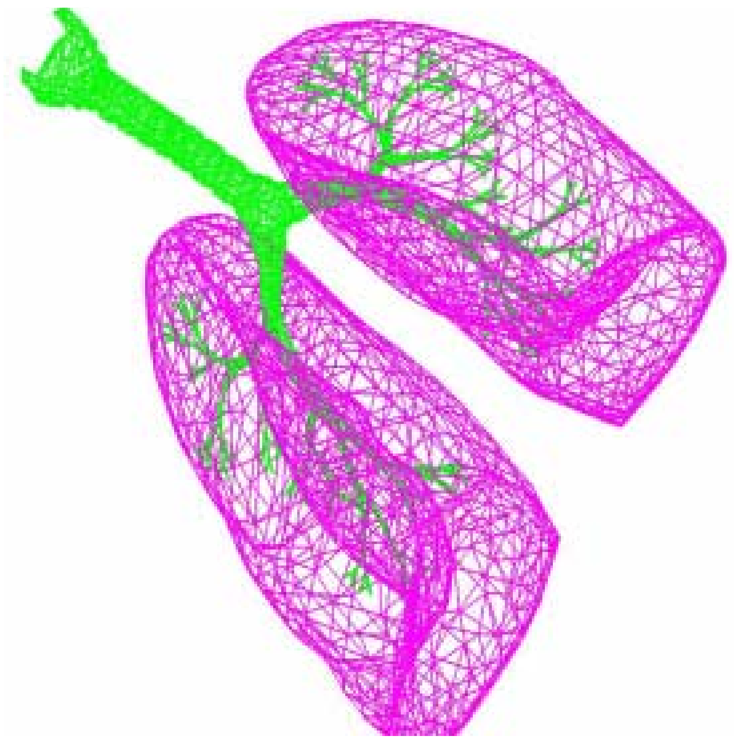
NWGrid Examples

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

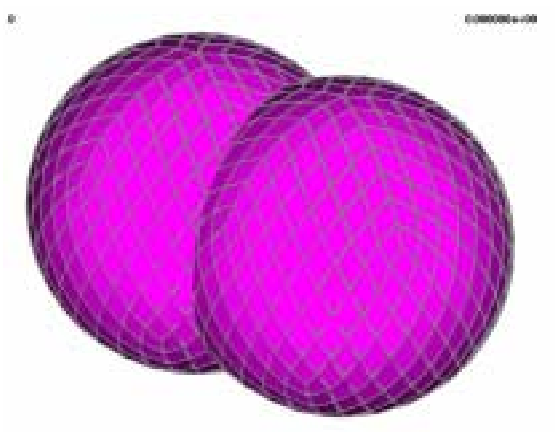
Introduction

For all these examples, lines that begin with an asterisk (*) are comments; keywords are in bold. Detailed descriptions of the ***NWGrid*** commands are given in the Command Reference for ***NWGrid***.

Example Command Segments

Sphere

* create the current mesh object
cmo / create / cmo1
* make 2 spheres
rzs / 8 / 10, 642 / 0.0 20.0 / 30.0, 150.0, 0.0 / 1
rzs / 8 / 10, 642 / 0.0, 20.0 / 50.0, 150.0, 0.0 / 1
* connect the points
zq / imt / 0, 0, 0 / 6
* write mesh to gmv file
dump / gmv / sphere.gmv / cmo1
* terminate processing
finish



Box

*Geometry
cmo / create / cmo1
* create box
surface / cube / reflect / box / 0., 0., 0. / 1., 1., 1.
* define geometric regions and material regions
region / inside / le cube
region / outside / gt cube
mregion / minside / le cube
mregion / moutside / gt cube

*Node Distribution
* create 125 points (5x5x5) in a plane above the unit cube
* place points on the boundaries in the x and y direction (1,1,1)
rz / xyz / 5, 5, 5 / 0., 0., 0., / 1., 1., 1., / 1, 1, 1

*Node Connectivity
* eliminate coincident or nearly coincident points
* 1,0,0 means all points
filter / 1, 0, 0
* identify interior interfaces and set material
* type for nodes and set constraints
setpts

```
* connect the points into a Delaunay tetrahedral mesh
* do not connect across material interfaces –
* add points if necessary to resolve material interfaces
search
* set material type for elements (tetrahedral) and setup
* parent/child relationship for interface points
settets

dump / gmv / box.gmv / cmo1
finish
```

After search, for a normal run the following message should appear:

There are 87 points that result in degeneracies or in wafer-thin tetrahedra, and therefore, require step 2.

Successfully eliminated all multimaterial connections.

The mesh is now complete!

