Voro++ GiD plug-in
Voronoy geometry plug-in

Voro ++ GiD plug-in tutorial

Table of Contents

1 Voro ++ GiD plug-in tutorial

Introduction

Voro is a Tcl/Tk GiD plug-in that wraps the voro++ open source program to create Voronoi tessellations (see math.lbl.gov/voro++)

The plug-in run in background the command line program and create a GiD geometric model, allowing then all kind of numerical simulations.

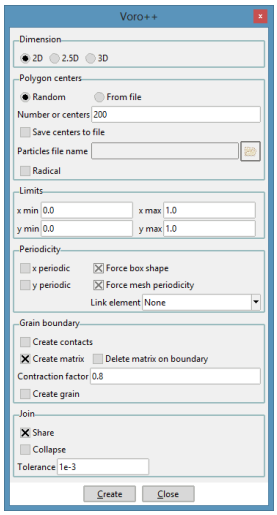
Installation:

Simply copy the plug-in obtained from http://www.gidhome.com/gid-plus/modules/plugins inside the /plugins GiD folder.

The plug-in includes voro compiled for Windows and Linux x32 (the x32 version is able to be run in x64 also)

Note: For Linux remember to set the execution flag to voro++ with the command chmod +x voro++

Then start GiD and a new menu 'Geometry->Create->Voronoi...' will appear to open this window:



Voro++ plug-in window

This tutorial will explain the meaning of the parameters.

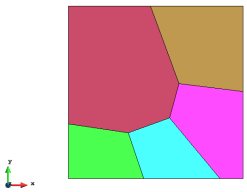
Note: most fields have a contextual help to describe them.

Pressing the *'Create'* button a GiD geometry will be created with the current window parameters.

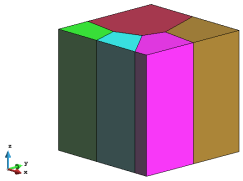
Dimension:

voro++ only generates 3D Voronoi polygons, but the plug-in expands its features to allow the 2D case and to the 2.5D case.

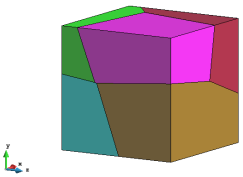
2.5D mean that is an intermediate case between 2D and 3D. It is like the 2D case but the geometry is an extrusrsion along axis z and then become 3D volumes:



2D



2.5D



3D

Polygon centers:

Random:

The plug-in generates randomly the specified amount of polygon centers inside a box. Each center will generate a Voronoy polygon (a volume in 3D or a surface in 2D),

For example the previous case are created with 4 random centers in 2D, 2.5D and 3D respectively.

From file:

It is also possible to write externally the centers in a simple text file, that contain a line by center with the center id and its x y z coordinates (z must be 0.0 for the 2D and 2.5D cases)

The use of the file is interesting in order to be able to repeat exactly the same geometry without any random factor.

For example:

```
1 0.832353753891007 0.36954164615345264 0.0
2 0.5130664270897706 0.1074400977731869 0.0
3 0.371065317360249 0.494788873705449 0.0
4 0.30237625599949447 0.03773458350344309 0.0
5 0.8710463786828547 0.6764865227399797 0.0
```

And then select the filename in the '*Particles file name*' field (picking the folder icon a dialog window will be opened to facilitate the file selection)

Number of centers:

Is the amount of centers to be generated randomly, in case of read 'From file' this field is not used (the windows disable the entry to avoid its edition)

Save centers to file:

This allow to generate the once the centers randomly and automatically save them to a file to be used next times (the output filename is the same field used for read)

Radical:

Enabling this option with each center is generated also randomly a weight (a positive real number) that do that the Voronoy polygon become closer to centers with higher weights.

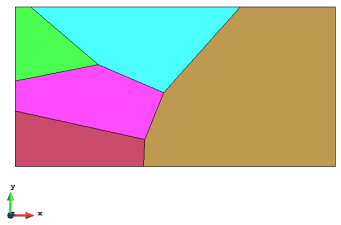
If the information is read from file, and extra number for the weight must be provided after the z coordinate of each center.

Limits:

The shape of the domain is a box with dimensions 1x1x1 units by default, it is possible to set the interval on each direction x y z.

In case of select 2D the z field will be hidden.

