shows a clear difference between linear and curvilinear results.

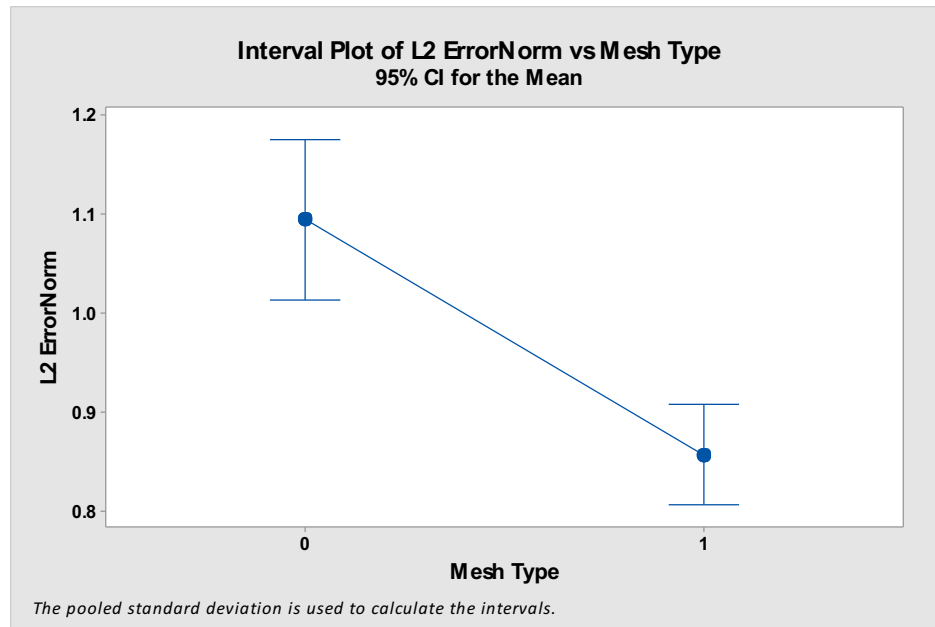


Figure 17. Interval plot for means of L2 error norm for mesh type.

More runs are being performed with a mesh density of 832, and the behavior of the linear vs. curvilinear mesh is being investigated in more detail. One type of analysis that can be performed with these data is an Interaction plot, which shows the mean values of the response (in these studies, the response is the L2 ErrorNorm) as a function of two parameters. To further understand the relationship between mesh density and mesh type, Figure 18 shows the interaction plot over these variables. For a mesh density of 32, the most reduction in the mean L2ErrorNorm is seen when going from a linear to curvilinear mesh. The improvement in the mean L2ErrorNorm is almost identical for mesh densities 84 and 184 and 832 (the lines are parallel). However, there is a lower mean error for mesh density of 184 than the mean error for mesh density of 832 (the purple line is above the green line). We expect the line for 832 to be below that of 184, but it is not. We are investigating this further. Note that our previous analyses, which had even fewer runs of the 832 mesh, showed us how sensitive these mean value lines are to the limited number of data points. We are currently running more configurations at a mesh density of 832.).

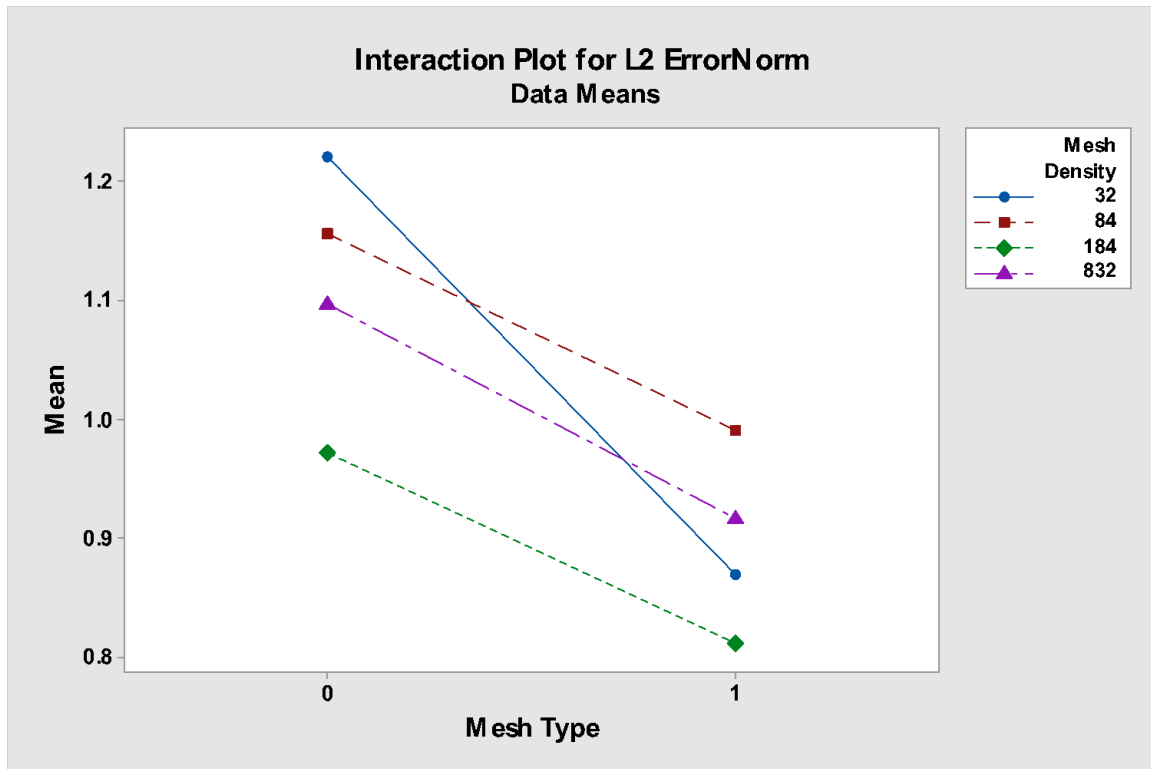


Figure 18. Interaction plot between mesh type and mesh density.

