GadGeo

User Guide

GeoDict release 2023

Published: December 28, 2022

https://doi.org/10.30423/userguide.geodict

Citation:
Jürgen Becker, Dennis Mosbach, Barbara Planas. GeoDict 2023 User Guide. GadGeo handbook. Math2Market GmbH, Germany, doi.org/10.30423/userguide.geodict

The handbooks in the User Guide series of Math2Market GmbH can be obtained from:

Math2Market GmbH
Richard-Wagner-Strasse 1
67655 Kaiserslautern
Germany

Phone: +49 631 205 605 0
Fax: +49 631 205 605 99
Email: info@math2market.de
Web: www.math2market.de

Introduction to GAD Objects

GeoDict Analytic Data

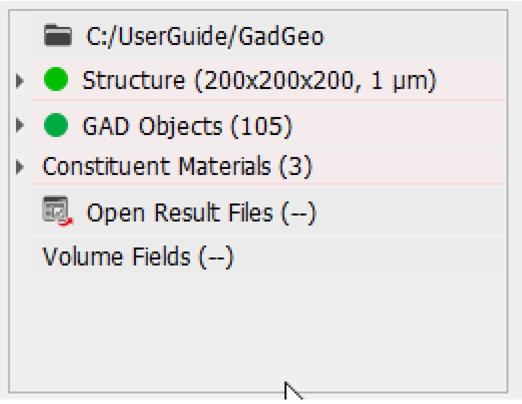
Structure models in GeoDict can be described in two fundamentally different ways.

On first glance, GeoDict uses a voxel grid to describe any 3D structure, and the voxel grid is used by its modules to compute the properties. When you import a 3D image into GeoDict, or when you create a structure with any of the Geo modules, a voxel grid is produced that represents the structure. While this description is very general, it lacks any further knowledge about how the material is composed of geometrical objects. Is this a fiber? What is the direction of it, the orientation of an object? The voxel grid cannot tell the answer to these questions.

However, when a structure model is virtually created with one of the GeoDict modules, at the time of creation all the geometrical information is known. To allow the user to make use of these information later on, it is not discarded, but stored additionally to the voxel grid whenever available.

These geometric objects like spheres, ellipsoids, and fibers in GeoDict are called **GAD**-objects, where **GAD** stands for GeoDict analytic data.

In the project and status section, on the left of the GeoDict GUI, a green dot indicates that this data is available for the current 3D structure.



The properties of **GAD** Objects, like their diameter or position, can be written to an ASCII-file, which has the file ending **.gad**. A **.gad** file, which is easy to read and to modify, contains all the information of a given geometry.

A **.gad**-file, written by GeoDict or an external tool, can be imported into GeoDict. During the import process, the analytic information is transformed into a voxel geometry (*voxelized*), where certain discretization parameters, like the voxel length, are required.

The GAD File Format

A **.gad** file is an ASCII file which can be opened with a text editor such as NotePad++, or WordPad. Since GeoDict 2023, the syntax is that of a python dictionary. In previous GeoDict versions, a different file syntax was used, and the old syntax can still be read and written by GeoDict 2023.

The general structure of a .gad file looks as follows:

```
# Example .gad File

{
  "Header" : {
    "Release"           : 2023,
    "Revision"          : 58894,
    "BuildDate"         : "23 Sep 2022",
    "CreationDate"      : "26 Sep 2022",
    "CreationTime"      : "14:20:50",
    "Creator"           : "becker",
    "Platform"          : "64 bit Windows"
  },

  "NumberOfObjects"   : 8,

  "Description"       : "GAD Grain Shapes",

  "Domain" : {
  [...]
  },

  "ConstituentMaterials" : {
  [...]
  },

  "MaterialDatabase" : {
  [...]
  },

  "Object1" : {
    "MaterialIDModel"     : "Constant",
    "MaterialID"          : 1,
    "Type"                : "Sphere",
    "Position"            : [4e-05,6.5e-05,4e-05],
    "Diameter"            : 2e-05,
    "Axis1"               : [0,1,0],
    "Axis2"               : [1,0,0]
  },

  [...]
}
```

Every gad-file written by GeoDict starts with a **Header** section containing details about the creator, the file, and the GeoDict version.

The keyword (or short **key**) **NumberOfObjects** specifies the number of objects in the file. The data describing each Object is given in a separate sections **Object1, Object2,** … later in the file.

The given **Description** is shown in the title of the main screen of the GeoDict GUI when the file is loaded.

In the **Domain** section, the size of the domain of the voxel geometry is specified, which allows GeoDict to directly load a .gad file and create a voxel geometry. The information is used as default, but can be changed during the import process when importing the gad-file with ImportGeo-Vol. The content of the Domain section is described in more detail with the Edit Domain command on page 48.

The **ConstituentMaterials** link the material ID with a material. The background material is defined in the **Domain** section, all other materials are defined in this section. All materials have a **Name**, a **Type** and an **Information** section. The name is a reference to materials of the GeoDict Material Database. The type is referring to a material state (solid, fluid, ..). The information could be either empty or a useful description of the material. Besides this, all properties defined in the GeoDict Material Database for this material are stored in this section:

In the **MaterialDatabase** section, the complete material database entries of all materials present in the structure are stored. The entries describing the materials are in the same format as the entries of GeoDict material database. This format is described in more detail in the Material Database handbook of this User Guide.

List of GAD Objects

The heart and most important part of the .gad file is the list of GAD objects.

The definition of a GAD object in a file is given in the **Object%1** section, where the symbol **%1** stands for the index of the object. This means that the keys and values of the first object are given in the **Object1** section, the keys and values of the second object are given in the **Object2** section, and so on. The **NumberOfObjects** given above has to correspond to the number of sections defined here.

Each GAD object has some common keys:

- The **MaterialIDModel** defines of how many different materials an object consists. Typically, it is just one ID and the value is **Constant**. Since GeoDict 2022, some non-constant patterns are also possible (see the description of the **Change Material ID Model** command on page 43 for details).
- The value after the key **MaterialID** specifies the material that the object is made of.
- The string after the key **Type** defines the type of the object, for example **Sphere**. The possible object types and the object-specific parameters are described in more detail with the Create / Add GAD Objects command on page 8.
- For all objects, the key **Position** defines the center of the objects. For objects with symmetric shapes this is also the center of mass. For objects with irregular shapes (Planar Polyhedron and Convex Polyhedron) the center is the center of the star-shaped object or the surrounding object, which may slightly differ from the center of mass.

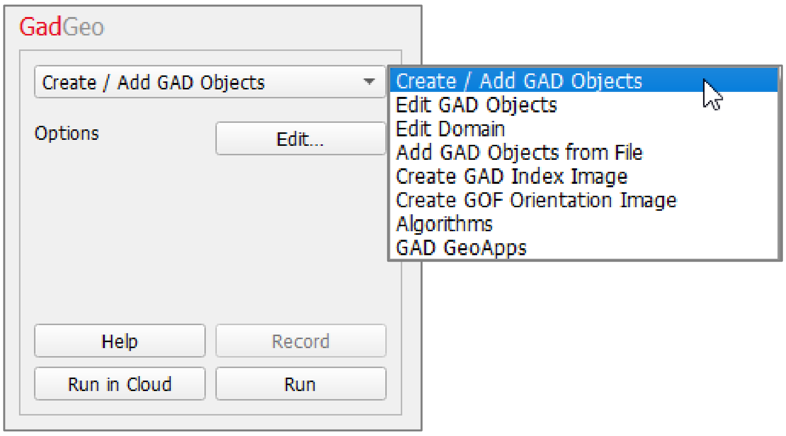
Other parameters depend on the object type and are given as **key-value** pairs where the **value** can be a floating number, an integer number, a Boolean value (true or false), a vector, or a string. Each **key-value** pair has to be put in a separate line, where the order of the lines is not important (the **key-value** pairs of an object can be interchanged). The possible object types and the object-specific parameters are described in more detail with the Create / Add GAD Objects command on page 8.

- If the key **Diameter** is used to specify a parameter of a gad-object, it is always possible to use instead the key **Radius** and half the value of the diameter. This is a convention respected by all gad-objects.

USING GADGEO

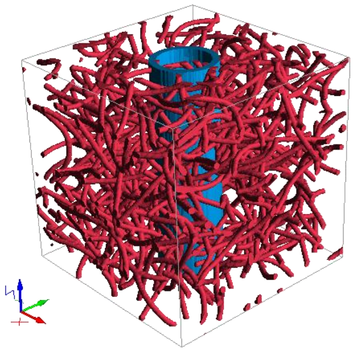
GADGEO SECTION

The heading GadGeo appears in the module section (to the left of the GeoDict GUI) when selecting **Model → GadGeo** in the menu bar to begin working with GadGeo.



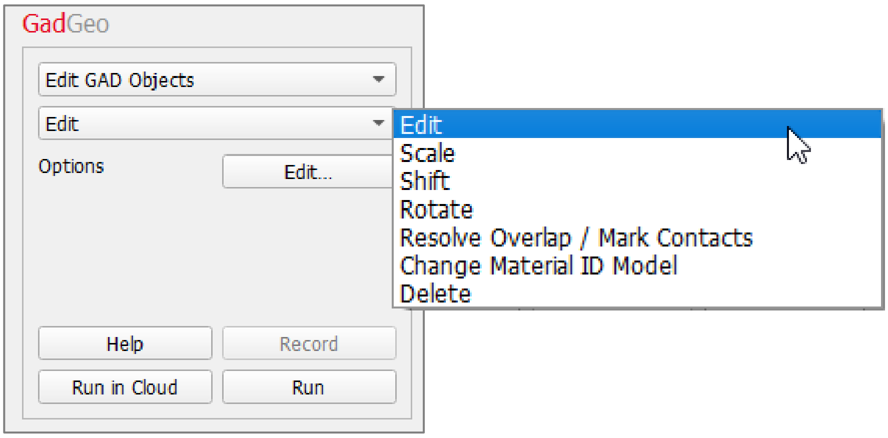
GadGeo can be used to **Create/Add GAD Objects**. Use this command to create new structures as a practical way for building structures consisting of few objects with deterministic properties. New forms of particles can be generated as combined objects.

Furthermore, this command can be used to add objects to existing structures, e.g., to add deterministic objects to stochastic (random) structures.

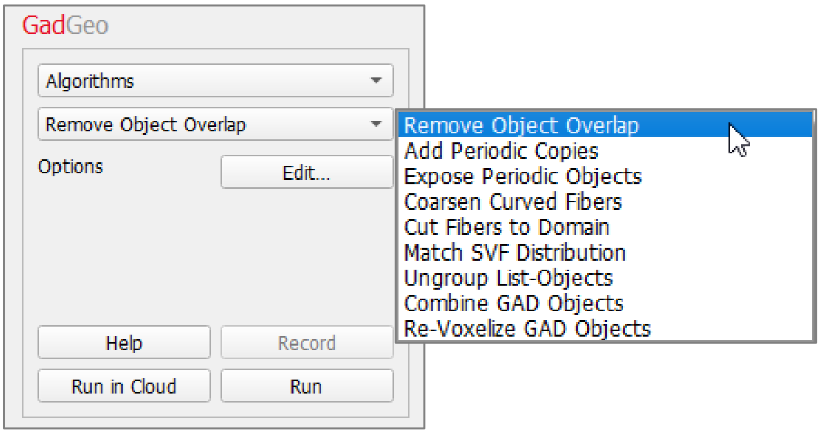


Under **Edit GAD Objects**, it is possible to **Edit**, **Scale**, **Shift**, **Rotate** or **Delete** objects. It is possible to edit material, size, and position of objects.

You can also call the **Resolve Overlap / Mark Contacts** and **Change Material ID Models** commands.



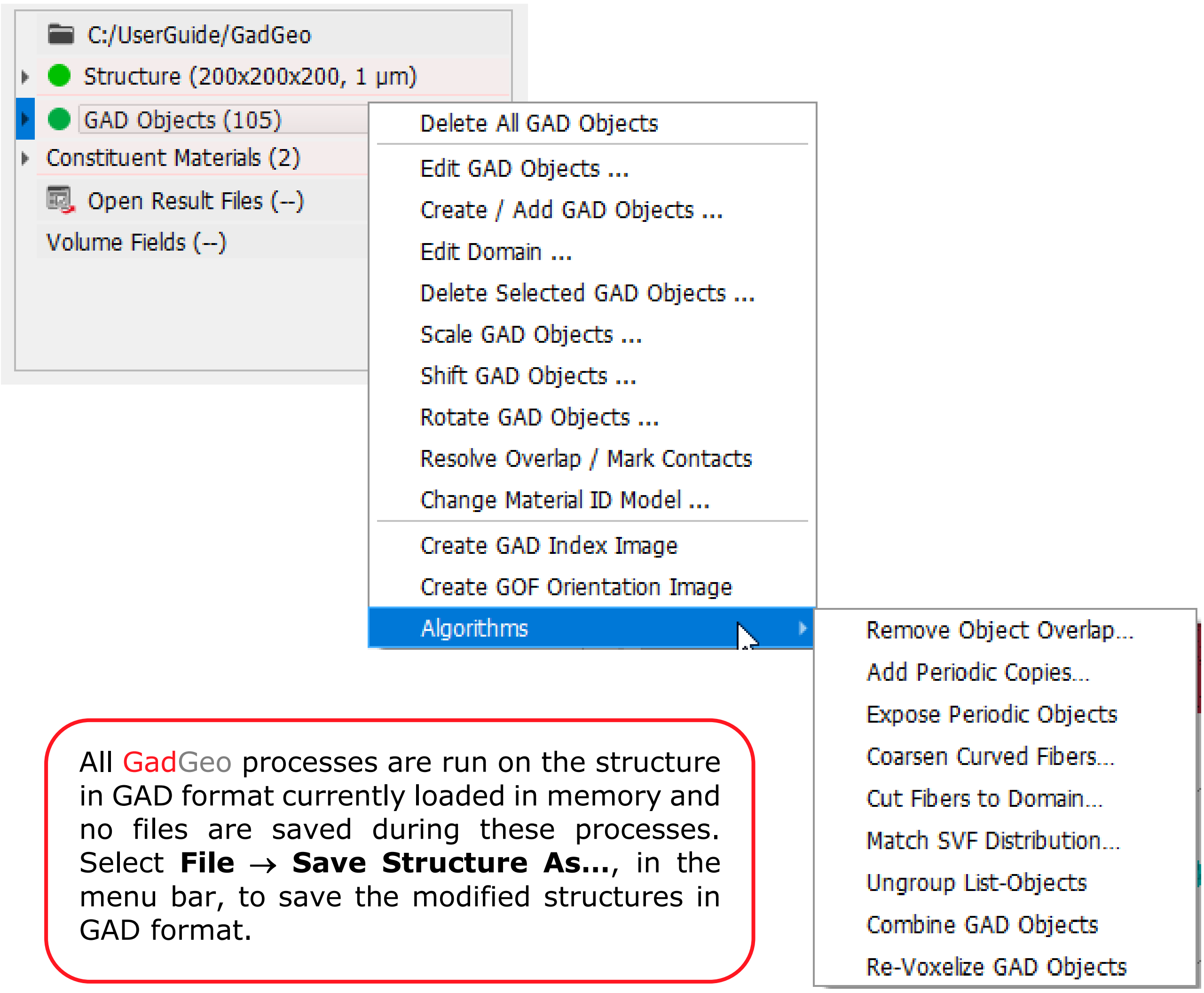
It is also possible to **Edit Domain**, to **Add GAD Objects from File,** to **Create GAD Index Image** or **Create GOF Orientation Image.**



In addition, **Algorithms** are available to **Remove Object Overlap**, to **Add Periodic Copies**, to **Expose Periodic Objects**, to **Coarsen Curved Fibers**, to **Cut Fibers to Domain,** to **Match SVF Distribution**, to **Ungroup List-Objects,** to **Combine GAD Objects** and to **Re-Voxelize GAD Objects** depending on the selection from the pull-down menu.

Additionally, under **GAD GeoApps**, four example scripts show how GadGeo commands can be used to create complex 3D structures.

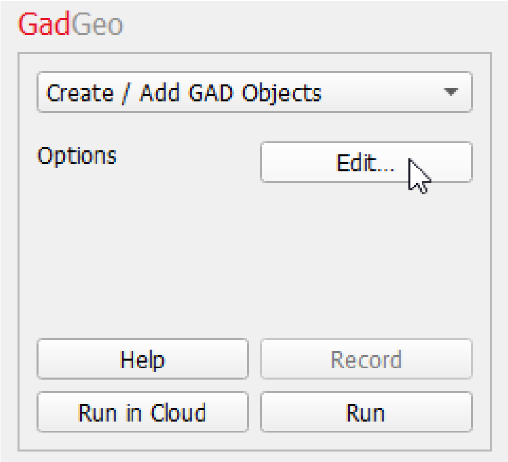
All GadGeo commands are also available through the context menu of the project status panel by right-clicking on the **GAD Objects** line.



All GadGeo processes are run on the structure in GAD format currently loaded in memory and no files are saved during these processes. Select **File → Save Structure As...**, in the menu bar, to save the modified structures in GAD format.

CREATE/ADD GAD-OBJECTS

When selecting **Create / Add GAD Objects** from the pull-down menu, basic objects can be added to an existing structure or used to create a new one, by defining object type, geometric parameters, and material.

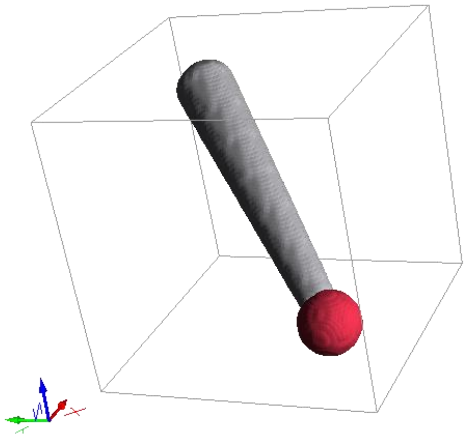


Alternatively, the command is also available from the **GAD Objects** context menu of the **Project Status** panel.

Clicking the **GadGeo Options**' **Edit...** button or choosing **Create / Add GAD Objects...** in the context menu opens the **Create/Add GAD Options** dialog box.

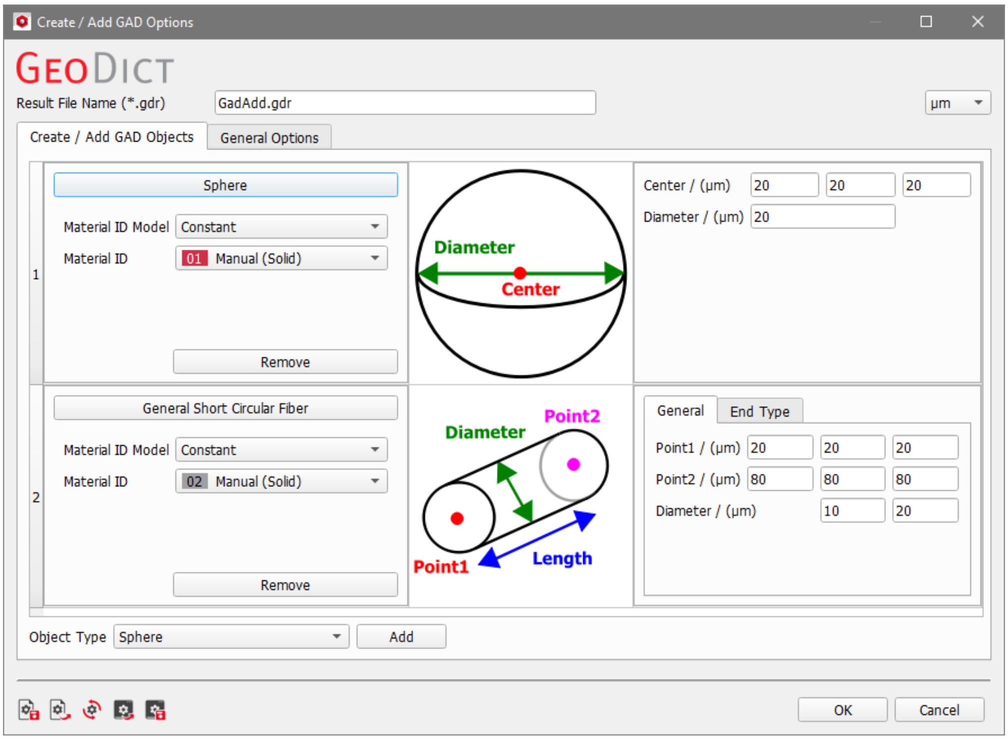
The dialog box contains the **Create / Add GAD Objects** tab, to select objects and their parameters, and the **General Options** tab, to set the size and domain parameters of the resulting structure.

Back in the GadGeo section, clicking **Run** with the default values produces an object that is the combination of a sphere and a general short circular fiber, resembling a matchstick.



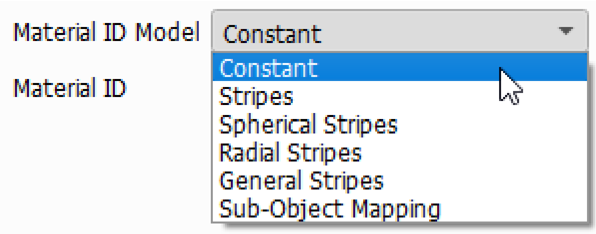
CREATE / ADD GAD OBJECTS

Object types and their geometric options are set through the **Create / Add GAD Objects** tab.

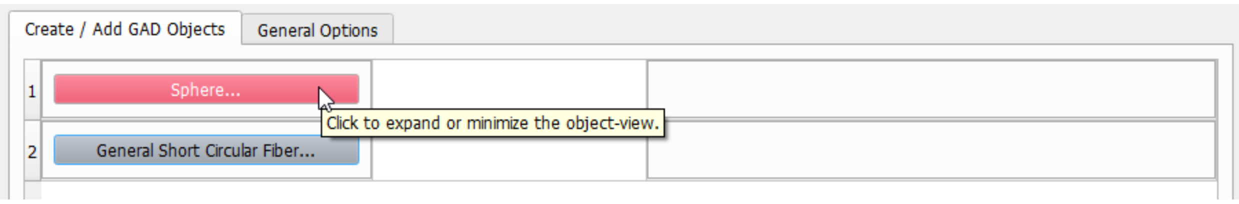


The left panel displays the number assigned to the object (**1**, **2**, **3**...), the Object Type (**Sphere**, **General Short Circular Fiber**, ...), and the object **Material ID** and **Material ID Model**. It is not possible to set the material itself here, but you may assign materials to a given ID using **Settings->Select Constituent Materials** in the main menu of GeoDict prior to using GadGeo.

By default, the **Material ID Model** is set to **Constant**, which means that the whole object is made of a single material. In the other models, the object consists of several constituent materials, (e.g. a bicomponent fiber). The constituent materials can be arranged in several ways, the detail are explained in the description of the **Change Material ID Model** command on page 43.



Clicking the button with the object type name once minimizes the complete panel for the object. Clicking it when the panel is minimized expands the complete panel.



Remove can be used to eliminate a certain object from the list and from the generation.

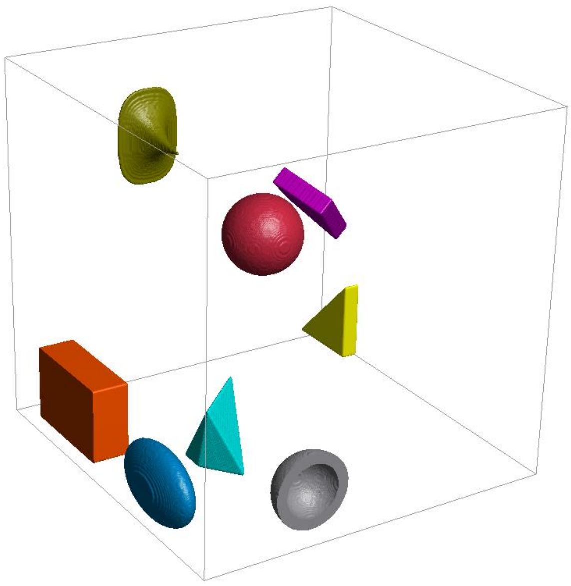
The center panel schematically shows the geometrical properties of basic objects, and the right panel determines the size and position of each basic object through values for **Center**, **Diameter**, **Length**, **Point**, etc.

At the bottom left of the dialog box, the **Object Type** pull-down menu and the **Create** button can be used to add the selected object type to the list of the structure's basic objects during the generation.

In the following, all available object types and their representation in the .gad file are explained.

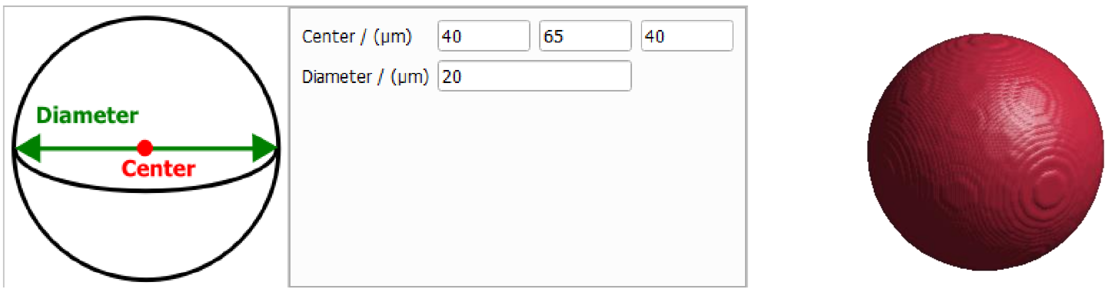
GRAIN SHAPES

Several geometric shapes are available to create objects consisting of grains: spheres, hollow spheres, ellipsoids, and their generalized form of superquadric particles, rectangular boxes, triangular prisms, and planar and convex polyhedrons.



SPHERE

A sphere is defined by the position of its center and a diameter, given by the vector after the key **Position** and the value after the key **Diameter** (**Radius**).

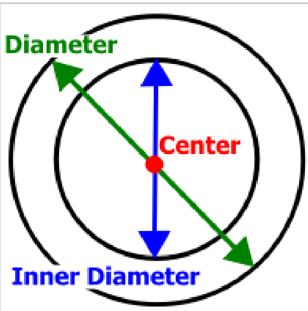


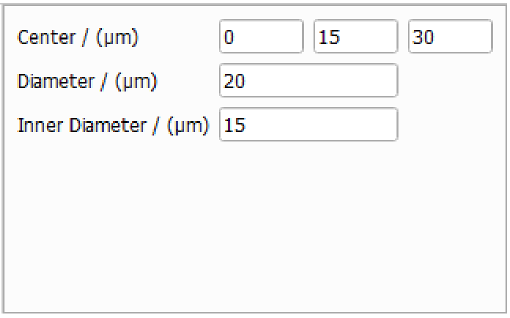
```
"Object1" : {
  "MaterialIDModel"     : "Constant",
  "MaterialID"          : 1,
  "Type"                : "Sphere",
  "Position"            : [4e-05,6.5e-05,4e-05],
  "Diameter"            : 2e-05,
  "Axis1"               : [0,1,0],
  "Axis2"               : [1,0,0]
},
```

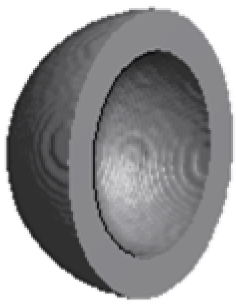
The additional parameters **Axis1** and **Axis2** in the .gad file describe the orientation of the object. The third coordinate axis is not given as an additional parameter, but given as the cross product of Axis1 and Axis2, such that the three axes form a right hand system. The orientation of a sphere obviously does not change the visible 3D geometry, rather, it is used to model anisotropic material behavior. Those parameters cannot be modified through the user interface.

Hollow Sphere

A hollow sphere is similar to a sphere, but has an inner diameter defining the void space inside the sphere, given by the value after the key **InnerDiameter** (or **InnerRadius**).





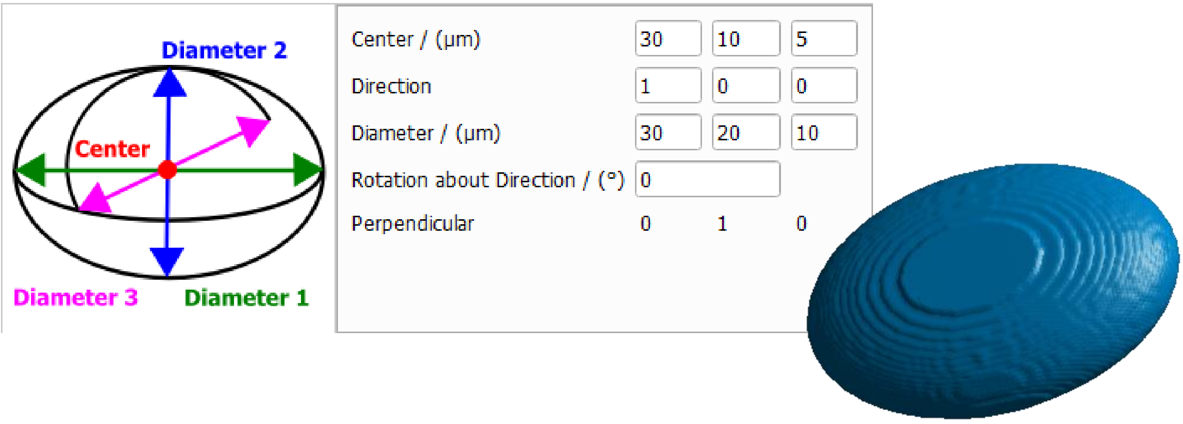


As for spheres, the additional parameters **Axis1** and **Axis2** in the .gad file describe the orientation of the object and cannot be modified through the user interface.

```
"Object2" : {
  "MaterialIDModel"     : "Constant",
  "MaterialID"          : 2,
  "Type"                : "HollowSphere",
  "Position"            : [0,1.5e-05,3e-05],
  "Diameter"            : 2e-05,
  "InnerDiameter"       : 1.5e-05,
  "Axis1"               : [0,1,0],
  "Axis2"               : [1,0,0]
},
```

Ellipsoid

An ellipsoid is defined by its **Center** position, given by the vector after the key **Position**, and three diameters, given by the values after the keys **Diameter1**, **Diameter2**, and **Diameter3**, in the direction of its main axes.




```
"Object3" : {
  "MaterialIDModel"      : "Constant",
  "MaterialID"           : 3,
  "Type"                 : "Ellipsoid",
  "Position"             : [3e-05,1e-05,5e-06],
  "Diameter1"            : 3e-05,
  "Diameter2"            : 2e-05,
  "Diameter3"            : 1e-05,
  "Axis1"                : [1,0,0],
  "Axis2"                : [0,1,0]
},
```

Two of these main axes are given in the gad-file by the vectors after the keys **Axis1** and **Axis2**. **Axis1** corresponds to the given **Direction** in the GUI, and **Axis2** to the **Perpendicular** calculated from the entered rotation angle. Again, the third main axis is determined through the cross product of **Axis1** and **Axis2**, such that a right hand coordinate system is formed.

