True*Grid*®Output Manual For NEK5000

A Guide and a Reference

by

Robert Rainsberger

Version 4.0.0

XYZ Scientific Applications, Inc.

May 23, 2019

Table of Contents

I. NEK5000 Output Guide

NEK5000 is a three-dimensional simulation code designed to simulate laminar, transitional, and turbulent incompressible or low Mach-number flows with heat transfer and species transport. The focus in this manual will be on those features in **True*Grid*®** that are specific to creating NEK5000 input data. The **True*Grid*®** User's Manual covers the creation of a mesh and will not be covered in this manual. This manual is incomplete in another sense because it cannot be used as a substitute for a working knowledge of NEK5000.

Element types

The typical element is a quadratic or tri-quadratic hexahedron brick. The 20 node serendipity quadratic hexahedral element is initially specified using the **quadratic** command. This element is sufficient when generating edges with curvature along the arc of a circle. Alternatively, the 27 node tri-quadratic hexahedral element, specified initially using the **triquadratic** command, is more versatile. It has the added feature of creating edges with curvature on a sphere.

Commands to create a NEK5000 Input File

The following is a list of **True*Grid*®** commands that can be used to produce features for the NEK5000 input file.

NEK5000 feature	True*Grid*® commands
choose NEK5000 format	**nek5000** initially
choose 20 node hexahedral elements	**quadratic** initially
choose 27 node hexahedral elements	**triquadratic** initially
set parameters and switches	**nekopts** in the control phase
assign inlets	**il, ili** in the part phase
assign outlets	**ol, oli** in the part phase
assign periodic conditions	**bb** with the **periodf** option in the part phase
assign arc or spherical edges	**geop**, **geopi** in the part phase
set tolerance for quadratic edges	**nektol** before the **write** command
merge coincident nodes	**stp** before the **write** command
change the default name of output file	**mof** before the **write** command
create the input file for NEK5000	**write** in the merge phase

The **nek5000** command must be issued before the **write** command. The **write** command can only be issued in the merge phase. The default file name for the input file to Nek5000 is trugrdo. Use the **mof** command before issuing the **write** command to change this file name.