Figure 19 shows the confidence intervals for all mesh densities and types. The confidence interval is larger for the mesh density 832. More data are being gathered to improve the confidence intervals for mesh density 832. One possible cause of the higher mean L2ErrorNorm from mesh density 832 is that these runs mainly include data from lx1 orders 3 and 5 for the 832 data points: there are only 2 data points with lx1 order of 7 for the 832 mesh. In contrast the other mesh densities have higher order data points (tens of runs at lx1 order greater than 5), which lowers their mean L2ErrorNorm. There are problems running the 832 mesh with a higher polynomial order due to increasing run times and frequent failures, but data are still being completed the data for this study.

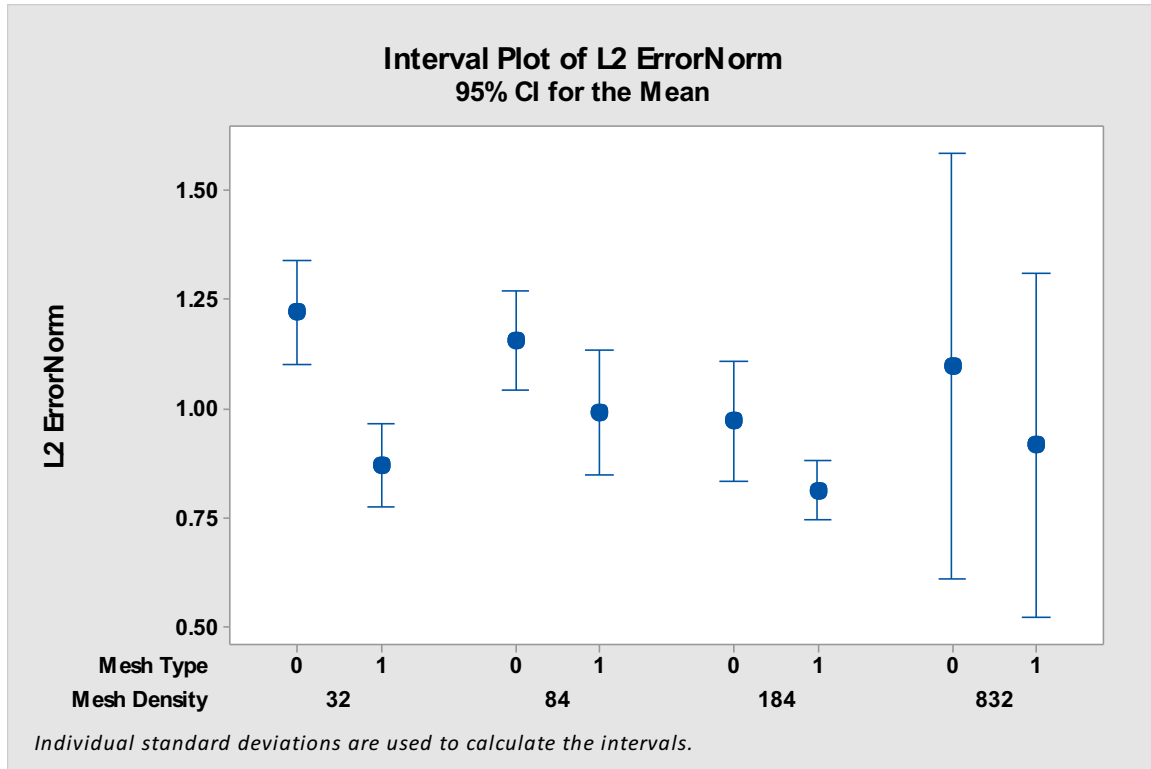


Figure 19. Summary plot of confidence intervals for the mesh type parameter.

In summary, the results from the extensive parameter studies performed with Dakota and Nek5000 indicate the following:

1. The filtering parameters p101 and p103 did not significantly affect the L2ErrorNorm: p103 was particularly insignificant.
2. Lx1 showed a dominant main effect on the L2ErrorNorm. As the levels of Lx1 increased from 3 to 10, the L2ErrorNorm decreased.
3. Both the mesh density and the mesh type did show significant main effects on the L2ErrorNorm. The mesh type showed a lower mean L2ErrorNorm for the curvilinear mesh vs. the linear mesh. The mesh density had a more complex pattern. At this point, the mean L2ErrorNorm values differ across the four levels of mesh density. The pattern is being investigated to determine why the mean error for mesh density of 832 is higher than that of mesh density 184. At this time, we believe it is due to having an insufficient number of runs at higher lx1 orders for the 832 mesh because of extensive run-times for these data points.

6. STARCCM+ SINGLE PIN THORS CALCULATIONS

The objective of this preliminary work is to simulate a turbulent flow in a THORS single wire wrap pin. The geometry was meshed with EXPRESS as described in Section 4.1. Using STARCCM+, the pressure drop was predicted for each of the nine inlet velocities and compared against experimental (Carbajo J. 2016) and correlation data (TODREAS 1986) (Novendstern 1972). Since STARCCM+ implements various turbulent flows, an analysis of sensitivity to the turbulent models was performed with Dakota to

investigate their impact on the numerical pressure drop values. Three different turbulent models were considered:

1. Realizable k - ϵ two-layer turbulent model (default k - ϵ turbulent model)
2. V2F variation of the k - ϵ turbulent model
3. The shear stress transport (SST) k - ω turbulent model with default settings

The Realizable k - ϵ turbulent model is an improvement over the standard k - ϵ model that is in theory unsuitable for applications in the viscous sublayer. The V2F variation of the k - ϵ model was designed to improve the accuracy of the k - ϵ model near walls and thus to better capture near-wall turbulent effects. The third turbulent model of interest is the well-established SST k - ω model, which is capable of resolving boundary layers.

For the STARCCM+ simulations, liquid sodium is used with constant density and viscosity properties. The simulation is run with an incompressible model with inlet/outlet conditions until the converging criteria are met. At the end of the run, the pressure values at the inlet and the outlet boundaries are retrieved and used to compute the pressure drop that is compared against experimental and correlation data. The plots are given as a function of the Reynolds number. Note that the same mesh is used when running STARCCM+ with the three different turbulent models and that STARCCM+ automatically applies a special wall treatment to give proper mesh thickness near the boundary walls.

A plot of the velocity magnitude obtained with STARCCM+ in a slice perpendicular to the flow direction is shown in Figure 20.

