GadGeo

User Guide

GEODICT Release 2026

Published: June 2026

GeoDict

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

GadGeo

1 GadGeo

Construct and manipulate material models

GadGeo allows to construct 3D geometries from basic 3D objects like spheres, ellipsoids, fibers, etc.

The module also contains all functions available to manipulate these 3D objects in GeoDict. You can create a 3D structure in one of the other structure generation modules of GeoDict (e.g., FiberGeo or GrainGeo) and then modify this structure in GadGeo. You may remove objects from the structure or add new objects to it. You may change the properties of any object in the structure (size, position, Material ID).

You can change the field of view or the resolution of the domain with GadGeo, too.

New in GeoDict 2026

The Export as GAD Objects 62 command does not directly export visualized results like particles, trajectories and streamlines anymore. Instead, you can use the new Convert to GAD 64 commands to convert the visualized results to GAD objects first, and then export the 3D structure.

Learn to use GadGeo:

- Introduction to GAD Objects 6
  Learn to distuiguish between geometric object descriptions and their representation in a 3D voxel mesh.
- Object Types 9
  Find descriptions and parameter lists of all available object types here.
- Create / Add GAD-Objects 31
  Create new objects or add objects to the current structure.
- Edit GAD Objects 41
  Modify the current objects in various ways.
- Edit Domain 53
  Change the domain settings.
- Create GAD Index Image 57
  Access the object index numbers.
- Create GOF Orientation Image 58
  Make the intrinsic orientation of the GAD objects available for subsequent simulations.
- Add & Import 59
  Import GAD objects from a file and add them into the current structure or domain.
- Export as GAD Objects 62
  Create a GAD file from the current objects.

Learn to use GadGeo:

- [Convert to GAD] 64
  Convert visualized results like particle trajectories or streamlines into GAD objects that become part of the 3D voxel structure.
- [Algorithms] 68
  Modify the existing objects through a collection of algorithms.

Join the conversation

Visit the Community on GeoDict Forum to be inspired and get answers to top questions.
Find illustrative application examples on the website.
Our tutorials and seminar videos give you a first hands-on experience. They are collected in the Learning Center.
Send your Support Request to our technical Support.

1.1 Introduction to GAD Objects

GeoDict Analytic Data (GAD)

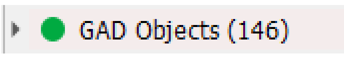
Structure models in GeoDict can be described in two fundamentally different ways.

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 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 **G**eoDict **A**nalytic **D**ata.

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, called GAD 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 2026.

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

```
# Example .gad File

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

  "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 60. The content of the Domain section is described in more detail with the Edit Domain 53 command.

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 chapter.

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**. Some non-constant patterns are also possible, see the description of the Change Material ID Model 48 command for details.
- The value after the key **MaterialID** specifies the Material ID 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 within the Create / Add GAD Objects 31 command.
- 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 in Create / Add GAD Objects 31.

- 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.

1.2 Object Types

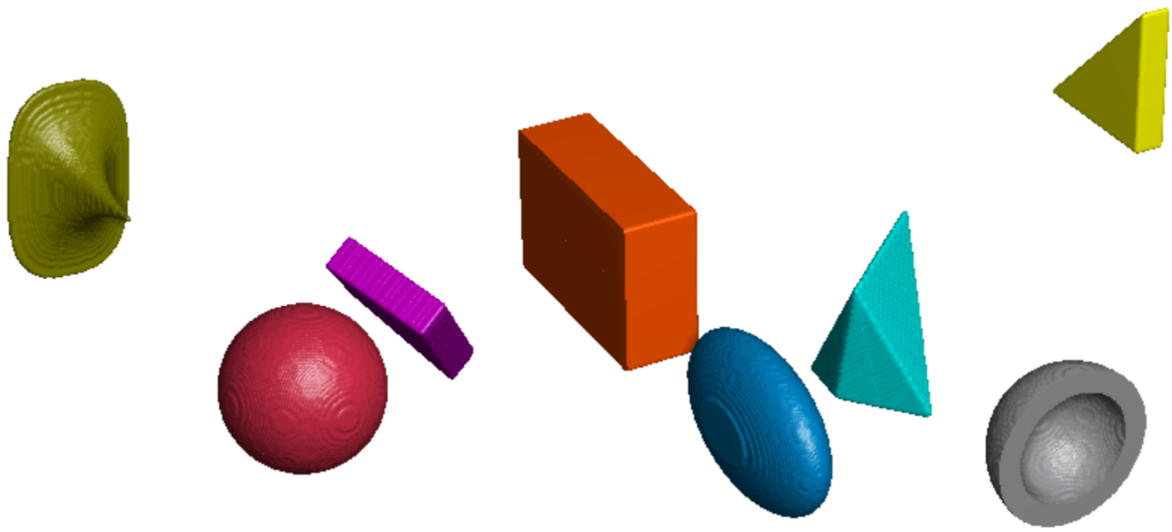
All available GAD object types, shapes, and profiles and their representation in the GAD file are explained in the following.

Shapes and profiles and their representations

- Grain Shapes 9
  Parameter definition for spheres (filled and hollow), ellipsoids, boxes, triangular prisms, superquadric particles, and polyhedrons (planar and convex).
- Fiber Shapes 16
  Parameter definition for short and long fibers, straight and curved, with fixed or varying diameters.
- Fiber Profiles 21
  Parameter definitions for various fiber cross sections.
- Combined Object 25
  Parameter definition for objects combining several standard objects into a new object.
- Intersected Object 28
  Paramter definition for objects created by intersection of multiple standard objects.

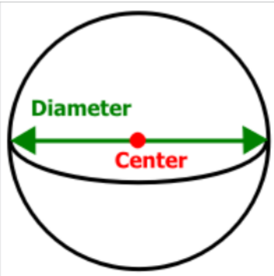
1.2.1 Grain Shapes

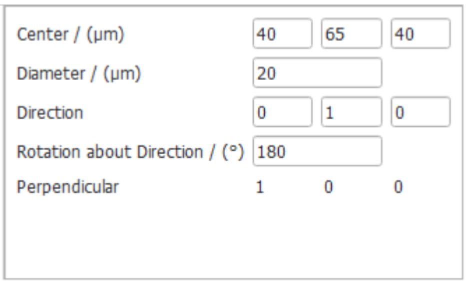
Several geometric shapes are available to create objects consisting of grains: spheres, hollow spheres, ellipsoids, and their generalized form of superquadric particles, rectangular boxes, triangular prisms, and planar and convex polyhedrons.

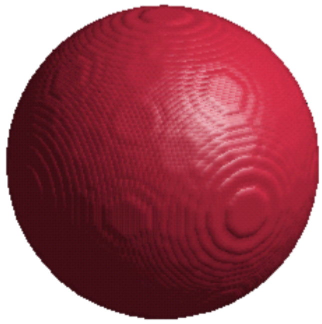


Sphere

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**).







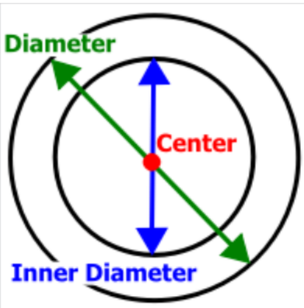
```
"Object1" : {
  "MaterialIDModel"     : "Constant",
  "MaterialID"          : 1,
  "Type"                : "Sphere",
  "Position"            : [4e-05,6.5e-05,4e-05],
  "Diameter"            : 2e-05,
  "Axis1"               : [0,1,0],
  "Axis2"               : [1,0,0]
},
```

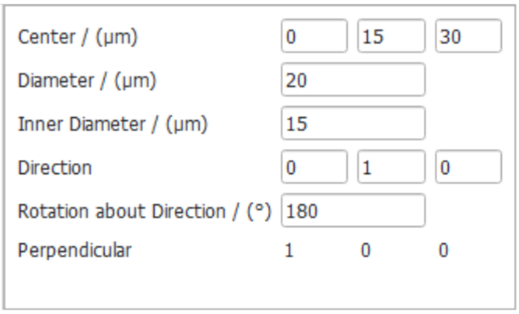
The parameters **Axis1** and **Axis2** in the .gad file describe the orientation of the object. The third coordinate axis is not given as an additional parameter, but given as the cross product of **Axis1** and **Axis2**, such that the three axes form a right hand system. **Axis1** corresponds to the given **Direction** in the GUI, and **Axis2** to the **Perpendicular** calculated from the entered rotation angle.

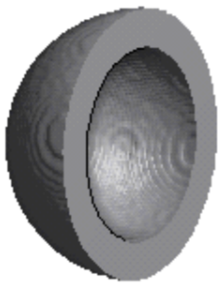
The orientation of a sphere obviously does not change the visible 3D geometry, rather, it can be used to model anisotropic material behavior.

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**).







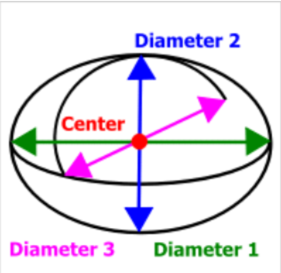
```
"Object2" : {
  "MaterialIDModel"     : "Constant",
  "MaterialID"          : 2,
  "Type"                : "HollowSphere",
  "Position"            : [0,1.5e-05,3e-05],
  "Diameter"            : 2e-05,
  "InnerDiameter"       : 1.5e-05,
  "Axis1"               : [0,1,0],
  "Axis2"               :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

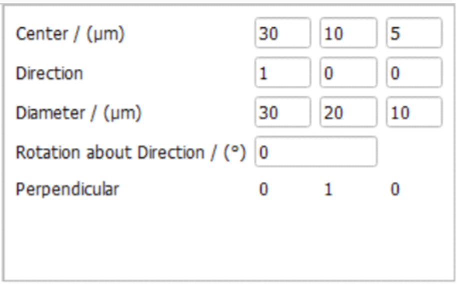
As for spheres, the parameters **Axis1** and **Axis2** in the .gad file describe the orientation of the object. The third coordinate axis is not given as an additional parameter, but given as the cross product of **Axis1** and **Axis2**, such that the three axes form a right hand system. **Axis1** corresponds to the given **Direction** in the GUI, and **Axis2** to the **Perpendicular** calculated from the entered rotation angle.

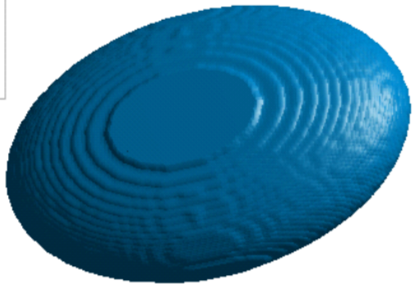
The orientation of a hollow sphere obviously does not change the visible 3D geometry, rather, it can be used to model anisotropic material behavior.

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]
},
```

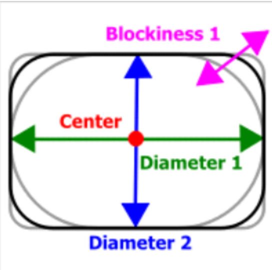
Two of these main axes are given in the gad-file by the vectors after the keys **Axis1** and **Axis2**. **Axis1** corresponds to the given **Direction** in the GUI, and **Axis2** to the **Perpendicular** calculated from the entered rotation angle. Again, the third main axis is determined through the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

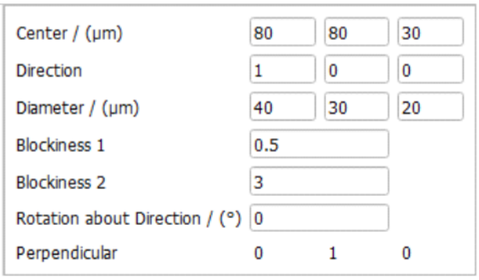
Superquadric Particle

The superquadric particle is implicitly defined by the formula:

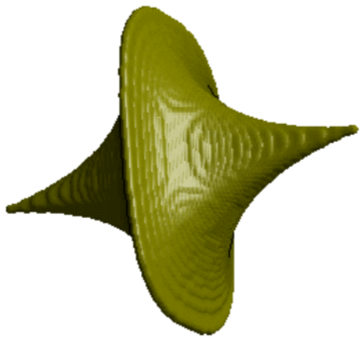
$$\left(\left|\frac{x}{d_1}\right|^{n_2} + \left|\frac{y}{d_2}\right|^{n_2}\right)^{\frac{n_1}{n_2}} + \left|\frac{z}{d_3}\right|^{n_1} = 1 \qquad (1)$$

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, and 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.






```
"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
},
```



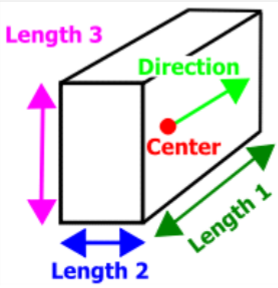
The three **Diameter values** correspond to the diameters d_1, d_2, d_3 in formula ((1) 12), **Blockiness 1** to the exponent n_1 and **Blockiness 2** to the exponent n_2. The **Position** of the particle is given through its **Center / Position**, and the orientation of the object. **Axis1** corresponds to the given **Direction** in the GUI, and **Axis2** to the **Perpendicular** calculated from the entered rotation angle.

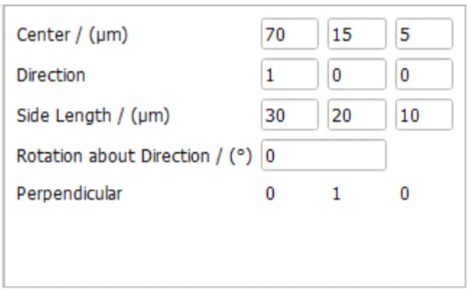
Box

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

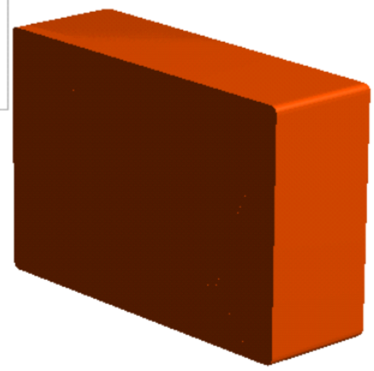
The **Center** position is given by the vector after the key **Position**, and the two main axes are given by the vectors after the keys **Axis1** and **Axis2**. **Axis1** corresponds to the given **Direction** in the GUI, and **Axis2** to the **Perpendicular** calculated from the entered rotation angle. 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**.





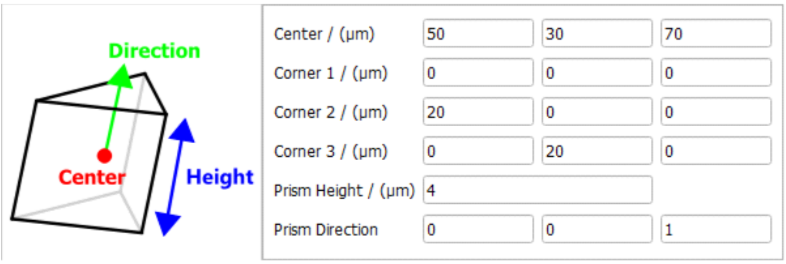

```
"Object4" : {
  "MaterialIDModel"    : "Constant",
  "MaterialID"         : 4,
  "Type"               : "Box",
  "Position"           : [7e-05,1.5e-05,5e-06],
  "Length1"            : 3e-05,
  "Length2"            : 2e-05,
  "Length3"            : 1e-05,
  "Axis1"              : [1,0,0],
  "Axis2"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



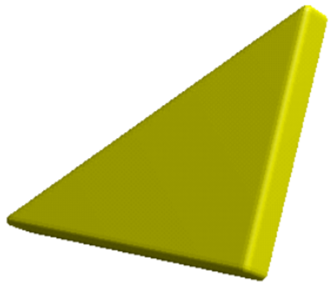
The additional parameters **CornerRadius** and **CornerMode** are used in conjunction for rounded boxes. **CornerRadius** describes the radius of the corner. **CornerMode Rounded** is used for rounded boxes, instead of the **CornerMode Sharp** that is used for normal boxes. Those parameters cannot be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

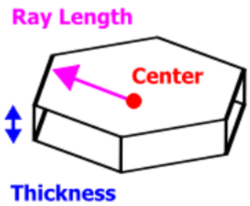


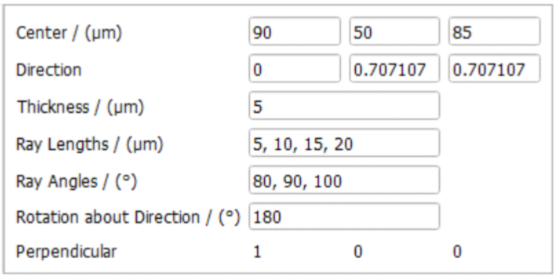
The **Center** (corresponds to the **Position** vector in the .gad file) describes the center of mass of the prism, the **Prism Height** (**Thickness**) the distance between the two parallel triangles. The **Prism Direction** (**Normal** in the .gad file) is the shift direction between the two parallel 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.

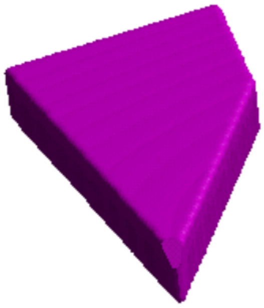
Planar Polyhedron

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)**.







```
"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 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.

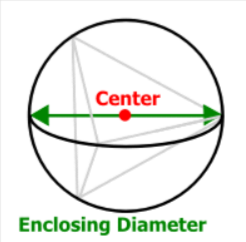
The angle between the first ray and the second ray is given by the value after the key **RayAngle1** and so on. In the user interface, the number of rays is given implicitly by the number of entered and comma-separated **Ray Lengths**.

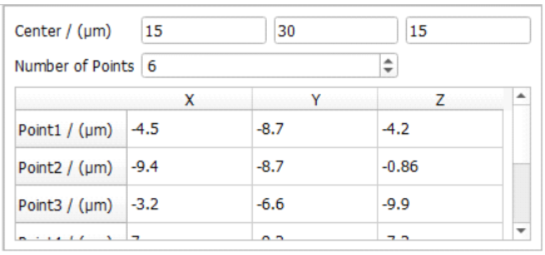
Convex Polyhedron

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**.

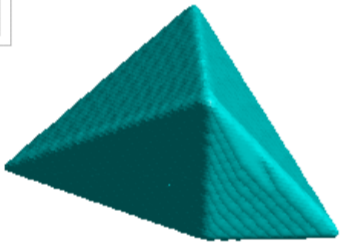
The coordinates of the first point, relative to the center, are given by the vector after the key **Point1**.

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.

1.2.2 Fiber Shapes

The names used for the fiber **Type** key are composed by the fiber shape, followed by the fiber profile (or cross-section) and the ending **Fiber**. The Fiber Profile 21 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**

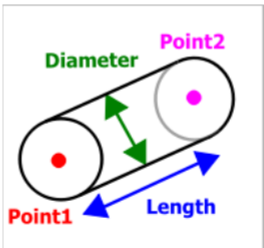
Profiles: **Circular**, **Hollow**, **Elliptical**, **Rectangular**, **Angular**, **Rosetta**, **Cellulose**, **Arbitrary**, **Strut**

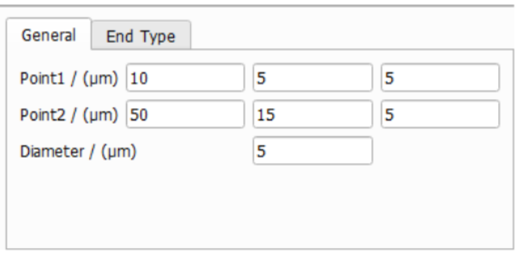
For example, **CurvedRosettaFiber** or **GeneralShortRectangularFiber** denote a **Type**.

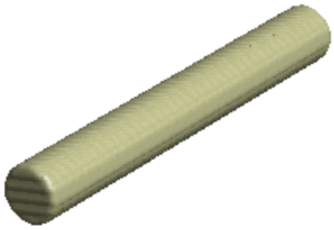
In the following, all fiber shapes are explained for the Circular profiles.

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







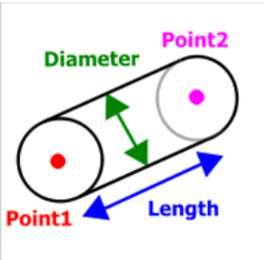
```
"Object1" : {
  "MaterialIDModel"      : "Constant",
  "MaterialID"           : 1,
  "Type"                 : "ShortCircularFiber",
  "Point1"               : [1e-05,5e-06,5e-06],
  "Point2"               : [5e-05,1.5e-05,5e-06],
  "FiberEndType1"        : "Flat", # Possible values: Flat, CuttingPlane,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

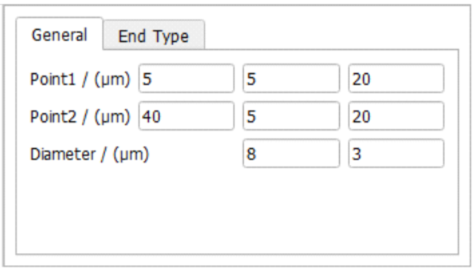
On the second tab, define the **End Type** as described here 21.

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**).





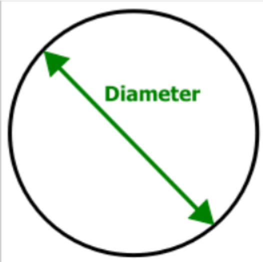
```
"Object2" : {
  "MaterialIDModel"      : "Constant",
  "MaterialID"           : 2,
  "Type"                 : "GeneralShortCircularFiber",
  "Point1"               : [5e-06,5e-06,2e-05],
  "Point2"               : [4e-05,5e-06,2e-05],
  "FiberEndType1"        : "Flat",
  "FiberEndType2"        :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

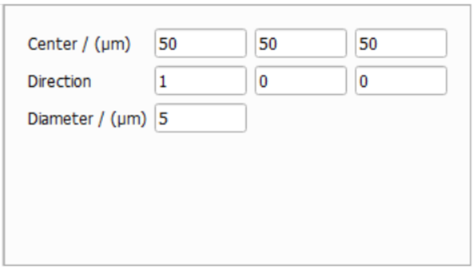
On the second tab, define the **End Type** as described here 21.

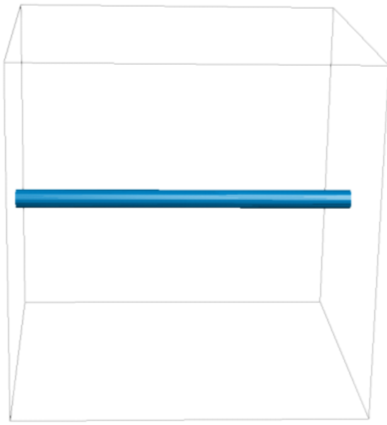
Infinite Fiber

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.





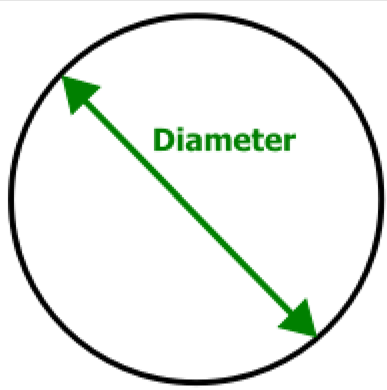


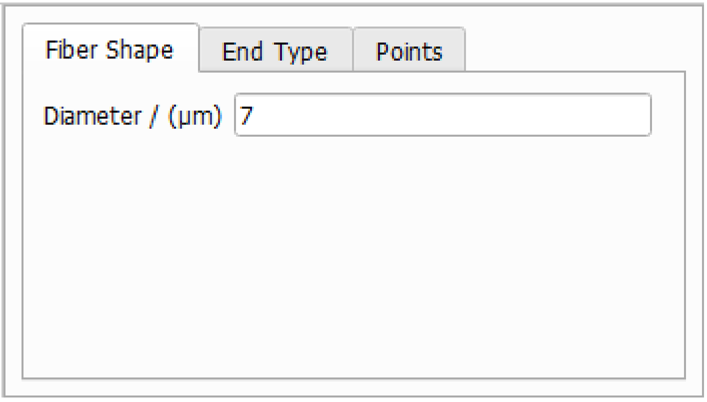
```
"Object3" : {
  "MaterialIDModel"    : "Constant",
  "MaterialID"         : 3,
  "Type"               : "InfiniteCircularFiber",
  "Position"           : [5e-05,5e-05,5e-05],
  "Direction"          : [1,0,0],
  "Diameter"           : 5e-06
},
```

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.

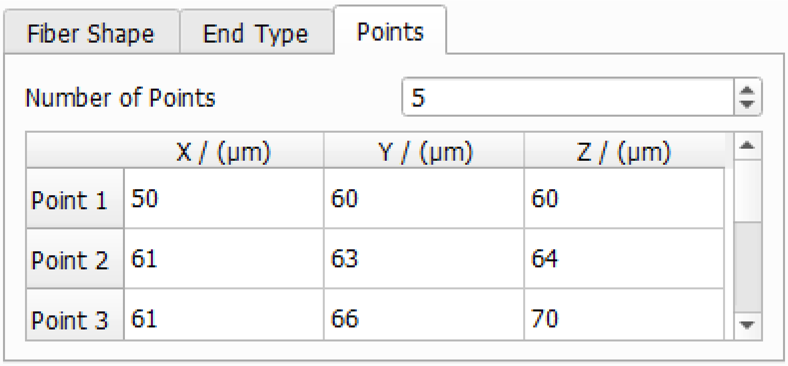




The circular fiber profile is again described by a single **Diameter**.

On the second tab, define the **End Type** as described here 21.

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

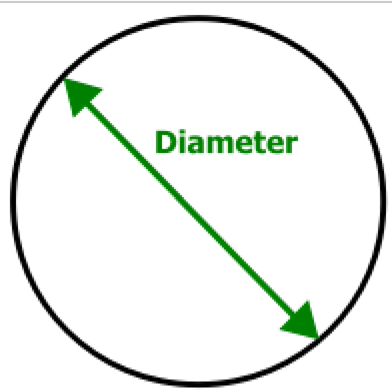


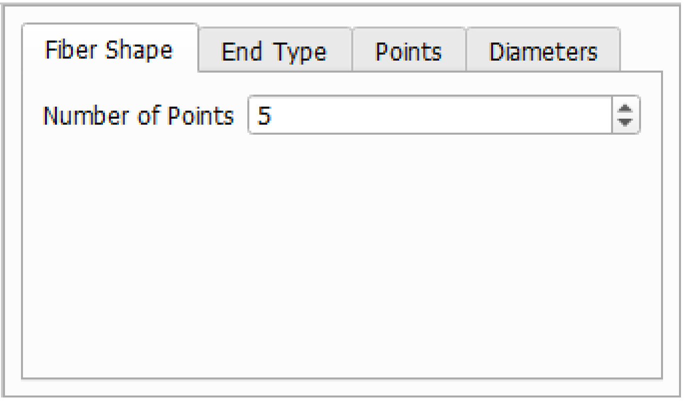
In the GAD file, the **NumberOfSegments** is given instead, which corresponds to the given **Number of Points** -1.