For example if *xmin* and *xmax* are changed to -1.0 and 1.0 respectively the shape becomes a rectangle of 2x1 instead of a square:



Rectangular domain

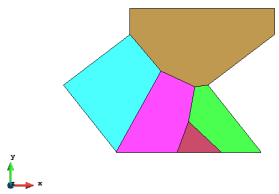
Periodicity:

x,y,z-periodic:

Some numerical simulations of materials require that the geometry have periodicity along the axis, to simulate a big domain with a small representative region that could be virtually repeated to fill the whole domain.

e.g. In 2D a geometry periodic in x direction mean than the lines of x-min boundary must match exactly the lines of the x-max boundary.

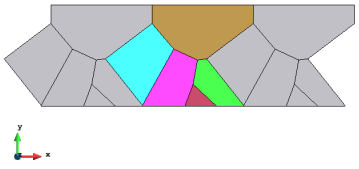
If we set 2D, with 5 centers, default limits and x-periodicity something like this could be obtained:



X-periodic non-rectangular shape

This shape show periodicity in x, but not in y. the area of all surface is exactly equal to 1.0x1.0.

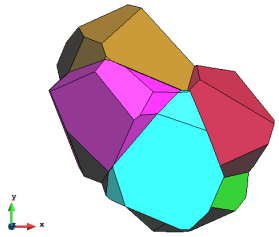
If we copy the surfaces with a translation of (1.0,0.0) and (-1.0,0.0) its x-periodicity is highlighted.



Two copies to remmark the periodicity

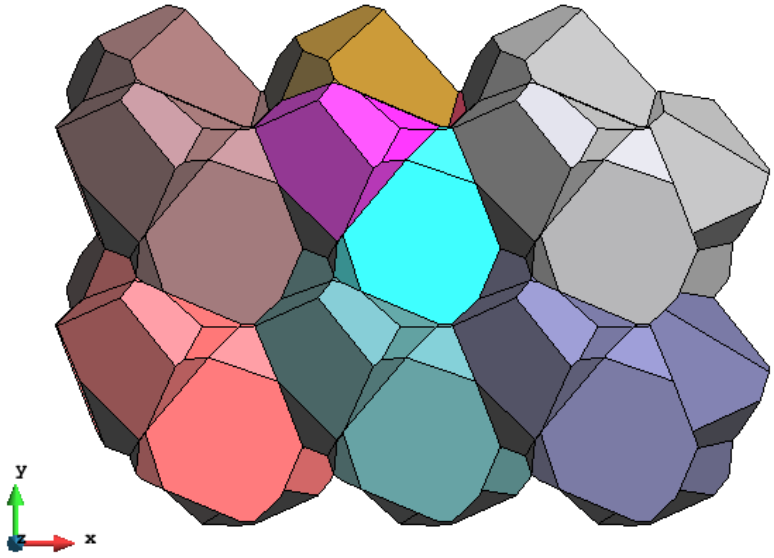
Although the shape has periodicity in x most simulation programs cannot apply easily periodical boundary conditions in the shape is not rectangular.

This is a 3D example with periodicity in x, y and z directions:



3D shape periodic in all directions

Its periodicity is not so easy to be seen visually, but if we copy it with a translation the faces match perfectly in all directions.



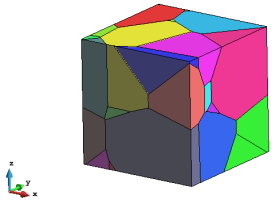
Several copies to remmark the periodicity

Force box shape:

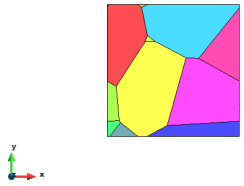
In any case, usually we prefer a periodic geometry with the shape of the bounding box; then the 'Force box shape' option must be set.

In this case the amount of volumes (in 3D) is bigger than the amount of selected centers; this is because several voronoy volumes must be cutted to fit the required box.

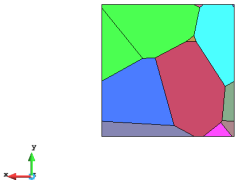
For example with 5 random centers we could obtain 28 volumes like the image.



3D view



Top view



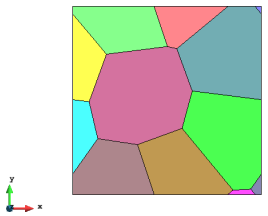
Bottom view

In could be seen that the top and bottom view are symmetric.

