

# MIN3P-THCm-USG User Manual Draft

Danyang Su<sup>a</sup>, Mingliang Xie<sup>a</sup>, K. Ulrich Mayer<sup>a</sup>

Kerry T.B. MacQuarrie<sup>b</sup>

<sup>a</sup>Department of Earth, Ocean and Atmospheric Sciences, University of British Columbia,  
Vancouver, BC, V6T 1Z4, Canada

<sup>b</sup>Department of Civil Engineering, University of New Brunswick, P.O. Box 4400,  
Fredericton, NB, E3B 5A3, Canada

**July 2019**

## Contents

<b>MIN3P-THCM-USG USER MANUAL.....</b>	<b>2.1-1</b>
<b>DRAFT .....</b>	<b>2.1-1</b>
<b>1    SYSTEM REQUIREMENT.....</b>	<b>1</b>
<b>2    INPUT FILE FOR MIN3P-THCM-USG.....</b>	<b>1</b>
2.1    SPATIAL DISCRETIZATION RELATED COMMANDS .....	1
2.1.1 <i>Mesh Format.....</i>	<i>1</i>
2.1.2 <i>Built-in mesh generation.....</i>	<i>2</i>
2.1.3 <i>Read mesh from file .....</i>	<i>21</i>
2.1.4 <i>Control volume method.....</i>	<i>22</i>
2.1.5 <i>Gradient reconstruction method.....</i>	<i>23</i>
2.1.6 <i>Gradient reconstruction spatial weighting .....</i>	<i>24</i>
2.1.7 <i>Gradient interpolation .....</i>	<i>25</i>
2.1.8 <i>Gradient reconstruction flux correction .....</i>	<i>26</i>
2.1.9 <i>Optional cell based parameters .....</i>	<i>27</i>
2.1.10 <i>Read initial condition from vtk file .....</i>	<i>28</i>
2.1.1 <i>Read other parameters from vtk file .....</i>	<i>28</i>
2.1.2 <i>Cell-node ordering control .....</i>	<i>28</i>
2.1.3 <i>Flow velocity output control.....</i>	<i>29</i>
2.1.4 <i>Scale factor for mesh output .....</i>	<i>30</i>
2.1.5 <i>Export mesh to PFLOTRAN mesh format.....</i>	<i>30</i>
2.1.6 <i>Example of spatial discretization block.....</i>	<i>31</i>
2.2    SPATIAL SELECTION RELATED COMMANDS.....	32
2.2.1 <i>Node selection.....</i>	<i>32</i>
2.2.2 <i>Cell selection .....</i>	<i>35</i>
2.2.3 <i>Constraints for general node selection .....</i>	<i>38</i>
2.2.4 <i>Cell selection constraints for general cell selection .....</i>	<i>38</i>
2.3    FLUX APPROXIMATION RELATED COMMANDS .....	39
2.3.1 <i>Multi-point flux approximation.....</i>	<i>39</i>
2.3.1 <i>Two-point flux approximation .....</i>	<i>40</i>
2.3.2 <i>Multi-point upstream weighting .....</i>	<i>41</i>
2.3.3 <i>Two-point upstream weighting .....</i>	<i>41</i>
2.4    INITIAL CONDITION RELATED COMMANDS .....	41
2.4.1 <i>Initial condition from external file.....</i>	<i>42</i>
2.5    BOUNDARY CONDITION RELATED COMMANDS .....	42
2.5.1 <i>Flux direction across domain boundary.....</i>	<i>42</i>
<b>3    OUTPUT FOR MIN3P-THCM-USG.....</b>	<b>43</b>
3.1    OUTPUT FILES .....	43
3.2    VELOCITY OUTPUT .....	43
<b>REFERENCES .....</b>	<b>45</b>

**If you are new to MIN3P-THCm, please read the general user manual written for structured grid version of MIN3P-THCm and then continue reading this user manual.**

This manual aims at users who are familiar with MIN3P-THCm structured grid version (Mayer 1999, Mayer et al., 2002). It focus on how to use the unstructured grid features of the latest MIN3P-THCm code. The unstructured grid version of MIN3P-THCm is named as MIN3P-THCm-USG. It supports desktop PC, workstation and supercomputer. The code can run on Window, Linux/Unix and OS X operation system. Please note that installation of the required library is system dependent and compiler dependent. Please refer to the related library website for detail information, if needed.

## **1 SYSTEM REQUIREMENT**

MIN3P-THCm-USG was developed based on the ParMIN3P-THCm, a parallel version of MIN3P-THCm. The following libraries are required for specific versions.

CGAL: the computational geometry algorithm library CGAL is only required if built-in mesh generation and 3D zone selection by polyhedral is needed.

PETSc-3.9: MPI and hybrid MPI-OpenMP parallel version of MIN3P-THCm-USG requires PETSc 3.9 and later version, configured with HDF5.

## **2 INPUT FILE FOR MIN3P-THCM-USG**

MIN3P-THCm-USG is based on MIN3P-THCm code and it use the same input file format as MIN3P-THCm. To use the unstructured grid features, user needs to add unstructured grid related commands into the input file. If user needs to run MIN3P-THCm-USG case for structured grid case, just comment all the unstructured grid related commands/data. This chapter gives a basic introduction about MIN3P-THCm-USG input file, including 1) spatial discretization related commands; 2) node/cell spatial selection related commands; and 3) boundary conditions related commands.

### **2.1 SPATIAL DISCRETIZATION RELATED COMMANDS**

The standard spatial discretization for structured grid code consists of keyword ‘spatial discretization’ and parameters for x-direction, y-direction and z-direction. The unstructured grid code can use both structured grid parameters and unstructured grid parameters. If structured grid is provided, the code runs in structured grid mode and if unstructured grid is provided, the code runs in unstructured grid mode. If the structured grid is provided in unstructured grid format, the code run in unstructured grid mode. The unstructured grid code can be generated by built-in mesh generation feature or by other professional software such as Gmsh.

#### **2.1.1 Mesh Format**

The cell-node ordering of provided external mesh, eight in vtk or gid mesh format, should

obey right hand thumb rule. The supported cell-node ordering is shown in Figure 1.

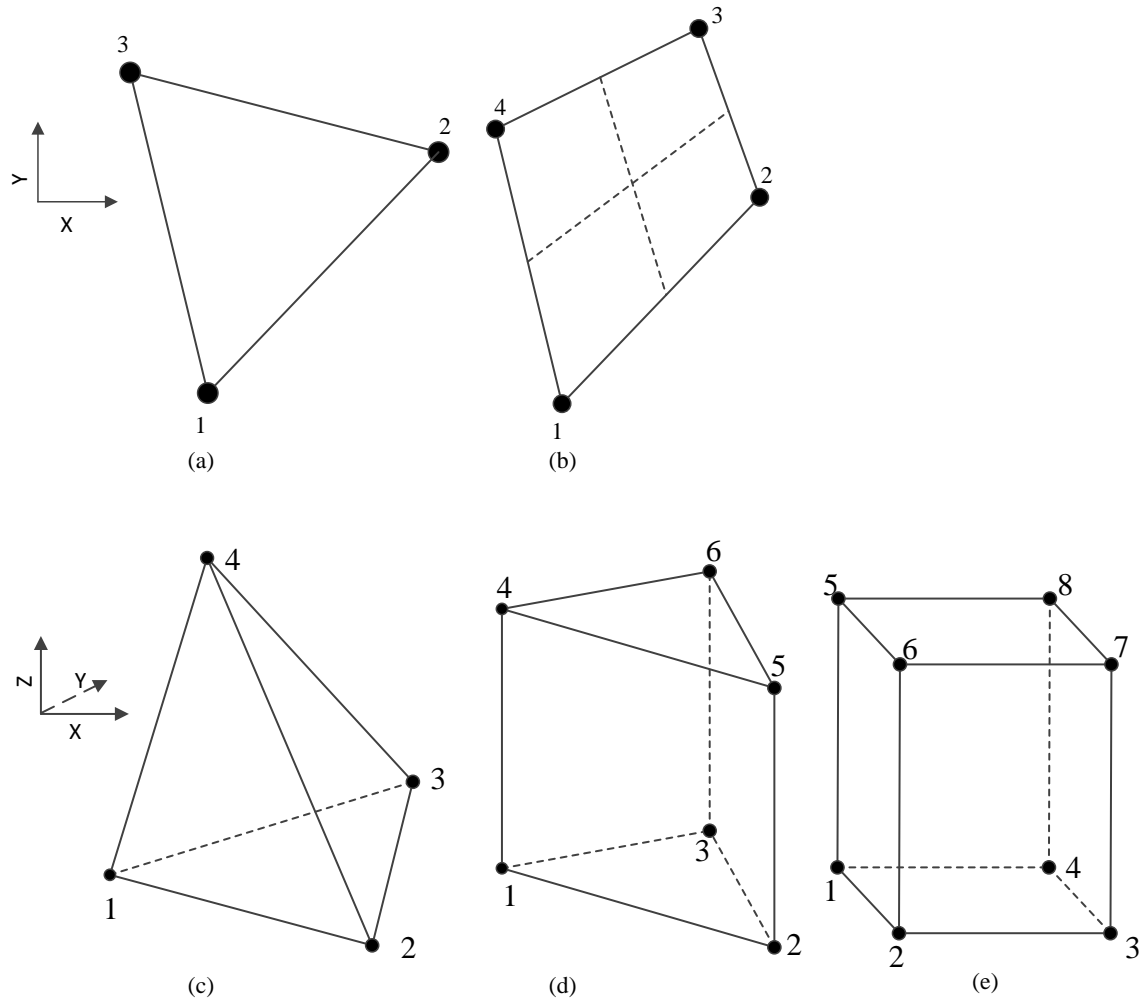


Figure 1 Illustration of cell-node ordering in 2D and 3D Cartesian coordinate system, (1) triangle cell, (b) quadrilateral cell, (c) tetrahedral cell, (d) prism (wedge) cell, and (e) hexahedral cell

### 2.1.2 Built-in mesh generation

Built-in unstructured grid mesh generation supports 2d triangulation, 3d triangulation, 2d constrained mesh generation, 3d polyhedron domain mesh generation, 3d multi domain mesh generation, 3d mesh generation based on gms mesh and elevation dataset. The default output file of mesh generation is input-prefix.vtk. For the format of vtk, please visit <https://www.vtk.org/wp-content/uploads/2015/04/file-formats.pdf>. The following built-in mesh generation features are supported.