```
"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.

On the second tab, define the **End Type** as described here 21.

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

Fiber Shape | End Type | Points | Diameters

	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

Fiber Shape | End Type | Points | Diameters

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

In 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.

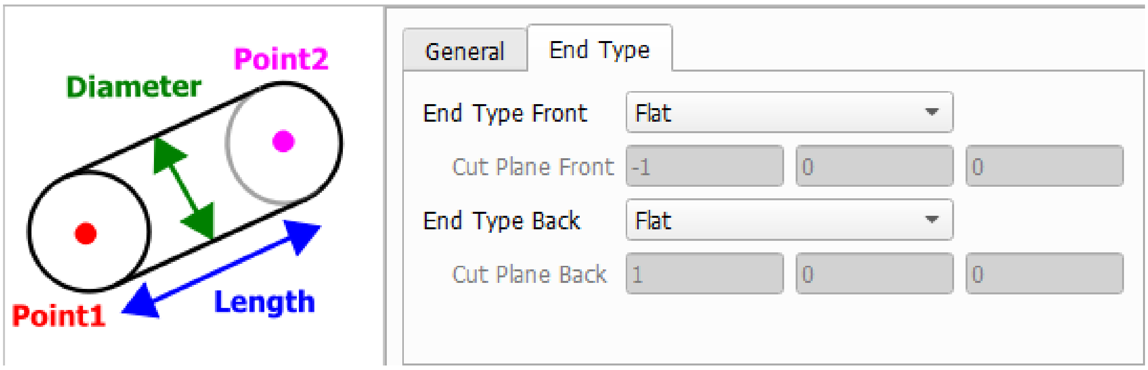
In the GAD file, the number of segments is given by the value after the key **NumberOfSegments**. The profile parameters are given in submaps: the parameters for the first segment are in **Segment1**, the parameters for the second are in **Segment2**, and so on.

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]
},
```

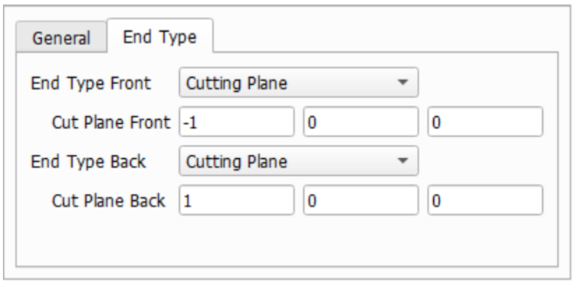
The **Short**, **GeneralShort**, **Curved** and **GeneralCurved** fiber types may end inside of the domain. For each of these fibers, you can define how the first and last fiber segment is terminated geometrically.

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 cutting plane. No additional parameters are needed.
- **CuttingPlane**: The fiber ending is planar, and the cutting 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.

1.2.3 Fiber Profiles

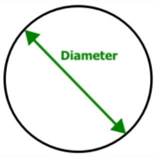
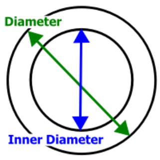
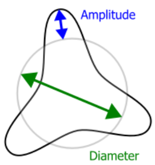
The profile of the circular fiber types can be described by a single diameter value, or two diameter values for each general segment. In general, for the **Infinite**, **Short**, or **Curved** fiber shapes only a single set of descriptors of a fiber profile is needed.

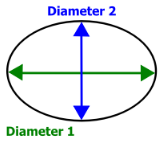
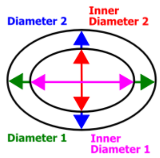
For the **GeneralShort** fiber shape, two sets of descriptors are needed, one for the start point of the fiber, and one for the end point of the fiber.

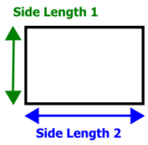
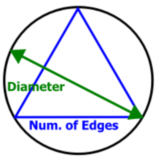
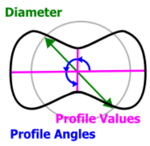
For the **GeneralCurved** fiber shape, two sets of descriptors are needed for each segment, one for the start point of a segment, and one for the end point of a segment. One fiber may be composed of several segments.

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).

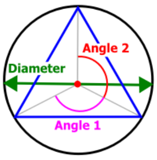
In detail, the parameters for the profile types are as follows:

Circular 	**Infinite** **Short** **Curved**	A circular profile is defined by one diameter, given by the value after the key **Diameter**.
	GeneralShort **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** **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 **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** **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 **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** **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 **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** **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 **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.

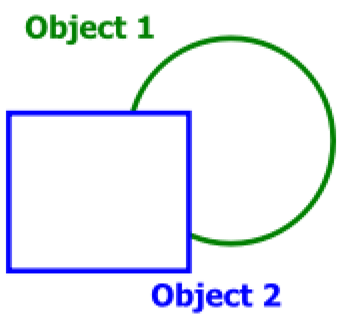
<table>
<tr>
<td rowspan="2">Rectangular

</td>
<td>Infinite
Short
Curved</td>
<td>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, where SideLength1 points in the direction of Perpendicular.</td>
</tr>
<tr>
<td>GeneralShort
GeneralCurved</td>
<td>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.</td>
</tr>
<tr>
<td rowspan="2">Angular

</td>
<td>Infinite
Short
Curved</td>
<td>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.</td>
</tr>
<tr>
<td>GeneralShort
GeneralCurved</td>
<td>The angular 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.</td>
</tr>
<tr>
<td>Arbitrary

</td>
<td>Infinite
Short
Curved</td>
<td>An arbitrary 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.

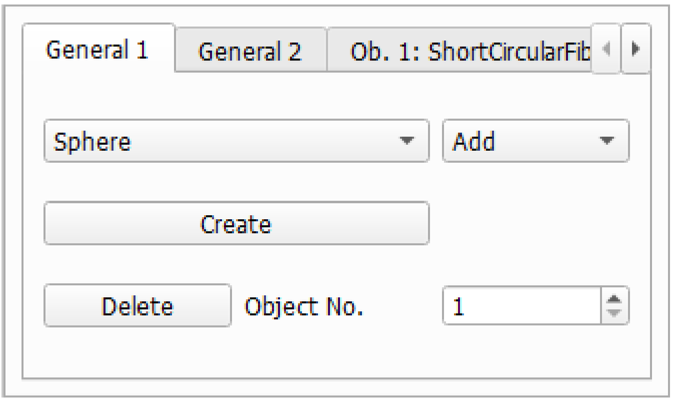
The Curved fiber shape is not editable in GadGeo and cannot be selected in the Create / Add GAD Objects command.</td>
</tr>
</table>

	GeneralShort **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** **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 **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.

1.2.4 Combined Object

A combined object is a combination of sub-objects, e.g., **Sphere** 9, **Ellipsoid** 11, or **Box** 12. Objects can be combined by adding different objects together, or by subtracting an object from another one.



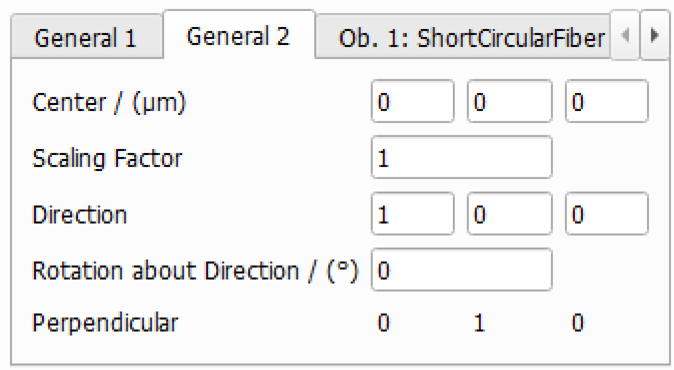


In the **General 1** tab, standard objects can be added to the combined type, or subtracted from the combined type. Upon startup, a Combined Object consists of a single sphere, which can be removed by deleting **Object No. 1** using the **Delete** button.

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** angle.

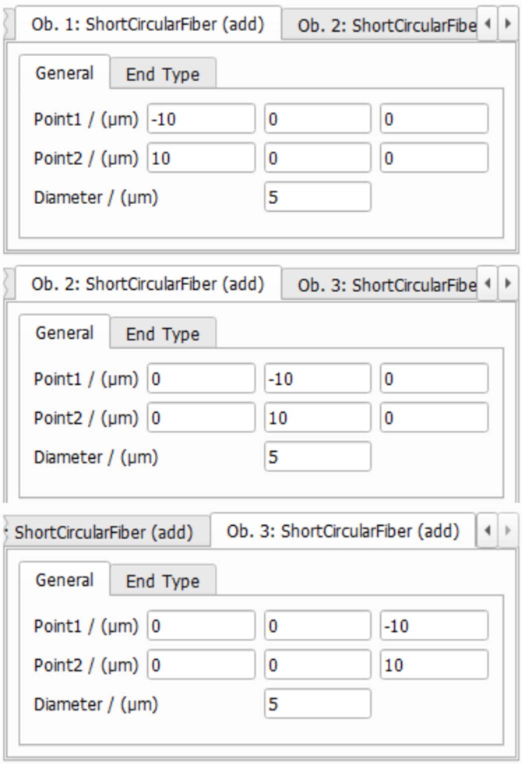
The z-axis is then already automatically defined as the cross product of the two other axes, such that a right hand system is formed. The **ScalingFactor** sets the relation between object sizes in the local object coordinates and the global Euclidean coordinate system. A **Scaling Factor** of 2 means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

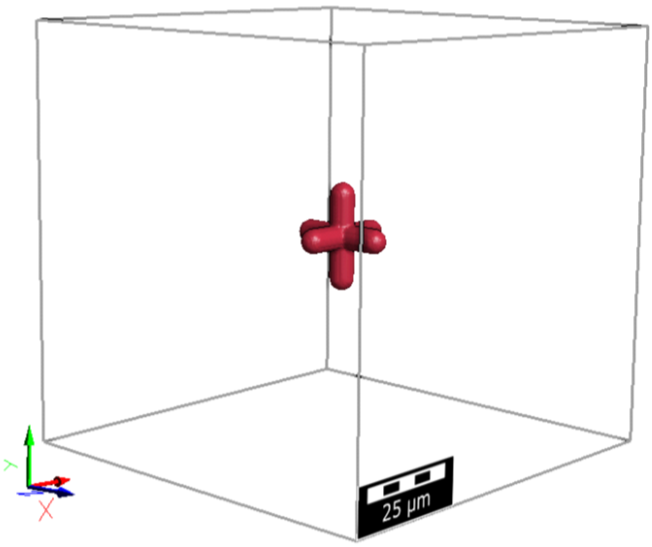
In the GAD file, the number of sub-objects is given by the value after the key **NumberOfSubObjects**.

```
"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.

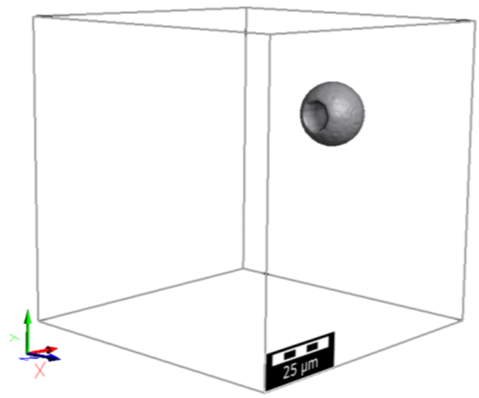
In the example above, three cylinders form a 3D cross as a new object type.



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:

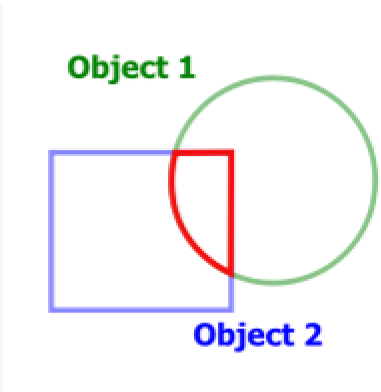
```
"Object2" : {
  "MaterialIDModel"        : "Constant",
  "MaterialID"             : 2,
  "Type"                   : "CombinedObject",
  "NumberOfSubObjects"     : 2,
  "Position"               : [1.5e-05,2e-05,2e-05],
  "ScalingFactor"          : 2,
  "Axis1"                  : [1,0,0],
  "Axis2"                  : [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
  }
},
```

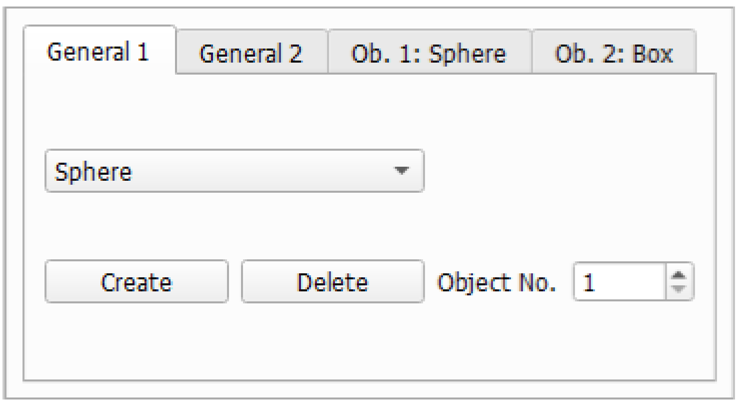


1.2.5 Intersected Object

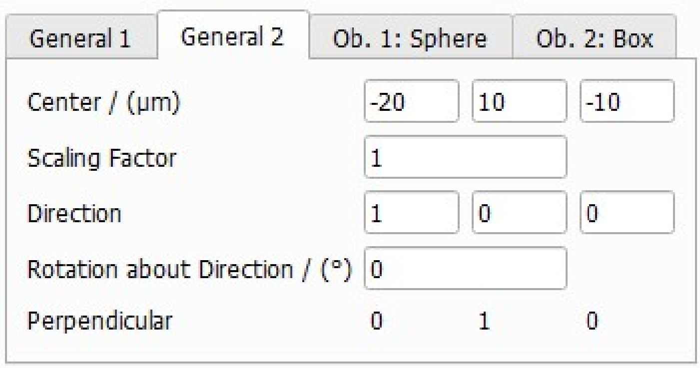
An intersected object is specified by an intersection of sub-objects. A sub-object may be any of the object types: **Sphere**, **Ellipsoid**, **Box**, etc.

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.

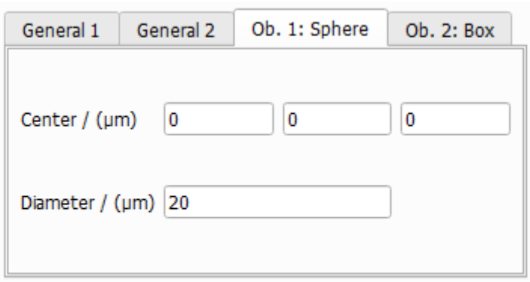


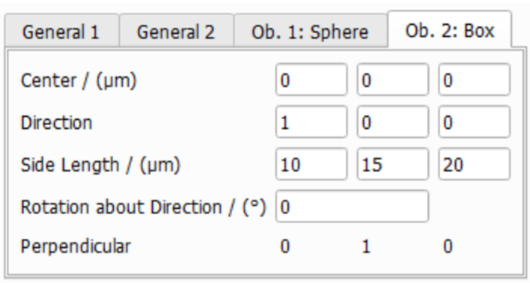
In the **General 2** tab, the position of the new object can be defined and the local coordinate system can be set up as described in the **Combined Object** 25 section.

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.

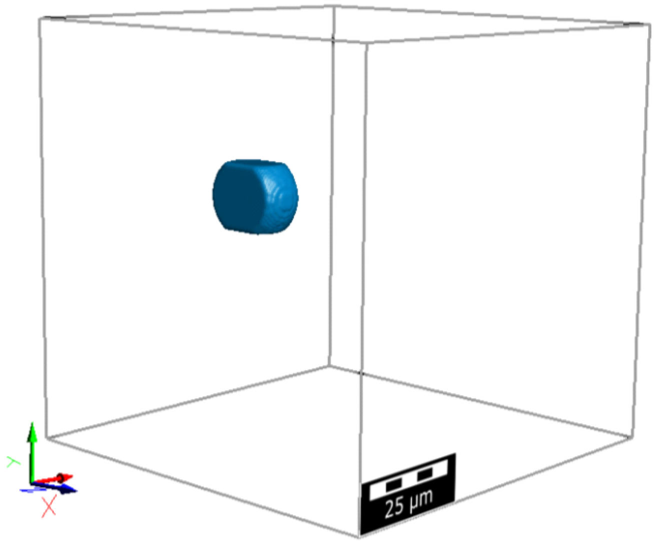
```
"Object3" : {
  "MaterialIDModel"        : "Constant",
  "MaterialID"             : 3,
  "Type"                   : "IntersectedObject",
  "NumberOfSubObjects"     : 2,
  "Position"               : [-2e-05,1e-05,-1e-05],
  "ScalingFactor"          : 1,
  "Axis1"                  : [1,0,0],
  "Axis2"                  : [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
  }
},
```

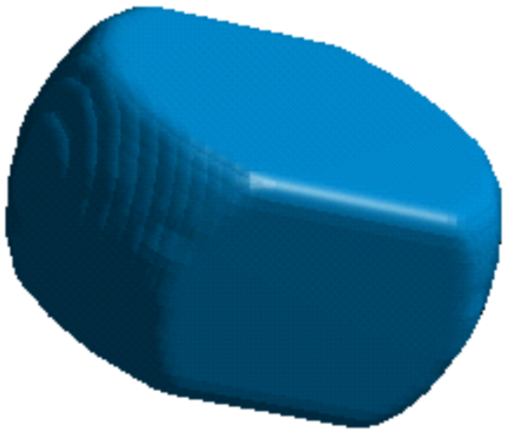




In the GAD file, the number of sub-objects is given by the value after the key **NumberOfSubObjects**.

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

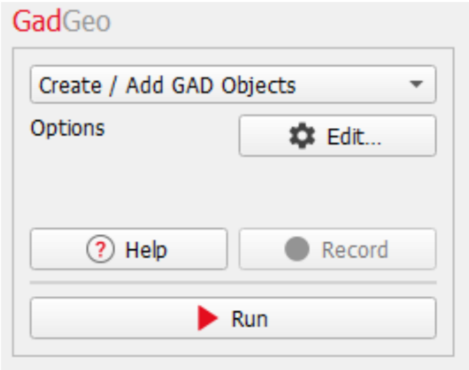

25 µm

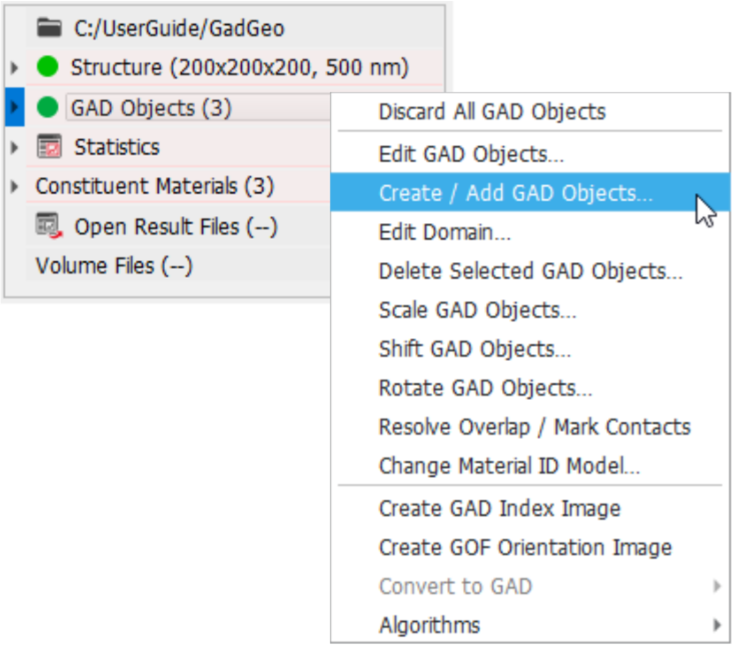


1.3 Create / Add GAD-Objects

Use the **Create / Add GAD Objects** command to add basic geometric objects to an existing structure or create a new structure by defining object type, geometric parameters, and material. New object shapes can be generated as combined objects.

The command is available from the GadGeo module panel or alternatively from the **GAD Objects** context menu of the **Project Status** section.

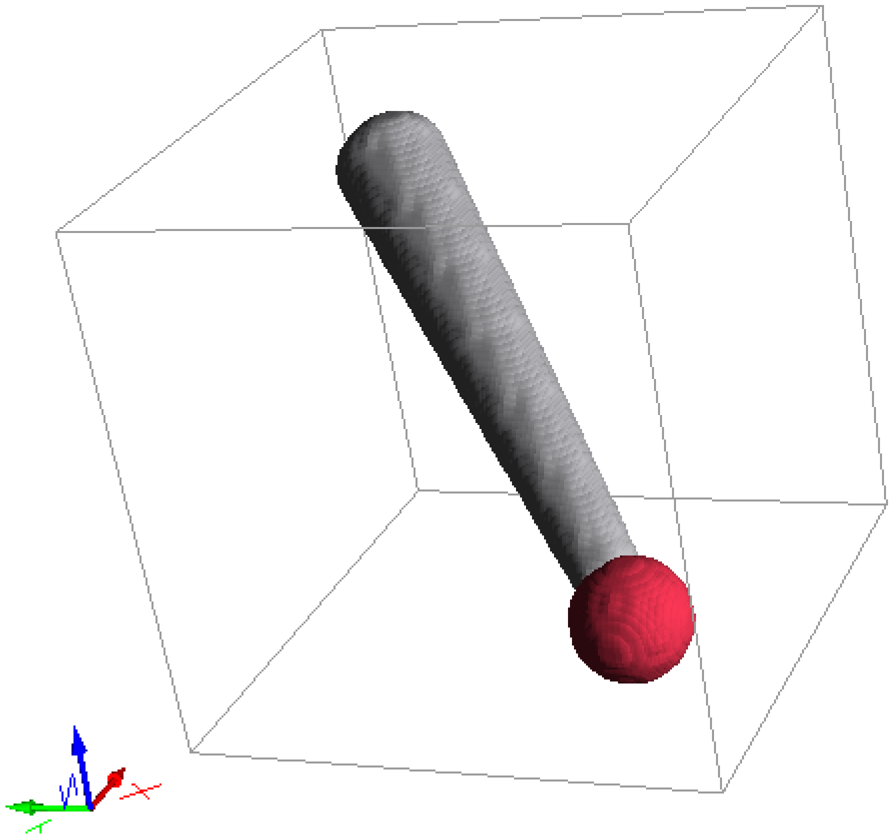




Clicking the **Options**' **Edit...** button or choosing **Create / Add GAD Objects...** in the context menu opens the **Create/Add GAD Options** dialog.

The dialog contains the Create / Add GAD Objects tab 32, to select objects and their parameters, and the General Options tab 34, to set the size and domain parameters of the resulting structure.

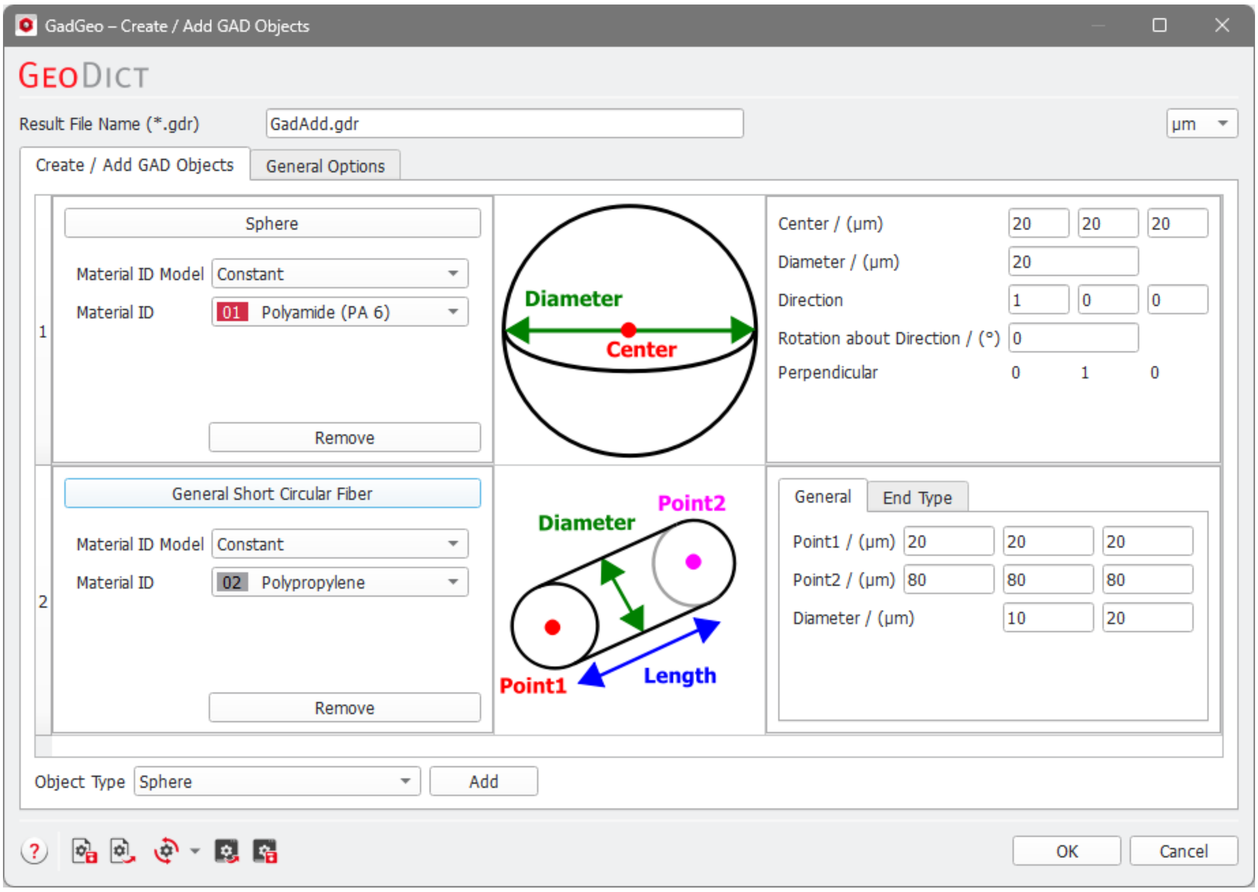
Back in the GadGeo section, clicking **Run** with the default values produces an object that is the combination of a sphere and a general short circular fiber, resembling a matchstick.



You can use this command also to create user-defined object types 36.

