

Nek5000 Tutorial

Simulation of a Simple Pipe with Cubit
and MOAB

Holler, David M.



Table of Contents

Introduction.....	1
Geometry.....	1
Meshing.....	2
Conversion with MOAB.....	3
Nek5000 Simulation.....	4
Summary.....	7
Acknowledgements.....	7
Appendix - Journal File.....	8

Introduction

This tutorial provides a step-by-step workflow for simulating laminar flow in a pipe with Nek5000 and the Mesh Oriented Database (MOAB). The meshing utility Cubit is used graphically, and a journal file containing the equivalent keyboard commands is also included. The requirements for this tutorial are:

- Understanding of basic meshing procedures (geometry > mesh > B.C.s > export)
- Basic Linux knowledge (path variables, terminal commands)
- Installation of Cubit GUI, any OS
- Configuration of MOAB somewhere that allows conversion and partitioning
- Installation of Nek5000 repository , preferably in a parallel configuration
- Successful run of the simple pipe MOAB example included in the Nek repository – this is to make sure the entire system is configured correctly and that the user knows how to use the Nek scripts
- Installation of VisIt for viewing the results

If this is the first attempt at a Nek/MOAB problem on a machine that hasn't been verified to work with Nek/MOAB, then errors may occur.

Geometry

After starting up Cubit, select the Geometry icon in the Command Panel, followed by the Manage Volumes icon. Select "Cylinder" from the drop down menu. The height option is also known as length, which is set to 5 in this example, along with a radius of .125. Click "Apply". This will make a 5 meter-long, 25 cm-diameter cylinder. Note that Cubit creates labeled surfaces and volumes automatically. Figure 1 shows the resultant geometry.

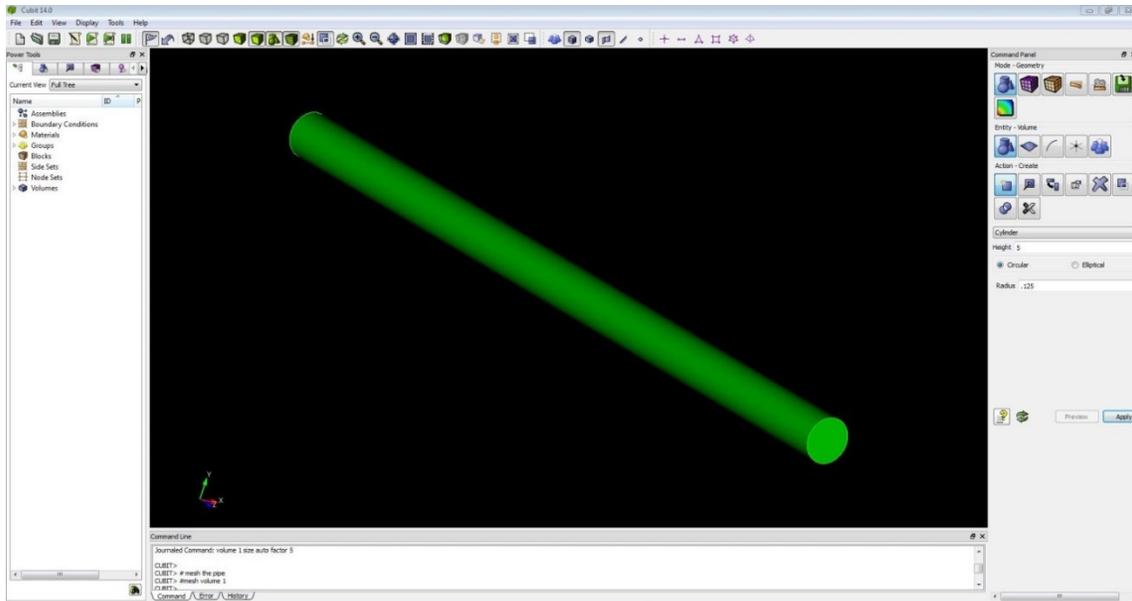


Figure 1. Finished geometry in Cubit.

Meshing

Meshing with Cubit can be easy, however trickier when dealing with more complex geometries. For this cylinder, the meshing process is short. The journal file (.jou) included at the end of this tutorial will create the entire mesh (except export) in less than 30 seconds. The majority of this section describes the (mostly) graphical method of creating the desired mesh.