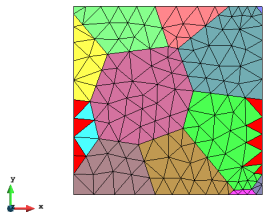
Force mesh periodicity:

Although the periodicity of a geometry when a mesh is generated the periodicity will be in general lost and the nodes and elements won't be periodic.

For example this 2D case is periodic in both directions, but the mesh is not periodic (there are four red-selected triangles on the left-side and five on the right-side)



Periodic geometry



Non-periodical mesh

In order to ensure periodicity in GiD is necessary a trick: to create a special geometric entity, called 'separated contact volume', that links two 'similar' surfaces and force that both meshes are 'similar', with a one-to-one relationship between its nodes.

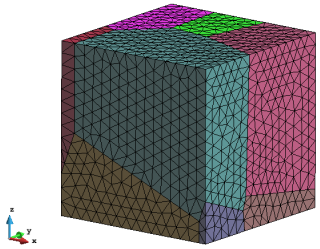
Both surfaces of a contact volume must have the same amount of lines and its shape must be very similar, usually they are equal with a 'rigid-body movement'.

Selecting '*Force mesh periodicity*' the pair of surfaces is automatically matched and the 'separated contact volumes' are created.

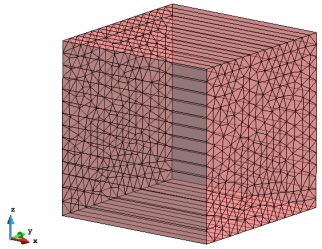
These special entities are created in a layer named 'Contacts_outer'.

The contact volumes will generate elements, usually prisms that link the triangles of its 'master' and 'slave' surfaces.

Next image show a geometry with x-periodicity in geometry and also in mesh. On the right the prisms along x are drawn (with transparency to show both faces)



X-periodical mesh



Extra prisms of the 'separated contact volumes'

Link element:

The contact volumes are necessary to ensure mesh periodicity, but its own elements could be generated, as prisms or as line-elements joining two nodes, or avoided (selecting *'link element'* as *'None'.*

Note:

In the 2D case the periodicity of the geometry could be achieved also, but nowadays the periodicity of the mesh is not possible, because the concept of 'separated contact surface' doesn't exists in current GiD versions. It must be implemented in future releases.

Fortunately 2D geometries are simpler than 3D ones, and periodicity could be forced with other tricks,

like force the same amount of structured mesh divisions on each pair of periodic lines.

Grain boundary:

Create contacts:

Depending on the kind of mechanical numerical simulation it could be required:

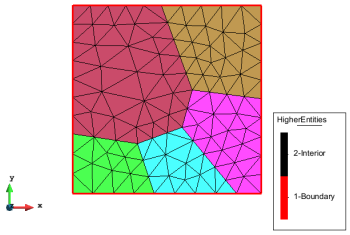
A) The polygons sharing its faces and the mesh nodes must be also shared between neighbor elements of two bodies.

B) Each polygon independent and not necessarily have 'conformal nodes' between neighbor bodies.

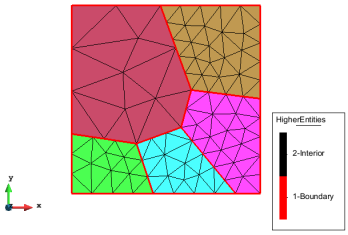
C) Each polygon with its own node, but exactly in the same location as the one of the neighbor body (an maybe extra 'degenerated contact elements' linking pairs of faces)

To fit the case A) must be set the 'Share' option and then points, lines and surfaces are created shared, other ways we are in the case B) without sharing entities between polygons.

Next images illustrate it in a 2D case drawing the amount of parent entities (View->Higherentities->Edges)



Case A - triangles sharing nodes

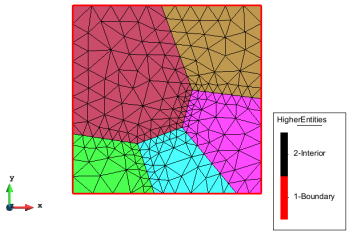


Case B - triangles whitout share nodes

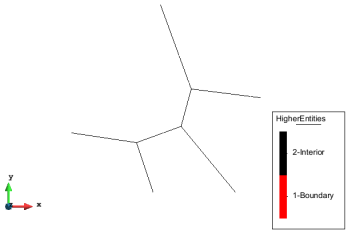
To fit the case C) it is necessary not to share nodes (unset *'Share'*) and force having conformal nodes in both bodies and create the extra 'contact element' between neighbor faces (a degenerated quadrilateral in 2D). To do it in GiD must create a special pseudo-geometrical entity called 'contact surface' between two 'similar curves' on the same space location.