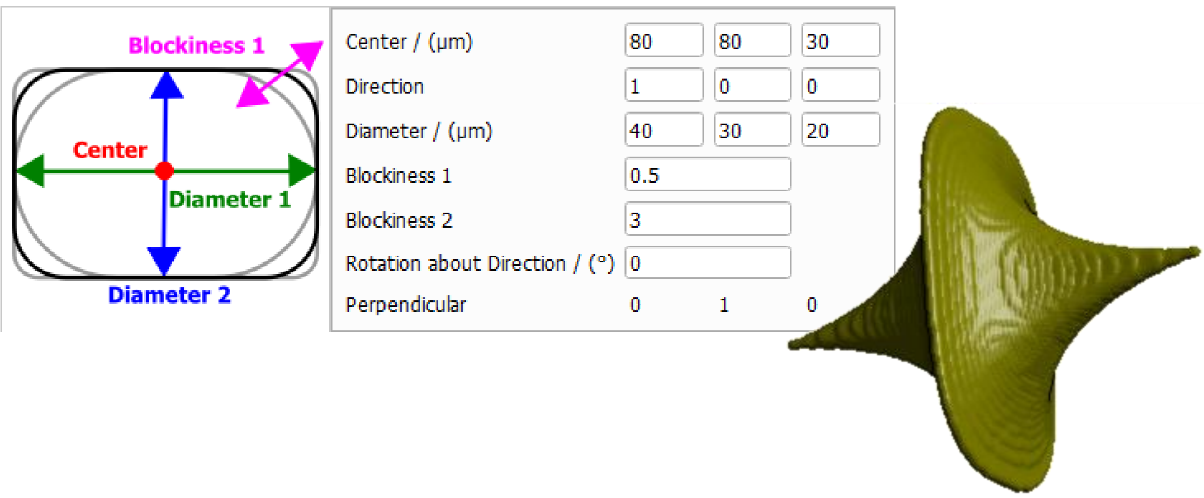
Superquadric Particle

The superquadric particle is implicitly defined by the formula:

$$\left(\left|\frac{x}{d_1}\right|^{n_2} + \left|\frac{y}{d_2}\right|^{n_2}\right)^{\frac{n_1}{n_2}} + \left|\frac{z}{d_3}\right|^{n_1} = 1 \tag{1}$$

For exponents $n_1 = n_2 = 1$ it takes the shape of an octahedron, for exponents $n_1 = n_2 = 2$ it takes the shape of an ellipsoid with diameters d_1, d_2, d_3, for higher exponents the shape approaches a rectangular box.

The exponent n_2 describes the shape in the xy-plane ($z = 0$), the exponent n_1 describes the shape in the xz and yz plane.



The three **Diameter values** correspond to the diameters d_1, d_2, d_3 in formula (1), **Blockiness 1** to the exponent n_1 and **Blockiness 2** to the exponent n_2.

The **Position** of the particle is given through its **Center / Position**, and the orientation of the object. **Axis1** corresponds to the given **Direction** in the GUI, and **Axis2** to the **Perpendicular** calculated from the entered rotation angle.

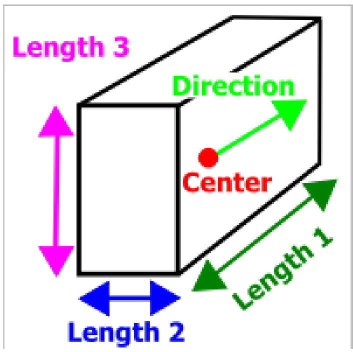
```
"Object8" : {
  "MaterialIDModel"          : "Constant",
  "MaterialID"               : 8,
  "Type"                     : "Superquadric",
  "Position"                 : [8e-05,8e-05,3e-05],
  "Diameter1"                : 4e-05,
  "Diameter2"                : 3e-05,
  "Diameter3"                : 2e-05,
  "Axis1"                    : [1,0,0],
  "Axis2"                    : [0,1,0],
  "BlockinessParameter1"     : 0.5,
  "BlockinessParameter2"     : 3
},
```

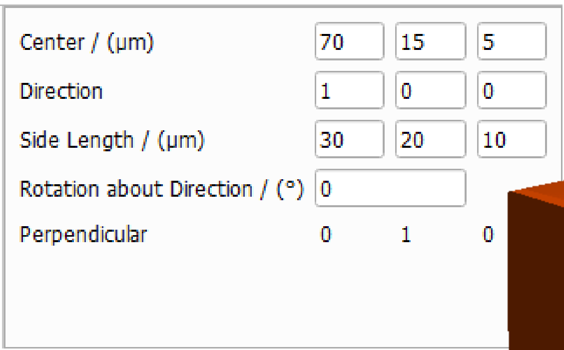
Box

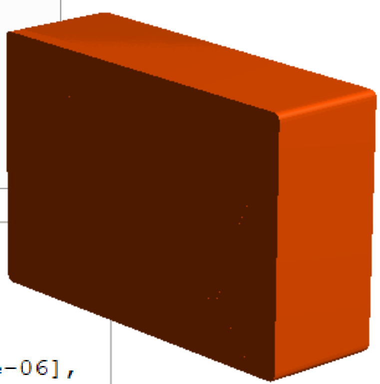
The definition of a box is similar to the definition of an ellipsoid.

The **Center** position is given by the vector after the key **Position**, and the two main axes are given by the vectors after the keys **Axis1** and **Axis2**. **Axis1** corresponds to the given **Direction** in the GUI, and **Axis2** to the **Perpendicular** calculated from the entered rotation angle. As always in GeoDict, the third main axis is determined through the cross product of **Axis1** and **Axis2**, such that a right hand coordinate system is formed.

The three **Side Lengths** of the box in the direction of the main axes are given by the values after the keys **Length1**, **Length2**, and **Length3**.





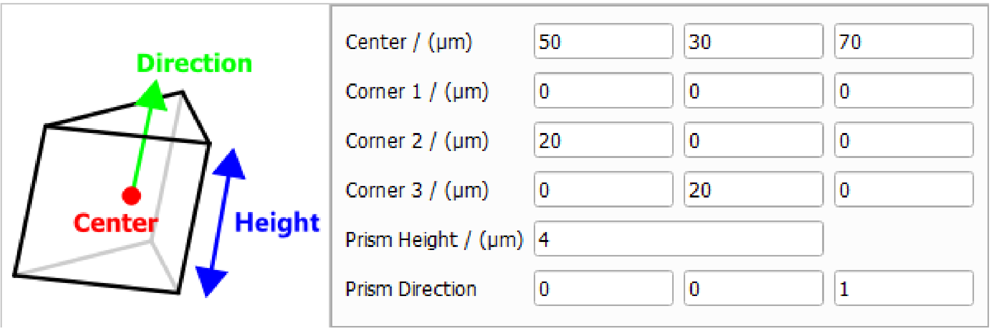


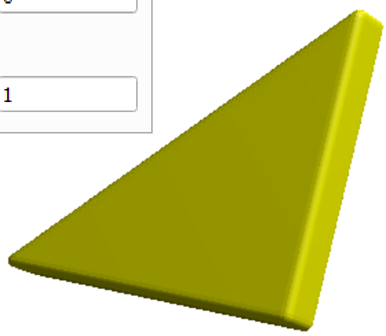
```
"Object4" : {
  "MaterialIDModel"     : "Constant",
  "MaterialID"          : 4,
  "Type"                : "Box",
  "Position"            : [7e-05,1.5e-05,5e-06],
  "Length1"             : 3e-05,
  "Length2"             : 2e-05,
  "Length3"             : 1e-05,
  "Axis1"               : [1,0,0],
  "Axis2"               : [0,1,0],
  "CornerMode"          : "Sharp",
  "CornerRadius"        : 0
},
```

The additional parameters **CornerRadius** and **CornerMode** are used in conjunction for rounded boxes. **CornerRadius** describes the radius of the corner. **CornerMode Rounded** is used for rounded boxes, instead of the **CornerMode Sharp** that is used for normal boxes. Those parameters cannot be modified through the user interface.

TRIANGLE

The Triangle object defines a triangular prism, which might be oblique or right-angled. The base triangle is given through **Corner 1** to **Corner 3** (**Point1** to **Point3** in the .gad file). The coordinates of the corner points are given relative to each other.





```
"Object7" : {
   "MaterialIDModel"       : "Constant",
   "MaterialID"            : 7,
   "Type"                  : "Triangle",
   "Position"              : [5e-05,3e-05,7e-05],
   "Point1"                : [0,0,0],
   "Point2"                : [2e-05,0,0],
   "Point3"                : [0,2e-05,0],
   "Thickness"             : 4e-06,
   "Normal"                : [0,0,1]
},
```

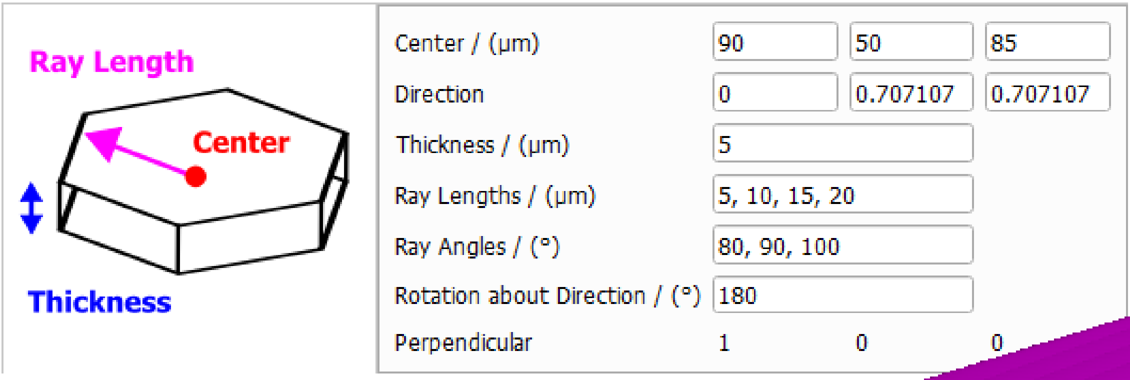
The **Center** (corresponds to the **Position** vector in the .gad file) describes the center of mass of the prism, the **Prism Height** (**Thickness**) the distance between the two parallel triangles. The **Prism Direction** (**Normal** in the .gad file) is the shift direction between the two parallel triangles. If the **Prism Direction** is perpendicular to the plane defined through the three corner points, the prism is right-angled, otherwise it is oblique.

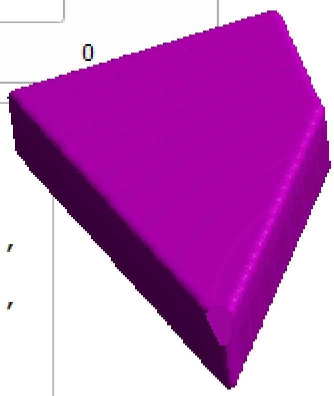
PLANAR POLYHEDRON

A planar polyhedron is defined by a number of points in a plane, forming a closed convex polyline, and a thickness perpendicular to the plane, given by the value after the key **Thickness**.

The plane of the points is defined by the vector after the key **Axis1 (Direction)**. This vector is perpendicular to the plane. The points in the plane are given as the endpoints of rays, starting at the point given by the vector after the key **Position (Center)**.

The number of rays is given by the value after the key **Rays**. The first ray points in the direction specified by the vector after the key **Axis2** (which corresponds to the **Perpendicular** vector automatically computed in the user interface), and it has the length given by the value after the key **RayLength1**. The other ray lengths are given accordingly.



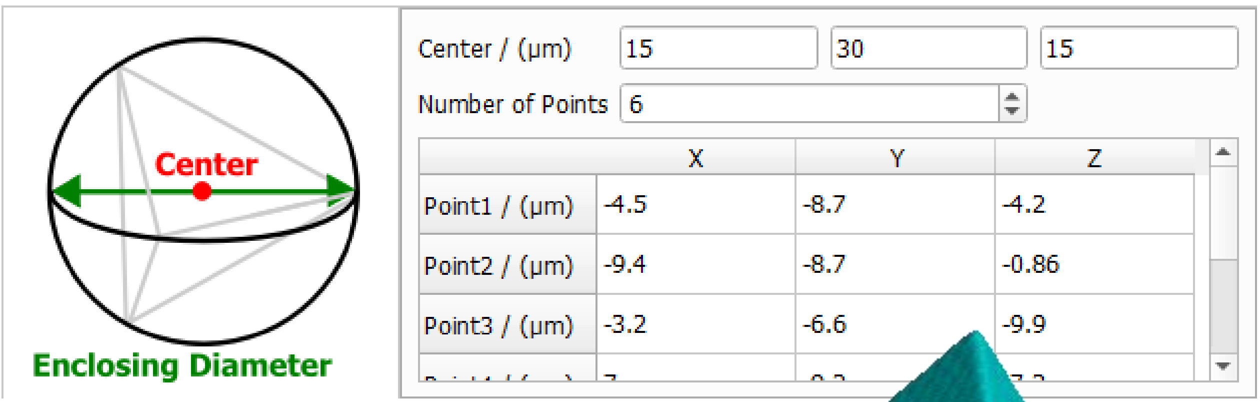


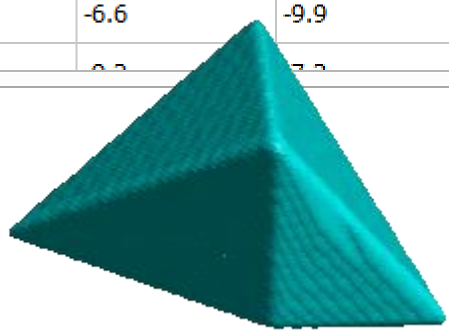
```
"Object5" : {
  "MaterialIDModel"     : "Constant",
  "MaterialID"          : 5,
  "Type"                : "PlanarPolyhedron",
  "Position"            : [9e-05,5e-05,8.5e-05],
  "Thickness"           : 5e-06,
  "Axis1"               : [0,0.707107,0.707107],
  "Axis2"               : [1,0,0],
  "Rays"                : 4,
  "RayLength1"          : 5e-06,
  "RayLength2"          : 1e-05,
  "RayLength3"          : 1.5e-05,
  "RayLength4"          : 2e-05,
  "RayAngle1"           : 80,
  "RayAngle2"           : 90,
  "RayAngle3"           : 100
},
```

The angle between the first ray and the second ray is given by the value after the key **RayAngle1** and so on. In the user interface, the number of rays is given implicitly by the number of entered and comma-separated **Ray Lengths**.

CONVEX POLYHEDRON

A convex polyhedron is defined by its center, given by the vector after the key **Position (Center)**, and some points, where the number of points is given by the value after the key **NumberOfPoints.**





The coordinates of the first point, relative to the center, are given by the vector after the key **Point1**.

Accordingly, the other relative coordinates are given by the values after the keys **Point2**, **Point3**, and so on. Using the specified points, the convex hull with the minimal volume is constructed. This hull defines the surface of the convex polyhedron. At least four points are needed to construct a convex hull.

```
"Object6" : {
  "MaterialIDModel"     : "Constant",
  "MaterialID"          : 6,
  "Type"                : "ConvexPolyhedron",
  "Position"            : [1.5e-05,3e-05,1.5e-05],
  "NumberOfPoints"      : 6,
  "Axis1"               : [1,0,0],
  "Axis2"               : [0,1,0],
  "Point1"              : [-4.5e-06,-8.7e-06,-4.2e-06],
  "Point2"              : [-9.4e-06,-8.7e-06,-8.6e-07],
  "Point3"              : [-3.2e-06,-6.6e-06,-9.9e-06],
  "Point4"              : [7e-06,-9.2e-06,-7.3e-06],
  "Point5"              : [-7.2e-06,9.5e-06,-9.9e-06],
  "Point6"              : [6.2e-06,9.4e-06,4.8e-06]
},
```

As for spheres, the additional parameters Axis1 and Axis2 in the .gad file describe the orientation of the object. Here, also, the visible 3D geometry does not depend on the orientation, the parameters are solely used to model anisotropic material behavior.

Fiber Shapes

The names used for the fiber **Type** key are composed by the fiber shape, followed by the profile (or cross-section) and the ending **Fiber**. The fiber profile is independent of the fiber shape.

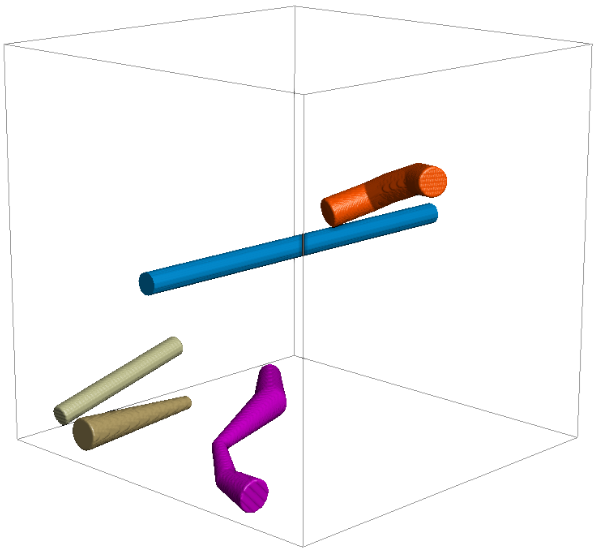
The fiber shapes are named **Short** (finite and straight), **Curved** (finite and curved), and **Infinite** (infinite and straight). For the **Short** and **Curved** fiber shapes exist more general types called **General Short** and **General Curved**, that allow for varying profile parameters along the length of the fiber.

In GeoDict, all combinations of the five different fiber shapes and the nine different fiber profiles are in principle possible, and might appear in a GAD file.

Shapes: **Short, GeneralShort, Infinite, Curved, GeneralCurved**

Profiles: **Circular, Hollow, Elliptical, Rectangular, Angular, Rosetta, Cellulose, Arbitrary, Strut**

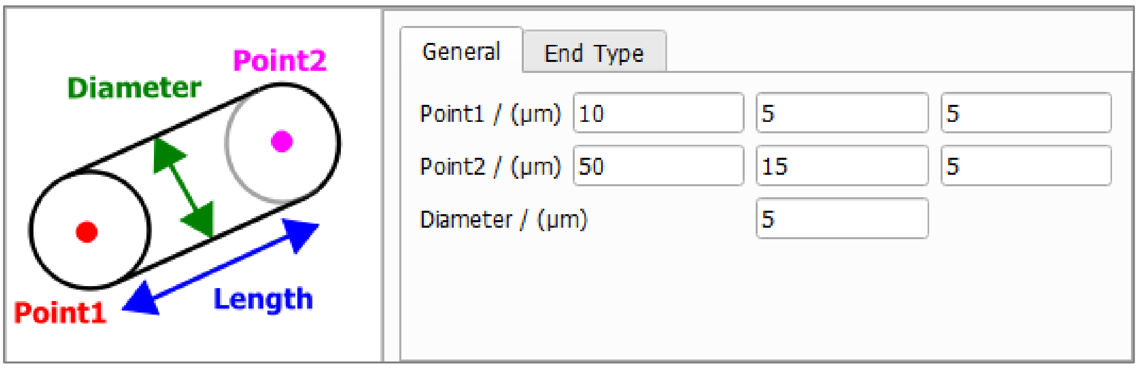
For example, **CurvedRosettaFiber** or **GeneralShortRectangularFiber** denote a **Type**.



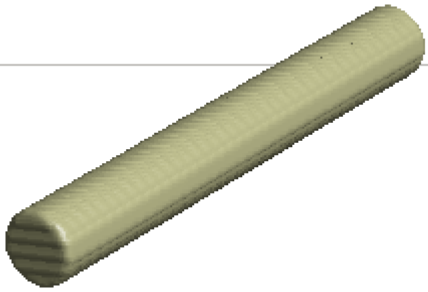
In the following, all fiber shapes are explained for the Circular profiles.

Short Fiber

A **Short** fiber is a straight fiber with a finite length. A short fiber is defined by two points, given by the vectors after the keys **Point1** and **Point2**, which specify the start and end points of the fiber.

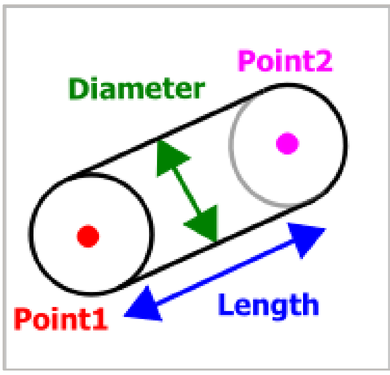


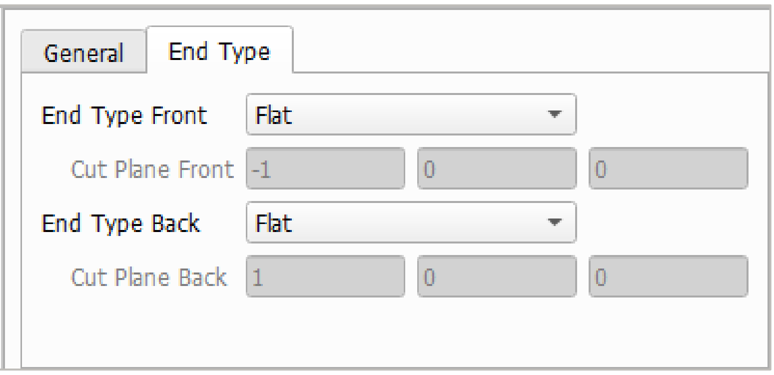
```
"Object1" : {
  "MaterialIDModel"      : "Constant",
  "MaterialID"           : 1,
  "Type"                 : "ShortCircularFiber",
  "Point1"               : [1e-05,5e-06,5e-06],
  "Point2"               : [5e-05,1.5e-05,5e-06],
  "FiberEndType1"        : "Flat", # Possible values: Flat, CuttingPlane, Rounded
  "FiberEndType2"        : "Flat", # Possible values: Flat, CuttingPlane, Rounded
  "Diameter"             : 5e-06
},
```



The circular fiber profile can be described by a single **Diameter**.

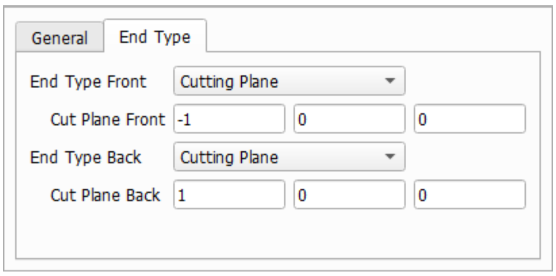
Fiber End Type





In general, the shape of the fiber endings is defined in the GAD format by the value after the keys **FiberEndType1 (End Type Front)** for the start point and **FiberEndType2 (End Type Back)** for the end point, which can have the following values:

- **Flat**: The fiber ending is planar, and the fiber direction is perpendicular to the according plane. No additional parameters are needed.
- **CuttingPlane**: The fiber ending is planar, and the according plane is perpendicular to a normal vector, specified by the vector after the keys **CuttingPlane1** (**Cut Plane Front**) for the start point and **CuttingPlane2** (**Cut Plane Back**) for the end point.




```
"FiberEndType1"       : "CuttingPlane",
"FiberEndType2"       : "CuttingPlane",
"CuttingPlane1"       : [-1,0,0],
"CuttingPlane2"       : [1,0,0],
```

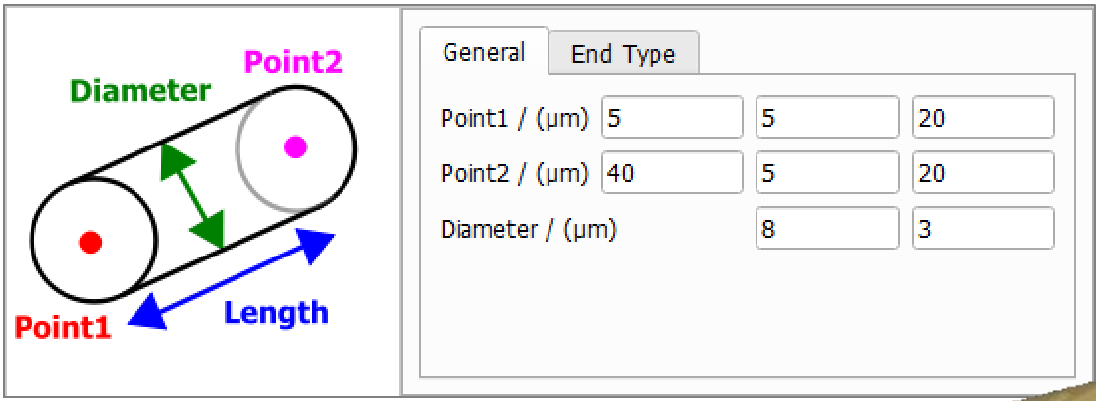
- **Rounded**: The fiber ending is rounded. No additional parameters are needed.

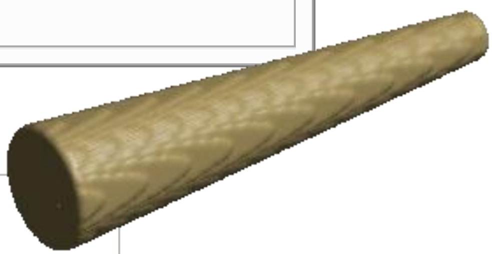
These options are available for all non-infinite fiber object types.

General Short fiber

The definition of the **GeneralShort** fiber is similar to the **Short** fiber one. The fiber is defined by two points, given by the vectors after the keys **Point1** and **Point2**.

The parameters of the fiber profile can change from the start to the end point. Hence, they have to be given for both points. For the circular fiber profile, this means that two diameters must be given, **Diameter1** (diameter at **Point1**) and **Diameter2** (diameter at **Point2**).





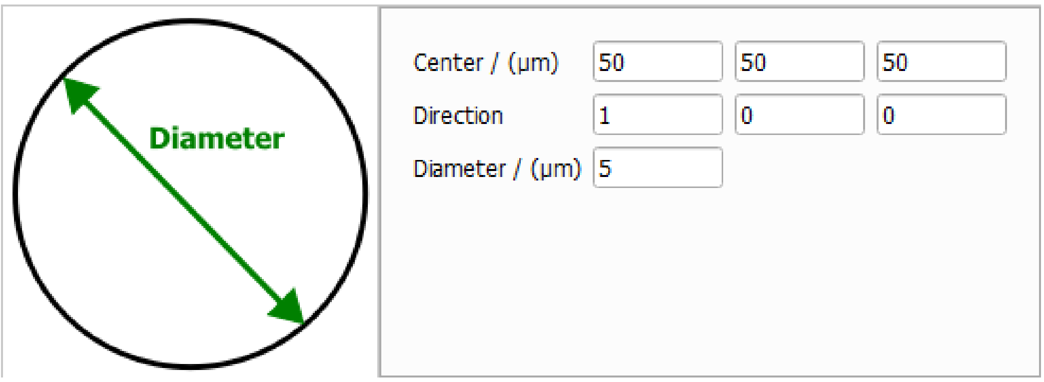
```
"Object2" : {
  "MaterialIDModel"     : "Constant",
  "MaterialID"          : 2,
  "Type"                : "GeneralShortCircularFiber",
  "Point1"              : [5e-06,5e-06,2e-05],
  "Point2"              : [4e-05,5e-06,2e-05],
  "FiberEndType1"       : "Flat",
  "FiberEndType2"       : "Flat",
  "Diameter1"           : 8e-06,
  "Diameter2"           : 3e-06
},
```

The fiber end types are defined as described above for the short fiber type.

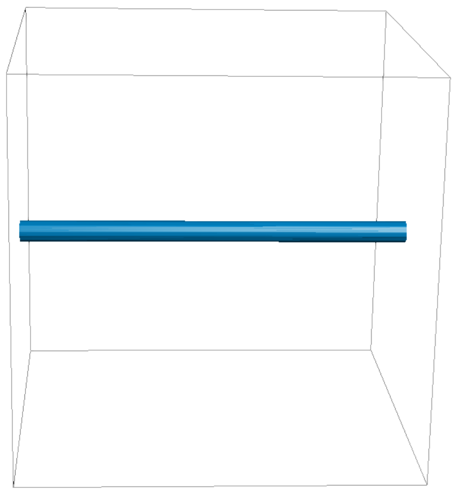
Infinite fiber

An **Infinite** fiber is a straight fiber with infinite length. The fiber is defined by one point, called **Center** in the user interface or **Position** in the GAD file, and one direction, given by the vector after the key **Direction**.

An infinite fiber cannot be created in a periodic domain.



```
"Object3" : {
  "MaterialIDModel"      : "Constant",
  "MaterialID"           : 3,
  "Type"                 : "InfiniteCircularFiber",
  "Position"             : [5e-05,5e-05,5e-05],
  "Direction"            : [1,0,0],
  "Diameter"             : 5e-06
},
```

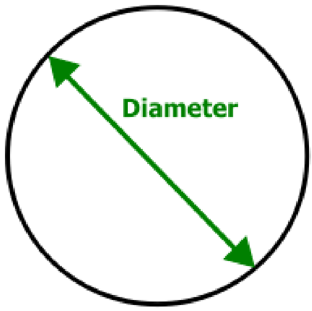


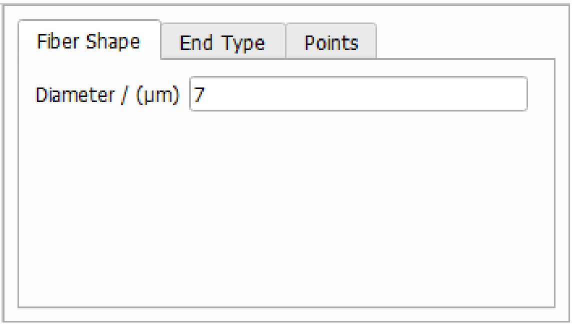
The circular fiber profile is again described by a single **Diameter**.

Naturally, infinite fibers do not have fiber end types.

Curved Fiber

A **CurvedFiber** is a non-straight finite fiber, composed of several linear segments. The linear segments form a polygonal line.

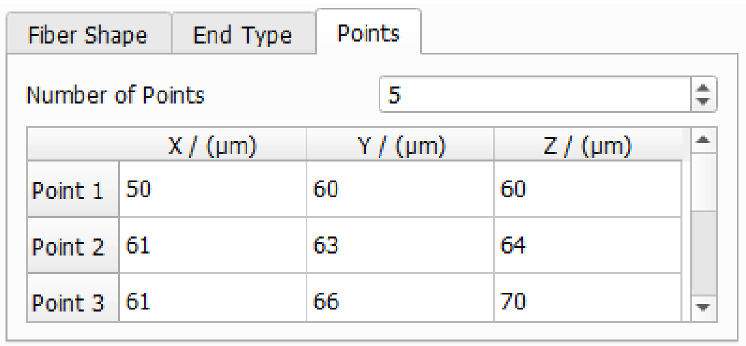




The circular fiber profile is again described by a single **Diameter**, and it is possible to select **Rounded Endings** in the user interface.