#### 2.1.2.1 Keywords

*'generate unstructured grid from structured parameters'*

or

*'generate unstructured grid from structured boundary'*

or

*'generate unstructured grid from file'*

[inputfile-prefix.usg] ; self-described unstructured grid generation file

or

*'use unstructured grid method'*

*'structured spatial discretization'*

This command is optional if mesh is read from external file. It is only required if mesh is generated from structured parameters or structured boundary.

The command *'generate unstructured grid from structured parameters'* triangulates structured grid into unstructured grid mesh by connecting the diagonal nodes of rectangular panel (2D) to build triangular mesh, or by connecting the diagonal nodes of rectangular box (3D) to build tetrahedral mesh; the command *'generate unstructured grid from structured boundary'* generates triangular mesh (2D) or tetrahedral mesh (3D) by using the averaged size of *delt\_x*, *delt\_y*, and *delt\_z* as control parameters of the triangular mesh cell (2D) or tetrahedral cell (3D); and the command *'generate unstructured grid from external file'* generates mesh based on the given self-described unstructured grid generation file, including the boundaries and constrains. The name of this self-described script is always *inputfile-prefix.usg*.

*'use unstructured grid method'*

The command *'use unstructured grid method'* is optional. It can be used to run simulation based on structured grid using unstructured grid method and generate results of spatial output in vtk format. An example is shown below.

*'spatial discretization'*

*'use unstructured grid method'*

*'structured spatial discretization'*

1 ;number of discretization intervals in x

21 ;number of control volumes in x

0. 1.0 ;xmin,xmax

1 ;number of discretization intervals in y

1 ;number of control volumes in y

0. 1.0 ;ymin,ymax

1 ;number of discretization intervals in z

41 ;number of control volumes in z

0. 2.00 ;zmin,zmax

### 2.1.2.2 Requirements

Optional if external mesh file is provided.

### 2.1.2.3 Notes

Mesh generation features is available under Unix/Linux/Mac OS system, not available under Windows OS. Code should be compiled with CGAL 4.7 + library.

### 2.1.2.4 Examples

#### 2.1.2.4.1 Input parameters for mesh generation based on the structured grid boundary

```

!*****
!example for mesh generation based on the structured grid boundary
!*****
'spatial discretization'
'generate unstructured grid from structured boundary'
'structured spatial discretization'
1                ;number of discretization intervals in x
31               ;number of control volumes in x
0.0 1371.6       ;xmin,xmax, 4500 feet
1                ;number of discretization intervals in y
1                ;number of control volumes in y
0. 1.0           ;ymin,ymax
1                ;number of discretization intervals in z
31               ;number of control volumes in z
0.0 914.4        ;zmin,zmax, 3000 feet
'done'
!*****
!end of example for mesh generation based on the structured grid boundary
!*****

```

The generated mesh based on the structured grid boundary is shown in Figure 2.

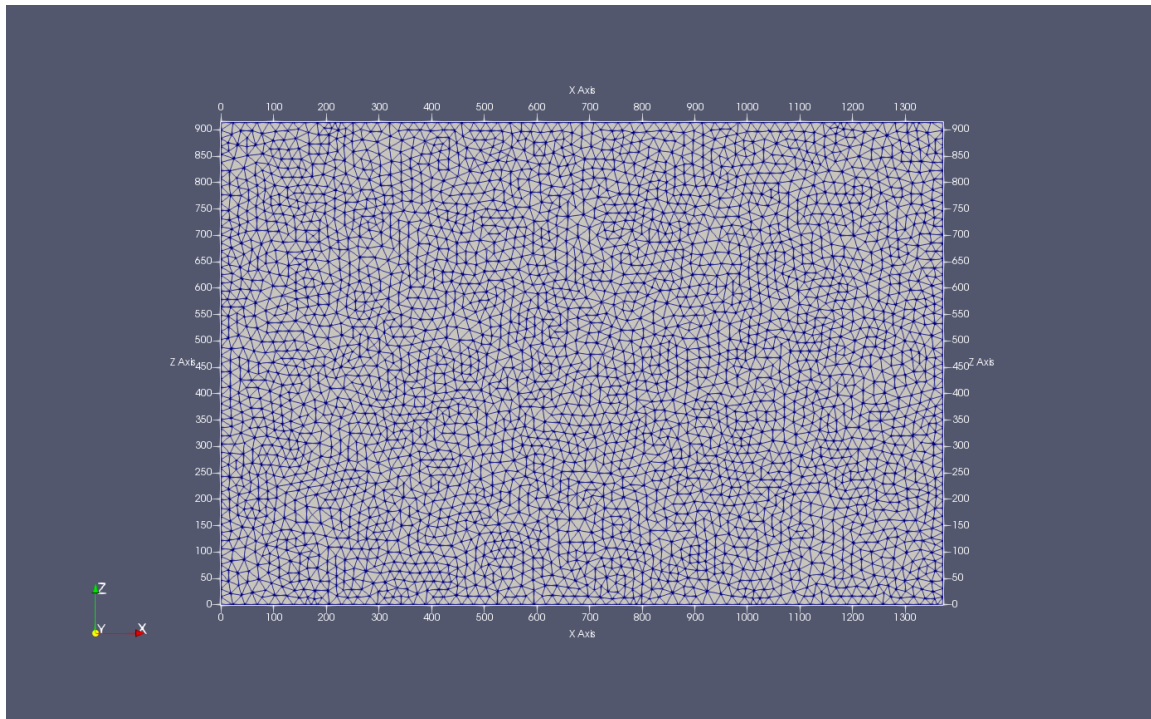


Figure 2 example for mesh generation based on the structured grid boundary

#### 2.1.2.4.2 Input parameters for 2d mesh generation based on self-described script

```
!*****
!input parameters for 2d Delaunay mesh generation
!*****
'polygon 2d delaunay mesh'

'coordinate projection'
'xz'          '!xy' or 'yz' or 'xz'

'number of constraints'
9

!constraint 1
'constraint type'
'closed polygon'

'number of vertices'
4

'vertices coordinate'
1.0, 0.0
0.0, 1.0
-1.0, 0.0
```

0.0, -1.0

*'end of constraint'*

!constraint 2

*'constraint type'*

*'closed polygon'*

*'number of vertices'*

4

*'vertices coordinate'*

3.0, 3.0

-3.0, 3.0

-3.0, -3.0

3.0, -3.0

*'end of constraint'*

!constraint 3

*'constraint type'*

*'open polygon'*

*'number of vertices'*

2

*'vertices coordinate'*

-2.8, 2.0

-2.8, -2.0

*'end of constraint'*

!constraint 4

*'constraint type'*

*'closed polygon with sub discretization interval'*

*'number of vertices'*

3

*'vertices coordinate'*

-2.5, -0.5

-2.5, -1.5

-1.0, -1.5

*'segment discretization interval'*

0.025



0.010

0.005

*'end of constraint'*

!constraint 5

*'constraint type'*

*'closed polygon with sub discretization size'*

*'number of vertices'*

3

*'vertices coordinate'*

2.5, -0.5

2.5, -1.5

1.0, -1.5

*'segment discretization size'*

50

100

200

*'end of constraint'*

!constraint 6

*'constraint type'*

*'open polygon with sub discretization interval'*

*'number of vertices'*

2

*'vertices coordinate'*

2.5, 0.0

3.0, 0.0

*'segment discretization interval'*

0.02

*'end of constraint'*

!constraint 7

*'constraint type'*

*'open polygon with sub discretization size'*

*'number of vertices'*

2

*'vertices coordinate'*

0.0, 2.0

0.0, 3.0

*'segment discretization size'*

20

*'end of constraint'*

!constraint 8

*'constraint type'*

*'closed arc with sub discretization size'*

*'number of vertices'*

3

*'vertices coordinate'*

-1.0, -1.0

1.0, -1.0

1.0, 1.0

*'arc discretization size'*

0.025

*'end of constraint'*

!constraint 9

*'constraint type'*

*'open arc with sub discretization size'*

*'number of vertices'*

3

*'vertices coordinate'*

2.5, 0.0

0.0, 2.5

-2.5, 0.0

*'arc discretization size'*

0.05

*'end of constraint'*

!set seed where the triangulation is excluded

*'number of seeds'*

2

*'seeds coordinate'*

0.0, 0.0

2.0, -1.4

*'global domain mesh criteria'*

0.125, 0.25

!shape criterion bound,default 0.125 corresponds to

!around 20.6 degree, size bound

*'mesh optimization iterations'*

10

!\*\*\*\*\*

!end of input parameters for 2d Delaunay mesh generation

!\*\*\*\*\*

The generated mesh based on the given external mesh script is shown in Figure 3.