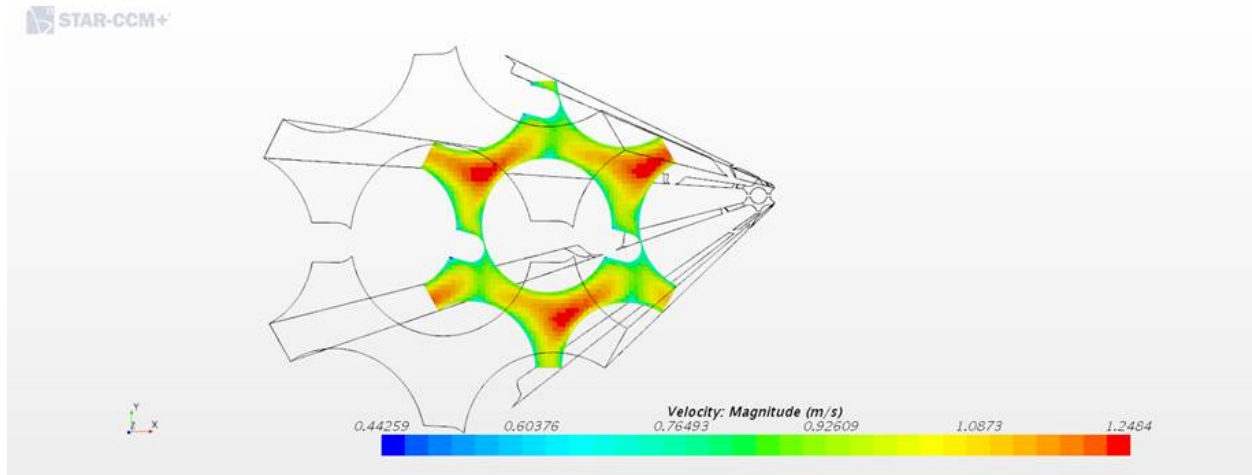


Figure 20. Cross section view of a turbulent flow in THORS single pin.

The experimental (THORS facility), numerical (STARCCM+) and correlation (Todreas and Novendstern) pressure drops are plotted as a function of the Reynolds number in Figure 21. Lines of plus or minus 40 % and 25 % of the experimental data are also plotted. On Figure 21 we observed that the predicted pressure drops with Todreas correlation best match the numerical pressure drops computed with the three different STARCCM+ turbulent models. It seems that the (SST) k - ω turbulent model matches Todreas correlation data with great accuracy for increasing Reynolds number. It is also noted that the pressure drops computed from Novendstern model under-predict the STARCCM+ pressure drops.

Still on Figure 21, we observe that the correlation data, the numerical data and the experimental data only agree for Reynolds number below $1e5$ within a reasonable uncertainty. When increasing the Reynolds number, the numerical and Todreas correlation pressure drops over-predict the experimental pressure drop by 40 %. On the other hand, pressure drop predicted with Novendstern correlation seem to give the best agreement with the experimental data within a 25 % uncertainty.

A plausible explanation to the large uncertainty between the experimental data and the predicted data is the large Reynolds numbers. The Todreas and Novendstern correlations were developed and tested for Reynolds numbers not larger than $1e5$ and $2e5$, respectively. In this study, the Reynolds number varies from $5e4$ to $3.5e5$. Also, to complete this study and draw more accurate conclusions, a mesh sensitivity analysis will have to be performed.

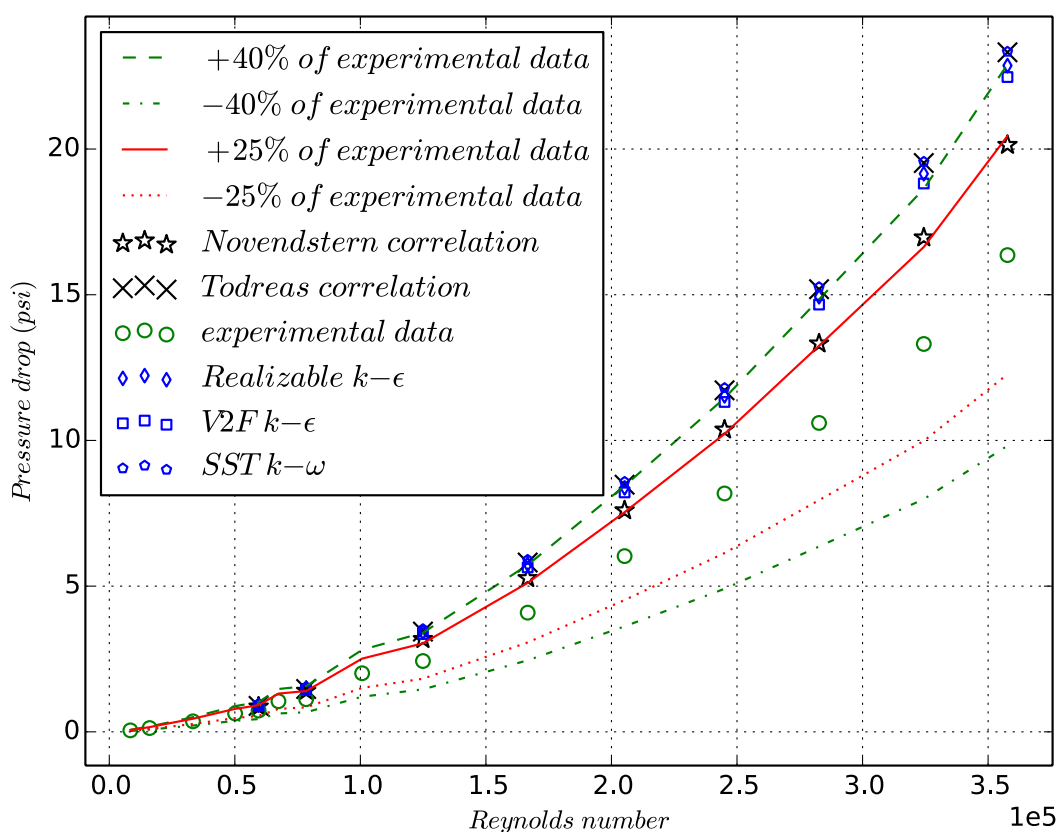


Figure 21 Numerical (STARCCM+), correlation (Todreas and Novendstern) and experimental pressure drop as a function of Reynolds number for a THORS single pin.

7. NEK5000 WIRE WRAPPED BUNDLE CALCULATIONS

The Nek5000 SEM code was used to perform the initial calculation with the meshes developed in Section 4. The primary objective of the calculations was to assess the code capability to produce a

converged pressure-velocity solution in the Reynolds number range of several hundred thousand, which is typical for the THORS experiments.