In the Command Panel, select Mesh and then Manage Volumes. For this example, the Intervals option will be used. Select the cylinder in the viewing window and leave the drop down list to “Auto”. The other options can be useful, but they are beyond the scope of this exercise. Leave the Auto Factor in the center between Fine and Coarse, and click Apply. This will create quite a coarse mesh, especially when you consider the physical size of the pipe. Any time you need to change the size factor (auto factor) you have to click Apply, or else the changes will not be applied to the volume. Once the aforementioned options are applied, click on “Mesh” and Cubit will mesh the cylinder. There should be approximately 2000 elements.

After the mesh is successfully created, the boundary conditions must be applied. For use with Nek5000, the actual B.C. ID numbers in Cubit are not used. Only “sidesets” are used; Nek5000 doesn’t use any physical B.C. conditions set by Cubit.

The mesh created has three surfaces:

- 1) one of the circular faces
- 2) the cylinder wall

3) the other circular face.

The surface numbering may differ, so be sure to know which surfaces are which before creating the sidesets. To create Sideset no. 100 on Surface 1, select the Exodus mode from the Command Panel, then Sideset and Manage. Choose "Add" from the drop-down menu and create the sideset. Repeat this action for Surfaces 2 & 3. The boundary conditions in Cubit are now complete.

The last step needed in Cubit before export is to designate the mesh elements as HEX27 elements. These elements are needed in order for Nek5000 to use the mesh. To convert the elements, select the Block option (next to the Sideset option under "Entity" in the Command Panel) and choose "Add" from the drop-down menu. Type in 1 in the Block ID box and select "Volume" in the options below. Select Volume 1 and click Apply. Volume 1 should now be a part of Block 1. Now the block can be converted to HEX27 elements. The easiest way to do this is to use the command line, where everything can be done once a user is familiar with the syntax. In the command line, type:

```
block 1 element type hex27
```

This will convert all elements within block 1 (volume 1) to a type that can be used by Nek5000. This method is especially useful when dealing with multiple volumes that may be difficult to see in the viewing window or select with a mouse.

To export this mesh, use File > Export. Select "Exodus (*.e* *.exo*)" from the drop-down menu, and type the following into the File Name box (substituting a unique filename):

```
<filename>.exo
```

When prompted with extra options, select 3d and type in the blocks that need exporting into the Block ID(s) box.

The ".exo" file extension is needed or else Cubit will export the mesh as a ".e" file by default. The latter format is not usable with MOAB and Nek5000. The final mesh before export can be seen in Figure 2.

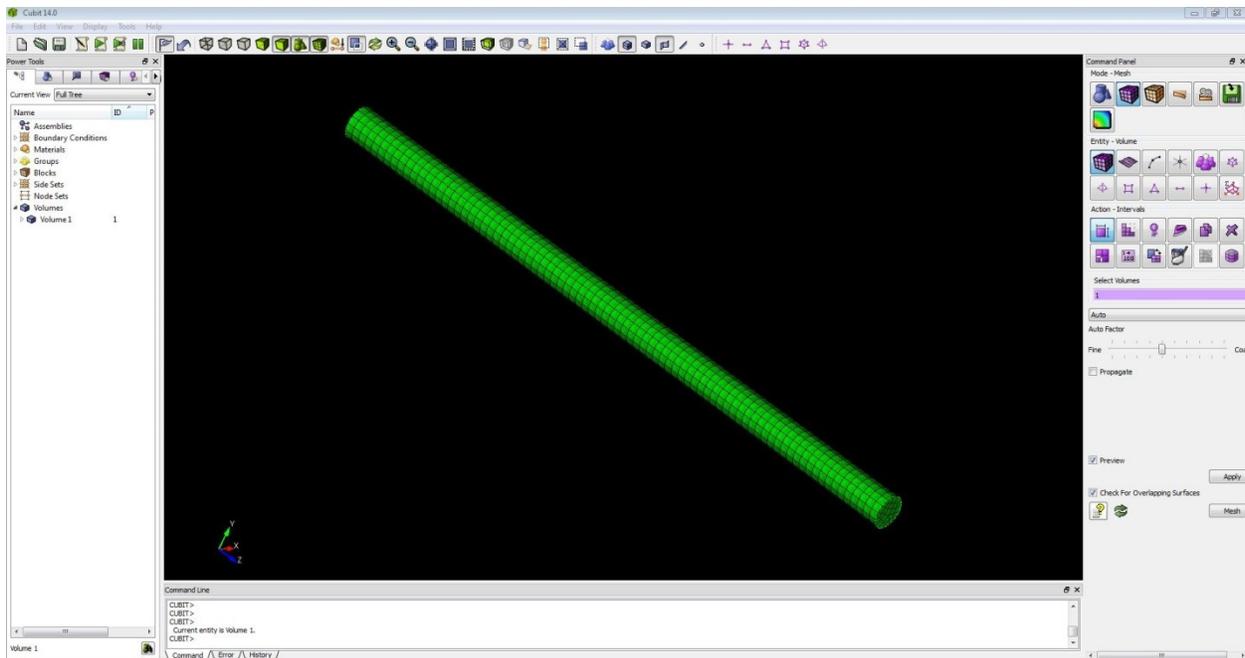


Figure 2. Finished HEX27 mesh in Cubit.