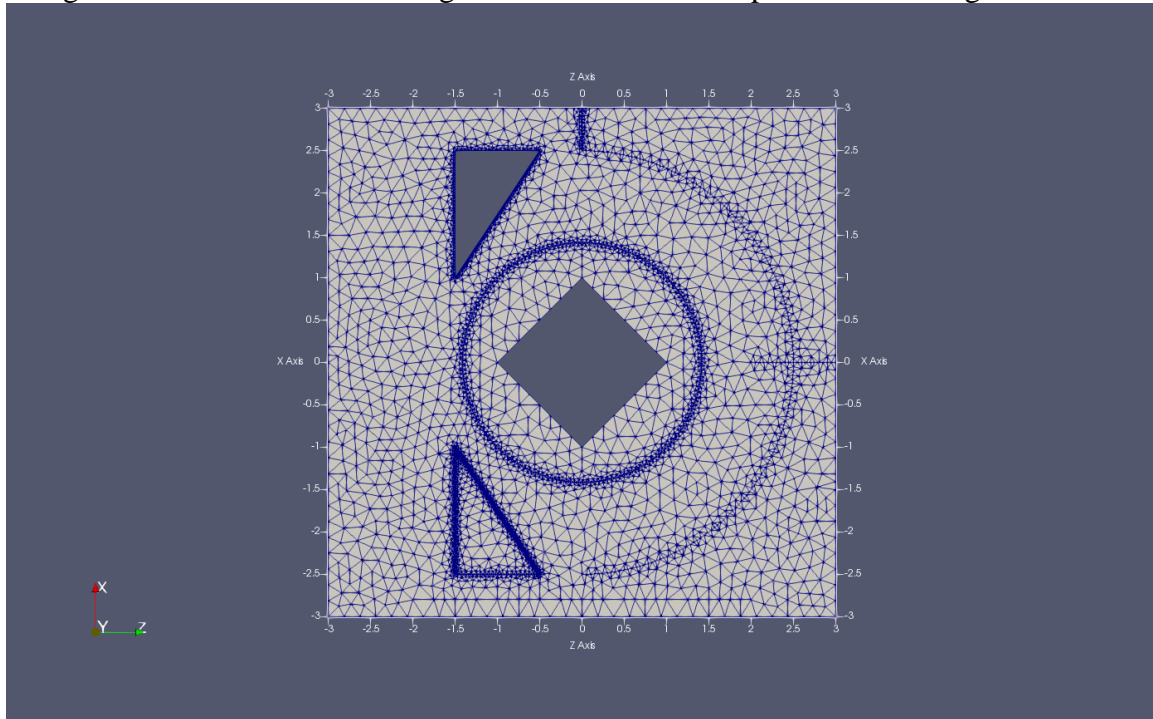


Figure 3 example for mesh generation based on the external mesh script

## 2.1.2.4.3 Input parameters for 2d delaunay mesh with mixed constraints

!\*\*\*\*\*

! input parameters for 2d delaunay mesh with mixed constraints

!\*\*\*\*\*

*'polygon 2d delaunay mesh'*

'coordinate projection'  
 'xz'                   !'xy' or 'yz' or 'xz'

'number of constraints'  
 7

!constraint : mixed boundary  
 'constraint type'  
 'closed mixed constraint'

!constraint 1: outer boundary  
 'constraint type'  
 'open polygon with sub discretization interval'

'number of vertices'  
 3

'vertices coordinate'  
 0.5 0.0  
 1.0 0.0  
 1.0 1.0

'segment discretization interval'  
 0.01  
 0.01

'end of constraint'

!constraint 2: arc outer boundary  
 'constraint type'  
 'open arc with sub discretization size'

'number of vertices'  
 3

'vertices coordinate'  
 1.0   1.0  
 0.647 1.293  
 0.5   2.0

'arc discretization size'  
 0.01

'end of constraint'

!constraint 3: outer boundary

*'constraint type'*

*'open polygon with sub discretization interval'*

*'number of vertices'*

3

*'vertices coordinate'*

0.5 2.0

0.0 2.0

0.0 1.0

*'segment discretization interval'*

0.01

0.01

*'end of constraint'*

!constraint 4: arc outer boundary

*'constraint type'*

*'open arc with sub discretization size'*

*'number of vertices'*

3

*'vertices coordinate'*

0.0 1.0

0.353 0.707

0.5 0.0

*'arc discretization size'*

0.025

*'end of constraint'*

*'end of mixed constraint'*

!constraint 5: line constraint

*'constraint type'*

*'open polygon with sub discretization interval'*

*'number of vertices'*

2

*'vertices coordinate'*

0.25 1.0

0.25 2.0

*'segment discretization interval'*

0.02

*'end of constraint'*

!constraint 6: line constraint

*'constraint type'*

*'open polygon with sub discretization interval'*

*'number of vertices'*

2

*'vertices coordinate'*

0.75 0.0

0.75 1.0

*'segment discretization interval'*

0.02

*'end of constraint'*

!constraint 7: inner boundary

*'constraint type'*

*'closed arc with sub discretization size'*

*'number of vertices'*

3

*'vertices coordinate'*

0.5 1.125

0.625 1.0

0.5 0.875

*'arc discretization size'*

0.01

*'end of constraint'*

!set seed where the triangulation is excluded

*'number of seeds'*

1

*'seeds coordinate'*

0.5, 1.0

*'global domain mesh criteria'*

33.0, 0.02

;shape criterion bound, size bound

*'mesh optimization iterations'*

10

!\*\*\*\*\*

!end of input parameters for 2d delaunay mesh with mixed constraints

!\*\*\*\*\*

The generated mesh based on the given external mesh script with mixed constraints is shown in Figure 4.

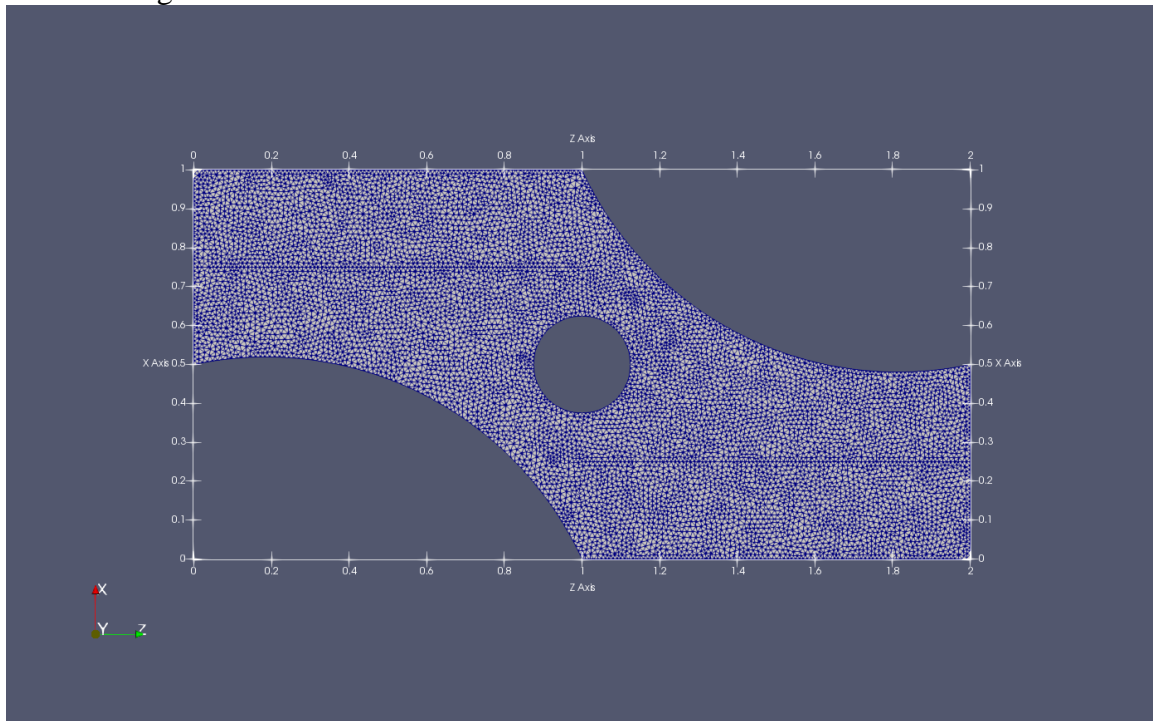


Figure 4 example for mesh generation based on the external mesh script with mixed constraints

#### 2.1.2.4.4 Input parameters for 3d delaunay mesh script

!\*\*\*\*\*

!example input parameters for 3d delaunay mesh generation

!\*\*\*\*\*

*'polyhedra 3d delaunay mesh'*

*'domain file path'*

*'domain\_polyhedron.off'*

!the polyhedron is provided in off format

*'global domain mesh criteria'*

0.25, 25.0, 0.25, 0.25, 2.0, 0.25 !mesh criteria in edge\_size, facet\_angel, facet\_size,  
!facet\_distance, cell\_radius\_edge\_ratio, global size

*'lloyd optimization parameters'*

10, 10.0 !maximum iteration and time limit

*'exude optimization parameters'*

10.0, 10.0 !sliver bound and time limit

*'perturb optimization parameters'*

10.0, 10.0 !sliver bound and time limit

!\*\*\*\*\*

!end of input parameters for 3d delaunay mesh generation

!\*\*\*\*\*

!\*\*\*\*\*

!example for domain\_polyhedron.off

!\*\*\*\*\*

OFF

8 6 0

-1.4603 -1.25555 0.500000

1.4603 -1.25555 0.500000

-1.4603 1.25555 0.500000

1.4603 1.25555 0.500000

-1.8603 1.45555 -0.500000

1.8603 1.65555 -0.500000

-1.8603 -1.45555 -0.500000

1.8603 -1.65555 -0.500000

4 0 1 3 2

4 2 3 5 4

4 4 5 7 6

4 6 7 1 0

4 1 7 5 3

4 6 0 2 4

!\*\*\*\*\*

!end of example for domain\_polyhedron.off

!\*\*\*\*\*

The generated mesh based on the given polyhedron surface mesh is shown in Figure 5.



