MeshAssist

Generated by Doxygen 1.8.13

Contents

1 Authors 1

2 File Index 3

2.1 File List . . . 3

3 File Documentation 5

3.1 exodus2semgeotech.c File Reference . . . 5

3.1.1 Detailed Description . . . 5

3.2 exodus2specfem2d.c File Reference . . . 7

3.2.1 Detailed Description . . . 7

3.3 exodus2specfem3d.c File Reference . . . 9

3.3.1 Detailed Description . . . 9

3.4 exodusold2specfem3d.c File Reference . . . 11

3.4.1 Detailed Description . . . 11

3.5 gid2semgeotech.c File Reference . . . 13

3.5.1 Detailed Description . . . 13

3.6 gocad2vtu.c File Reference . . . 14

3.6.1 Detailed Description . . . 14

3.7 vti2cell.c File Reference . . . 15

3.7.1 Detailed Description . . . 15

3.8 vtk1d2jou.f90 File Reference . . . 16

3.8.1 Detailed Description . . . 17

3.9 vtk2d2jou.c File Reference . . . 17

3.9.1 Detailed Description . . . 17

3.10 xyz2jou.f90 File Reference . . . 18

3.10.1 Detailed Description . . . 18

Index 19

Chapter 1

Authors

Hom Nath Gharti

- formerly at Institute of Engineering, Tribhuvan University, Nepal
- formerly at NORSAR, Norway
- Department of Geosciences, Princeton University, USA

hgharti_AT_princeton_DOT_edu

Leah Langer

- Department of Geosciences, Princeton University, USA

llanger_AT_princeton_DOT_edu

Uno Vaaland

- Program in Applied & Computational Mathematics, Princeton University, USA

uvaaland_AT_princeton_DOT_edu

Michael Roth

- NORSAR, Norway

michael_AT_nosar_DOT_no

Jeroen Tromp

- Department of Geosciences, Princeton University, USA
- Program in Applied & Computational Mathematics, Princeton University, USA

jtromp_AT_princeton_DOT_edu

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

exodus2semgeotech.c
Converts ASCII exodus file to SPECFEM3D_GEOTECH files 5
exodus2specfem2d.c
Convert ASCII exodus file to SPECFEM2D format 7
exodus2specfem3d.c
Converts ASCII exodus file to SPECFEM3D files 9
exodusold2specfem3d.c
Converts old ASCII exodus file to SPECFEM3D files 11
gid2semgeotech.c
Converts ASCII Gid mesh file to SPECFEM3D_GEOTECH files 13
gocad2vtu.c
Converts GOCAD ASCII file to VTU file 14
vti2cell.c
This file converts VTI file to VTU file 15
vtk1d2jou.f90
Converts VTK 1D file to CUBIT/Trelis journal file 16
vtk2d2jou.c
Converts VTK file consisting of 2D mesh to CUBIT/Trelis journal file 17
xyz2jou.f90
Coverts UTM/XYZ file to CUBIT/Trelis journal file 18

Chapter 3

File Documentation

3.1 exodus2semgeotech.c File Reference

Converts ASCII exodus file to SPECFEM3D_GEOTECH files.

Functions

- void **removeExtension** (char ∗, char ∗)
- int **get_int** (int ∗, char ∗, char ∗)
- int **look_int** (int ∗, char ∗, char ∗)
- int **getfirstquote** (char ∗, char ∗)
- int **main** (int argc, char ∗∗argv)

3.1.1 Detailed Description

Converts ASCII exodus file to SPECFEM3D_GEOTECH files.

This program converts the Binary (provided that ncdump command exists) or ASCII exodus file exported from the CUBIT/Trelis to several mesh files required by the SPECFEM3D_GEOTECH package. The ncdump commad is a part of NetCFD library which is generally installed already in LINUX, which can be downloaded freely from https://www.unidata.ucar.edu/software/netcdf/

Author

Hom Nath Gharti (hgharti_AT_princeton_DOT_edu)

Dependencies:

stringmanip.c: string manipulation routines

Compile:

gcc exodus2semgeotech.c -o exodus2semgeotech

Usage:

exodus2semgeotech <inputfile> [OPTIONS]

Example: exodus2semgeotech tunnel.e -bin=1
or
exodus2semgeotech tunnel.txt

Options:

- -fac: Use this option to multiply coordinates with some factor. This is important for unit conversion, e.g., to convert m to km use -fac=0.001 [DEFAULT 1]
- -bin: Use this option if you want to convert exodus binary directly, provided that the command "ncdump" is in the path. The command "ncdump" is a part of netCDF library that can be downloaded freely from http://www.unidata.ucar.edu/downloads/netcdf/index.jsp. Use -bin=1 for binary or -bin=0 for ascii file. [DEFAULT 0]

Issues:

- - This does not work with older verion of CUBIT. For the older version use exodusold2semgeotech.c.

