

# Introduction to the FreeCAD GDML Workbench

Keith Sloan & Munther Hindi

15<sup>th</sup> Geant4 Space Users Workshop  
Pasadena, December 2023

# GDML

GDML: Geometry Description Markup Language.

- A specialized XML-based language designed for describing the geometries of detectors for physics experiments.
- Integrated into Geant4 since release 9.2
- Need to set `GEANT4_USE_GDML = 1` when compiling
- GDML Workbench: A tool for importing GDML files into (Free)CAD geometries and exporting FreeCAD geometries to GDML

## Abbreviated GDML file structure

```
<?xml version="1.0" encoding="UTF-8"?>
<gdml xsi:noNamespaceSchemaLocation="schema/gdml.xsd">
    <define>
        ...
        <position name="TrackerinWorldpos" unit="mm" x="0" y="0" z="100"/>
    </define>
    <materials>
        ...
        <element name="Nitrogen" formula="N" Z="7.">
            <atom value="14.01"/>
        </element>
        <material formula="" name="Air" >
            <D value="1.290" unit="mg/cm3"/>
            <fraction n="0.7" ref="Nitrogen" />
            <fraction n="0.3" ref="Oxygen" />
        </material>
    </materials>
    <solids>
        ...
        <box lunit="mm" name="Tracker" x="50" y="50" z="50"/>
    </solids>
    <structure>
        ...
        <volume name="World" >
            <materialref ref="Air" />
            <solidref ref="world" />
            <physvol>
                <volumeref ref="Tracker" />
                <positionref ref="TrackerinWorldpos"/>
                <rotationref ref="TrackerinWorldrot"/>
            </physvol>
        </volume>
    </structure>
    <setup name="Default" version="1.0" >
        <world ref="World" />
    </setup>
</gdml>
```

# FreeCAD - an Open Source CAD Application

[freecad.org](http://freecad.org)

FreeCAD

Features Download Blog Documentation Contribute Community 

## Current stable version: 0.21.1

Select your desired platform (note that all downloads are for 64-bit systems):



**Windows**

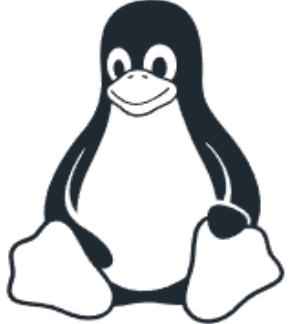
[64-bit installer](#)  
[64-bit portable \(.7z\)](#)

Windows 7 is the minimum supported version. For more info



**Mac**

[ARM \(M1/M2\) disk image](#)  
[Intel disk image](#)  
[MacOS 12 and older \(unsigned\)](#)

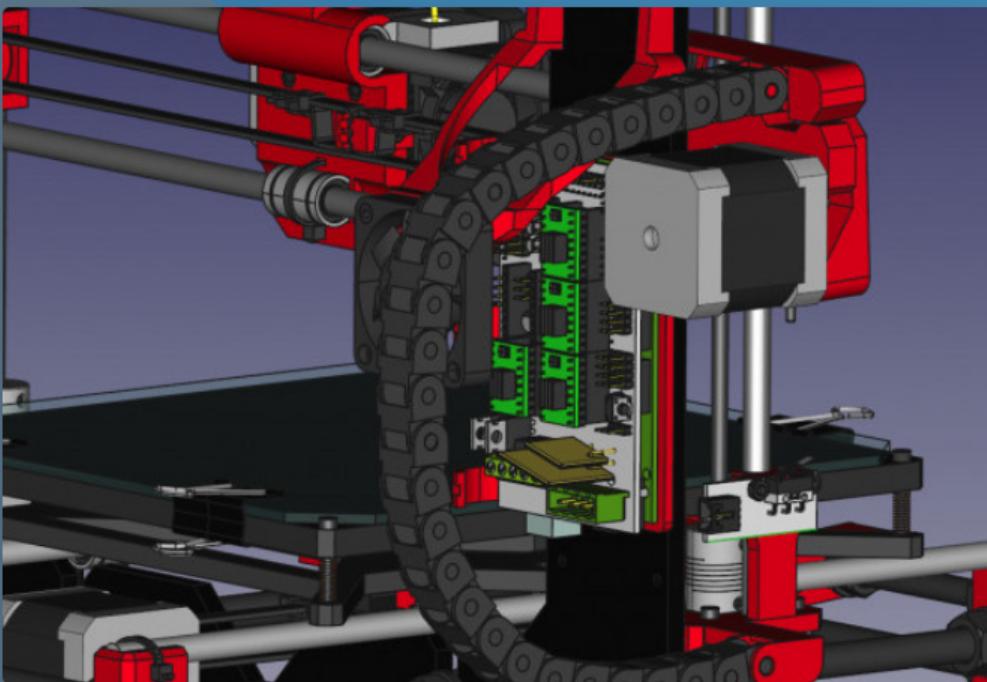


**Linux**

[x86\\_64 ApplImage](#)  
[aarch64 ApplImage](#)

For distro-specific instructions, such as Ubuntu PPA, and other

# Some Features(from the FreeCAD web site):



## A powerful solid-based geometry kernel

FreeCAD features an advanced geometry engine based on [Open CASCADE Technology](#). It supports solids, [Boundary Representation](#) (BRep) objects, and [Non-uniform rational basis spline](#) (NURBS) curves and surfaces, and offers a wide range of tools to create and modify these objects, including complex [Boolean](#) operations, [fillets](#), shape cleaning and [much more](#).

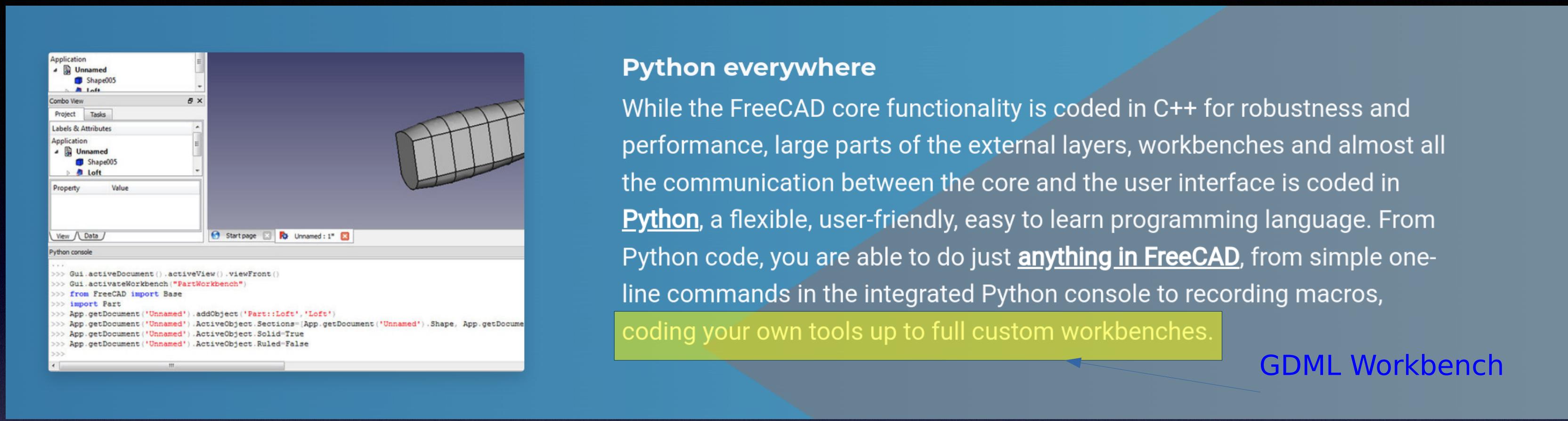
## A wi(l)dly parametric environment

All FreeCAD objects are natively parametric, meaning their shape can be based on [properties](#) such as numeric values, texts, on/off buttons, or even other objects. All shape changes are recalculated on demand, recorded by an undo/redo stack, and allow to maintain a precise modelling history.

Properties of one object can drive the value of properties of other objects, allowing complex, custom parametric chains that could only exist in your wildest dreams. New parametric objects are [easy to code](#).

The screenshot shows the FreeCAD interface with a 3D view of a rectangular block with a circular pocket. The 'Pocket' feature is selected in the tree view. To the right, a vertical flowchart shows the parametric history: Sketch -> Pad -> Pocket.

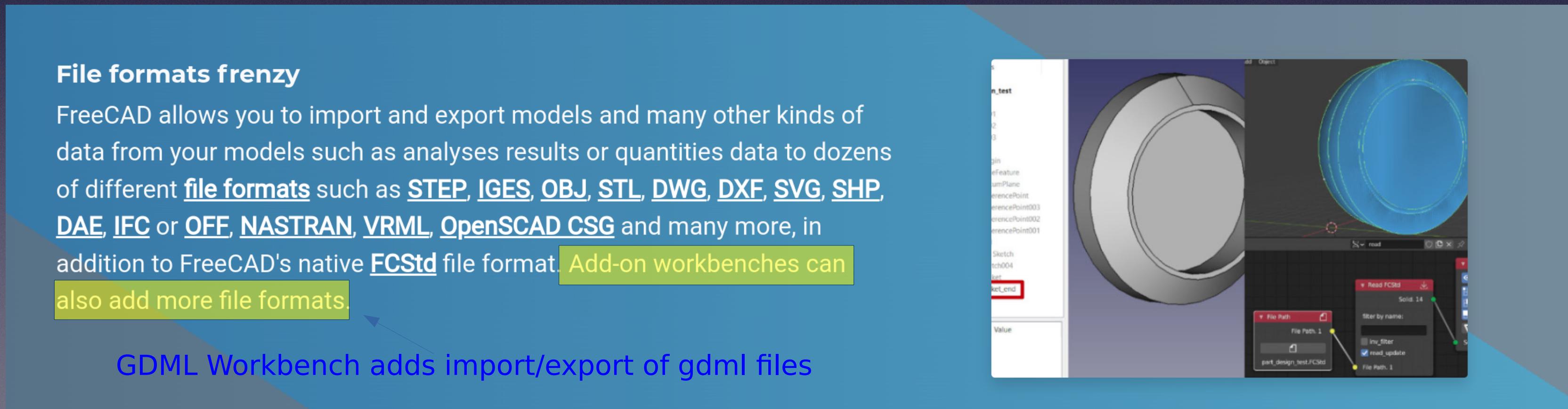
# FreeCAD features, continued



## Python everywhere

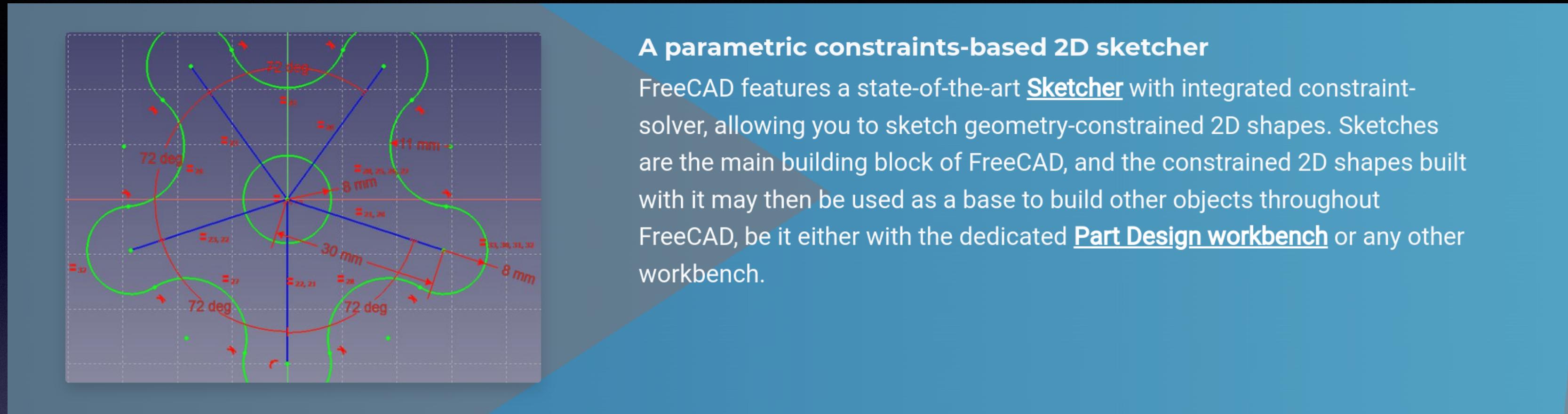
While the FreeCAD core functionality is coded in C++ for robustness and performance, large parts of the external layers, workbenches and almost all the communication between the core and the user interface is coded in **Python**, a flexible, user-friendly, easy to learn programming language. From Python code, you are able to do just anything in FreeCAD, from simple one-line commands in the integrated Python console to recording macros, coding your own tools up to full custom workbenches.

GDML Workbench



GDML Workbench adds import/export of gdml files

# FreeCAD features, continued



## A parametric constraints-based 2D sketcher

FreeCAD features a state-of-the-art [Sketcher](#) with integrated constraint-solver, allowing you to sketch geometry-constrained 2D shapes. Sketches are the main building block of FreeCAD, and the constrained 2D shapes built with it may then be used as a base to build other objects throughout FreeCAD, be it either with the dedicated [Part Design workbench](#) or any other workbench.

- Sketch parameters can have numeric values (with units) or come from a FreeCAD supported spreadsheet. As usual, cells can have aliases and reference other cells.
- Sketches can be converted to 3D solids by (a) extruding them (b) revolving them, (c) sweeping them along paths.



Functionality of FreeCAD accessed through **Workbenches**:  
Workbench: a set of tools + GUI grouped for a certain task.

