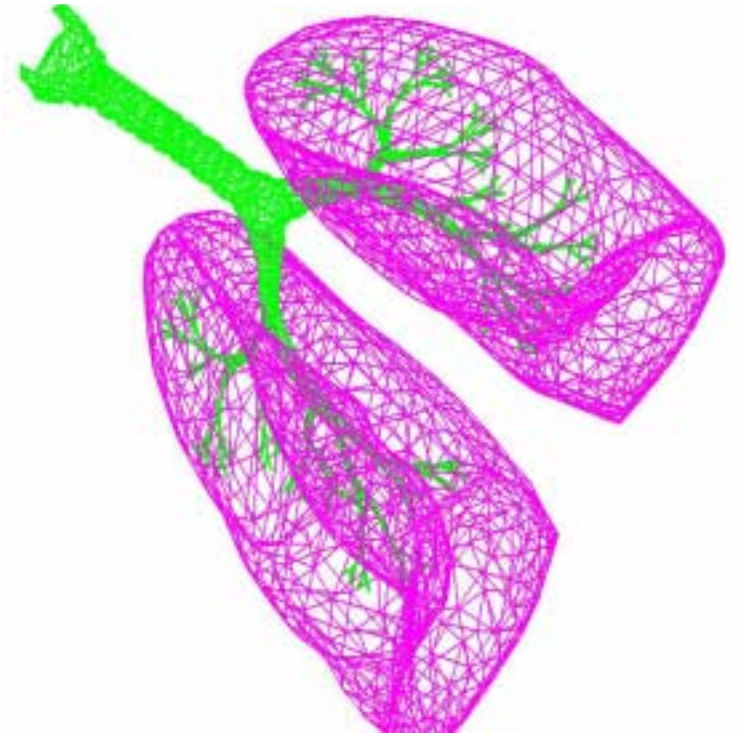
NWGrid
Users Manual
The *N*orth*W*est *Grid* Generation Code

Web Site: http://www.emsl.pnl.gov/nwgrid



Dr. Harold E. Trease
Lynn L. Trease
Applied Mathematics Group
Theory, Modeling, and Simulation Directorate
***William R. Wiley,* Environmental Molecular Sciences Laboratory**
Pacific Northwest National Laboratory
Operated by Battelle for the US Department of Energy

TABLE OF CONTENTS

1.0 Introduction

NWGrid is a hybrid n-dimensional grid generation system. The Applied Mathematics Group at Pacific Northwest National Laboratory (PNNL) has developed this tool support the Laboratory's computational science efforts in chemistry, biology, engineering and environmental (subsurface and atmospheric) modeling. ***NWGrid*** is the grid generation system, which is designed for multi-scale, multi-material, multi-physics, time-dependent, 3-D, hybrid grids that are either statically adapted or evolved in time. ***NWGrid's*** capabilities include static and dynamic grids, hybrid grids, managing colliding surfaces, and grid optimization [using reconnections, smoothing, and adaptive mesh refinement (AMR) algorithms]. ***NWGrid's*** data structure can manage an arbitrary number of grid objects, each with an arbitrary number of grid attributes. ***NWGrid*** uses surface geometry to build volumes by using combinations of Boolean operators and order relations. Point distributions can be input, generated using either ray shooting techniques or defined point-by-point. Connectivity matrices are then generated automatically for all variations of hybrid grids.

1.1 Overview

NWGrid is more than a grid generator. It is a setup code which incorporates arbitrary grid attributes, boundary conditions and initial conditions, in addition to the typical grid generation capability.

The grid generation system in ***NWGrid*** is composed of several grid generation algorithms, which work cooperatively to produce a final hybrid computational mesh. The grid generation algorithms that are currently built into ***NWGrid*** include a Voronoi mesh generator, a Delaunay mesh generator, an Adaptive Mesh Refinement (AMR) mesh generator, and a (block) structured mesh generator. In addition, we have reconnection algorithms, remapping algorithms, integrated translation algorithms, and merging algorithms to help blend meshes together. The set of geometric elements supported includes tetrahedra, pyramids, prisms, and hexahedra. The code also includes AMR octree elements (i.e., parents and children) and general polyhedral elements.