Basic steps starting from the CUBIT/TRELIS:

step 1: prepare mesh in TRELIS/CUBIT

- Define material regions using "Blocks"

  For example:
  block 1 add volume 1
  block 2 add volume 2 3

  will assign material region 1 to volume 1 and material region 2 to volumes 2 and 3. These material regions will be used to define material properties in "nummaterial_velocity_file". This program will NOT generate "nummaterial_velocity_file". The file "nummaerial_veolicty_file" must be created to run SPECFEM3D!

- Define surface boundary conditions using "Nodesets" or "Sidesets" – nodal boundary conditions must be defined using node set -> each node set name must contain the corresponding BC names as defined in char *ns_bcname[] below
  e.g., node set name can be front_nsbcux or front_nsbcux_nsbcuy etc. – surface boundary conditions must be defined using side set -> each side set name must contain the corresponding BC names as defined in char *ss_bcname[] below
  e.g., side set name can be front_ssbcux or front_ssbcux_ssbcuy etc.

  For example:
  sideset 1 add surface 1
  sideset 1 name 'bottom_ssbcux_ssbcuy_ssbcuz'

  will define a surface in which all displacement components are prescribed.

  Note: All the above commands can also be executed using TRELIS/CUBIT GUI. "sideset 1 name 'bottom_ssbcux_ssbcuy_ssbcuz'" is equivalent to clicking 'sideset 1' and renaming.

step2: export mesh file as exodus file say "tunnel.e" (use 3D option)

step3: convert "tunnel.e" to SPECFEM3D files

exodus2semgeotech tunnel.e -bin=1

There will be several output files:

- ∗_coord_? : coordinates file => total number of nodes followed by nodal coordinate ? (? -> x, y, z)
- ∗_connectivity : element file => total number of elements followed by connectivity list
- ∗_material_id : material file => total number of elements followed by material IDs
- ∗_??bcu? : node IDs which have u? defined as the boundary conditions (?? -> ns or ss, ? -> x, y, z). Total number of entities (nodes or faces) followed by element ID and surface nodes.

3.2 exodus2specfem2d.c File Reference

Convert ASCII exodus file to SPECFEM2D format.

Functions

- void **removeExtension** (char ∗, char ∗)
- int **get_int** (int ∗, char ∗, char ∗)
- int **look_int** (int ∗, char ∗, char ∗)
- int **getfirstquote** (char ∗, char ∗)
- int **shape** (double, double, double ∗∗)
- int **check_normal** (double [3][4], double [3])
- int **main** (int argc, char ∗∗argv)
- int **isclockwise** (int n, double x[n], double z[n])

3.2.1 Detailed Description

Convert ASCII exodus file to SPECFEM2D format.

This program converts the ASCII exodus file exported from the CUBIT to several input files required by the SPECFEM2D program. Currently, this program only handles the 2D quadrilateral elements with four nodes. The binary exodus file (e.g., .e file) needs to be converted into ASCII file, generally using a free console application "ncdump" which is a part of the netCDF library, and can be downloaded from http://www.unidata.ucar.edu/downloads/netcdf/index.jsp. Please see the detail steps below.

Author

Hom Nath Gharti (hgharti_AT_princeton_DOT_edu)

Dependencies:

stringmanip.c: string manipulation routines

Compile:

gcc exodus2specfem2d.c -o exodus2specfem2d -lm

Usage:

exodus2specfem2d <inputfile> [OPTIONS]

Example: exodus2specfem2d mesh.e -bin=1
or
exodus2specfem2d mesh.txt

Options:

- -fac: Use this option to multiply coordinates with some factor. this is important for unit conversion, e.g., to convert m to km use -fac=0.001 [DEFAULT 1]
- -bin: Use this option if you want to convert exodus binary directly, provided that the command "ncdump" is in the path. The command "ncdump" is a part of netCDF library that can be downloaded freely from http://www.unidata.ucar.edu/downloads/netcdf/index.jsp. use -bin=1 for binary or -bin=0 for ascii file. [DEFAULT 0]
- -order: Use this option to check the connectivity order and make sure that the connectivity is in counterclockwise order. Use -order=1 for checking or -order=0 for no checking [DEFAULT 0].
- -head: Use this option to attach head of input file to output file names. Use -head=1 to attach header or -head=0 not to attach [DEFAULT 0]
- -tomo: Use this option for tomography model. Since tomography model uses negative identifiers, this option will write negative block IDs. Use -tomo=1 to make negative block IDs or -tomo=0 not to make [DEFAULT 0]