If it doesn't say the above, there will be multimaterial tets.

Adaptive Mesh Refinement (AMR)

```
*set up geometry
modbld / mat_1 / amr / initial / minlevel=0 / maxlevel=2 / sizemix=40.0
modbld / mat_2 / amr / initial / minlevel=0 / maxlevel=2 / sizemix=40.0
…..
modchk
modval
dump / gmv / level2.gmv / 3dmesh

modbld / mat_1 / amr / initial / minlevel=0 / maxlevel=3 / sizemix=20.0
modbld / mat_2 / amr / initial / minlevel=0 / maxlevel=3 / sizemix=20.0
…..
modchk
modval
dump / gmv / level3.gmv / 3dmesh

gtg / amrtox3d
hextotet / 0 / cmo_tet / cmotmp
dump / gmv / level3_tet.gmv / cmo_tet
```

Refine with AMR

```
read / gmv / grid.gmv / cmo1
* coordinates of area to refine further
* amr 123 refines in xyz
* amr 1 refines only in x, 2 only in y, 3 only I z
* amr 12 refines in xy, 13 in xz
eset / ref10 / geom. / xyz / 1, 0, 0 /171062.2, 237312.2, 1330.625 / &
      171137.8, 237387.8, 1432.441
refine / constant / itetclr / linear / element / eset, get, ref10 / &
      -1., 0., 0. / inclusive / amr 123
rmpoint / compress
filter / 1, 0, 0
geniee
gtg / hybrid
```

Doping Nodes from an Infiltration Map onto a Regular Grid

* read in connected grid
read / gmv / 2dconnect.gmv / cmo_tri1

* create cmos of mesh type hex and select it
cmo / create / cmo1 / / / hex
cmo / select / cmo1

* distribute points by building brick mesh and generating nearest
* neighbor connectivity matrix
* create tet grid from hex – 6 tet per hex
* reset node values for itp
rzbrick / xyz / 61 91 2 / 533340. 4046780. –1. / 563340. 4091780. 1.1 / 1 1 1
hextotet / 6 / cmo_tet1 / cmo1
resetpts / itp

* select triangular cmos
* calculate the dope table pointer which will be put into the
* idop array created in the com_tri1 mesh
cmo / select / cmo_tri1
doping / integer2 / imt1 / set / 1 0 0 / cmo_tet1 / imt1 / minp / create
dump / gmv / tri1.gmv / cmo_tri1

* condense avs output file to just the attributes
* by putting 0 0 1 on end of dump
dump / avs / tri1_att.inp / cmo_tri1 / 0 0 1
finish

2-D Slice (from tets to triangles)

read / avs / tet.inp/ cmo_tet

* create cmos for triangular grids
cmo / create / cmo_tri

* extract off the bottom of the plane (z=-1)
extract / plane / threepts /545723.4 4075061.4 –1. / &
533673.0 4047320.0 –1. / &
0.0 0.0 –1. / &
1 0 0 / cmo_tri / cmo_tet
dump / gmv / tri.gmv / cmo_tri

2-D Surface from xyz Plane

read / avs / grid.inp / cmo1
*xic, yic, or zic for xyz plane
extract / isosurf / yic / 51.0 / 1, 0, 0 / cmo2 / cmo1
filter / 1, 0, 0
geniee
dump / gmv / 2d_grid.gmv / cmo1

Sort the i-, j-, k-indexes and Get min/max Attribute Values

read / avs / tri.inp / cmo_tri
sort / xyz / bins
cmo / printatt / cmo_tri **/ -all- / minmax**

Cut out a Part of a Grid

pset / inner **/ geom** / **xyz** / 1 0 0 / 170600.0 236820.0 918.5 / &
171600.0 237820.0 1581.5
pset / notinner / not inner
rmpoint / pset get notinner / **exclusive**
rmpoint / compress
resetpts / itp

Translate a Grid to 0 0 0 Coordinates

trans / 1, 0, 0 / zero / xyz / 1, 1, 0

Build the Connectivity Matrix

* create triangular cmo and read in avs file
cmo / create / cmo / -def- / -def- / tri
read / avs / flux.inp / cmo

* set attributes in cmo for imt to all 1
cmo / setatt / cmo / imt / 1 0 0 / 1

* build the connectivity matrix
search2d

*set attributes in cmo for itetclr to all 1
cmo / setatt / cmo / itetclr / 1 0 0 / 1

* reconnect grid and add points on interfaces
recon / 0
resetpts / itp

dump / gmv / 2dconnect.gmv / cmo

Get the External Boundary

extract / external / 19 / 1, 0, 0 / cmo2d / cmo1
rmpoint / compress

Smoothing

average_coord_plane / 1, 0, 0 / 100

Convert from between File Formats

read / x3d / grid.x3d / 3dmesh
*to convert x3d file to an avs file
dump / avs / grid.inp / 3dmesh
*to convert x3d file to a gmv file
dump / gmv / grid.gmv / 3dmesh

Partition Grid

read / gmv / grid.gmv / cmo1
* partition into 8 sections
partition / -def- / 8
dump / gmv / grid_part8.gmv / cmo1

Complete Input Files by Project Type

Adaptive Mesh Refinement (AMR)

* Convert an xdl file to an AMR file.