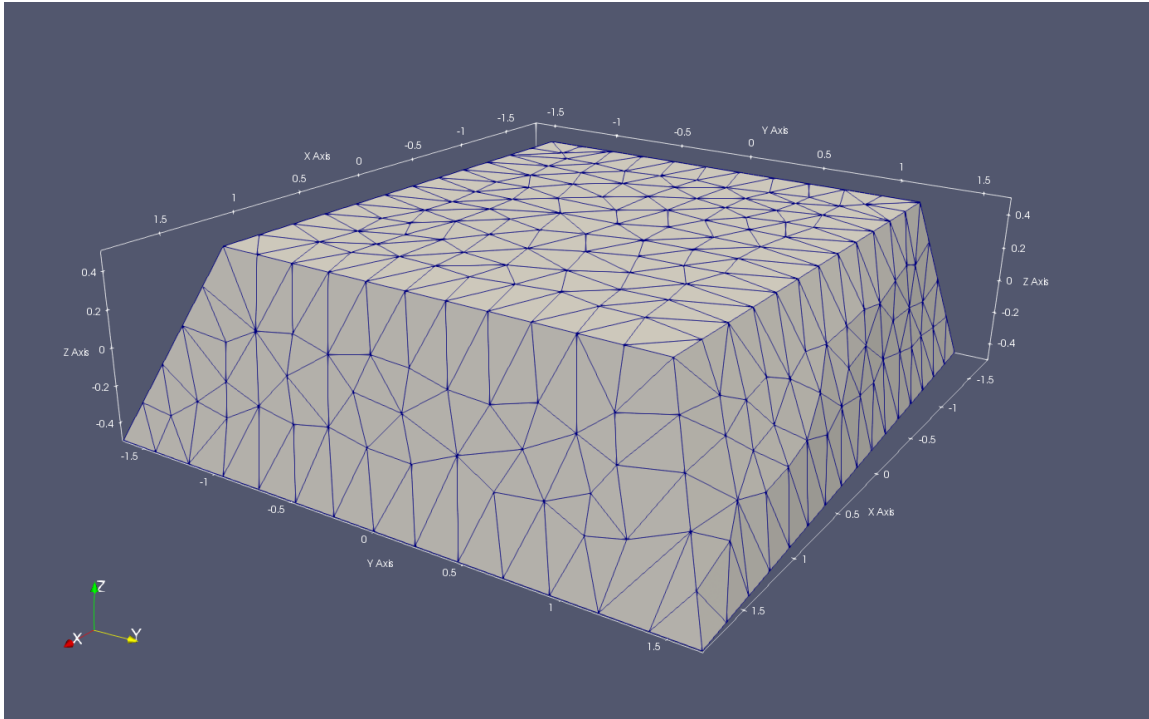


Figure 5 example for mesh generation based on the given polyhedron surface mesh file.

#### 2.1.2.4.5 Input parameters for multidomain 3d delaunay mesh script

```
!*****
!input parameters for multidomain 3d delaunay mesh generation
!*****
```

*'multidomain 3d delaunay mesh'*

*'global domain bottom left coordinates'*  
0.0, 0.0, 0.0

*'global domain length in xyz direction'*  
128.0, 128.0, 128.0

*'global domain minimum discretization in xyz direction'*  
1.0, 1.0, 1.0

*'global domain mesh criteria'*  
5.0, 25.0, 5.0, 5.0, 5.0, 5.0   !mesh criteria in edge\_size, facet\_angle, facet\_size  
!facet\_distance, cell\_radius\_edge\_ratio, global\_size

*'number of subdomains'*  
4

*'subdomain id'*

*1*

*'subdomain name'*

*'sub-1'*

*'subdomain discretization size'*

*5.0*

*'subdomain file path'*

*'../multi\_domains/subdomain-1.off'*

*'end of subdomain'*

*'subdomain id'*

*2*

*'subdomain name'*

*'sub-2'*

*'subdomain discretization size'*

*4.0*

*'subdomain file path'*

*'../multi\_domains/subdomain-2.off'*

*'end of subdomain'*

*'subdomain id'*

*3*

*'subdomain name'*

*'sub-3'*

*'subdomain discretization size'*

*3.0*

*'subdomain file path'*

*'../multi\_domains/subdomain-3.off'*

*'end of subdomain'*

*'subdomain id'*

*4*

*'subdomain name'*

*'sub-4'*

*'subdomain discretization size'*

2.0

*'subdomain file path'*

*'../multi\_domains/subdomain-4.off'*

*'end of subdomain'*

*'number of polyline features'*

4

*'polyline type'*

*'open'*

*'number of vertices'*

2

*'vertices coordinate'*

0.44,0.44,0.44

0.44,0.44,127.05

*'end of polyline'*

*'polyline type'*

*'open'*

*'number of vertices'*

2

*'vertices coordinate'*

127.05, 0.44, 0.44

127.05, 0.44, 127.05

*'end of polyline'*

*'polyline type'*

*'open'*

*'number of vertices'*

2

*'vertices coordinate'*

127.05, 127.05, 0.44

127.05, 127.05, 127.05

*'end of polyline'*

*'polyline type'*

*'open'*

*'number of vertices'*

2

*'vertices coordinate'*

0.44, 127.05, 0.44

0.44, 127.05, 127.05

'end of polyline'

'lloyd optimization parameters'

10, 10.0 !maximum iteration and time limit

'exude optimization parameters'

10.0, 10.0 !sliver bound and time limit

'perturb optimization parameters'

10.0, 10.0 !sliver bound and time limit

!\*\*\*\*\*

!end of input parameters for multidomain 3d delaunay mesh

!\*\*\*\*\*

The subdomain off files are not included here. The final mesh generated is shown in Figure 6(a).

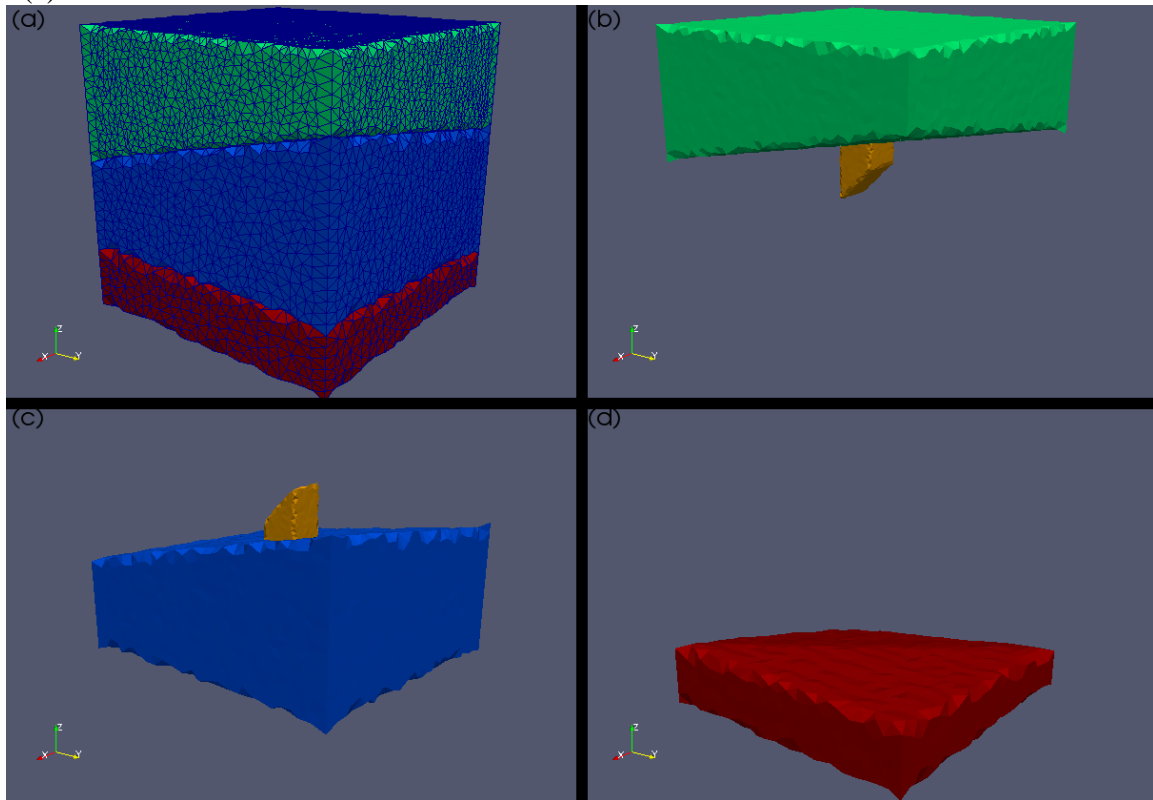


Figure 6 example for mesh generation based on multiple subdomains

#### 2.1.2.4.6 Input parameters for gms 3d mesh script

!\*\*\*\*\*

!input parameters for gms 3d mesh generation

!\*\*\*\*\*

!GMS 3d mesh supports the original gms mesh layout or the triangulated  
!prism mesh. The original gms mesh can be 2d triangle mesh or 2d  
!quadrilateral mesh or mixed triangle quadrilateral mesh. The generated  
!3d mesh is either prism or hexahedra mesh or mixed prism hexahedra mesh.