Base Workbenches: installed with FreeCAD and maintained by them

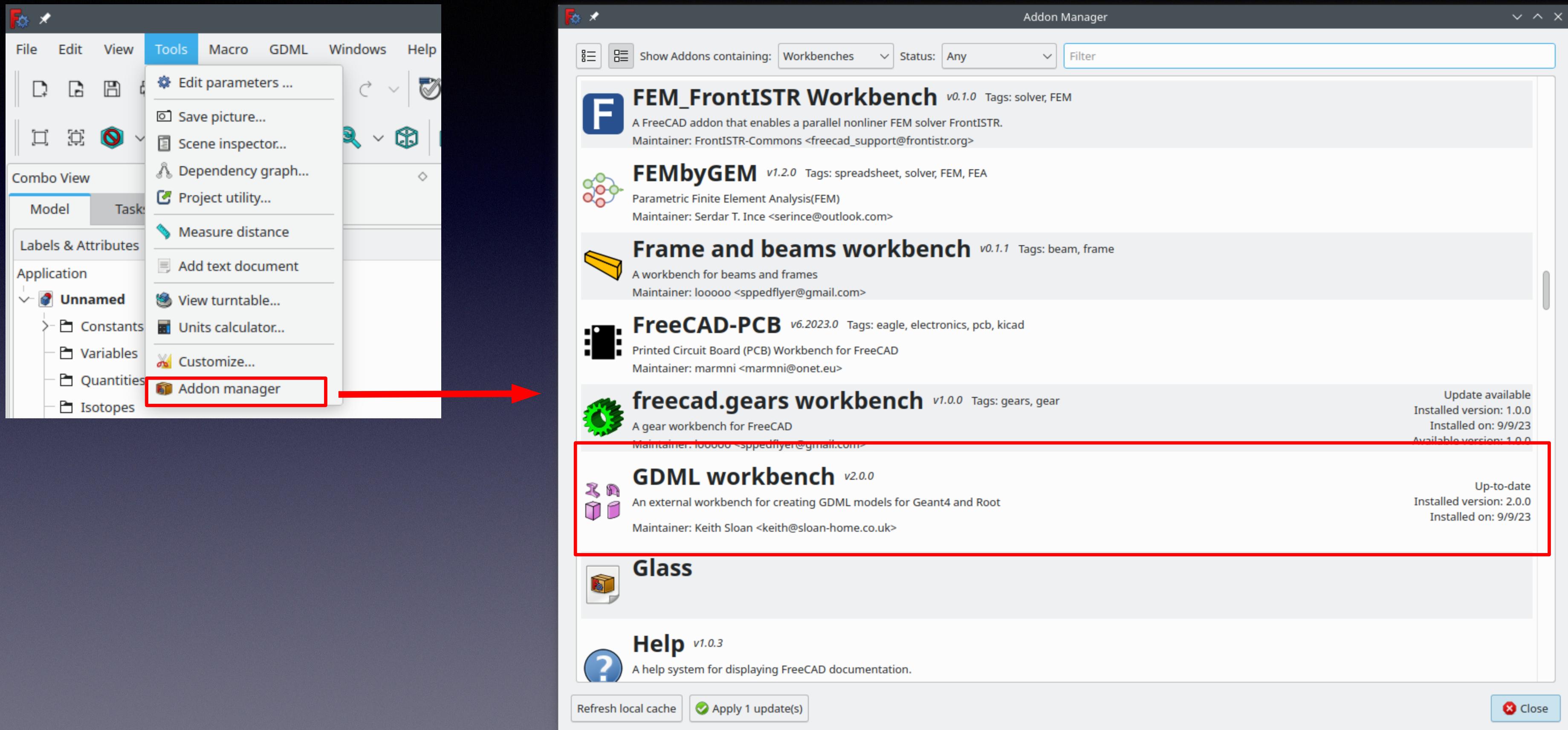
## External Workbenches: Developed by the community and maintained outside FreeCAD

# GDML: An external Workbench

## Examples



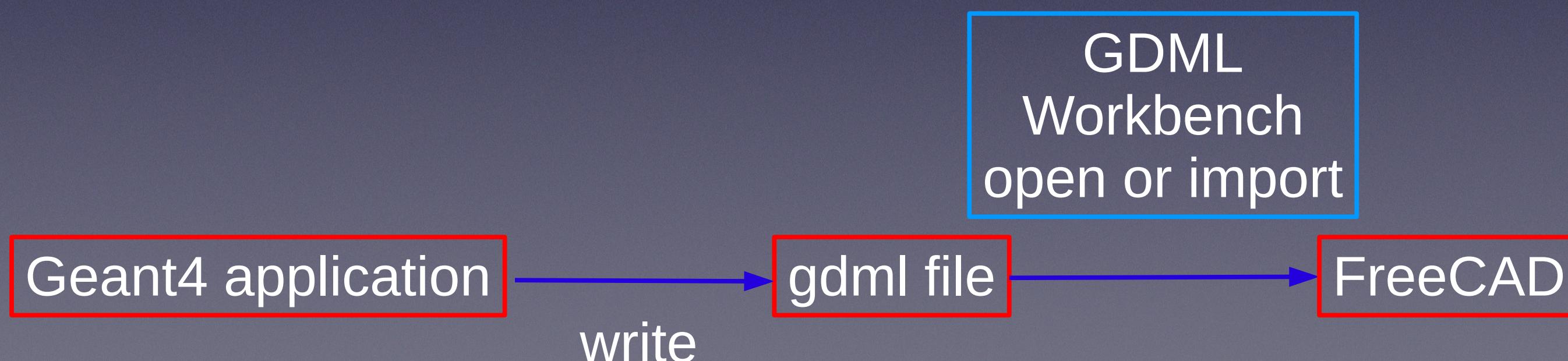
# External Workbenches Installed Via Tools→Addon Manager



In this presentation we concentrate on going from FreeCAD to gdml/Geant4:

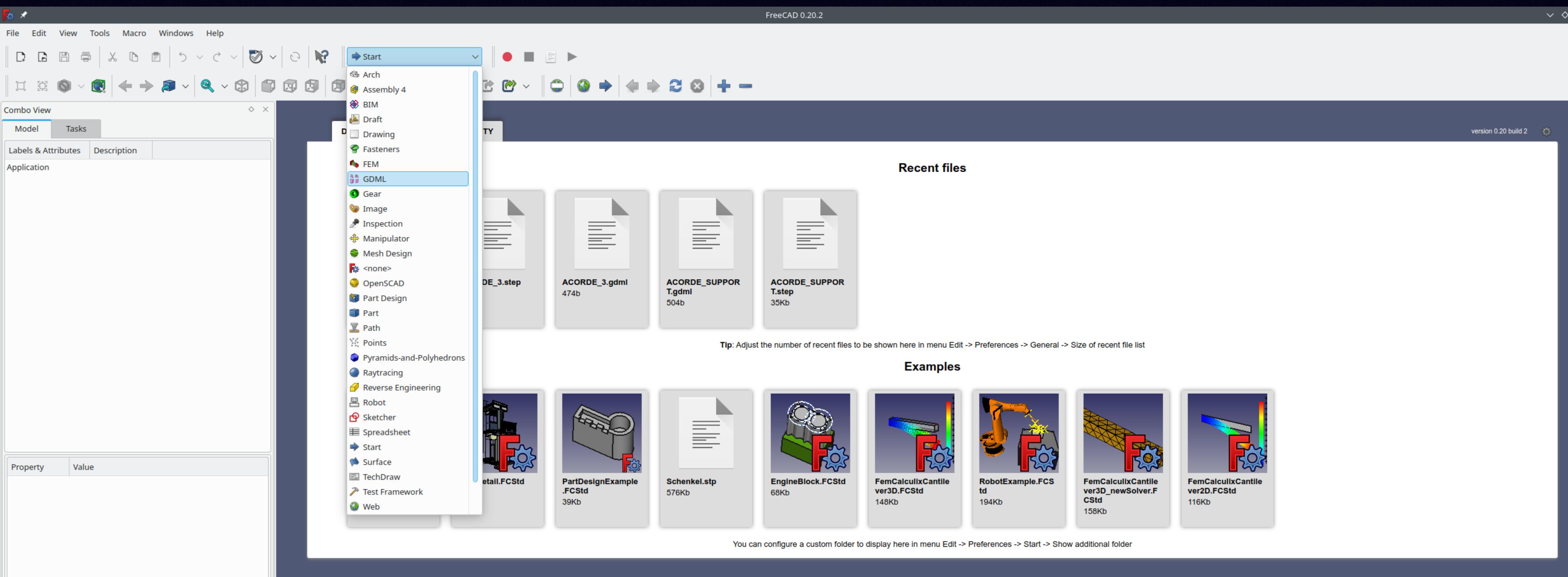


But one can also go from a Geant4 geometry to FreeCAD (see example G01):



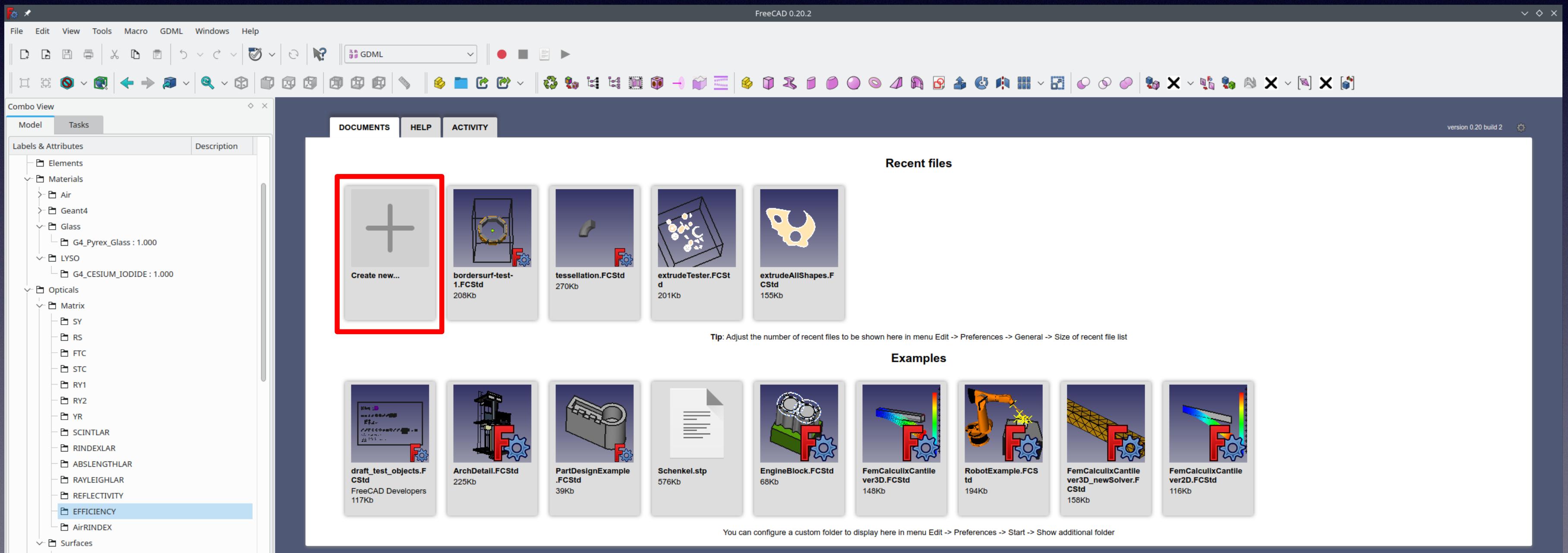
# Creating a gdml file from scratch

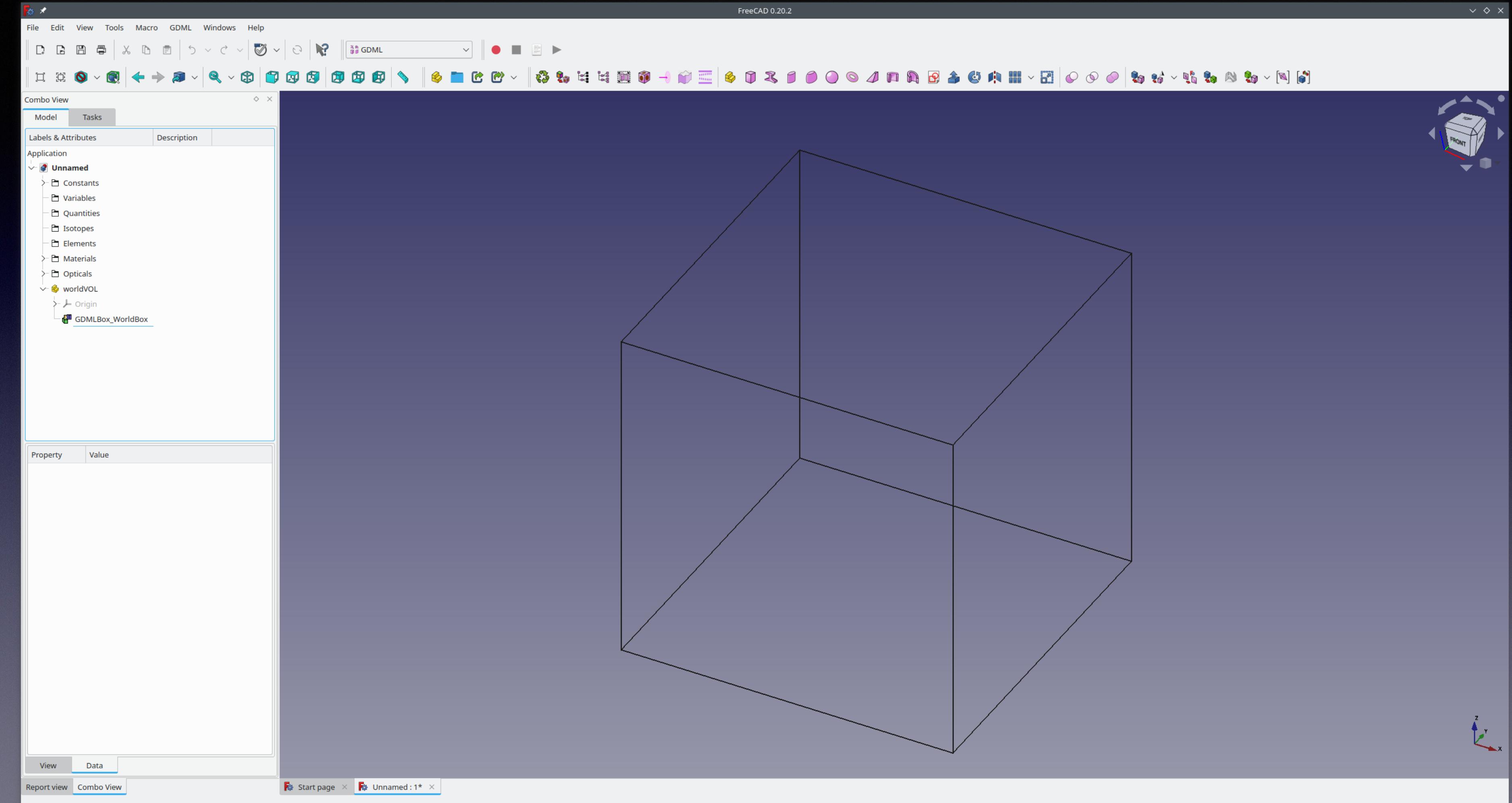
1. Launch FreeCAD
2. Select GDML Workbench