* Read an xdl file called xdl_file, into a cmo called amr_table,
* to use as a lookup table.
*
read / xdl / xdl_file / amr_file
*
* Create a new cmo called cmo_amr_grid, that will contain the amr grid.
*
cmo / create / cmo_amr_grid
*
* Build an equation-of-state (eos) model for each material and
* assign eos numbers.
*
modbld / 75 / eos / sesame / eosid = 75 / rho0 = 1.000000e+00
*

```
* Build an opacity model for each material and assign opacity numbers.
*
modbld / 75 / opacity / sesame / opcid = 10075
*
* Build an amr refinement model for each material region and
* define the following entries:
*          minlevel          - the minimum refinement level
*          maxlevel          - the maximum refinement level
*          sizeint           - the maximum edge length along a material
*                              interface
*          sizemix- the maximum edge length across a material
*                              interface
*          amr_table         - the name of the amr lookup table
*          background        - the name of the background material
*
modbld / 75 / amr / initial / minlevel = 1 /  &
                           maxlevel = 4 / &
                           sizeint = 1.9921993e-01 / &
                           sizemix = 1.9921993e-01 / &
                           amr_table = amr_table / &
                           background = bground
*
modbld / 7670 / amr / initial / minlevel = 1 / &
                           maxlevel = 11 / &
                           sizeint = 1.6175573e-03 / &
                           sizemix = 1.6175573e-03 / &
                           amr_table = amr_table / &
                           background = bground
*
modbld / 8180 / amr / initial / minlevel = 1 / &
                           maxlevel = 5 / &
                           sizeint = 6l8919820e-02 / &
                           sizemix = 6.8919820e-02 / &
                           amr_table = amr_table / &
                           background = bground
*
* Check the syntax and parameters of the model commands against
* the model dictionary.
*
modchk
*
* Build the background grid. A Cartesian, orthogonal, equally spaced
* grid. Then connect it into a logical grid.
*
rz / xyz / 23  137  1 / &
          0.0000000e+00, -5.90000000e+01, 0.00000000e+00 / &
          2.2000000e+01,  7.70000000e+01, 0.00000000e+00 / &
          1, 1, 1
rzbrick / xyz / 23  137  1 / 0, 0, 0 / connect
*
* Initialize the whole grid to the background material.
*
zq / imd / 0, 0, 0 / bground
*
* Build the models for the background material.
*
modbld / bground / eos / sesame / eosid = 503 / rho0 = 1.0000000e+00
modbld / bground / opacity / sesame / opcid = 15030
modbld / bground / amr / initial / minlevel = 1 / &
                           maxlevel = 1 / &
                           sizeint = 1.0000000e-06 / &
                           sizemix = 1.0000000e-06 / &
                           amr_table = amr_table
                           background = bground
```

```
*
* Assign node types and initial element materials and compress
* the data structure.
*
setpts
settets
rmpoint / compress
*
* Create an initial material map on the background grid by
* using the amr lookup table as a reference.
*
mapgrid / -def- / amr_table / -all / bground
*
* Partition the mesh for n-processors.
*
partition / -def- / 1
*
* Check the models, and then evaluate the amr model using
* the criteria defined in the amr model. This will generate
* the amr grid.
*
modchk
modval / amr
modval / amr
*
* Check the amr grid to make sure the refinement levels
* satisfy the 1-level difference rule (this call is for the
* parallel version).
*
check_amr_level
check_amr_level
*
* Map the density, temperature, specific internal energy, and
* velocities from the amr look-up table to the amr grid that
* was generated.
*
cmo / addatt / -def- / density / vdouble / scalar / nelements / &
        linear / permanent / rgx / 0.0
doping / table / density / set / 1, 0, 0 / amr_table / density / &
        linear / rty / xy
cmo / addatt / -def- / temperature / vdouble / scalar / nelements / &
        linear / permanent / rgx / 0.0
doping / table / temperature / set / 1, 0, 0 / amr_table / temperature / &
        linear / rty / xy
cmo / addatt / -def- / sie / vdouble / scalar / nelements / &
        linear / permanent / rgx / 0.0
doping / table / sie / set / 1, 0, 0 / amr_table / sie / linear / rty / xy
cmo / addatt / -def- / xdot / vdouble / scalar / nelements / &
        linear / permanent / rgx / 0.0
doping / table / xdot / set / 1, 0, 0 / amr_table / xdot / linear / rty / xy
cmo / addatt / -def - / ydot / vdouble / scalar / nelements / &
        linear / permanent / rgx / 0.0
doping / table / ydot / set / 1, 0, 0 / amr_table / ydot / linear /rty / xy
*
* Assign the density, temperature, and specific internal energy
* energy to the background material.
*
eset / background / zq / imd / 1, 0, 0 / bground
zq / density / eset, get, background / 1.293e-03
zq / temperature / eset, get, background / 0.010
zq / sie / eset, get, background / 0.0
*
* Write a x3d and gmv file.
*
```

```
dump / x3d / x3d_to_amr_2d.x3d
dump / gmv / x3d_to_amr_2d.gmv
*
finish
```