6   May 23, 2019   **True*Grid*®** Output Manual For NEK5000

II. NEK5000 EXAMPLE

This is a two part model of a 90 degree portion of an annulus. It is made in two parts so that a transitional region is automatically formed to increase the mesh density between the two parts. The wall formed at the inner radius is also identified with edges that are on arcs. The same is true for the face at the outer radius. The faces of the mesh along the x-axis (angle 0) and the faces of the mesh along the y-axis (angle 90) are paired as periodic boundary conditions. The bottom (z=0) is an inlet and the top (z=.5) is an outlet.

```
quadratic           c generate 20 node hexahedral elements
nek5000             c output file will be in the NEK5000 format
nektol 1.0e-04      c specify tolerance for non-linear edges

c start first block part with 9 by 12 by 12 elements
cylinder 1 10;1 13;1 13;.1 .5 0 90 0 .5

bb 2 1 1 2 2 2 1; c master side of transition block boundary

bb 1 1 1 2 1 2 2 periodf;       c master side of periodic bc
bb 1 2 1 2 2 2 2 periodf rz -90; c slave side of periodic bc

il 1 1 1 2 2 1              c inlet bc
ol 1 1 2 2 2 2             c outlet bc

geop 1 1 1 1 2 2 j arc      c non-linear edges on arcs

endpart                    c end of first part

c start second block part with 9 by 24 by 12 elements
cylinder 1 10;1 25;1 13;.5 1 0 90 0 .5

trbb 1 1 1 1 2 2 1;  c slave side of transitional block boundary

bb 1 1 1 2 1 2 3 periodf;       c master side of periodic bc
bb 1 2 1 2 2 2 3 periodf rz -90; c slave side of periodic bc

il 1 1 1 2 2 1              c inlet bc
ol 1 1 2 2 2 2             c outlet bc

geop 2 1 1 2 2 2 j arc      c non-linear edges on arcs

endpart                    c end of first part
```
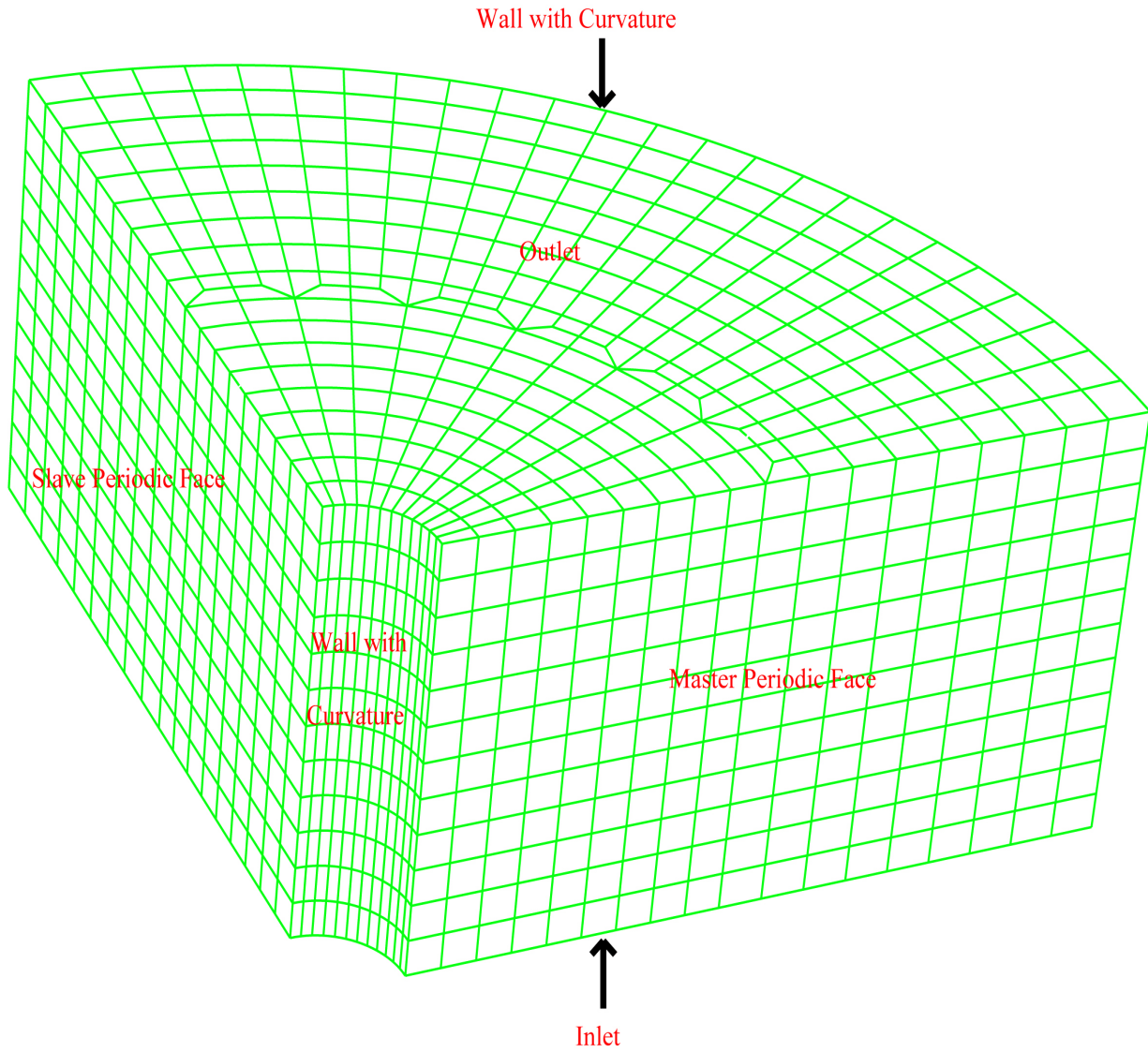
```
merge                              c enter the merge phase to output
stp .001                           c merge coincident nodes
write                              c write the NEK5000 file
```


Example

May 23, 2019   **True*Grid*®** Output Manual For NEK5000

III. NEK5000 OUTPUT REFERENCE

The syntax for commands are described below were literals are highlighted in **bold**. Symbols to be substituted are *italicized*. Each command is described by an entry like the following:

Command Syntax Conventions

When an arbitrarily long list of arguments are required, a semi-colon terminates the list. When a semi-colon is found in the description of an option or command, this indicates such a list. It is common to have a list inside another list. Each list must have a terminating semi-colon. This is analogous to parenthesis in algebraic expressions where the opening parenthesis must be balanced with a closing parenthesis. In this case, the keyword that initiates a list of items must be balanced with a closing semi-colon. Sometimes a short list of arguments and options can be repeated indefinitely, forming a list. The set of arguments and options that can be repeated are placed in square brackets. Sometimes the abbreviation *#_things* is used to mean "number of things". Each command is described by an entry like the following:

command   summary description

command *arguments*   brief description of functionality
   with brief descriptions of what the *arguments* should be.
      indentation is used to indicate a list of options to the *arguments*

Some commands in the part phase require a region specification. The region selects a face of the mesh, among other things. Others may require a progression specification. The progression selects multiple faces, among other things. In the merge phase, such commands require an option. In all of these cases, a portion of the mesh is identified.

Remarks

When present, the Remarks section describes the command in even greater detail. It may describe the context in which the command is normally used, and other commands used in association with this command. It may describe side effects. It may describe other, similar commands. In many cases, it includes a description of where to find the command in the menus.

Examples

When present, this shows the exact use of the command. If you use the dialogues, this command will be generated by simple selection options with the mouse and entering data where indicated. The command, as shown here, will appear in the session file for later reuse and possible modification.

You can also enter the command into the text window or insert it into a command file to be run in batch mode.

Only commands that are unique to the NEK5000 format are included below. All other commands are common to many applications and are included in the **True*Grid*®** User's Manual.

nek5000   NEK5000 format for the output file - control or merge phase

nek5000 (no arguments)

nektol   tolerance for non-linear edges

nektol *tolerance*

Remarks

This tolerance is used when midside-nodes are being determined. **True*Grid*®** always generates a midside-node for every edge in the mesh. However, only those midside-nodes that are not co-linear with the end nodes of the edge are considered for curved sides in the NEK5000 output file. When nektol is set to a value, then the midside-node must be at least the specified tolerance from being linear to be considered curved.

This is an absolute distance (not relative).

The default value is 1.0e-04.

Set this to zero if all edges are to be considered curved.

This can be sensitive. If a block has curvature on one side only, then there may be a threshold somewhere in the middle, due to interpolation of the mesh, where the edges transition from being curved to being (almost) linear, based on the value of nektol.

geop   geometry property by region - part phase

geop *region type property*
        where *type* can be
            i   to apply the geometric property to i-edges
            j   to apply the geometric property to j-edges
            k   to apply the geometric property to k-edges

where *property* can be

arc	each edge in the specified direction is on an arc of a circle
fsp	each edge of the face in the specified direction is on a sphere

Remarks

All element edges within the specified region will be identified in the NEK5000 output file as being on the type of geometry selected. If an edge is on an arc, the radius will be determined from the coordinates of the edge nodes. If an edge is on a sphere, the radius and center are determined from the edge and neighboring nodes. The edge does not have to be explicitly attached to an arc of a circle or projected to a cylinder or sphere. Care must be taken to be sure to shape these non-linear edges to achieve the desired arc or sphere calculation. Note that the **fsp** option requires that the region specified in this command is a face - not an single edge.

geopi   geometry property by progression - part phase

geopi *progression type property*

    where *type* can be

i	to apply the geometric property to i-edges
j	to apply the geometric property to j-edges
k	to apply the geometric property to k-edges

    where *property* can be

arc	each edge is on an arc of a circle
fsp	each edge of the face in the specified direction is on a sphere

Remarks

All element edges within the specified region will be identified in the NEK5000 output file as being on the type of geometry selected. If an edge is on an arc, the radius will be determined from the coordinates of the edge nodes. If an edge is on a sphere, the radius and center are determined from the edge and neighboring nodes. The edge does not have to be explicitly attached to an arc of a circle or projected to a cylinder or sphere. Care must be taken to be sure to shape these non-linear edges to achieve the desired arc or sphere calculation. Note that the **fsp** option requires that the index progression specified in this command select faces - not edges.