From a data structure standpoint, the entire code is a Scientific Relational DataBase Management System (SRDBMS) composed of five memory management layers. The highest layer is the Relational DataBase Management System (RDBMS); the next layer is the DataBase Management System; the next two lower levels include the Storage Block Memory Management System (SBMM) and the Memory Management System (MMS); and at the lowest level, everything is processed by the C memory management functions (malloc, realloc and free). The code system uses the **METIS** dynamic partitioning software for determining the weighted distribution of functional and data driven work to allocate the blocks of the partitioned grid to a set of parallel processors. **PYTHON** has been fully integrated into NWGRID as a backplane, and it acts as either the controlling code or as an embedding language processor. The SRDBMS allows for multiple Mesh Objects (MOs), of which one is called

the Current Mesh Object (CMO). MOs can be merged, appended, inserted, intersected, diffed, etc. to create a final mesh.

1.2 Capabilities

1.2.1 Geometry

- Surfaces and curves are input from CAD packages through STL files.
- Faceted surfaces composed of triangles and/or quadrilaterals can be used as geometry, such as STL files, AVS files, etc.
- Boolean operators and order relations (like greater than, less than, equal to, etc., as well as and, or, and not) are used to manipulate/combine surfaces and curves to define geometric regions and material regions that have arbitrary attributes associated with them.

1.2.2 Node Distribution

- Nodes can be placed and manipulated one-by-one if the user desires (and some want to do this).
- We typically use ray shooting or ray tracing from a source to a target to distribute nodes on surfaces and within volumes.
- Node distributions can be imported into the code in various formats.

1.2.3 Node Connectivity

- NWGrid has multiple search engines to create connectivity matrices for random distributions of nodes.
- Voronoi/Delaunay meshes for polyhedral, tetrahedral or triangular grids.
- Nested AMR grids for hexahedral or quadrilateral grids.
- Several point addition algorithms (assuming we have an existing background grid) can be used to add a feature in the form of an additional node distribution.
- Converting higher order elements into simplex figures. Like hexahedrons, prisms or pyramids to tetrahedrons or quadrilaterals to triangles.

1.2.4 Grid Object Manipulation

- NWgrid supports an arbitrary number of grid objects.
- Grid objects can be added, merged, intersected, etc.

- Data from one grid object can be copied into another.
- A new grid object can be derived from an existing one.
- An arbitrary number of grid attributes can be added and/or deleted at run time.
- Intersections of higher dimensional grid objects by lower dimensional objects produce lower dimensional grid objects, such as the intersection of a 3D grid with a 2D plane to produce a 2D hybrid grid.

2.0 Getting Started

2.1 Executing NWGrid

NWGrid can be run in a command-line, interactive mode or in a batch-mode.

2.1.1 Running in Command-Line, Interactive Mode

In command-line, each command can be entered one-by-one, an existing input file can be run, or several lines can be pasted in at once.

To execute ***NWGrid*** interactively, in a command-line mode, type:

machinename% **nwgrid** *[assumes the executable is aliased to nwgrid]*

The user will be prompted to enter each ***NWGrid*** command.

Enter a command

An input file can also be run by starting up ***NWGrid*** interactively and at the prompt entering:

Enter a command
infile < *input_filename*

If there is a **finish** command at the end of the input file, ***NWGrid*** will exit. If not, the above prompt will return.

2.1.2 Running in Batch-Mode

To execute ***NWGrid*** in batch-mode, with an existing ***NWGrid*** input file, type:

machinename% **nwgrid <** *input_filename*

2.2 Exiting *NWGrid*

The session is ended by typing the ***NWGrid*** command, **finish** at the prompt.

Enter a command
finish

2.3 *NWGrid* Input File

The input file for ***NWGrid*** is made up of commands (a complete listing is in the Command Reference Section), in an asci file. Following is the command syntax and conventions used in this manual for commands.

2.3.1 Command Syntax

- Commands begin with a keyword followed by parameters. Delimiters include comma, slash, equal sign, or blank. (',' '/' '=' ' '). Blanks on either side of other delimiters are ignored. Leading blanks are ignored. Commas are usually used for parameters that belong to the same logical set such as first point, last point, stride. Slashes are usually used to separate sets of parameters.

- Lines are a maximum of 80 characters long. To continue a line, put a “&” as the last character of the line to be continued. A command can be up to 1024 characters long.

- All names (**surface, region, pset**, etc.) should be limited to 32 characters.

- Commands usually are typed in all lower case. Names are case sensitive.