The fiber end types are defined as described above for the short fiber type.

The polygonal center line is described by a **Number of Points** and the positions of **Point1**, **Point2**, … in the table below.

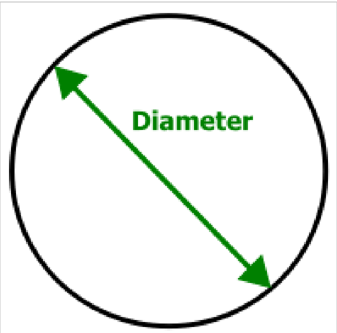


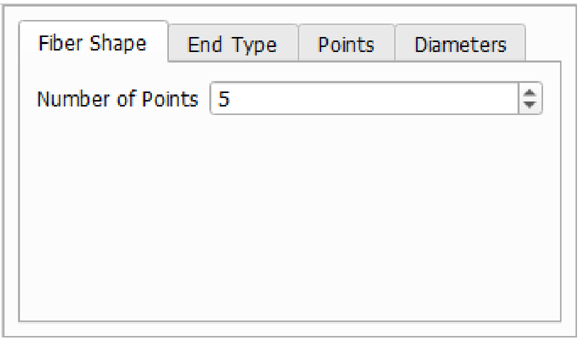
In the GAD file, the **NumberOfSegments** is given instead, which corresponds to the given **Number of Points** -1.

```
"Object4" : {
  "MaterialIDModel"     : "Constant",
  "MaterialID"          : 4,
  "Type"                : "CurvedCircularFiber",
  "NumberOfSegments"    : 4,
  "FiberEndType1"       : "Flat",
  "FiberEndType2"       : "Flat",
  "Diameter"            : 7e-06,
  "Point1"              : [5e-05,6e-05,6e-05],
  "Point2"              : [6.1e-05,6.3e-05,6.4e-05],
  "Point3"              : [6.1e-05,6.6e-05,7e-05],
  "Point4"              : [6e-05,7e-05,8e-05],
  "Point5"              : [5.6e-05,7.1e-05,8.8e-05]
},
```

General Curved Fiber

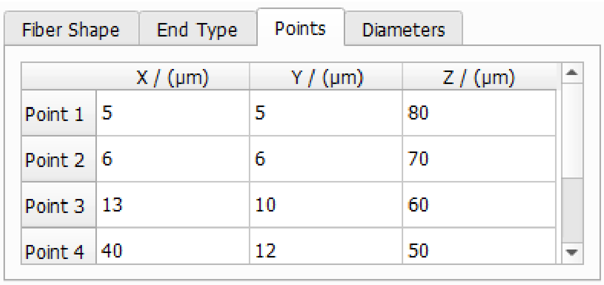
The **GeneralCurvedFiber** is the most general fiber type supported by GeoDict. The fiber is composed of several linear segments, and the fiber profile can change over the length of the fiber.



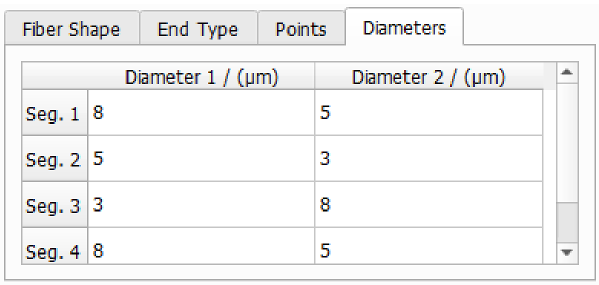


On the first tab, enter the **Number of Points** describing the polygonal center line.

The fiber end types are defined as described above for the short fiber type.



	X / (µm)	Y / (µm)	Z / (µm)
Point 1	5	5	80
Point 2	6	6	70
Point 3	13	10	60
Point 4	40	12	50



	Diameter 1 / (µm)	Diameter 2 / (µm)
Seg. 1	8	5
Seg. 2	5	3
Seg. 3	3	8
Seg. 4	8	5

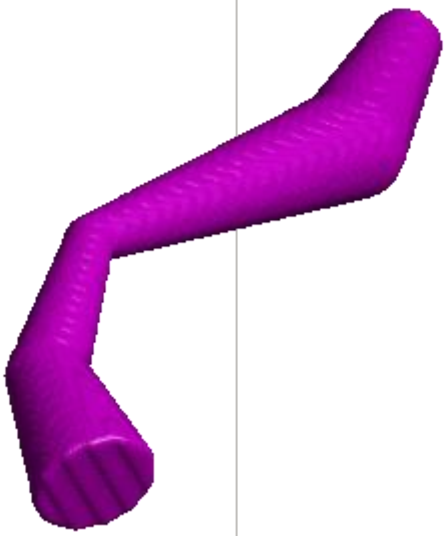
On the third tab, enter the coordinates of each point.

On the fourth tab, define the fiber profile for each segment, where the number of segments corresponds to the number of points minus one. For the circular fiber profile, this demands defining two diameters, one diameter at the start of a segment, and one at the end of a segment. It is not necessary that the diameter at the end of one segment matches to the diameter at the beginning of the next segment, but if they do, there will be a smooth transition of diameters along the fiber.

In the GAD file, the number of segments is given by the value after the key **NumberOfSegments**. The profile parameters are given in submaps: the parameters for the first segment are in **Segment1**, the parameters for the second are in **Segment2**, and so on.

The linear segments form a polygonal line. The coordinates of the points, which define the polygonal line, are given by the vectors after the keys **Point1**, **Point2**..., where **NumberOfSegments**+1 points are specified.

```
"Object5" : {
  "MaterialIDModel"      : "Constant",
  "MaterialID"           : 5,
  "Type"                 : "GeneralCurvedCircularFiber",
  "NumberOfSegments"     : 4,
  "FiberEndType1"        : "Flat",
  "FiberEndType2"        : "Flat",
  "Segment1" : {
    "Diameter1"          : 8e-06,
    "Diameter2"          : 5e-06
  },
  "Segment2" : {
    "Diameter1"          : 5e-06,
    "Diameter2"          : 3e-06
  },
  "Segment3" : {
    "Diameter1"          : 3e-06,
    "Diameter2"          : 8e-06
  },
  "Segment4" : {
    "Diameter1"          : 8e-06,
    "Diameter2"          : 5e-06
  },
  "Point1"               : [5e-06,5e-06,8e-05],
  "Point2"               : [6e-06,6e-06,7e-05],
  "Point3"               : [1.3e-05,1e-05,6e-05],
  "Point4"               : [4e-05,1.2e-05,5e-05],
  "Point5"               : [5e-05,1.5e-05,4e-05]
},
```



Fiber profiles

In the previous section, the profile of the circular fiber types could be described by a single diameter value, or two diameter values for each general segment.

In general, for the **Infinite**, **Short**, or **Curved** fiber shapes only a single set of descriptors of a fiber profile are needed.

For the **GeneralShort** fiber shape, two sets of descriptors are needed, one for the start point of the fiber, and one for the end point of the fiber.

For the **GeneralCurved** fiber shape, two sets of descriptors are needed for each segment, one for the start point of a segment, and one for the end point of a segment. One fiber may be composed of several segments.

If parameters change from the start to the end of the fiber (or a fiber segment), the parameters are linearly interpolated along the fiber (or the segment). Most interesting is when the perpendicular changes, for example of an elliptical profile.

In this case, the fiber is twisted, and the orientation of the fiber profile changes linearly along the fiber (torsion).

In detail, the parameters for the profile types are as follows:

- **Circular**:
  - **Infinite**, **Short**, or **Curved**: A circular profile is defined by one diameter, given by the value after the key **Diameter**.
  - **GeneralShort** or **GeneralCurved:** A circular profile is defined by the diameter at the start, given by the value after the key **Diameter1**, and the diameter at the end, given by the value after the key **Diameter2**.

- **Hollow:**
  - **Infinite**, **Short**, or **Curved**: A hollow profile is defined by one diameter, given by the value after the key **Diameter,** and one inner diameter, given by the value after the key **InnerDiameter**. The inner diameter describes the hollow part in the fiber center.
  - **GeneralShort** or **GeneralCurved:** A hollow profile is defined by one diameter and one inner diameter at the start, given by the values after the keys **Diameter1** and **InnerDiameter1**, and one diameter and one inner diameter at the end, given by the values after the keys **Diameter2** and **InnerDiameter2**.

- **Rosetta:**
  - **Infinite**, **Short**, or **Curved**: The contour of a Rosetta profile is defined by a circle where an additional sine-oscillation is added. The diameter of the circle is given by the value after the key **Diameter** and the amplitude of the sine function by the value after the key **AmplitudeFraction** multiplied by the **Diameter**. The number of maxima of the sine function is given by the value after the key **NumberOfLeaves**. The orientation of the profile is defined by the vector after the key **Perpendicular**.
  - **GeneralShort** or **GeneralCurved:** The Rosetta profile is defined by one diameter and one sine amplitude at the start, given by the values after the keys **Diameter1** and **AmplitudeFraction1**, and by one diameter and one sine amplitude at the end, given by the values after the keys **Diameter2** and

AmplitudeFraction2. The orientation at the start is given by the vector after the key **Perpendicular1** and the orientation at the end by the vector after the key **Perpendicular2**. The number of maxima of the sine function is given by the key **NumberOfLeaves**.

- **Elliptical:**
  - **Infinite**, **Short**, or **Curved**: An elliptical profile is described by two diameters, given by the values after the keys **Diameter1** and **Diameter2,** and a perpendicular, given by the vector after the key **Perpendicular**. **Diameter1** and **Diameter2** specify the profile size in the direction of the two main axes of the elliptical profile. The vector after the key **Perpendicular** is perpendicular to the direction of the fiber and points in the direction of **Diameter1**.
  - **GeneralShort** or **GeneralCurved:** An elliptical profile is defined by two diameters at the start, given by the values after the keys **Diameter1** and **Diameter2**, and two diameters at the end, given by the values after the keys **Diameter3** and **Diameter4**. The direction of **Diameter1** is given by the vector after the key **Perpendicular1** and the direction of **Diameter3** by the vector after the key **Perpendicular2**. When the values for **Perpendicular1** and **Perpendicular2** are different, the fiber (or segment) is twisted.

- **Cellulose:**
  - **Infinite**, **Short**, or **Curved**: The definition of a cellulose profile is similar to the one for the according elliptical profile. Additionally, it has two inner diameters, specifying the hollow part in the fiber center, given by the values after the keys **InnerDiameter1** and **InnerDiameter2**. **InnerDiameter1** points in the direction of **Diameter1** and **InnerDiameter2** points in the direction of **Diameter2**. The vector after the key **Perpendicular** is perpendicular to the direction of the fiber and points in the direction of **Diameter1**.
  - **GeneralShort** or **GeneralCurved:** The definition of a cellulose profile is similar to the one for the according elliptical profile. Additionally, it has two inner diameters at the start, given by the values after the keys **InnerDiameter1** and **InnerDiameter2,** and two inner diameters at the end, given by the values after the keys **InnerDiameter3** and **InnerDiameter4**. **InnerDiameter1** points in the direction of **Diameter1**, **InnerDiameter2** points in the direction of **Diameter2**, **InnerDiameter3** points in the direction of **Diameter3**, and **InnerDiameter4** points in the direction of **Diameter4**. The direction of **Diameter1** is given by the vector after the key **Perpendicular1** and the direction of **Diameter3** by the vector after the key **Perpendicular2**. When the values for **Perpendicular1** and **Perpendicular2** are different, the fiber (or segment) is twisted.

- **Rectangular:**
  - **Infinite**, **Short**, or **Curved**: The shape of a rectangular profile is defined by the two side lengths of a rectangle, given by the values after the keys **SideLength1** and **SideLength2**. The orientation of the profile perpendicular to the fiber direction is given by the vector after the key **Perpendicular**, were **SideLength1** points in the direction of **Perpendicular**.
  - **GeneralShort** or **GeneralCurved:** The shape of a rectangular profile is defined by the two side lengths of a rectangle at the start, given by the values after the keys **SideLength1** and **SideLength2**, and the two side lengths at the end, given by the values after the keys **SideLength3** and **SideLength4**.

The orientation of **SideLength1** perpendicular to the fiber direction is given by the vector after the key **Perpendicular1** and the orientation of **SideLength3** by the vector after the key **Perpendicular2**.

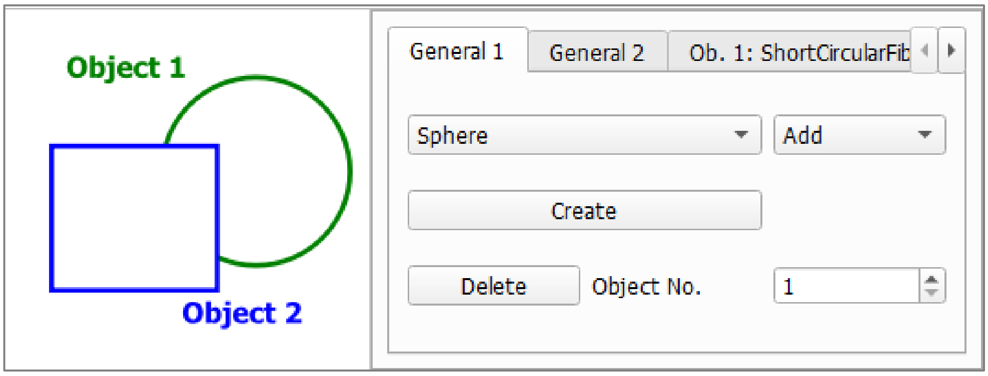
- **Angular:**
  - **Infinite**, **Short**, or **Curved**: The angular profile has the contour of a regular polygon enclosed by a circle, where all vertices of the polygon lie on the circle. The circle is defined by a diameter, given by the value after the key **Diameter**. The number of edges of the polygon is given by the value after the key **NumberOfEdges**. The orientation of the profile is defined by the vector after the key **Perpendicular**.
  - **GeneralShort** or **GeneralCurved:** The profile is described by one diameter at the start, given by the value after the key **Diameter1**, and one diameter at the end, given by the value after the key **Diameter2** (the meaning of the diameter is explained for the simple angular profile). The orientation at the start is given by the vector after the key **Perpendicular1** and the orientation at the end by the vector after the key **Perpendicular2**. The number of edges of the angular profile is given by the value after the key **NumberOfEdges**.

- **Arbitrary:**
  - **Infinite**, **Short**, or **Curved**: The profile is described by a comma-separated **Radius** list and an **Angle** list. The first listed radius is achieved at an angle of 0°, the following radii are achieved at the given angles. The orientation of the profile is defined by the vector after the key **Perpendicular**. The **Curved** fiber shape is not editable in GadGeo and cannot be selected in the **Create / Add GAD Objects** command.
  - **GeneralShort** or **GeneralCurved:** The profile at the start is described by **Radius1** and **Angle1**, the profile at the end is described by **Radius2** and **Angle2**. The orientation at the start is given by the vector after the key **Perpendicular1** and the orientation at the end by the vector after the key **Perpendicular2**.
    The **GeneralCurved** fiber shape is not editable in GadGeo and cannot be selected in the **Create / Add GAD Objects** command.

- **Strut:**
  - **Infinite**, **Short**, or **Curved**: The triangular profile is described by the **Diameter** and an **Angle** list with two entries. The orientation of the profile is defined by the vector after the key **Perpendicular**. These fiber shapes are not editable in GadGeo and cannot be selected in the **Create / Add GAD Objects** command.
  - **GeneralShort** or **GeneralCurved:** The profile at the start is described by **Diameter1** and **Angle**, the profile at the end is described by **Diameter2** and **Angle**. The orientation at the start is given by the vector after the key **Perpendicular1** and the orientation at the end by the vector after the key **Perpendicular2**.
    Only the **GeneralCurved** type is editable in GadGeo and can be selected in the **Create / Add GAD Objects** command.

COMBINED OBJECT

A combined object is a combination of sub-objects, i.e., any object type explained until now (**Sphere**, **Ellipsoid**, **Box** …). Objects can be combined by adding different objects together, or by subtracting an object from another one.

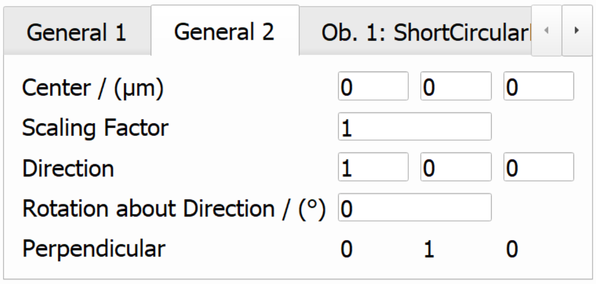


In the **General 1** tab, standard objects can be added to the combined type, or subtracted from the combined type. Upon startup, a Combined Object consists of a single sphere, which can be removed by deleting **Object No.** 1 using the **Delete** button.

It is important to distinguish between **Delete** and **Subtract**. While **Delete** removes an object from the list of objects, **Subtract** subtracts the defined object from an added object. With this, it is for example possible to create a cylindrical hole through a sphere by adding a **Sphere** and subtracting a **ShortCircularFiber** as shown in the example below.

For each object that is created here (by clicking **Create**), a new tab widget will appear, where the tab header states the object number, the type name, and if the object is added or subtracted.

In the **General 2** tab, the position of the **Center** of the new object can be defined and the main orientation vector can be defined through the **Direction** vector.



Center, **Scaling Factor**, **Direction**, and **Perpendicular** set up a local coordinate system for the object. **Center** (called **Position** in the GAD file) is the origin of the local coordinate system, **Direction** (**Axis1** in the GAD file) is the x-axis and **Perpendicular** (**Axis2)** the y-axis, whose position can be freely chosen in a plane by setting the **Rotation about Direction**.

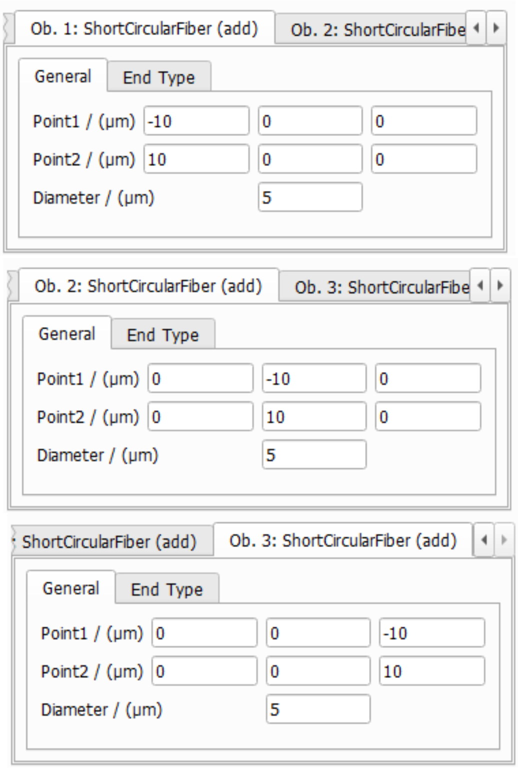
The z-axis is then already automatically defined as the cross product of the two other axes, such that a right hand system is formed. The **ScalingFactor** sets the relation between object sizes in the local object coordinates and the global Euclidean coordinate system. A **Scaling Factor** of 2 means that objects will appear twice as large as given in the local coordinate system.

All coordinates entered in the Object tabs will use this local coordinate system.

The standard objects that create the combined object are defined in the subsequent tabs.

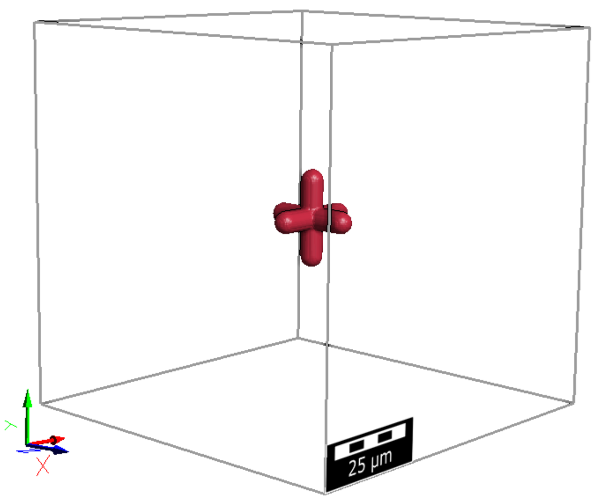
In the GAD file, the number of sub-objects is given by the value after the key **NumberOfSubObjects**.

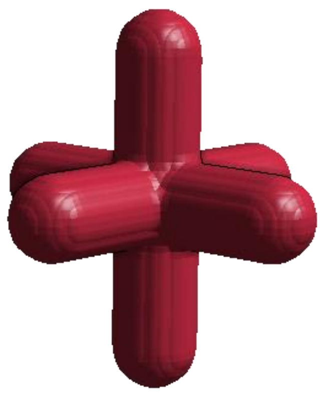
```
"Object1" : {
  "MaterialIDModel"        : "Constant",
  "MaterialID"             : 1,
  "Type"                   : "CombinedObject",
  "NumberOfSubObjects"     : 3,
  "Position"               : [0,0,0],
  "ScalingFactor"          : 1,
  "Axis1"                  : [1,0,0],
  "Axis2"                  : [0,1,0],
  "SubObject1" : {
    "Operation"            : "add",
    "Type"                 : "ShortCircularFiber",
    "Point1"               : [-1e-05,0,0],
    "Point2"               : [1e-05,0,0],
    "FiberEndType1"        : "Rounded",
    "FiberEndType2"        : "Rounded",
    "Diameter"             : 5e-06
  },
  "SubObject2" : {
    "Operation"            : "add",
    "Type"                 : "ShortCircularFiber",
    "Point1"               : [0,-1e-05,0],
    "Point2"               : [0,1e-05,0],
    "FiberEndType1"        : "Rounded",
    "FiberEndType2"        : "Rounded",
    "Diameter"             : 5e-06
  },
  "SubObject3" : {
    "Operation"            : "add",
    "Type"                 : "ShortCircularFiber",
    "Point1"               : [0,0,-1e-05],
    "Point2"               : [0,0,1e-05],
    "FiberEndType1"        : "Rounded",
    "FiberEndType2"        : "Rounded",
    "Diameter"             : 5e-06
  }
},
```



Each sub-object has the additional key **Operation** with the value **add** or **subtract**, with **Operation add** being the default. The operation specifies if a sub-object is added or subtracted from the geometry of the combined object.

In the example above, three cylinders form a 3D cross as a new object type.

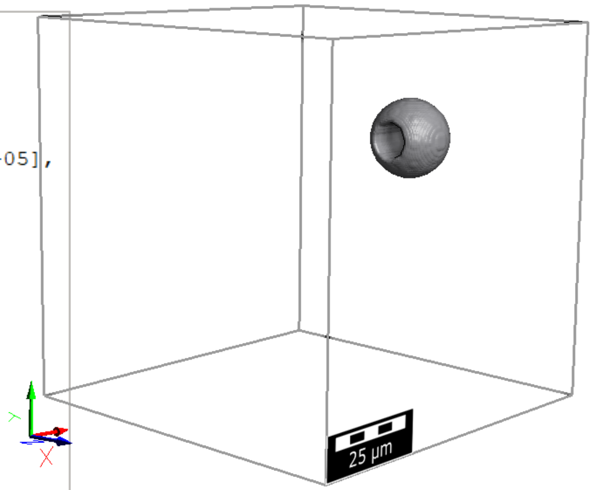




As already mentioned above, the coordinates entered in these tabs refer to the local coordinate system of the object, not the global coordinates of the domain.

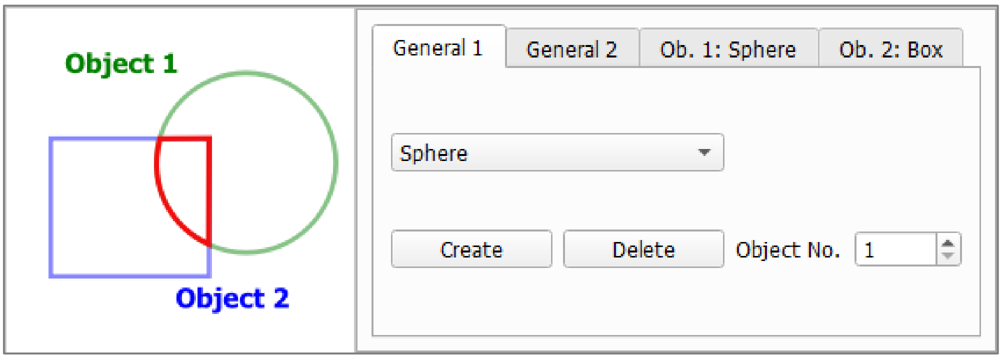
Besides the possibility to add standard objects to a new combination, **subtract** allows to cut away parts of an object. Using a combined object with a **ShortCircularFiber** subtracted from a **Sphere** results in the following shape:

```
"Object2" : {
  "MaterialIDModel"        : "Constant",
  "MaterialID"             : 2,
  "Type"                   : "CombinedObject",
  "NumberOfSubObjects"     : 2,
  "Position"               : [1.5e-05,2e-05,2e-05],
  "ScalingFactor"          : 2,
  "Axis1"                  : [1,0,0],
  "Axis2"                  : [0,1,0],
  "SubObject1" : {
    "Operation"            : "add",
    "Type"                 : "Sphere",
    "Position"             : [0,0,0],
    "Diameter"             : 1e-05,
    "Axis1"                : [1,0,0],
    "Axis2"                : [0,1,0]
  },
  "SubObject2" : {
    "Operation"            : "subtract",
    "Type"                 : "ShortCircularFiber",
    "Point1"               : [-1e-05,0,0],
    "Point2"               : [1e-05,0,0],
    "FiberEndType1"        : "Flat",
    "FiberEndType2"        : "Flat",
    "Diameter"             : 5e-06
  }
},
```



Intersected Object

An intersected object is specified by an intersection of sub-objects. A sub-object may be any of the object types: **Sphere**, **Ellipsoid**, **Box**, etc.

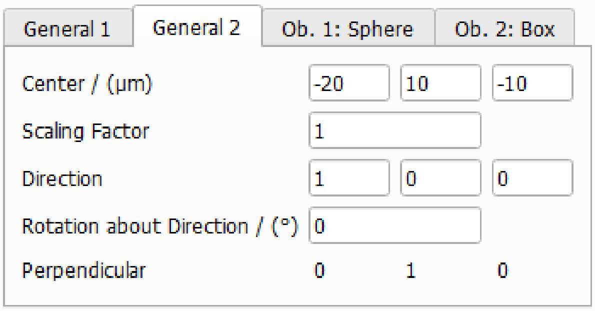


In the **General 1** tab, standard objects can be added to the intersection. Upon startup, an IntersectedObject consists of a single sphere, which can be removed by deleting **Object No.** 1 using the **Delete** button.

For each object that is created here (by clicking **Create**), a new tab widget will appear, where the tab header states the object number and the type name.

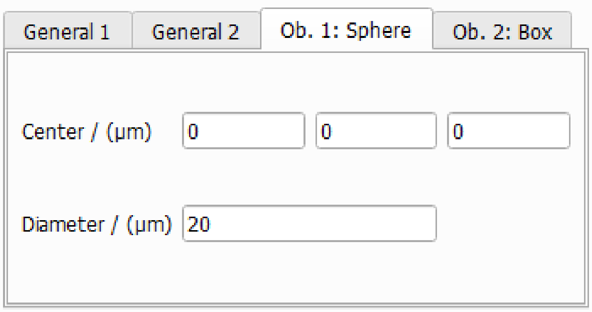
In the **General 2** tab, the position of the new object can be defined and the local coordinate system can be set up as described in the **CombinedObject** section above.

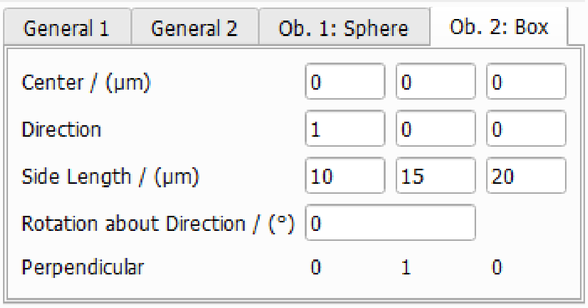
All coordinates entered in the Object tabs will use this local coordinate system.



The standard objects that create the intersected object are defined in the subsequent tabs.

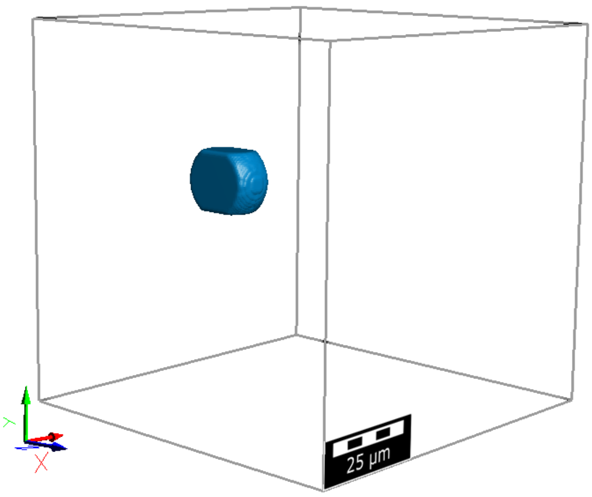
```
"Object3" : {
  "MaterialIDModel"          : "Constant",
  "MaterialID"               : 3,
  "Type"                     : "IntersectedObject",
  "NumberOfSubObjects"       : 2,
  "Position"                 : [-2e-05,1e-05,-1e-05],
  "ScalingFactor"            : 1,
  "Axis1"                    : [1,0,0],
  "Axis2"                    : [0,1,0],
  "SubObject1" : {
    "Type"                   : "Sphere",
    "Position"               : [0,0,0],
    "Diameter"               : 2e-05,
    "Axis1"                  : [1,0,0],
    "Axis2"                  : [0,1,0]
  },
  "SubObject2" : {
    "Type"                   : "Box",
    "Position"               : [0,0,0],
    "Length1"                : 1e-05,
    "Length2"                : 1.5e-05,
    "Length3"                : 2e-05,
    "Axis1"                  : [1,0,0],
    "Axis2"                  : [0,1,0],
    "CornerMode"             : "Sharp",
    "CornerRadius"           : 0
  }
},
```

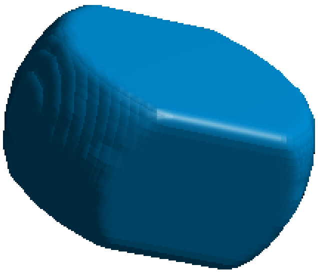




In the GAD file, the number of sub-objects is given by the value after the key **NumberOfSubObjects**.