Basic steps starting from TRELIS:

Step 1: prepare mesh in TRELIS/CUBIT

- Define material regions using "Blocks"

  For example:
  block 1 add surface 1
  block 2 add surface 2 3

  will assign material region 1 to surface 1 and material region 2 to surfaces 2 and 3. These material regions will be used to define material properties in "Par_file". This program will NOT generate "Par_file". The file "Par_file" must be created to run SPECFEM2D!

- Define element type to be QUAD4

  For example:
  block all element type quad4

  NOTE: If the element types are SHELL or SHELL4, "Default" or 3D option should be selected during export. If the element type is QUAD or QUAD4, 3D option should be selected. With default or 2D data, it saves only X and Y coordinates which is not always correct. Make sure that the node ordering is strictly anticlockwise (no longer necessary!) for all the elements in CUBIT.

- Define surface boundary conditions using "Sidesets"

  For example:
  sideset 1 add curve 1
  sideset 1 name 'free_surface_file'

  will define a free or absorbing surface boundary condition on surface. Similary,
  sideset 2 add curve 3
  sideset 2 name 'absorbing_surface_file'

  will define absorbing boundary condition on the curve 3. Note: All the above commands can also be executed using TRELIS/CUBIT GUI. "sideset 1 name 'free_surface_file'" is equivalent to clicking sideset 1 and renaming.

Step 2: export mesh file as exodus file say "mesh.e" (always use 3D option!)

Step 3: convert "mesh.e" to SPECFEM2D files

exodus2specfem2d mesh.e -bin=1

There will be several output files:

- coordinates : coordinates file => total number of nodes followed by nodal coordinate ? (? -> x, y, z)
- connectivity : element file => total number of elements followed by connectivity list
- materials : material file => total number of elements followed by material IDs
- surface∗ : sourface boundary condition files => total number of elements followed by element ID and surface nodes

3.3 exodus2specfem3d.c File Reference

Converts ASCII exodus file to SPECFEM3D files.

Functions

- void **removeExtension** (char ∗, char ∗)
- int **get_int** (int ∗, char ∗, char ∗)
- int **look_int** (int ∗, char ∗, char ∗)
- int **getfirstquote** (char ∗, char ∗)
- int **check_normal** (double [3][4], double [3])
- int **main** (int argc, char ∗∗argv)

3.3.1 Detailed Description

Converts ASCII exodus file to SPECFEM3D files.

This program converts the Binary (provided that ncdump command exists, type ncdump to check whether ncdump command exists.) or ASCII exodus file exported from the TRELIS/CUBIT to several mesh files required by the SPE↩ CFEM3D Cartesian package. The ncdump commad is a part of NetCFD library which is generally installed already in LINUX, which can be downloaded freely from https://www.unidata.ucar.edu/software/netcdf/

Author

Hom Nath Gharti (hgharti_AT_princeton_DOT_edu)

Dependencies:

stringmanip.c: string manipulation routines

Compile:

gcc exodus2specfem3d.c -o exodus2specfem3d

Usage:

exodus2specfem3d <inputfile> [OPTIONS]

Example: exodus2specfem3d tunnel.e -bin=1
or
exodus2specfem3d tunnel.txt

Options:

- -fac: Use this option to multiply coordinates with some factor. This is important for unit conversion, e.g., to convert m to km use -fac=0.001 [DEFAULT 1]
- -bin: Use this option if you want to convert exodus binary directly, provided that the command "ncdump" is in the path. The command "ncdump" is a part of netCDF library that can be downloaded freely from http://www.unidata.ucar.edu/downloads/netcdf/index.jsp. use -bin=1 for binary or -bin=0 for ascii file. [DEFAULT 0]
- -norm: Use this option to check the normal of the faces in order to make sure that the surface nodes are in the right order. Use -norm=1 for checking or -norm=0 for no checking [DEFAULT 0]. Normally this is not necessary.
- -head: Use this option to attach head of input file to output file names. Use -head=1 to attach header or -head=0 not to attach [DEFAULT 0]
- -tomo: Use this option for tomography model. Since tomography model uses negative identifiers, this option will write negative block IDs. Use -tomo=1 to make negative block IDs or -tomo=0 not to make [DEFAULT 0]

Issues:

- - This does not work with older verion of CUBIT. For the older version use exodusold2specfem3d.c.