- The three parameters: first point, last point, stride can have integer values which refer to actual sequential point numbers or they can have the character-string values:

- **pset**, **get**, *name* where *name* has been defined by a previous **pset** command.

- The triplet 1, 0, 0 refers to all points.

- The triplet 0, 0, 0 refers to the set of points defined in the last geometry command

- Comments are identified by * in the first column. Comments are parsed; avoid using special characters especially `&' in comments.

- To separate commands on the same line use a semicolon (;).

2.3.2 Manual Conventions

NWGrid Users Manual conventions, for the command descriptions include:

- **Bold** is used for literal keywords such as **addmesh, cmo,** and **region** and for literal parameters such as **itetclr, lt, avs,** etc.

- Symbols Meanings:

  - [ ] surround optional parameters

  - | signifies alternate choices

  - , or / separates parameters

2.4 Help

2.4.1 On-line Help

Help can be obtained for any ***NWGrid*** command when working in the interactive mode by entering:

help *keyword*

An example session, showing how to get help for any ***NWGrid*** command follows:

```
machinename% nwgrid
*       * * * * * * * * * * * * * * * * * * * * * * * * *
*       *                                               *
*       *    Program:  NWGrid                           *
*       *    Version:  0.069                            *
*       *    Compiled: 97/02/04 at 10:46:52             *
*       *                                               *
*       *                                               *
*       *    This run was processed                     *
*       *     on day:  Fri Sep 24 1999                  *
*       *     at time: 13:44:37                         *
*       *                                               *
*       * * * * * * * * * * * * * * * * * * * * * * * * *

Output log file: outx3dgen
Command log file: logx3dgen

Using the LOCAL dictionary: x3ddict

Enter a command
      help rzs
            1  help rzs

The current code name is:  generate

        Builds a sphere by generating coordinates of points and also
           modifies zoning by ratio-zoning point distributions. See
           the RZ command for more details.  The ITYPE flag defines what
           type of sphere will be generated.
         ITYPE=1 generates a sphere by gridding the faces of a cube and
           then projecting the vertex onto a sphere.
         ITYPE=2 generates a sphere by subdividing an icosahedron placed
           on the surface of a sphere (this algorithm was suggested by
           John Baumgardner and Paul Fredrickson.
         ITYPE=5 produces a cubic  hexahedral zoning.
         ITYPE=6 produces a triangluar hexahedral zoning.
         ITYPE=7 produces a octahedron hexahedral zoning.
         ITYPE=8 produces a icosahedron hexahedral zoning.

         FORMAT:  RZS/itype=1/# of radii/# of points per radius/
          (cont)      inner radius, outer radius/ xcen,ycen,zcen /
          (cont)      absolute(reflective) zoning flag /
          (cont)      ratio zoning flag / ratio zoning factor
```

2.4.2 Web-Based Help

There is a Command Index on the ***NWGrid/NWPhys*** internal web site that gives the description and complete format for each ***NWGrid*** command.

http://www.emsl.pnl.gov/nwgrid

The user documentation listed in the Documentation section of this document is also available internally at this site.

2.5 Log Files

NWGrid will produce two files, outx3dgen and logx3dgen. These contain detailed output information and the list of commands respectively.

3.0 Visualization Tools

3.1 General Mesh Viewer (GMV)

The recommended visualization tool is GMV. It is a 3-D visualization tool designed to view simulation data from any type of structured or unstructured mesh. A User's Manual can be found under:

http://laws.lanl.gov/XCM/gmv/GMVHome.html

The executable can be found under:

For Linux (Mesa and OGL versions):

/msrc/proj/nwgrid/bin/gmv_linuxMesa
/msrc/proj/nwgrid/bin/gmv_linuxogl

For SGI (Mesa and OGL versions):

/msrc/proj/nwgrid/bin/gmv_sgiMesa
/msrc/proj/nwgrid/bin/gmv

3.2 Oso

One recommended way to create a geometry is by using Oso. It is an interactive tool that lets you visually see what you are creating, as you create it. A User's Manual can be found under:

http://www-xdiv.lanl.gov/x8/oso

The executables can be found under:

For Linux:

/msrc/proj/nwgrid/bin/osolinux

For SGI:

/msrc/proj/nwgrid/bin/ososgi

4.0 Data Structures

The data structures that ***NWGrid*** uses are based on grid objects. We can have any number of grid objects that are active at any one time. Each grid object is user extensible to include whatever attributes a user needs to add (or delete). Grid objects