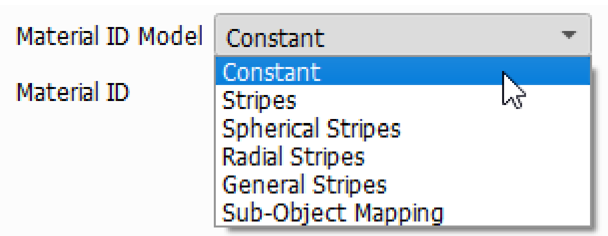
1.3.1 Create / Add GAD Objects Tab

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

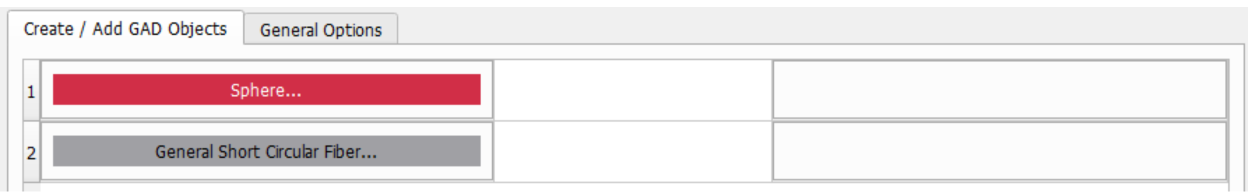


The left panel displays the number assigned to the object (**1**, **2**, **3**...), the Object Type 9 (**Sphere**, **General Short Circular Fiber**, ...), and the object **Material ID** and **Material ID Model**. It is not possible to set the material itself here, but you may assign materials to a given 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 48.



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.



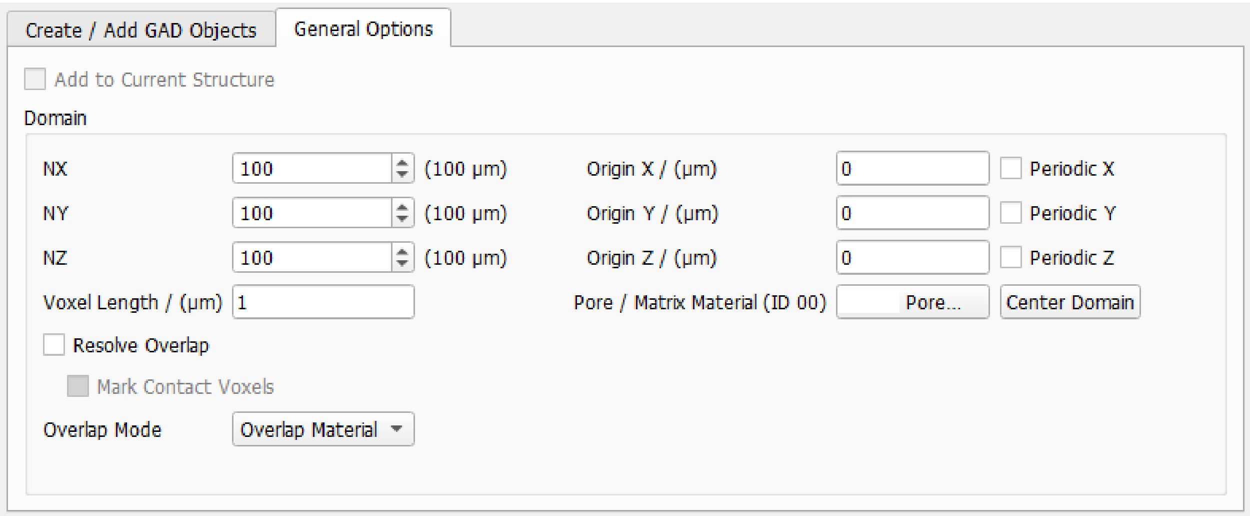
Remove can be used to eliminate a certain object from the list and from the generation.

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, the **Object Type** pull-down menu and the **Add** button can be used to add the selected object type to the list of the structure's basic objects during the generation. The available object types are explained in detail here 9.

1.3.2 General Options Tab

Check **Add to Current Structure** to add the newly defined objects to the current structure. In this case, all existing GAD objects will be kept and the domain parameters are already fixed and no longer editable. Uncheck this box to create a new 3D structure that contains only the objects defined in this dialog.



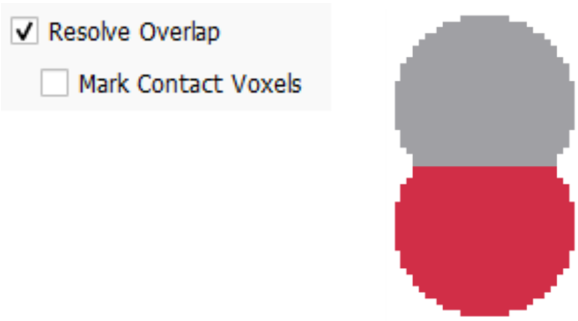
NX, **NY**, and **NZ** define the number of voxels in X-, Y-, and Z-direction, and **Voxel Length** establishes the voxel size. **Origin X**, **Origin Y**, and **Origin Z** control the Cartesian origin of the structure, and checking **Periodic X**, **Periodic Y**, and/or **Periodic Z** allows generating 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.

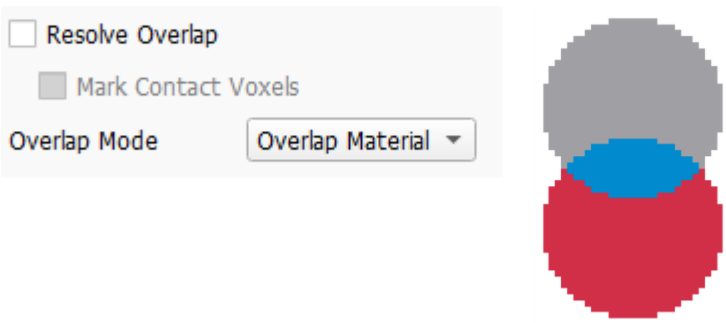
If **Resolve Overlap** is chosen, the overlap is resolved after the structure is generated. Every voxel in the overlap area is assigned to one of the objects. This assignment is done based on a watershed algorithm on the overlap area.



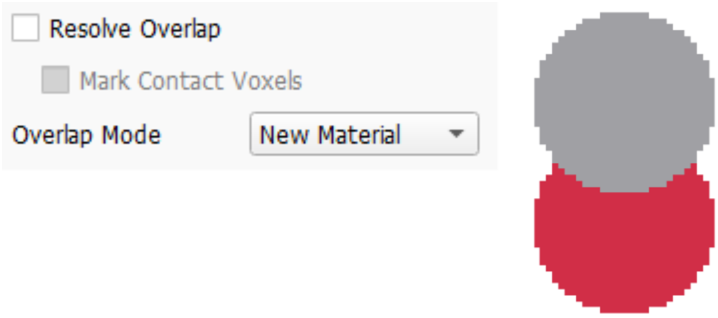
If additionally, **Mark Contact Voxels** is checked, a **Contact Voxels Material ID** must be selected. Voxels on the boundary between two objects are then given this Material ID.

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.

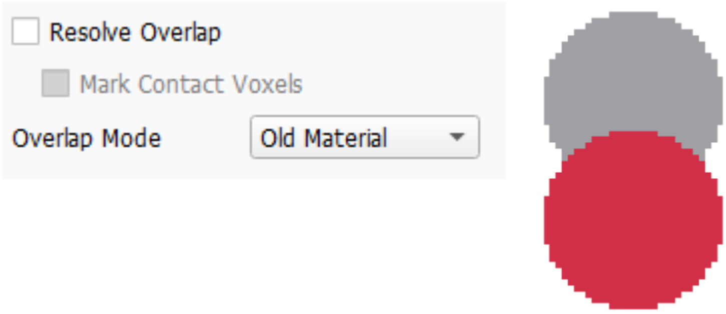
The overlap areas can be assigned to be the default **Overlap Material**. The ID of the overlap area is obtained by the bit-wise sum of the IDs of the overlapping materials.



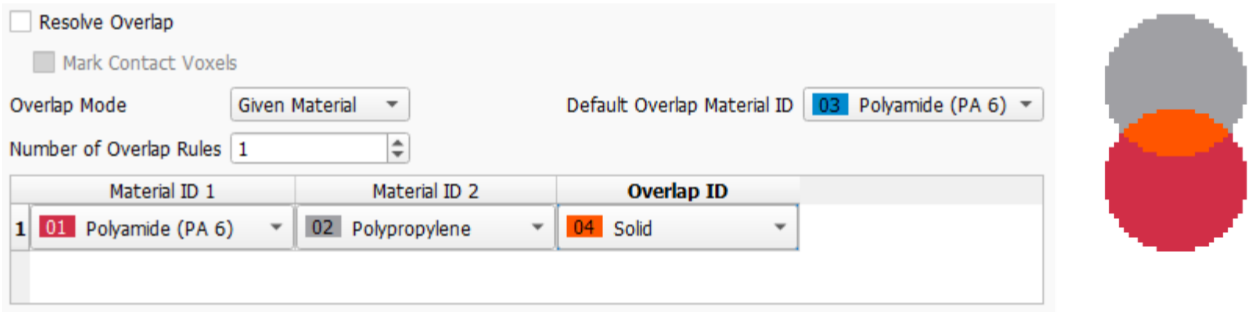
The overlap can take the ID of the **New Material**. Objects are added in the order as listed under the Create/Add GAD Options tab. This means that in this case, the overlap area is of the same material as the object that was added later.



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.



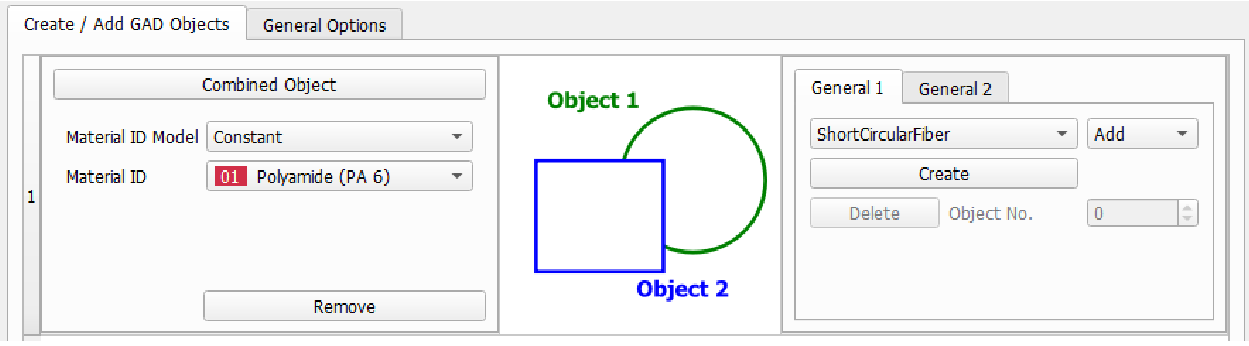
With **Given Material**, you can freely select a material from the GeoDict Material Database. Select a **Default Overlap Material ID**, and define an arbitrary **Number of Overlap Rules**. The default Material ID is used, whenever none of your user defined overlap rules applies. In the case shown below, overlap between Material ID 01 and Material ID 02 will be assigned to Material ID 04. Any other overlap, e.g., between Material ID 01 and Material ID 01 will be assigned to Material ID 03. The objects are added in the order defined in the Create / Add GAD Objects tab 32 and the overlap rules are applied after each added object. Thus, in locations where three or more objects overlap, the resulting Material ID may depend on the order in which the objects are added and the overlap rules are applied.



1.3.3 Creating User-Defined Object Types

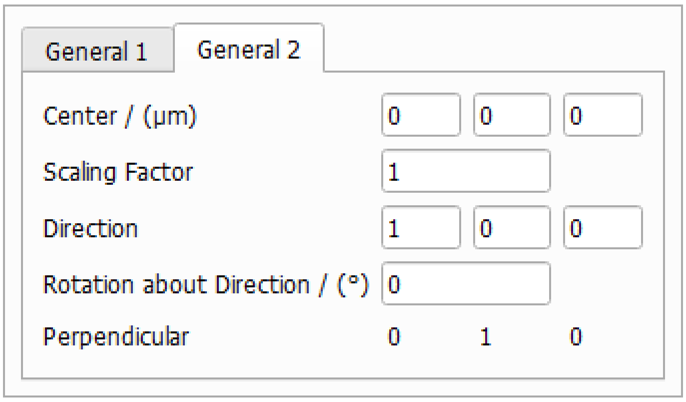
Select the object type CombinedObject to define a user-defined object type that can then be used, e.g., in GrainGeo alternatively to the built-in object types when creating a 3D structure model. For this, create a single CombinedObject with the GadGeo -Create/Add GAD Objects command and save it as a .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.

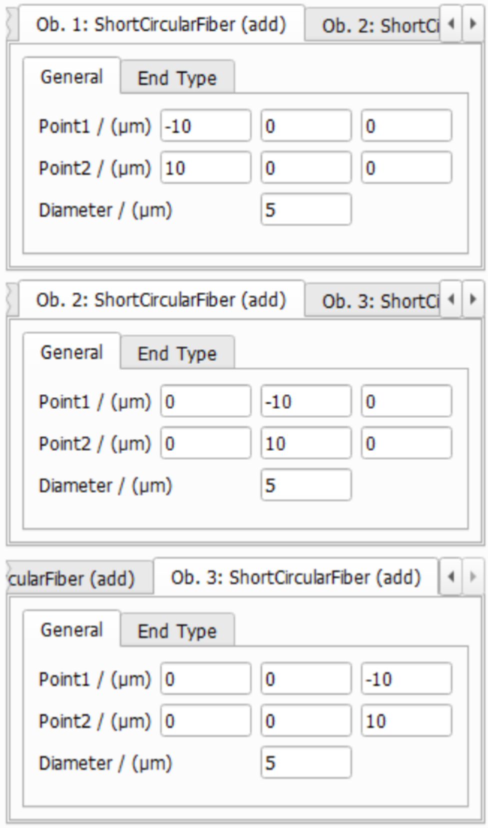


Center and **Direction** will also be used to describe the position of the new object when using it in a structure generator like GrainGeo, or inside the GAD file. Therefore, when creating a new object type in the default coordinate system, the object should be defined such that its center of mass lies 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 here, three cylinders form a 3D cross as a new object type:

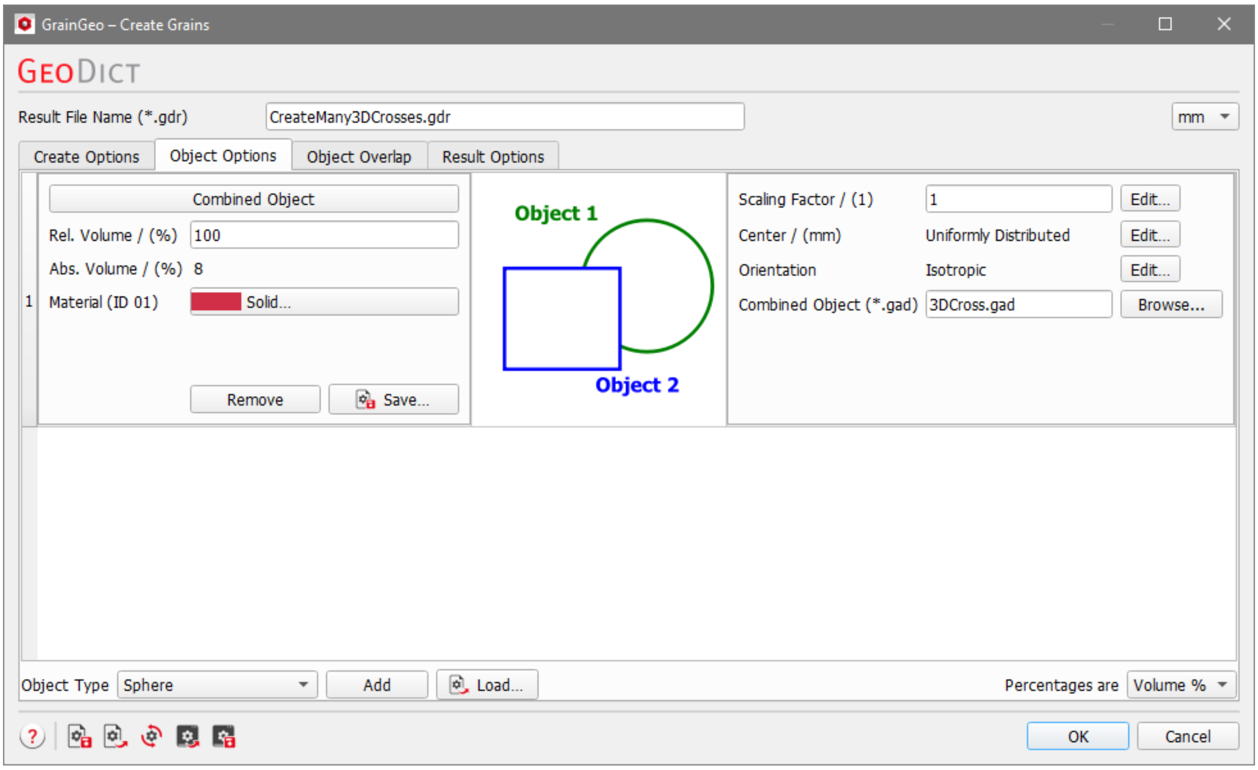
```
"Object1" : {
  "MaterialIDModel"          : "Constant",
  "MaterialID"               : 1,
  "Type"                     : "CombinedObject",
  "NumberOfSubObjects"       : 3,
  "Position"                 : [0,0,0],
  "ScalingFactor"            : 1,
  "Axis1"                    : [1,0,0],
  "Axis2"                    : [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
  }
},
```



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.

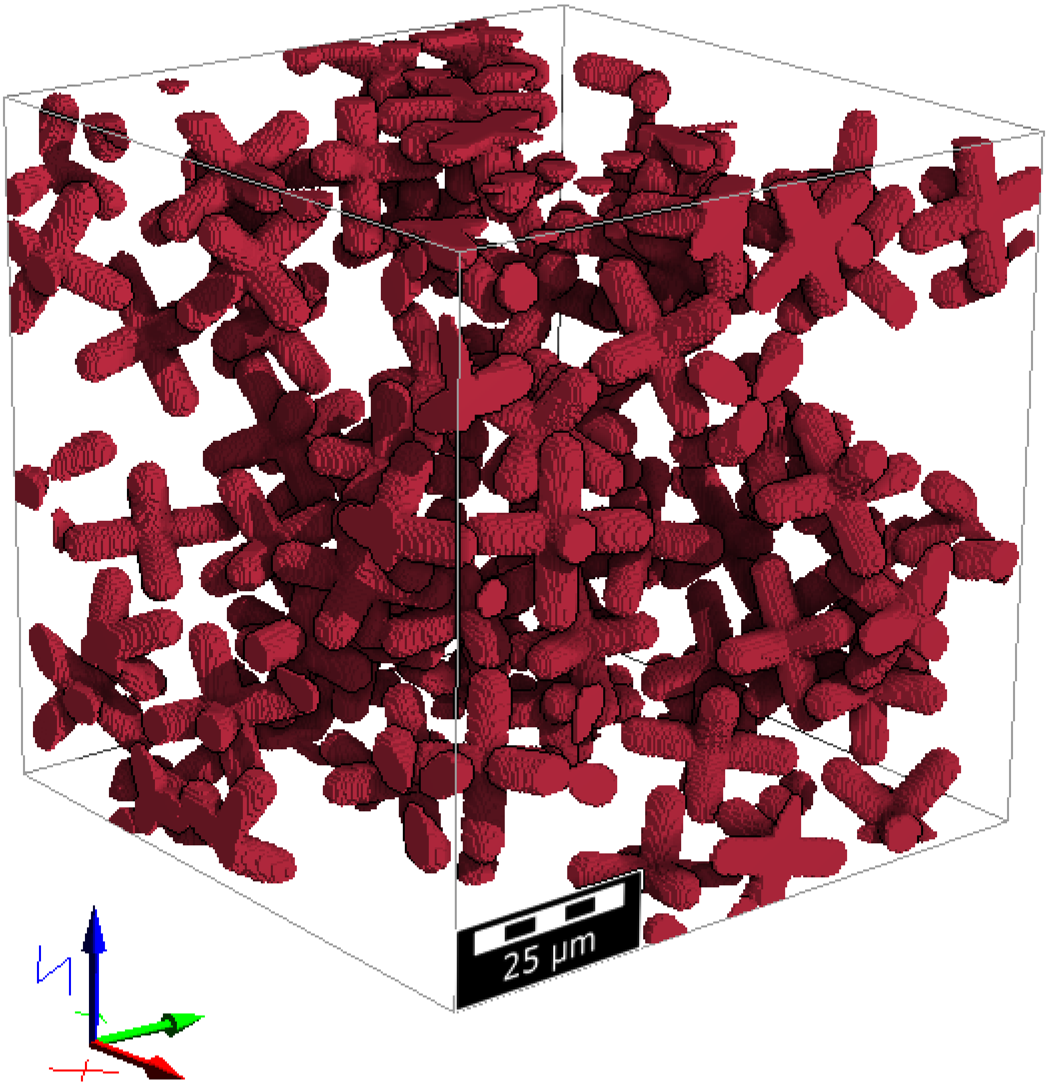
After the parameters have been entered, close the dialog and create the structure. Then, save the structure as *.gad file using **File → Save Structure As…** from the main menu. To be able to use the file later as user-defined combined object, it is important that only a single **CombinedObject** is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



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.

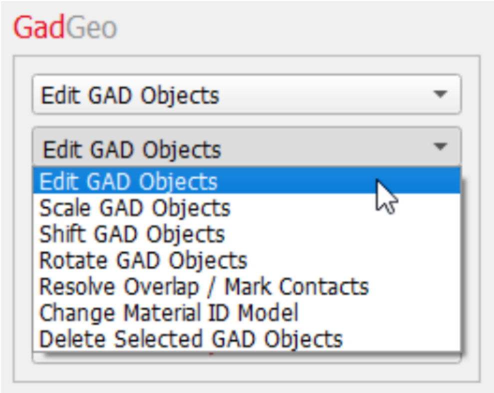


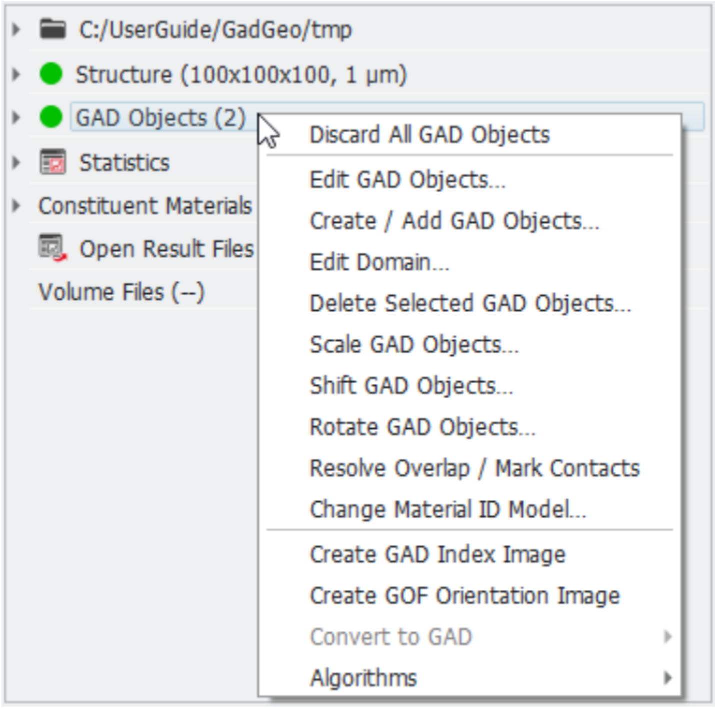
Closing the dialog and clicking **Generate** creates a structure consisting of the user defined type. Note that, although a CombinedObject consists of several sub-objects, it is not possible to select different materials or Material IDs for the different parts here. However, it is possible to change that in a post-processing step using the Change Material ID Model command 48.

1.4 Edit GAD Objects

Objects of previously generated structure models in GAD format can be edited after **Edit GAD Objects** is selected from the upper pull-down menu in the GadGeo section.

Seven commands are selectable in the second pull-down menu in the GadGeo section. The same commands are also available from the **GAD Objects'** context menu in the Project Status section located above:





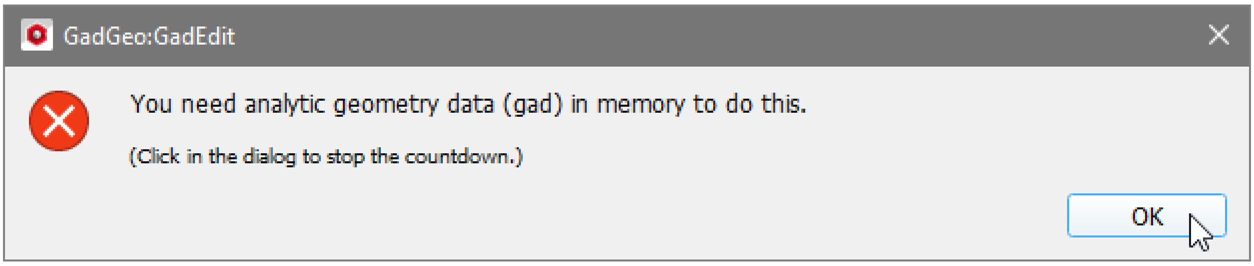
Click the **Edit...** button to set the options for this change. After setting the options, return to the GadGeo section and click **Run** to make the changes effective.

Important! The **Edit GAD Objects** commands modify the current structure and no result files are saved. Select **File** → **Save Structure As...**, in the menu bar, to save the modified structures in GAD format.

The **Edit GAD Objects** commands can only be used if the structure's objects are analytic and their parameters are known to GeoDict. This is indicated by a green dot in front of **GAD Objects** in the **Project Status** section. If a red dot is shown, no analytic information about the objects is available, and the structure is only defined through its voxel grid.

Structure models obtained from segmented 3D images are typically not analytic, and no GAD data is available. Similar, commands like **GrainGeo - Grow Sediment** or **GrainGeo - Add Binder** create non-analytic 3D structures.

In this case, when you try 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.