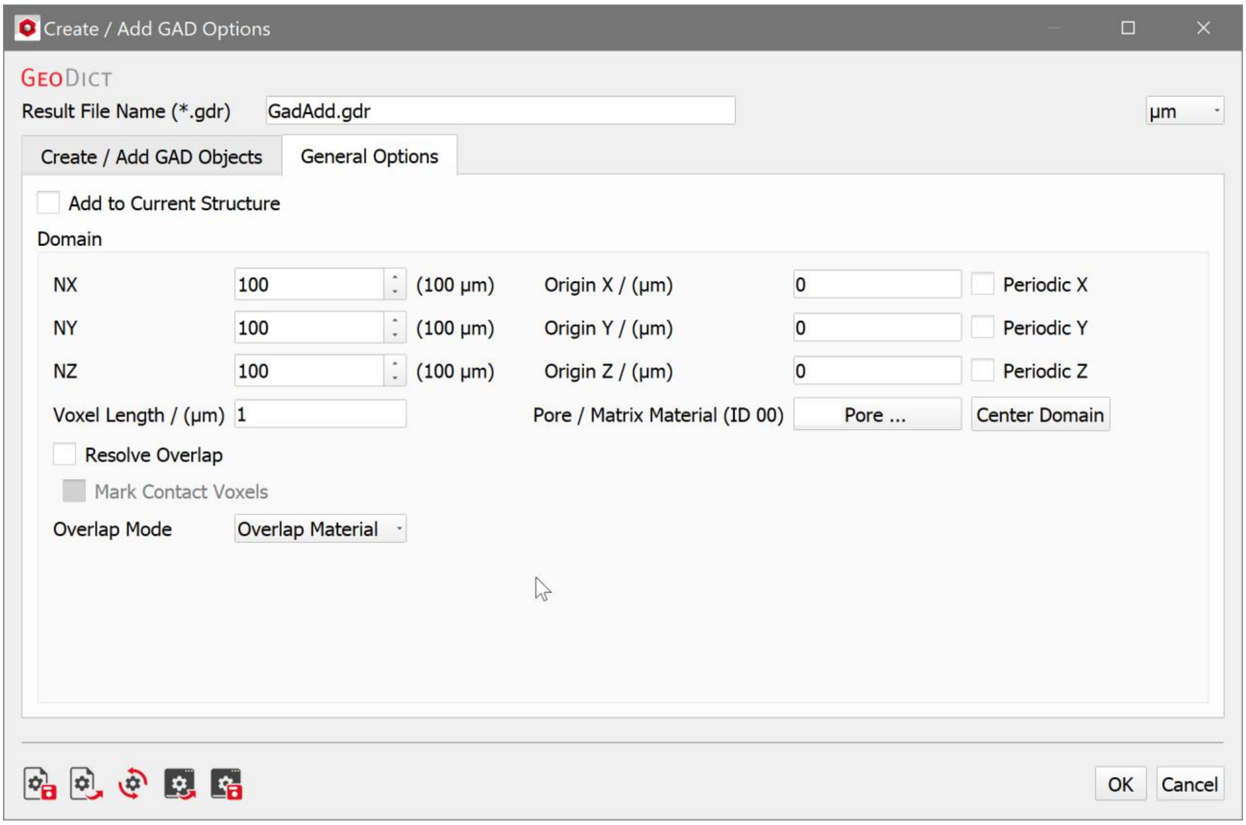
In the example shown here, a **Sphere** is intersected with a **Box**.





General Options

Checking **Add to Current Structure** results in a new object being added to the currently loaded structure. **NX**, **NY**, and **NZ** define the number of voxels in X-, Y-, and Z-direction, and **Voxel Length** establishes the voxel size.



Origin X, **Origin Y**, and **Origin Z** control the Cartesian origin of the structure, and checking **Periodic X**, **Periodic Y**, and/or **Periodic Z** allows generating periodic structures.

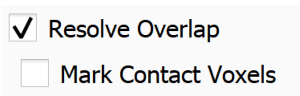
Click **Center Domain** to move the structure so that the Cartesian coordinates' origin (0,0,0) lies in its center.

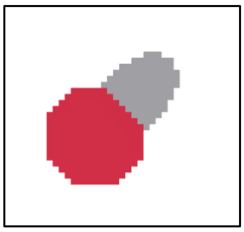
Choose the material of the pore space or the matrix around the fibers from the **Pore/Matrix Material (ID 00)** pull-down menu. When the material with Material ID 00 is chosen to be **Pore**, the objects in the structure are embedded in empty pore space. Otherwise, the material filling the matrix can be chosen to be any of the **Solid**, **Porous**, or **Fluid** materials currently in the GeoDict Material Database.

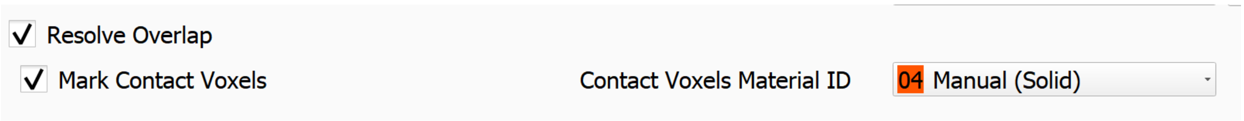
Handling overlap between structures

Generated objects may overlap, and a decision has to be made how the overlap will be treated. Overlapping areas may be assigned to one of the objects or form an "overlap object" themselves.

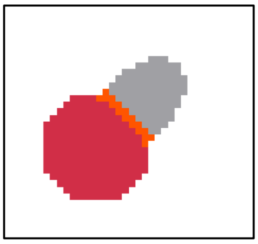
If **Resolve Overlap** is chosen, the overlap is resolved after the structure is generated. Every voxel in the overlap area is assigned to one of the objects. This assignment is done based on a watershed algorithm on the overlap area.





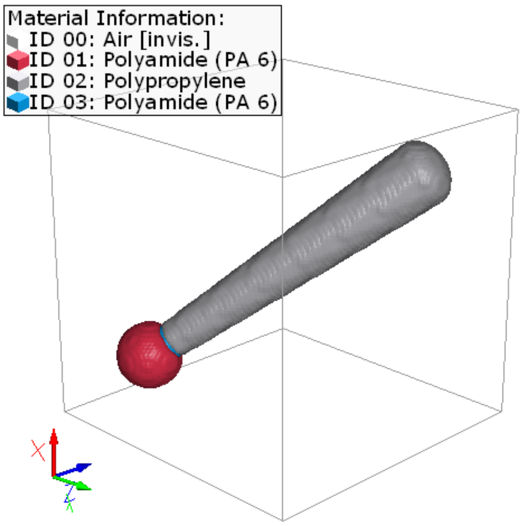


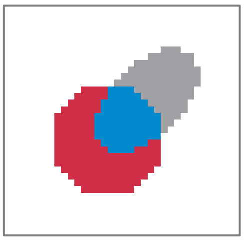
If additionally, **Mark Contact Voxels** is checked, a **Contact Voxels Material ID** must be selected. Voxels on the boundary between two objects are then given this material ID.



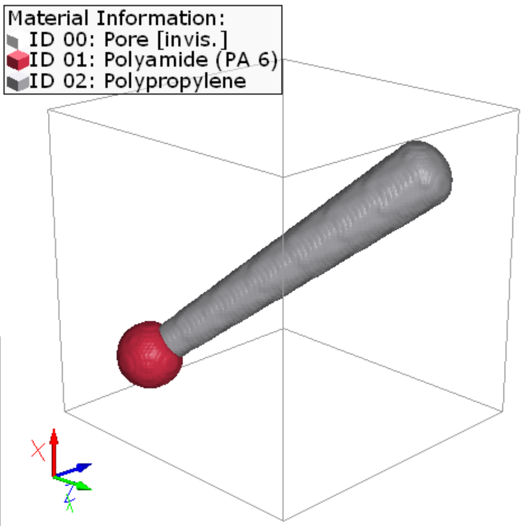
If the overlap stays unresolved, overlapping areas form additional *overlap objects*. The **Overlap Mode** defines how a material ID is assigned to these *overlap objects*.

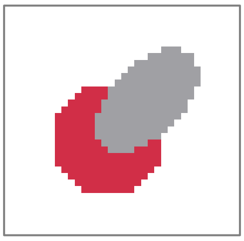
The overlap areas can be assigned to be the default **Overlap Material**. The ID of the overlap area is obtained by the bitwise sum of the IDs of the overlapping materials.



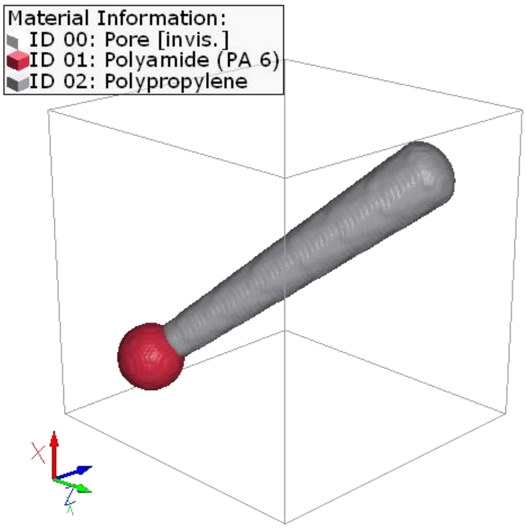


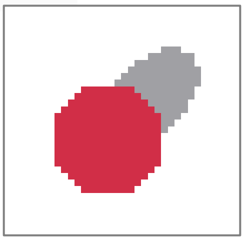
The overlap can take the ID of the **New Material**. Objects are added in the order as listed under the Create/Add GAD Options tab. This means that in this case, the overlap area is of the same material as the object that was added later.



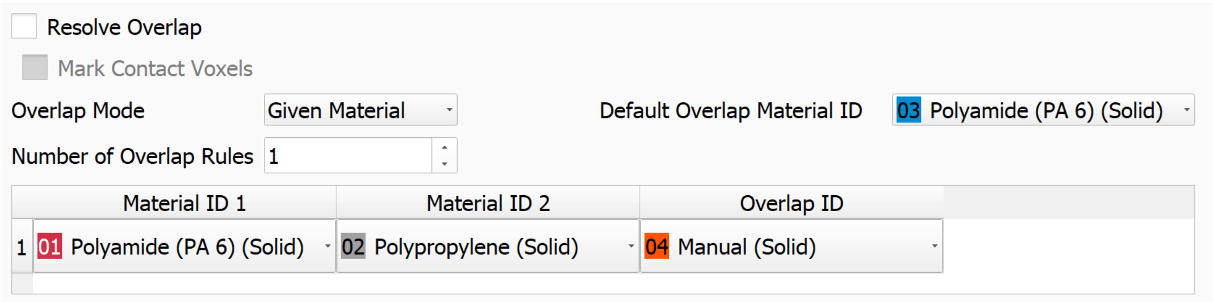


The overlap can take the ID of the **Old Material.** Objects are added in the order as listed under the Create/Add GAD Options tab. This means that in this case, the overlap area is of the same material as the object that was added earlier.



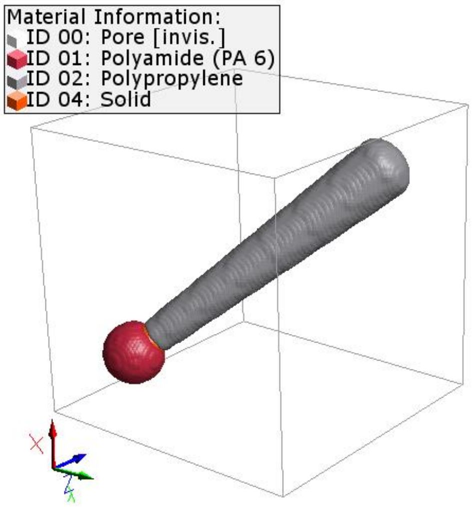


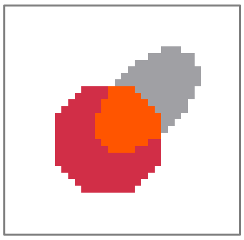
The overlap can be also chosen to be a **Given Material** freely selected by the user from the GeoDict Material Database.



In this case, a **Default Overlap Material ID** can be chosen, and an arbitrary **Number of Overlap Rules** can be defined. The default material is used, whenever none of the user defined overlap rules applies.

As explained above in page 7, clicking **Run** applies these selected options.



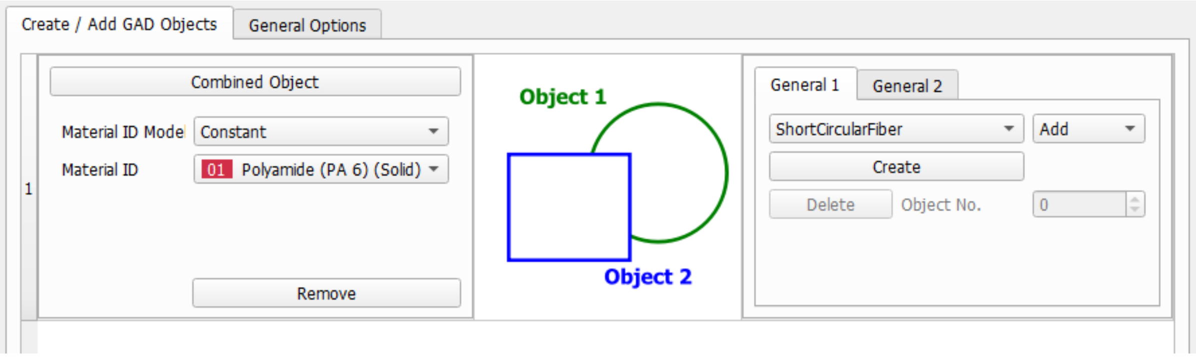


Creating User-Defined object types

The object type CombinedObject can be used to define a user-defined object type that can then be used, e.g., in GrainGeo alternatively to the built-in object types when creating a 3D structure model.

For this, create a single CombinedObject with the GadGeo -Create/Add GAD Objects command and save it as a .gad file as follows:

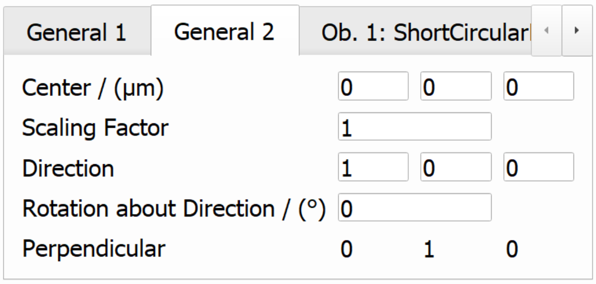
At first, remove all objects from the **Create/Add GAD Objects** tab and create a new CombinedObject.



In the **General 1** tab, standard objects can be added to the combined type, or subtracted from the combined type. Upon startup, a CombinedObject consists of a single sphere, which can be removed by deleting **Object No.** 1 using the **Delete** button. For each object that is created here (by clicking **Create**), a new tab widget will appear, where the tab header states the object number, the type name, and if the object is added or subtracted.

In the **General 2** tab, the position of the **Center** of the new object can be defined and the main orientation vector can be defined through the **Direction** vector.

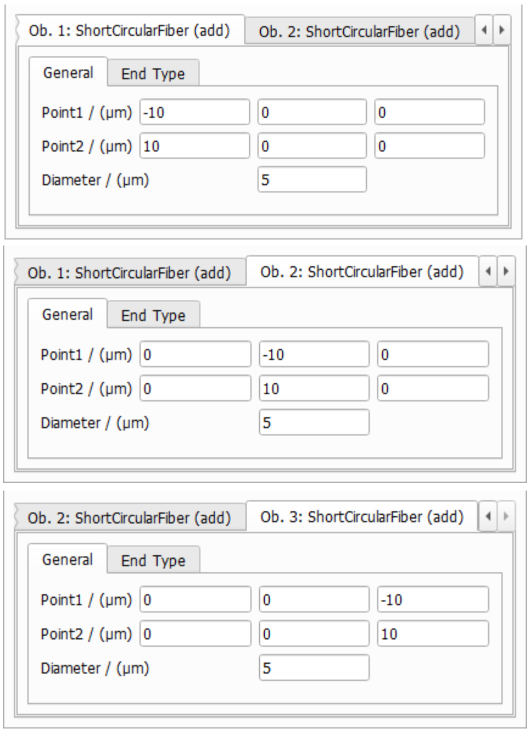
When using GadGeo to define a user-defined object, the global position and rotation of the sample object is not relevant, so for simplicity, it is recommended to use the default cartesian coordinate system also as local coordinate system.



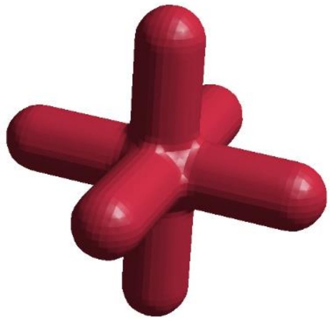
Center and **Direction** will also be used to describe the position of the new object when using it in a structure generator like GrainGeo, or inside the GAD file. Therefore, when creating a new object type in the default coordinate system, the object should be defined such that its center of mass lies in the origin (0,0,0) and the main direction of the object is the x-axis (1,0,0). If defined otherwise, using the object type in GrainGeo may give unexpected results, e.g. when defining an orientation distribution or a distribution of the object locations.

Be aware that GeoDict will not check if the so defined center is the center of mass of the new object or even if the center is inside of the object.

The standard objects that create the new combined object are defined in the subsequent tabs.



In the example to the left, three cylinders form a 3D cross as a new object type:

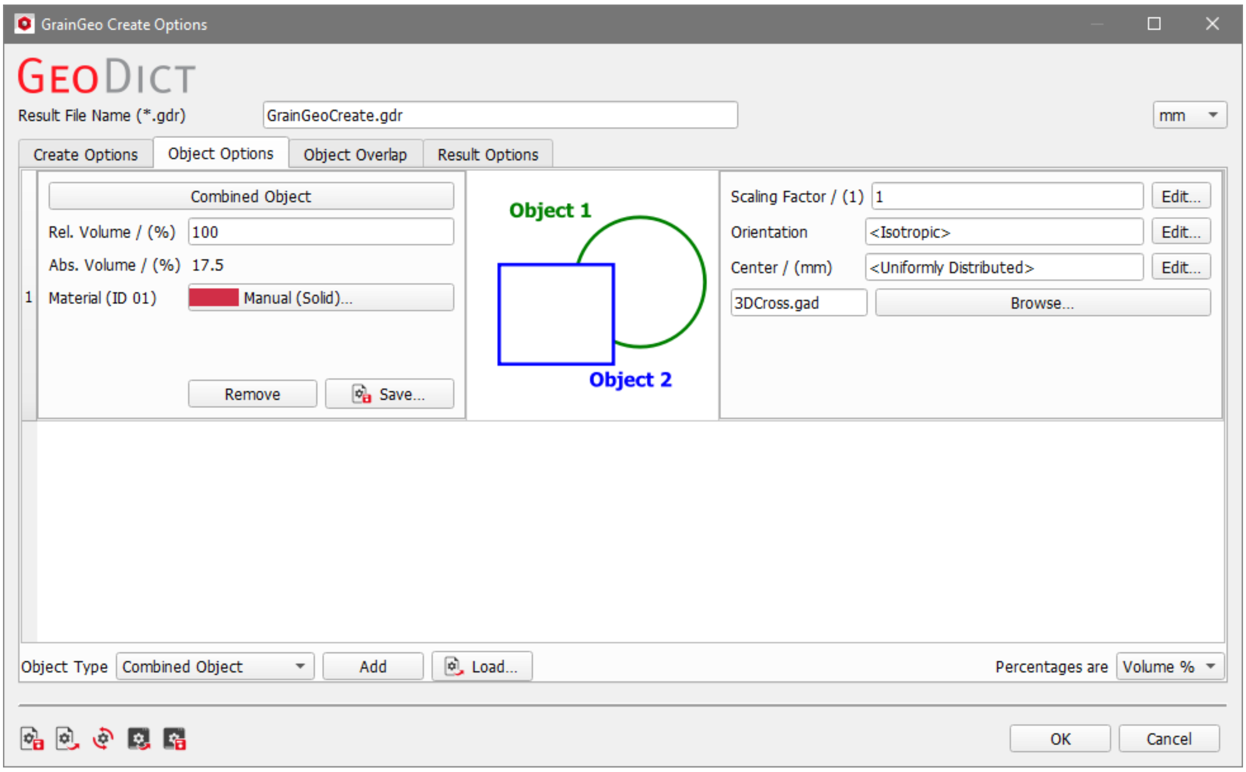


As already mentioned above, the coordinates entered in these tabs refer to the local coordinate system of the object, not the global coordinates of the domain.

After the parameters have been entered, close the dialog and create the structure. Then, save the structure as *.gad file using **File->Save Structure As...** from the main menu. To be able to use the file later as user-defined combined object, it is important that only a single **CombinedObject** is stored inside of the file.

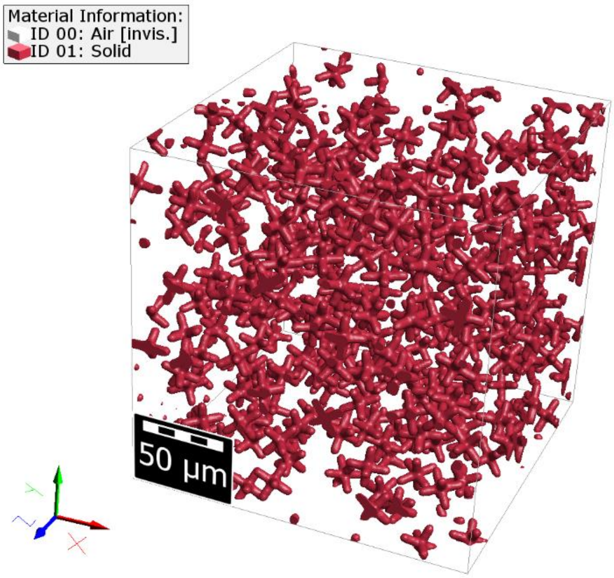
After the file has been stored, it is possible to use it in GeoDict's structure generators. For example, in GrainGeo **Create Grains** choose **Combined Object** as **Object Type**, **Add** it and then **Browse...** for the *.gad file just created.


```
"Object1" : {
  "MaterialIDModel"        : "Constant",
  "MaterialID"             : 1,
  "Type"                   : "CombinedObject",
  "NumberOfSubObjects"     : 3,
  "Position"               : [0,0,0],
  "ScalingFactor"          : 1,
  "Axis1"                  : [1,0,0],
  "Axis2"                  : [0,1,0],
  "SubObject1" : {
    "Operation"            : "add",
    "Type"                 : "ShortCircularFiber",
    "Point1"               : [-1e-05,0,0],
    "Point2"               : [1e-05,0,0],
    "FiberEndType1"        : "Rounded",
    "FiberEndType2"        : "Rounded",
    "Diameter"             : 5e-06
  },
  "SubObject2" : {
    "Operation"            : "add",
    "Type"                 : "ShortCircularFiber",
    "Point1"               : [0,-1e-05,0],
    "Point2"               : [0,1e-05,0],
    "FiberEndType1"        : "Rounded",
    "FiberEndType2"        : "Rounded",
    "Diameter"             : 5e-06
  },
  "SubObject3" : {
    "Operation"            : "add",
    "Type"                 : "ShortCircularFiber",
    "Point1"               : [0,0,-1e-05],
    "Point2"               : [0,0,1e-05],
    "FiberEndType1"        : "Rounded",
    "FiberEndType2"        : "Rounded",
    "Diameter"             : 5e-06
  }
},
```



The **Center** and **Orientation** used in the structure generator refer to the center point and the direction defined previously in the **General 2** tab (the origin and the x-axis of the local coordinate system of the object).

Besides this, a variation of the object size is possible with the **Scaling Factor**. Here, it is not possible anymore to access or vary the individual sizes of the standard objects that make up the combined object.

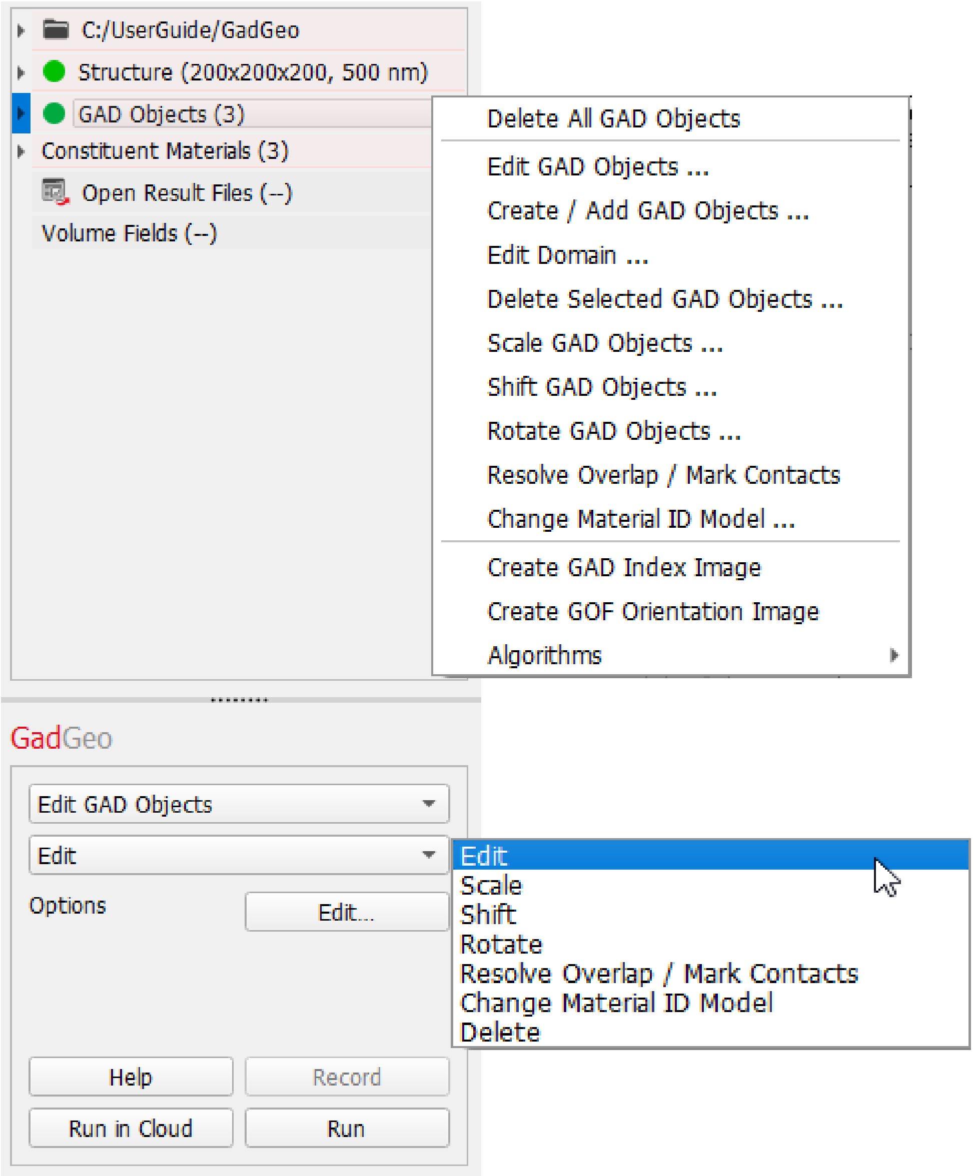


Closing the dialog and clicking **Generate** creates a structure consisting of the user defined type. Note that, although a CombinedObject consists of several sub-objects, it is not possible to select different materials or material IDs for the different parts here. However, it is possible to change that in a post-processing step using the Change Material ID Model command (see page 43)

Edit GAD Objects

With **Edit GAD Objects**, objects of previously generated structure models in GAD format can be edited.

Seven commands are selectable in the pull-down menu (**Edit**, **Scale**, **Shift**, **Rotate, Resolve Overlap / Mark Contacts, Change Material ID Model** or **Delete**) in the GadGeo section. The same commands are also available from the **Objects'** context menu in the Project Status section located above:



Edit is available as **Edit GAD Objects...**

Scale is available as **Scale GAD Objects...**

Shift is available as **Shift GAD Objects...**

Rotate is available as **Rotate GAD Objects...**

Delete is available as **Delete Selected GAD Objects...**

Click the **Options' Edit...** button to set the options for this change. After setting the options, return to the GadGeo section and **Run** to make the changes effective.

GadGeo modifies GAD objects. That means it can only be used if the structure's objects are analytic and so, their parameters are known to GeoDict. This is indicated by a green dot in front of **GAD Objects** in the **Status** section. If a red dot is shown, no information about the objects is available, and the structure is only defined through its voxel grid. Such structures are voxelized representations, occurring e.g. when importing tomographic images or after using certain commands in some GeoDict generating modules, e.g. **GrainGeo** → **Grow Sediment** or **GrainGeo** → **Add Binder**, or modifying modules, e.g. ProcessGeo.

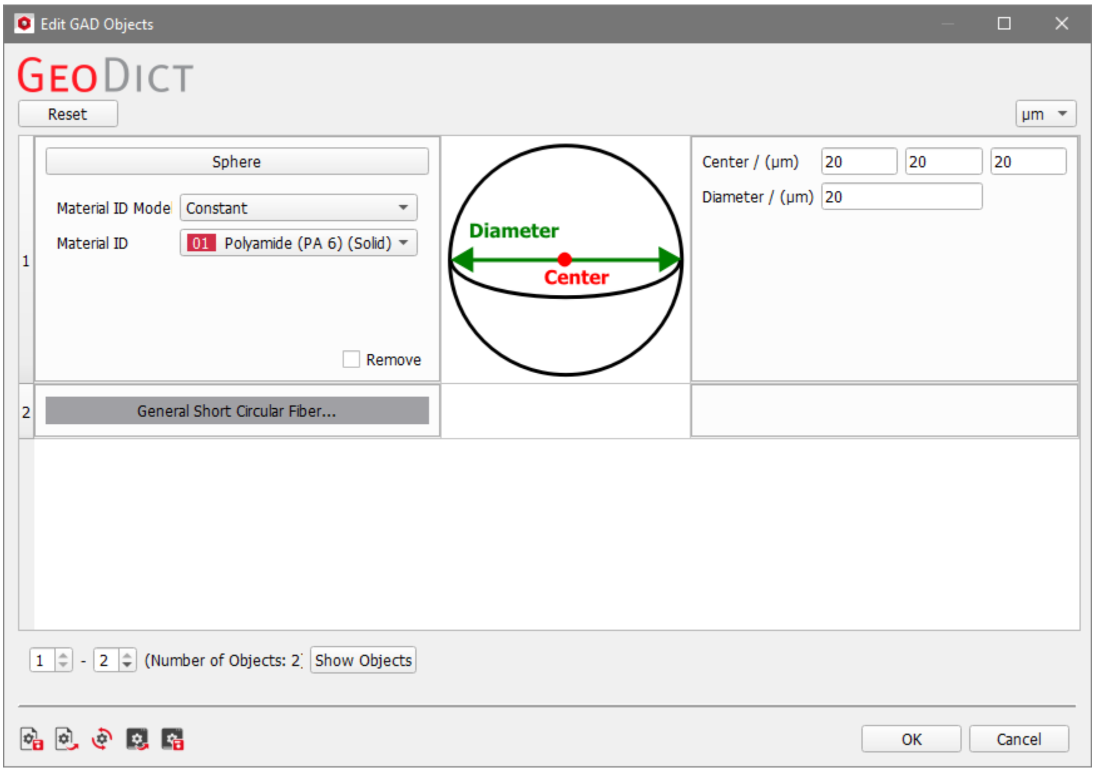
In this case, when the user tries to change a structure (**Scale**, **Shift**, **Rotate**, and **Delete**) a warning message appears when clicking the corresponding button and the operation is not performed.