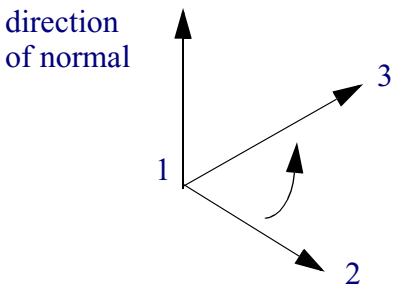
can be operated on to perform functions like adding, merging, intersection, etc. Basically, ***NWGrid*** is a combination of a language driver (either Fortran, C, or C++) coupled to the ***NWGrid*** grid generation toolbox, which uses an Object DataBase Management System (ODBMS). The ODBMS that ***NWGrid*** uses is a hierarchy of a RDBMS, a Common Array System, a SBMS, and a MMS. All of these are used to manage the run-time definable data structures that define a grid object. The collection of grid objects is managed as a collection of Mesh Objects (MOs), where the selected MO is called the Current Mesh Object (CMO). The set of grid object operators that we use are compress, copy, create, derive, list, move, read, delete, select, status, verify, and write.

The default Mesh Object is named *3dmesh.* For simple problems the user must supply only a **cmo/create/**mesh_object_name command. There is no limit on the number of Mesh Objects that can be defined, but at any time there is only one 'current' or 'active' Mesh Object. For more advanced problems, such as those requiring more than one Mesh Object or requiring extensions to the basic Mesh Object template, the Mesh Object(s) is(are) manipulated via the **cmo** commands which are described in the core commands section. For example, additional user defined attributes may be added to a Mesh Object by using the **cmo/addatt** command, or the 'active' Mesh Object can be changed using the **cmo/select** command.

For more information on mesh objects see the ***NWGrid*** Data Structures Reference.

5.0 Coordinate System

The right hand rule is used to determine normals to planes and to sheet surfaces. The first two points determine the first vector and the first and third point determine the second vector. By curling the fingers of the right hand from the first vector toward the second vector, the right thumb will point in the direction of the normal.



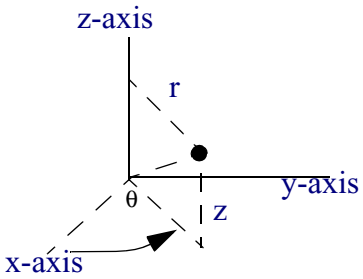
Three coordinate systems are used.

5.1 Cartesian Coordinates

xyz refers to the standard Cartesian coordinate system.

5.2 Cylindrical Coordinates

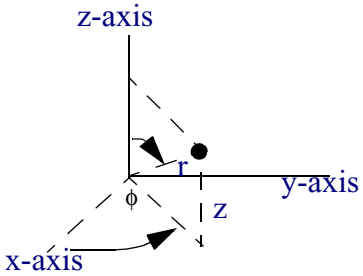
rtz refers to a cylindrical coordinate system aligned along the z-axis, where r is the radius measured from the zaxis, t (theta) is the angle measured in the xy-plane from the positive x-axis toward the positive y-axis and z is the height measured from the xy-plane.



position of ● determined by:
r radius
t theta (angle from x-axis)
z height

5.3 Spherical Coordinates

rtp refers to a spherical coordinate system, where r is the radius measured from the origin, p (phi) is the angle in the xy-plane measured from the positive x-axis toward the positive y-axis, t (theta) is the angle measured from the positive z-axis to the positive y axis.



position of ● determinted by:
r radius measured from origin
t theta (measured from positive z-axis)
p phi (measured from positive x-axis)

6.0 Command Descriptions

NWGrid has over 120 commands, with an infinite number of sequences and parameters. Following is a brief description of the core set of ***NWGrid*** commands and of all the other commands. The complete command format, parameters, and examples can be found in the ***NWGrid*** Command Reference.

6.1 Core Set of Commands

Following are the core set of NWGrid commands needed to run most problems.