7.1 SINGLE-PIN THORS SIMULATION

The single pin wire wrap mesh required conversion to the native Nek5000 format (.rea file in Nek5000 nomenclature) in order to be run with the code. The mesh generation section above outlines the process. After the conversion, rea2re2 software, part of the Nek5000 utilities, was used to separate the mesh from the .rea file and write it in a binary format. Then another utility (genmap) was used to partition the mesh. The calculations were run in parallel on 320 cores. The problem is in nondimensional form. The length scale is the pin diameter together with an average velocity of one, and a density of one. The velocity scale is 6.93 (an average number for THORS specific velocity range), and the resultant Re is $\sim 160,000$.

Because the geometry is a periodic section of the entire bundle representing a single pin in hexagonal configuration of six neighboring pins, the sides form open boundaries that are not aligned with the global coordinates. Currently Nek5000 does not allow periodic conditions to be set on such boundaries. This limitation required approximation of the periodic conditions with symmetries. When using symmetry conditions in Nek5000 that are not aligned with the coordinates, the stress formulation for viscosity is required. Only the PN – PN-2 method in Nek5000 implements the stress formulation, so the code use for that method is limited.

The initial runs of the case with symmetry boundaries appeared to be unstable, and a back-up case was set up substituting the symmetry conditions with walls. Both cases (symmetry and walls) use inlet–outlet conditions along the main axis of flow direction. The model spans over 50 length scales in that direction, which allows for proper application of the inlet–outlet conditions. Table 4 gives the most important calculation parameters for both cases.

Table 4. Summary of run parameters for THORS single-pin mesh

Side boundary conditions	Method	Viscosity	GLL order (initial)	Stabilizing filter weight	Pressure solver	CFL
Symmetry	PN - PN-2	Stress formulation	4	0.5	PCG	0.25
Wall	PN-PN	Constant	2	0.2	GMRES	2.0

The case with symmetry boundaries runs very instable and easily diverges on velocity. The traditionally used GMRES solver for pressure was replaced with the PCG solver. This provided a convergence on the pressure, but the iterations increased to several hundred. This slowed down the calculation required for each time step significantly. The starting Re number was 500 (no convergence occurred initially on higher Re numbers), and gradually it increased to 1000. Convergence on larger Re numbers could not be achieved, regardless of the attempts made by changing a number of parameters: limiting the CFL to 0.25, reducing the time step, increasing the run time for a constant Re, varying the weight of the mode filtering, etc. Based on experience with wire pins (TODREAS 1986), the flow regime is most likely still laminar or transitional.

Figure 22 presents a plot of velocity contours at six positions along the main flow axis. The contours show the contact points of the central pin wire with the adjacent rods. The velocity enters the domain with a flat (constant) profile and then redistributes due to the flow channel geometry reaching a magnitude that is more than twice as high at some spots. Using the central pin wire as a reference, it can be seen that the

highest velocity area is always on the left side of the wire. The same is valid for the other pin/wires. This indicates that a swirl forms because of the wire. Unfortunately, due to limitations in Nek5000, it is not possible to run the same model with periodic boundaries to find the differences in the flow patterns between periodic and symmetry boundary conditions. Work is also being performed in parallel on a larger domain model that can provide the lines of flow symmetry. Comparison of those lines to the result obtained with the single-pin model may provide more insight on the flow distribution in the bundle. One of the goals of this calculation is to define the minimum representative computational domain sufficient to describe the flow in an infinite lattice of pins. This would reduce the computational effort for flow analyses, taking into account the complicated geometry that is unsuitable for meshing, the large number of pins in a fuel bundle, and the relatively high number of grid elements per unit of volume.

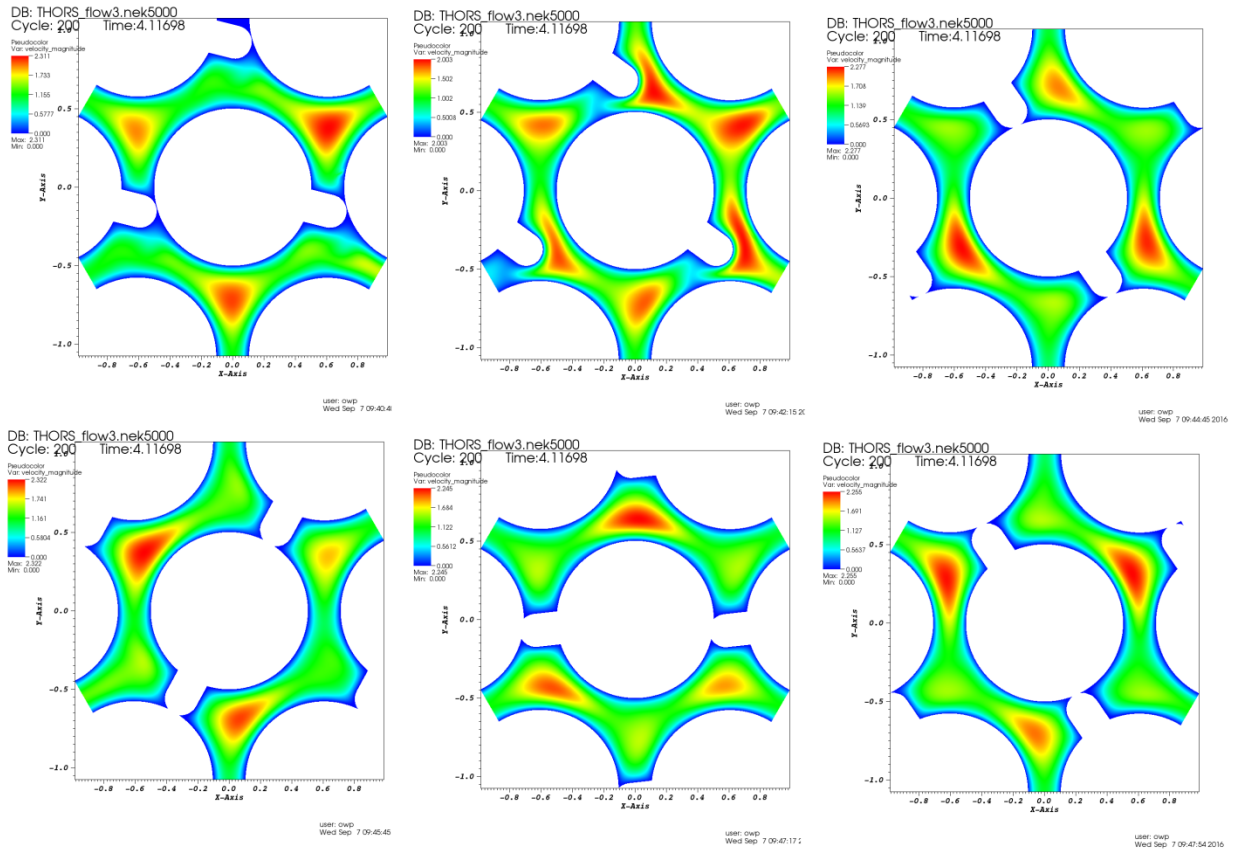


Figure 22. Velocities at six axial positions of the single pin mesh, symmetry boundaries on the periodic faces.