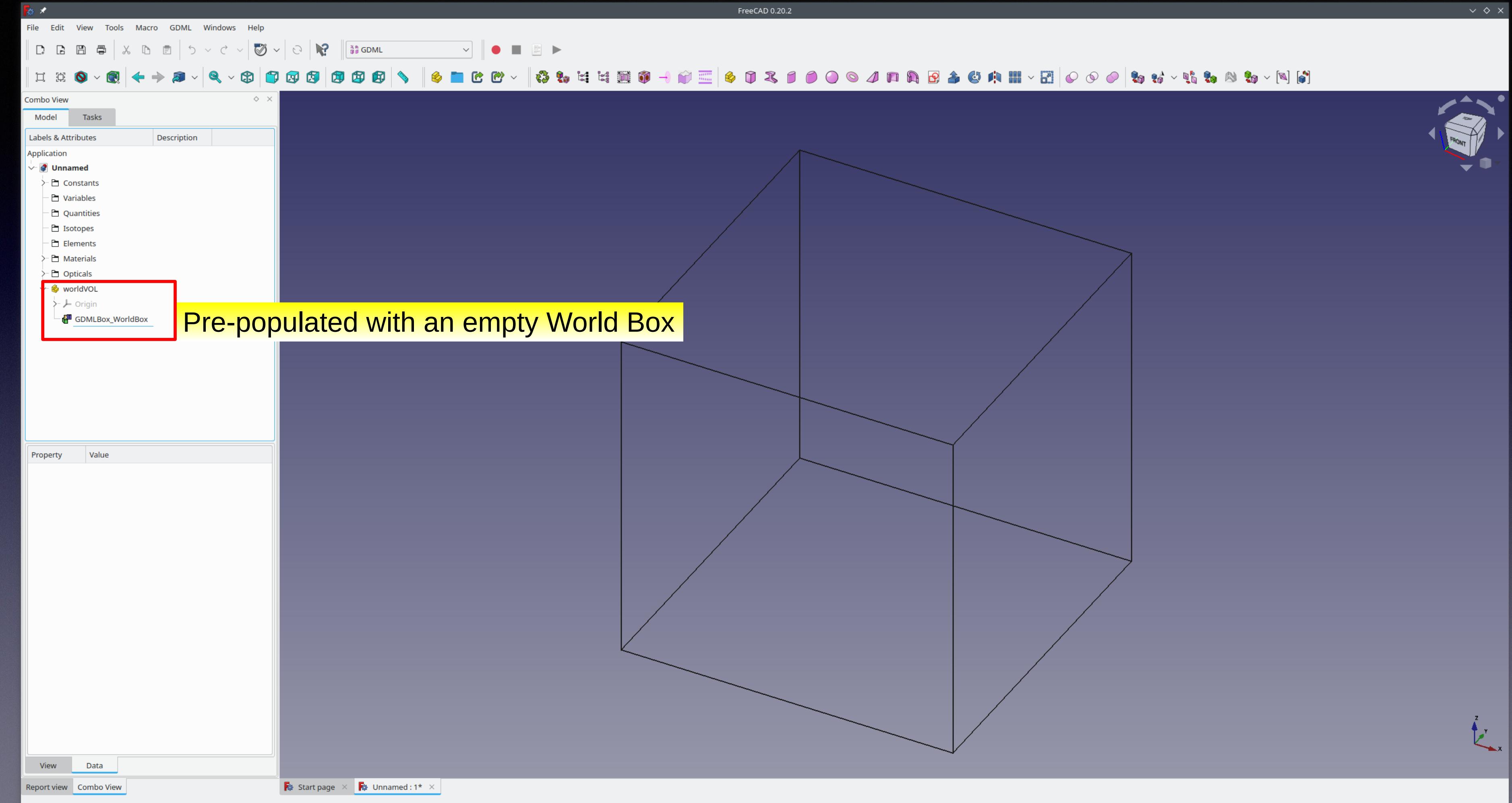


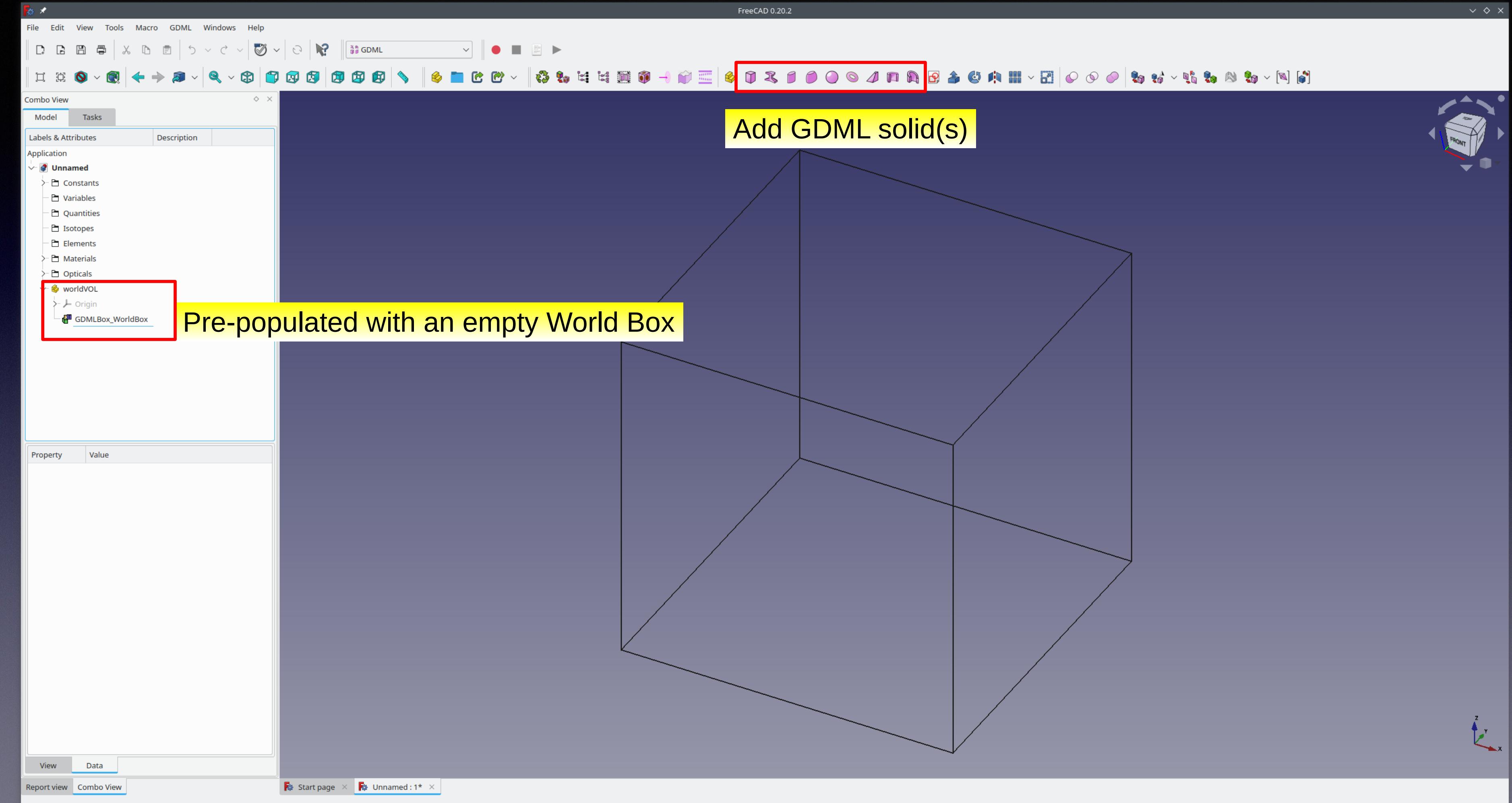
# Creating a gdml file from scratch, continued