Brief Description of Core Set of Commands

Core Commands	Description

Brief Description of Core Set of Commands(Continued)

addmesh	Joins two meshes together at their common interface to produce a third mesh
check_amr_level	Check the consistency of AMR levels.
cmo	The current mesh object (cmo) operations.
doping	Maps node/element attributes between CMOs.
dump	Writes out the ***NWGrid*** data to a file of the specified format.
eset	Defines (or overwrites) a specified set of elements.
extract	Slices a 3D mesh object to obtain a 2D mesh object.
geniee	Create the element connectivity list (jtet).
gtg	Grid-to-grid mapping function.
filter	Filter (delete) points that are closer than the distance specified by the user or duplicate points.
finish	Stops the code so that is can terminate gracefully.
help	Activates the help package for the current code.
hextotet	Create a tet grid from a hex grid or a triangle grid from a guad grid.
infile	Read input commands from a file.
mapgrid	Maps node/element attributes between CMOs.
modbld	Defines model parameters for material regions.
modchk	Checks the consistency of the model parameters.
modval	Evaluates the defined models.
mregion	Define a material region from the set of surface command definitions by logically combining the surface names.
partition	Decomposes a mesh into N-parts for parallel computing.
pset	Give a name to a selected set of points.
read	Reads in a file of the specified format.
refine	Refine a mesh by adding point on edges, center of faces or center of elements.
region	Define a geometric region from the set of surface command definitions by logically combining the surface names.

Brief Description of Core Set of Commands(Continued)

regnpts	Generates points in a region previously defined by the region command.
rm	Removes any points that are within the specified point range and specified piece of space.
rmpoint	Removes a specified list of points from a point distribution.
rotateln	Rotates a point distribution about a line.
rotatept	Rotates a point distribution about a point
rz	Builds the geometry by generating coordinates of points and also modifies zoning by ratio-zoning point distributions.
rzbrick	Builds a brick mesh and generates a nearest neighbor connectivity matrix.
rzs	Build a sphere by generating coordinates of points and modifies zoning by ratio-zoning point distributions.
rzst	Generates points on the surface of spheres or parts of spheres defined by the input parameters.
scale	Scales a point.
search	Generates the connectivity list.
setpts	Sets point types and material regions.
settets	Set element color, itetclr and create child points at interior interfaces. Can also be used to set the node color from the element color.
smooth	Smooths 2- or 3-D mesh objects. Adaptive smoothing (to values of specified fields) or non-adaptive smoothing is available.
surface	Defines a boundary surface of the type **free, intrface**, or **reflect**.
surfpts	Generates points on boundary surfaces previously defined by the surface or region command.
title	Used to set the problem title or to retrieve the problem title.
trans	Translates a selected set of points in x,y,z space by picking one specific point in the set of points and moving it to new coordinates with a linear translation.
zq	Sets (or prints) zone quantities.

6.2 Additional Commands

Following is a list of additional ***NWGrid*** commands that are used less frequently.

Brief Description of Additional Commands

assign	Assign a value to a code value.
box	Defines the six reflective boundaries that make up a box.
bricktyp	Defines tye type of point distribution to be used in the next part of the brick mesh generation. The default is no brick mesh.
coggeshl	Initialize one of the Coggeshell test problems.
conenb	Defines the reflective boundaries in r-, theta-, phi-geometry.
coordsys	Define a local coordinate system to be in effect until another coordinate system is defined or the normal coordinate system is reset.
copypts	Copy a point distribution.
cutreg	Selects an existing region to be used as a cutting surface in the plotreg command.
edit	Prints an edit of various quantities based on the value of the option argument, the point limits, and/or a material specification.
ether	Distribute boundary velocity in region or volume.
extrafl	Extracts a list of points contained in a region and changes the point types to be reflective boundary points.
field	Manipulates one or more specified fields in the Current Mesh Object.
flbound	Files the quantities associated with a reflective boundary.
interp	Used to interpolate to a given (logical) line, plane or volume.
intersect	Creates a new CMO from the intersection of two existing CMOs.
linkmaze	Provides a link to the MAZE generator of DYNA.
linksab	Provides a link to the SABRINA.