The case with walls runs more stable, and the flow is in turbulent regime. The simulation reached 76 seconds run time converges at Re numbers below 12,000. The time step decreases with increase of the Re number, but it stays in the order of one millisecond average with a CFL number of about 0.5. A further increase of the Re number leads to an increase of the divergence term ($\text{DIV}[V]$) and finally leads to the velocity solver crashing. The pressure solution stays converged within the allowed number of iterations (usually below 100) and accuracy ($5e-5$).

Figure 23 shows a snapshot of the instantaneous velocity at two axial locations near 50 length scales mark. One of the locations is at the level where the wire comes in contact with the pin, and the other location is downstream. Both show increased velocity intensity in the vicinity of the wire, with a magnitude of twice the mean velocity. The result seems to look qualitatively close to the run with

symmetry's boundary, with some more developed turbulent (eddy) flow structures. The wall type boundary condition at the side faces results in zero velocity (Figure 23), and it differs from the run with symmetry condition shown in Figure 22.

The single wire pin calculations are the first attempt to use mesh for Nek5000 generated by commercial meshing software. They demonstrate that one can mesh, set up, and run a practical problem with relatively high geometrical difficulty in Nek5000. The encountered difficulties with the problem convergence point to the Nek5000 requirement for high mesh quality. Further improvement of the mesh quality and the used solution techniques is necessary.

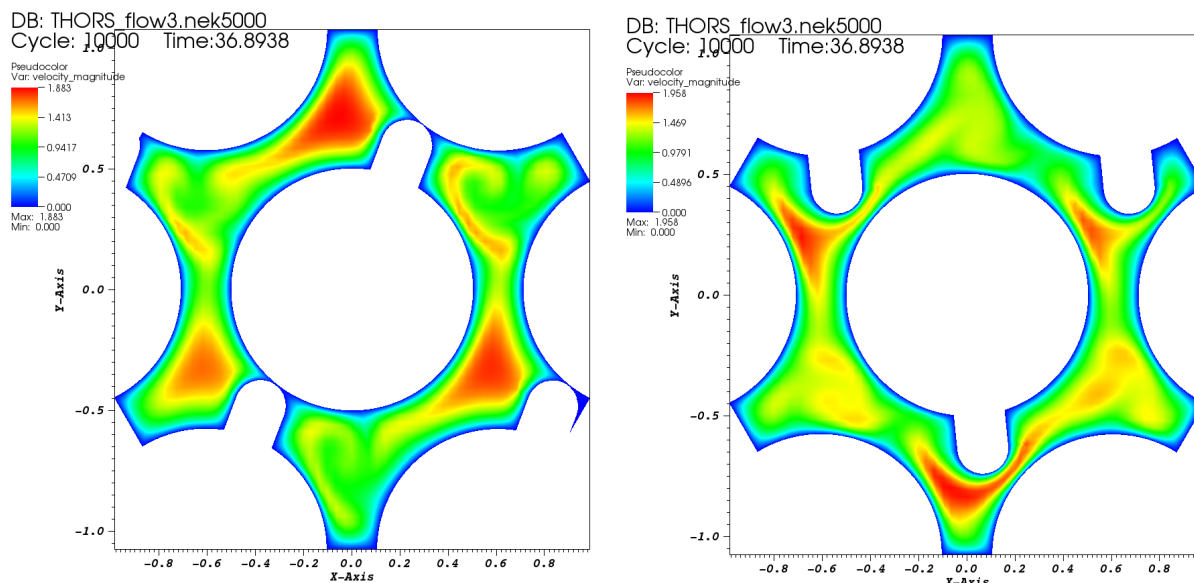


Figure 23. Single pin wire wrap simulation, wall boundaries on the periodic faces

7.2 7-PIN BUNDLE SIMULATIONS

This section presents numerical results of a turbulent flow in a 7-pin THORS bundle obtained with Nek5000 for a Reynolds number of 160 k. The long-term objective is to numerically predict the pressure drop for the THORS geometry and compare the numerical results against experimental data and correlation-computed data. This report only assesses the capabilities of the Nek5000 CFD code to produce a converged solution for a Reynolds number of 160 k using the mesh presented in Section 4.2.

The procedure to obtain a converged pressure-velocity solution with Nek5000 at high Reynolds number requires multiple steps. It is advised to run Nek5000 at a low Reynolds number first, say 10 k, obtained a converged solution, and use the converged solution at a restart file for the next run with a higher Reynolds number. We followed this procedure to obtain a converged pressure-velocity solution in a 7-pin bundle mesh with a Reynolds number of 160 k. We successively run Nek5000 with Reynolds number of 10, 100 and 160 k, each time using the output file of the previous run as a restart file for the next run.

The code was run for 200 seconds with a CFL of 0.5 on 40 nodes of 32 processors each. The Pn-Pn formulation was used with fourth and sixth polynomial orders and periodic boundary conditions. A spatial average velocity of 1 is set at the bottom ($z=0$) of the geometry. Plots of the time average velocity

magnitude are given in Figure 24 to Figure 27 for different position along the z-axis and for polynomial order $lx1 = 4$ and 6.