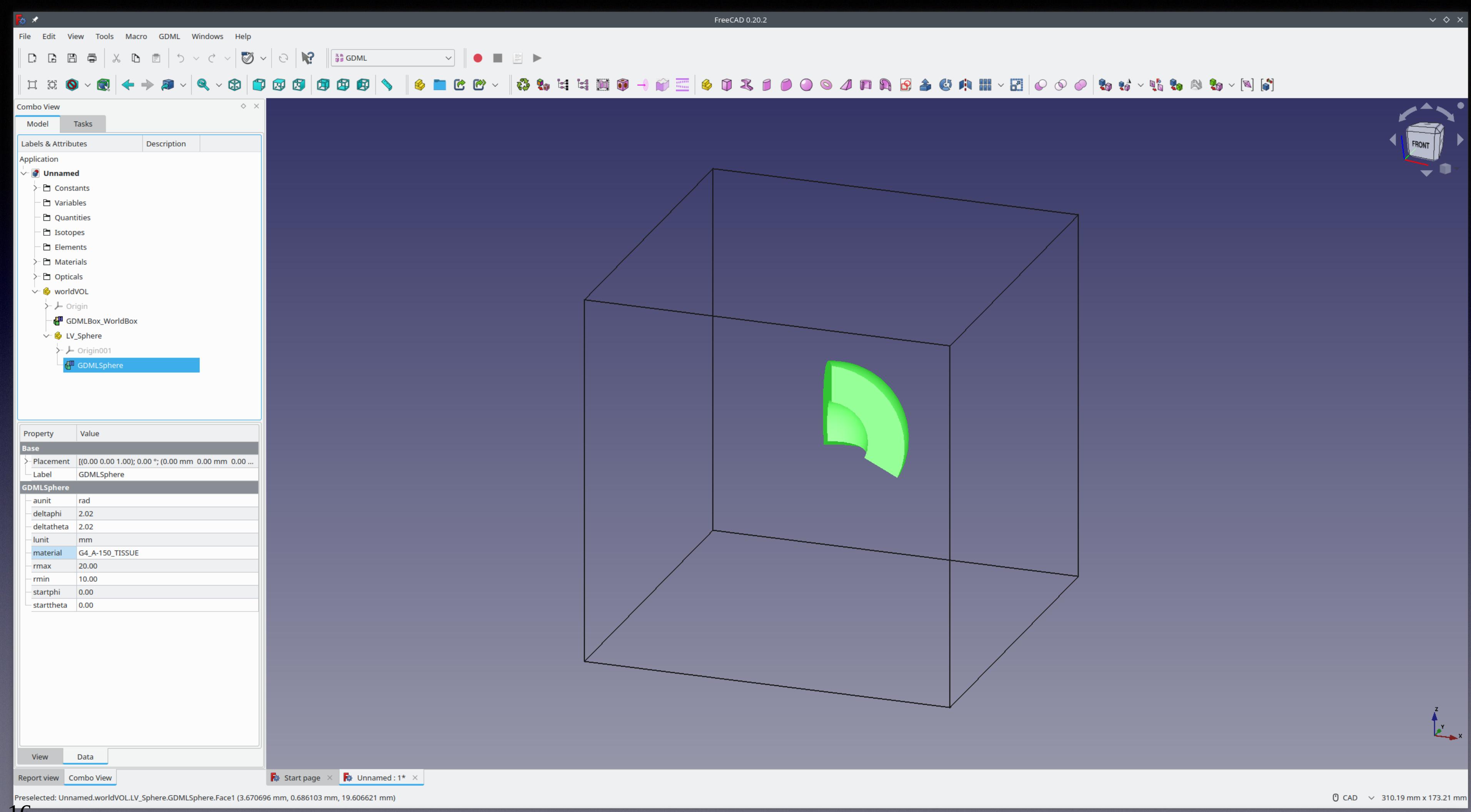
## 3. Create New document

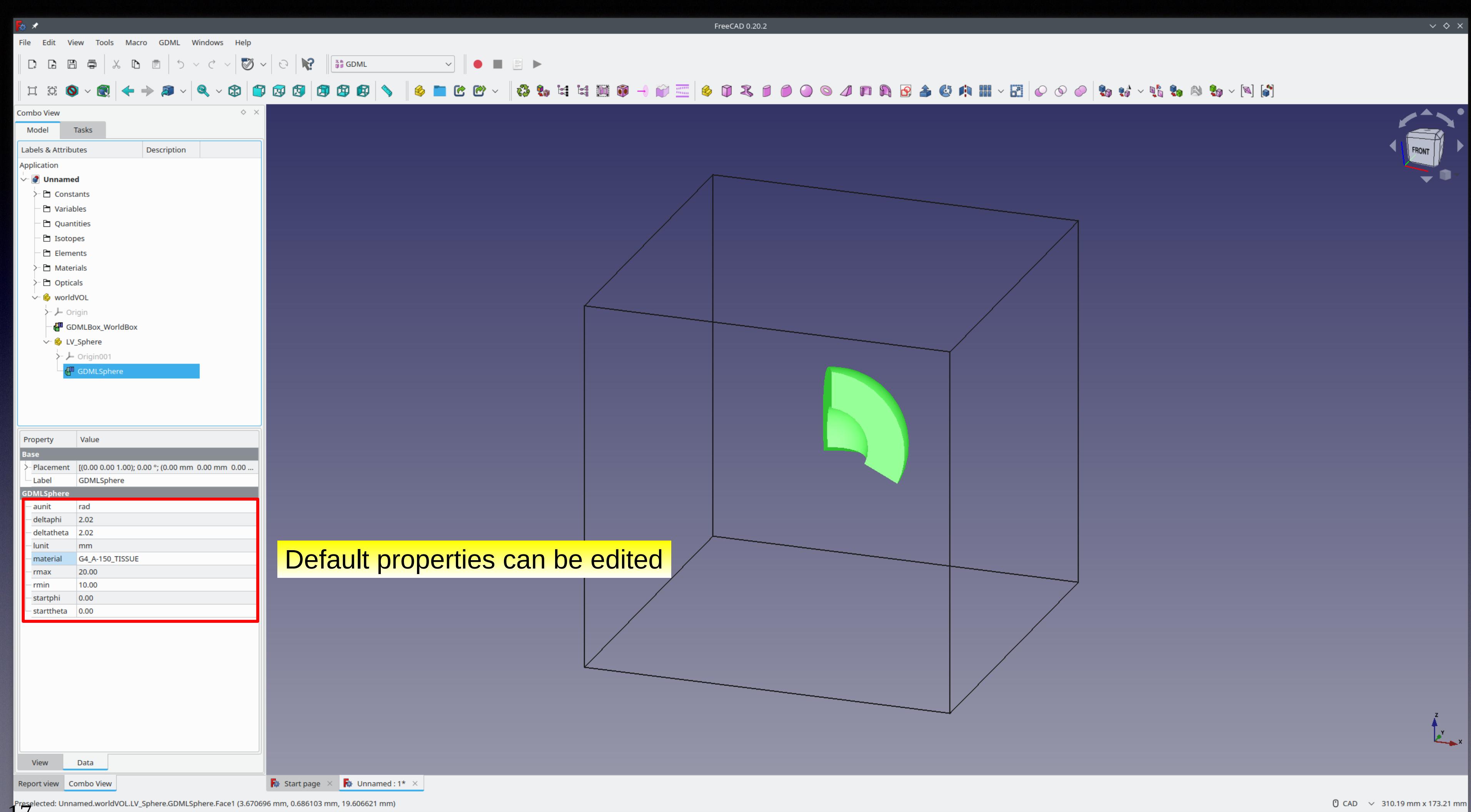












FreeCAD 0.20.2

File Edit View Tools Macro GDML Windows Help

Combo View

Model Tasks

Labels & Attributes Description

Application

- Unnamed
  - Constants
  - Variables
  - Quantities
  - Isotopes
  - Elements
  - Materials
  - Opticals
- worldVOL
  - Origin
  - GDMLOBox\_WorldBox
    - LV\_Sphere
      - Origin001
      - GDMLSphere

This tree element in the document exports to:

- a) a <solid>
- b) a <volume>
- c) a <physvol>

Default properties can be edited

Property Value

Base

- Placement [(0.00 0.00 1.00); 0.00 °; (0.00 mm 0.00 mm 0.00 ...)
- Label GDMLSphere

GDMLSphere

aunit	rad
deltaphi	2.02
deltatheta	2.02
lunit	mm
material	G4_A-150_TISSUE
rmax	20.00
rmin	10.00
startphi	0.00
starttheta	0.00

View Data

Report view Combo View Start page Unnamed : 1\*

Preselected: Unnamed.worldVOL.LV\_Sphere.GDMLSphere.Face1 (3.670696 mm, 0.686103 mm, 19.606621 mm)

0 CAD 310.19 mm x 173.21 mm

The screenshot shows the FreeCAD 0.20.2 interface with the 'GDML' workspace active. On the left, the 'Combo View' window displays the document tree under the 'Model' tab. A red box highlights the 'GDMLSphere' entry under 'LV\_Sphere'. Below it, the properties panel shows various parameters for the sphere, with a red box highlighting the 'material' field set to 'G4\_A-150\_TISSUE'. A blue arrow points from a yellow callout box containing text about export options to the 'GDMLSphere' entry in the tree. Another yellow callout box contains the text 'Default properties can be edited' and points to the properties panel. The main workspace shows a 3D view of a green sphere centered within a gray wireframe cube representing the world volume. A coordinate system is visible in the bottom right corner.

# Materials can be added in two ways:

The screenshot shows a CAD-like interface for managing 3D models. A large 3D view window in the center displays a gray rectangular volume containing a green semi-spherical object. To the left of the main window is a 'Combo View' panel. The top part of this panel is a tree view labeled 'Labels & Attributes' under 'Model'. It shows a hierarchy: Variables, Quantities, Isotopes, Elements, Materials (which is expanded to show Geant4, worldVOL, and Origin), and GDMLBox\_WorldBox. Below this is a 'Property' table with a single row for 'Base'. The 'Value' column shows 'Placement [(0.00 0.00 1.00); 0.00 °; (0.00 mm 0.00 mm 0.00 mm)]' and 'Label GDMLSphere'. Under 'GDMLSphere', there are several properties: aunit (rad), deltaphi (2.02), deltatheta (2.02), lunit (mm), material (highlighted with a red border), rmax (highlighted with a red border), rmin, startphi, starttheta, and others. The 'material' dropdown contains a list of materials, with 'G4\_A-150\_TISSUE' selected. The 'rmax' dropdown also contains 'G4\_A-150\_TISSUE' and other options like G4\_ACETONE, G4\_ACETYLENE, etc. At the bottom of the 'Data' tab in the 'Combo View' panel, there are tabs for 'View' and 'Data', with 'Data' being the active tab. The status bar at the bottom shows 'Report view', 'Combo View', 'Start page', and 'Unnamed : 1\*'. On the right side of the interface, there is a small 3D preview showing a cube with a 'FRONT' label and a coordinate system (x, y, z). A yellow callout box points to the 'material' dropdown in the 'Data' tab, with the text '1) From drop list in 'material' property'.

1) From drop list in 'material' property

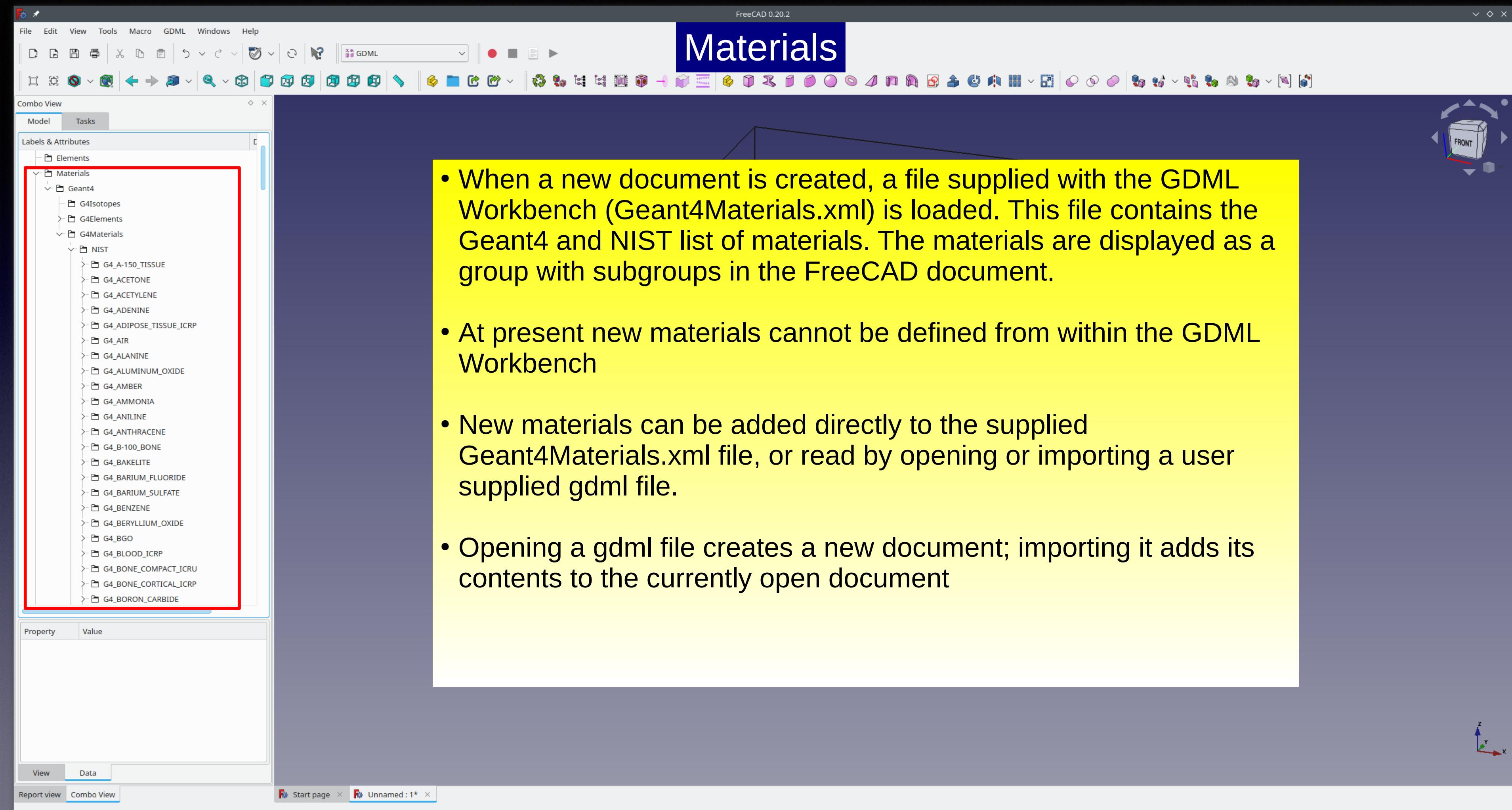
# Materials can be added in two ways:

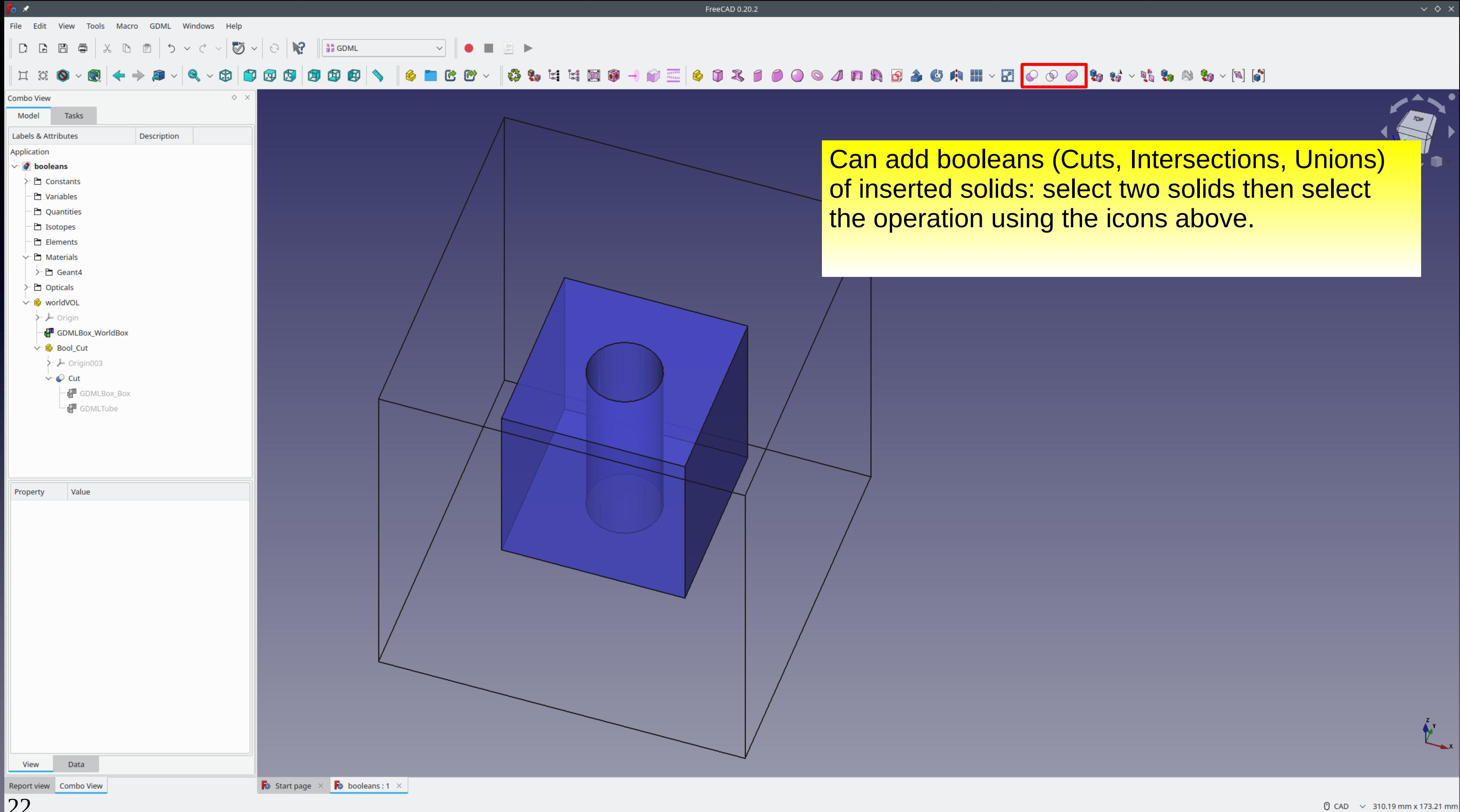
2) Using 'set material' icon.