'gms 3d mesh'

'mesh type'

!'original'

!or

'prism'

'gms mesh file path'

'../domain\_gms/grid\_R3.gms'

'number of layers'

39

'gms layer file path from top to bottom'

'../domain\_gms/top\_NOAA\_1234\_7531\_DEM\_BATH\_REGIONAL.txt'

'../domain\_gms/top\_SURF\_R1\_1\_01\_Hamilton\_Gp.txt'

'../domain\_gms/top\_SURF\_R1\_1\_02\_Dundee.txt'

'../domain\_gms/top\_SURF\_R1\_1\_03\_Detroit\_River\_Gp.txt'

'../domain\_gms/top\_SURF\_R1\_1\_04\_Bois\_Blanc.txt'

'../domain\_gms/top\_SURF\_R1\_1\_05\_Bass\_Islands.txt'

'../domain\_gms/top\_SURF\_R1\_1\_06\_G\_Unit.txt'

'../domain\_gms/top\_SURF\_R1\_1\_07\_F\_Unit.txt'

'../domain\_gms/top\_SURF\_R1\_1\_08\_F\_Salt.txt'

'../domain\_gms/top\_SURF\_R1\_1\_09\_E\_Unit.txt'

'../domain\_gms/top\_SURF\_R1\_1\_10\_D\_Unit.txt'

'../domain\_gms/top\_SURF\_R1\_1\_11\_B\_and\_C\_Units.txt'

'../domain\_gms/top\_SURF\_R1\_1\_12\_B\_Anhydrite\_\_Salt.txt'

'../domain\_gms/top\_SURF\_R1\_1\_13\_A\_2\_Carbonate.txt'

'../domain\_gms/top\_SURF\_R1\_1\_14\_A\_2\_Evaporite.txt'

'../domain\_gms/top\_SURF\_R1\_1\_15\_A\_1\_Carbonate-A1\_UC.txt'

'../domain\_gms/top\_SURF\_R1\_1\_15\_A\_1\_Carbonate-A1\_C.txt'

'../domain\_gms/top\_SURF\_R1\_1\_16\_A\_1\_Evaporite-A1\_E.txt'

'../domain\_gms/top\_SURF\_R1\_1\_16\_A\_1\_Evaporite-A0.txt'

'../domain\_gms/top\_SURF\_R1\_1\_17\_Guelph.txt'

'../domain\_gms/top\_THIC\_R1\_1\_17\_Goat\_Island.txt'

'../domain\_gms/top\_THIC\_R1\_1\_17\_Gasport.txt'

'../domain\_gms/top\_THIC\_R1\_1\_17\_Lions\_Head.txt'

'../domain\_gms/top\_SURF\_R1\_1\_18\_Reynales\_\_Fossil\_Hill.txt'

```
'../domain_gms/top_SURF_R1_1_19_Cabot_Head.txt'
'../domain_gms/top_SURF_R1_1_20_Manitoulin.txt'
'../domain_gms/top_SURF_R1_1_21_Queenston.txt'
'../domain_gms/top_SURF_R1_1_22_Georgian_Bay_Blue_Mtn.txt'
'../domain_gms/top_SURF_R1_1_23_Cobourg.txt'
'../domain_gms/top_SURF_R1_1_24_Sherman_Fall.txt'
'../domain_gms/top_SURF_R1_1_25_Kirkfield.txt'
'../domain_gms/top_SURF_R1_1_26_Coboconk.txt'
'../domain_gms/top_SURF_R1_1_27_Gull_River.txt'
'../domain_gms/top_SURF_R1_1_28_Shadow_Lake.txt'
'../domain_gms/top_SURF_R1_1_29_Cambrian.txt'
'../domain_gms/top_SURF_R1_1_30_Precambrian.txt'
'../domain_gms/top_SURF_R1_1_31_UW-Precambrian.txt'
'../domain_gms/top_SURF_R1_1_32_Bot-Precambrian.txt'
'../domain_gms/top_SURF_R1_1_33_Bottom.txt'
```

'output layer range'

1, 39

'output surface mesh'

'../domain\_gms/surface.off'

!\*\*\*\*\*

!end of input parameters for gms 3d mesh generation

!\*\*\*\*\*

The detail layer files are ignored here. The final mesh converted from GMS 3D mesh is shown in Figure 7. The converted hexahedral mesh is shown in Figure 7(a) and (c) while the converted prism mesh is shown in Figure 7(b) and (d).

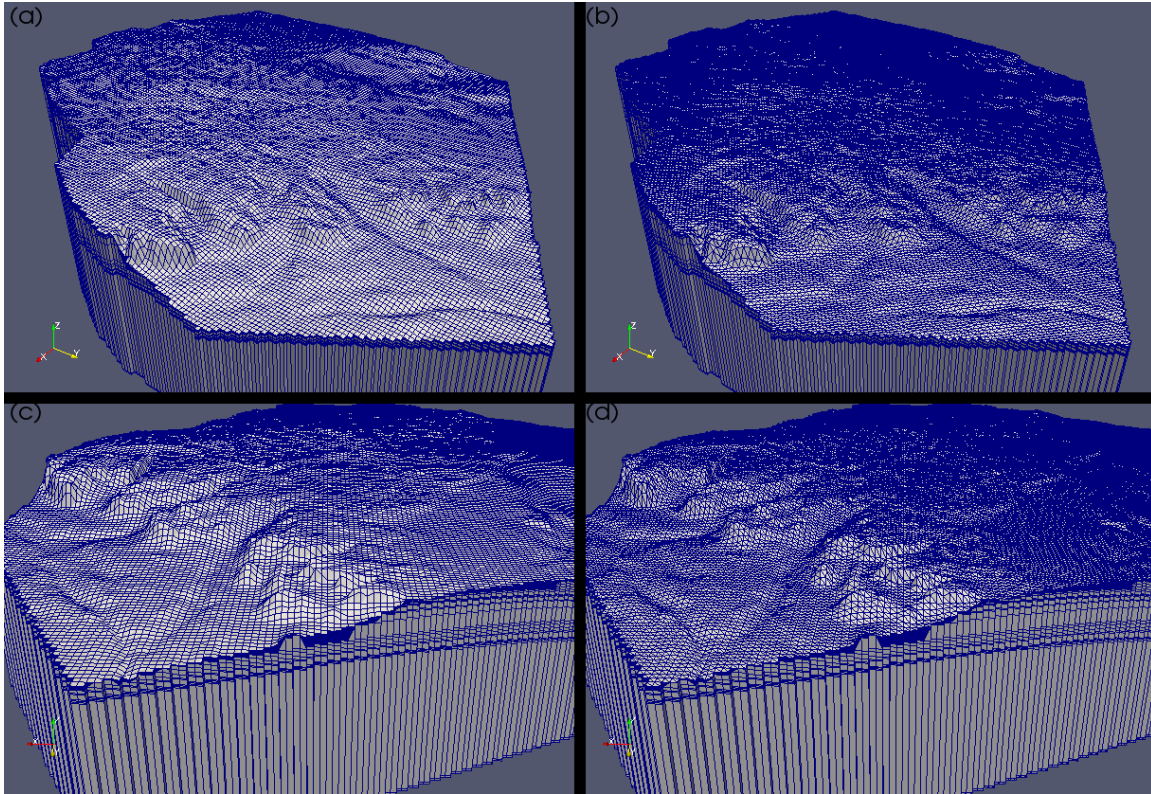


Figure 7 example of mesh converted from GMS mesh files.

### 2.1.3 Read mesh from file

The unstructured grid can read in external file as mesh file. By using external mesh file, built-in mesh generation is not required anymore. Two mesh formats are supported, vtk mesh file and gid mesh file. Vtk mesh file is supported by lots of open source software for both preprocessing and post-processing, e.g., Gmsh, paraview. Gid mesh file is supported by commercial software Gid. For the gid mesh format, please visit <https://www.gidhome.com/>. MIN3P-THCm-USG supports triangle mesh and quadrilateral mesh in 2D, tetrahedron, prism and hexahedron mesh in 3D. It is recommended to use external mesh file as user has more control on mesh quality.

#### 2.1.3.1 Keywords

*'read unstructured grid from file'*

Or

*'read unstructured grid from gid mesh file'*

The command *'read unstructured grid from file'* reads in external mesh file in vtk file format (input-prefix.vtk) and *'read unstructured grid from gid mesh file'* reads in external mesh file in msh file format (input-prefix.msh).

### 2.1.3.2 Requirements

Required if built-in mesh generation is not used.  
Optional if built-in mesh generation is used.

### 2.1.3.3 Notes

The coordinates in the mesh file should follow right-hand rule. For triangular or quadrilateral mesh, all the cell-node numbering should be in counterclockwise ordering.

### 2.1.3.4 Examples

*'read unstructured grid from file'*

#### 2.1.4 Control volume method

For the vertex-centered approach, there are two methods available to define the control volume. The most common is to join the midpoint of each edge of the primary grid with the center of the control volume. This method is often referred to as the median-dual method (MD), as shown in Figure 8(a). Another option is to use the Voronoi diagram (VD) method, as shown in Figure 8(b). In this case, the division points of the dual grid are the circumcentres of the respective elements. For the VD method, the segment that joins the two adjacent grids is perpendicular to their common edge and crosses this edge at the midpoints. However, this method is only valid for triangular meshes in 2D. An additional requirement for this method is that triangular faces in 2D cannot have obtuse angles.

Both the median-dual control volume method and the Voronoi diagram control volume method are available from the current code. The discretization of the VD method is similar to the discretization of structured grid methods, with all control volume interfaces orthogonal to the node connections (edges). To make the approach compatible with different cell types, the MD method is used to define dual control volumes throughout this report and the VD method is treated as a reduced form (subset) of the MD method.

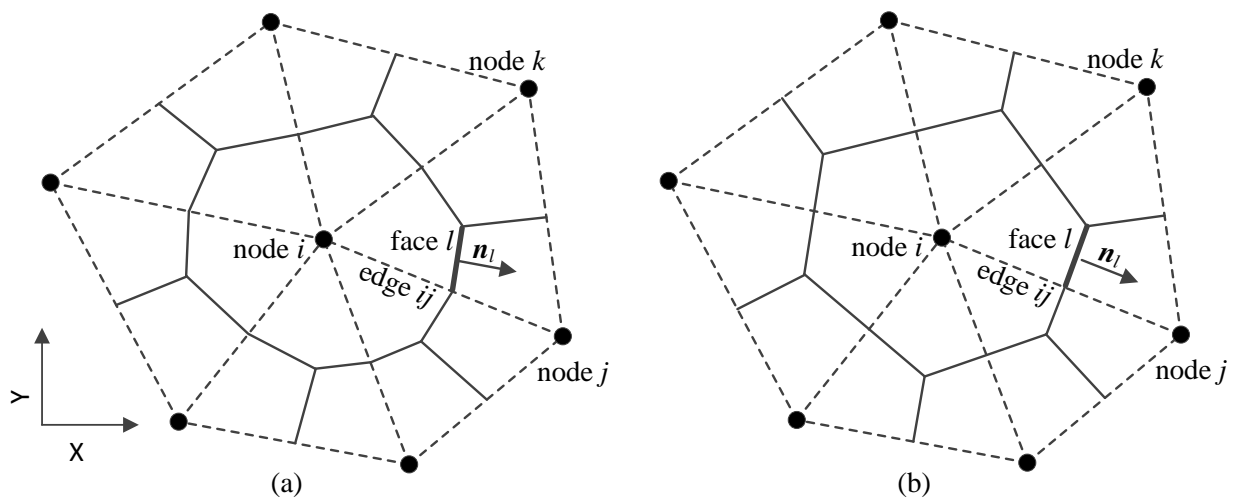




Figure 8 Finite volume dual grid constructed from a primary triangular grid according to (a) the median-dual method and (b) the Voronoi diagram method

#### 2.1.4.1 Keywords

*'control volume method'*

*'voronoi diagram'*

Or

*'median dual'*

*'allow obtuse cells'*

#### 2.1.4.2 Requirements

By default, *'median dual'* method is used as this method is available for all the cell types. *'voronoi diagram'* is only available for triangular mesh (2D) and tetrahedral mesh (3D). Generally, *'voronoi diagram'* method has limitation that there is no obtuse cell (cell whose circumcenter is out of itself) in the mesh. In MIN3P-THCm-USG code, obtuse cell is allowed for VD method that mid-point of the longest triangle edge (2D) or center of the largest cell face is treated as the 'circumcentres'.