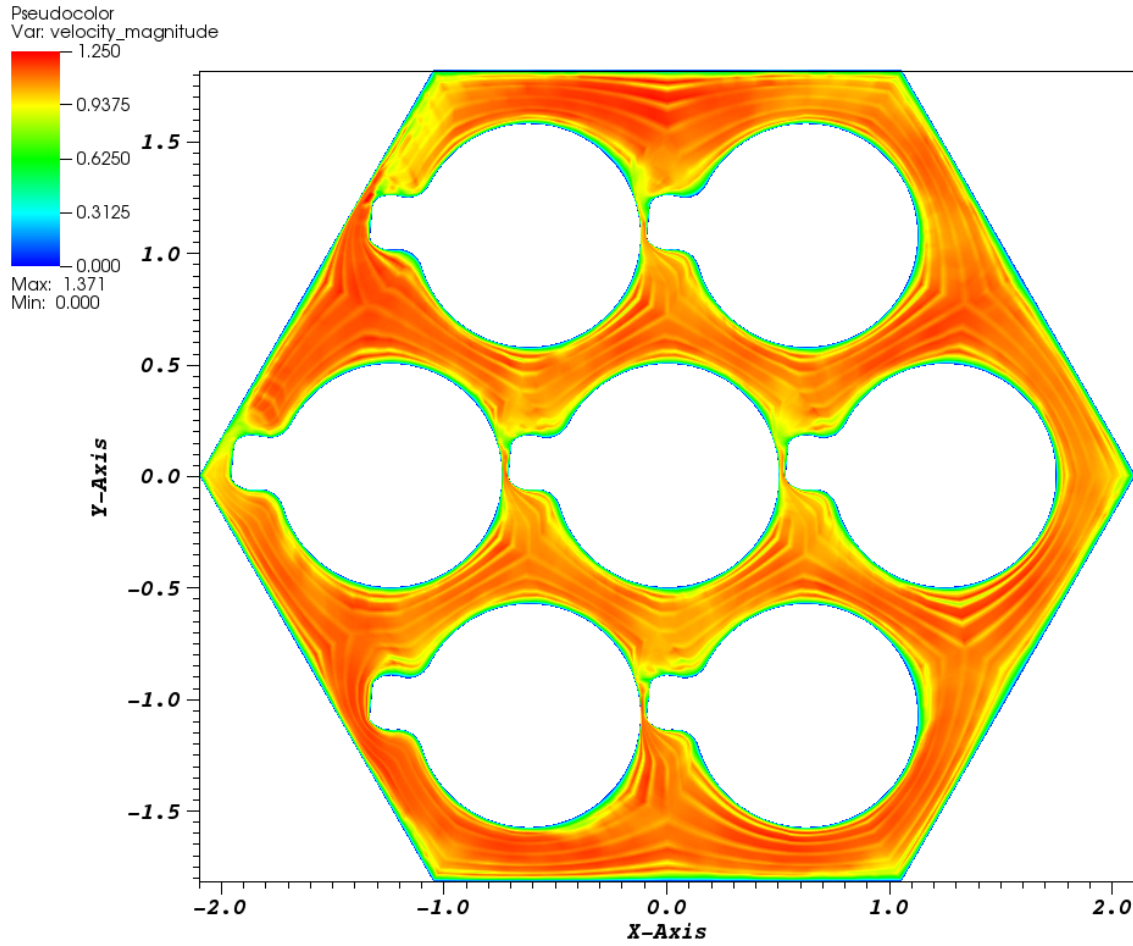


Figure 24 Time average velocity magnitude at $z = 0$ (bottom) simulated with $lx1=4$.

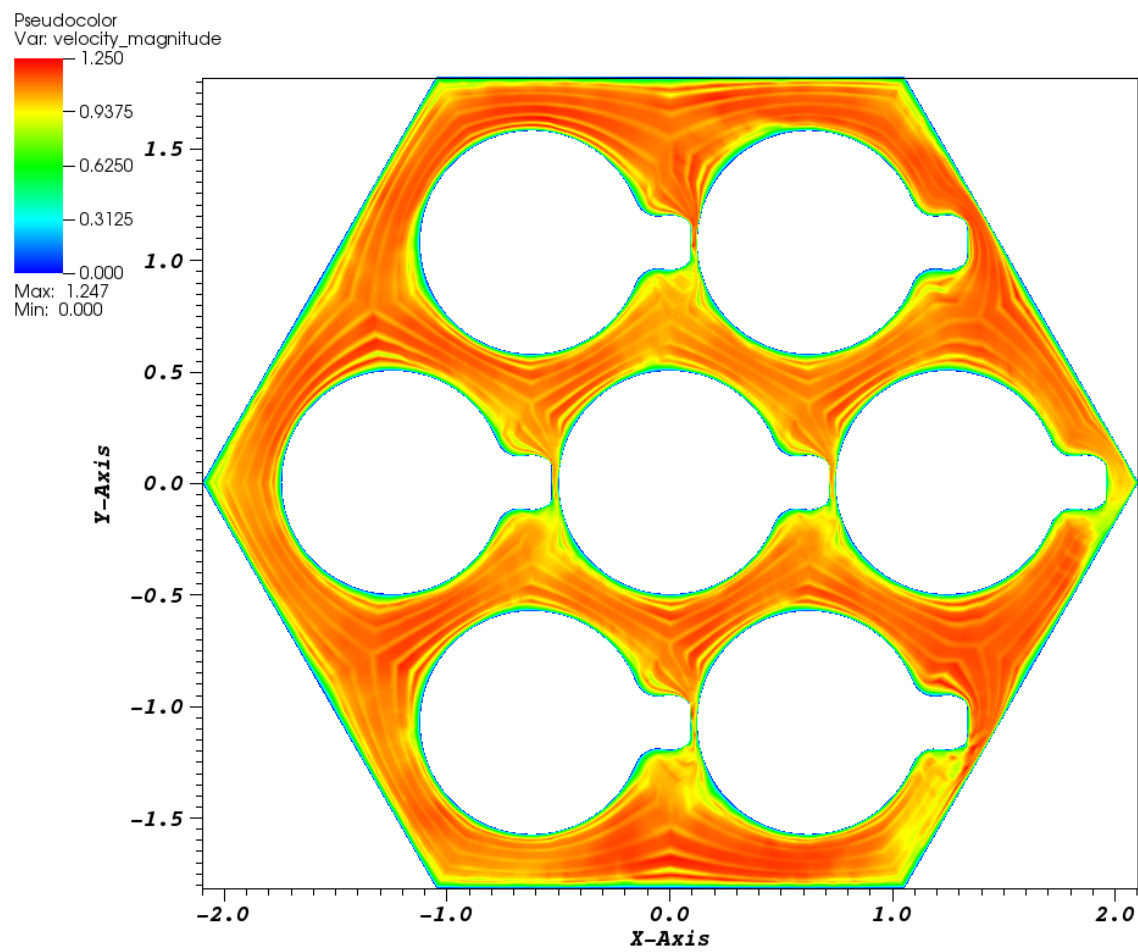


Figure 25 Time average velocity magnitude at $z = 25$ (middle) simulated with $lx1=4$.