Commands available in the Edit GAD Objects menu

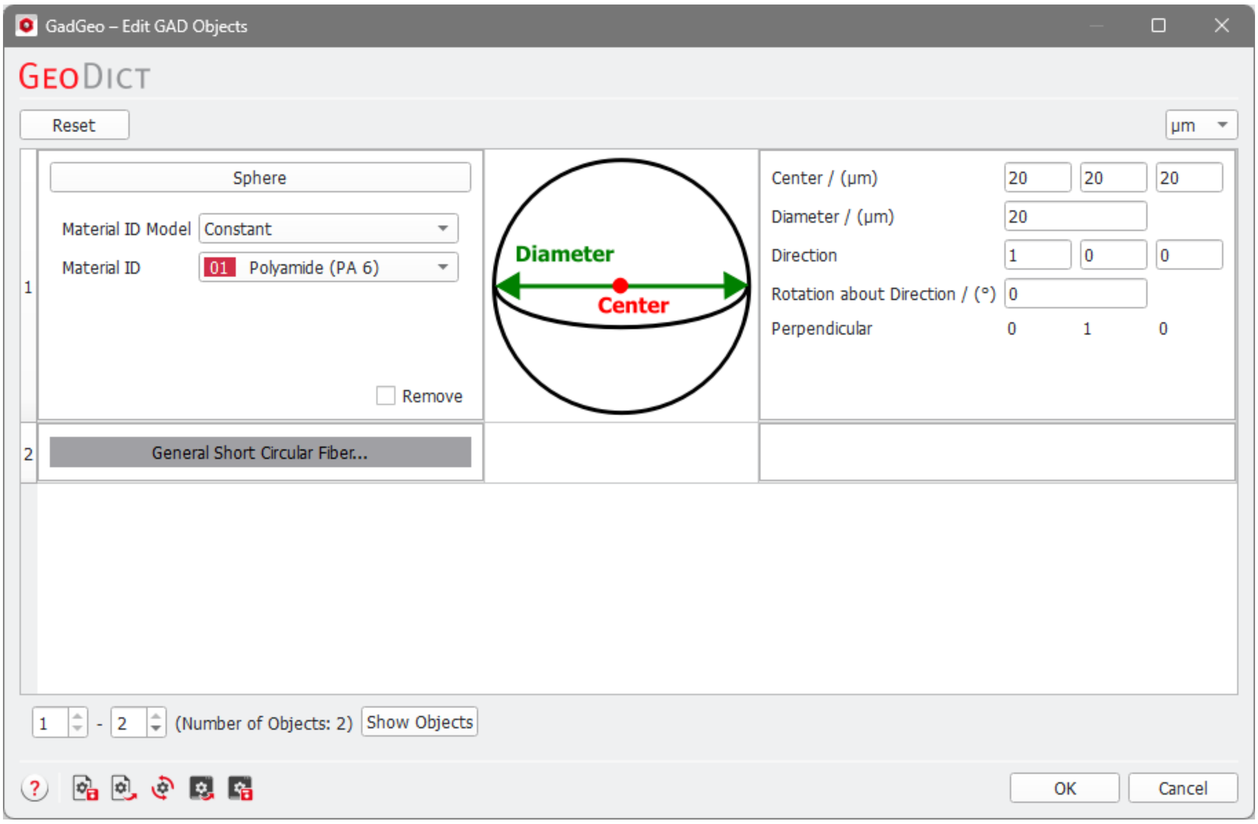
- Edit GAD Objects 42
  Modify object parameters (size, material) of a specific GAD object from a list of all existing objects.
- Scale GAD Objects 44
  Shrink or enlarge all GAD objects by a given factor.
- Shift GAD Objects 45
  Shift all GAD objects in a given direction.
- Rotate GAD Objects 46
  Rotate the coordinate system about given axes.
- Resolve Overlap / Mark Contacts 47
  Change the overlap mode of GAD objects into the Resolve Overlap mode and mark contact areas.
- Change Material ID Model 48
  Create objects that consist of different Material IDs.
- Delete Selected GAD Objects 51
  Remove all GAD objects with a given Material ID, material or object type.

1.4.1 Edit GAD Objects

The **Edit GAD Objects** command allows to modify the current 3D structure by changing the geometrical parameters of each individual object.

Click the **Edit...** button to open the **Edit GAD Objects** dialog. In it, every single object in the structure is listed with all geometrical properties.

Click on the **Sphere...** (or any other object type) button to open the corresponding widget and see all details:

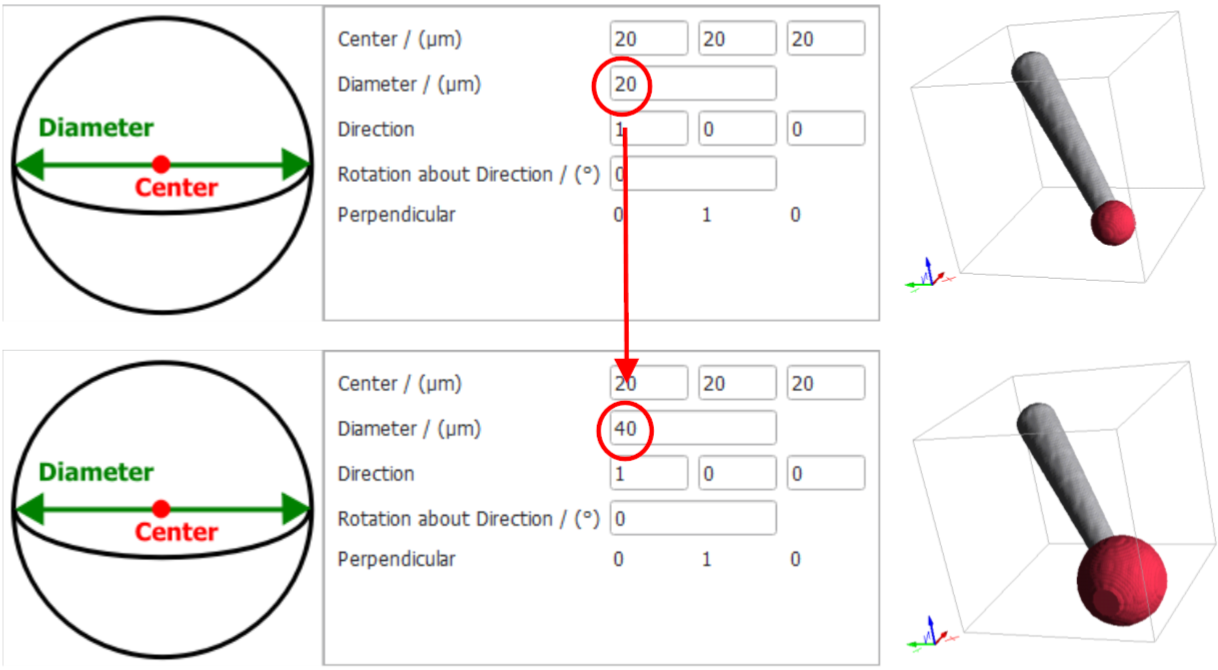


The dialog gives the choice to **Remove** single objects altogether, to change the Material ID assigned to any of them, as well as to modify their size, position, orientation, shape, and more (The available object types are described in detail here 9).

At the bottom left, you can decide to limit the range of objects 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, click **Run** to apply the modifications to the current structure.

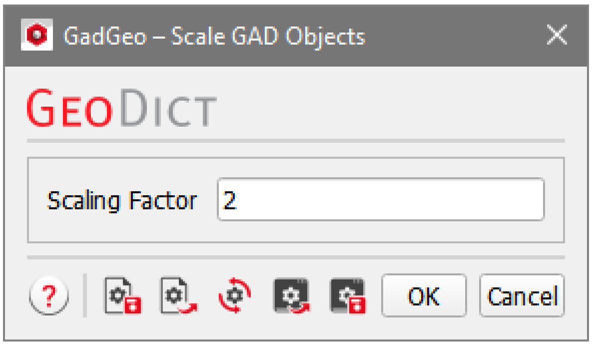
For example, observe the effect of increasing the **Diameter** of the red sphere from 20 to 40 µm in the matchstick structure.



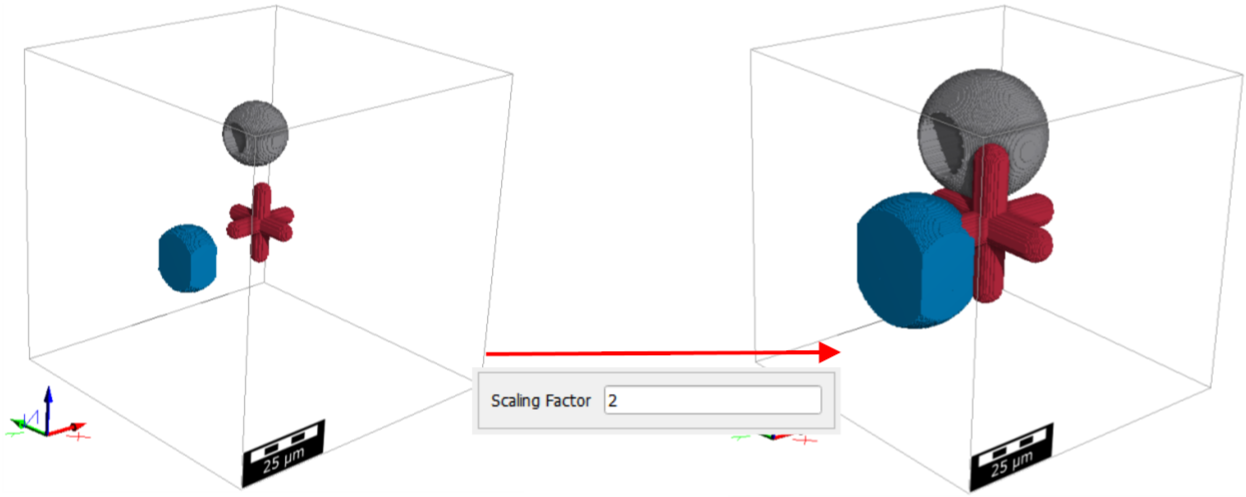
1.4.2 Scale GAD Objects

The **Scale GAD Objects** command allows to modify the current 3D structure by changing the size of each individual object by a fixed scaling factor.

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



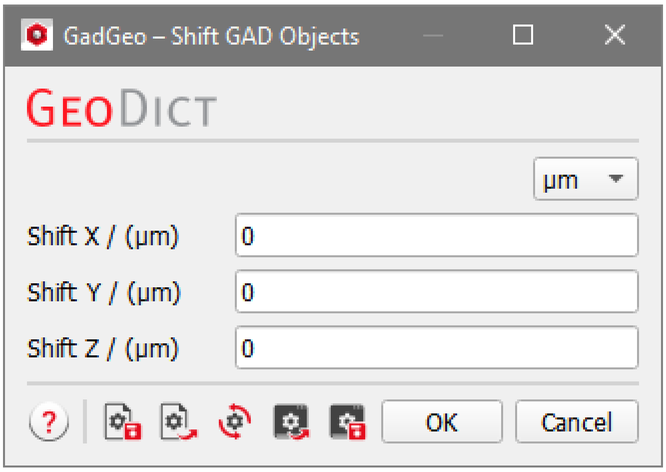
Enter the desired **Scaling Factor** for the objects in the structure and click **OK** to return to the **GadGeo** section. Clicking **Run** changes the size of the objects by the given scaling factor, but not the size of the structure's domain or its resolution.



1.4.3 Shift GAD Objects

The **Shift GAD Objects** command allows to modify the current 3D structure by shifting each individual object in a fixed direction.

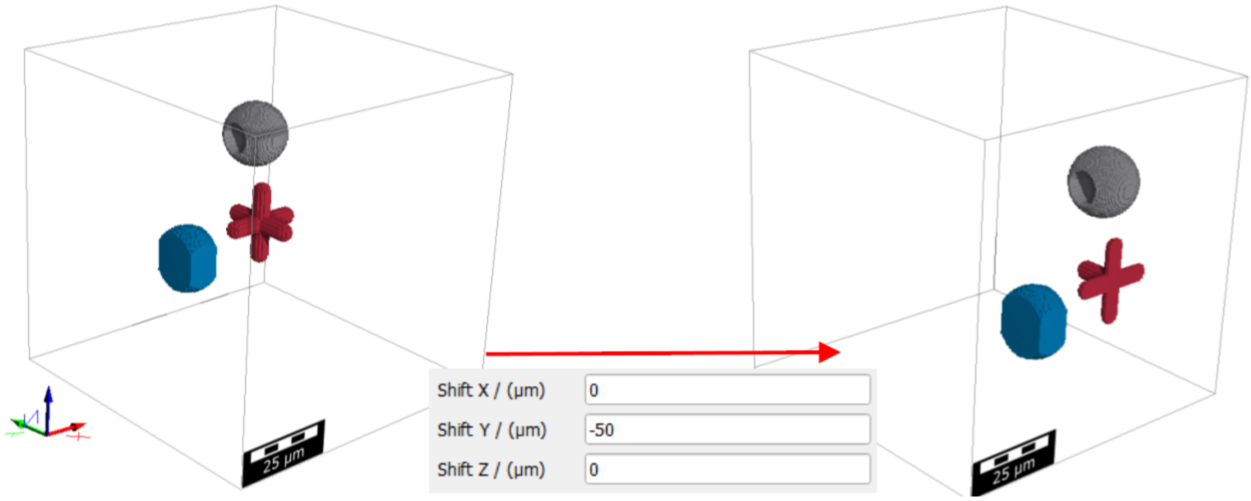
After selecting **Shift GAD Objects** from the pull-down menu, and clicking the **Edit...** button, the **Shift GAD Objects** dialog opens.



At the top right, you can change the unit.

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. Click **Run** to move the objects in the structure away from the Cartesian axis origin by the given distance.

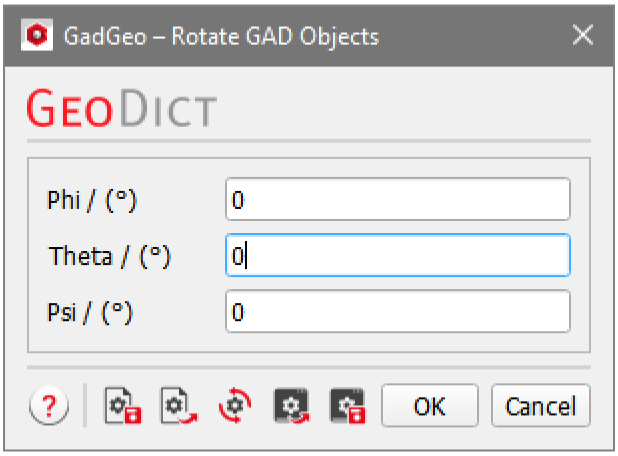
Observe the shift of the structure by 50 µm in the negative Y-direction. Objects may be (partially) shifted outside of the domain.



1.4.4 Rotate GAD Objects

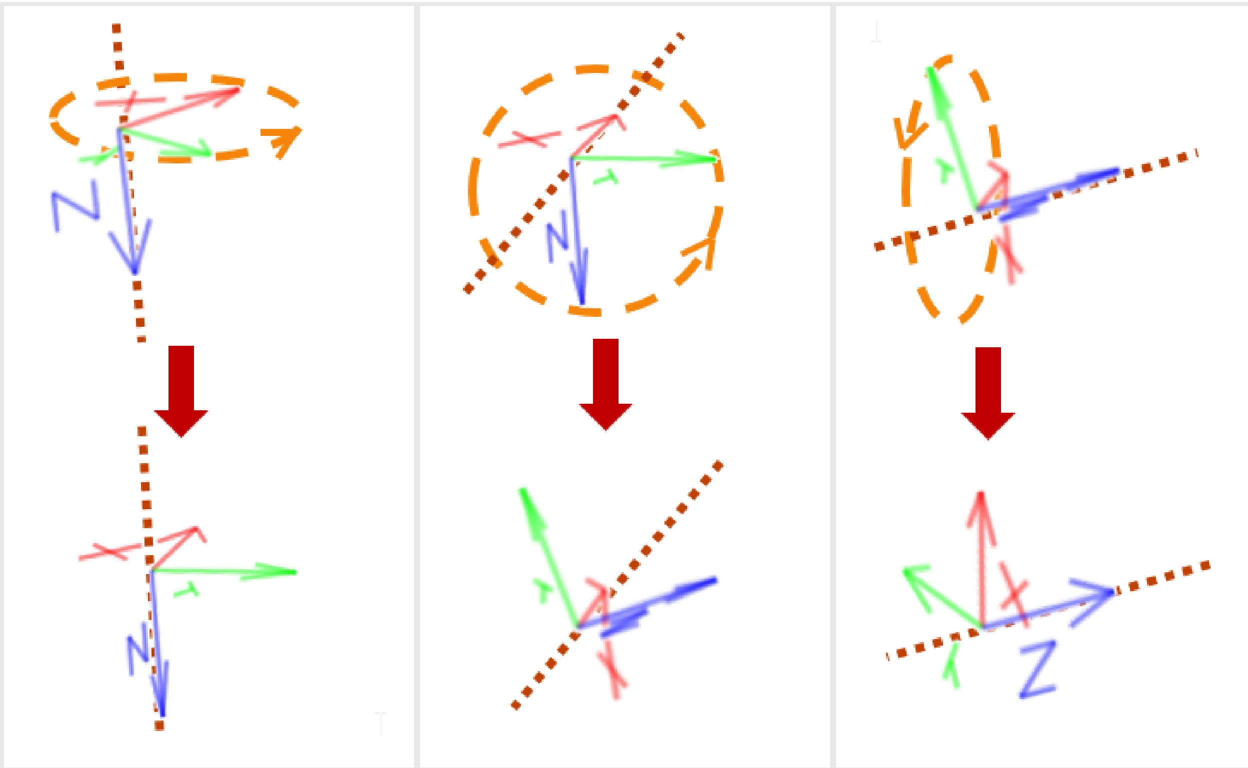
The **Rotate GAD Objects** command allows to modify the current 3D structure by rotating each individual object around the domain center. The rotation is defined through it's Euler angles.

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

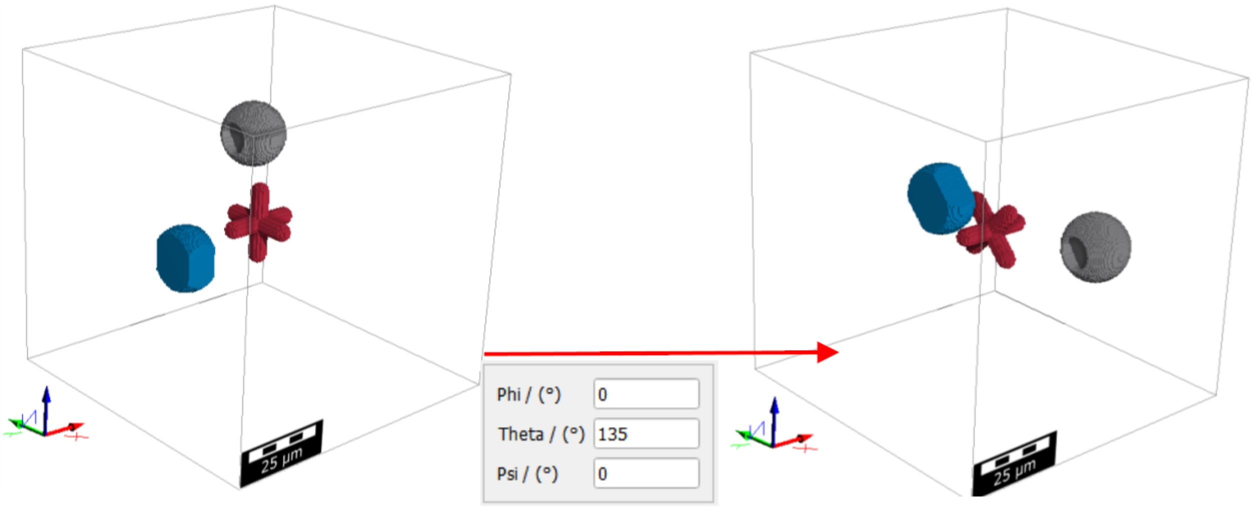


Enter the Euler angles **Psi**, **Theta**, and **Phi** by which the objects in the structure should rotate and click **OK** to return to the GadGeo section. Click **Run** to rotate the objects in the structure by the given rotation angles.

z-axis fixed	new x-axis fixed	new z-axis fixed
Phi rotates about the existing z-axis.	Theta rotates about the new x-axis.	Psi rotates about the new z-axis



Observe the turning of the structure by 135° about the (new) X-axis (Theta). No rotation was previously applied around the (old) Z-axis (Phi). The origin of the axes is located in the center of the domain.



1.4.5 Resolve Overlap / Mark Contacts

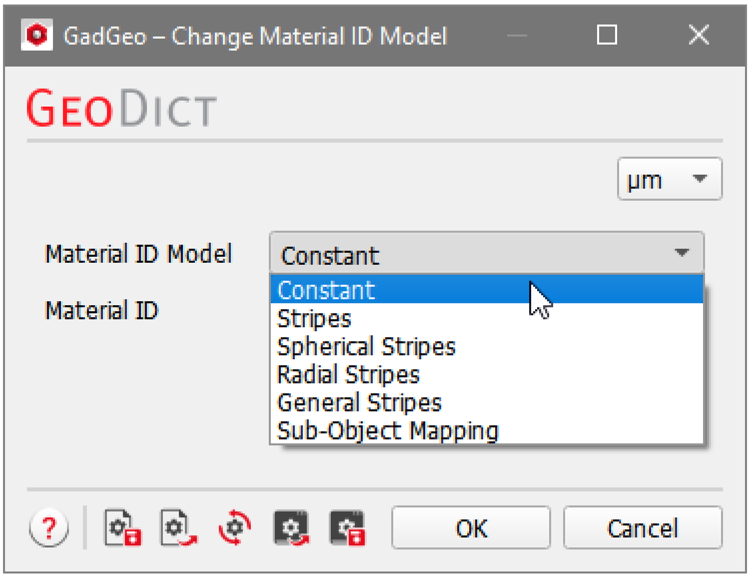
When running the **Resolve Overlap / Mark Contacts** command, GeoDict changes the overlap behavior 34 as if **Resolve Overlap** and **Mark Contact Voxels** had been selected upon creation of the structure. This command has no parameters, GeoDict automatically chooses a free Material ID as **Contact Voxel Material ID**.

The same effect can also be achieved using the Edit Domain 55 command.

1.4.6 Change Material ID Model

With the **Change Material ID Model** command it is possible to modify the current GAD objects such that each object may consist of multiple Material IDs.

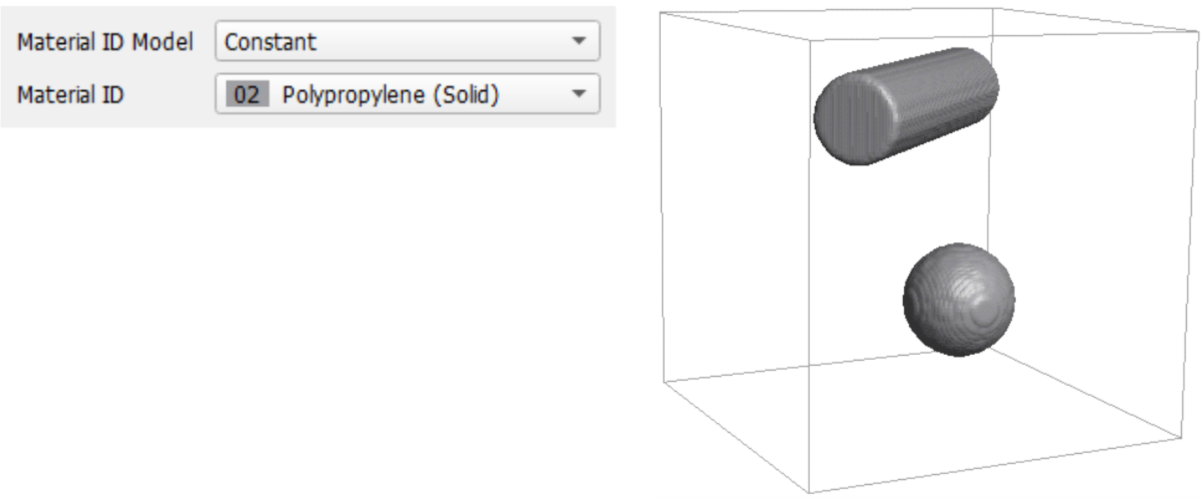
After selecting **Change Material ID Model** from the pull-down menu, and clicking the **Edit...** button, the **Change Material ID Model** dialog opens.



Choose between six different Material ID models. The chosen model will be applied to all GAD objects when the **Run** button is clicked.

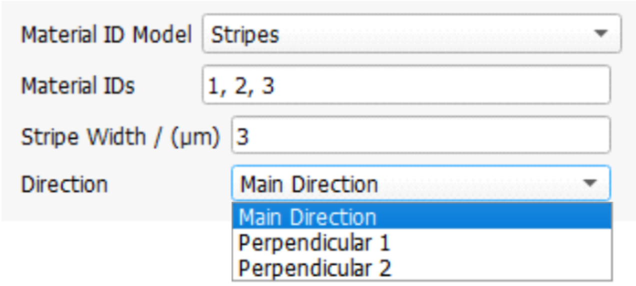
Constant

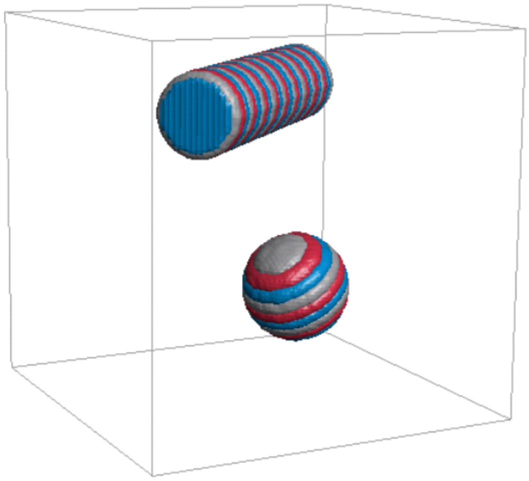
With the **Material ID Model** set at the default **Constant**, each object is assigned the single selected Material ID.



Stripes

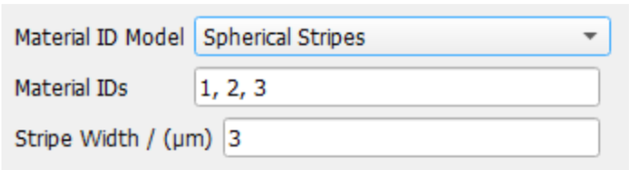
Stripes creates stripes of different materials in a direction given by one of the intrinsic directions of the object.