Non-analytic data structures can be edited using the  icon in the Toolbar.

EDIT

When the user selects **Edit** and clicks the **Edit...** button, the **Edit GAD Objects** dialog opens. In it, every single object in the structure is listed, with all geometrical properties.

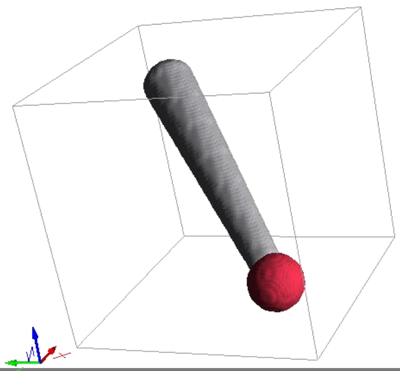
Clicking on the **Sphere** button in the example below opens the corresponding widget and all details are shown:

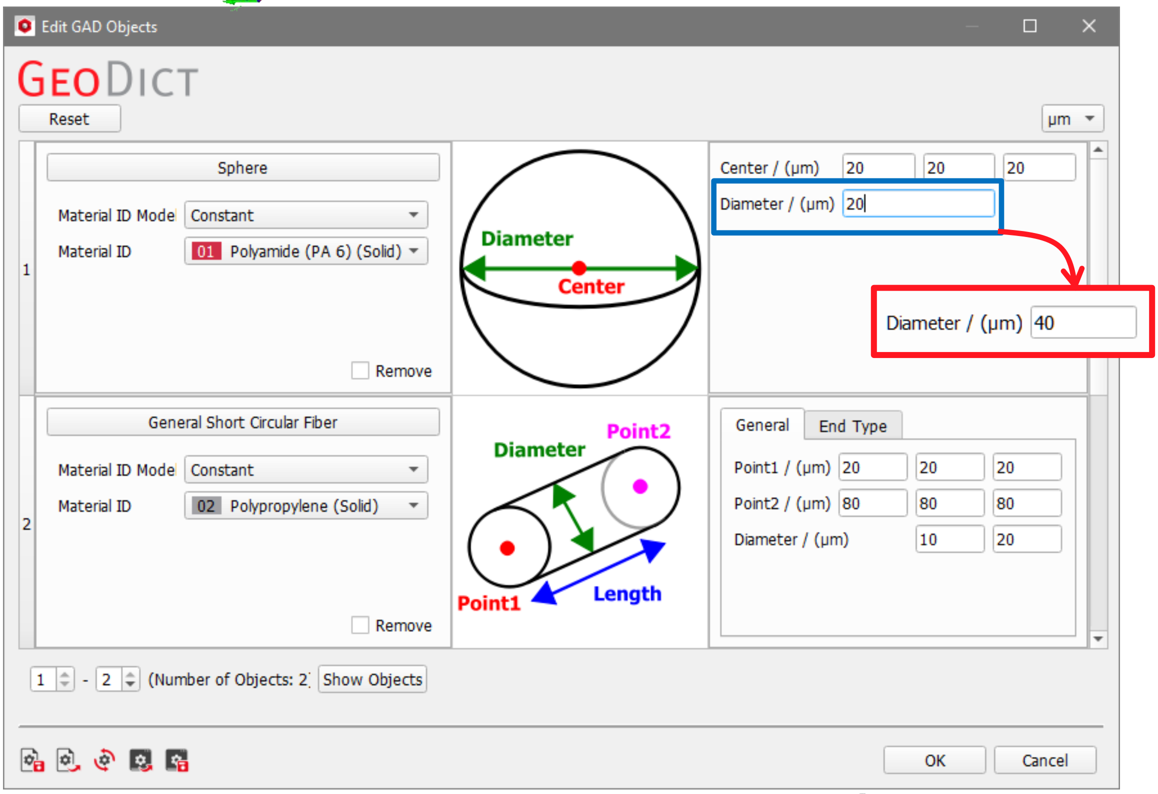


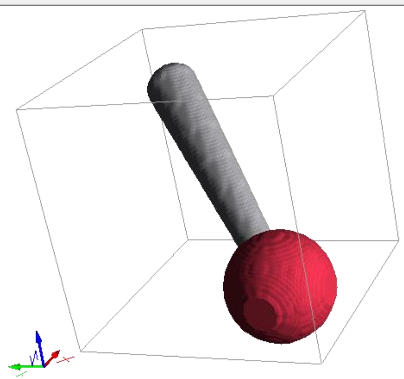
The dialog gives the choice to **Remove** single objects altogether, to change the material ID assigned to any of them, as well as to modify their size, position, orientation, shape, etc.

At the bottom left, the user can decide to limit the range of objects that are shown in the dialog by selecting the numbers and clicking **Show Objects**. At the upper left, click **Reset** to return the dialog to the original parameters of the listed objects.

When finished editing, click **OK** to return to the GadGeo section. There, clicking **Run** applies the modifications to the objects in the structure according to the user's choices.





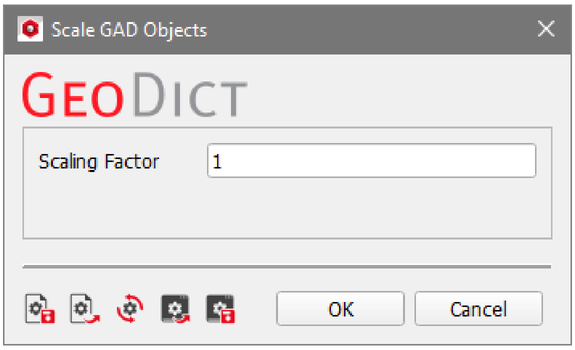


For example, observe the effect of increasing the **Diameter** of the red sphere from 20 to 40 µm in the matchstick structure created with the default settings and shown first in page 7.

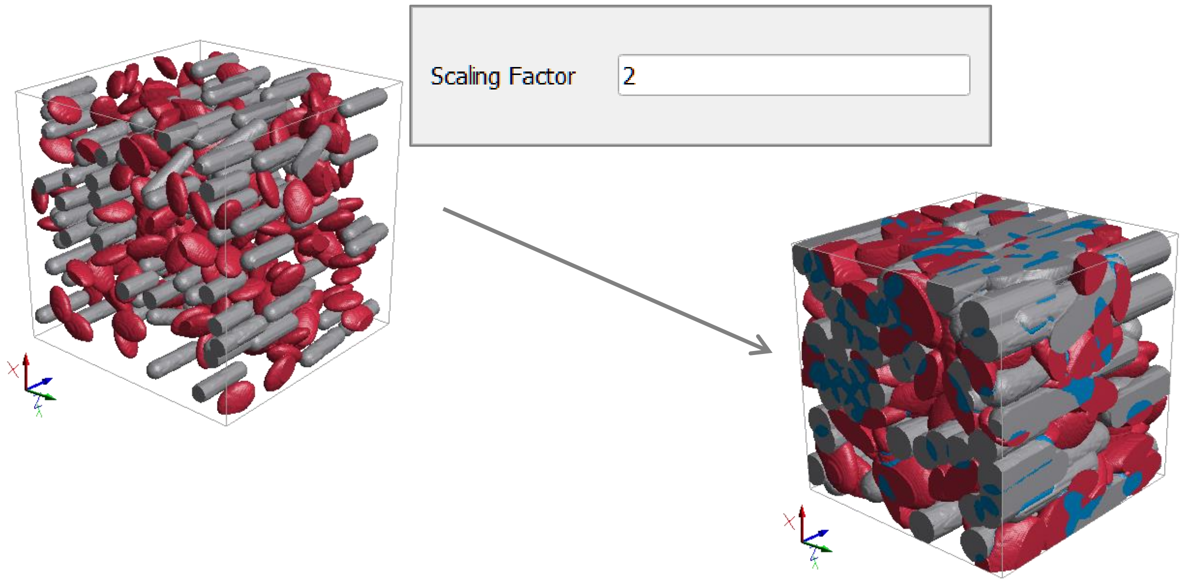
Scale

After selecting **Scale** from the pull-down menu, and clicking the **Edit...** button, the **Scale GAD Objects** dialog opens.

Enter the desired **Scaling Factor** for the objects in the structure and click **OK** to return to the **GadGeo** section.

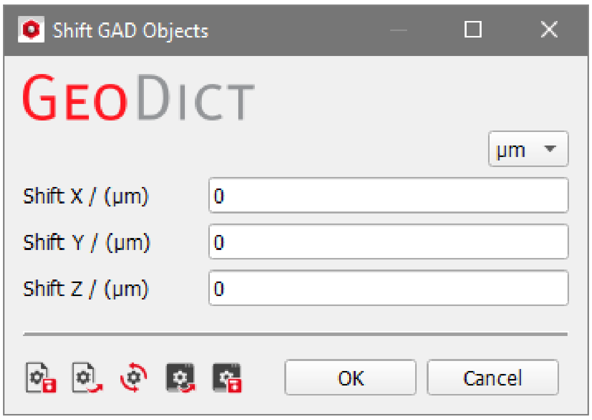


Clicking **Run** changes the size of the objects by the given scaling factor, but not the size of the structure's domain or its resolution.



SHIFT

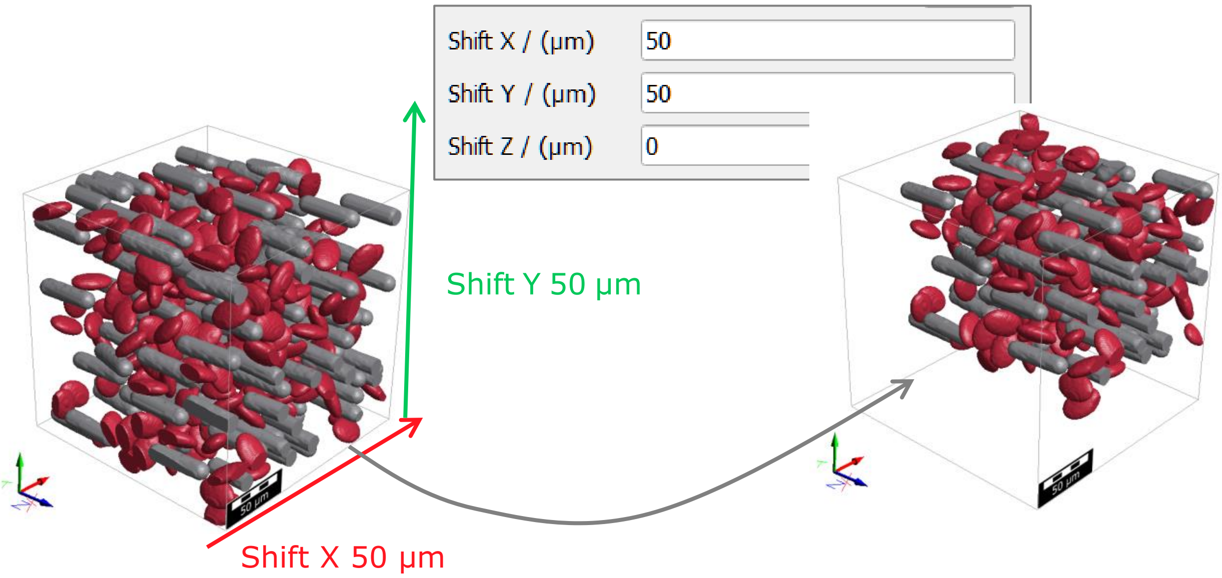
After selecting **Shift** from the pull-down menu, and clicking the **Edit...** button, the **Shift GAD Objects** dialog opens.



Enter the desired **Shift** in **X**, **Y**, and **Z** direction for the objects in the structure and click **OK** to return to the GadGeo section.

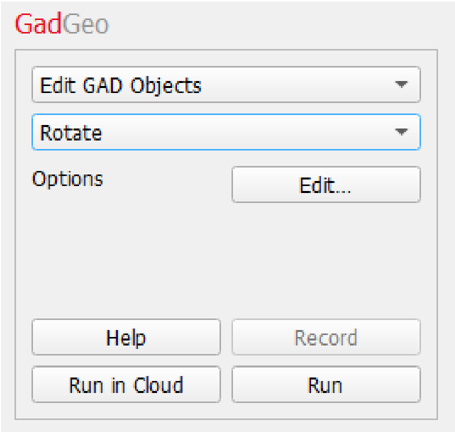
Clicking **Run** moves the objects in the structure away from the Cartesian axis origin by the given distance.

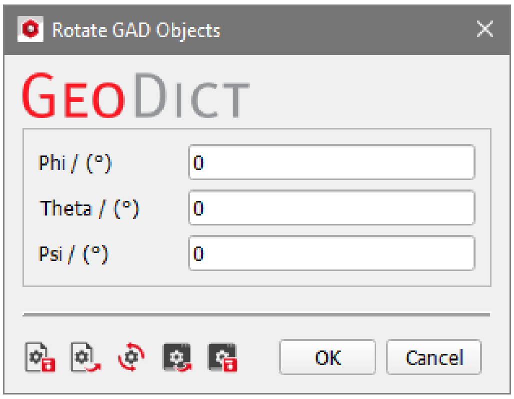
Observe the relocation of the structure away from origin 50 µm in the X-direction and 50 µm in the Y-direction. Entering negative values returns the original structure position.



ROTATE

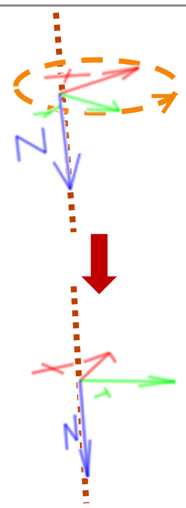
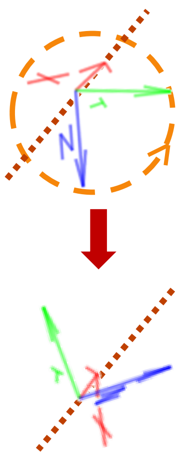
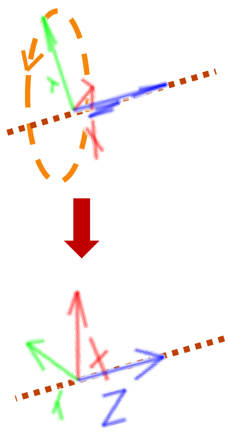
After selecting **Rotate** from the pull-down menu, and clicking the **Edit...** button, the **Rotate GAD Objects** dialog box opens.



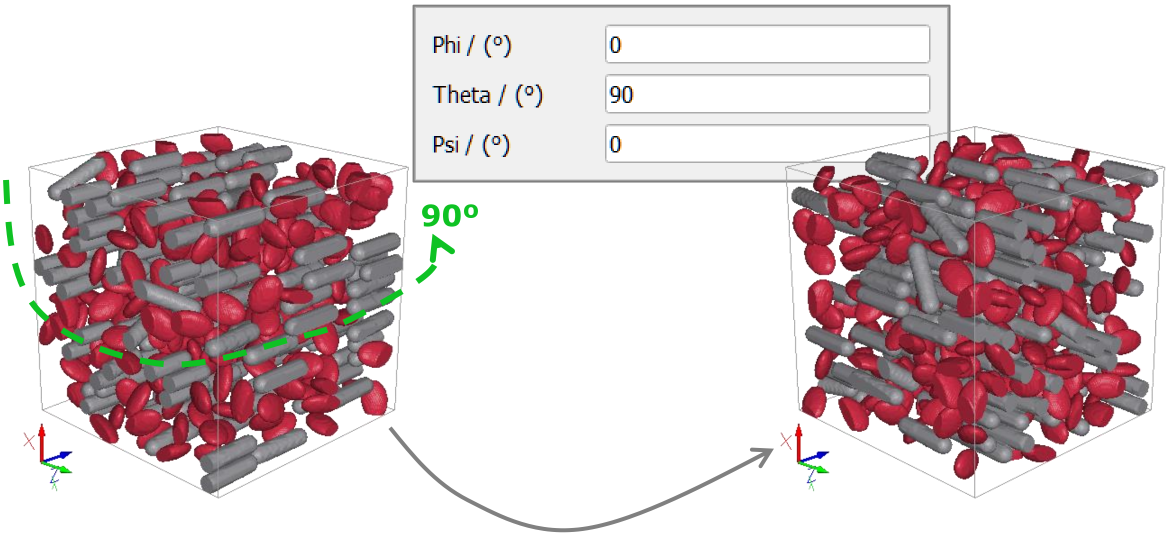


Enter the Euler angles **Psi**, **Theta**, and **Phi** by which the objects in the structure should rotate and click **OK** to return to the **GadGeo** section.

Clicking **Run** turns the objects in the structure by the given rotation angles.

z axis fixed	new x axis fixed	new z axis fixed
Phi applies rotation about existing z-axis	**Theta** applies rotation about the new x-axis	**Psi** applies rotation about the new z-axis
		

Observe the turning of the structure by 90° about the (new) X-axis (Theta). No rotation was previously applied around the (old) Z-axis (Phi).



Resolve Overlap / Mark Contacts

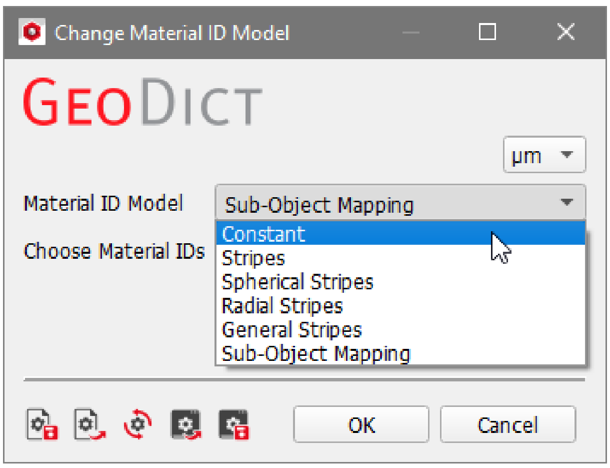
When running the **Resolve Overlap / Mark Contacts** command, GeoDict changes the overlap behavior as if **Resolve Overlap** and **Mark Contact Voxels** had been selected upon creation of the structure (see page 30 for a description of these options). This command has no parameters, GeoDict automatically chooses a free material ID as **Contact Voxel Material ID**.

The same effect can be achieved using the **Edit Domain** command (see page 48).

Change Material ID Model

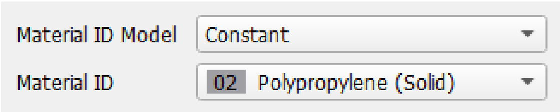
With the **Change Material ID Model** command it is possible to create objects that consist of different material IDs. This command is new in GeoDict 2022.

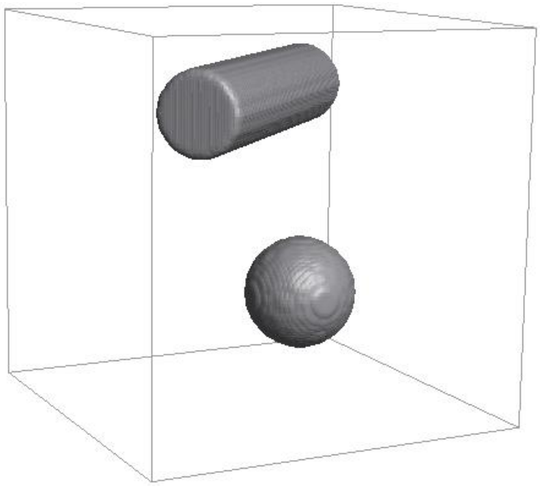
When clicking the **GadGeo Options Edit...** button, the **Change Material ID Model** dialog opens.



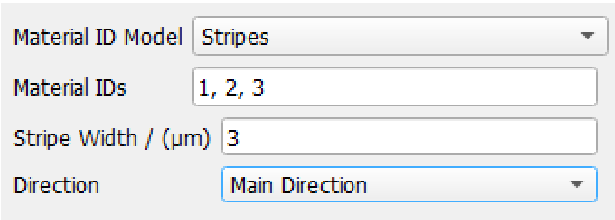
The user may choose between six different material ID models. The chosen model will be applied to all GAD objects when the **Run** button is clicked.

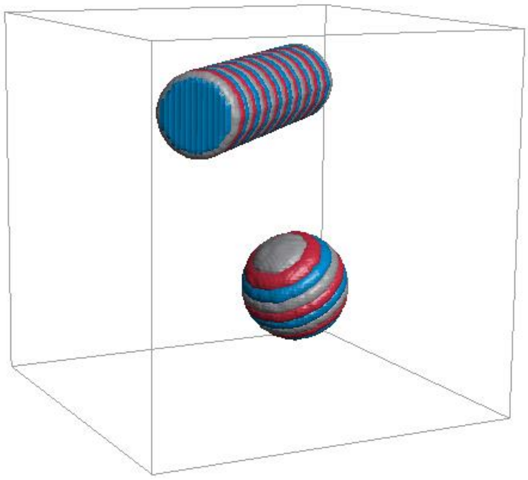
With the **Material ID Model** set at the default **Constant**, each object is assigned the single selected material ID.



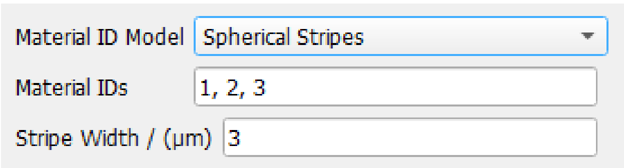


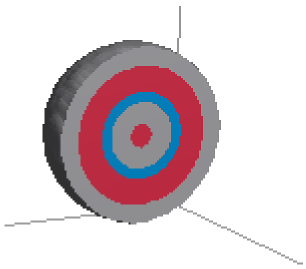
Stripes creates stripes of different materials in a direction given by one of the intrinsic directions of the object.



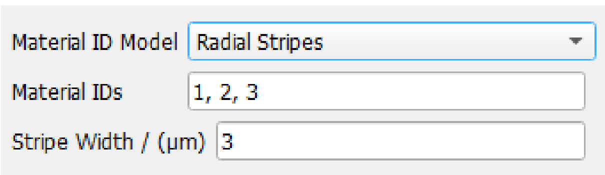


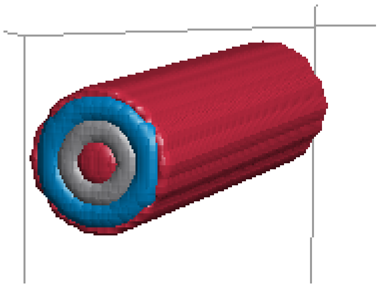
Spherical Stripes creates stripes of different material IDs around the center of the object.



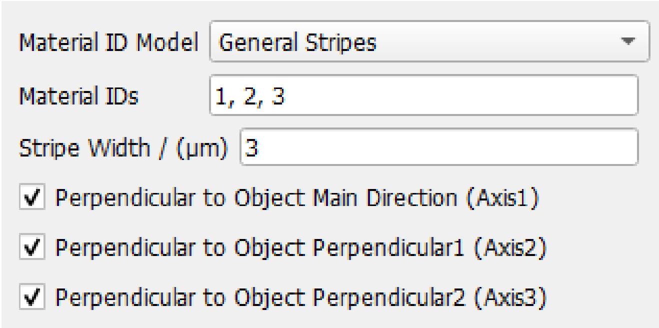


Radial Stripes creates stripes of different material IDs around the main axis of the object.

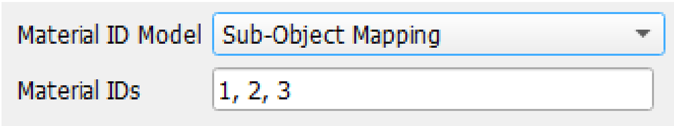




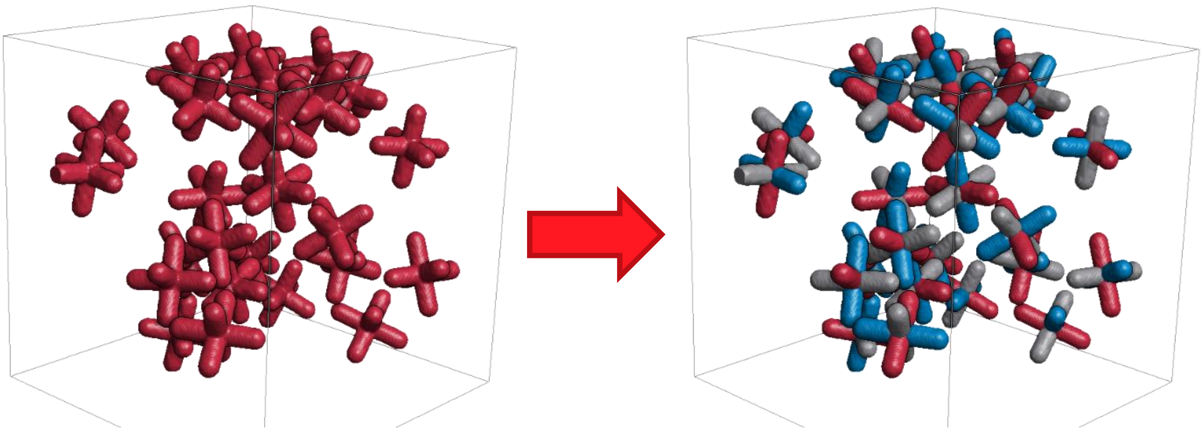
General Stripes is a generalization of the other models. For each axis, set if there should be a stripe pattern when moving along this axis. Thus, when all axes are checked on, spherical stripes will be created, if only axes 2 and 3 are checked, radial stripes are created, and if only one axis is checked, a stripe pattern along this axis will be created.



Sub-Object Mapping allows to select different material IDs for the different parts of a CombinedObject.

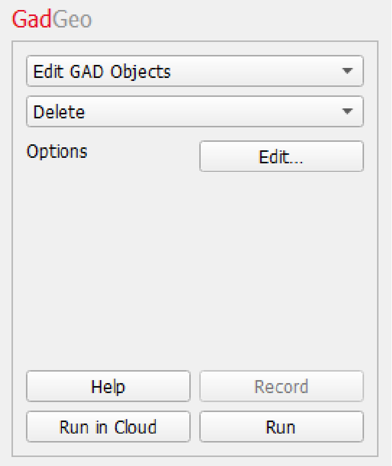


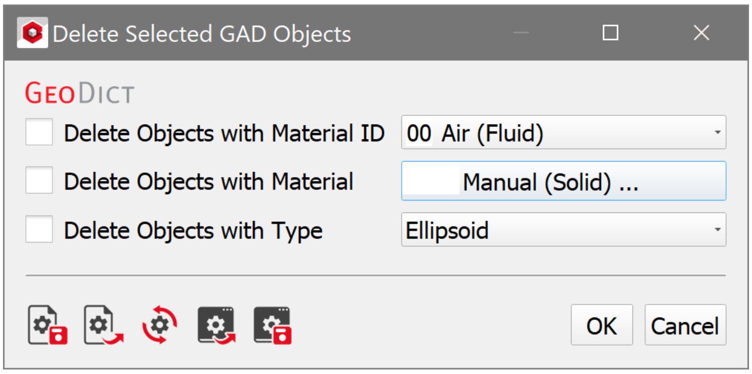
This is particularly useful for User-Defined Objects and structures created using those objects as described on page 33. For the 3D cross example that consists of three **ShortCircularFiber** GAD sub-objects, every sub-object can get a different material ID assigned.



DELETE

After selecting **Delete** from the pull-down menu, and clicking the **Edit...** button, the **Delete Selected GAD Objects** dialog opens.