Basic steps starting from the TRELIS:

step 1: prepare mesh in TRELIS/CUBIT

- Define material regions using "Blocks"

  For example:
  block 1 add volume 1
  block 2 add volume 2 3

  will assign material region 1 to volume 1 and material region 2 to volumes 2 and 3. These material regions will be used to define material properties in "nummaterial_velocity_file". This program will NOT generate "nummaterial_velocity_file". The file "nummaerial_veolicty_file" must be created to run SPECFEM3D!

- Define surface boundary conditions using "Sidesets"

  For example:
  sideset 1 add surface 1
  sideset 1 name 'free_or_absorbing_surface_file_zmax'

  will define a free or absorbing surface boundary condition on surface 1 which lies at the top of the volume (zmax). similary,

  sideset 2 add surface 3
  sideset 2 name 'absorbing_surface_file_bottom'

  will define absorbing boundary condition on the surface 3 which lies at the bottom of the volume (zmin). Note: All the above commands can also be executed using TRELIS/CUBIT GUI. "sideset 1 name 'free_or↩ _absorbing_surface_file_zmax'" is equivalent to clicking sideset 1 and renaming.

step2: export mesh file as exodus file say "tunnel.e" (use 3D option)

step3: convert "tunnel.e" to SPECFEM3D files

exodus2specfem3d tunnel.e -bin=1

There will be several output files:

- nodes_coords_file : coordinates file => total number of nodes followed by nodal coordinate ? (? -> x, y, z)
- mesh_file : element file => total number of elements followed by connectivity list
- materials_file : material file => total number of elements followed by material IDs
- surface_file∗ : sourface boundary condition files => total number of elements followed by element ID and surface nodes

3.4 exodusold2specfem3d.c File Reference

Converts old ASCII exodus file to SPECFEM3D files.

Functions

- void **removeExtension** (char ∗, char ∗)
- int **get_int** (int ∗, char ∗, char ∗)
- int **look_int** (int ∗, char ∗, char ∗)
- int **getfirstquote** (char ∗, char ∗)
- int **check_normal** (double [3][4], double [3])
- int **main** (int argc, char ∗∗argv)

3.4.1 Detailed Description

Converts old ASCII exodus file to SPECFEM3D files.

This program converts the Binary (provided that ncdump command exists) or ASCII exodus file exported from the old CUBIT to several mesh files required by the SPECFEM3D package.

Author

Hom Nath Gharti (hgharti_AT_princeton_DOT_edu)

Dependencies:

stringmanip.c: string manipulation routines

Compile:

gcc exodusold2specfem3d.c -o exodusold2specfem3d

Usage:

exodusold2specfem3d input_file [OPTIONS]

Example: exodusold2specfem3d tunnel.txt
or
exodusold2specfem3d tunnel.e -fac=0.001 -bin=1

Options:

- -fac: use this option to multiply coordinates. this is importantn for unit conversion, e.g., to convert m to km use -fac=0.001
- -bin: use this option if you want to convert exodus binary directly, provided that the command ncdump is in the path. ncdump is a part of netCDF library that can be downloaded freely from http://www.unidata.↩ucar.edu/downloads/netcdf/index.jsp. use -bin=1 for binary or -bin=0 for ascii file.
- -norm: use this option to check the normal of the faces. use -norm=1 for checking or -norm=0 (default) for no checking

Issues:

- - This does not work with new verion of Trelis/CUBIT. For the new version use exodus2specfem3d.c.

 # Basic steps starting from the CUBIT:

step 1: prepare mesh in CUBIT

- define material regions using "Blocks"

  For example:
  block 1 add volume 1
  block 2 add volume 2 3

  will assign material region 1 to volume 1 and material region 2 to volumes 2 and 3. These material regions will be used to define material properties in "nummaterial_velocity_file". this program will NOT generate "nummaterial_velocity_file". the file "nummaerial_veolicty_file" must be created to run SPECFEM3D!

define surface boundary conditions using "Sidesets"