#### 2.1.4.3 Notes

For high quality mesh, either triangular mesh (2D) or tetrahedral mesh (3D), VD method is preferable as this method generally cause less cross diffusion error.

#### 2.1.4.4 Examples

*'control volume method'*

*'voronoi diagram'*

*'allow obtuse cells'*

### 2.1.5 Gradient reconstruction method

Gradient reconstruction is an internal algorithm to estimate the gradient on the control volume interface. The code supports the following methods: *'least square'*, *'green gauss'*, *'least square second order'*, *'least square third order'*, and *'least square fourth order'*. *'least square'* and *'green gauss'* methods estimate gradient based on the node and other nodes linked to this node and then interpolate gradient of control volume interface by the surrounding nodes. *'least square second order'* to *'least square fourth order'* estimate gradient based on cell and other cells surrounding this cell and the gradient of the control volume interface is calculated directly without interpolation. *'least square third order'* and *'least square fourth order'* also provide high order flux correction term.

**2.1.5.1 Keywords**

*'gradient reconstruction method'*

*'least square'*

Or

*'green gauss'*

Or

*'least square second order'*

Or

*'least square third order'*

Or

*'least square fourth order'*

**2.1.5.2 Requirements**

Optional if use the default *'least square'* gradient reconstruction method.

Required if use other gradient reconstruction methods.

**2.1.5.3 Notes**

Gradient reconstruction method *'green gauss'* only supports 2D triangular mesh. Other gradient reconstruction methods can support all cell types. Generally, *'least square second order'* to *'least square fourth order'* methods require more memory and computing time than *'least square'* and *'green gauss'* methods. For high quality mesh, all these methods can generate same results while for low quality mesh, they have different behavior.

**2.1.5.4 Examples**

*'gradient reconstruction method'*

*'green gauss'*

**2.1.6 Gradient reconstruction spatial weighting**

Gradient reconstruction spatial weighting is a control parameter for *'least square'*, *'least square second order'* to *'least square fourth order'*. It uses inverse distance weighting method. User can select different weighting factor.

**2.1.6.1 Keywords**

*'gradient reconstruction spatial weighting'*

*'inverse distance weighting factor'*

1

**2.1.6.2 Requirements**

Not required for *'green gauss'* gradient reconstruction method.

Optional for *'least square'* and *'least square second order'* to *'least square fourth order'* gradient reconstruction methods.

*'inverse distance weighting factor'* is also an optional option. The default inverse distance weighting factor is 1.

### 2.1.6.3 Notes

### 2.1.6.4 Examples

*'gradient reconstruction spatial weighting'*

*'inverse distance weighting factor'*

2

### 2.1.7 Gradient interpolation

By default, gradient reconstruction is calculated based on the mid-point of the edge that links two nodes. User can also select separated control volume interface as the point where gradient is calculated or interpolated. Gradient interpolation works only for *'green gauss'* and *'least square'* methods. By default, only two nodes (edge points) are used to interpolate the gradient at the edge mid-point (or separated interface centers). User can select gradient interpolation based on neighbor cells.

#### 2.1.7.1 Keywords

*'gradient interpolation based on neighbour cells'*

*'use separated interface centers'*

or

*'use control volume half face center'*

#### 2.1.7.2 Requirements

*'gradient interpolation based on neighbour cells'* is an optional command for *'least square'* and *'green gauss'* gradient reconstruction methods. By default, this option is not used if the keyword is not provided.

*'use separated interface centers'* is an optional command for all gradient reconstruction methods. By default, edge mid-point is used instead of separated interface centers.

#### 2.1.7.3 Notes

Generally, select edge mid-point is a better choice for most of cases. *'use separated interface centers'* causes bigger numerical error in flux calculation when mesh quality is not good. For structured grid method, all the gradient is calculated at the edge mid-point. *'use control volume half face center'* has the same effect as *'use separated interface centers'*

#### 2.1.7.4 Examples

*'gradient interpolation based on neighbour cells'*

*!'use separated interface centers'*

#### 2.1.8 Gradient reconstruction flux correction

Unlike structured grid that the control volume interface is always perpendicular to the edge, the control volume interface is not always perpendicular to the edge. In this case, cross diffusion error will be caused when calculate the interface flux. By including cross diffusion term, this error can be reduced. Control of cross diffusion term is available for all the gradient reconstruction methods. For fine mesh, especially when mesh is orthogonal, the cross diffusion terms can self-canceled.

For the flux calculation, by default second order Taylor's theorem is used. User can include higher order correction term for this calculation. This command is only available for *'least square third order'* and *'least square fourth order'*

##### 2.1.8.1 Keywords

*'include cross diffusion term'*

or

*'exclude cross diffusion term'*

*'include high order flux correction term'*

or

*'exclude high order flux correction term'*

Separated control of cross diffusion term for flow, heat transport and reactive transport is also available. The following keywords are used to include or exclude cross diffusion term for flow, heat transport and reactive transport.

*'include cross diffusion term for flow'*

or

*'exclude cross diffusion term for flow'*

*'include cross diffusion term for heat transport'*

or

*'exclude cross diffusion term for heat transport'*

*'include cross diffusion term for reactive transport'*

or

*'exclude cross diffusion term for reactive transport'*

##### 2.1.8.2 Requirements

*'include cross diffusion term'* is optional. By default, cross diffusion term is used for flow,

heat transport and reactive transport.

*'include high order flux correction term'* works only for high order least square method. By default, this option is not used.

### 2.1.8.3 Notes

### 2.1.8.4 Examples

*'include cross diffusion term'*

*'exclude cross diffusion term in flow'*

*'include high order flux correction term'*

### 2.1.9 Optional cell based parameters

MIN3P-THCm structured grid version assign all the physical parameters to the node. For the interface flux calculation, those physical parameters, e.g., hydraulic conductivity, need to be averaged. MIN3P-THCm-USG provides options to assign those physical parameters to the cell to avoid parameter averaging on the control volume interface. Available parameters include hydraulic conductivity, permeability, relative permeability, dispersivity for both heat transport and reactive transport.

#### 2.1.9.1 Keywords

*'cell based hydraulic conductivity'*

*'cell based permeability'*

*'cell based relative permeability'*

*'cell based dispersivity'* ;Set dispersivity of both heat and reactive transport to true

*'cell based dispersivity for heat transport'*

*'cell based dispersivity for reactive transport'*

#### 2.1.9.2 Requirements

All the *'cell based parameters'* are optional. By default, *'node based parameters'* are used.

### 2.1.9.3 Notes

### 2.1.9.4 Examples

*'cell based hydraulic conductivity'*

### 2.1.10 Read initial condition from vtk file

By default, MIN3P-THCm-USG read the data files in the same format as structured grid. The difference is that the node ordering in the data files should be the same as the node ordering in the mesh file.

#### 2.1.10.1 Keywords

*'read initial condition from vtk file'*

#### 2.1.10.2 Requirements

Optional, this command is used to read initial condition for flow, density dependent flow and heat transport. The file name should be prefix.ivs.vtk.

#### 2.1.10.3 Notes

#### 2.1.10.4 Examples

*'read initial condition from vtk file'*

### 2.1.1 Read other parameters from vtk file

By default, MIN3P-THCm-USG read the data files in the same format as structured grid. The difference is that the node ordering in the data files should be the same as the node ordering in the mesh file. The following keywords allow user to read parameters from vtk file format for unstructured grid version instead of tecplot data format for structured grid version.

#### 2.1.1.1 Keywords

*'read ... from vtk file'*

#### 2.1.1.2 Requirements

Optional, this command is used to read specified parameters (e.g., mineral volume fraction) The file name should be prefix.abc.vtk where 'abc' is the extension file name used for the structured grid version.

#### 2.1.1.3 Notes

#### 2.1.1.4 Examples

*'read initial mineral volume fractions from vtk file'*

### 2.1.2 Cell-node ordering control

By default, the counterclockwise cell-node ordering is required. In case the cell-node

ordering is clockwise, user can use the following keyword to reverse the node ordering.

#### 2.1.2.1 Keywords

*'reverse cell-node ordering'*

#### 2.1.2.2 Requirements

Optional, this command is used for mesh with clockwise cell-node ordering.

#### 2.1.2.3 Notes

### 2.1.3 Flow velocity output control

Different from structured grid code that exports velocity at the control volume interface, the unstructured grid code exports velocity at the nodes. This is mainly because the unstructured grid has much more complex structure than structured grid. So as to keep all the output in the same data format, the velocity of each node is interpolated from the control volume interface flux directly linked to this node. However, this interpolation sometimes may cause numerical error when the gradient around this node, or hydraulic conductivity around this node, is highly different. In this case, user can use the gradient calculated directly based on the gradient reconstruction method and use the minimum hydraulic conductivity around this node to calculate the velocity based on Darcy's law. This can help to avoid extreme value caused by velocity interpolation.