Unstructured Grids

```
* Convert an xdl file to an unstructured mesh file.

* Read an xdl file called xdl_file, into a cmo called amr_table0,
* to use as a lookup table.
*
read / xdl / xdl_file / amr_table0
rmpoint / compress
zq / itp1 / 1, 0, 0 / 0
*
* Re-read the xdl file and build the unstructured mesh flag.
*
read / xdl / xdl_file / amr_table
rmpoint / compress
zq / itp1 / 1, 0, 0 / 0
*
* Assign boundary node flags. Cleanup the mesh by filtering
* duplicate nodes, converting it to a hybrid mesh, and
* compressing the data structures.
*
setpts
filter / 1, 0, 0
geniee
gtg / hybrid
rmpoint / compress
*
* Map density, temperature, pressure, and the he-burn parameters
* from the original xdl file.
*
cmo / addatt / -def- / density / vdouble / scalar / nelements / &
        linear / permanent / fgx / 0.0
doping / table / density / set / 1, 0, 0 / amr_table0 / density /
        linear / rty / xy
cmo / addatt / -def- / temperature / vdouble / scalar / nelements / &
        linear / permanent / fgx / 0.0
doping / table / temperature / set / 1, 0, 0 / amr_table0 / temperature /
        linear / rty / xy
cmo / addatt / -def- / ptot / vdouble / scalar / nelements / &
        linear / permanent / fgx / 0.0
doping / table / ptot / set / 1, 0, 0 / amr_table0 / ptot /
        linear / rty / xy
cmo / addatt / -def- / hebdst / vdouble / scalar / nelements / &
        linear / permanent / fgx / 0.0
cmo / addatt / -def- / hebinv / vdouble / scalar / nelements / &
        linear / permanent / fgx / 0.0
cmo / addatt / -def- / hebstr / vdouble / scalar / nelements / &
        linear / permanent / fgx / 0.0
*
* Assign boundary node flags. Cleanup the mesh by filtering
* duplicate nodes, converting it to a hybrid mesh, and
* compressing the data structures.
*
filter / 1, 0, 0
geniee
gtg / hybrid
*
* Partition the mesh into 16 partitions (1 partition per processor)
```

```
* in preparation for mapping to a parallel computer.
*
partition / -def- / 16 / 1
*
* Write a x3d and gmv file.
*
dump / x3d / x3d_to_2d.x3d
dump / gmv / x3d_to_2d.gmv
*
finish
```