For example:
sideset 1 add surface 1
sideset 1 name 'free_or_absorbing_surface_file_zmax'

will define a free or absorbing surface boundary condition on surface 1 which lies at the top of the volume (zmax). similary,

sideset 2 add surface 3
sideset 2 name 'absorbing_surface_file_bottom'

will define absorbing boundary condition on the surface 3 which lies at the bottom of the volume (zmin). Note: All the above commands can also be executed using TRELIS/CUBIT GUI. "sideset 1 name 'free_or_absorbing_surface↩_file_zmax'" is equivalent to clicking sideset 1 and renaming.

step2: export mesh file as exodus file say "tunnel.e" (use 3D option)

step3: convert "tunnel.e" to SPECFEM3D files

exodusold2specfem3d tunnel.e -bin=1

There will be several output files:

- nodes_coords_file : coordinates file => total number of nodes followed by nodal coordinate ? (? -> x, y, z)
- mesh_file : element file => total number of elements followed by connectivity list
- materials_file : material file => total number of elements followed by material IDs
- surface_file∗ : sourface boundary condition files => total number of elements followed by element ID and surface nodes

3.5 gid2semgeotech.c File Reference

Converts ASCII Gid mesh file to SPECFEM3D_GEOTECH files.

Functions

- void **removeExtension** (char ∗, char ∗)
- int **get_int** (int ∗, char ∗, char ∗)
- int **look_int** (int ∗, char ∗, char ∗)
- int **getfirstquote** (char ∗, char ∗)
- int **main** (int argc, char ∗∗argv)

3.5.1 Detailed Description

Converts ASCII Gid mesh file to SPECFEM3D_GEOTECH files.

This program converts the ASCII GiD mesh file to several mesh files required by the SPECFEM3D_GEOTECH package. GiD (www.gidhome.com) is a commercial pre and post processor for numerical simulations.

Author

Hom Nath Gharti (hgharti_AT_princeton_DOT_edu), Zhenzhen Yan

Dependencies:

stringmanip.c: string manipulation routines

Compile:

gcc gid2semgeotech.c -o gid2semgeotech

Usage:

gid2semgeotech <inputfile> <OPTIONS>

Example: gid2semgeotech gid2semgeotech_example.dat
or
gid2semgeotech gid2semgeotech_example.dat -fac=0.001

Options:

- -fac: Use this option to multiply coordinates with a certain factor. This is useful for unit conversion, e.g., to convert m to km use: -fac=0.001

Basic steps starting from GID:

step1: Export mesh file in ASCII format "mesh.dat"

step2: Produce mesh and BC files

gid2semgeotech mesh.dat
OR
gid2semgeotech mesh.dat 1000.0
There will be several output files:

- coord_? : Total number of nodes followed by nodal coordinate ? (? -> x,y,z)
- _connectivity : Total number of elements followed by connectivity list
- _material_id : Total number of elements followed by material IDs
- ??bcu? : node IDs which have u? = 0 as the boundary conditions (?? -> ns or ss, ? -> x,y,z)

3.6 gocad2vtu.c File Reference

Converts GOCAD ASCII file to VTU file.

Functions

- int **main** (int argc, char **argv)

3.6.1 Detailed Description

Converts GOCAD ASCII file to VTU file.

This program converts the GOCAD ASCII file (3-noded triangular meshes) to VTK XML .vtu binary file (unstructured mesh file) which can be visualized/processed in ParaView or VTK.

Author

Hom Nath Gharti (hgharti_AT_princeton_DOT_edu)

Dependencies:

stringmanip.c

Compile

- in parent folder, type: make OR
- in src/ folder, type gcc gocad2vtu.c -o gocad2vtu

Usage:

./bin/gocad2vtu <inputfile> [OPTIONS]

Example: ./bin/gocad2vtu ./input/gocad2vtu_example.ts

Options:

- -fac: Use this option to multiply the coordinates by certain factor, this is helpful for unit conversion, e.g. for m to km use 0.001, for km to m use 1000, example: gocad2vtu T2_horizon.ts -fac=0.001

Notes:

- Output .vtu file is binary, therefore endianness of the processor architechture is important.
- This program automatically identify the endianness and write the output accordingly. Hence if you run and process/visualize .vtu file in the architecture with different endianness there may be an error.

3.7 vti2cell.c File Reference

This file converts VTI file to VTU file.

Functions

- int **comp_float** (const void *a, const void *b)
- int **main** (int argc, char **argv)

3.7.1 Detailed Description

This file converts VTI file to VTU file.

This program converts the 2D/3D Binary VTK XML .vti file to unstructured mesh files (.vtu). This program also generates the mesh files required by SPECFEM2D and SPECFEM3D. Note that the file formats in SPECFEM2D and SPECFEM3D are different. This should be made same format as soon as possible. For this, source codes within the decompose folder of SPECFEM3D and cubit2specfem3d.py need to be changed.