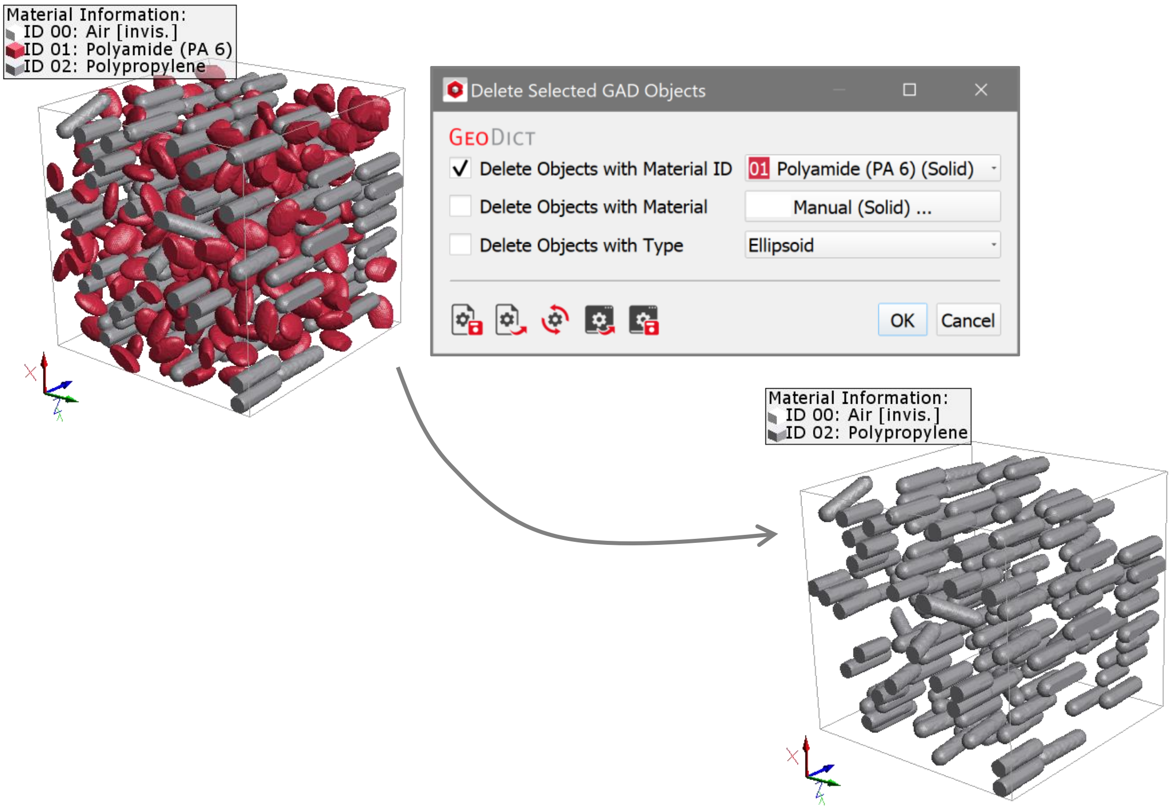




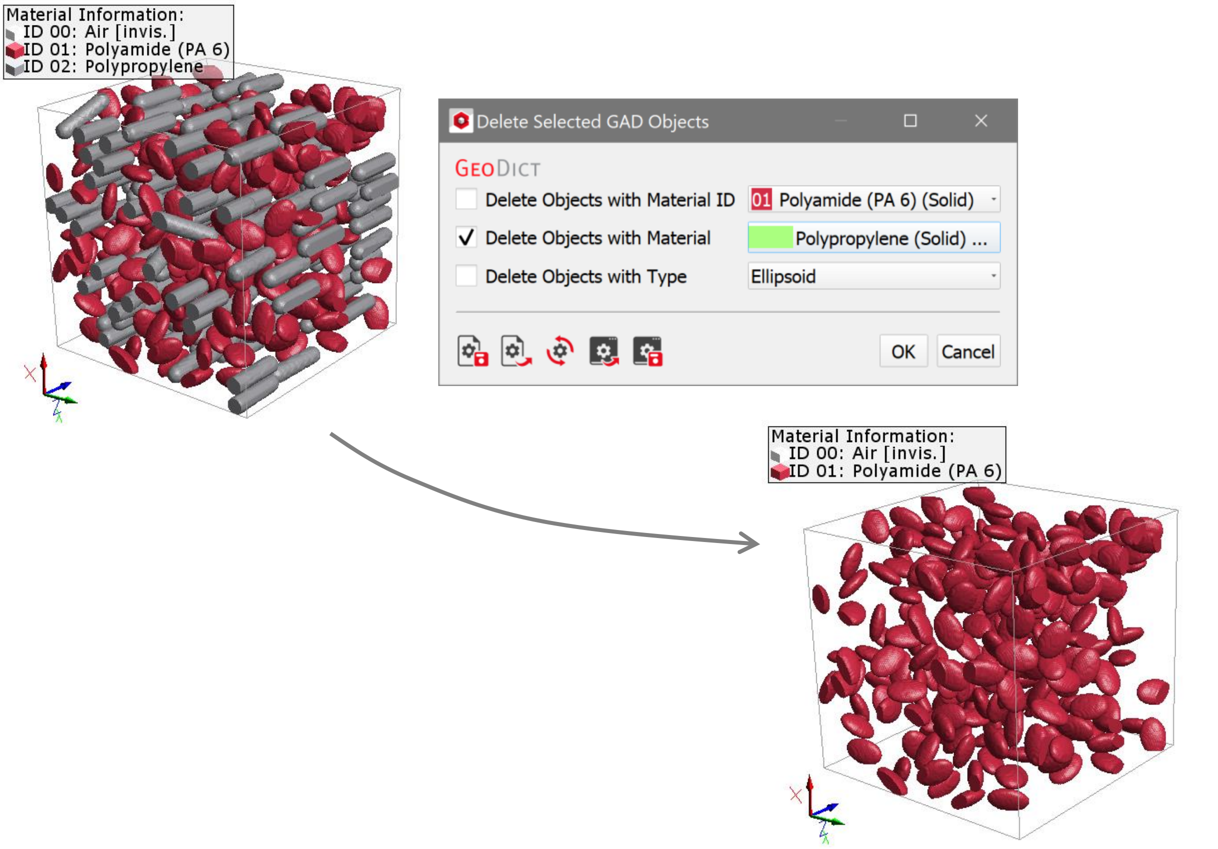
Decide whether to delete objects with a certain material ID, a certain material, or a certain object type. Then select the material ID, the material, or the object type from the pull-down menus. Click **OK** to return to the GadGeo section.

Click **Run** to delete the objects in the structure according to your choice. It is possible to check multiple of the possibilities.

Observe the disappearance of the objects with material ID 01 (here shown in red) in the structure.



Observe the disappearance of the objects with material Polypropylene (here the fibers) in the structure, leaving only objects made of Polyamide.



Edit Domain

In GeoDict, the GAD objects are converted into a voxel geometry. The details of this conversion are defined by the **Domain**. The domain determines the resolution, the size and position of the bounding box, the background materials, and the overlap behavior of objects.

Whenever a structure is created or loaded in GeoDict, the corresponding domain parameters are stored. They can be accessed and changed through the **Edit Domain** command.

In a GAD file, those parameters are defined in the **Domain** section.

GeoDict knows and uses three different ways to specify the domain size, defined by the value after the key **DomainMode**:

- **DomainMode VoxelNumber:** The voxel length, specified by the value after the key **VoxelLength**, and the number of voxels in the three spatial directions, specified by the values after **NX**, **NY** and **NZ**, are given.
- **DomainMode Length:** The voxel length specified by the value after the key **VoxelLength**, and the lengths of the domain in the three spatial directions, specified by the values after the keys **LengthX**, **LengthY** and **LengthZ**, are given. The number of voxels (**NX**, **NY** and **NZ**) is calculated automatically.

```
"Domain" : {
  "PeriodicX"             : False,
  "PeriodicY"             : False,
  "PeriodicZ"             : False,
  "OriginX"               : [0, "m"],
  "OriginY"               : [0, "m"],
  "OriginZ"               : [0, "m"],
  "VoxelLength"           : [1e-06, "m"],
  "DomainMode"            : "VoxelNumber",
  "NX"                    : 100,
  "NY"                    : 100,
  "NZ"                    : 100,
  "OverlapMode"           : "GivenMaterial",
  "Material" : {
    [...]
  },
  "OverlapMaterialID"     : 3,
  "NumOverlapRules"       : 0,
  "HollowMaterialID"      : 0,
  "PostProcessing" : {
    "ResolveOverlap"      : False,
    "MarkContactVoxels"   : False,
    "ContactMaterialID"   : 15
  }
},
```

- **DomainMode VoxelNumberAndLength:** This mode is similar to **DomainMode VoxelNumber**, but additionally the physical lengths of the domain are specified by the values after the keys **LengthX**, **LengthY** and **LengthZ**. This mode is used when wanting to save a difference of the physical lengths (**LengthX**, **LengthY**, and **LengthZ**) to the lengths given by the voxel length and the number of voxels (**VoxelLength**, **NX**, **NY**, and **NZ**).

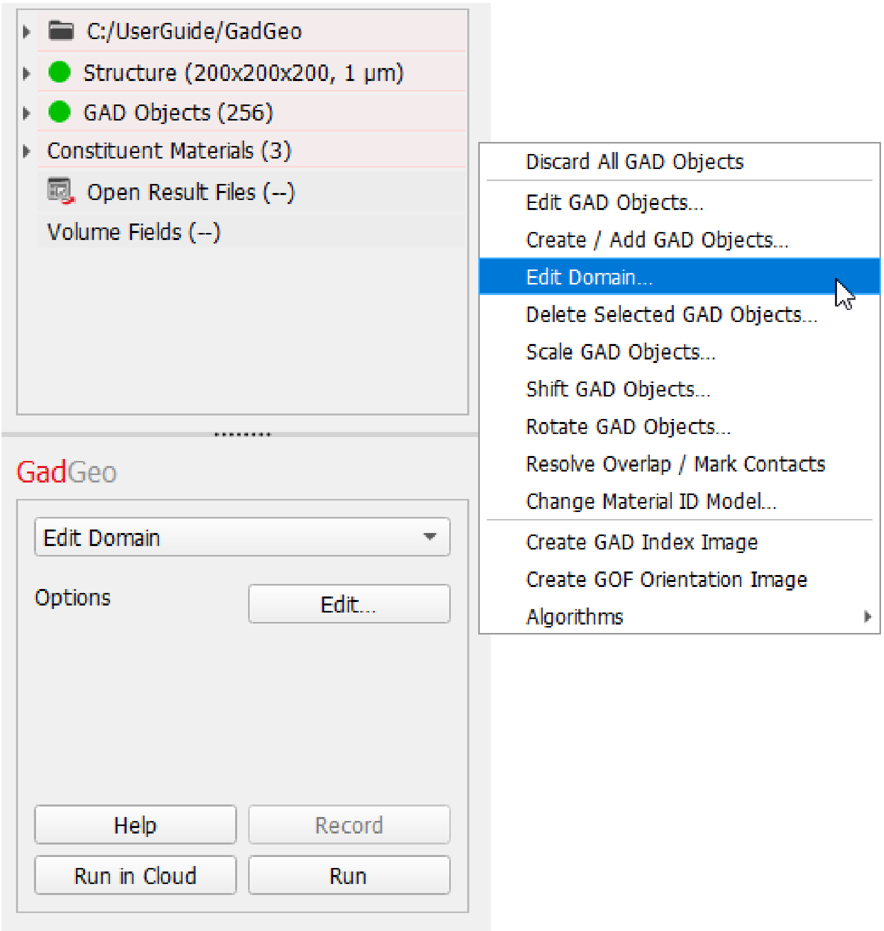
Additional to the size of the domain, the origin of the geometry is specified. A shift of the origin from (0.0, 0.0, 0.0) is given by the values after the keys **OriginX**, **OriginY**, and **OriginZ**.

The domain of the voxel geometry may be periodic in one or more of the three spatial dimensions. This is specified by the values after the keys **PeriodicX**, **PeriodicY** and **PeriodicZ**, which may be **False** or **True**. If the domain is periodic in the X-directions, this means that an object leaving the domain through one of the two domain boundaries in X-direction reenters the domain from the other boundary in X-direction. Periodicity in Y- and Z-direction is defined accordingly.

With **Edit Domain**, the position and resolution of the domain can be changed. Changing the domain leaves the absolute size and position of the objects themselves unchanged, but it might change which part of an object is shown inside of the voxel grid, and how well it is resolved. Basically, **Edit Domain** allows to do the following tasks:

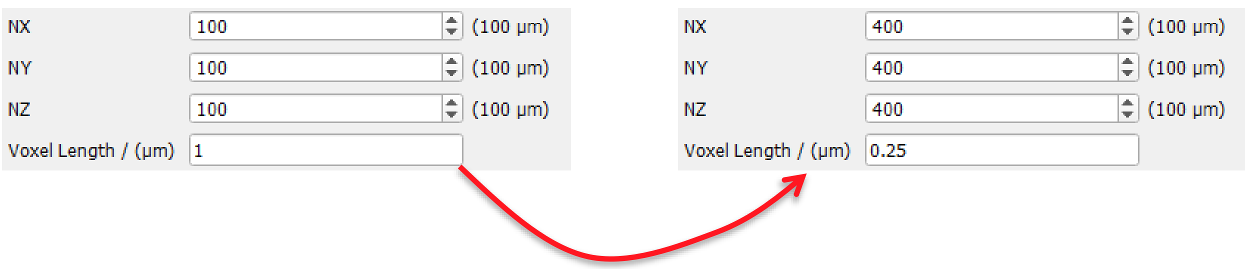
1. Changing the resolution.
2. Shifting the field of view.
3. Distinguishing background and empty parts of hollow objects.
4. Redefine overlap behavior

Alternatively, the command is also available from the **GAD Objects** context menu as **Edit Domain...**



Change Resolution

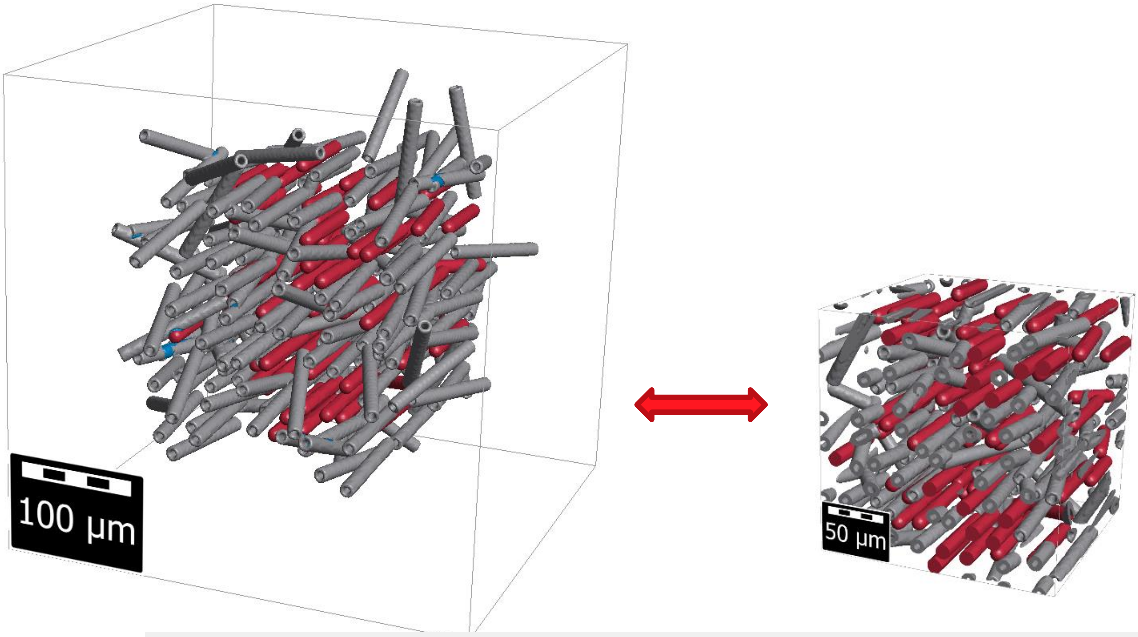
Edit Domain can be used to change the resolution by simultaneously dividing the **Voxel Length** and multiplying the number of voxels in **NX, NY, NZ** direction by the same factor.



Shift the Field of View

By changing the values of **NX, NY, NZ** and the position of the **Origin** it is possible to move the field of view, thus changing the cut-out used as the current 3D structure model. The **Origin** defines the position of the front lower left corner of the bounding box, which is usually (0,0,0) by default.

NX	400	(400 µm)	Origin X / (µm)	-100	Periodic X
NY	400	(400 µm)	Origin Y / (µm)	-100	Periodic Y
NZ	400	(400 µm)	Origin Z / (µm)	-100	Periodic Z
Voxel Length / (µm)	1		Pore / Matrix Material (ID 00)	Air (Fluid) ...	Center Domain



NX	200	(200 µm)	Origin X / (µm)	0	Periodic X
NY	200	(200 µm)	Origin Y / (µm)	0	Periodic Y
NZ	200	(200 µm)	Origin Z / (µm)	0	Periodic Z
Voxel Length / (µm)	1		Pore / Matrix Material (ID 00)	Air (Fluid) ...	Center Domain

Checking the **Periodic X**, **Periodic Y**, and **Periodic Z** boxes applies periodicity in one or several directions. Checking them has the effect that the objects ending on one side of the volume reappear on the opposite side.

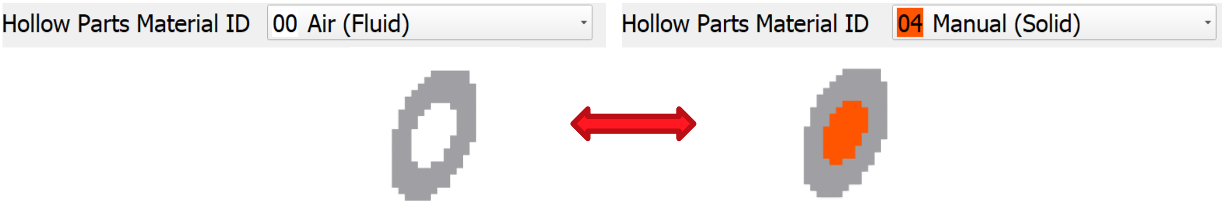
When clicking **Center Domain**, the domain will be automatically centered around the coordinate origin (0,0,0).

Distinguish between Background and Hollow Parts

It is possible to reassign the **Pore / Matrix Material (ID 00)** as explained above in pages 30.

Pore / Matrix Material (ID 00)	Air (Fluid) ...

If some objects of the structure contain hollow parts, e.g. hollow fibers, hollow spheres, or cellulose fibers, it is also possible to change the material ID of those hollow parts, which is by default the same material as the background material.

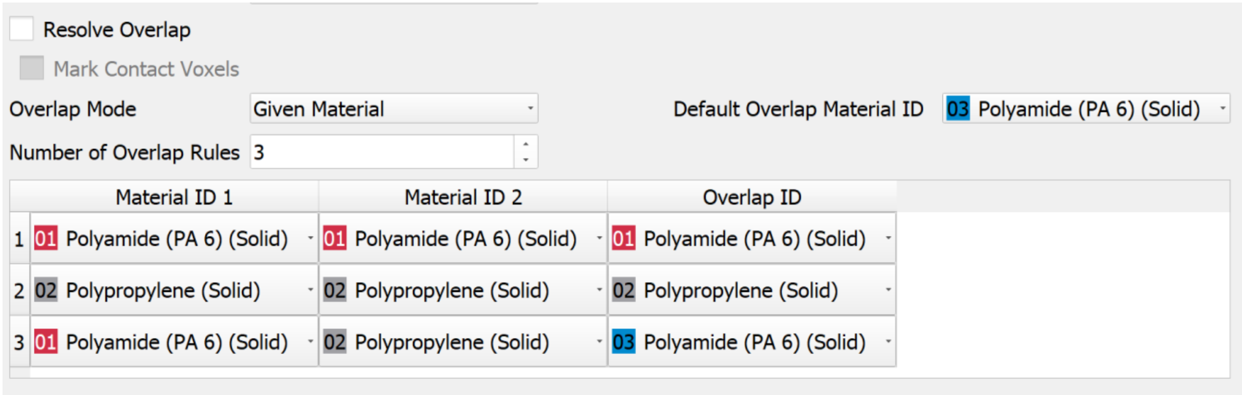


Changing the **Hollow Parts Material ID** allows to distinguish between the interior of those objects and the pore space on the outside.

In a GAD file, a change of these parameters will change the parameters stored in the **Material** section and the **HollowMaterialID**.

Redefine Overlap Behavior

When the 3D structure model is created with one of the structure generation modules of GeoDict, the user has to determine how overlapping objects should be treated. With GadGeo **Edit Domain**, the chosen settings can be changed afterwards, and other modes for the overlap can be set without changing the position of the objects themselves.



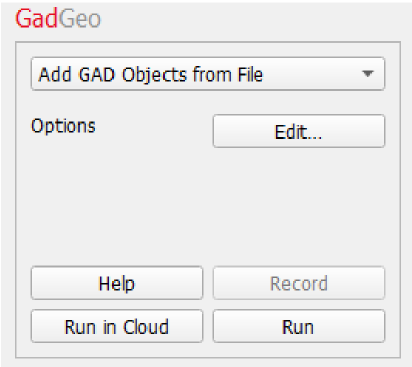
The **Edit Domain** dialog allows to change the settings for **Resolve Overlap** and the **Overlap Mode** and allows to choose all the overlap modes that are accessible when creating a GAD structure. The modes are explained in detail in pages 30ff above.

All defined overlap rules are also stored in the GAD file, in sections **OverlapRule...**. Therefore, these settings are always stored together with the structure, when it is saved either as a .gad or .gdt file.

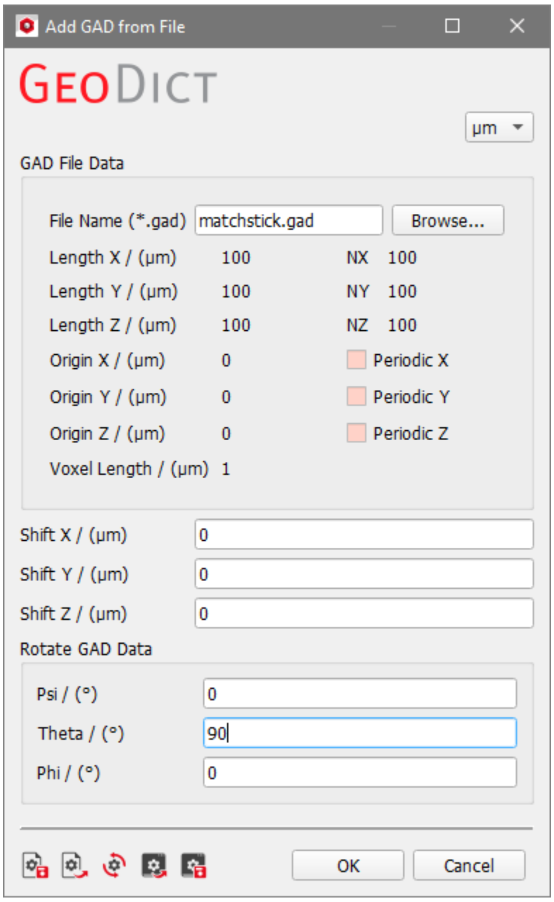
```
"OverlapMode"                  : "GivenMaterial",
"OverlapMaterialID"            : 3,
"NumOverlapRules"              : 3,
"OverlapRule1" : {
  "MaterialID1"                : 1,
  "MaterialID2"                : 1,
  "OverlapMaterialID           : 1
},
"OverlapRule2" : {
  "MaterialID1"                : 2,
  "MaterialID2"                : 2,
  "OverlapMaterialID"          : 2
},
"OverlapRule3" : {
  "MaterialID1"                : 1,
  "MaterialID2"                : 2,
  "OverlapMaterialID"          : 3
},
"PostProcessing" : {
  "ResolveOverlap"             : False,
  "MarkContactVoxels"          : False,
  "ContactMaterialID"          : 15
}
```

Add GAD Objects from File

After selecting **Add GAD Objects from File** from the pull-down menu in the GadGeo section, the current structure can be combined with objects stored in another GAD file. This command is not available from the **GAD Objects**' context menu.



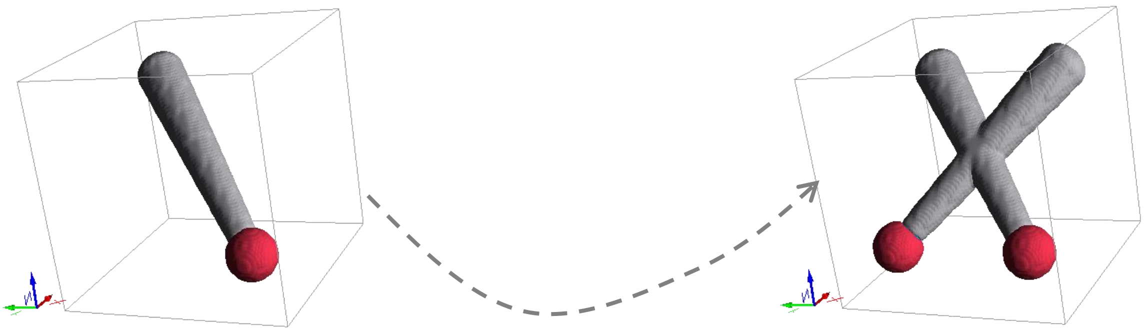
Click the **Options**' **Edit...** button to open the **Add GAD from File** dialog box.



Then, click the **Browse...** button to search for a file in GAD format to add.

Before adding the new objects from the GAD file to the current structure, the new objects may be shifted and rotated with respect to their position stored inside the GAD file.

For example, the matchstick structure in page 7 could be rotated by 90° and added to itself. As shown in the dialog box on the right, simply **Browse...** for the GAD file and enter a **Theta** value of 90°.



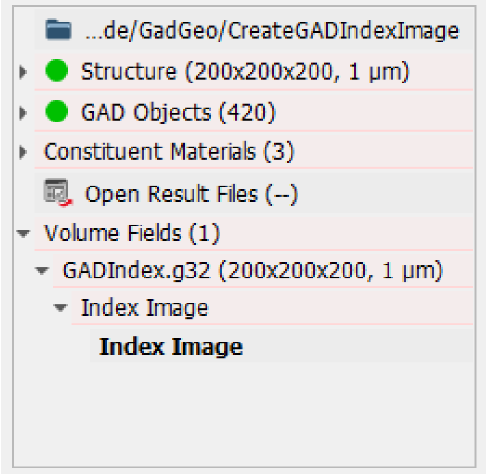
Create GAD Index Image

In GeoDict, each GAD object has an index number. This numbers can be accessed with the help of the **Create GAD Index Image** command. This command creates a 3D volume field, where each voxel contains a 32 bit unsigned integer value describing the index of the object that this voxel belongs to.

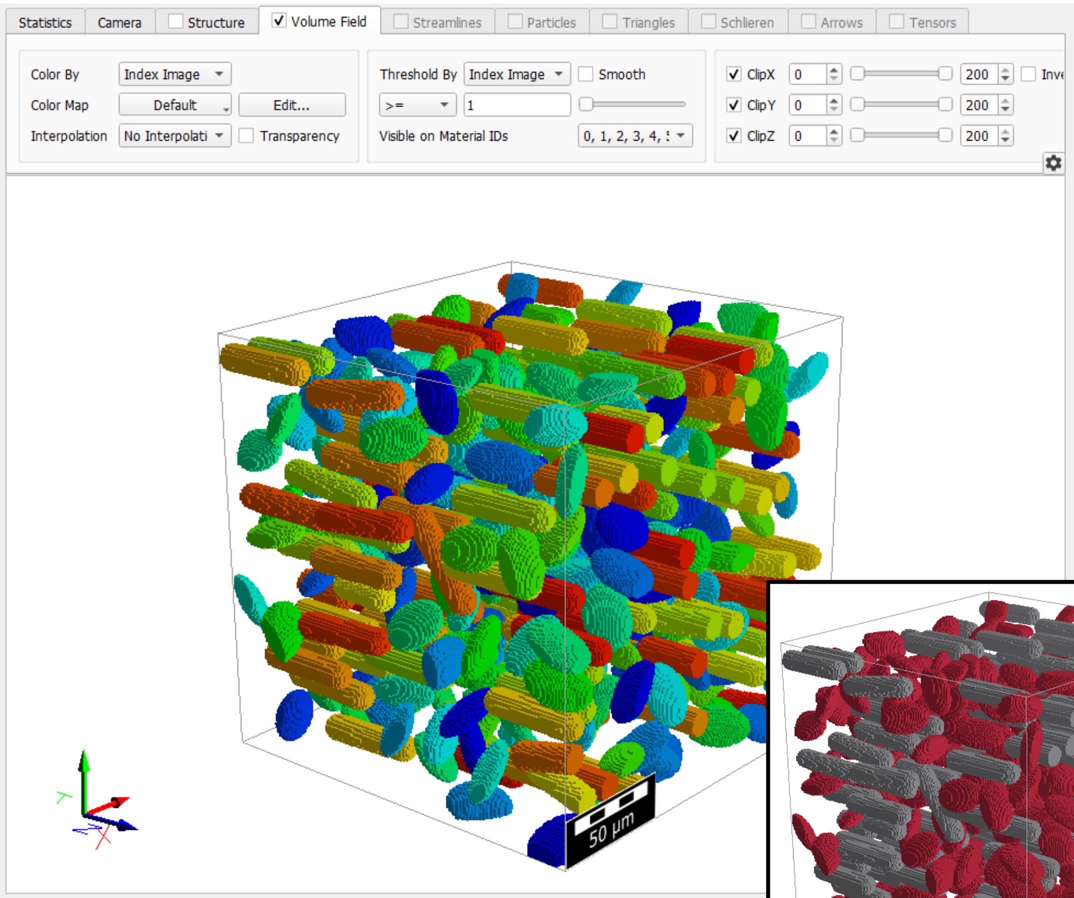
This command requires no input parameters, so there are no parameters to be set.

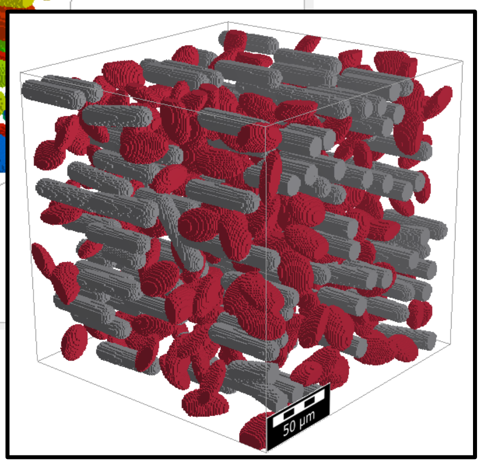
When the command is run, no (result) files are created. The created index image is available in GeoDict as a volume field named GADIndex.g32.

By a right click on the GADIndex.g32 line, the context menu allows to save the index image as .g32 file.



An index image can be visualized as other volume fields, the screenshot below shows the index number field with small index numbers in blue and high index numbers in red.





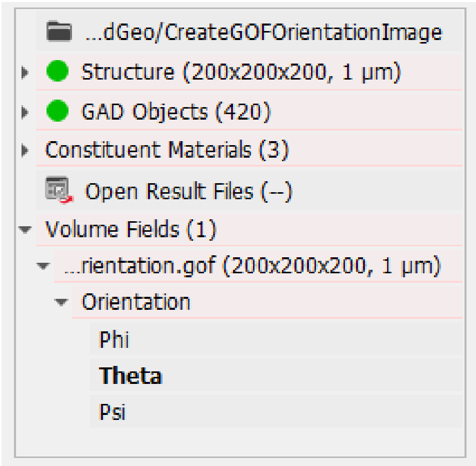
Create GOF Orientation Image

Every GAD object has an intrinsic orientation, which is defined through its principal direction (key Axis1), a first perpendicular direction (key Axis2) and a second perpendicular direction forming a RHS coordinate system.

This orientation can be described in form of the Euler angles (see page 42) required to rotate the cartesian coordinate system into the intrinsic one. Thus, for an intrinsic coordinate system described through Axis1 = (1,0,0) and Axis2 = (0,1,0), all Euler angles would be zero.

The local orientation of a voxel is needed for some simulations (e.g. in ElastoDict) and can be stored in a 3D field that contains the three Euler angles for each voxel.

The **Create GOF Orientation Image** command creates this volume field. This command requires no input parameters, so there are no parameters to be set.