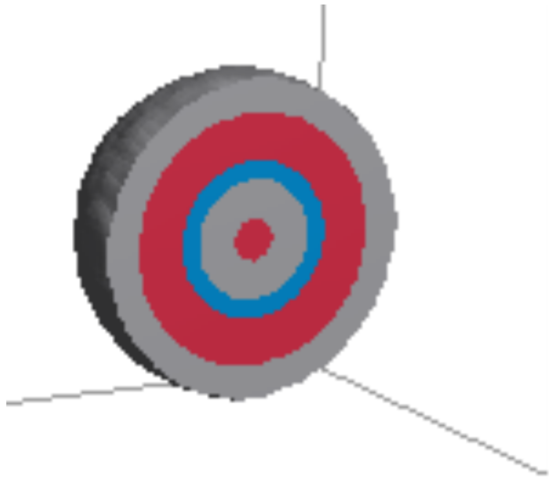




Spherical Stripes

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

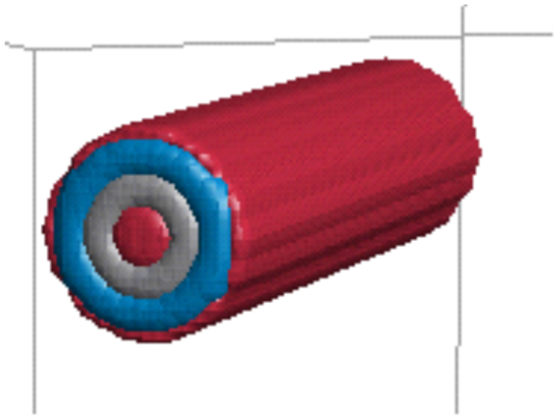




Radial Stripes

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

Material ID Model	Radial Stripes
Material IDs	1, 2, 3
Stripe Width / (µm)	3



General Stripes

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, 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.

Material ID Model	General Stripes
Material IDs	1, 2, 3
Stripe Width / (µm)	3

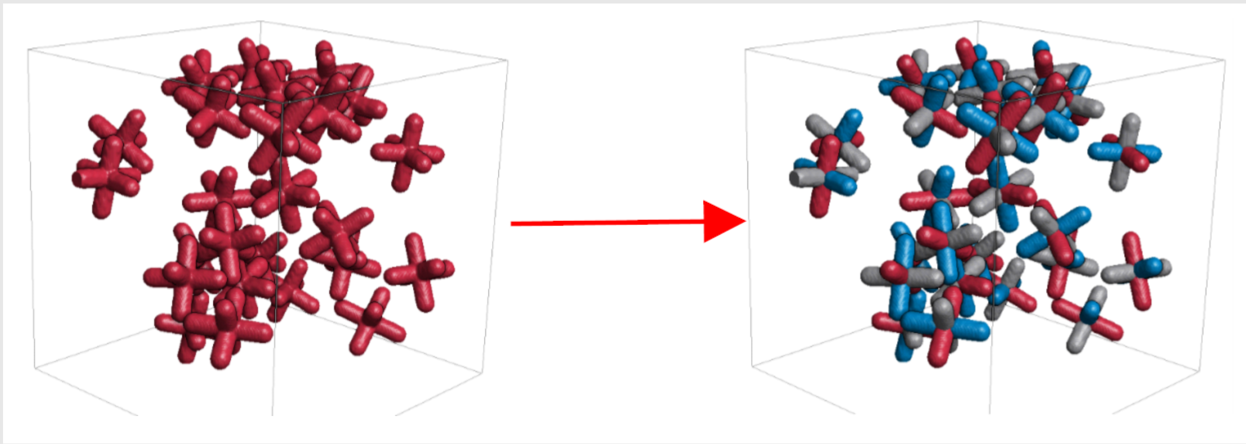
- [x] Perpendicular to Object Main Direction (Axis1)
- [x] Perpendicular to Object Perpendicular1 (Axis2)
- [x] Perpendicular to Object Perpendicular2 (Axis3)

Sub-Object Mapping

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

Material ID Model	Sub-Object Mapping
Material IDs	1, 2, 3

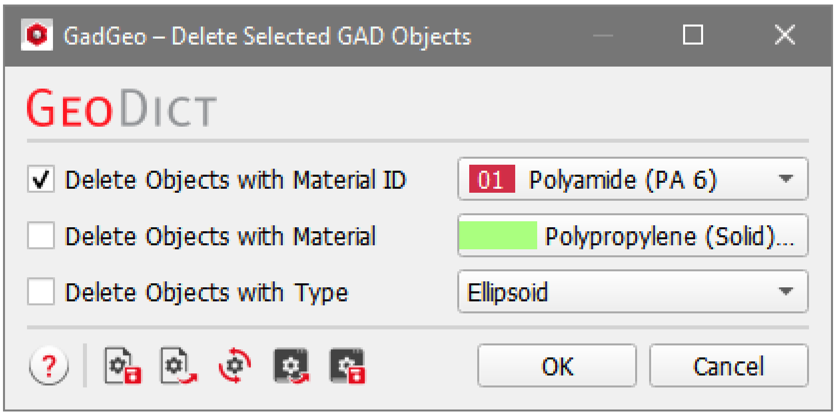
This is particularly useful for User-Defined Objects and structures created using those objects as described here 36. For the 3D cross example that consists of three **ShortCircularFiber** GAD sub-objects, every sub-object can get a different Material ID assigned.



1.4.7 Delete Selected GAD Objects

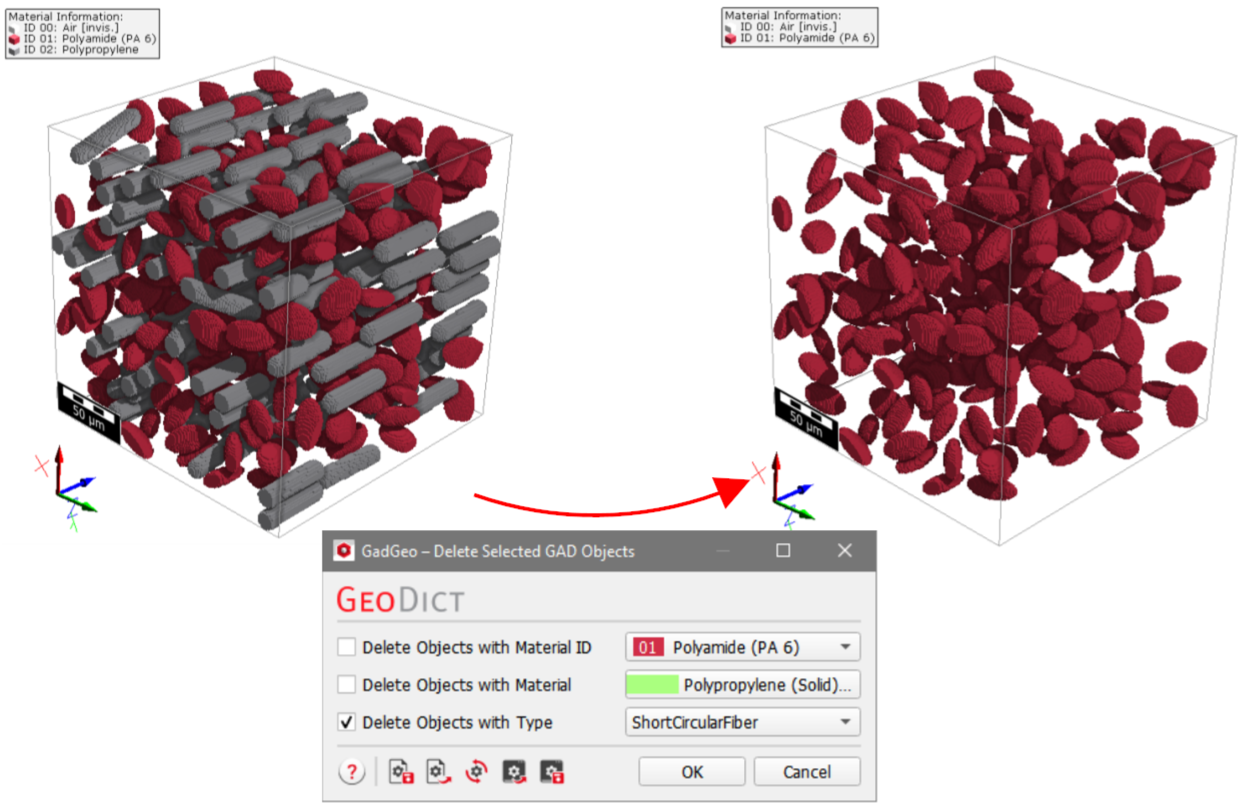
The **Delete Selected GAD Objects** command allow to remove a selection of objects from the current 3D structure. The selection can be based on material, Material ID, or on the object type.

After selecting **Delete Selected GAD Objects** 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 type **ShortCircularFiber** (here shown in gray) in the structure.


Material Information:
ID 00: Air [invis.]
ID 01: Polyamide (PA 6)
ID 02: Polypropylene
50 µm
Material Information:
ID 00: Air [invis.]
ID 01: Polyamide (PA 6)
50 µm
GadGeo – Delete Selected GAD Objects
GeoDict
Delete Objects with Material ID
01 Polyamide (PA 6)
Delete Objects with Material
Polypropylene (Solid)...
Delete Objects with Type
ShortCircularFiber
OK
Cancel

1.5 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 which is additionally available from the **GAD Objects** context menu.

> The **Edit Domain** command can only be used if the structure's objects are analytic and their parameters are known to GeoDict. This is indicated by a green dot in front of **GAD Objects** in the **Project Status** section. If a red dot is shown, no analytic information about the objects is available, and the domain cannot be edited.

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

```
"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
  }
},
```

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.
- **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**. Use this mode if you want 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**).

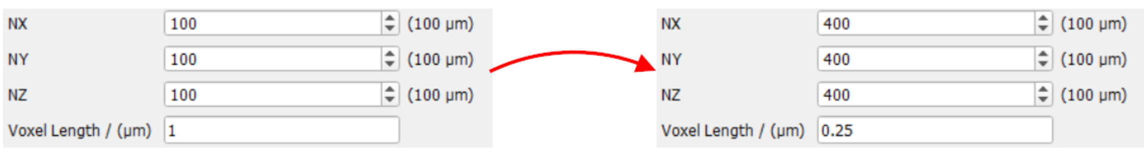
Additional to the size of the domain, the origin of the geometry is specified. A shift of the origin from (0.0, 0.0, 0.0) is given by the values after the keys **OriginX**, **OriginY**, and **OriginZ**.

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 re-enters 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:

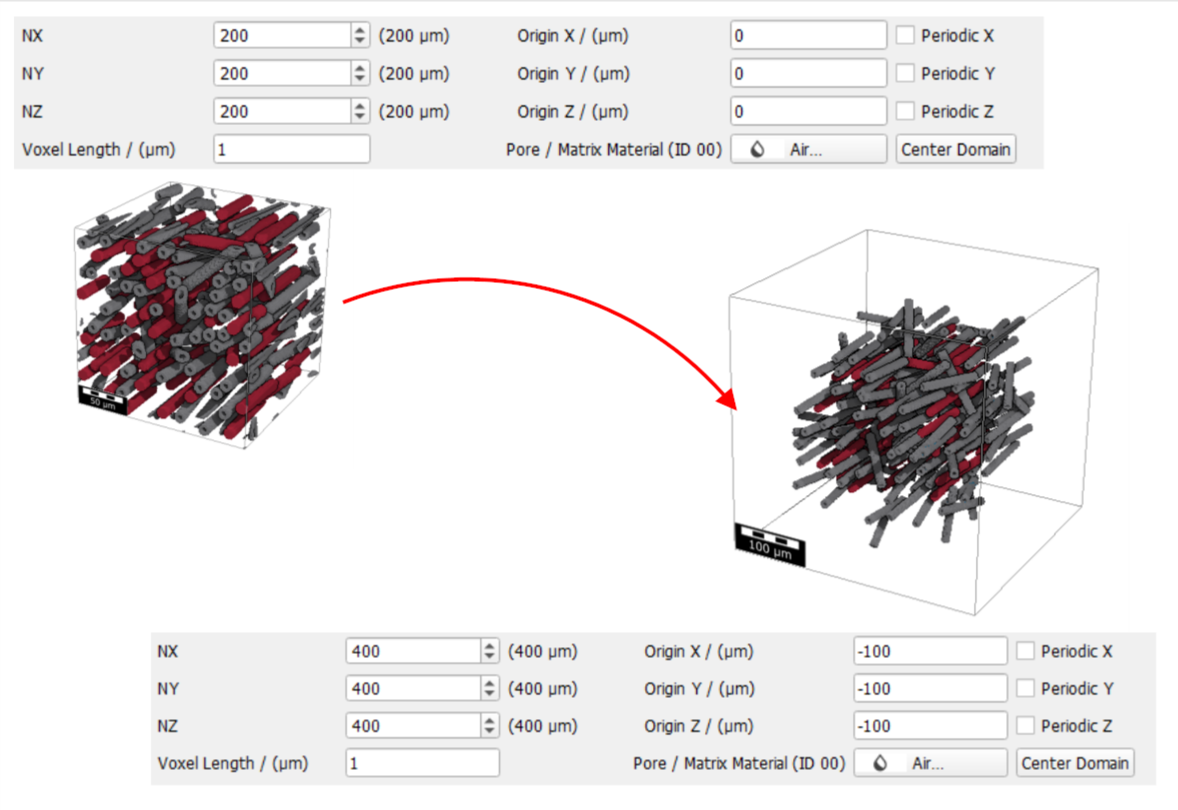
Change resolution

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

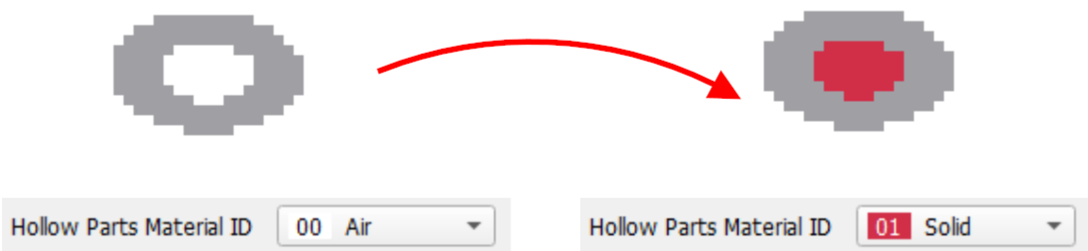
By changing the values of **NX**, **NY**, **NZ** and the position of the **Origin** it is possible to move the field of view, thus changing the cut-out used as the current 3D structure model. The **Origin** defines the position of the front lower left corner of the bounding box, which is usually (0,0,0)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



Distinguish between background and hollow parts

It is possible to reassign the **Pore / Matrix Material (ID 00)** as explained here 34.

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.

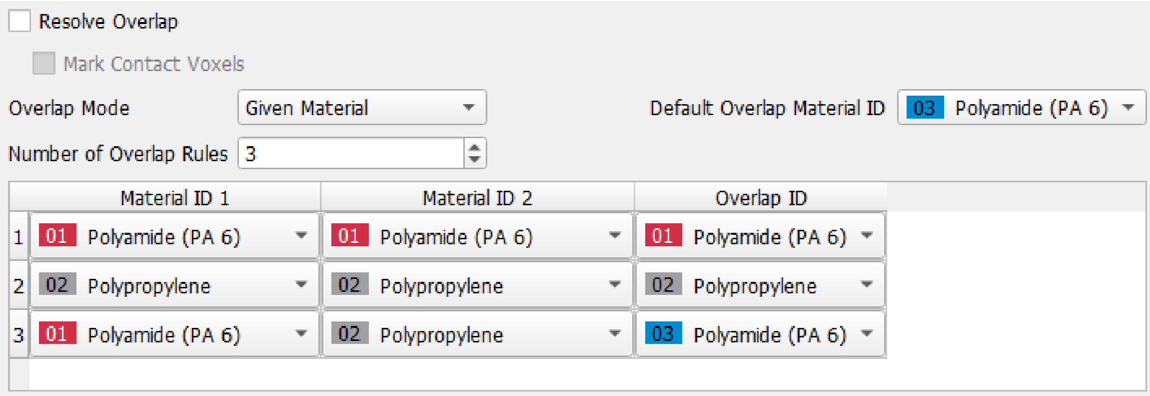


Changing the **Hollow Parts Material ID** allows to distinguish between the interior of those objects and the pore space on the outside. In a GAD file, a change of these parameters will change the parameters stored in the **Material** section and the **HollowMaterialID**.

Redefine overlap behavior

When you create a 3D structure model with one of the structure generation modules of GeoDict, you have to decide how overlapping objects are 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 here 34.

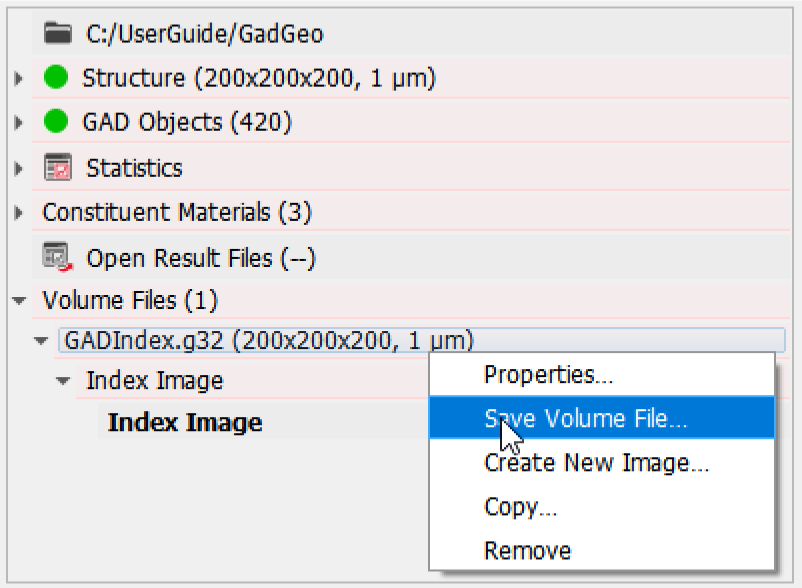
All defined overlap rules are also stored in the GAD file, in sections **OverlapRule..**. Therefore, these settings are always stored together with the structure, when it is saved either as a .gad or .gdt file.

1.6 Create GAD Index Image

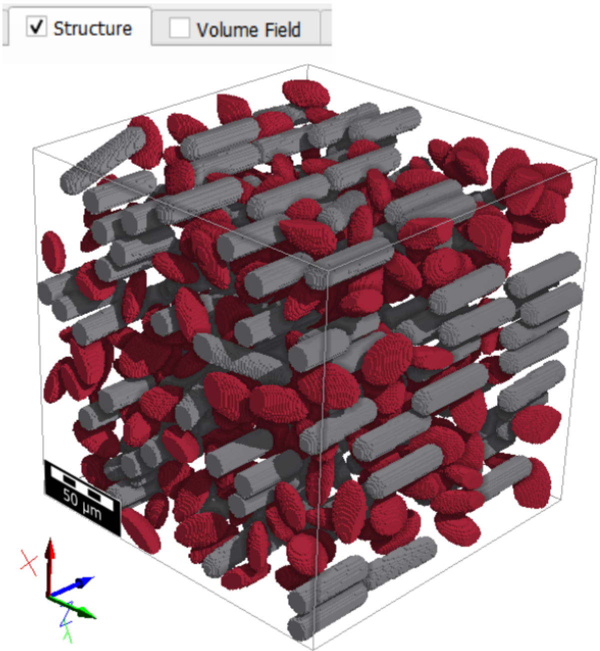
In GeoDict, each GAD object has an index number. This numbers can be accessed with the help of the **Create GAD Index Image** command. This command creates a 3D volume field, where each voxel contains a 32 bit unsigned integer value describing the index of the object that this voxel belongs to.

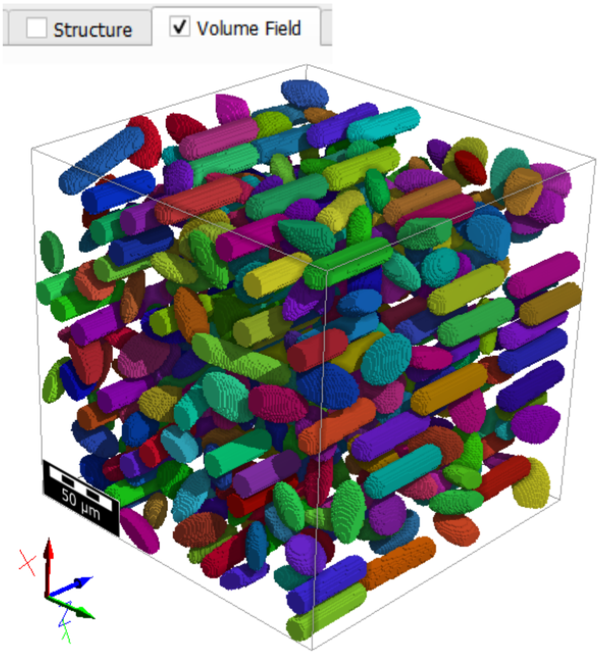
This command requires no input parameters, so there are no parameters to be set.

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 image below shows the index number field with small index numbers in blue and high index numbers in red.





1.7 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.

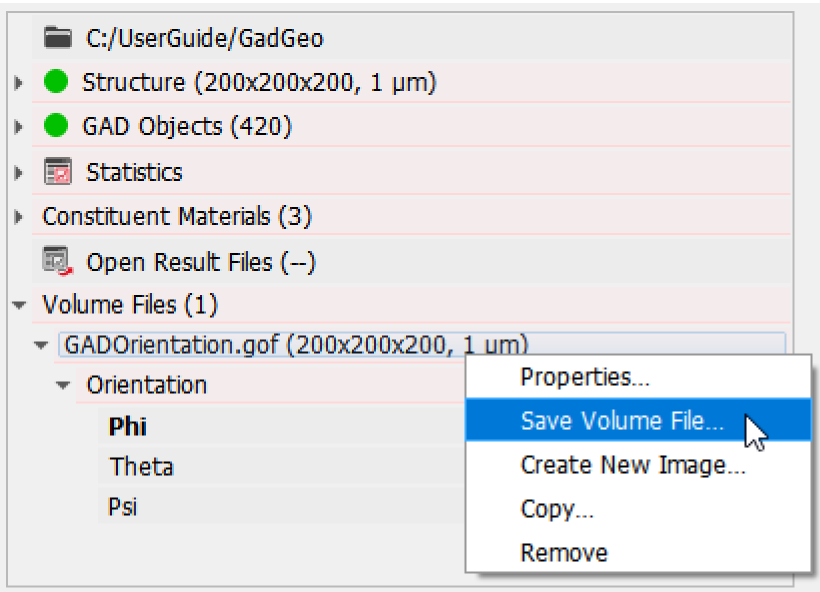
This orientation can be described in form of the Euler angles 46 required to rotate the cartesian coordinate system into the intrinsic one. Thus, for an intrinsic coordinate system described through Axis1 = (1,0,0) and Axis2 = (0,1,0), all Euler angles would be zero.

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.

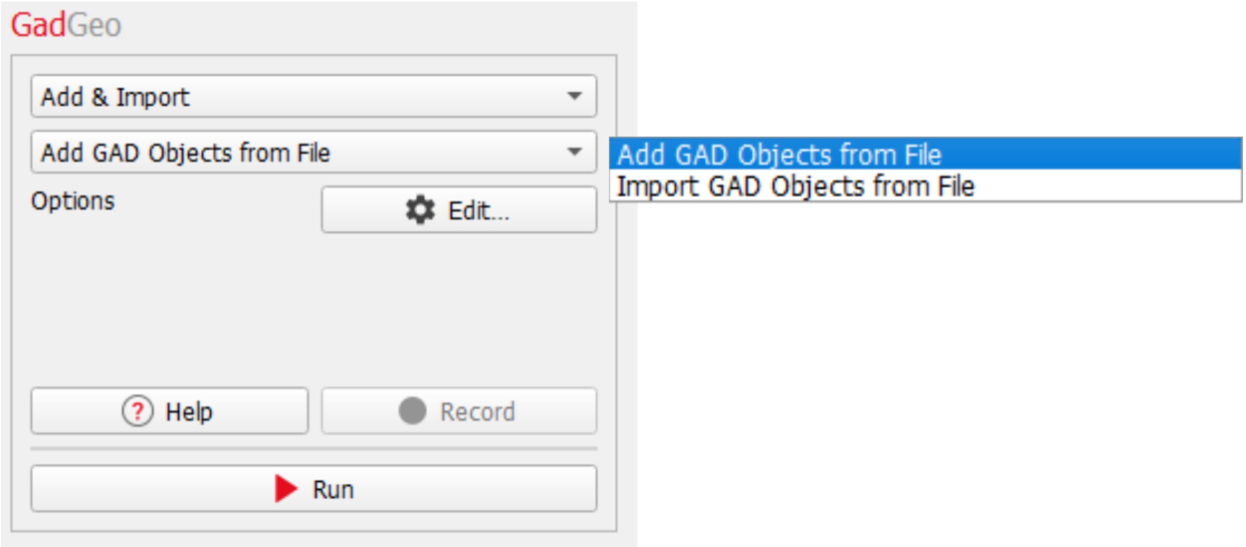
When the command is run, no (result) files are created. The created field is available in GeoDict as a volume field named GADOrientation.gof.

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



1.8 Add & Import

Two commands are available in the pulldown menu.



Add GAD Objects from File 59 allows you to combine the current structure with objects stored in another .gad file. This command is not available from the **GAD Objects**' context menu.

Import GAD Objects from File 60 allows you to import objects stored in a .gad file into a modified domain. The same effect might be achieved by loading the .gad file and then using the **Edit Domain** command afterwards. This command was available through the ImportGeo-Base module in GeoDict versions 2023 or earlier.