Brief Description of Additional Commands (Continued)

list	Prints a list of selected problem information.
log	Indicates where the messages to the terminal will be directed.
logadd	Add two logical blocks of coordinates.
mataddpt	Flags materials in which the user wishes to add mass points.
matchbdy	Join matching boundary faces together.
matrec	Adds materials to the material reconnection list.
memory	Estimates how much memory is needed to set up a problem, based on how many mass points will be used. Must be a GOOD guess.
macro	Defines a macro to be used by the MACROUSE command for the MOV IE command.
macrodef	Defines the name of a macro.
macroend	Terminates a macro.
macrouse	Executes a MACRO that has been defined with MACRO command.
merge	Merges two points together.
modbld	Sets up the MODELs for a particular problem (e.g., EOS, HE, etc.).
modchk	Checks the completeness and consistency of the MODELs that have been defined for a particular problem.
modhelp	Checks the MODELs for a particular problem (e.g., EOS, HE, etc.).
modpost	Prints a summary sheet for each model.
modprt	Prints a summary sheet for each model.
modset	Initializes the MODELs for a particular problem.
modval	An evaluation of all the PARTs of the current problem is done.
nb	Prints/sets the number of reflective boundaries for a region.
newdict	Replaces the current parameter dictionary values with the ones from the default parameter dictionary.

Brief Description of Additional Commands (Continued)

newmesh	Resets the generator so that another region can be defined.
nn	Prints information about the nearest neighbors of a given point.
nnblock	Prints the vector block summary for the tetrahedrons associated with one, two, or three mass points.
nnsurf	This is the main command for generating the connectivity list.
nntet	Describes all you could possibly want to know about a tetrahedron.
np	Prints a summary sheet on the point distribution for a region.
offsetsurf	Offsets a surface by normal surface motion by a distance dist.
parallel	Invokes a parallel section of inputfile commands.
perturb	Perturbs a point distribution according to SIN distributions.
plotreg	Saves, replaces, stores, or retrieves sets of pstatus points.
pregion	Saves, replaces, stores, or retrieves sets of **pstatus** points.
process	Assigns the processes and processor number that run this job.
pstatus	Saves, removes, retrieves, or replaces a specified set of points.
pyoungs	Perturbs a point distribution according to SIN distributions.
quadxy	Defines an arbitrary, logical quad of points in 2D (xy) space.
quadxyz	Defines an arbitrary, logical cube of points in 3D (xyz) space.
random	Randomizes a set of points within a set of restrictions.
recon	Reconnects a mesh to restore Delaunay criterion.
resetpts	Resets parent point type to max of its children.
restart	Restarts the code with a new dump file.
rmpoint	Removes a specified list of points from a point distribution.

Brief Description of Additional Commands (Continued)

rmregion	Removes points that lie within the specified region.
rmsphere	Removes a sphere of points from a point distribution.
rmsurf	Removes points that lie in, on or in and on the specified surface.
rzd	Refines a contour or generates points between two contours.
rzdsab	Zones between surfaces generated by SABRINA.
sabangle	Generate a file containing a set of angles that SABRINA will use as its angular distribution for generating points.
sabplot	Generate a file containing SABRINA body and cell data converted from **surface** and **region** or **mregion** definitions.
sample	Used to tally sampling on or off and to set the sampling interval (ms).
scalec	Scale a set of points to a contour.
setjtet	Sets jtet values at tetrahedron faces formed by sets of specified points.
source	Defines source tables for energy, temperature, pressure, or velocity source.
sphbound	Defines a reflective boundary with a r-, theta-, phi-command set.
surf3d	Generate points on a general 3D reference surface using a set of points as shooting rays.
tettest	Performs various options on the jtet array.
unit	Checks and changes the physical units for a given problem.
variable	Reads and saves a value for a user specified variable name.
view3	Sets viewing angle and distance from the coordinate origin.
wedge	Defines the reflective boundaries that makeup a wedge.
window	Sets the maximum and minimum plotting window.
x3d	Execute another ***NWGrid*** controllees from the current controllee.
z	Prints the status of a given point.

Brief Description of Additional Commands (Continued)

zqread	Reads a file of global arrays data.
zqwrite	Writes a file of global arrays data.

7.0 Error Messages

NWGrid displays alot of messages. Unless it actually says there's an error or aborts, it should be assumed to be running correctly.

Error Messages Descriptions

8.0 Documentation

The following ***NWGrid*** documents can be found on the internal, ***NWGrid/NWPhys*** web site. **http://www.emsl.gov/nwgrid**

NWGrid Installation Guide
NWGrid Tutorial
NWGrid Examples
NWGrid Command Reference
NWGrid Data Structures Reference
NWGrid Programmers Reference

9.0 Contact

Contact Lynn Trease, llt@pnl.gov, for further assistance and questions.