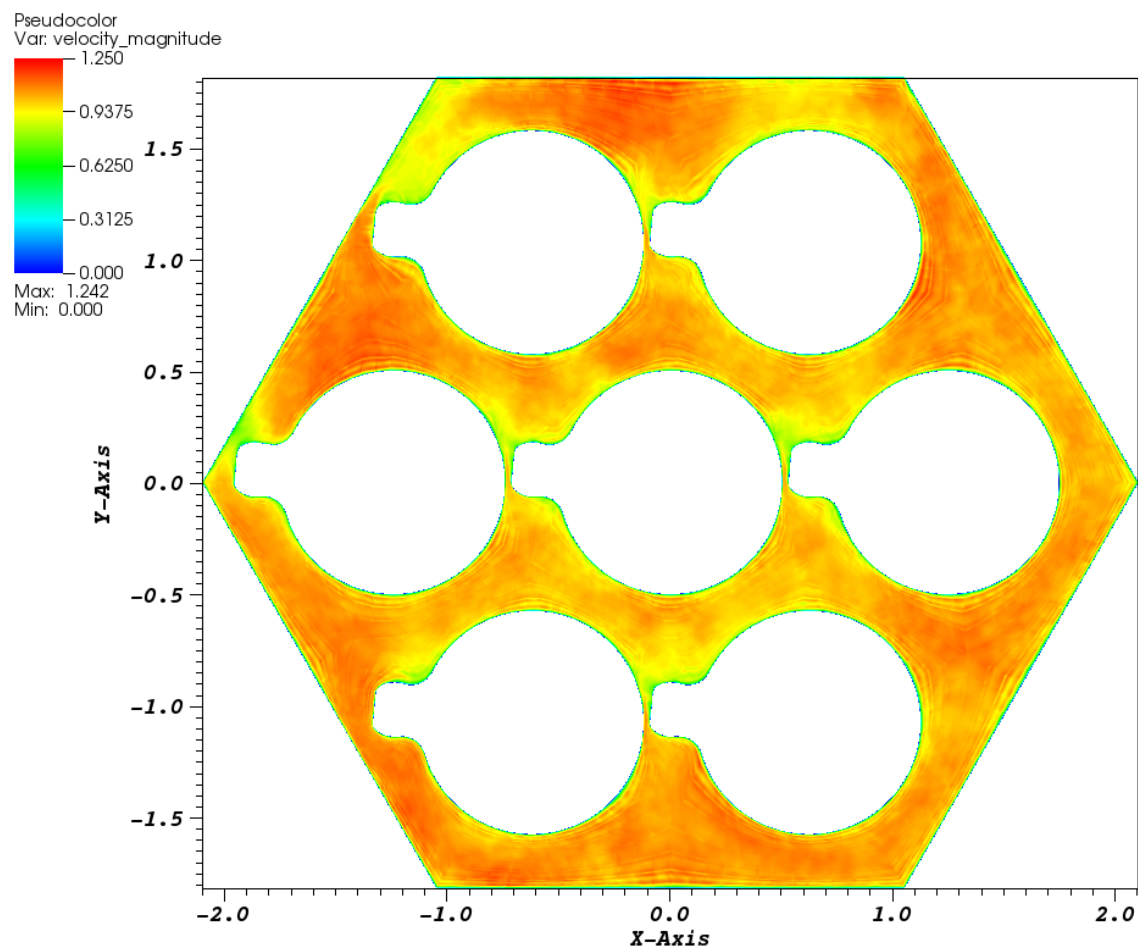


Figure 26 Time average velocity magnitude at $z = 0$ (bottom) simulated with $lx1=4$.