1.8.1 Add GAD Objects from File

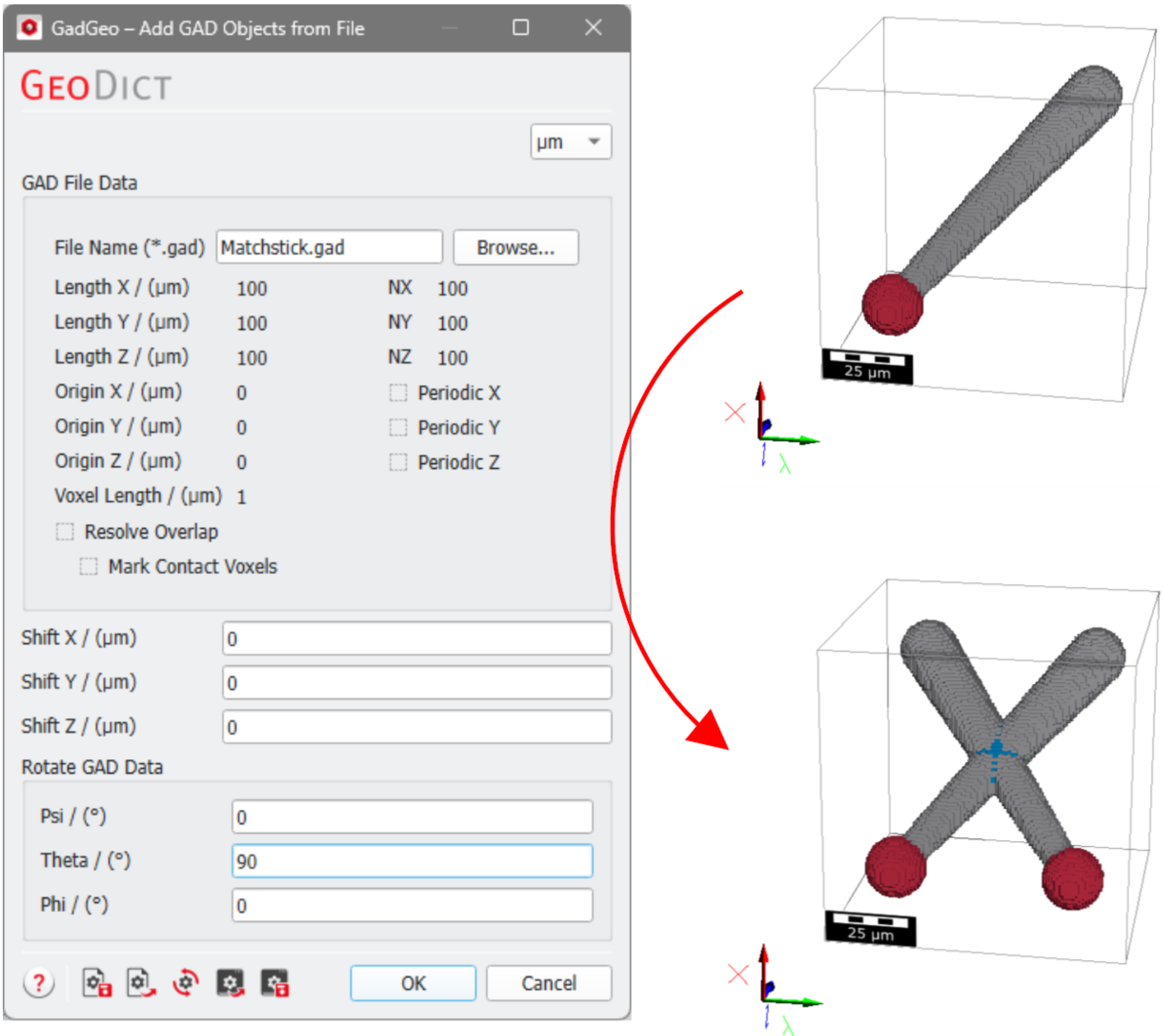
This command allows you to combine the current structure with objects stored in another .gad file.

Click the **Options**' **Edit...** button to open the **Add GAD Objects from File** dialog. Then, click the **Browse...** button to search for a file in GAD format to add.

The domain size stored in the .gad file is shown in the dialog and cannot be modified. Be aware that the current structure may have a different domain.

Before adding the new objects from the GAD file to the current structure, the new objects may be shifted 45 and rotated 46 with respect to their position stored inside the GAD file. It is not possible to use this command to load a .gad file, it can only be used to add objects to the current structure.

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

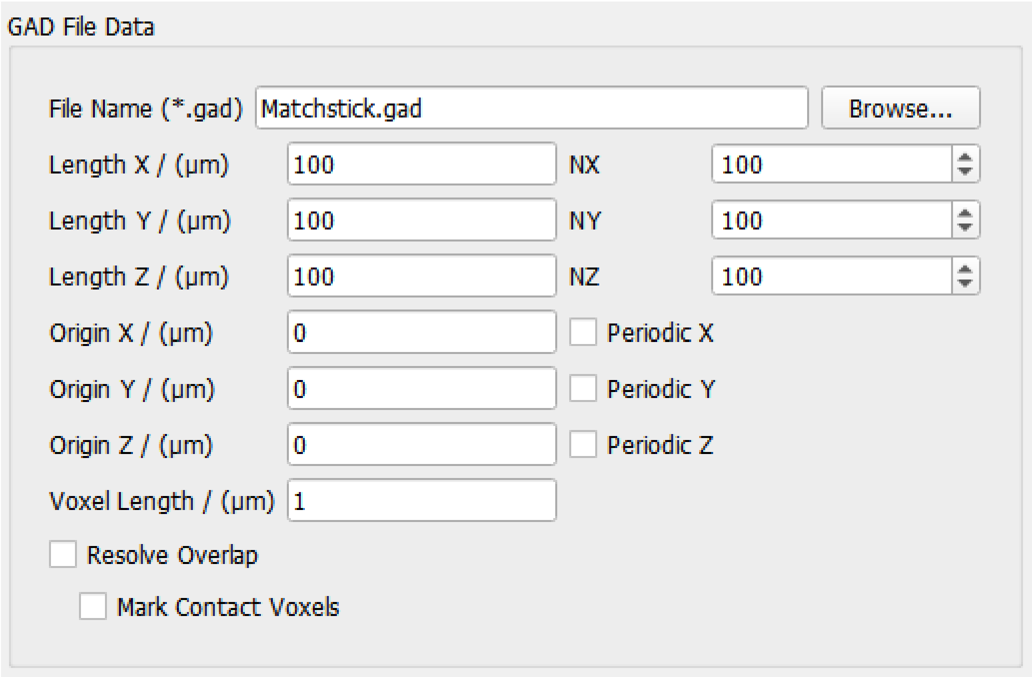


1.8.2 Import GAD Objects from File

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 ( ). This way, the .gad file is imported with the domain options specified in the **Domain** parameters (see the Edit Domain 53 command).

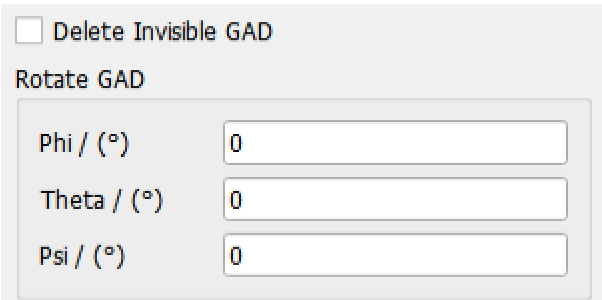
However, to load the .gad file with values different to those specified in the **Domain**, parameters, the **Import GAD Objects from File** command may be used.

Click the **Options**' **Edit...** button to open the **Import GAD Objects from File** dialog. **Browse...** for the .gad file to import. 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 here 34. 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.

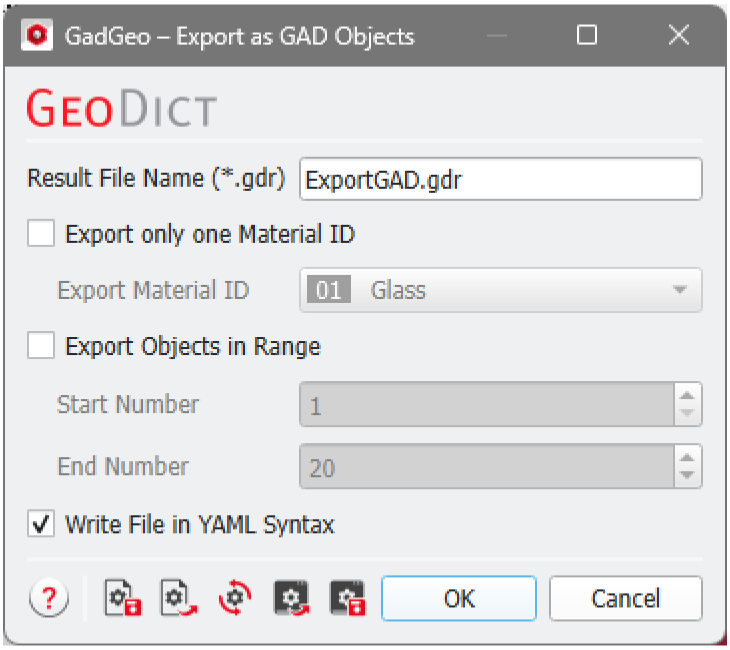
It is also possible to rotate the GAD objects directly when importing, achieving the same results as with the Rotate GAD Objects 46 command.

1.9 Export as GAD Objects

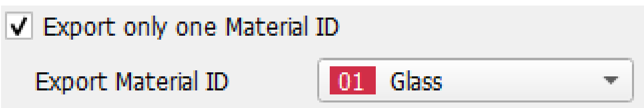
This command was part of the ExportGeo-Base module in versions 2023 or earlier. With it, you can create a .gad file that contains a selection of GAD objects present in the current structure.

Click the **Options**' **Edit...** button to open the **Export as GAD Objects** dialog.

There, select a **Result File Name**. The exported file is stored next to this *.gdr result file with the same base name and suffix *.gad. By default, all GAD objects are included in the file. The number of stored objects can be reduced by checking **Export only one Material ID** and/or **Export Objects in Range**.

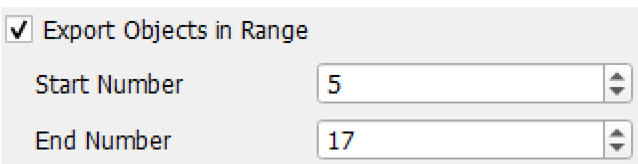


Export only one Material ID

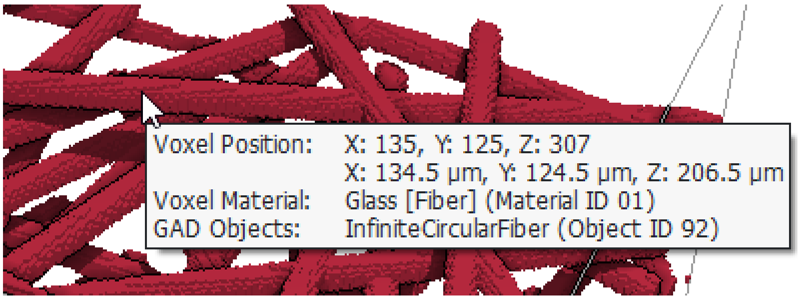


Check **Export only one Material ID** and select the **Export Material ID** to store only GAD objects of the selected Material ID in the written .gad file.

Export Objects in Range



Check **Export Objects in Range** and select **Start Number** and **End Number** to store only the GAD objects with an object ID in the selected range. The object IDs are, e.g., shown in the Edit GAD Objects command dialog or in the pop-up widget shown when the mouse pointer is rested over a certain object in the visualization area.



Write File in YAML Syntax

This option is checked by default. If unchecked, the .gad file is stored in the file syntax of version 2022 or earlier.

1.10 Convert to GAD

You can visualize particle trajectories or particle positions computed in FilterDict or AddiDict in GeoDict's visualization area. GeoDict distinguishes between the 3D voxel structure and the results computed on that structure. So when you save or export the 3D structure, the visualized results are not part of this structure.

Sometimes it is desirable to include particles, trajectories or streamlines in the exported structure file. To enable this, you can convert visualized results into GAD objects which are automatically added to the voxel grid with the commands in this menu. Afterwards, you can export or save this structure, e.g., with the Export as GAD Objects 62 command.

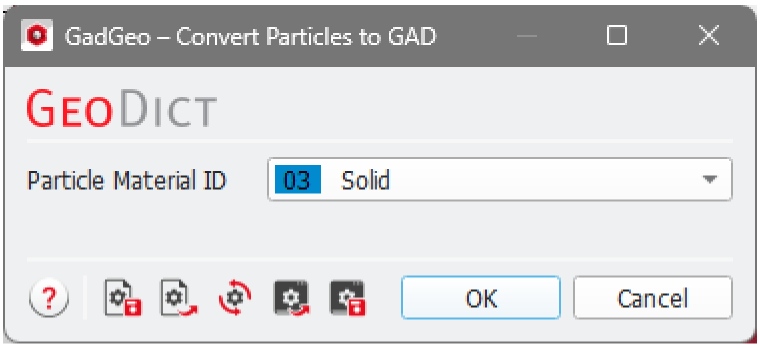
Commands available in the Convert to GAD menu

- Convert Particles to GAD 64
  Convert visualized particles into spherical GAD objects.
- Convert Trajectories to GAD 65
  Convert particle trajectories into cylindrical (CurvedCircularFiber) GAD objects.
- Convert Streamlines to GAD 66
  Convert streamlines into cylindrical (CurvedCircularFiber) GAD objects.

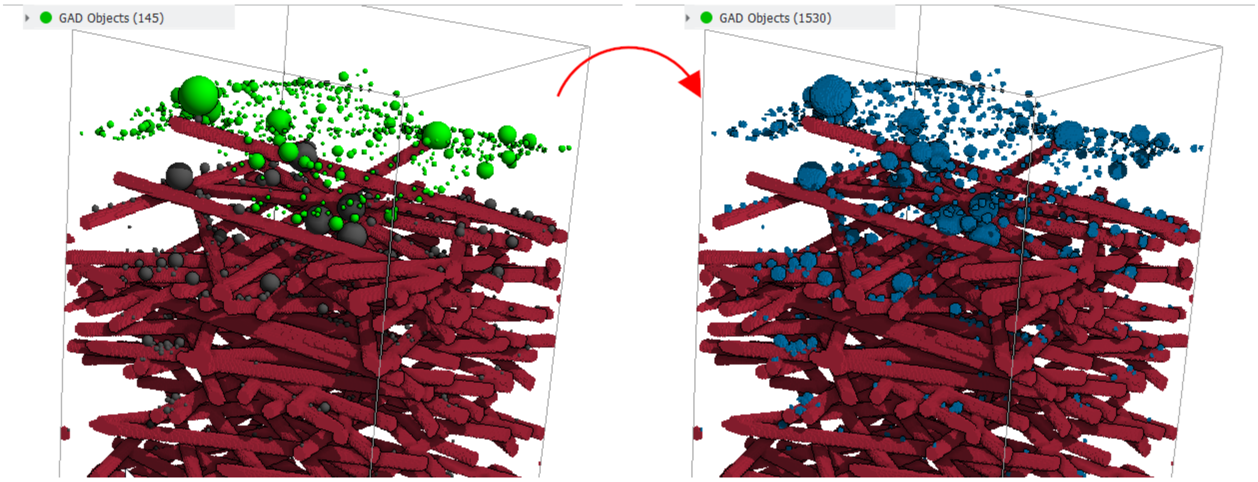
1.10.1 Convert Particles to GAD

The **Convert Particles to GAD** command converts the particles currently shown in the visualization area into spherical objects and adds them to the 3D structure. The command only has an effect when a particle (*.gpp) or particle trajectory (*.gpt) file is loaded and visualized, e.g., from a FilterDict or AddiDict result file.

Click the **Options**' **Edit...** button to open the **Convert Particles to GAD** dialog. Select a Material ID, close the dialog and click **Run** to convert the particles.



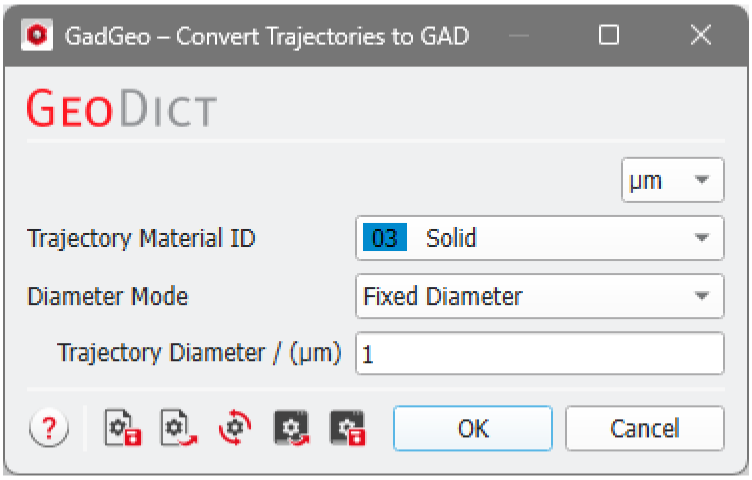
All particles are saved as **Sphere** objects at their current visualized position. This means that it is possible to add a particle at any point along its trajectory depending on the slider positions in the **Particles** visualization tab. If multiple sets of particles are visualized (e.g., deposited particles in dark gray and moving particles in green), they are simultaneously converted to GAD objects with this command using the same selected Material ID.



1.10.2 Convert Trajectories to GAD

The **Convert Trajectories to GAD** command converts the trajectories currently shown in the visualization area into curved circular fibers and adds them to the 3D structure. The command only has an effect when a particle trajectory (*.gpt) file is loaded and visualized, e.g., from a FilterDict or AddiDict result file.

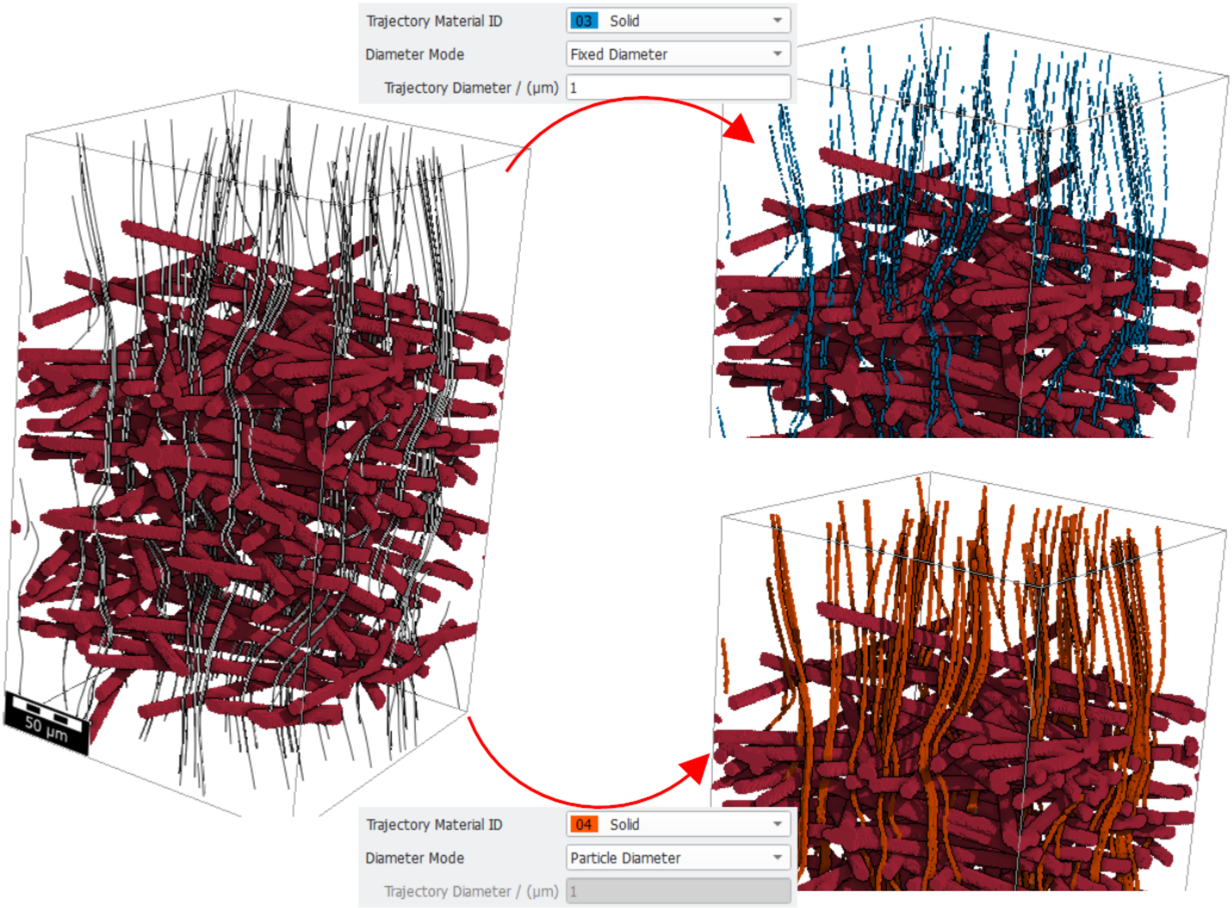
Click the **Options**' **Edit...** button to open the **Convert Trajectories to GAD** dialog.



The fiber diameter of the created GAD objects either corresponds to the **Particle Diameter** or you may select a **Fixed Diameter** for all trajectories. The command will turn each trajectory (a one-dimensional line) into a three-dimensional object by converting it into a curved circular fiber with the trajectory as the center line. The curved circular fiber will also be added to the current voxel grid. Be aware that only fibers with a diameter larger than several voxels are resolved well in the voxel grid.

Only the trajectories currently selected for visualization in the **Particles** View Controls tab will be converted into GAD objects.

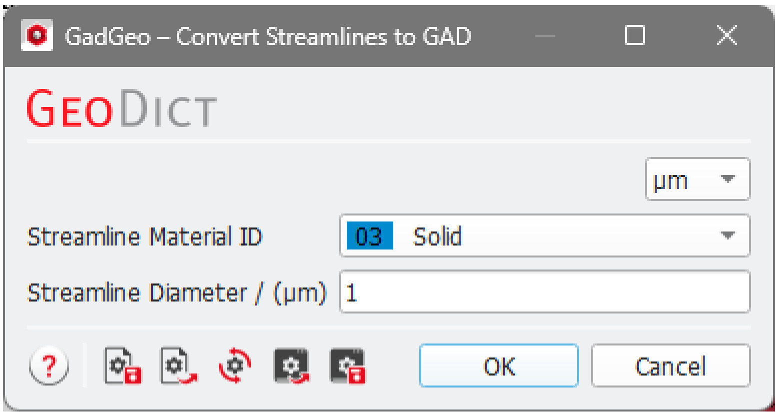
Close the dialog and click **Run** to convert the particle trajectories into **CurvedCircularFiber** GAD objects.



1.10.3 Convert Streamlines to GAD

The **Convert Streamlines to GAD** command converts the streamlines currently shown in the visualization area into curved circular fibers and adds them to the 3D structure. The command only has an effect when a flow field (*.vap) file is loaded and visualized as streamlines.

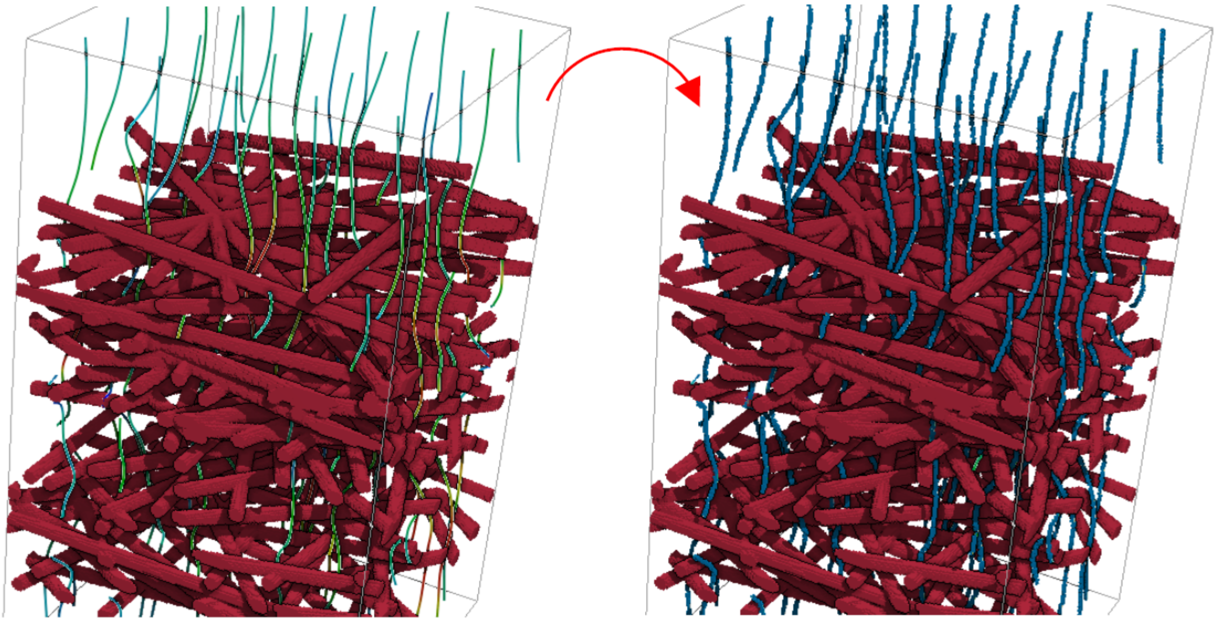
Click the **Options**' **Edit...** button to open the **Convert Streamlines to GAD** dialog.



The diameter of the created GAD objects corresponds to the selected **Streamline Diameter**. The command will turn each streamline (a one-dimensional line) into a three-dimensional object by converting it into a curved circular fiber with the streamline as the center line. The curved circular fiber will also be added to the current voxel grid. Be aware that only fibers with a diameter larger than several voxels are resolved well in the voxel grid.

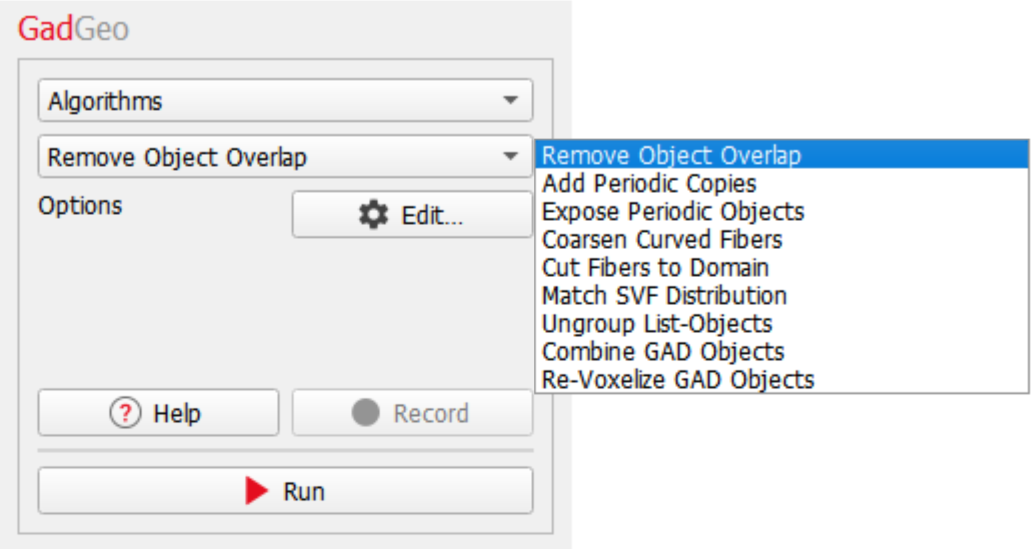
Only the streamlines currently selected for visualization in the **Streamlines** View Controls tab are converted into GAD objects.

Close the dialog and click **Run** to convert the streamlines into **CurvedCircularFiber** GAD objects.