nekopts   set switches and assign values to parameters - control phase

nekopts [*parameter value*] ;

  where the *parameter* and *value* can be

ifflow **t** or **f** (default is true)
ifheat **t** or **f** (default is false)
iftran **t** or **f** (default is false)
ifnav1 **t** or **f** (default is false)
ifnav2 **t** or **f** (default is false)
ifnav3 **t** or **f** (default is false)
ifnav4 **t** or **f** (default is false)
ifnav5 **t** or **f** (default is false)
ifnav6 **t** or **f** (default is false)
ifnav7 **t** or **f** (default is false)
ifnav8 **t** or **f** (default is false)
ifnav9 **t** or **f** (default is false)
ifnav10 **t** or **f** (default is false)
ifnav11 **t** or **f** (default is false)
iftmsh1 **t** or **f** (default is false)
iftmsh2 **t** or **f** (default is false)
iftmsh3 **t** or **f** (default is true)
iftmsh4 **t** or **f** (default is true)
iftmsh5 **t** or **f** (default is true)
iftmsh6 **t** or **f** (default is true)
iftmsh7 **t** or **f** (default is true)
iftmsh8 **t** or **f** (default is true)
iftmsh9 **t** or **f** (default is true)
iftmsh10 **t** or **f** (default is true)
iftmsh11 **t** or **f** (default is true)
iftmsh12 **t** or **f** (default is true)
ifaxis **t** or **f** (default is false)
ifstrs **t** or **f** (default is true)
iflomach **t** or **f** (default is false)
ifmgrid **t** or **f** (default is false)
ifmvbd **t** or **f** (default is false)
ifchar **t** or **f** (default is false)
ifsync **t** or **f** (default is false)
ifuservp **t** or **f** (default is false)
aneden *value*
anevis *value*
anebet *value*
anegth *value*
anepgr *value*
p06 *value*
anerho *value*
anecon *value*

p09 *value*
anefin *value*
anenst *value*
anedt *value*
aneioc *value*
aneiot *value*
aneios *value*
anepss *value*
aneaxi *value*
anegri *value*
aneint *value*
anenor *value*
anediv *value*
anehel *value*
anenps *value*
anetlr *value*
anetla *value*
anecou *value*
anetor *value*
anenab *value*
anemhd *value*
aneuse *value*
anenpe *value*
anenbe *value*
p33 *value*
p34 *value*
p35 *value*
anexmg *value*
anengr *value*
anenr2 *value*
anenr3 *value*
p40 *value*
p41 *value*
p42 *value*
p43 *value*
p44 *value*
p45 *value*
p46 *value*
p47 *value*
p48 *value*
p49 *value*

p50 *value*
p51 *value*
anehip *value*
p53 *value*
p54 *value*
p55 *value*
p56 *value*
p57 *value*
p58 *value*
p59 *value*
p60 *value*
p61 *value*
p62 *value*
anewds *value*
p64 *value*
p65 *value*
p66 *value*
p67 *value*
p68 *value*
p69 *value*
p70 *value*
p71 *value*
p72 *value*
p73 *value*
p74 *value*
p75 *value*
p76 *value*
p77 *value*
p78 *value*
p79 *value*
p80 *value*
p81 *value*
p82 *value*
p83 *value*
p84 *value*
p85 *value*
p86 *value*
p87 *value*
p88 *value*
p89 *value*
p90 *value*
p91 *value*

p92 *value*
p93 *value*
p94 *value*
p95 *value*
p96 *value*
p97 *value*
p98 *value*
p99 *value*
p100 *value*
p101 *value*
p102 *value*
p103 *value*
p104 *value*
p105 *value*
p106 *value*
p107 *value*
p108 *value*
p109 *value*
p110 *value*
p111 *value*
p112 *value*
p113 *value*
p114 *value*
p115 *value*
anenlx *value*
anenly *value*
anenlz *value*

Remarks

Only assign values to switches and parameters that are applicable. All the rest of the parameters will be set to zero.