Conversion with MOAB

Now that a convertible file is present, load or move this file (if necessary) to the place where MOAB is configured to convert and partition and have a terminal window open. The tools “mbconvert” and “mbpart” should already be in a /bin folder on your workstation, with a path variable set to find that folder. With the pipe.exe file in the current terminal directory, issue the command:

```
mbconvert pipe.exe pipe.h5m
```

This will create a mesh file that can be read by Nek5000. This mesh is now ready for serial applications. If running on parallel, partition the mesh into however many pieces are needed. The syntax is:

```
mbpart <NP> <input file name> <output file name>
```

“NP” is the number of processors desired, which must be consistent throughout this setup process. The next step is to setup the Nek5000 files for a simulation.

Nek5000 Simulation

The basic workflow of a Nek simulation is as follows:

- 1) Compile the problem with the *makenek* script
- 2) Execute *nek*, *nekb*, or *nekbmpi* to solve the problem

- 3) Execute *visnek* to form a “.nek5000” database file
- 4) Load the database file into VisIt

The process with Cubit and MOAB requires the user to change parameters in the *makenek*, “.rea”, “.usr”, and “SIZE” files. They can be edited in any order, but all must be done before compiling. It’s recommended to copy the /examples/moab folder elsewhere and edit the files. As mentioned before, the best way to run original Nek/MOAB problems is to copy and use the directory in the provided MOAB examples.

The MAKENEK Script

For parallel use, there are three places in this script that need editing:

- 1) Pre-processor symbol list (line 16): insert MOAB in place of the question mark and un-comment the line.

```
# pre-processor symbol list
```

```
# (set PPLIST=? to get a list of available symbols)
```

```
PPLIST="MOAB"
```

- 2) Path to MOAB/iMESH (line 52): un-comment the line and point to your installation of MOAB.

```
# MOAB/iMESH path
```

```
MOAB_DIR="$HOME/install/moab"
```

- 3) Compiler links (line 54): choose the correct compiler and un-comment the line to make the link active.

For serial use, there are three additional changes that must be made to this file:

- Line 9, change mpif77 to the Fortran (77 or later) compiler installed
- Line 12, change mpicc to the C compiler installed
- Line 26, uncomment the line to turn off MPI

Keep in mind that there is a limit to the number of gridpoints / core that Nek can handle, and parallel computation is necessary for larger meshes.

The .rea file

This file contains property and mesh data, and for this tutorial, only a small section is focused on. The first thing to check is the logic switch at line 113, “IFMOAB”. This should’ve already been on true (T) if this file was copied from /examples/moab.

The next section, Mesh Data, needs the most changes. Start with line 117, which should already say:

```
input.h5m
```

This line should contain the name of the .h5m file that was converted and partitioned (if parallel) with mbconvert (and mbpart). Replace this line with the name of the new mesh file.

The next line, line 118, tells Nek how many sets (Cubit calls them blocks) of materials are present. Leave this line at one fluid set, zero solid sets. The other MOAB example, /examples/moab_conjht, simulates conjugate heat transfer with a fluid and a solid in one mesh. The .rea file in that example includes both a fluid set (block) and a solid set. Line 119 contains the Cubit ID given to the fluid set (block). Leave line 120 as is.

Line 121 should also stay the same, because there are 3 B.C. sets (sidesets) in both the Nek example and the pipe made in this tutorial. If more or fewer sidesets are present, then this number will need to be changed.

The final three lines that this tutorial is concerned with are the next three, which contain the Cubit ID, material property indicator, and boundary condition type. Check your Cubit file to see if the sidesets differ from the ones present, and change them accordingly. Leave the fluid property indicators, and confirm the B.C. types. Nek5000 has an assortment of boundary conditions available, but only the simple ones are used for this pipe.

The .usr file

The only thing that should be changed is line 164, which denotes the naming system of the output .h5m files. This change is not necessary for simulation, but may help in the way of data processing and organization. Be sure to only change the file name prefix inside the single quotations, and always leave the “.h5m” extension there as well. It’s recommended to also leave two zeroes at the end of the prefix too, because then Nek knows to use consecutive numbers to label the output .h5m files. The number of output .h5m files is equal to the number of processors that the Nek simulation was run on.