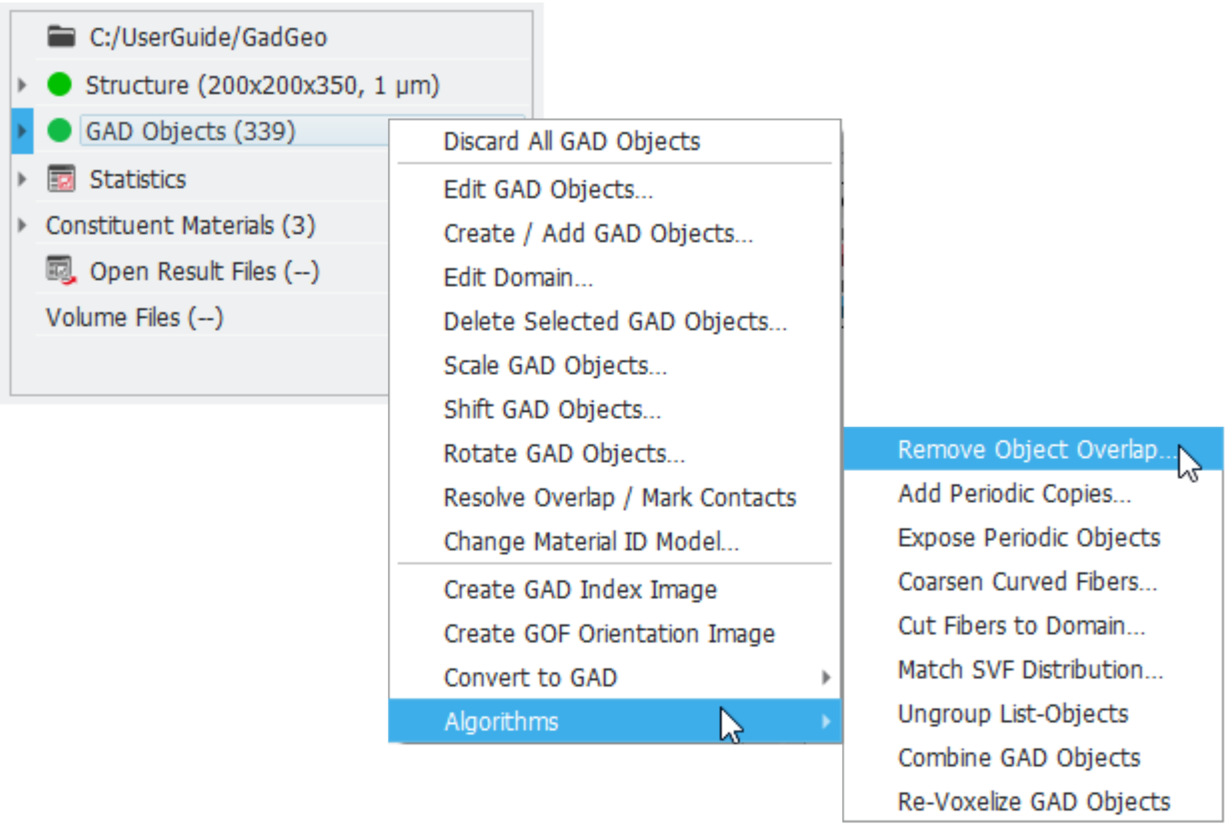


1.11 Algorithms

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



All of them are also accessible from the **GAD Objects** project status section menu.



Important! The commands under **Algorithms** modify the current structure and no result files are saved. Select **File** → **Save Structure As...**, in the menu bar, to save the modified structures in GAD format.

Commands available in the Algorithms menu

- Remove Object Overlap 69
  Shift, rotate or deform objects to reduce the overlap areas between objects.
- Add Periodic Copies 71
  Repeat the structure periodically.
- Expose Periodic Objects 72
  Dissolve periodic reappearances of objects crossing periodic boundaries.
- Coarsen Curved Fibers 73
  Reduce the number of fiber segments in curved fibers.
- Cut Fibers to Domain 75
  Shorten fibers to the current domain.
- Match SVF Distribution 76
  Re-distributes the GAD objects to achieve a predefined solid volume fraction distribution.
- Ungroup List-Objects 80
  Split objects composed of multiple segments or components into separate objects.
- Combine GAD Objects 81
  Create a single combined object out of all current objects.
- Re-Voxelize GAD Objects 83
  Re-create the voxel strcuture from the current list of GAD objects.

1.11.1 Remove Object Overlap

If the current 3D structure consists of grains or fibers that intersect (or overlap) with each other, you can use the **Remove Object Overlap** command to modify the current 3D structure such that the overlap between the individual objects is minimized.

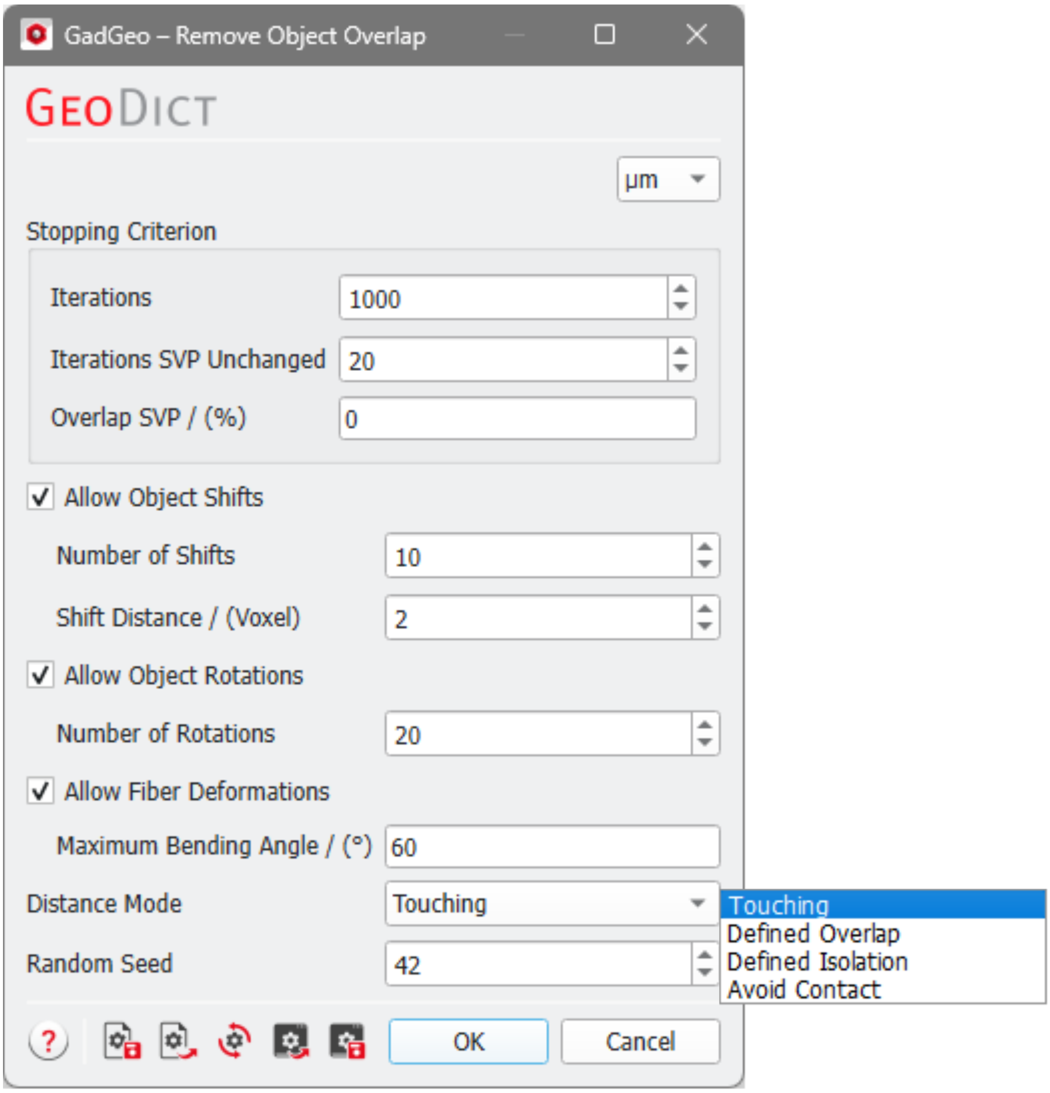
The options of **Remove Object Overlap** are similar to the choices available under the **Generation and Overlap Mode** panel of FiberGeo for fibrous structures (FiberGeo Create Options dialog) and GrainGeo for structures with granular objects (GrainGeo Create Grains Options dialog).

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 percentage 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.



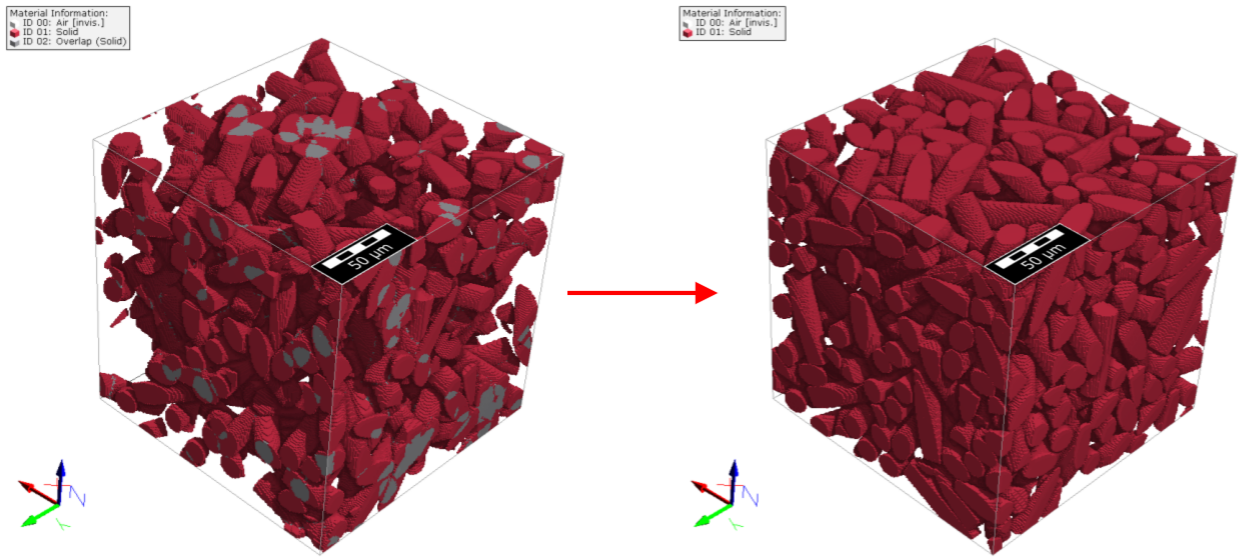
Choose the Distance Mode between **Touching**, **Defined Overlap**, **Defined Isolation** and **Avoid Contact** to set the separation that should be achieved between objects. For **Defined Overlap**, set the **Distance** that objects are still allowed to overlap. For **Defined Isolation**, set the minimal **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 Object Overlap** dialog. The gray areas correspond to the overlap between the red fibers.

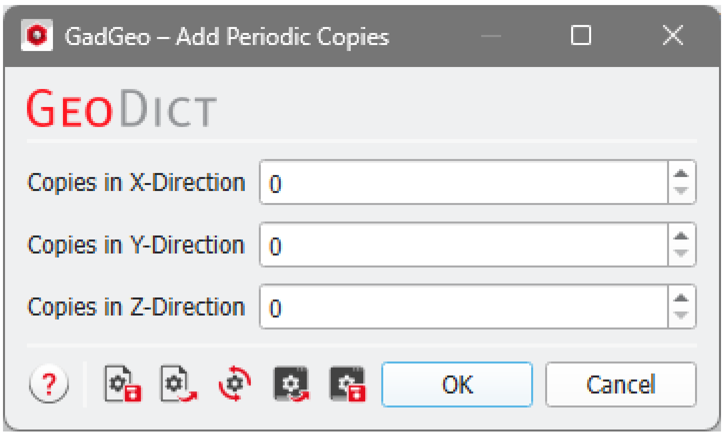
Observe how the shape and the position of the fibers have been modified to avoid the overlap.



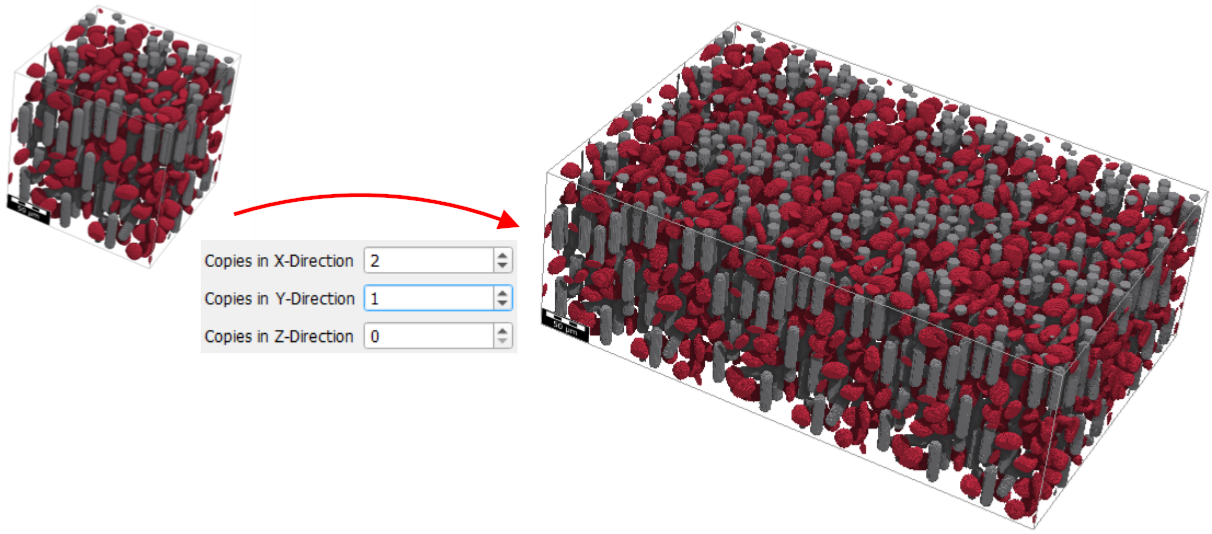
Depending on the stopping criterion reached, it is possible that the remaining structure still contains some overlap. The current overlap settings decide which material ID the overlap material gets. You may use the Edit Domain 55 command to redefine the overlap behavior, e.g., to mark overlapping parts with a different material ID.

1.11.2 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.



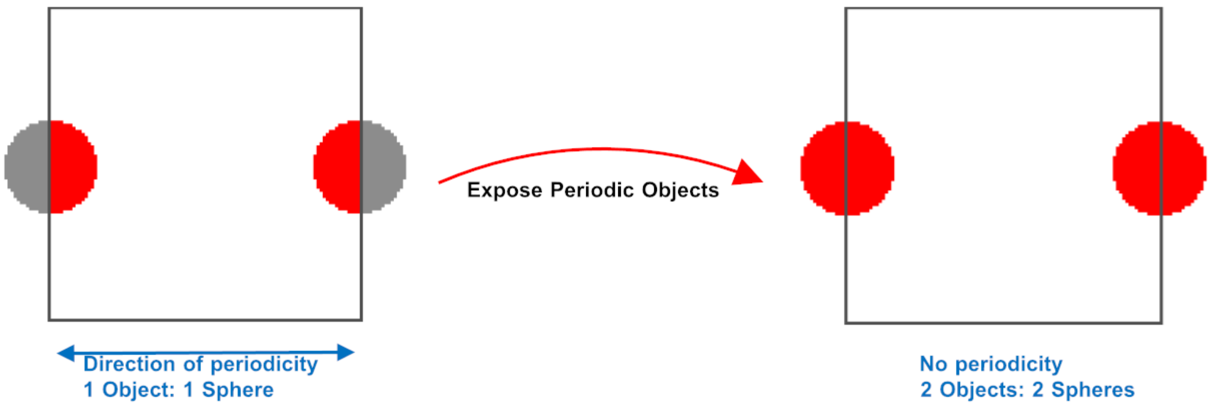
To add periodic copies, the original structure model should be periodic in the direction(s) to be copied. Otherwise, the resulting structure may show unwanted boundary effects. Here, periodic copies of the structure are added in the X-direction (2 copies) and the Y-direction (1 copy).



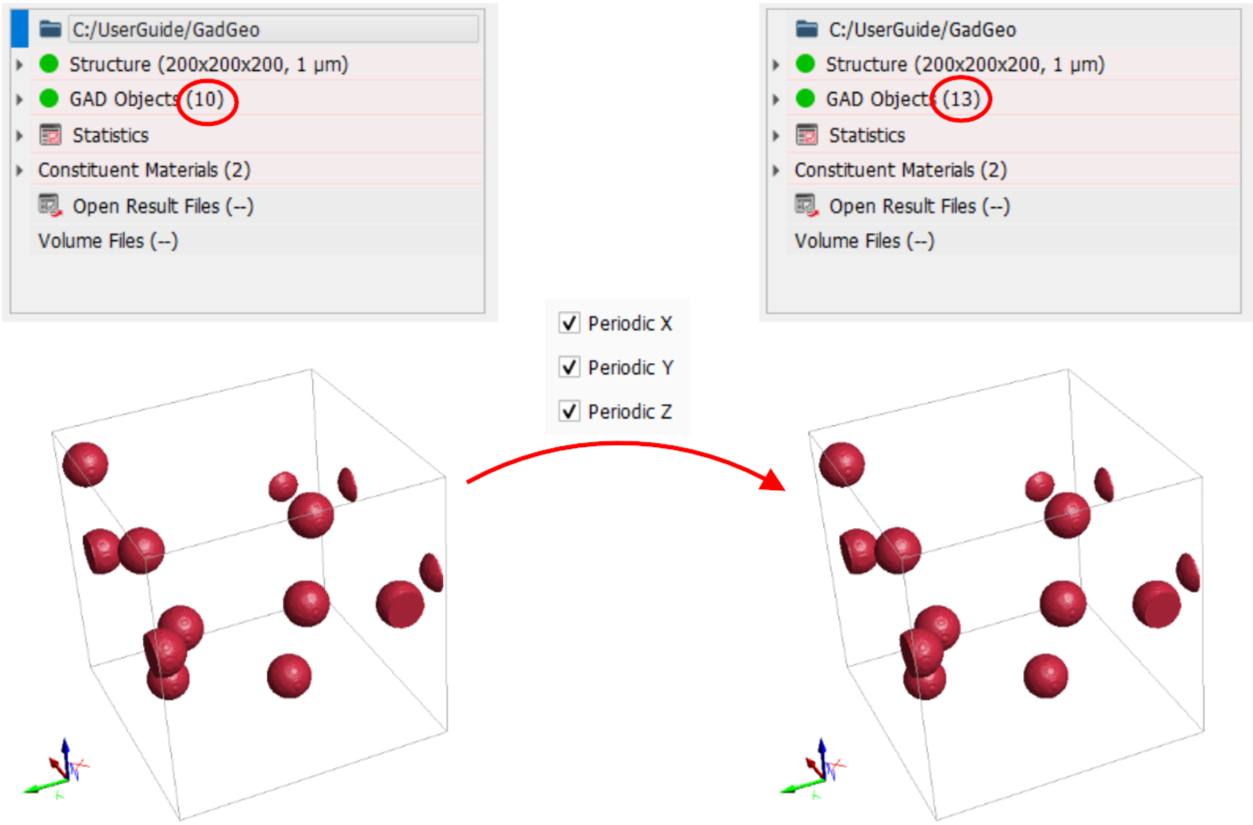
1.11.3 Expose Periodic Objects

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.

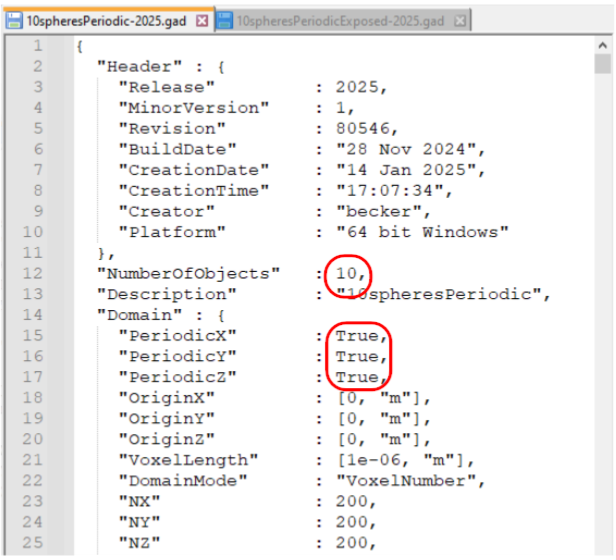
Therefore, for periodic GeoDict-generated structures, **Expose Periodic Objects** produces non-periodic structures in which the interrupted periodic objects in the structure are added to the structure's analytic description (GAD).



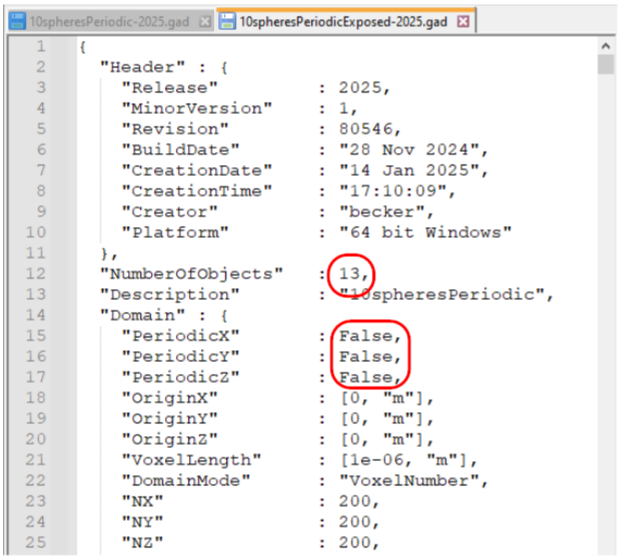
Although no difference is observable when simply visualizing the structure, the result of exposing periodic objects is the loss of periodicity and a higher number of objects in the structure. The higher number of objects after performing this process is directly observable in the number of **Objects**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




```
{
  "Header" : {
    "Release"          : 2025,
    "MinorVersion"     : 1,
    "Revision"         : 80546,
    "BuildDate"        : "28 Nov 2024",
    "CreationDate"     : "14 Jan 2025",
    "CreationTime"     : "17:07:34",
    "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,
```



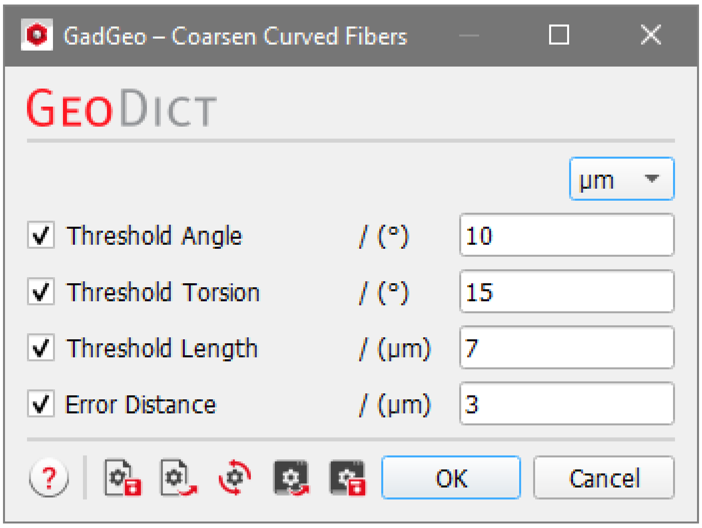

```
{
  "Header" : {
    "Release"          : 2025,
    "MinorVersion"     : 1,
    "Revision"         : 80546,
    "BuildDate"        : "28 Nov 2024",
    "CreationDate"     : "14 Jan 2025",
    "CreationTime"     : "17:10:09",
    "Creator"          : "becker",
    "Platform"         : "64 bit Windows"
  },
  "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,
```

1.11.4 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 MeshGeo or ExportGeo-CAD. Clicking the **Options**' **Edit...**

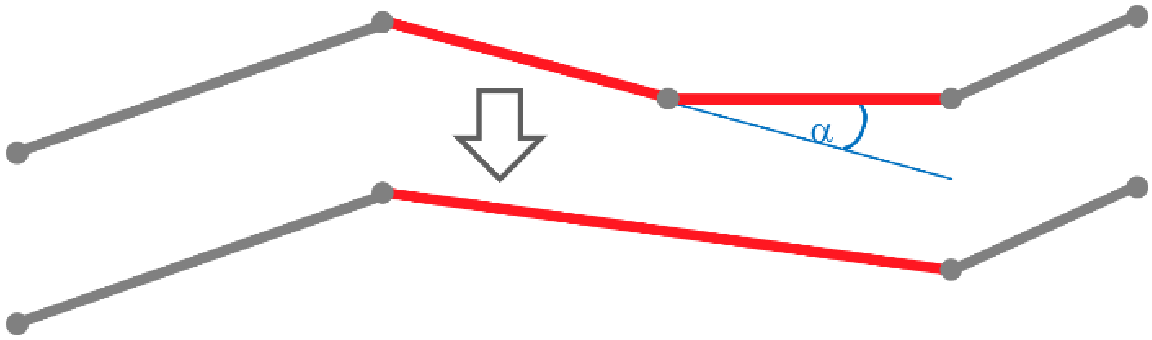
button opens the **Coarsen Curved Fibers** dialog where criteria for merging two segments into one can be entered.

The available **units** (m, cm, mm, µm, nm, and Inch) are selectable from the top right pull-down menu. When the **Threshold Angle**, **Threshold Torsion**, **Threshold Length** and/or **Error Distance** 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.

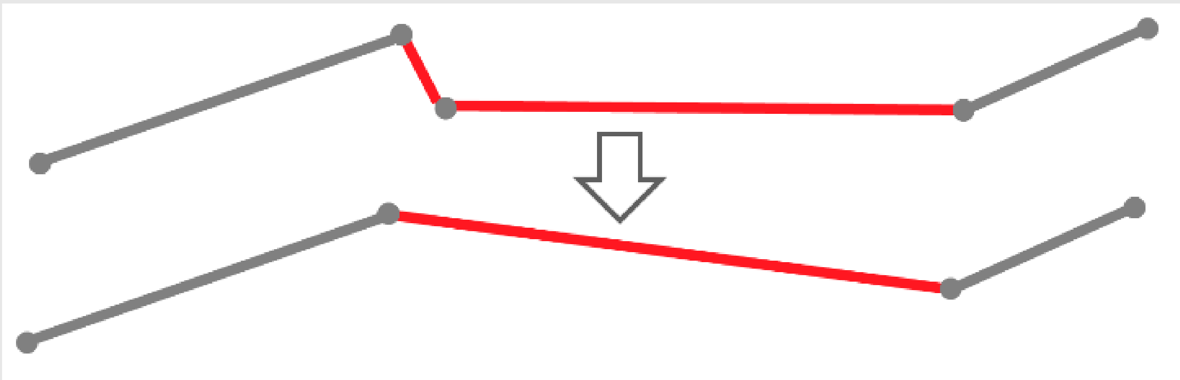


Threshold Torsion

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.