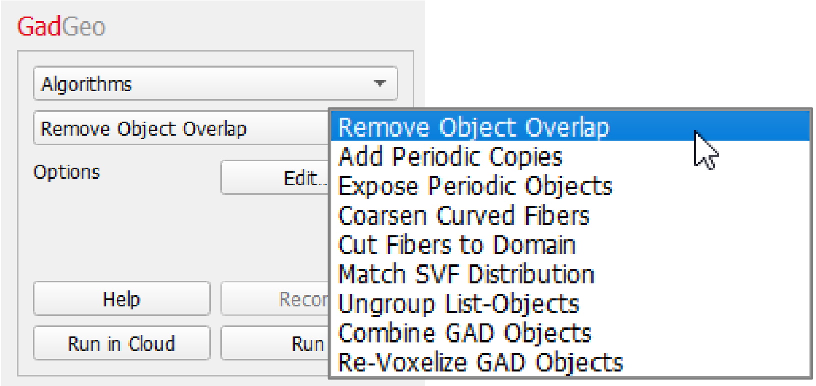


When the command is run, no (result) files are created. The created field is available in GeoDict as a volume field named GADOrientation.gof.

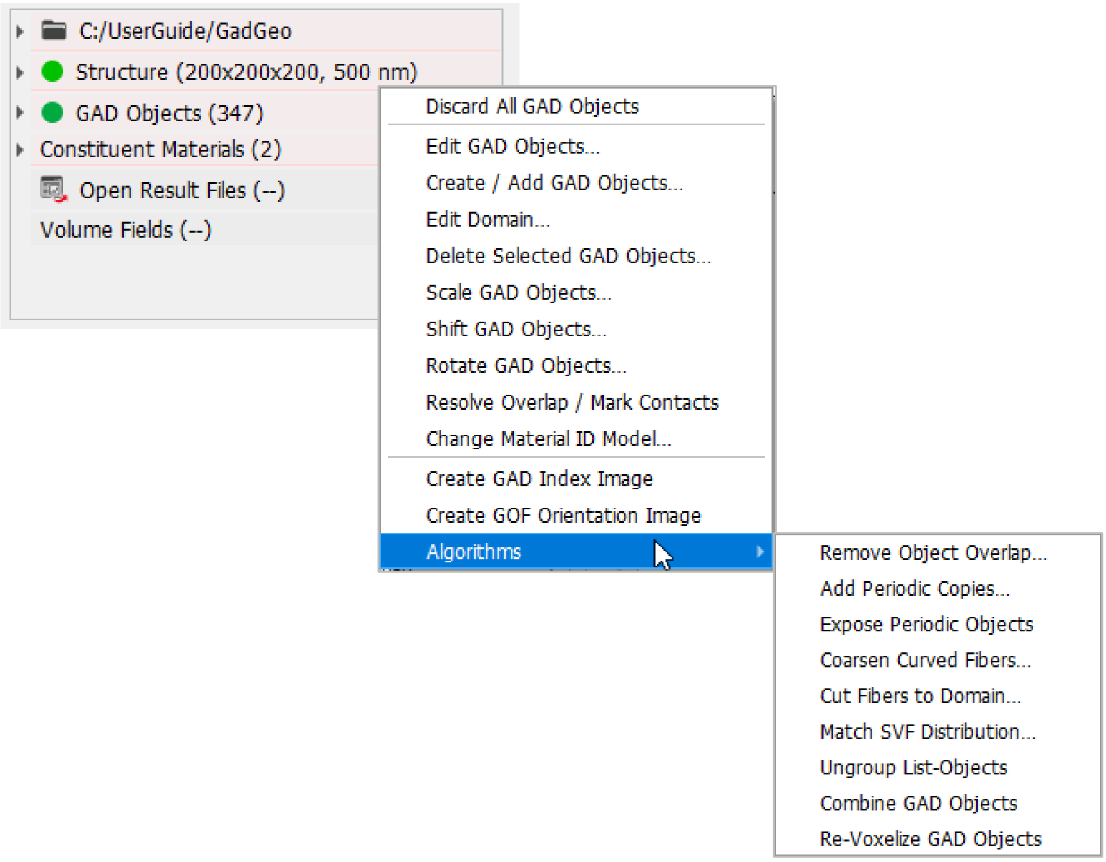
By a right click on the GADOrientation.gof line, the context menu allows to save the index image as .gof file.

ALGORITHMS

Nine different commands are selectable in the pull-down menu under **Algorithms.**



All of them are also accessible from the **Objects** context menu.



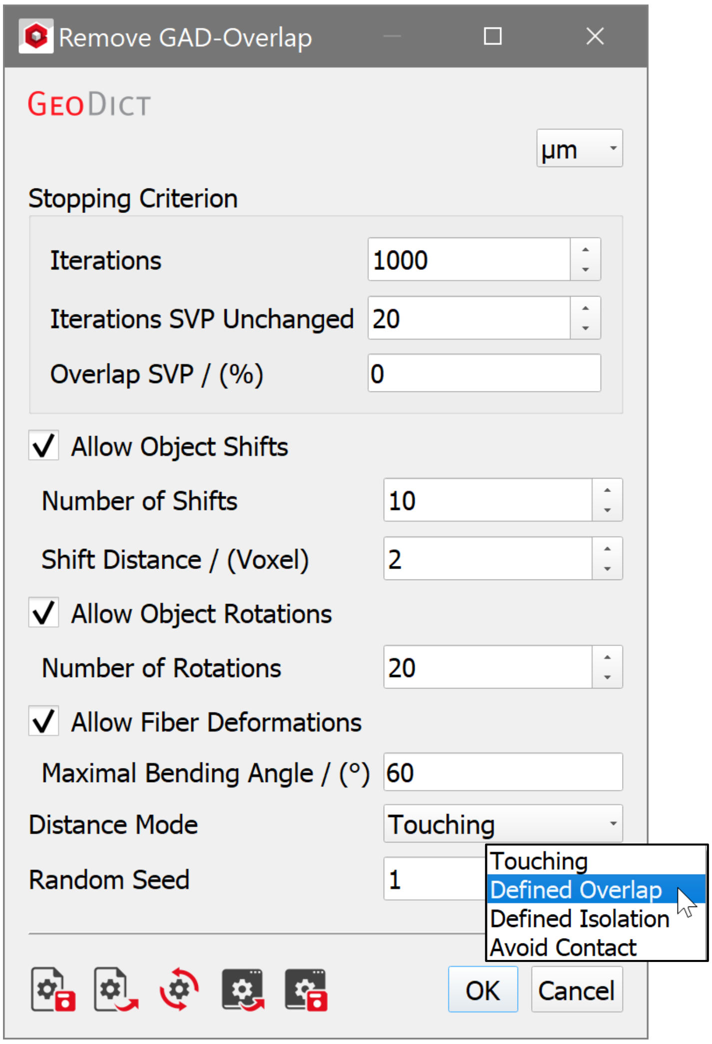
REMOVE OBJECT OVERLAP

The options of **Remove Object Overlap** are similar to choices available under the **Generation and Overlap Mode** panel of FiberGeo for fibrous structures (FiberGeo Create Options dialog box) and GrainGeo for structures with granular objects (GrainGeo Create Grains Options dialog box)

By means of **Remove Object Overlap**, objects can be detached from each other in 3D structures after they have been generated.

Removing overlap in dense structures is a complex process that requires some time, depending on the imposed restrictions. In the **Stopping Criterion** panel, select the maximal number of **Iterations** for the algorithm to finish, how many iterations should be allowed when the solid volume fraction does not change anymore (**Iterations SVP Unchanged**) and the maximal overlap permitted for the process to finish (**Overlap SVP / (%)**).

The overlap of objects is removed by shifting, rotating, and/or deforming the objects.



Choose which processes should be carried out and the way in which they should be done.

After checking **Allow Object Shifts**, choose the allowed **Number of Shifts** and the magnitude of the shifts (**Shift Distance**) in voxels.

In a similar way, select to **Allow Object Rotations** and the **Number of Rotations**, or to **Allow Fiber Deformations**, and then the **Maximal Bending Angle** for the deformations.

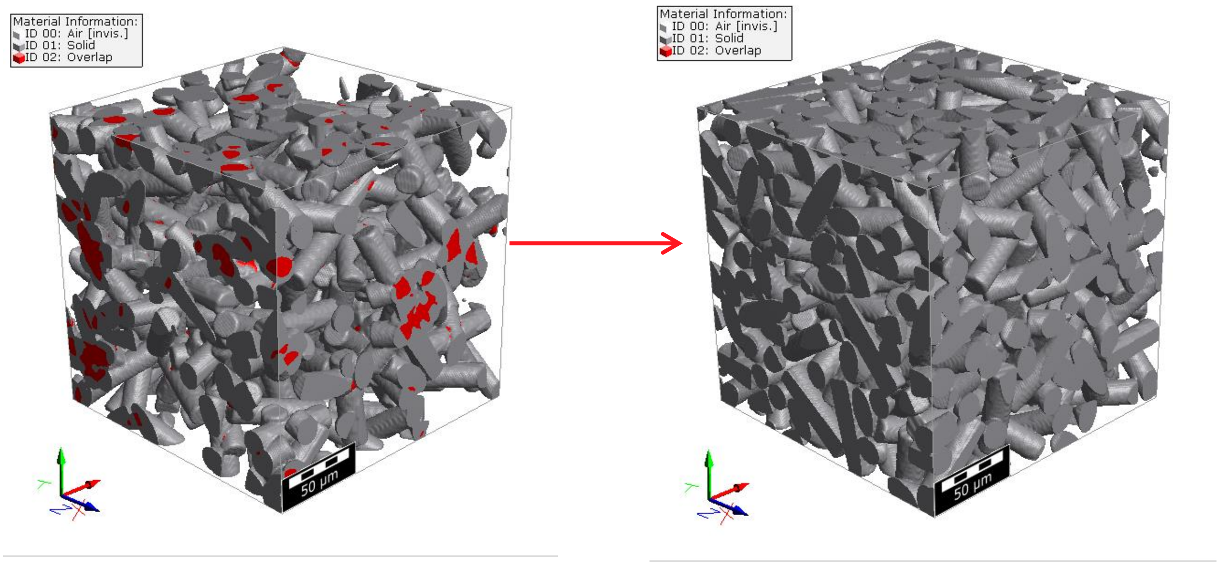
Choose the Distance Mode between **Touching**, **Defined Overlap**, **Defined Isolation** and **Avoid Contact** to set the separation that should be achieved between objects. For Defined Overlap, set the **Distance** that objects are still allowed to overlap. For Defined Isolation, set the minimal **Distance** required between two objects.

Overlap and distances between objects are determined based on the voxel representation of the GAD objects, not between the analytic objects themselves. Therefore, the outcome is resolution dependent, and objects not overlapping in a certain resolution might slightly overlap in a different voxel grid.

Various realizations of the overlap removal with the specified options can be obtained by changing the **Random Seed**. If all other settings are equal, overlap removal operations with the same **Random Seed** value produce exactly the same structure.

For example, overlap of fibers can be removed from a fibrous structure with SVP of 35%, using the default values in the **Remove GAD-Overlap** dialog box. The red areas correspond to the overlap between the grey fibers.

Observe how the shape and the position of the fibers have been modified to avoid the overlap.

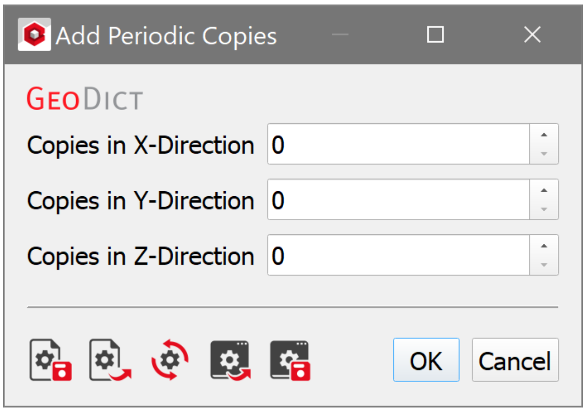


When the overlap has not been able to be completely removed after the stopping criterions have been applied, a message appears indicating this and showing the percentage of remaining overlap.

Add Periodic Copies

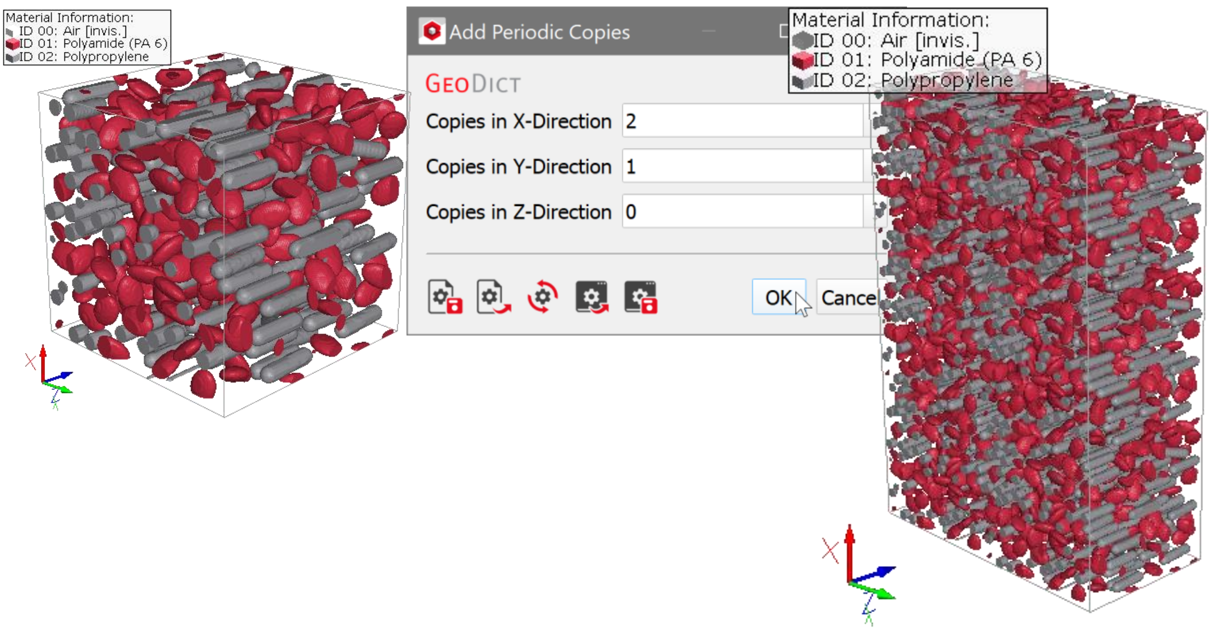
The algorithm to **Add Periodic Copies** allows to periodically repeat the structure model in the chosen direction.

In contrast to the **Process-Repeat**, command available in the ProcessGeo module, the resulting structure still contains analytic data and can be saved in GAD format.



To add periodic copies, the original structure model should be periodic in the direction(s) to be copied. Otherwise, the resulting structure may show unwanted effects.

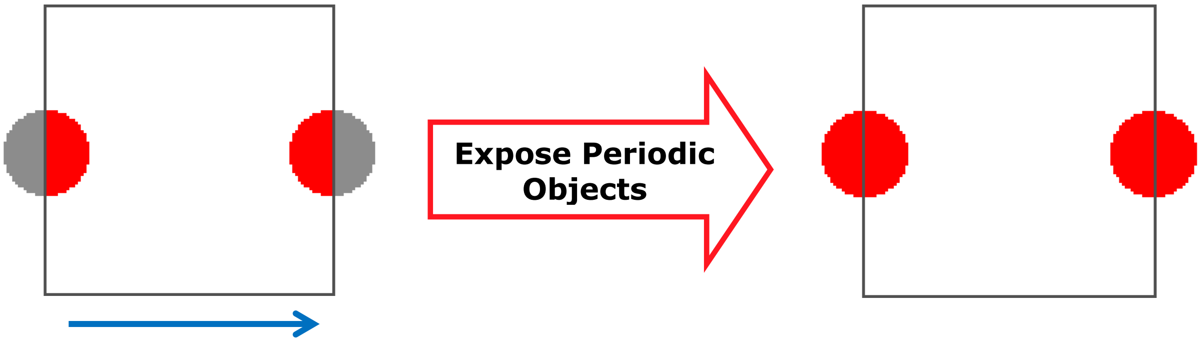
Here, periodic copies of the structure are added in the X-direction (2 copies) and the Y-direction (1 copy).



Expose Periodic objects

When a structure is created, e.g. in GrainGeo, using periodic boundary conditions, objects cut on one side of the domain reappear on the opposite side. This means that a single GAD object appears cut into several small pieces in the 3D structure. If the GAD objects are exported into other software, e.g. using ExportGeo-CAD, those periodic reappearances may not be taken into account.

Therefore, for periodic GeoDict-generated structures, **Expose Periodic Objects** produces non-periodic structures in which the interrupted periodic objects in the structure are added to the structure's analytic description (GAD).



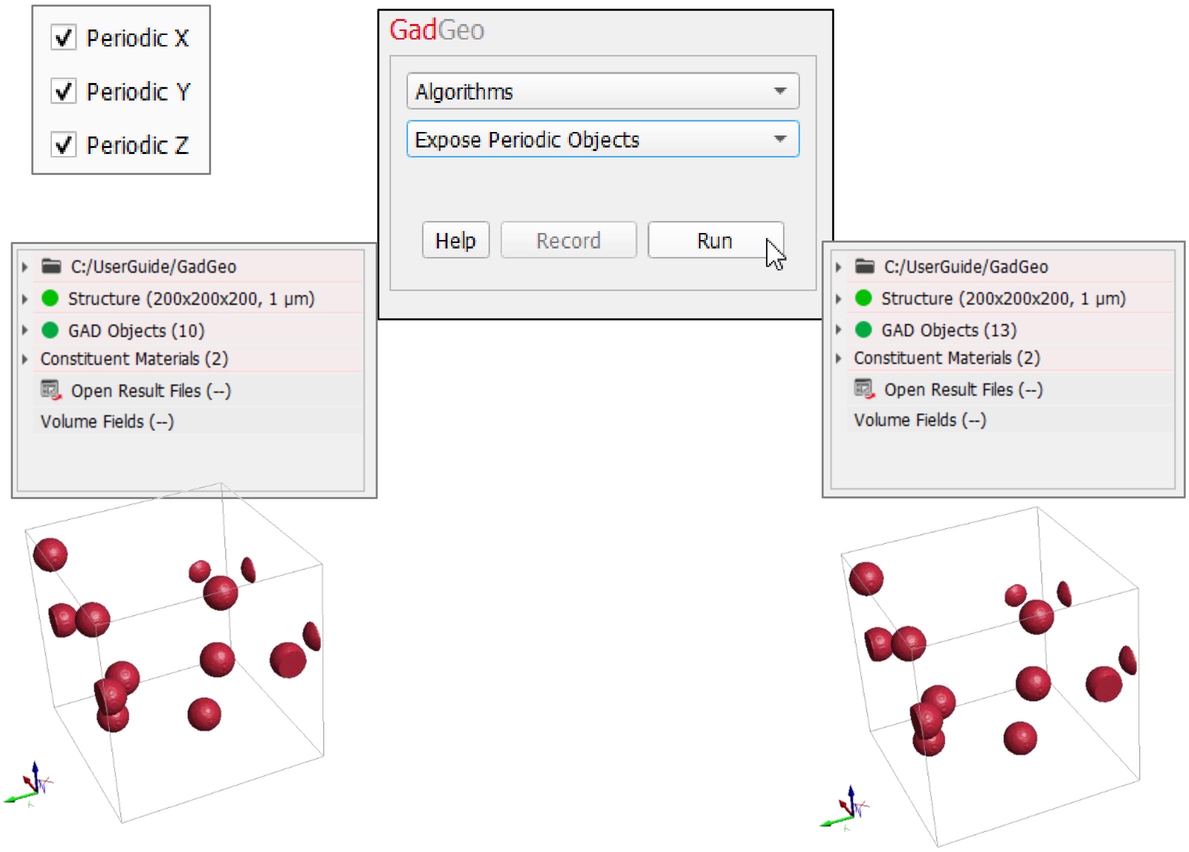
Direction of periodicity

1 Object: 1 Sphere
(made of 2 half spheres)

No periodicity

2 Objects: 2 spheres

Although no difference is observable when simply visualizing the structure, the result of exposing periodic objects is the loss of periodicity and a higher number of objects in the structure. The higher number of objects after performing this process is directly observable in the number of **Objects** in the Project Status section.



The effect is also clearly observable when saving the structure as GAD file before and after exposing periodic objects and opening them with a text editor. See how PeriodicX, PeriodicY, and PeriodicZ are now listed as *false* and the number of objects has increased from 10 to 13.

10spheresPeriodic.gad | 10spheresPeriodicExposed.gad

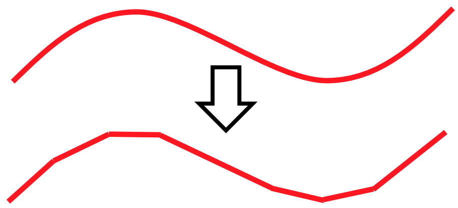
```
{
  "Header" : {
    "Release"          : 2023,
    "Revision"         : 59085,
    "BuildDate"        : "29 Sep 2022",
    "CreationDate"     : "11 Oct 2022",
    "CreationTime"     : "17:20:40",
    "Creator"          : "becker",
    "Platform"         : "64 bit Windows"
  },
  "NumberOfObjects"                       : 10,
  "Description"                           : "10spheresPeriodic",
  "Domain" : {
    "PeriodicX"                           : True,
    "PeriodicY"                           : True,
    "PeriodicZ"                           : True,
    "OriginX"                             : [0, "m"],
    "OriginY"                             : [0, "m"],
    "OriginZ"                             : [0, "m"],
    "VoxelLength"                         : [1e-06, "m"],
    "DomainMode"                          : "VoxelNumber",
    "NX"                                  : 200,
    "NY"                                  : 200,
    "NZ"                                  : 200,
    "OverlapMode"                         : "OverlapMaterial", # Possible v
    "Material" : {
      "Type"                              : "Fluid"
```

10spheresPeriodic.gad | 10spheresPeriodicExposed.gad

```
{
  "Header" : {
    "Release"          : 2023,
    "Revision"         : 59085,
    "BuildDate"        : "29 Sep 2022",
    "CreationDate"     : "11 Oct 2022",
    "CreationTime"     : "17:20:53",
    "Creator"          : "becker",
    "Platform"         : "64 bit Windows"
  },
  "NumberOfObjects"                       : 13,
  "Description"                           : "10spheresPeriodic",
  "Domain" : {
    "PeriodicX"                           : False,
    "PeriodicY"                           : False,
    "PeriodicZ"                           : False,
    "OriginX"                             : [0, "m"],
    "OriginY"                             : [0, "m"],
    "OriginZ"                             : [0, "m"],
    "VoxelLength"                         : [1e-06, "m"],
    "DomainMode"                          : "VoxelNumber",
    "NX"                                  : 200,
    "NY"                                  : 200,
    "NZ"                                  : 200,
    "OverlapMode"                         : "OverlapMaterial", # Possible v
    "Material" : {
      "Type"                              : "Fluid"
```

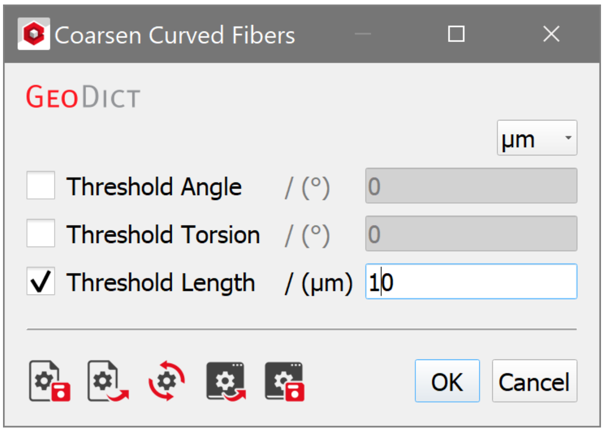
Coarsen curved fibers

Coarsen Curved Fibers allows decreasing the number of segments used in curved fibers, and thus, reduce the size of the corresponding structure files in GAD format. This also reduces the file size of STL files exported with ExportGeo-CAD or MeshGeo.



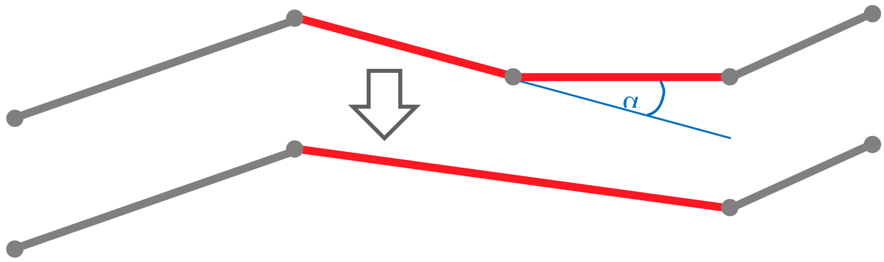
Clicking the **Options' Edit...** button opens the **Coarsen Curved Fibers** dialog where criteria for merging two segments into one can be entered.

The available **units** (m, mm, nm, and µm) are selectable from the top right pull-down menu. When the **Threshold Angle**, **Threshold Torsion**, and/or **Threshold Length** boxes are checked, values can be entered in the fields.



Threshold angle

When the supplementary angle between two fiber segments is smaller than the value entered for the threshold angle, the two segments are merged. The process is repeated until no supplementary angle smaller than the threshold is left in the structure's fibers.

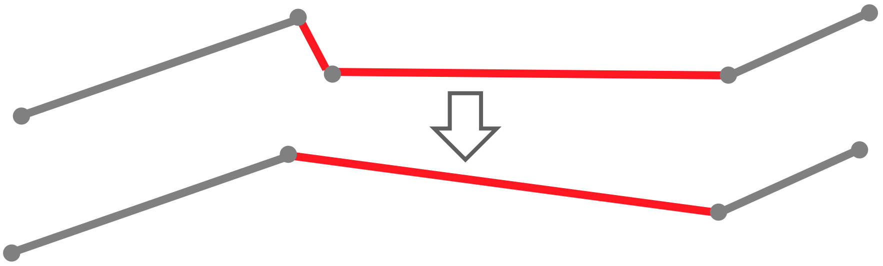


Threshold torsion

When the torsion within one segment is smaller than the entered threshold value, this segment is merged with its neighbor.

Threshold Length

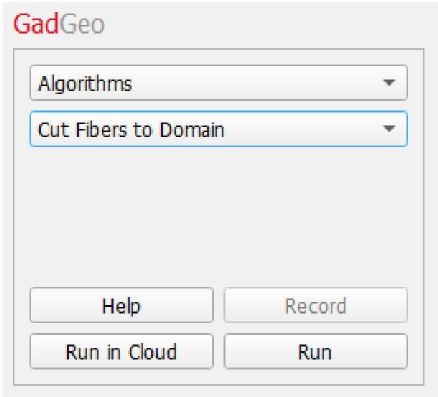
Segments shorter than the entered threshold value are merged with their neighbors.



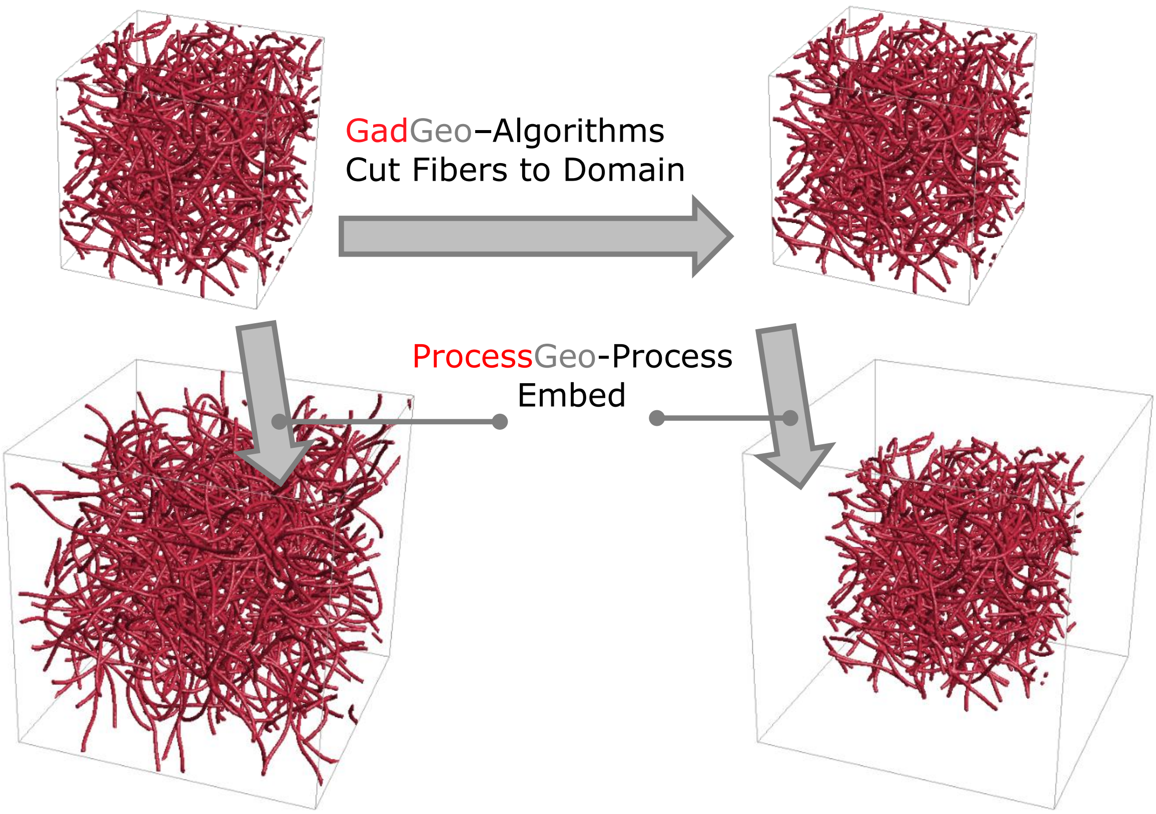
Cut Fibers to Domain

With **Cut Fibers to Domain**, all fibers in the domain are cut so that their endpoints are in the current domain. Infinite fibers are converted to finite fibers. Other objects (spheres, ellipsoids, …) are not affected.

This option is useful when exporting the fibers as objects, e.g., for meshing them with MeshGeo, or for combining them with other structures while maintaining the object information, e.g., in LayerGeo.



The effect of cutting fibers to the domain can best be observed when embedding the structure with ProcessGeo-Embed.