Allows incremental search

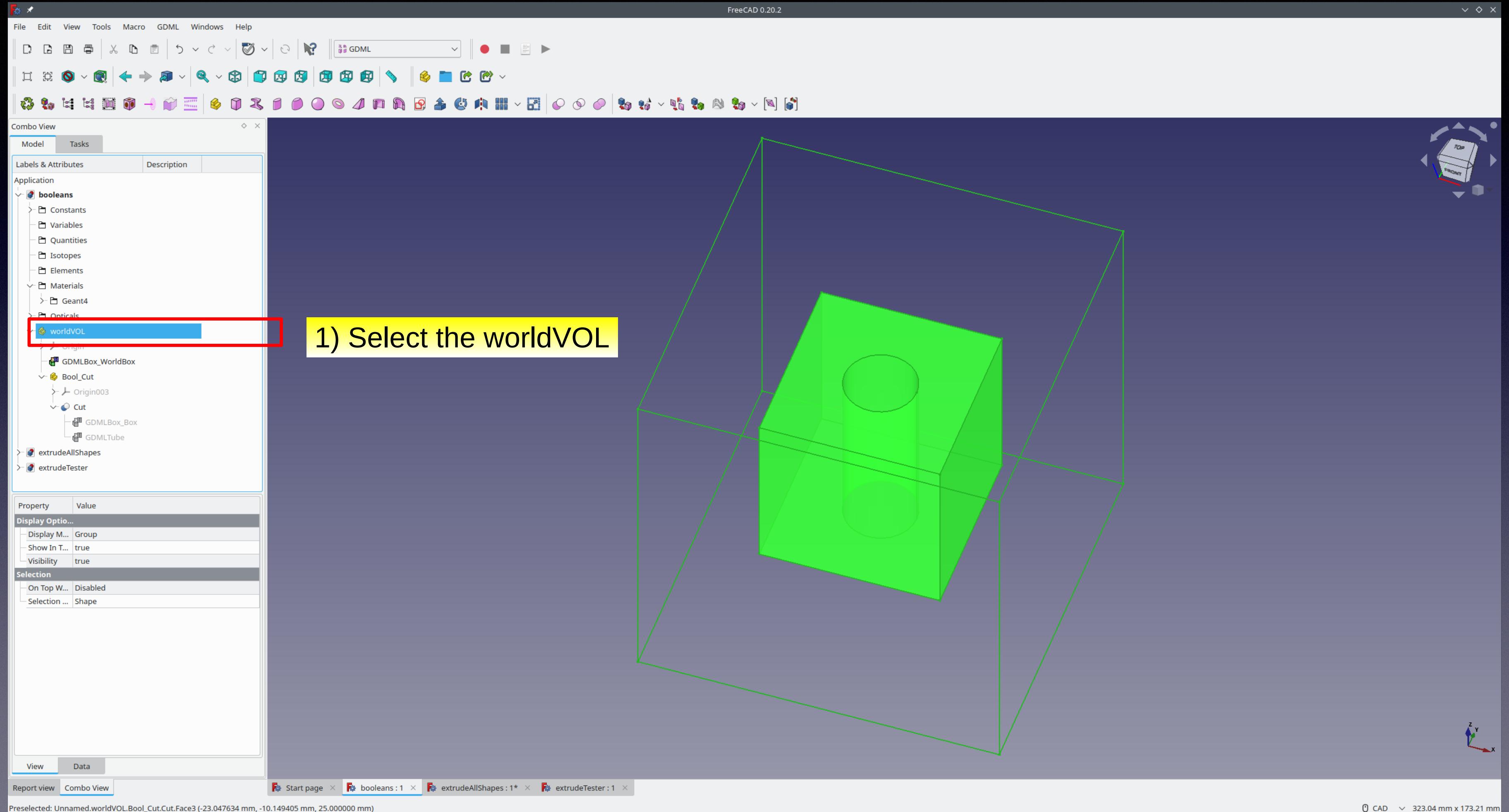
Allows adding material property to non-GDML solids.

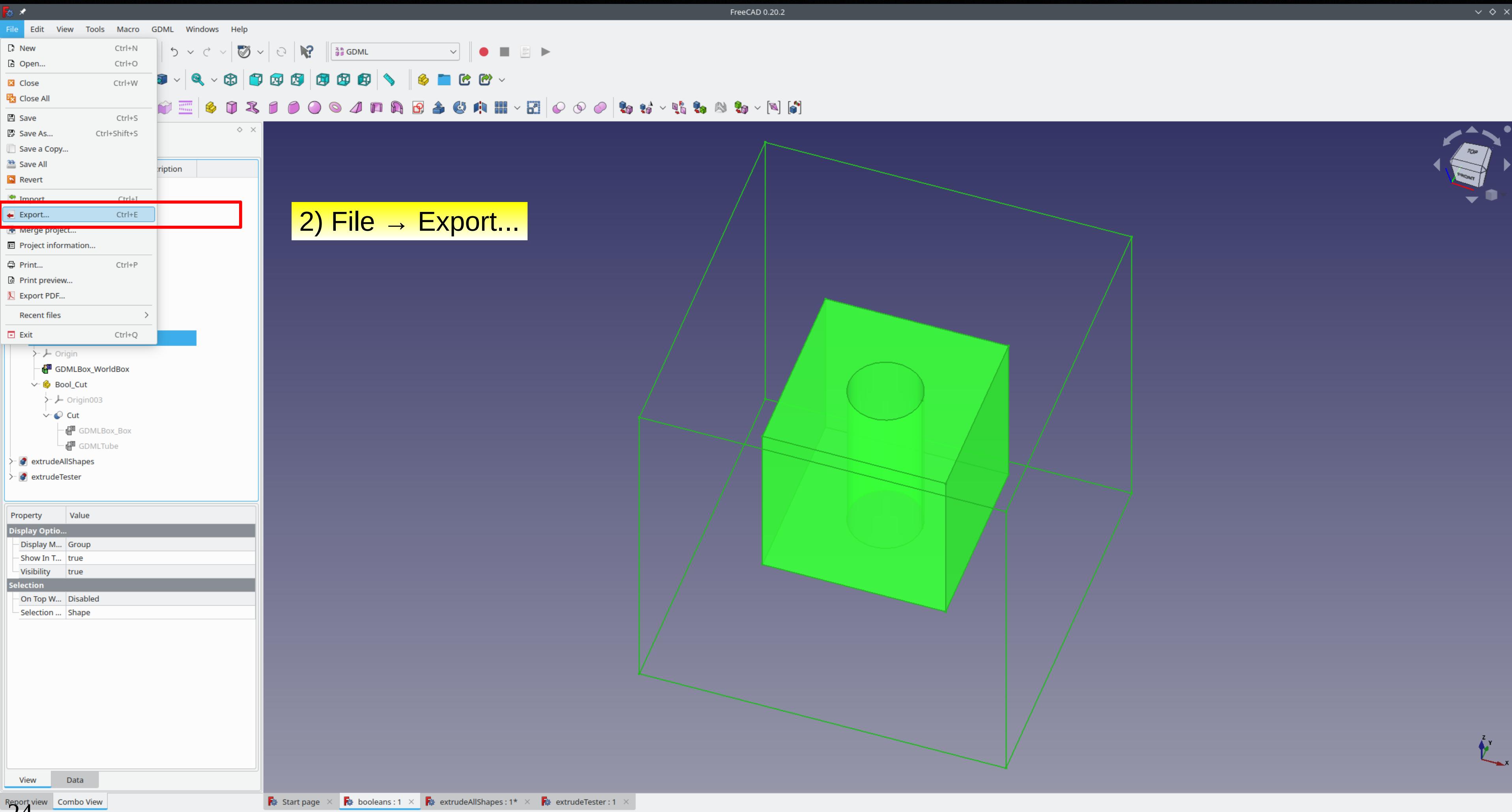
The screenshot shows the Geant4 Workbench interface. On the left, there's a tree view of the model structure under 'Model'. A red box highlights the 'Materials' section, and a red arrow points from the 'Set GDML Material' icon in the toolbar to this box. The central workspace displays a 3D scene with a large gray cube and a smaller gray sphere. A green slice is visible on the sphere. A 3D coordinate system (x, y, z) is at the bottom right. The bottom status bar shows various application icons and the time '1:04 PM 11/9/23'.

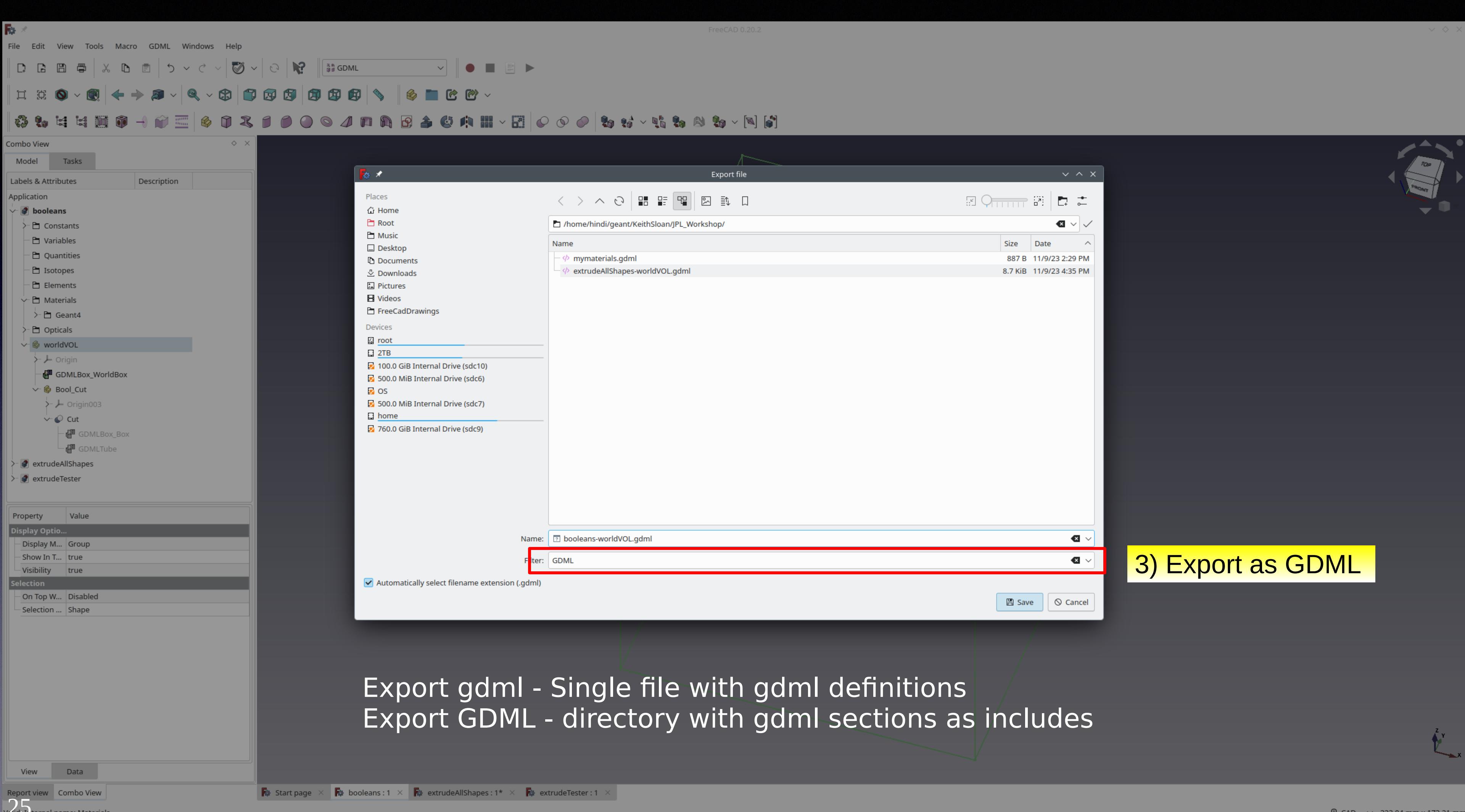




# Export the document to gdml:



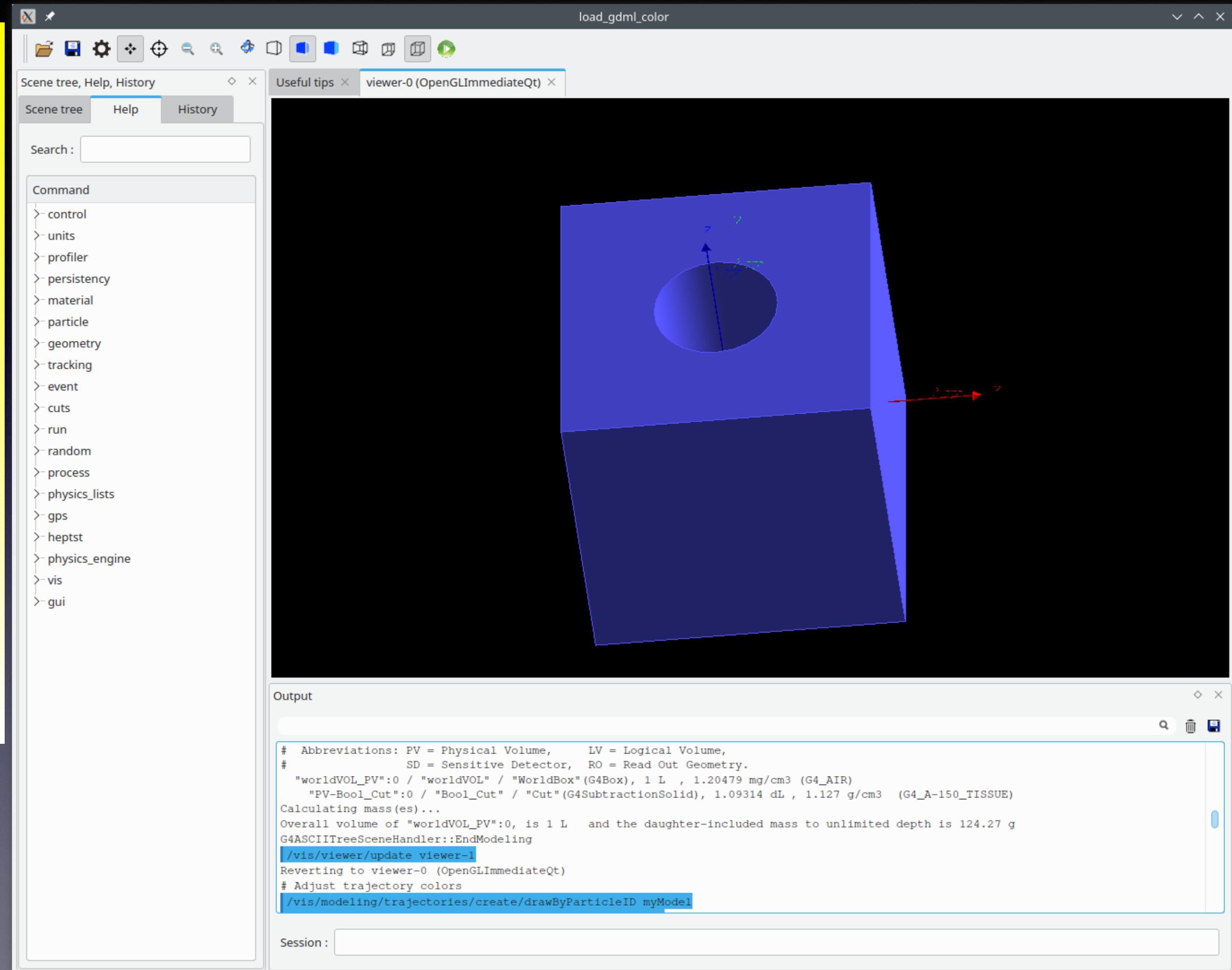




Can view the exported file using the `load_gdml` application provided with Geant4 in `examples/persistency/gdml/G01`