Author

Hom Nath Gharti (hgharti_AT_princeton_DOT_edu)

Dependencies:

stringmanip.c

Compile:

gcc vti2cell.c -o vti2cell -lm

Usage:

vti2cell <inputfile> [OPTIONS]

Example: vti2cell py_plane_model.vti

Options:

- -fac=factor (real) Use this option to multiply the coordinates by certain factor, this is helpful for unit conversion, e.g. for m to km use 0.001, for km to m use 1000 Example: vti2cell2d py_plane_model.vti -fac=1000
- -xmat=exclusion material id/s (integer/s) Use this option to exclude certain region of the model, e.g. exclusion of air. Appropriate id/s should be supplied, id s are number orderd according to the value of corresponding material properties and numbered starting from 1. This way, lowest value will have id 1 and so on. Example: vti2cell py_plane_model.vti -xmat=1,2 This command will exclude the regions with material id 1 and 2.

  Example: vti2cell py_plane_model.vti -fac=1000 -xmat=1 This command multiply the coordinates by 1000 and exclude the region with material id 1
- -step=step size (integer) Use this option to coarsen the mesh. This value represent the number of grids to be used as 1 element, e.g., if you want to make 2 grids as 1 element, use -step=2
- -zup=z axis direction indicator (integer) Use this option to indicate whether the Z axis direction is up

Toto:

- make uniformity for 2D,3D, e.g., writing and reading coordinates

3.8 vtk1d2jou.f90 File Reference

Converts VTK 1D file to CUBIT/Trelis journal file.

Functions/Subroutines

- program **vtk1d2jou**

3.8.1 Detailed Description

Converts VTK 1D file to CUBIT/Trelis journal file.

This program read ASCII vtk files with unstructured grid of lines and points only, and removes the redundant lines. The redundant nodes can be removed within the paraview itself using 'Clean to Grid' filter.

Hom Nath Gharti (hgharti_AT_princeton_DOT_edu

Compile:

gfortran vtk1d2jou.f90 -o vtk1d2jou

Usage:

vtk1d2jou <input_file>

vtk1d2jou vtk1d2jou_example.vtk

3.9 vtk2d2jou.c File Reference

Converts VTK file consisting of 2D mesh to CUBIT/Trelis journal file.

Functions

- void **removeExtension** (char *, char *)
- int **main** (int argc, char **argv)

3.9.1 Detailed Description

Converts VTK file consisting of 2D mesh to CUBIT/Trelis journal file.

This program converts ASCII VTK file consisting of triangular/quadrilateral mesh into CUBIT/Trelis Journal file.

Author

Hom Nath Gharti (hgharti_AT_princeton_DOT_edu)

Dependences:

stringmanip.c: string manipulation routines

Compile:

gcc vtk2d2jou.c -o vtk2d2jou

Usage:

vtk2d2jou input_file

Example: vtk2d2jou vtk2d2jou_example.vtk

3.10 xyz2jou.f90 File Reference

Coverts UTM/XYZ file to CUBIT/Trelis journal file.

Functions/Subroutines

- program **xyz2jou**

3.10.1 Detailed Description

Coverts UTM/XYZ file to CUBIT/Trelis journal file.

This program converts UTM or XYZ file to CUBIT journal file. The UTM or XYZ file contains the three columns of X, Y, and Z coordinates, respectively.

Hom Nath Gharti (hgharti_AT_princeton_DOT_edu

Compile:

- in parent folder, type: make OR
- in src/ folder, type gfortran xyz2jou.f90 -o xyz2jou

Usage:

./bin/xyz2jou input_file [OPTIONS]

Example: ./bin/xyz2jou ./input/xyz2jou_example.utm

Options:

- -nx: Use this option if you know the number of points in a line along X axis . This will speed up the processing. For example, -nx=100. If it is not defined, nx is automatically determined.
- -nskip: Use this option if you want to skip (downsample) certain number of successive points. This will skip along both X and Y axes. For example, -nskip=2. [DEFAULT 0].

Index

exodus2semgeotech.c, 5
exodus2specfem2d.c, 7
exodus2specfem3d.c, 9
exodusold2specfem3d.c, 11

gid2semgeotech.c, 13
gocad2vtu.c, 14

vti2cell.c, 15
vtk1d2jou.f90, 16
vtk2d2jou.c, 17

xyz2jou.f90, 18