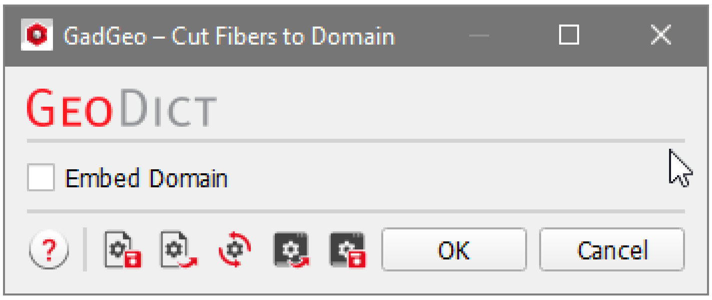
Error Distance

Points on the fiber are removed if the distance between the removed point and the new line is lower than the given **Error Distance** (Douglas–Peucker algorithm).

1.11.5 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.

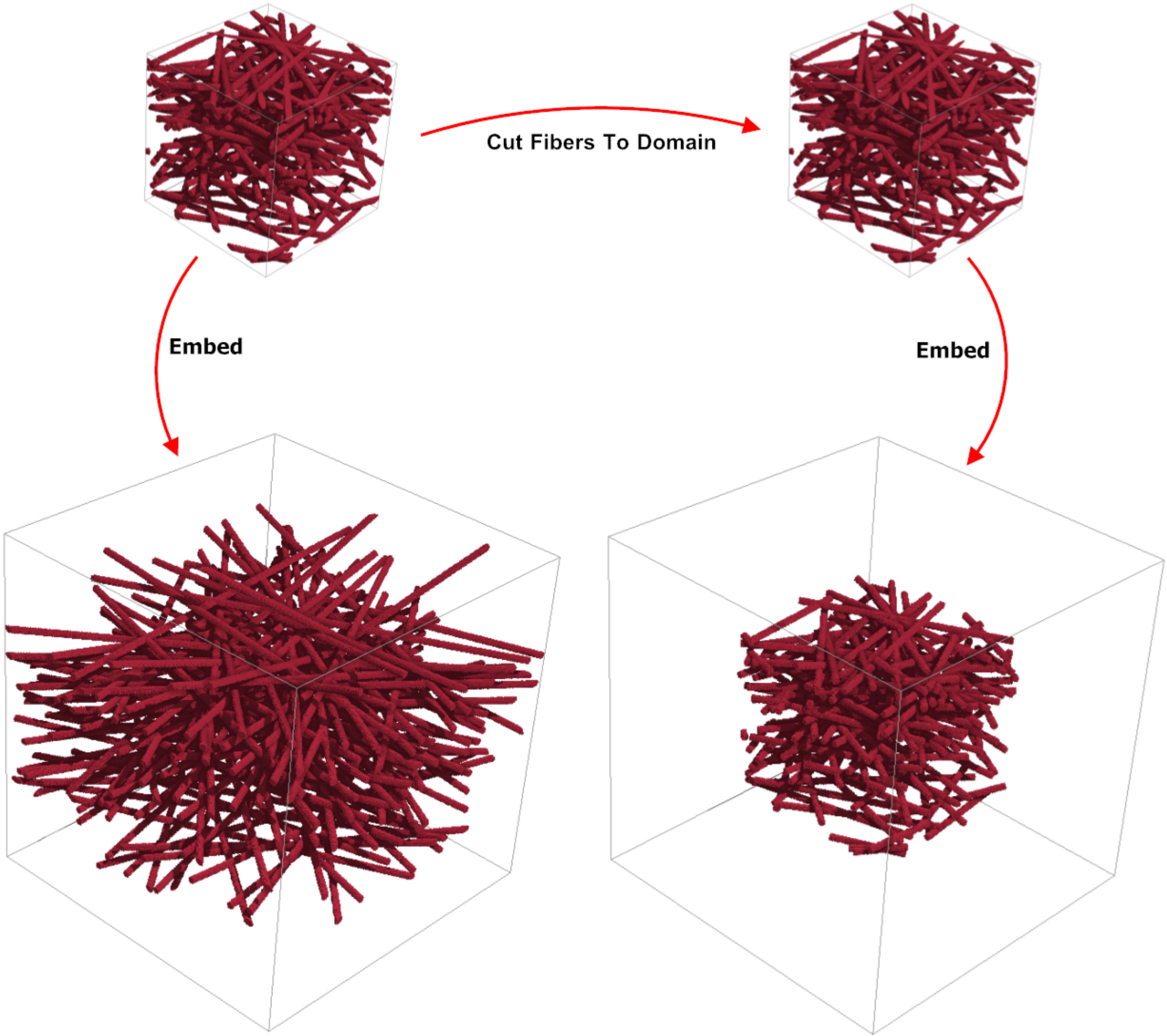
This option is useful when exporting the fibers as objects, e.g., for meshing them with MeshGeo, or for combining them with other structures while maintaining the object information, e.g., in LayerGeo. Click the **Options' Edit...** button to open the **Cut Fibers to Domain** dialog.



The command will cut the fiber center-lines at the domain boundary. For periodic geometries, the resulting voxel geometry will not change by this. For non-periodic geometries, the voxel geometry inside of the domain may change near the fiber cut-points at the domain boundary.

To avoid this, enable **Embed Domain** to move the fiber cut-points outwards from the domain boundary by the maximal fiber radius. In this case, the fiber center-lines will end outside of the domain and the voxel geometry inside of the domain will not change. This option is available only for non-periodic domains.

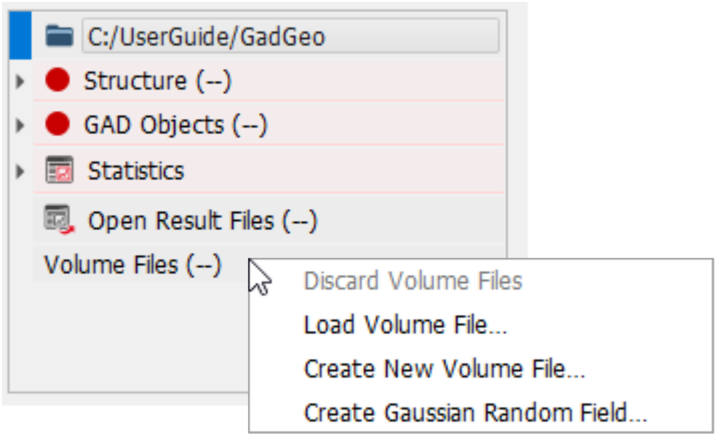
The effect of cutting fibers to the domain can best be observed when embedding the structure with ProcessGeo-Embed.



1.11.6 Match SVF Distribution

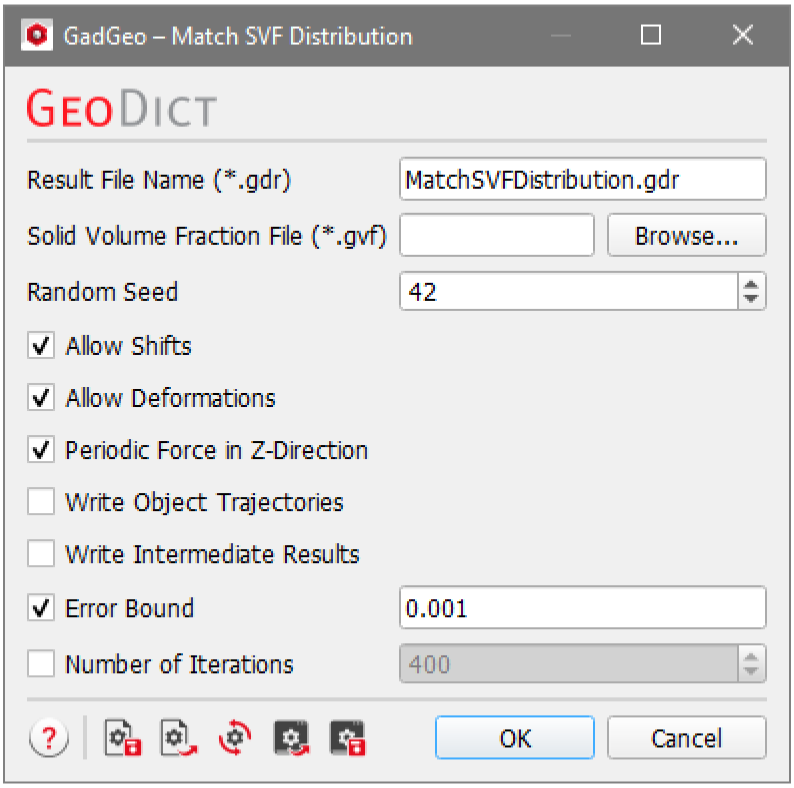
The **Match Solid Volume Fraction (SVF) Distribution** command allows to re-distribute the GAD objects of the current structure such that a desired inhomogeneous solid volume fraction distribution is reached. A specialized form of this algorithm is also available in FiberGeo and GrainGeo, where a Gaussian 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 Files** project status section entry (this feature is only available if FiberGeo or GrainGeo is licensed).



Alternatively, use **3D Inhomogeneity** in MatDict to create a distribution based on a CT-scan or a mask structure (see example 79) 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 is 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 (Stopping Criterion)

In each step, the objects are shifted and deformed 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 solid volume fraction (SVF) has changed in the last iteration and to define a tolerance value for which the simulation should be stopped. This is the recommended stopping criterion.

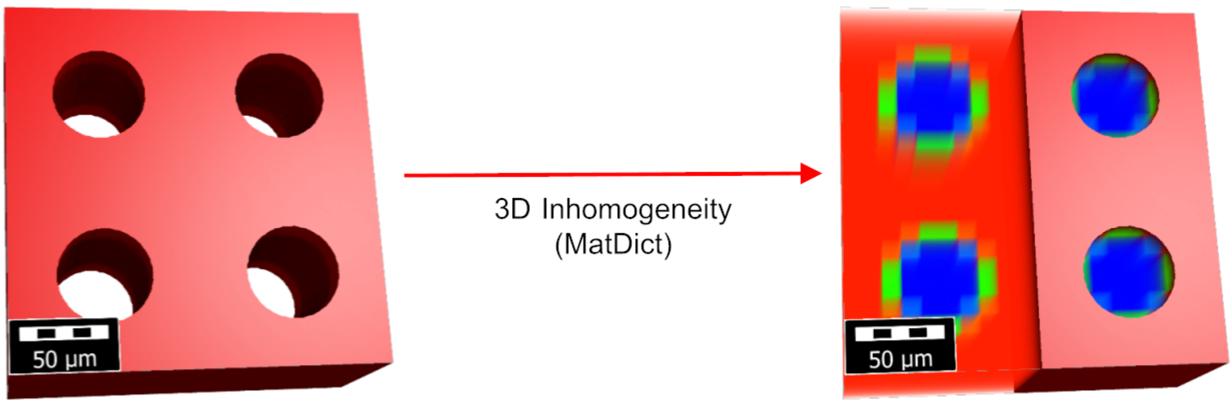
Number of Iterations (Stopping Criterion)

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

Example

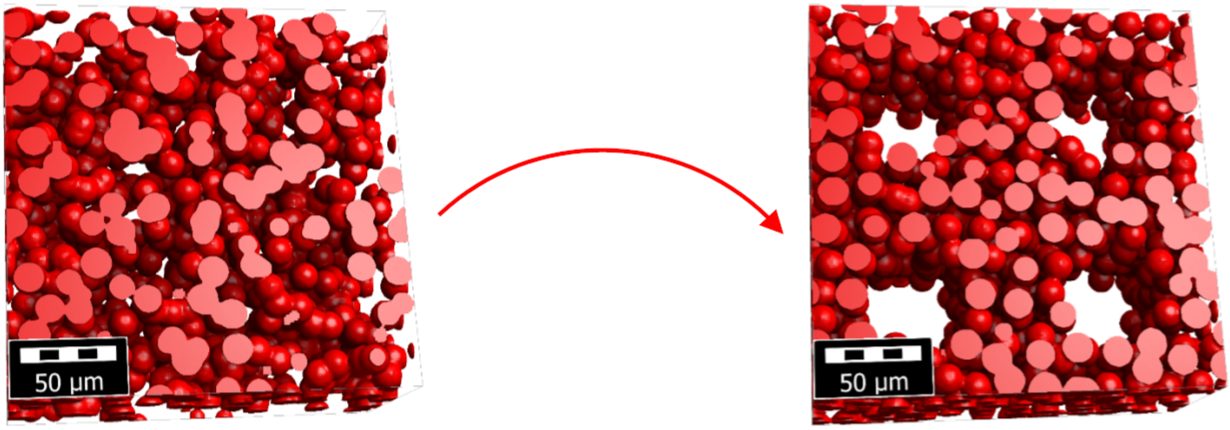
One way to use the **Match SVF Distribution** command is to create a mask structure and to convert this mask into a solid volume fraction file with a lower resolution (*.gvf). This file is afterwards used to change a base structure to follow the solid volume fraction distribution described by the mask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

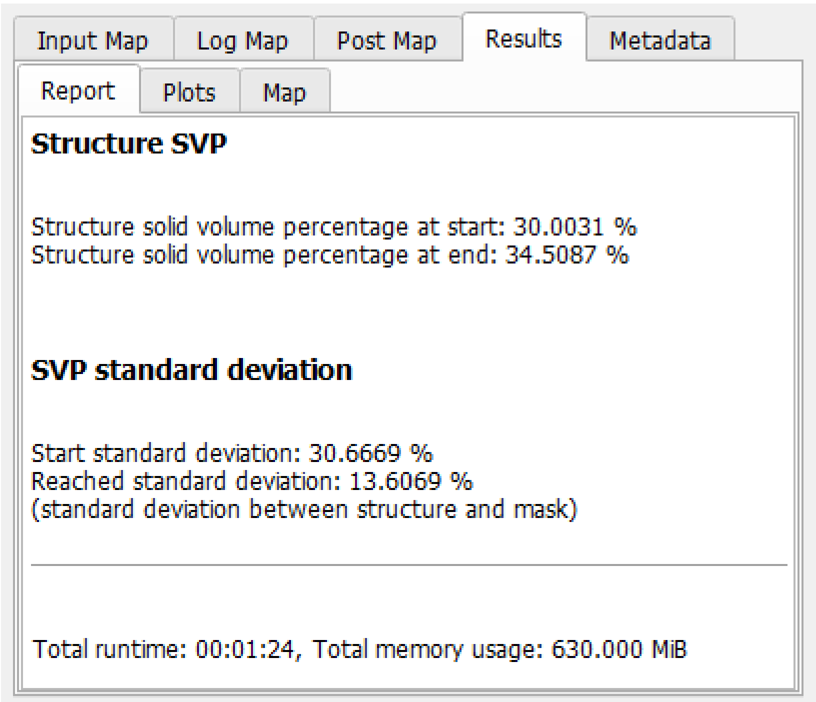


Now, we create a base structure of homogeneously distributed spheres with GrainGeo, where the mean solid volume fraction is 30%. The grain structure is loaded and the created solid volume fraction file (*.gvf) is chosen in the **Match SVF Distribution** dialog. For the rest, we keep the default settings.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



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.



1.11.7 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 section entry).

The following object types consist of multiple segments or objects and are split into their respective segments when this command is run:

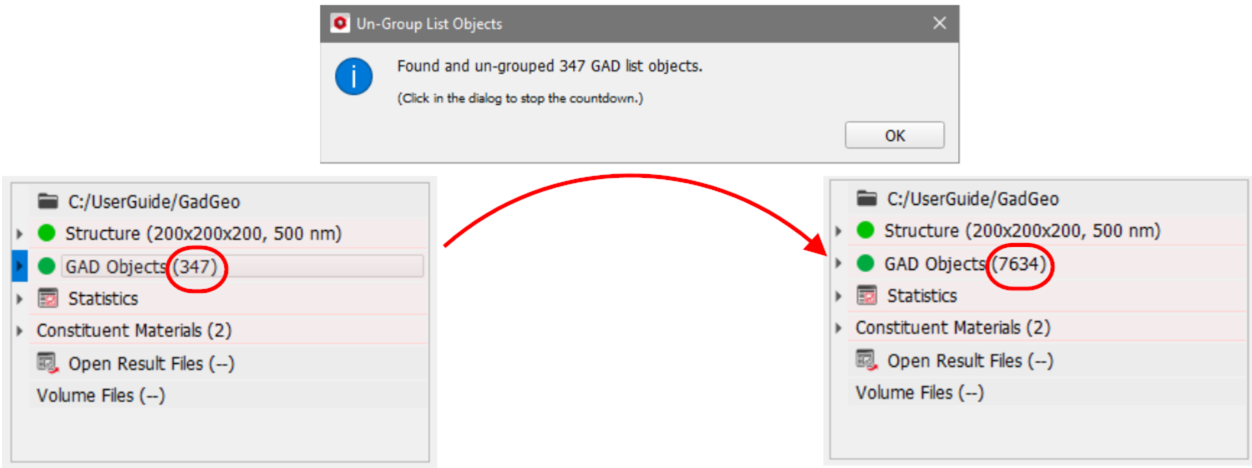
- CurvedXXXXFiber 18, where XXXX may be any one of the profile types Circular, Hollow, Rosetta, Elliptical, Cellulose, Rectangular, Angular, Arbitrary.

- GeneralCurvedXXXXFiber 19, where XXXX may be any one of the profile types Circular, Hollow, Rosetta, Elliptical, Cellulose, Rectangular, Angular, Arbitrary.
- CombinedObject 25
- IntersectedObject 28

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.

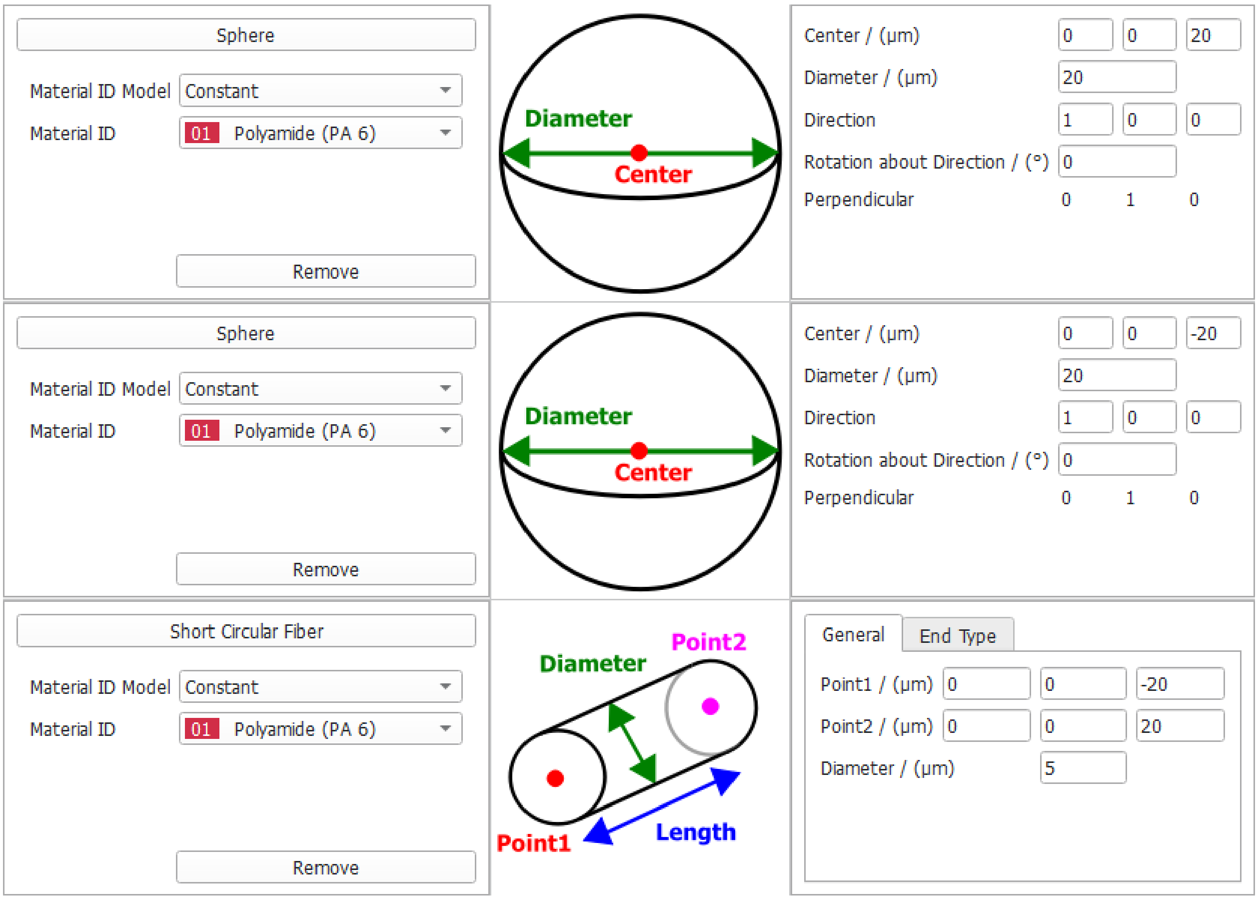


1.11.8 Combine GAD Objects

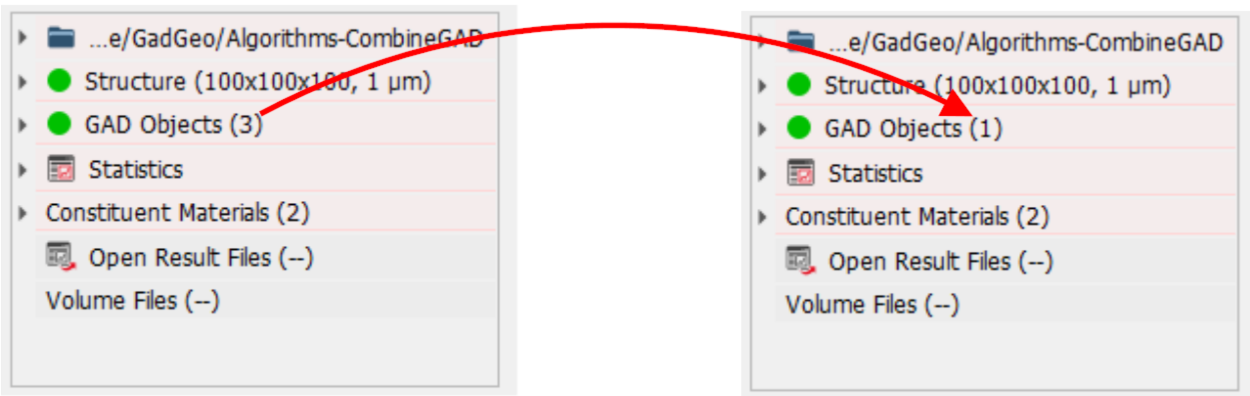
The Combine GAD Objects command combines all objects into a single CombinedObject.

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.

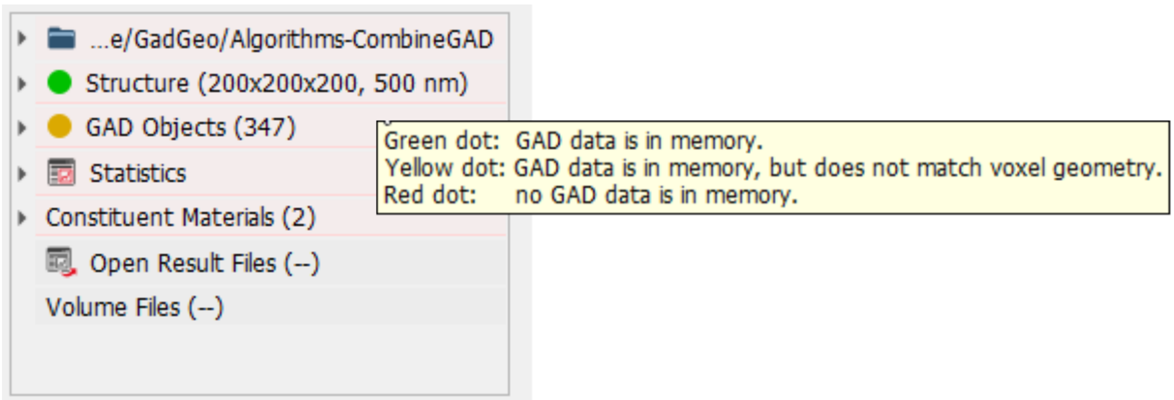


If this object is now stored in a .gad file, only a single object of the CombinedObject type is present. This file can then be used as a user-defined object type 36.

```
"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
  }
},
```

1.11.9 Re-Voxelize GAD Objects

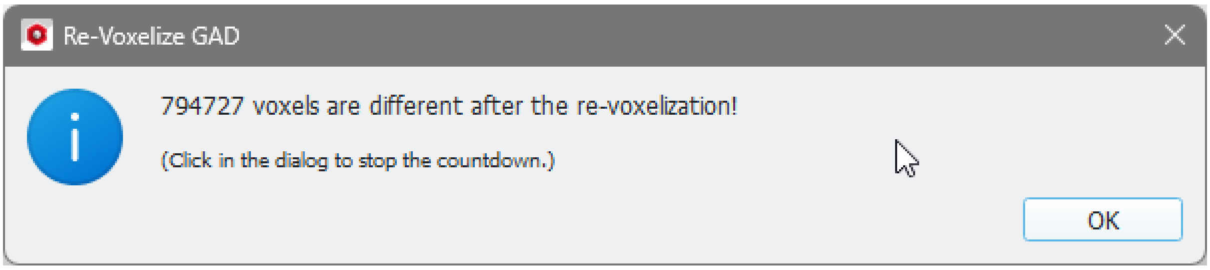
It is possible that the current list of GAD objects does not fully describe the current voxel structure. This can happen if binder (which has no gad representation) is added to a fiber or grain structure, or if the voxel structure has been modified in any other way. This is shown by a yellow dot in the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

A pop-up window informs you how many voxels have a different Material ID after the command was executed.



Citation: Math2Market GmbH, 2026: GeoDict 2026 User Guide, GadGeo, Math2Market GmbH, Germany, doi.org/10.30423/userguide.geodict

MATH
2 MARKET

Math2Market GmbH
Richard-Wagner-Str. 1, 67655 Kaiserslautern / Germany
www.math2market.com