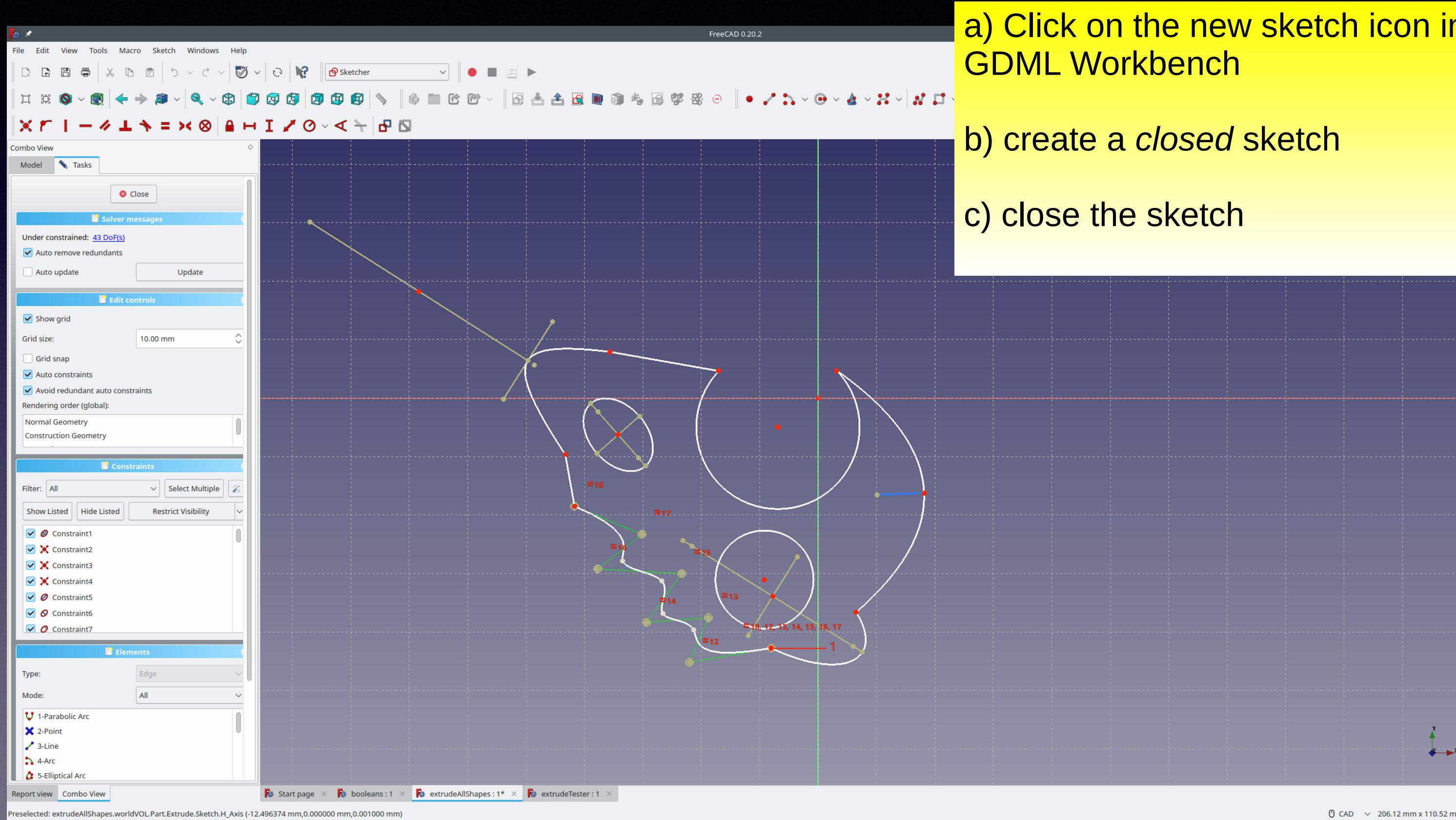
We export the FreeCAD color of solids by adding an `<auxiliary auxtype="Color">` tag to each of the `<volumes>` in the gdml file.

We have a modified version of `load_gdml`, `load_gdml_color` that processes that auxiliary tag. We can add the modified file to the github if there is interest.



# Adding non-GDML solids to the GDML document

## 1. Extrusions

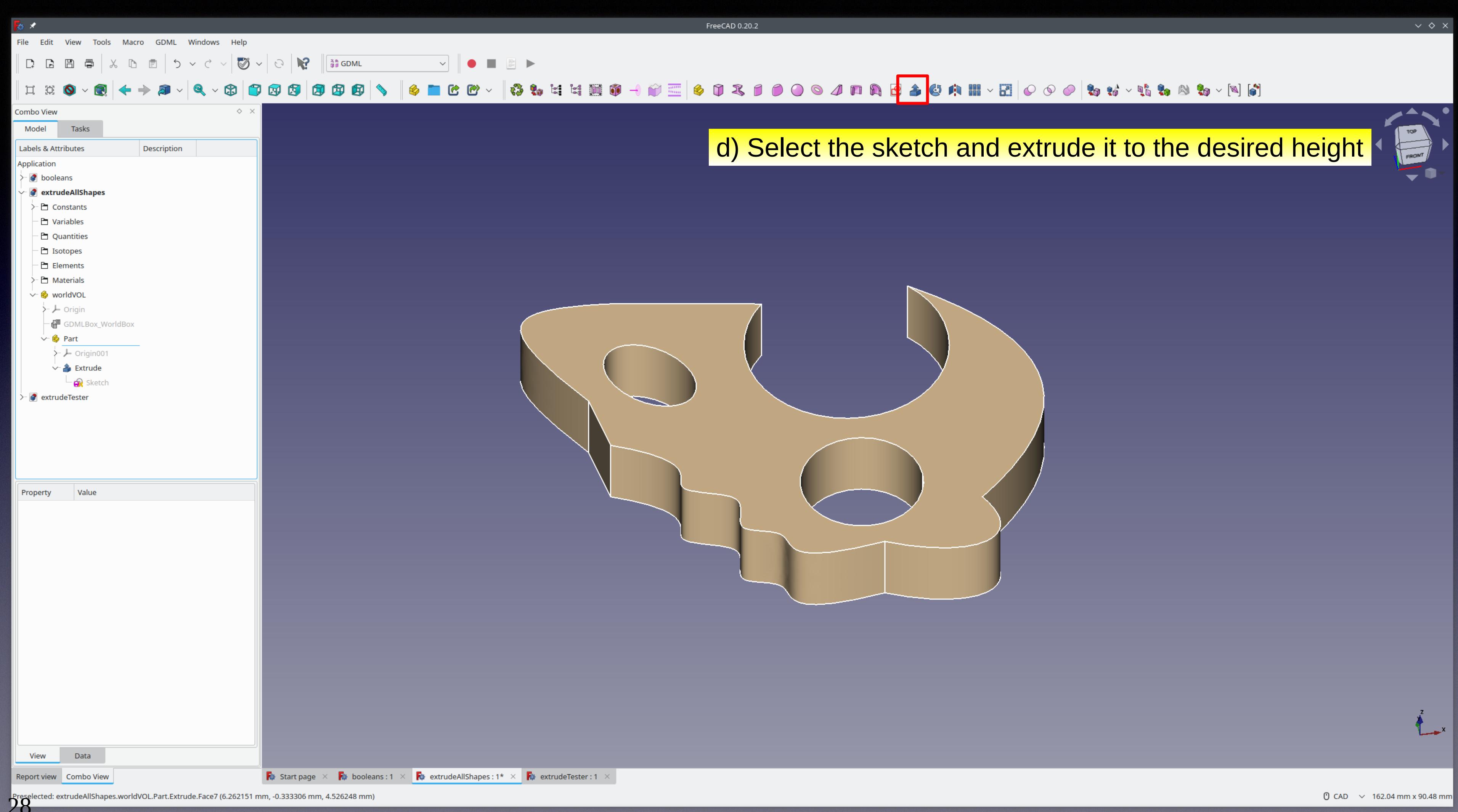


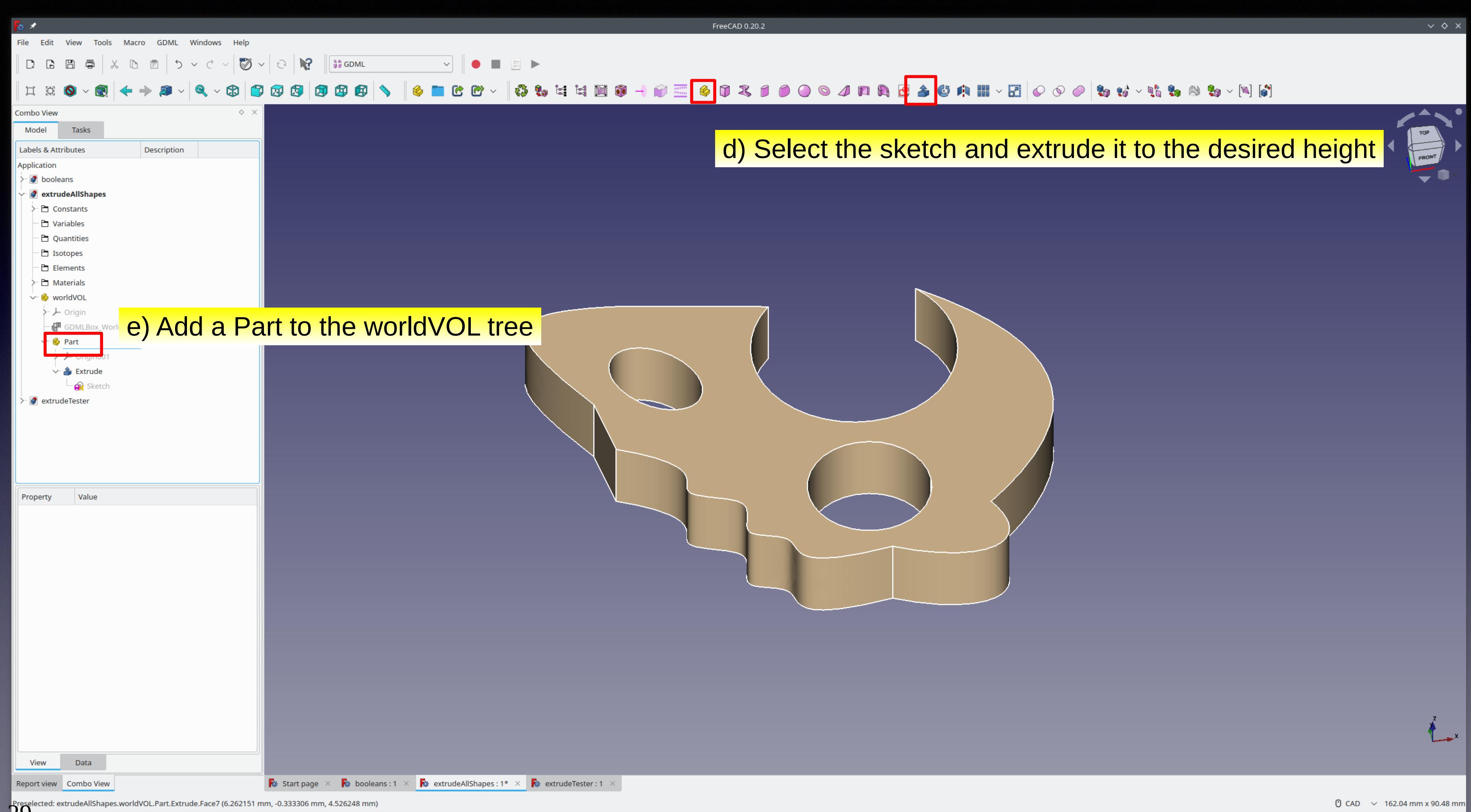
a) Click on the new sketch icon in the GDML Workbench