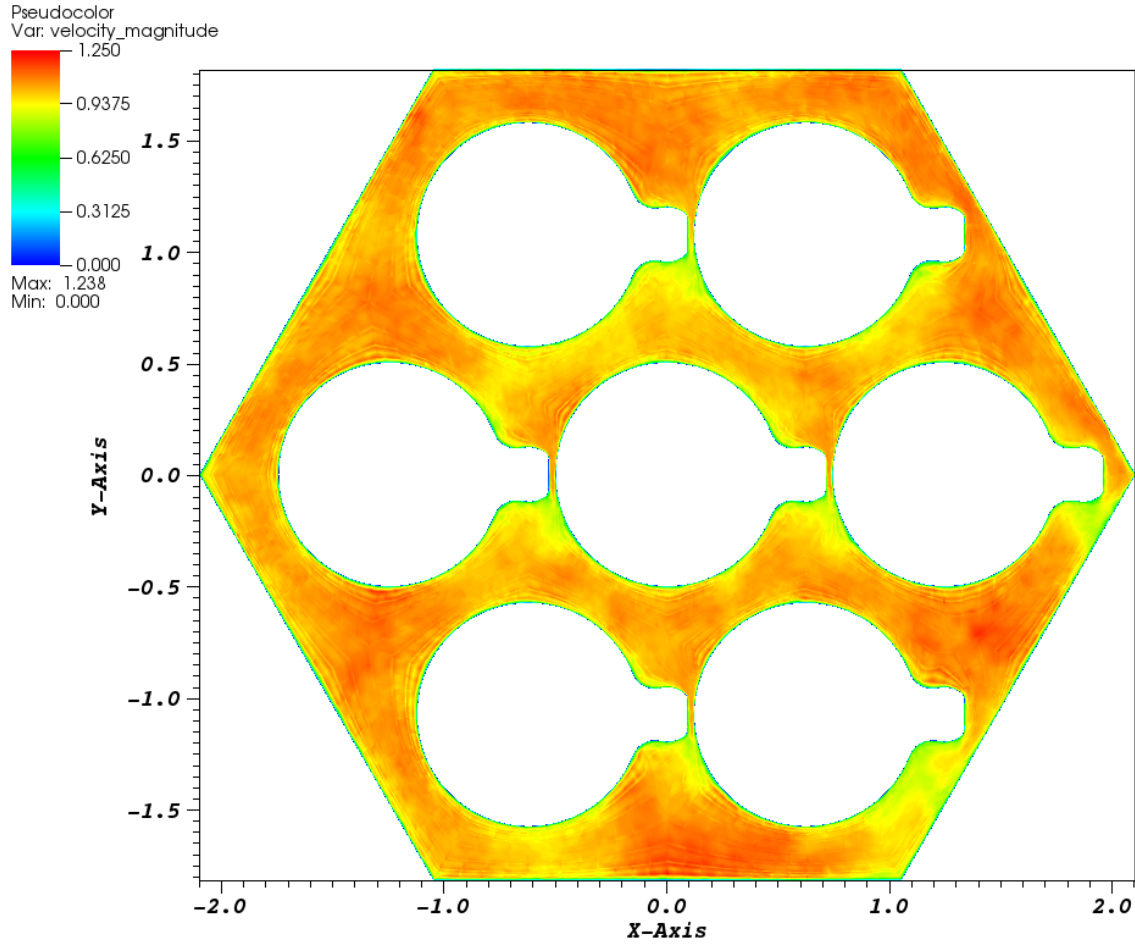


Figure 27 Time average velocity magnitude at $z = 0$ (bottom) simulated with $lx1=6$.

From Figure 24 to Figure 27, it is observed that the time average velocity magnitude is better resolved when increasing the polynomial order, i.e. the number of degree of freedom. The numerical solution obtained with $lx1 = 4$ shows a mesh dependency that seems to be attenuated when increasing $lx1$ to 6. We can still notice some mesh effects on the numerical solution next to the wall boundaries.

This behavior is somehow expected as the mesh resolution needs to be increased with the Reynolds number to capture the flow features. In the future, we are planning on running the same simulation with a polynomial order high enough to make the solution mesh independent.

8. CONCLUSIONS

This report demonstrates a coupling of Dakota with CFD codes by presenting various sensitivity analyses for turbulent flows in a 3-D pipe and in a single pin mesh. Using Dakota and Nek5000, a calibration study and sensitivity analyses for a turbulent flow was performed in a 3-D pipe by computing the L2 error norm between the experimental and numerical data. Nek5000 demonstrates that good accuracy can be achieved when the simulation is compared to experimental data. An analysis of sensitivity to the mesh and to three input parameters was performed, namely the polynomial order $lx1$, the filtering parameters $p101$, and $p103$. Results indicate that the polynomial order has a dominant effect on the L2 error norm. The L2 error norm decreases as the polynomial order, i.e. the number of degree of freedom increases. The filtering parameters $p101$ and $p103$ do not significantly affect the L2 error norm. On the other hand, the mesh type, curvilinear vs. linear, and the mesh density, affect the L2 error norm. When using a curvilinear mesh, the obtained L2 error norm is lower than a linear mesh for a given set of other input parameters ($p101$, $p103$, and $lx1$). It was observed that the mesh density has a more complex uncertainty pattern that will require further investigation to draw final conclusions.

A sensitivity analysis to three turbulent models (Realizable and VF2 $k-\epsilon$, and SST $k-\omega$) was also conducted for a turbulent flow in a single wire wrapped pin (THOR) geometry. After generating the mesh using the HEXPRESS software, the pressure drop was simulated with the STAR-CCM+ CFD commercial code. The numerically predicted pressure drop with the Realizable $k-\epsilon$, V2F $k-\epsilon$ and SST $k-\omega$ turbulent models were in good agreement with the pressure drops obtained by the Todreas correlation (TODREAS 1986) for all Reynolds numbers. On the other hand, the pressure drops computed with Novendstern model under-predicts the numerical pressure drops obtained with STARCCM+. We also observed that the experimental and predicted (numerically and with correlations) pressure drops only match for low Reynolds numbers, i.e. $Re < 1e5$, within a reasonable uncertainty. The differences could be due to either the inaccuracy of the correlations in the range of Reynolds numbers of interest or the quality of the THORS mesh used in the STARCCM+ simulations. They need to be investigated further.

In the final part of this project, simulations of a single-pin and 7-pin bundle were performed using Nek5000. The objective of these simulations was to demonstrate that it is possible to use Nek5000 to mesh, set up, and run a practical problem.

As a path forward, further simulations will be run to improve the statistics on the mesh density 832. Sensitivity analysis to other input parameters related to the numerical methods will also be performed, for example. The work conducted on the single- and 7-pin bundles meshed for UQ and sensitivity analysis can be leveraged to perform work similar to that conducted for the 3-D pipe.

9. BIBLIOGRAPHY

- Adams, B M, et al. *Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.0 User's Manual*. Sandia National Laboratory, Albuquerque: Sandia Technical Report , 2014.
- Carbajo J., Popov E., Pointer D. *Review of Legacy THORS Fuel Bundle Experiments as Validation Basis for NEK5000 Simulations of Wire-Wrapped Sodium-Cooled Fast Reactor (SFR)* . ORNL-TM, 2016.
- CD-adapco. *STAR-CCM+*. January 01, 2000. <http://www.cd-adapco.com/products/star-ccm@>.
- Chun, Moon-Hyum, and Kyong-Won Seo. "An experimental study and assessment of existing friction factor correlations for wire-wrapped fuel assemblies." *Annals of Nuclear Energy* 28 (2001): 1683-1695.
- Delchini, Marc-Olivier, Emilian Popov, and David Pointer. *Dakota Uncertainty Quantification Methods Applied to the CFD code Nek5000*. Oak Ridge National Laboratory, Oak Ridge: Oak Ridge National Laboratory, 2016.
- Fisher, Paul F, James W Lottes, and Stefan G Kerkermeier. *nek5000 Web page*. January 01, 2008. <http://nek5000.mcs.anl.gov>.
- Novendstern, E.H. "Turbulent flow pressure drop model for fuel rod assemblies utilizing a helical wire-wrap spacer system." *Nucl. Eng. Des.* 22 (1972): 19-27.
- TODREAS, Shih-Kuei CHENG and Neil E. "HYDRODYNAMIC MODELS AND CORRELATIONS FOR BARE AND WIRE-WRAPPED." *Nuclear Engineering and Design* 92 (1986) 227-251 (Nuclear Engineering and Design 92 (1986) 227-251), 1986.