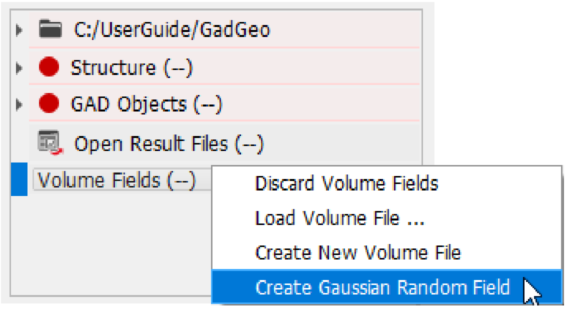
MATCH SVF DISTRIBUTION

The **Match Solid Volume Fraction (SVF) Distribution** command allows to create structures with an inhomogeneous solid volume distribution. A specialized form of this algorithm is also available in FiberGeo and GrainGeo, where a Gaussian random field is created and used directly in the generation step to achieve an inhomogeneous distribution.

Here, the algorithm can be used in a more general way.

Required input is a generated structure with GAD object information and periodic domain, and an arbitrary solid volume fraction distribution, defined through a GeoDict Volume Fraction (.gvf) input file.

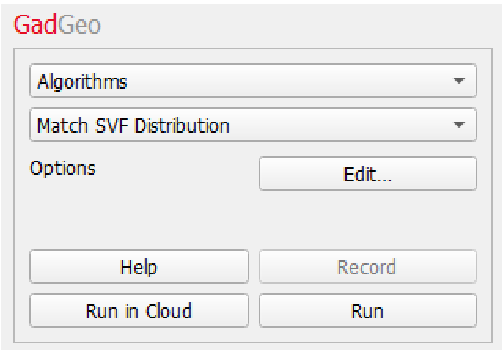
Such a file is created in FiberGeo or GrainGeo when the **Match Solid Volume Fraction (SVF) Distribution** was chosen.

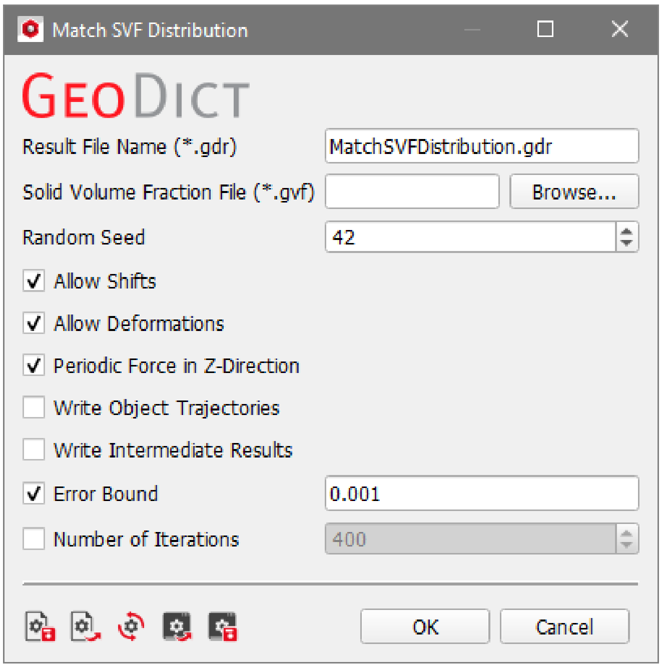


It can also be created using the **Create Gaussian Random Field** command available from the context menu of the **Volume Fields** entry (this feature is only available if licensed).

Alternatively, use **3D Inhomogeneity** in MatDict to create a distribution based on a CT-scan or a mask structure (see example below) or use the GeoPython library or other tools to create a file with the desired input values yourself.

Clicking the **Options**' **Edit...** button opens the **Match SVF Distribution** dialog where additional parameters can be entered.





RESULT FILE NAME

As usual, the name entered here is used for the created .gdr file and the corresponding folder.

Solid Volume Fraction File

Click **Browse...** to search for the .gvf file here. The chosen file must match to the current structure in the sense that the domain size (in physical length, not in the number of voxels) must be the same.

The absolute SVF values in the file are not important, as the distribution in the file is scaled to match the mean solid volume fraction of the loaded structure. The solid volume fraction file should have a coarser resolution as the structure file, otherwise the computation time will become large. For many cases a 10 times coarser resolution works well, meaning it has a voxel length of 10 µm when the original structure has a voxel length 1 µm.

Random Seed

The algorithm tries to achieve the given SVF distribution by modifying the GAD objects. This is done in a random order and the given random seed determines the order. Thus, different random seeds will lead to different results.

Allow Shifts

If checked, objects may be shifted to achieve the desired SVF distribution.

Allow Deformations

If checked, the objects will be deformed to achieve the desired SVF distribution. This applies only to objects that can be deformed at all. For example, fibers can be deformed, but not spherical grains.

Periodic Force in z-Direction

If checked, the force is periodic in all three directions, which is recommended for most cases.

If unchecked, the force is periodic in X- and Y-direction, but not in Z-direction. In that case, the objects are repelled from the domain boundary in Z-direction. This requires that some empty space be available at the top and bottom of the structure.

Write Object Trajectories

Stores the object movement in a GeoDict trajectory file (*.gpt), which can be visualized in GeoDict.

Write Intermediate Results

Stores intermediate results, e.g., for the structure after each iteration.

Error Bound

In each step, the objects are shifted and deformed by a certain value and the difference to the desired solid volume fraction is computed. Afterwards, the new difference is used to do the next iteration. This enables to compute how much the SVF has changed in the last iteration and define a certain value for which the simulation should be stopped. This is the recommended stopping criterion.

Number of Iterations

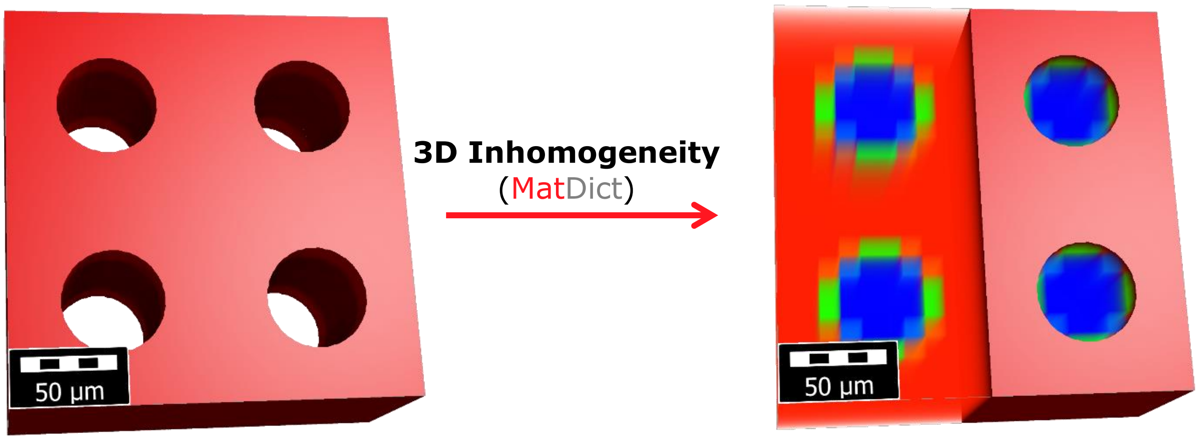
The process can also be stopped after a certain number of iterations has been done.

Match SVF Distribution Example

How to use **Match SVF Distribution** requires some more detailed explanations.

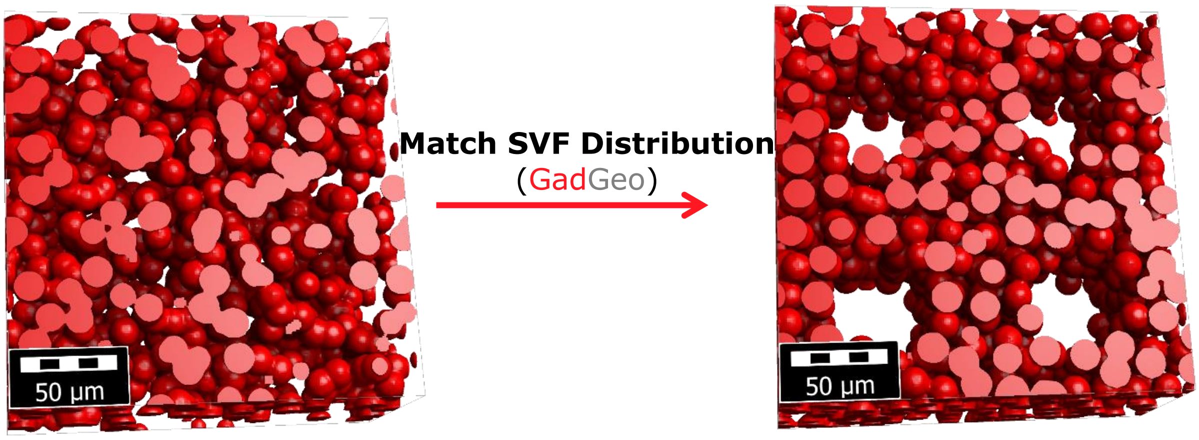
One way to use the algorithm is to create a mask structure and to convert this mask into a solid volume fraction file with a lower resolution (*.gvf). This file is afterwards used to change a base structure to follow the solid volume fraction distribution described by the mask file. This workflow is shown here.

First, a mask structure is created. Here, a perforated foil created with GridGeo is used. Afterwards, this mask is converted into a solid volume fraction distribution file using **Material Statistics -> 3D Inhomogeneity** with default settings in MatDict.



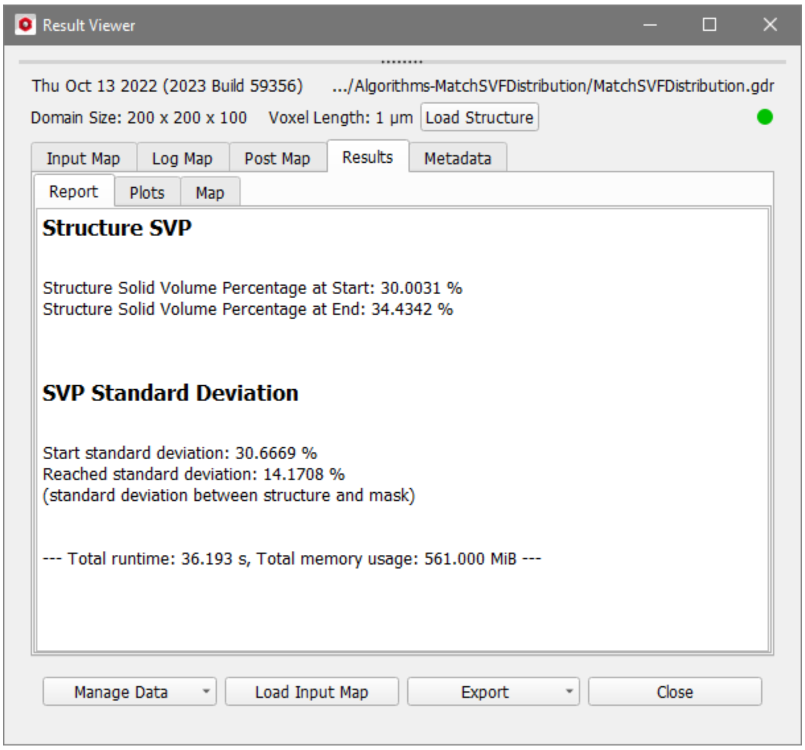
Now, we create a base structure of homogeneously distributed spheres with GrainGeo, where the mean solid volume fraction is 30%. The grain structure is loaded and the created solid volume fraction file (*.gvf) is chosen in the **Match SVF Distribution** dialog. Otherwise, we keep the default settings. Now, we run the Match SVF Distribution command.

The base structure, created with GrainGeo, and the structure matching the relative solid volume fraction distribution of the foil, are shown below.



RESULTS

As shown in the example, the main result is a structure matching the desired SVF distribution. Additionally, a *.gdr file is created, which contains information about how well the solid volume fraction distribution was matched and how the mean solid volume fraction has changed during the matching process.



Ungroup List-Objects

The **Ungroup List-Objects** command splits GAD objects composed of multiple segments or sub-objects into separate objects. To use this command, it is necessary that the analytic data of the current 3D structure is available (i.e., a green dot is shown in front of the **GAD Objects** status box entry).

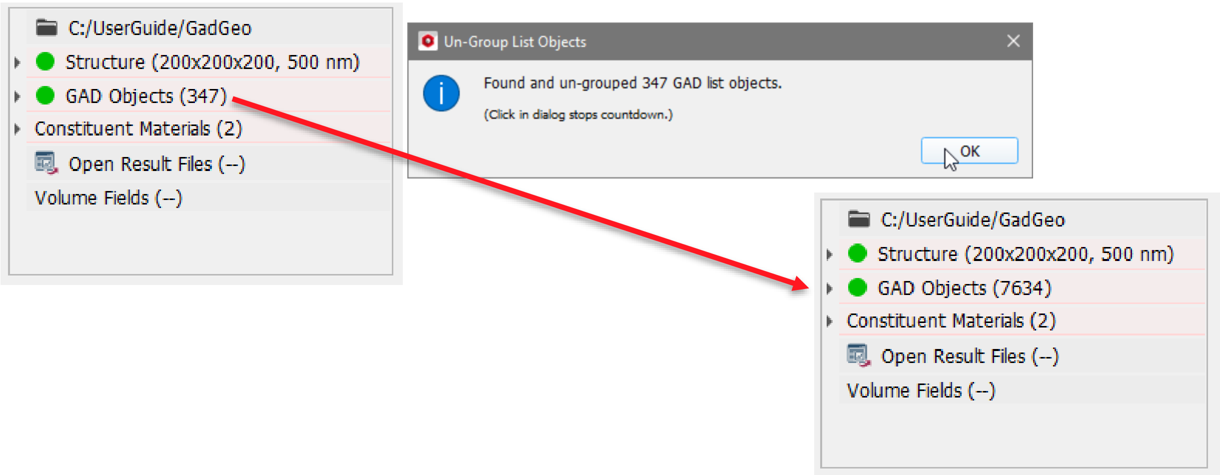
The following object types (see also the respective description of the objects in the Create /Add GAD Objects section on pages 8ff) consist of multiple segments or objects and are split into their respective segments when this command is run:

- CurvedXXXXFiber, where XXXX may be any one of the profile types Circular, Hollow, Rosetta, Elliptical, Cellulose, Rectangular, Angular, Arbitrary.
- GeneralCurvedXXXXFiber, where XXXX may be any one of the profile types Circular, Hollow, Rosetta, Elliptical, Cellulose, Rectangular, Angular, Arbitrary.
- CombinedObject
- IntersectedObject

Other object types present in the current 3D structure will remain unchanged.

The **Ungroup List-Objects** command does not require additional parameters. Therefore, no **Edit...** button is available, and the command can simply be executed by clicking **Run**.

When running the command, a pop-up message informs about the number of objects changed and it is observable that the overall number of objects is increased.

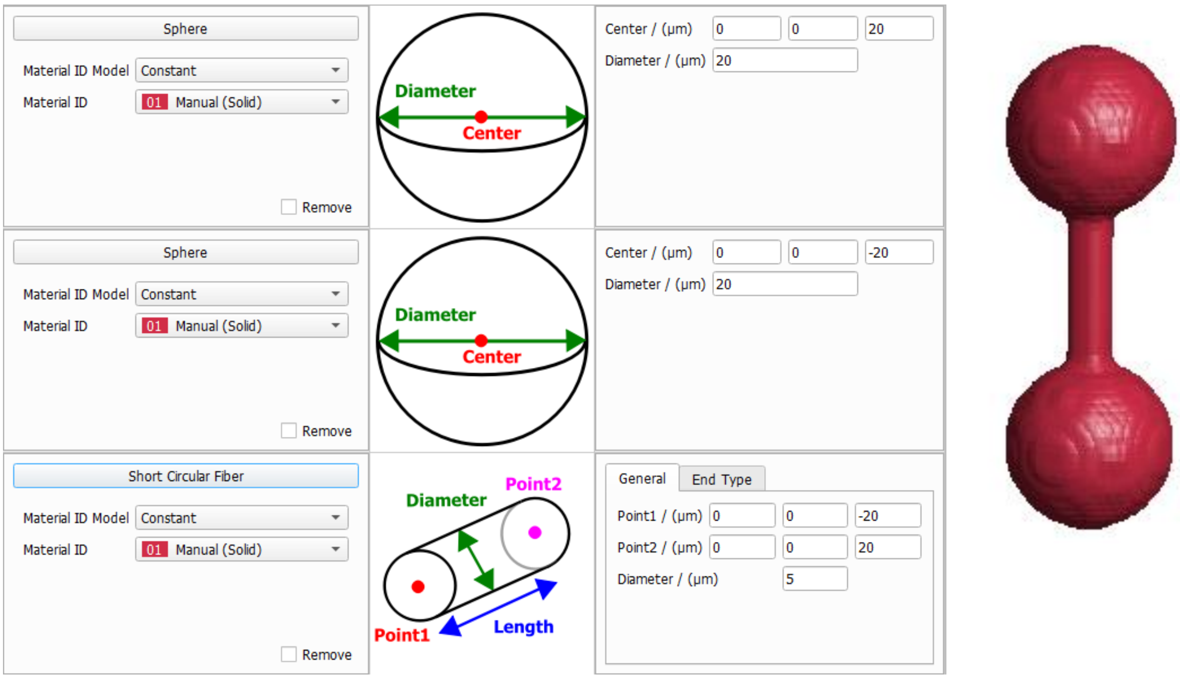


Combine GAD Objects

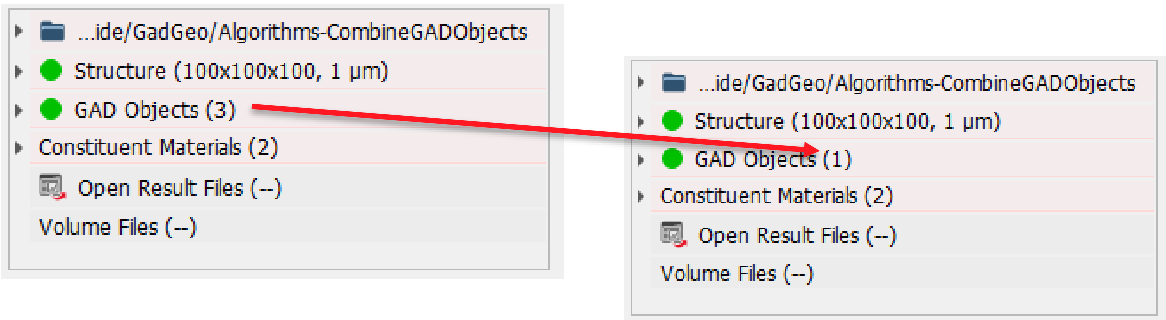
The Combine GAD Objects command combines all objects into a single CombinedObject.

The **Combine GAD Objects** command does not require additional parameters. Therefore, no **Edit…** button is available, and the command can simply be executed by clicking **Run**.

As example, consider two spheres and a cylinder that together form a dumbbell.



Initially, these are three independent objects, and in the status section, 3 GAD Objects are shown. After clicking **Run,** the three objects are turned into a single object

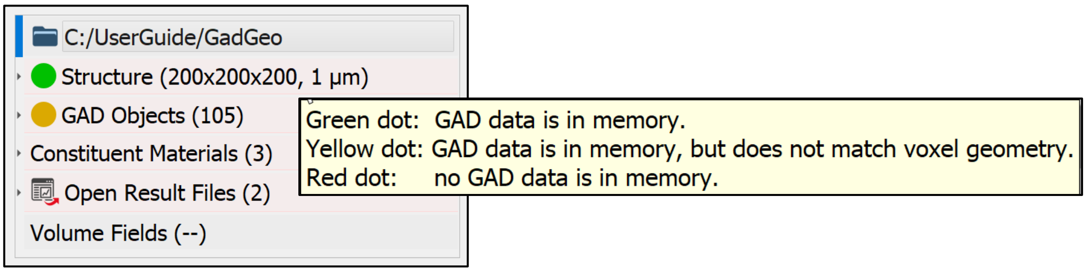


If this object is now stored in a .gad file, only a single object of the CombinedObject type is present. This file can then be used as a user-defined object type as described on page 33.

```
"Object1" : {
  "MaterialIDModel"         : "SubObjectMapping",
  "MaterialID"              : [1,1,1],
  "Type"                    : "CombinedObject",
  "NumberOfSubObjects"      : 3,
  "Position"                : [0,0,0],
  "ScalingFactor"           : 1,
  "Axis1"                   : [1,0,0],
  "Axis2"                   : [0,1,0],
  "SubObject1" : {
    "Operation"             : "add",
    "Type"                  : "Sphere",
    "Position"              : [0,0,2e-05],
    "Diameter"              : 2e-05,
    "Axis1"                 : [1,0,0],
    "Axis2"                 : [0,1,0]
  },
  "SubObject2" : {
    "Operation"             : "add",
    "Type"                  : "Sphere",
    "Position"              : [0,0,-2e-05],
    "Diameter"              : 2e-05,
    "Axis1"                 : [1,0,0],
    "Axis2"                 : [0,1,0]
  },
  "SubObject3" : {
    "Operation"             : "add",
    "Type"                  : "ShortCircularFiber",
    "Point1"                : [0,0,-2e-05],
    "Point2"                : [0,0,2e-05],
    "FiberEndType1"         : "Flat",
    "FiberEndType2"         : "Flat",
    "Diameter"              : 5e-06
  }
},
```

Re-Voxelize GAD Objects

It is possible that the current list of GAD objects does not fully describe the current voxel structure. This can happen if binder (which has no gad representation) is added to a fiber or grain structure, or if the voxel structure has been modified in any other way. This is shown by a yellow dot in the project status window:

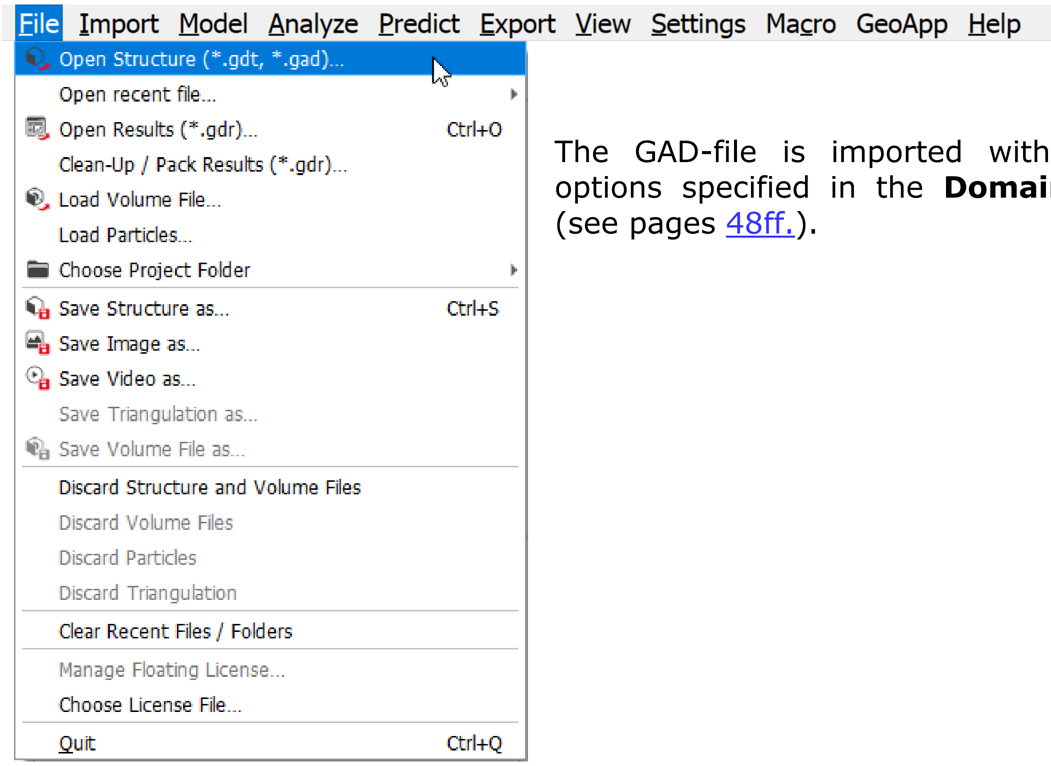


In this situation, it is possible to run **Re-Voxelize GAD Objects** and re-create the voxel structure that is a representation of the current GAD objects.

The **Re-Voxelize GAD Objects** command does not require additional parameters. Therefore, no **Edit…** button is available, and the command can simply be executed by clicking **Run**.

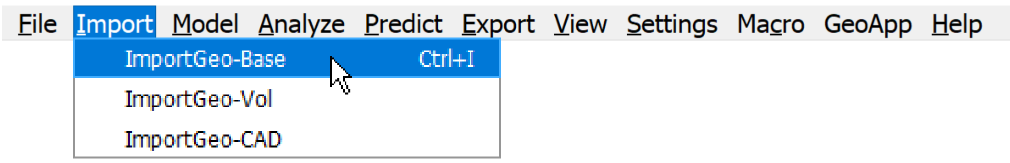
IMPORTING GAD DATA

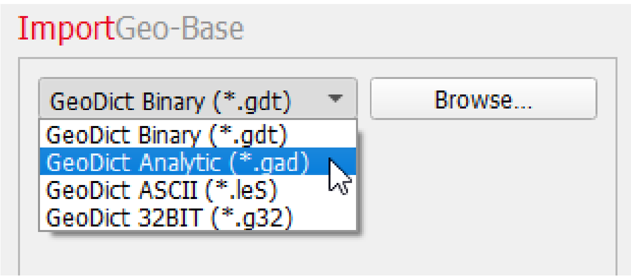
The simplest way to open gad-files in GeoDict is by selecting **File → Open Structure (*.gdt, *.gad) ...** in the menu bar, or by clicking its icon in the Toolbar ( ).



The GAD-file is imported with the domain options specified in the **Domain** parameters (see pages 48ff.).

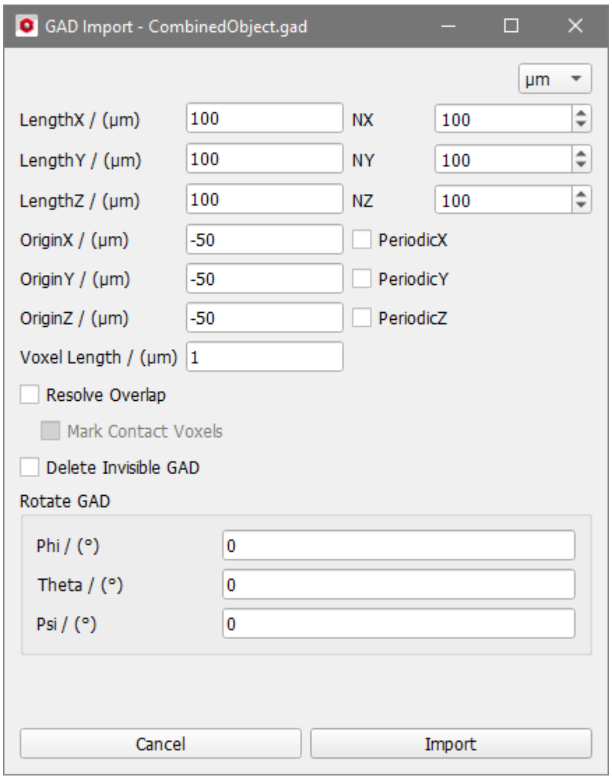
However, to load the GAD file with values different to those specified in the **Domain**, parameters, open it using ImportGeo-Base. Select **Import → ImportGeo-Base** in the menu bar.





Then, select the gad-file format (**GeoDict Analytic (*.gad)**) from the pull-down menu in the **ImportGeo-Base** section, and browse for the GAD file you want to import.

The **GAD Import** dialog opens, and the gad-file information is loaded. The domain information is automatically entered from the **Domain** in the file and may be changed.



By changing the side **Length** and **Origin**, it is possible to import only a cut-out of the material. By changing the **Voxel Length,** it is possible to increase or decrease the resolution of the imported 3D structure.

By choosing **Resolve Overlap**, the overlapping areas will be assigned to the closest object as described on page 30. In this case, it is then again possible to **Mark Contact Voxels** with a special material ID.

If **Delete Invisible GAD** is checked, all objects which are completely outside of the chosen domain will not be imported.

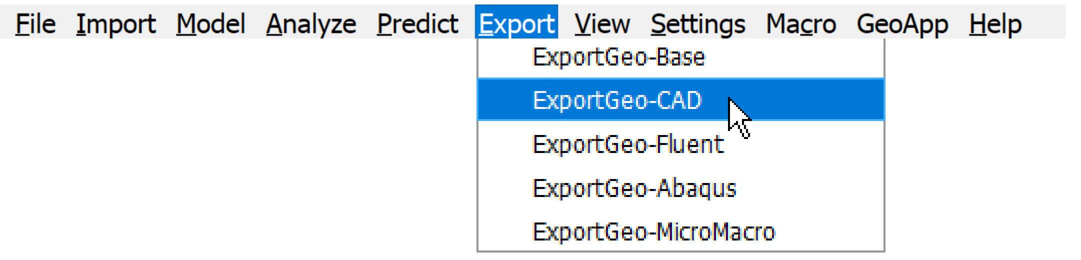
Then, the GAD file is imported with the new entered domain parameters when clicking **Import**.

The import with the ImportGeo-Base module additionally allows the rotation of the analytic data during the import.

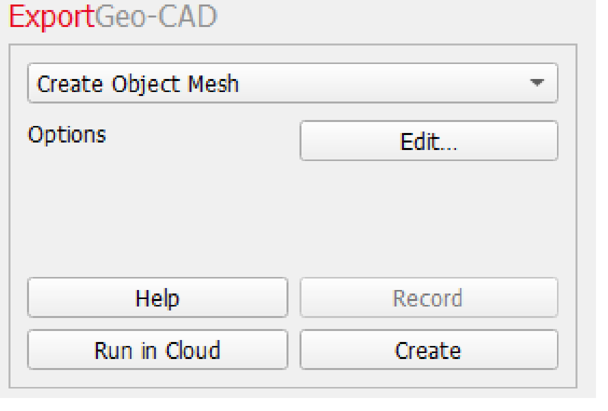
EXPORTING GAD DATA

With GeoDict's ExportGeo-CAD module, analytic data can be exported to various file formats.

Having a structure in GAD format loaded in the memory, select **Export** → **ExportGeo-CAD** in the menu bar.



To directly export the GAD objects, and not the voxelized structure, choose **Create Object Mesh** in the **ExportGeo-CAD** section.



For more information, consult the ExportGeo-CAD- MeshGeo handbook of this User Guide.

https://doi.org/10.30423/userguide.geodict

Technical documentation:

Jürgen Becker

Dennis Mosbach

Barbara Planas

MATH 2 MARKET

Math2Market GmbH

Richard-Wagner-Str. 1, 67655 Kaiserslautern, Germany
www.geodict.com