#### 2.1.3.1 Keywords

*'variation coefficient for velocity average'*

*0.5d0 ; below this threshold, use velocity interpolation*

#### 2.1.3.2 Requirements

Optional, this command is used to avoid extreme velocity value caused by velocity interpolation. If variation coefficient of interface velocity around the node is smaller than the threshold, use velocity interpolation, otherwise, calculate the velocity based on the hydraulic head gradient and hydraulic conductivity of this node based on Darcy's law. By default, threshold 0.5d0 is used.

#### 2.1.3.3 Notes

#### 2.1.3.4 Examples

*'variation coefficient for velocity average'*

*0.5d0 ; below this threshold, use velocity interpolation*

Alternatively, user can choose Perot's method (Blair Perot, 2000, Conservation properties of unstructured staggered mesh schemes, Journal of Computational Physics, 159, 58-89) for variably saturated flow velocity output. It was observed that this method does not

always produce good reconstruction, e.g., for `stript_usg` case, the reconstruction velocity is not as good as previous version which is based on the gradient reconstruction of the node. Also, the advective and diffusive velocity in reactive transport need further check.

#### 2.1.3.5 Keywords

*'use perot velocity reconstruction'*

#### 2.1.3.6 Requirements

Optional. Try this option when *'variation coefficient for velocity average'* does not work well.

#### 2.1.3.7 Notes

#### 2.1.3.8 Examples

*'use perot velocity reconstruction'*

### 2.1.4 Scale factor for mesh output

Mesh coordinates can be scaled for spatial output. This is optional as user can use postprocessing software to do mesh scaling.

#### 2.1.4.1 Keywords

*'scale factor for mesh output'*

*1.0 1.0 10.0 ;scale factor for x, y and z axis.*

#### 2.1.4.2 Requirements

Optional.

#### 2.1.4.3 Notes

#### 2.1.4.4 Examples

*'scale factor for mesh output'*

*1.0 1.0 10.0 ;scale factor for x, y and z axis.*

### 2.1.5 Export mesh to PFLOTTRAN mesh format

MIN3P-THCm-USG can export mesh to PFLOTTRAN mesh format so that user can verify the code using the same mesh. Please note PFLOTTRAN uses cell-centered control volume method, which is different with MIN3P-THCm's node-centered control volume method.



**2.1.5.1 Keywords**

*'export mesh to pflotran mesh format'*

**2.1.5.2 Requirements**

Optional, this command is used to generate pflotran mesh file for verification purpose only.

**2.1.5.3 Notes****2.1.5.4 Examples**

*'export mesh to pflotran mesh format'*

**2.1.6 Example of spatial discretization block**

! Data Block 3: spatial discretization

! -----

!

*'spatial discretization'*

!'generate unstructured grid from structured parameters'

!'generate unstructured grid from structured boundary'

!'generate unstructured grid from file'

*'read unstructured grid from file'*

!'read unstructured grid from gid mesh file'

*'gradient reconstruction method'*

!'least square third order'

*'green gauss'*

*'include cross diffusion term'*

*'include high order flux correction term'*

*'gradient reconstruction spatial weighting'*

*'inverse distance weighting factor'*

2

!'gradient interpolation based on neighbour cells'

*'control volume method'*

*'voronoi diagram'*

!'median dual'

*'allow obtuse cells'*

*'cell based hydraulic conductivity'*

*'cell based dispersivity for reactive transport'*

*'variation coefficient for velocity average'*

0.5d0 ; below this threshold, use velocity interpolation

*'structured spatial discretization'*

1 ;number of discretization intervals in x

31 ;number of control volumes in x

0.0 1371.6 ;xmin,xmax, 4500 feet

1 ;number of discretization intervals in y

1 ;number of control volumes in y

0. 1.0 ;ymin,ymax

1 ;number of discretization intervals in z

31 ;number of control volumes in z

0.0 914.4 ;zmin,zmax, 3000 feet

*'done'*

## 2.2 SPATIAL SELECTION RELATED COMMANDS

MIN3P-THCm-USG can use 'extent of zone' in the same way as MIN3P-THCm structured grid version. It also supports advanced node/cell selection by polygon (2D), polyhedron (3D), ID list and other constraints.

### 2.2.1 Node selection

#### 2.2.1.1 Keywords

*'extent of zone: boundary nodes in polygon'* ;2D only

Only boundary nodes that locate in the given polygon are selected

*'extent of zone: internal nodes in polygon'* ;2D only

Only internal nodes that locate in the given polygon are selected

*'extent of zone: all nodes in polygon'* ;2D only

All nodes that locate in the given polygon are selected

*'extent of zone: boundary nodes in polyhedron'* ;3D, need CGAL

Only boundary nodes that locate in the given polyhedron are selected

*'extent of zone: internal nodes in polyhedron'* ;3D, need CGAL

Only internal nodes that locate in the given polyhedron are selected

*'extent of zone: all nodes in polyhedron'* ;3D, need CGAL

All nodes that locate in the given polyhedron are selected

*'extent of zone: boundary nodes by id'* ;2D and 3D

Only boundary nodes with listed node ids are selected

*'extent of zone: internal nodes by id'* ;2D and 3D

Only internal nodes with listed node ids are selected

*'extent of zone: all nodes by id'* ;2D and 3D

All nodes with listed node ids are selected

The above commands should be followed by entity (polygon, polyhedron, or id list) data, either in the input file or from external file, as shown below. Note polyhedron data file can only be read from external file due to its complexity.

*'read data'*

*...data list...*

!or

*'read data from file'*

*...data file path...*

### 2.2.1.2 Requirements

The standard rectangular panel/box selection 'extent of zone' is also available in MIN3P-THCm-USG. The above command is a better choice for complex geometry.

### 2.2.1.3 Notes

Please note, polyhedron selection needs CGAL library compiled to the code that it is only support Linux/Unix OS.

### 2.2.1.4 Examples

#### 2.2.1.4.1 Example of node selection by polygon

*'extent of zone'*

*0.0 3.0 0.0 0.0 0.0 2.0*

!

! same as

!

*'extent of zone: all nodes in polygon'*

*'read data'*

```

4           ;number of nodes in polygon
0.0 0.0 0.0 ;coordinates of polygon vertices
3.0 0.0 0.0
3.0 0.0 2.0
0.0 0.0 2.0

```

```

!
! same as
!
'extent of zone: all nodes in polygon'
'read data from file'
'polygon.txt'

```

!where polygon.txt is a file with following data

```

4
0.0 0.0 0.0
3.0 0.0 0.0
3.0 0.0 2.0
0.0 0.0 2.0

```

#### 2.2.1.4.2 Example of node selection by polyhedron

```

'extent of zone: boundary nodes in polyhedron'
'read data from file'
'cube.off'

```

```

!
!cube.off is a file with following data input
!

```

```

OFF
# cube.off
# A cube
8 6 12
1.0 0.0 1.0
0.0 1.0 1.0
-1.0 0.0 1.0
0.0 -1.0 1.0
1.0 0.0 0.0
0.0 1.0 0.0
-1.0 0.0 0.0
0.0 -1.0 0.0
4 0 1 2 3
4 7 4 0 3
4 4 5 1 0
4 5 6 2 1
4 3 2 6 7

```

4 6 5 4 7

#### 2.2.1.4.3 Example of node selection by node ids

*'extent of zone: all nodes by id'*

*'read data'*

4 ;number of nodes

2 ;ids

8

10

15

!same as

*'extent of zone: all nodes by id'*

*'read data from file'*

*'idlist.txt'*

idlist.txt is a file with following data

4 ;number of nodes

2 ;ids

8

10

15

### 2.2.2 Cell selection

Cell selection is carried out in a similar way as node selection. If the cell center is inside a given entity, then the cell is treated as inside the given entity. It does not require all the cell nodes to be inside the given entity.

#### 2.2.2.1 Keywords

*'extent of zone: boundary cells in polygon'*

;2D only

Only boundary cells that locate in the given polygon are selected

*'extent of zone: internal cells in polygon'*

;2D only

Only internal cells that locate in the given polygon are selected

*'extent of zone: all cells in polygon'*

;2D only

All cells that locate in the given polygon are selected

*'extent of zone: boundary cells in polyhedron'*

;3D, need CGAL

Only boundary cells that locate in the given polyhedron are selected

*'extent of zone: internal cells in polyhedron'*

;3D, need CGAL

Only internal cells that locate in the given polyhedron are selected

*'extent of zone: all cells in polyhedron'* ;3D, need CGAL  
 All cells that locate in the given polyhedron are selected

*'extent of zone: boundary cells by id'* ;2D and 3D  
 Only boundary cells with listed cell ids are selected

*'extent of zone: internal cells by id'* ;2D and 3D  
 Only internal cells with listed cell ids are selected

*'extent of zone: all cells by id'* ;2D and 3D  
 All cells with listed cell ids are selected

*'read data'*  
*...data list...*  
 !or  
*'read data from file'*  
*...data file path...*

### 2.2.2.2 Requirements

Optional. The above command is a better choice for complex geometry and only applicable to cell selection, e.g., assign cell based hydraulic conductivity.

### 2.2.2.3 Notes

Please note, polyhedron selection needs CGAL library compiled to the code that it is only support Linux/Unix OS.