The SIZE file

Open the SIZE file, and locate the “lmt” variable (line 8, space 40). This variable controls the maximum number of elements per processor, and for

convenience, use the number of elements. The number of elements can be found in Cubit, or by typing the command

```
mbsize <filename>.h5m
```

in the same directory as <filename>.h5m and where MOAB is configured. Replace the number equal to the "lclt" variable with the number of elements in the mesh file, save the SIZE file, and exit. See the Nek5000 website for a complete description of the variables in this file.

Solving the Problem

Running the compilation and solver for MOAB problems is very similar to the preloaded examples, except the former tend to have larger output logs and the *makenek* script has a few different steps to take. Errors are more likely because of the larger pool of compilers and libraries being accessed during computation. Configuration errors are also likely, especially for new users. Keep in mind that extra .h5m files will also be in the output directory. Figure 3 shows the velocity magnitude solution of this pipe.

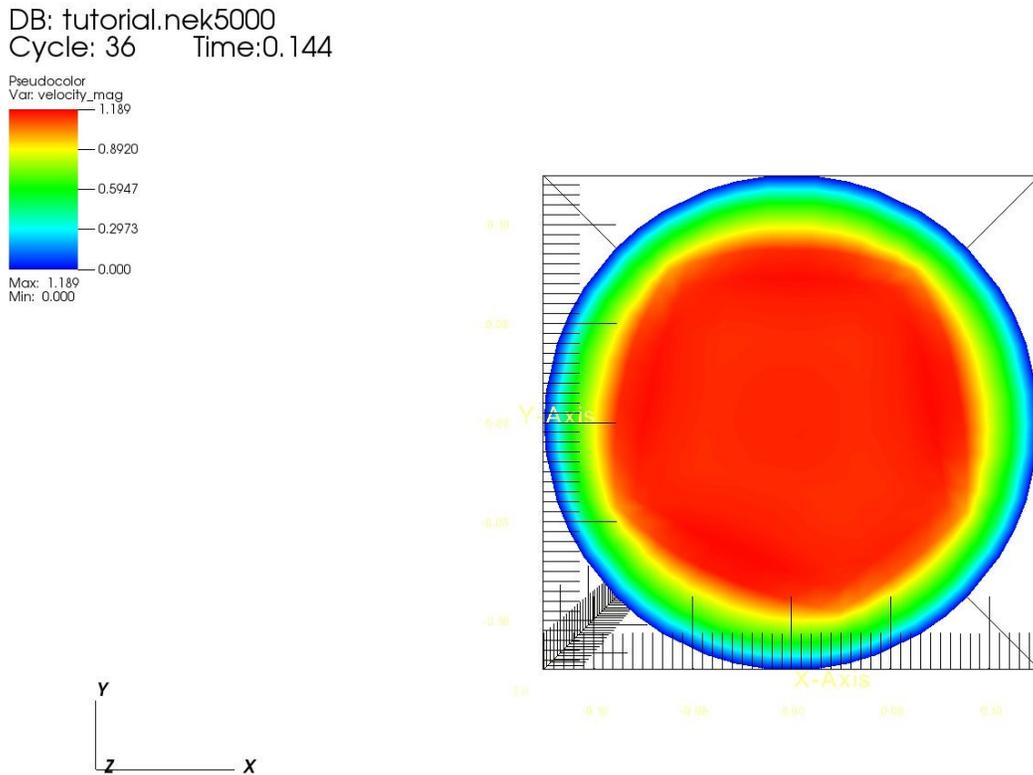


Figure 3. Velocity solution of tutorial problem.

Summary

In the process described earlier, a complete and working mesh for a simple pipe was created in Cubit for use with Nek5000. Any inquiries about detailed information in the Nek routine should be directed to the website.

Acknowledgements

This work was made possible with the help from multiple faculty members of Argonne's MCS division, along with support from NE and IT.

Appendix - Journal file

The following syntax will create the same file as described in this tutorial, except for the exportation and in a fraction of the time.

```
# start a new session
```

```
reset
```

```
# create cylinder
```

```
create cylinder height 5 radius 0.125
```

```
# apply automatic meshing scheme with a size factor of 5 (smaller  
size=finer mesh)
```

```
volume 1 size auto factor 5
```

```
# mesh the pipe
```

```
mesh volume 1
```

```
# make the three sidesets
```

```
sideset 100 surface 1
```

```
sideset 200 surface 2
```

```
sideset 300 surface 3
```

```
# designate all elements as HEX27, necessary for Nek5000
```

```
block 1 volume 1
```

```
block 1 element type hex27
```