Finite Element Grids

```
* Convert an xdl file to an finite element mesh file.

* Read an xdl file called xdl_file, into a cmo called amr_table0,
* to use as a lookup table.
*
read / xdl / xdl_file / amr_table0
rmpoint / compress
zq / itp1 / 1, 0, 0 / 0
*
* Extract the free surface. Then rotate it into a 3-D surface
* to be used as geometry to flag free surface boundary
* nodes in the 3d mesh.
*
pset / n0free / geom. / xyz / 1, 0, 0 / 0.01, -1000,0, -1000.0 / &
                                        1000.0, 1000.0, 1000.0
extract / external / -all- / pset, get, n0free / cmo1_1d0 / amr_table0
rotateln / 1, 0, 0 / nocopy / 0.0, -1000.0, 0.0 / 0.0, 1000.0, 0.0 / 5.0
pset / p1 / seq / 1, 0, 0
*
rotateln / pset, get, p1 / copy / 0.0, -1000.0, 0.0 / 0.0, 1000.0, 0.0 / -5.0
pset / p2 / seq / 2, 0, 0
rzbrick / interp / relative / 0 / pset, get, p1 / pset, get, p2 / cmo1_2d0
*
* Re-read the xdl file and make it a 5 degree taco mesh by
*rotating about the y-axis by 5 degrees.
*
read / xd1 / xdl_file / amr_table
rmpoint / compress
zq / itp1 / 1, 0, 0 / 0
rotateln / 1, 0, 0 / nocopy / 0.0, -1000.0, 0.0 / 0.0, 1000.0, 0.0 / 5.0
pset / p1 / seq / 1, 0, 0
*
rotateln / pset, get, p1 / copy / 0.0, -1000.0, 0.0 / 0.0, 1000.0, 0.0 / -5.0
pset / p2 / seq / 0, 0, 0
rzbrick / interp / relative / 0 / pset, get, p1 / pset, get, p2 / cmo1_3d
filter / 1, 0, 0
geniee
gtg / hybrid
*
cmo / select / cmo1_3d
*
* Build the geometry for the outer free surface and the two
* reflective planes at 0 degrees and 5 degrees. This geometry
* will be used to flag free surface and reflective boundary nodes.
*
surface / free2 / free / sheet / cmo1_2d0
surface / plane1 / reflect / plane / 0.0, 0.0, 0.0 / 1.0, 0.0, 0.0 / &
                                        0.0, 1.0, 0.0
surface / plane2 / reflect / plane / 0.0, 0.0, 0.0 / 0.0, 1.0, 0.0 / &
                                        1.0, 0.0, -0.087488664
region / reg1 / lt free2 and le plane1 or le plane2
region / reg3 / ge free2
```

```
*
* Initialize all the node materials and node types to zero.
* Then test against the geometry to build the node types.
* Then color the nodes basaed on the element colors.
*
zq / imt / 1, 0, 0 / 0
zq / itp / 1, 0, 0 / 0
geniee
setpts
settets / color_points
*
* Map density, temperature and pressure from the xdl file to
* the 3D mesh. Then map these quantities to the nodes.
*
cmo / addatt / -def- / density / vdouble / scalar / nelements / &
        linear / permanent / gx / 0.0
doping / table / density / set / 1, 0, 0 / amr_table0 / density / &
        linear / rty / xy
cmo / addatt / -def- / temperature / vdouble / scalar / nelements / &
        linear / permanent / gx / 0.0
doping / table / temperature / set / 1, 0, 0 / amr_table0 / temperature / &
        linear / rty / xy
cmo / addatt / -def- / ptot / vdouble / scalar / nelements / &
        linear / permanent / gx / 0.0
doping / table / ptot / set / 1, 0, 0 / amr_table0 / ptot / &
        linear / rty / xy
*
cmo / addatt / -def- / rho / / / / / / cgx
cmo / addatt / -def- / t / / / / / / cgx
cmo / addatt / -def- / p / / / / / / cgx
copypts / -def- / -def- / 1, 0 / 1, 0, 0 / rho / density
copypts / -def- / -def- / 1, 0 / 1, 0, 0 / t / temperature
copypts / -def- / -def- / 1, 0 / 1, 0, 0 / p / ptot
*
* Cleanup the mesh by filtering duplicate nodes and removing
* degenerate elements by converting it to a hybrid grid.
*
filter / 1, 0, 0
gtg / hybrid
*
* Partition the mesh into 16 partitions (1 partition per processor)
* in preparation for mapping to a parallel computer.
*
partition / -def- / 16, 1 / nodes
*
* Write a x3d and gmv file
*
dump / x3d / x3d_to_fe.x3d
dump / gmv / x3d_to_fe.x3d
*
finish
```

Engineering Problems

NWGrid uses the following procedure for engineering grid generation. Surface geometry is imported from a Computer Aided Design (CAD) system through Stereo Lithography (STL) files, constructive solid geometry primitives, or facetted surfaces. An intermediate code called Oso can be used to edit the geometry. The surface geometry is formed into closed volumes using binary relations (<, _, >, _, =) and Boolean operators (Ø,Ù,Ú). AMR rules are specified, a background mesh is supplied, and the number of processors for running the problem is given. The types of AMR rules include characteristic scale length, refinement levels, etc. The background can be any hybrid unstructured and/or (block) structured mesh. METIS partitions and allocates the grid generation work to the number of processors. The mesh is then generated. Smoothing algorithms and local refinement algorithms can be used to fine tune the final grid before it is exported to an application code.

Documentation

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

NWGrid Installation Guide
NWGrid Users Manual
NWGrid Tutorial
NWGrid Command Reference
NWGrid Data Structures Reference
NWGrid Programmers Reference

Contact

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