### 2.2.2.4 Examples

#### 2.2.2.4.1 Example of cell selection by polygon

*'extent of zone'*  
 0.0 3.0 0.0 0.0 0.0 2.0  
  
 !  
 ! same as  
 !  
*'extent of zone: all cells in polygon'*  
*'read data'*  
 4 ;number of nodes in polygon  
 0.0 0.0 0.0 ;coordinates of polygon vertices  
 3.0 0.0 0.0  
 3.0 0.0 2.0  
 0.0 0.0 2.0  
  
 !  
 ! same as

!  
*'extent of zone: all cells in polygon'*  
*'read data from file'*  
*'polygon.txt'*

!polygon.txt is a file with following data

4  
 0.0 0.0 0.0  
 3.0 0.0 0.0  
 3.0 0.0 2.0  
 0.0 0.0 2.0

#### 2.2.2.4.2 Example of node selection by polyhedron

*'extent of zone: boundary cells in polyhedron'*  
*'read data from file'*  
*'cube.off'*

!  
 !cube.off is a file with following data

!  
*OFF*  
 # cube.off  
 # A cube  
 8 6 12  
 1.0 0.0 1.0  
 0.0 1.0 1.0  
 -1.0 0.0 1.0  
 0.0 -1.0 1.0  
 1.0 0.0 0.0  
 0.0 1.0 0.0  
 -1.0 0.0 0.0  
 0.0 -1.0 0.0  
 4 0 1 2 3  
 4 7 4 0 3  
 4 4 5 1 0  
 4 5 6 2 1  
 4 3 2 6 7  
 4 6 5 4 7

#### 2.2.2.4.3 Example of cell selection by cell ids

*'extent of zone: all cells by id'*  
*'read data'*

4 ;number of cells

```

2                ;ids
8
10
15
!same as
'extent of zone: all cells by id'
'read data from file'
'idlist.txt'

```

!idlist.txt is a file with following data

```

4                ;number of cells
2                ;ids
8
10
15

```

### 2.2.3 Constrains for general node selection

MIN3P-THCm-USG provides node selection 'extent of zone' with constrains such as boundary nodes only, internal nodes only and all nodes. This command should be used in the same format as 'extent of zone' in structured grid.

#### 2.2.3.1 Keywords

*'extent of zone: boundary nodes only'*

Only boundary nodes that locate in the given zone are selected.

*'extent of zone: internal nodes only'*

Only internal nodes that locate in the given zone are selected.

*'extent of zone: all nodes'*

All nodes that locate in the given zone are selected, same as 'extent of zone'.

#### 2.2.3.2 Requirements

Optional.

#### 2.2.3.3 Notes

#### 2.2.3.4 Examples

*'extent of zone: boundary nodes only'*

0.0 1.0 0.0 2.0 4.0 5.0

### 2.2.4 Cell selection constrains for general cell selection

Similar as node selection constrains, this command selected cells with given constrains.



**2.2.4.1 Keywords**

*'extent of zone: boundary cells only'*

Only boundary cells that locate in the given zone are selected

*'extent of zone: internal cells only'*

Only internal cells that locate in the given zone are selected. Cell center is used to define if the cell is in or out of the given zone.

*'extent of zone: all cells'*

All cells in the given zone are selected.

**2.2.4.2 Requirements**

Optional.

**2.2.4.3 Notes****2.2.4.4 Examples**

*'extent of zone: boundary cells only'*

0.0 1.0 0.0 2.0 4.0 5.0

**2.3 FLUX APPROXIMATION RELATED COMMANDS**

For the structured mesh, two-point flux approximation and upstream weighting is a standard method. For the unstructured mesh, two-point flux approximation cannot generate correct results in the simulation with anisotropic material properties. The stand two-point upstream weighting may also violate the monotonicity and cause difficulty in convergence.

**2.3.1 Multi-point flux approximation****2.3.1.1 Keywords**

*'multi-point flux approximation'*

*'multi-point flux approximation for flow'*

*'multi-point flux approximation for heat transport'*

*'multi-point flux approximation for reactive transport'*

**2.3.1.2 Requirements**

Optional. By default, multi-point flux approximation is used for unstructured mesh.

**2.3.1.3 Notes**

'multi-point flux approximation' keyword applies to flow, heat transport and reactive transport.

**2.3.1.4 Examples**

Add the following keywords into data block 'spatial discretization'

*'multi-point flux approximation'*

*'multi-point flux approximation for flow'*

*'multi-point flux approximation for heat transport'*

*'multi-point flux approximation for reactive transport'*

**2.3.1 Two-point flux approximation****2.3.1.1 Keywords**

*'two-point flux approximation'*

*'two -point flux approximation for flow'*

*'two -point flux approximation for heat transport'*

*'two -point flux approximation for reactive transport'*

**2.3.1.2 Requirements**

Required if two-point flux approximation is to be used. By default, multi-point flux approximation is used for unstructured mesh.

**2.3.1.3 Notes**

'two-point flux approximation' keyword applies to flow, heat transport and reactive transport.

**2.3.1.4 Examples**

Add the following keywords into data block 'spatial discretization'

*'two-point flux approximation'*

*'two -point flux approximation for flow'*

*'two -point flux approximation for heat transport'*

*'two -point flux approximation for reactive transport'*

### 2.3.2 Multi-point upstream weighting

Multi-point upstream weighting is a default option for unstructured mesh when 'upstream weighting' is used.

#### 2.3.2.1 Keywords

No keyword is required.

#### 2.3.2.2 Requirements

None.

#### 2.3.2.3 Notes

#### 2.3.2.4 Examples

### 2.3.3 Two-point upstream weighting

To use two-point upstream weighting, user needs to specify the following keyword.

#### 2.3.3.1 Keywords

*'two-point upstream weighting'*

#### 2.3.3.2 Requirements

Required if standard two-point upstream weighting is to be used.

#### 2.3.3.3 Notes

#### 2.3.3.4 Examples

Add the following keyword into data block 'control parameters - variably saturated flow'

*'two-point upstream weighting'*

## 2.4 INITIAL CONDITION RELATED COMMANDS

In MIN3P-THCm structured grid version, initial condition can be read from external file in tecplot format. In unstructured grid version, the external file can be in tecplot format or HDF5 binary format. The number of processors used to read external file can be different from the number of processors used to generate those external files.

### 2.4.1 Initial condition from external file

#### 2.4.1.1 Keywords

*'read initial condition from file'*

!or

*'read initial condition from vtk file'*

!or

*'read initial condition from hdf5 file'*

#### 2.4.1.2 Requirements

Optional if initial condition is set through zone selection.

#### 2.4.1.3 Notes

The file name should be prefix.ivs for standard format or prefix.ivs.vtk for vtk file format or prefix.ivs.h5 for hdf5 file format. A corresponding domain mesh file in hdf5 format (prefix.domain.h5) should be provided together.

#### 2.4.1.4 Examples

*'read initial condition from hdf5 file'*

## 2.5 BOUNDARY CONDITION RELATED COMMANDS

In MIN3P-THCm structured grid version, when applying Newmann boundary condition (second type, e.g., flow flux, heat flux, atmospheric boundary evaporation flux), the flux term is assumed to be perpendicular to the interface. In MIN3P-THCm-USG, if not specified with flux direction, Newmann boundary type is also assumed to be perpendicular to the interface. For complex geometry boundary, user need to define the flux direction if flux is not perpendicular to the geometry boundary.

### 2.5.1 Flux direction across domain boundary

#### 2.5.1.1 Keywords

*'boundary flow direction'*

!or

*'boundary flux direction'*

!followed by direction vector

$x, y, z$  ;vector  $x\mathbf{i}+y\mathbf{j}+z\mathbf{k}$

#### 2.5.1.2 Requirements

Optional if flux is perpendicular to the geometry boundary.

### 2.5.1.3 Notes

If geometry boundary outer unit normal vector is  $\mathbf{na}$ , flux direction is vector  $\mathbf{nb}$ , Newmann boundary value is  $f$ , the valid internal boundary value is  $f \frac{|\mathbf{na} \cdot \mathbf{nb}|}{|\mathbf{na}| * |\mathbf{nb}|}$ , if both  $\mathbf{na}$  and  $\mathbf{nb}$  are unit vectors, this form can be reduced to  $f |\mathbf{na} \cdot \mathbf{nb}|$ .

### 2.5.1.4 Examples

*'boundary flux direction'*

1.0 0.0 0.0 ;flux is in x-direction

## 3 OUTPUT FOR MIN3P-THCM-USG

### 3.1 OUTPUT FILES

Unlike MIN3P-THCm structured grid version that uses tecplot data format in both spatial output and transient output, MIN3P-THCm-USG uses vtk file format for spatial output and tecplot data format for transient output. Transient output in MIN3P-THCm-USG uses tecplot data format as vtk is not designed for this kind of data file. Format of vtk file is available at <https://www.vtk.org/wp-content/uploads/2015/04/file-formats.pdf>.

For transient output, the output file name remains the same as MIN3P-THCm structured grid version. For spatial output, the output file name adds '.vtk' as the extension file name. For example, if the output file name in MIN3P-THCm structured grid version is output\_1.gsp, it is changed to output\_1.gsp.vtk in MIN3P-THCm-USG. User can use this file to post-process the results in Paraview or other softwares (e.g., VisIt) that can support vtk format. For ParaView, please visit <http://www.paraview.org/> for more information.

For the MPI and hybrid MPI-OpenMP parallel version, the spatial output of unstructured grid use HDF5+XDMF format which can support high efficient parallel IO. The restart file is also in HDF5+XDMF format when run the code in parallel. If the output is output\_1.gsp.vtk in the sequential version and OpenMP version, it is changed to output\_1.gsp.vtk.h5 and output\_1.gsp.vtk.xmf. User can view this file in ParaView by loading xmf file.

The mesh file can be stored separately to save space as the mesh data and structure in all the spatial output is the same. To use this feature, add *'use separated mesh and result data'* in the *'output control'* block. In this case, user needs to put the result data together with mesh data prefix\_domain.h5 together to do post-processing.

### 3.2 VELOCITY OUTPUT

Velocity output in MIN3P-THCm-USG is different with MIN3P-THCm. For the structured grid version, the velocity at the control volume interface is chosen while for the unstructured grid version, the velocity at the node (control volume center) is used.



## REFERENCES

- Mayer, K.U., 1999. A numerical model for multicomponent reactive transport in variably saturated porous media. PhD Thesis. University of Waterloo.
- Mayer, K.U., Frind, E.O., Blowes, D.W., 2002. Multicomponent reactive transport modeling in variably saturated porous media using a generalized formulation for kinetically controlled reactions. *Water Resour. Res.* 38, 1174, doi: 10.1029/2001WR000862.