This is done selecting *'Create contacts'*, then automatically the contact surfaces are created

In this case drawing higherentities looks like the case A, but really nodes are duplicated and there are quadrilaterals with zero height in the contact parts. The contact surfaces and its quadrilaterals are created in the layer 'Contacts_inner', the image on the right is like the one on the left but hiding the rest of layers.



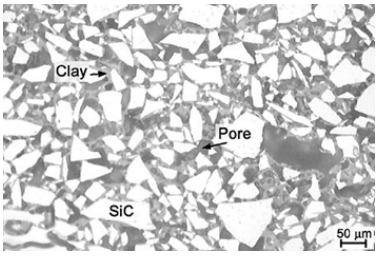
Case C - triangles and quadrilaterals



Case C - degenerated contact quadrilaterals

Create matrix:

Some materials have a microstructure with grains surrounded by a matrix of a different material, like the next image:

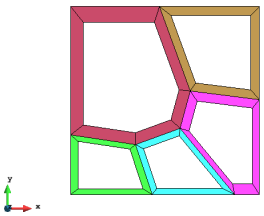


Micro-structure photograph of a ceramic material

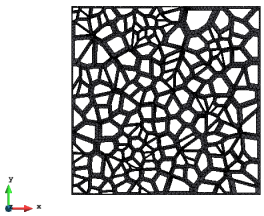
The Voronoi polygons could represent more less the grains of poli-crystals structures, but previous examples create a matrix with zero depth.

The *'Create matrix'* option generates a modified geometry, extruding the surfaces between grains, and then creates a continuous matrix around them.

The depth of the extrusion depends on the *'Contraction factor'* (the surface is contracted by this factor inside the grain)



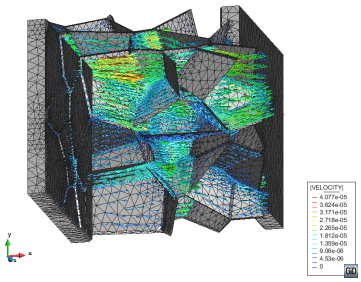
Matrix created by extrussion with factor=0.8



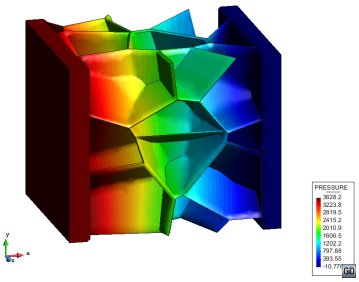
Matrix with 200 random polygons

The same operation could be done in 3D, for example the next images show a simulation done with Kratos (flow in the porous media of the matrix)

The extruded layers touching the boundary have been deleted, selecting the *'Delete matrix on boundary'* option.



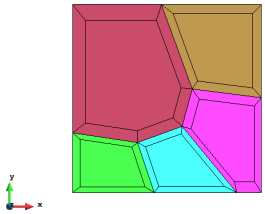
Vectors of the velocity of the fluid



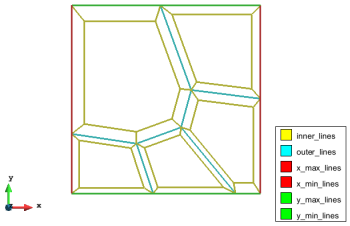
Color map of the pressure

In previous images, only the matrix is created, but the grain is volume is missing. If we want to create also these volume grains (surfaces in 2D) is necessary to select *'Create grain'.*

In order to facilitate the assignation of simulation properties (e.g. input velocity, output pressure,...) some groups are automatically created.



Matrix and grains



Entities of groups drawn by colors.

Join

Share:

As is explained before, the *'Share'* option controls that the geometrical entities are created shared by neighbor polygons or each one with its own copy.

Note: the 'Share' option must be unset if '*Create contacts*' is used.

Collapse:

Selection this setting, after create all geometry, an extra operation is done to find 'duplicated' entities based on a given *'Tolerance'* and delete them sharing a single copy. This could provide a result similar than the *'Share'* option, but could also be used to delete 'short segments' collapsing them to a single point (e.g. to avoid concentration of too small elements).

The tolerance must be used carefully, a big tolerance could degenerate the whole geometry to a point!!

Note: the collapse must not be used when creating contact volumes, because these contacts will be deleted by the collapse, and close parts will be joined.