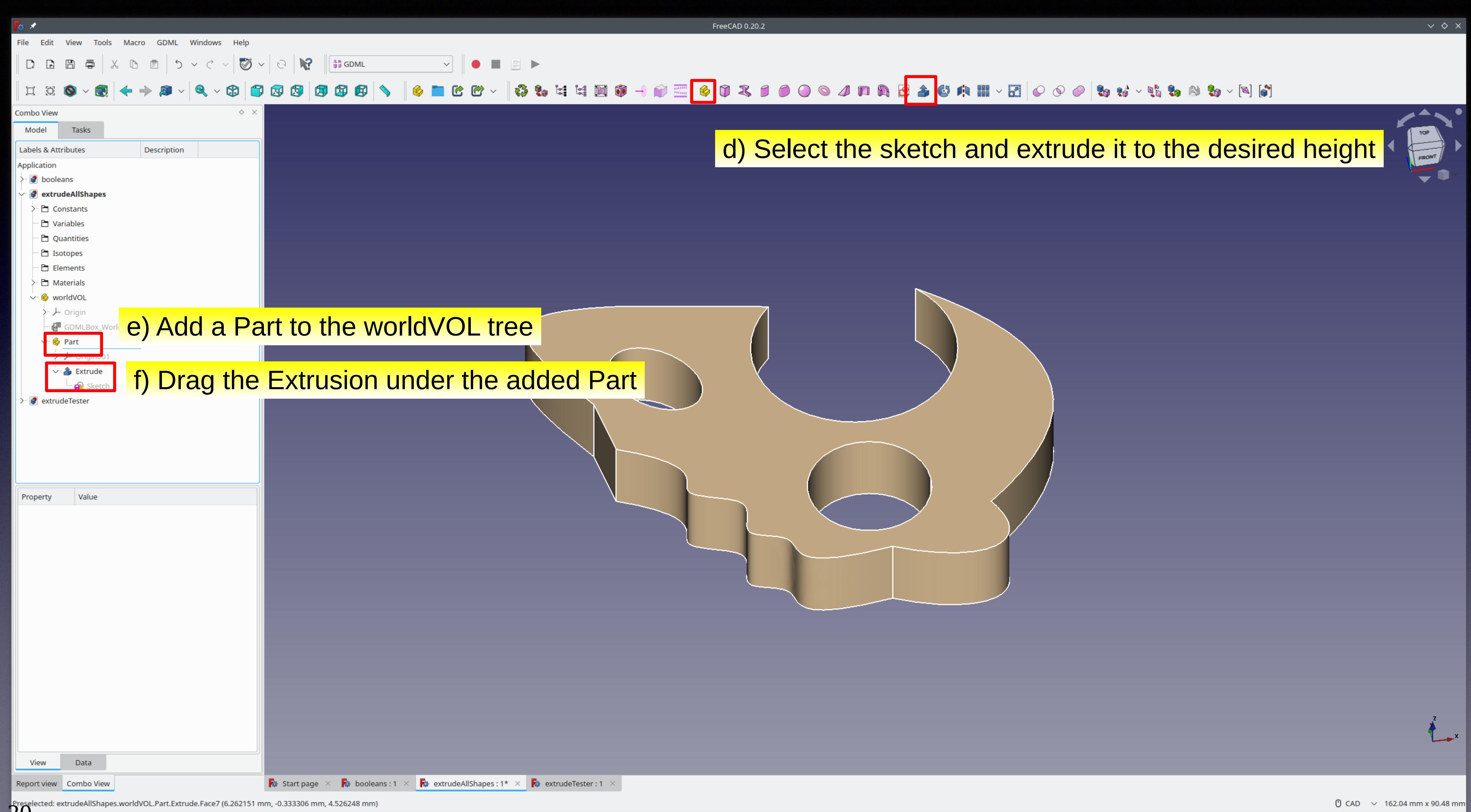
b) create a *closed* sketch

c) close the sketch





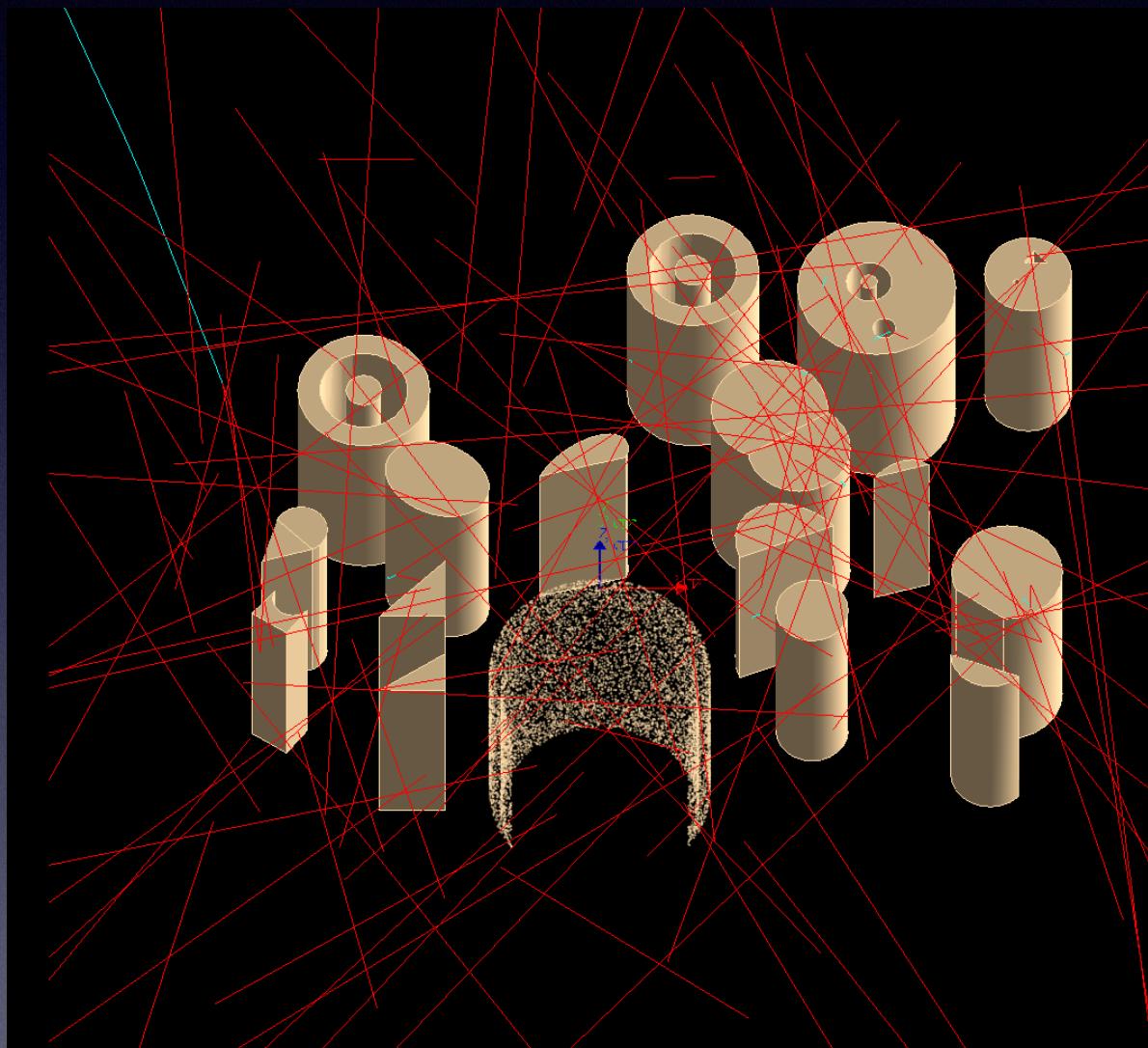




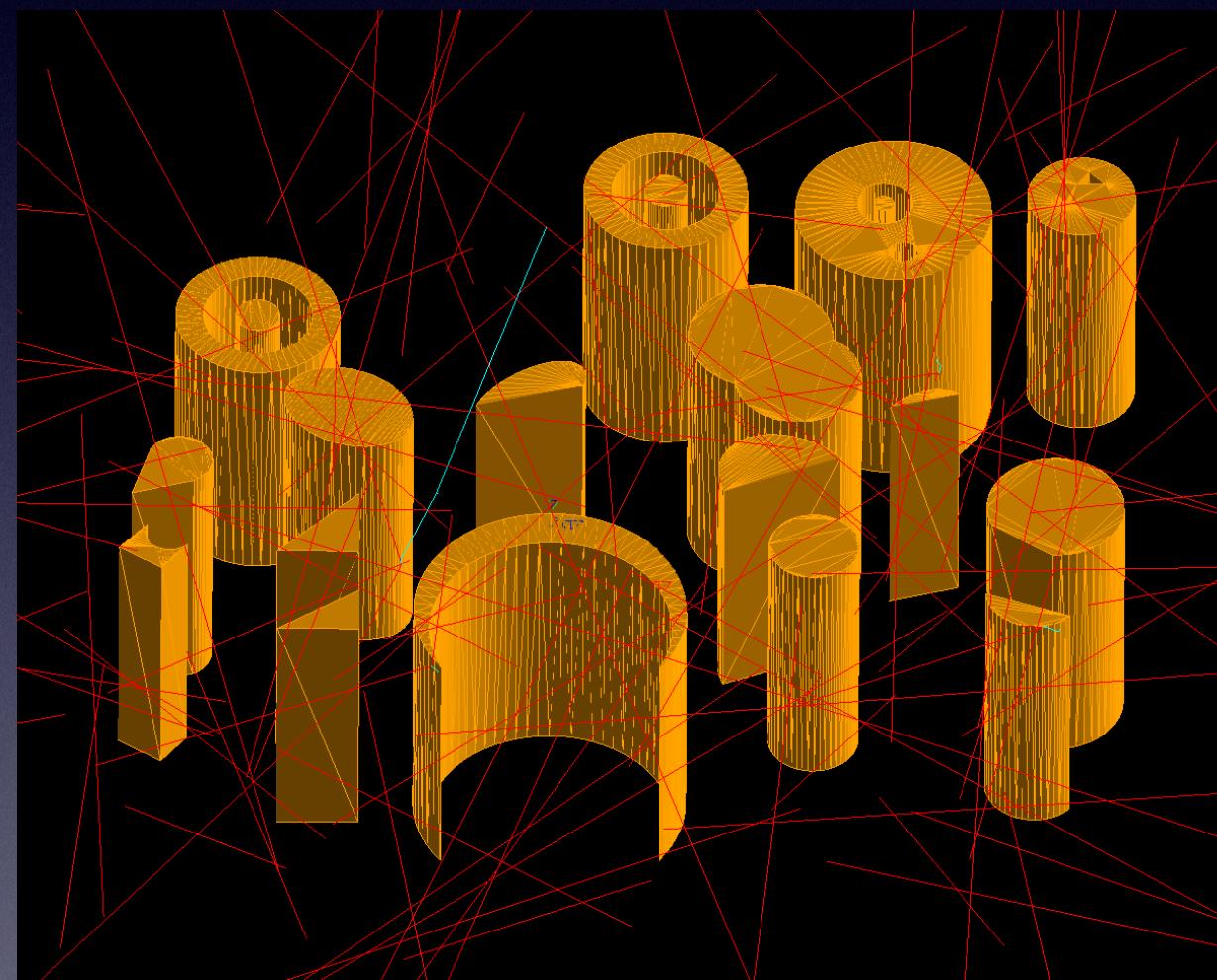
# Adding non-GDML solids to the GDML document

## 1. Extrusions

To the extent possible, extrusions are exported as booleans of GDML solids (Circles → Tubes, Polygons → Xtru, Ellipses → ElTube, ...). We believe this produces faster Geant4 simulations than conversion to a Tessellation:



Exported as booleans of Solids



Exported as tessellations

**With  $10^6$  geantinos:**

**extrusions: 7.25 s**

**tesselation: 17.35 s**

**(Factor of 2.4 faster with  
booleans)**

**With  $10^6$  2 MeV gamma rays:**

**extrusion: 17.35 s**

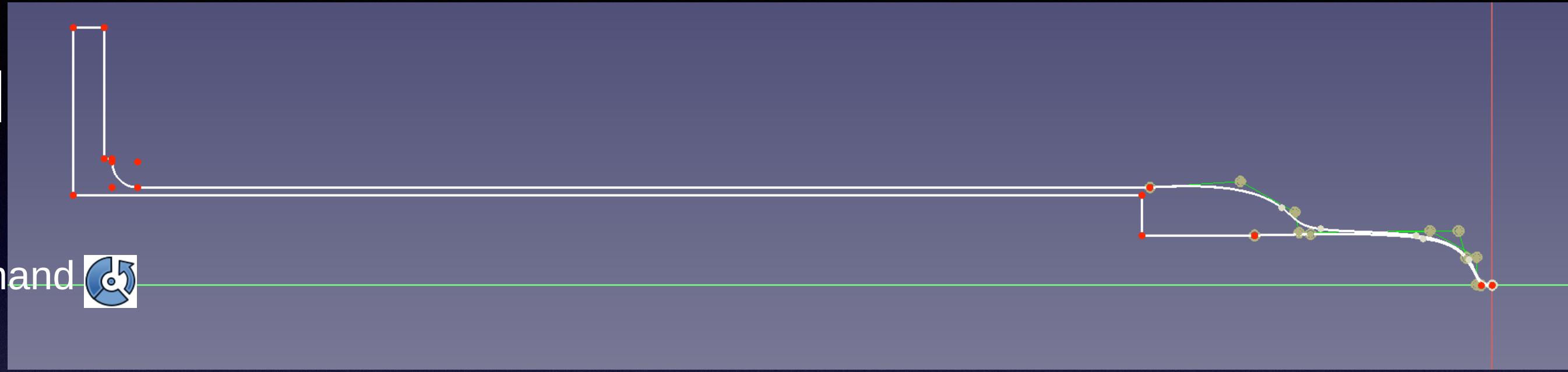
**tessellation: 146 s**

**(Factor of 8.4 faster with  
booleans)**

# Adding non-GDML solids to the GDML document

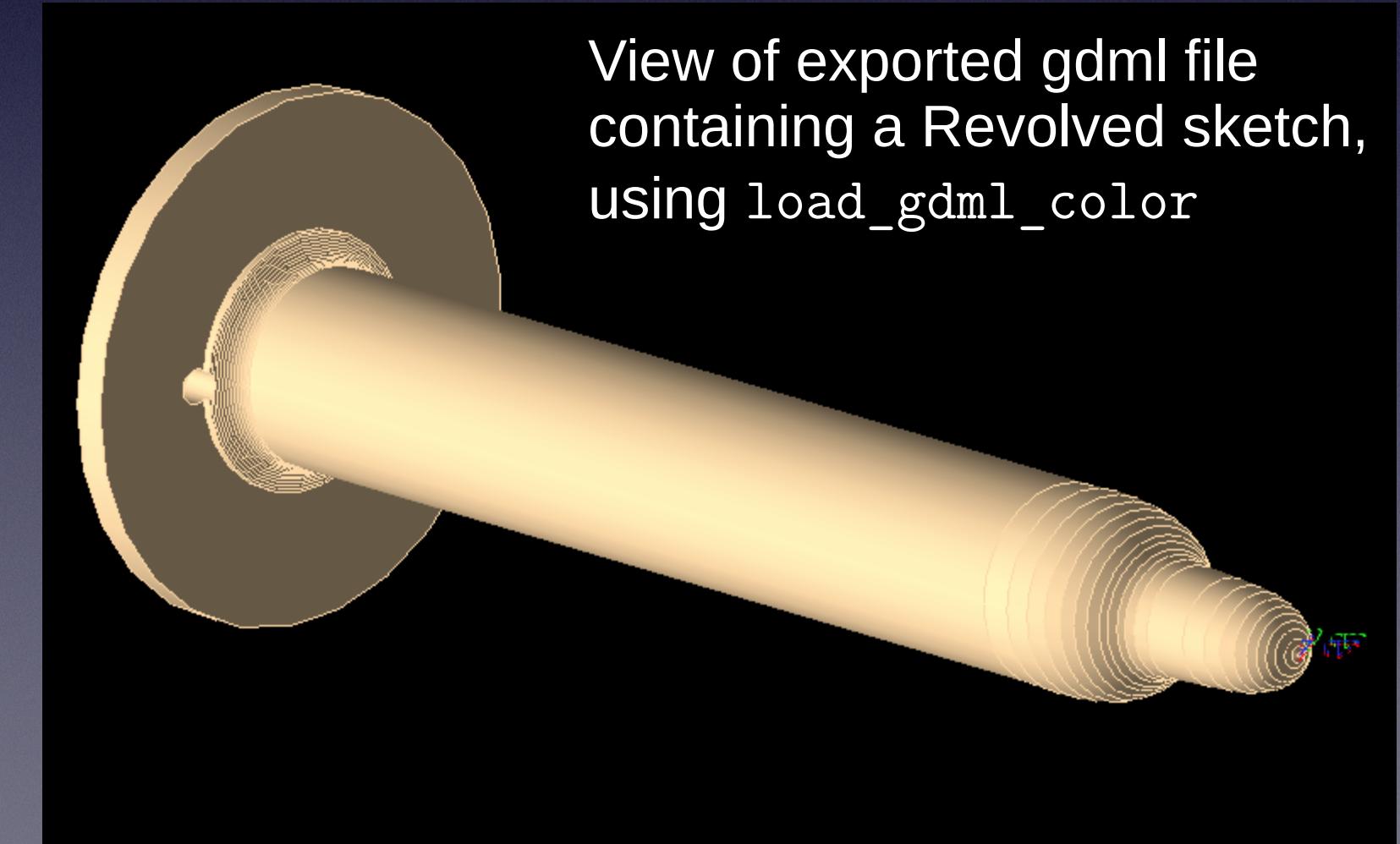
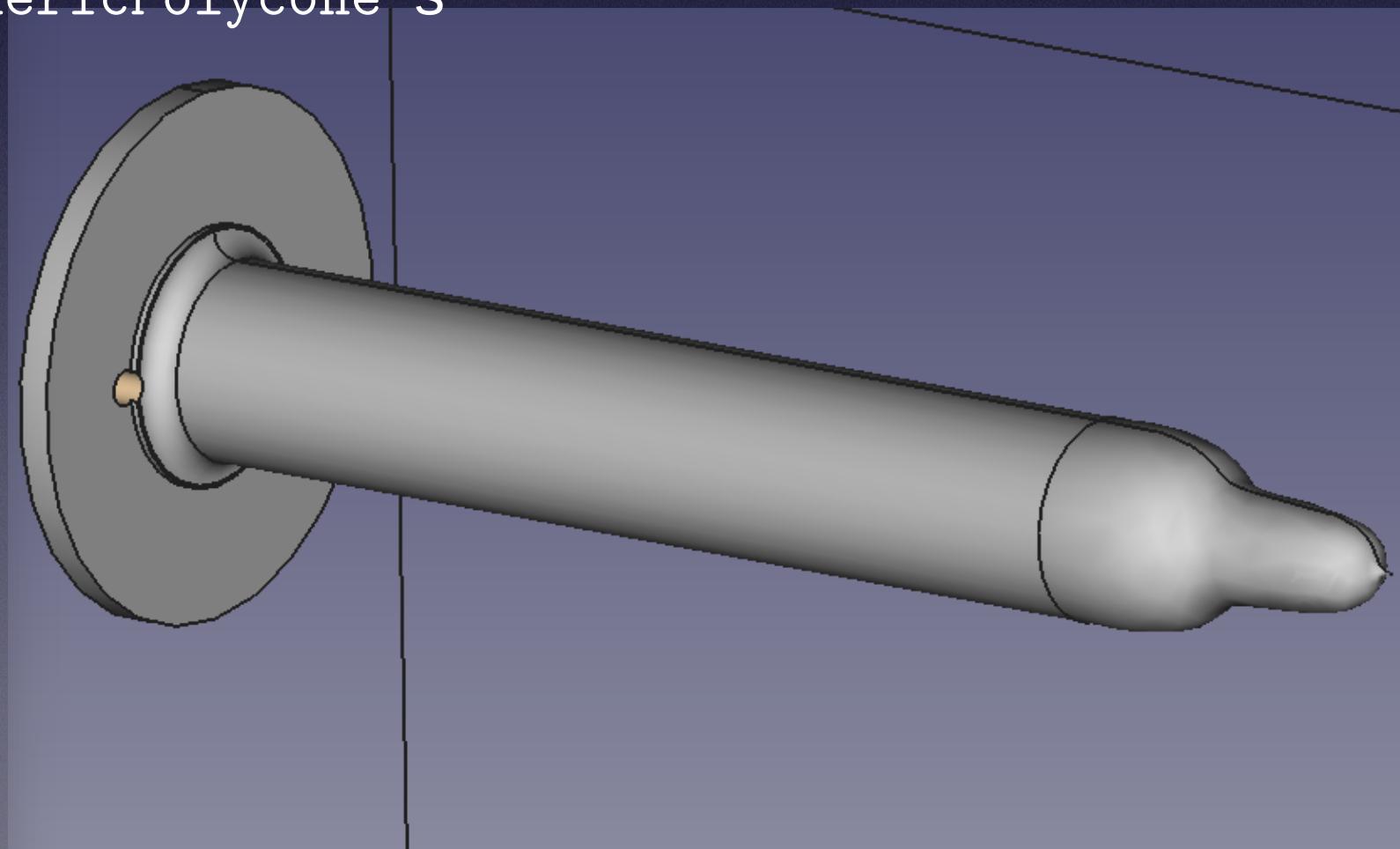
## 2. Revolve

Surfaces of revolution added by creating curve(s) in Sketcher, 



and then using the revolve command 

Revolves are exported as gdml genericPolycone's



View of exported gdml file containing a Revolved sketch, using load\_gdml\_color

# Adding non-GDML solids to the GDML document

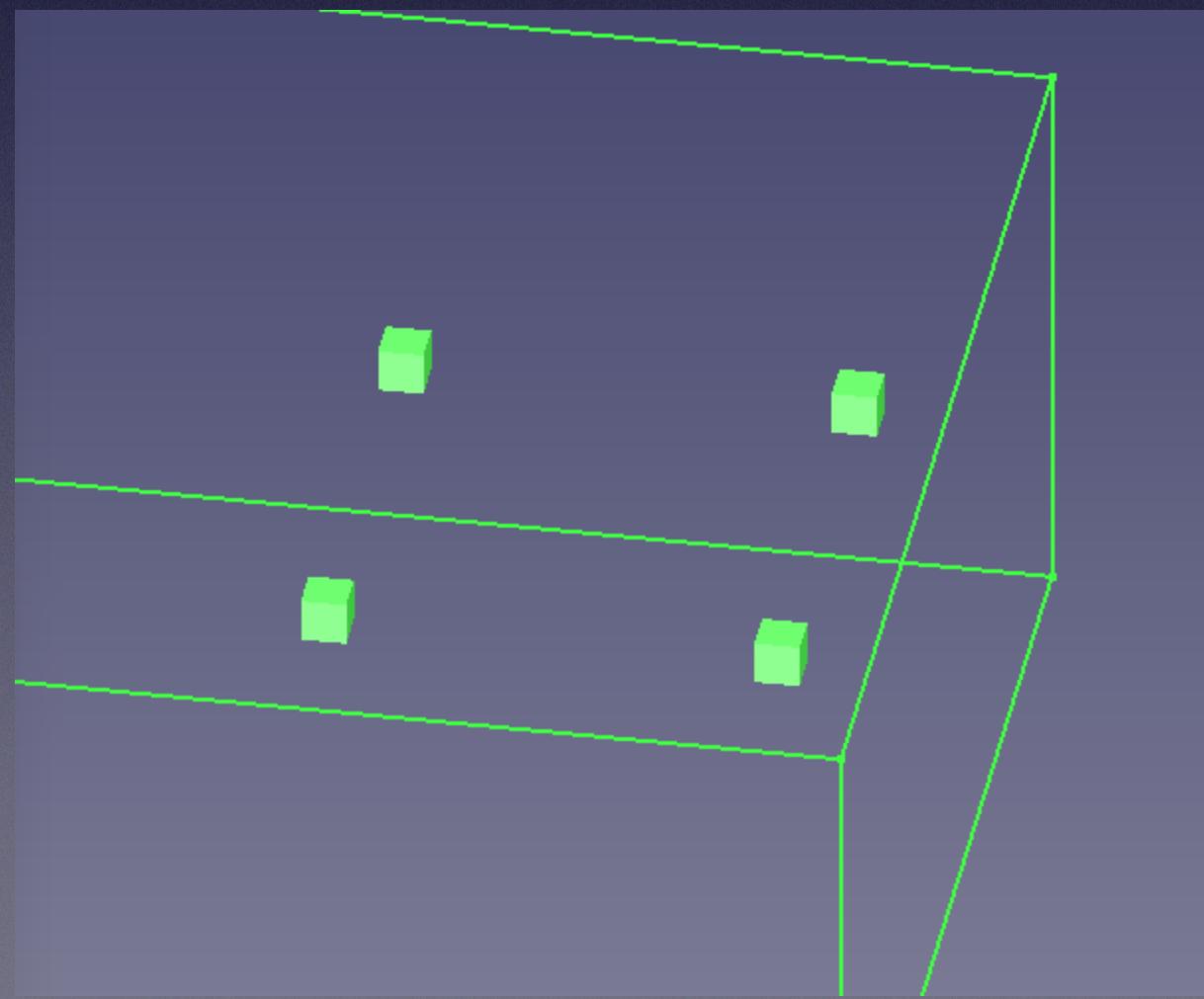
## 3. Arrays

### a. Orthogonal Arrays

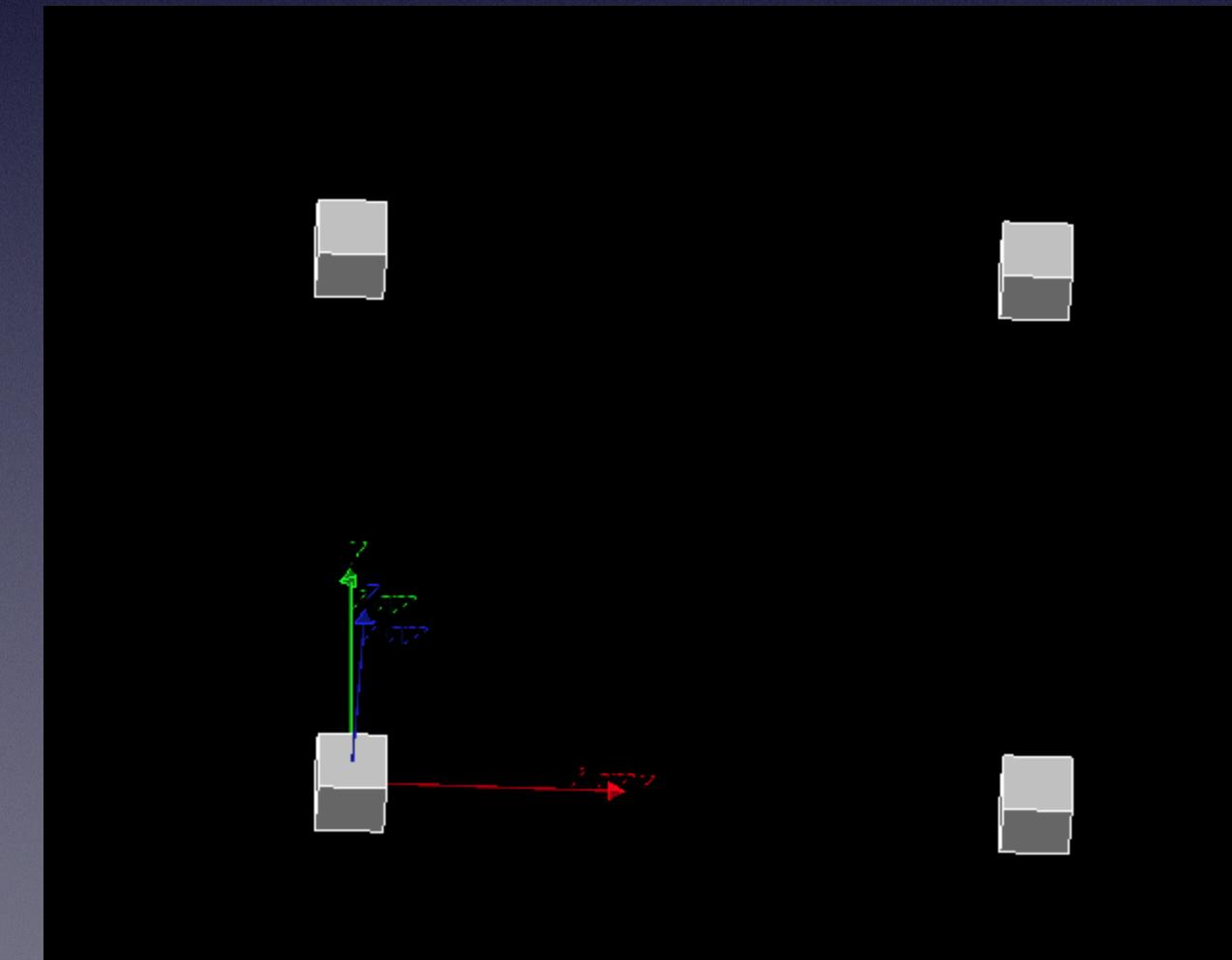
Orthogonal arrays are added by selecting the object and executing the orthogonal array command



Note: Arrays of solids are exported as a single multiunion solid and will all have the same material



View of exported gdml file containing an orthogonal (rectangular) array, using load\_gdml\_color

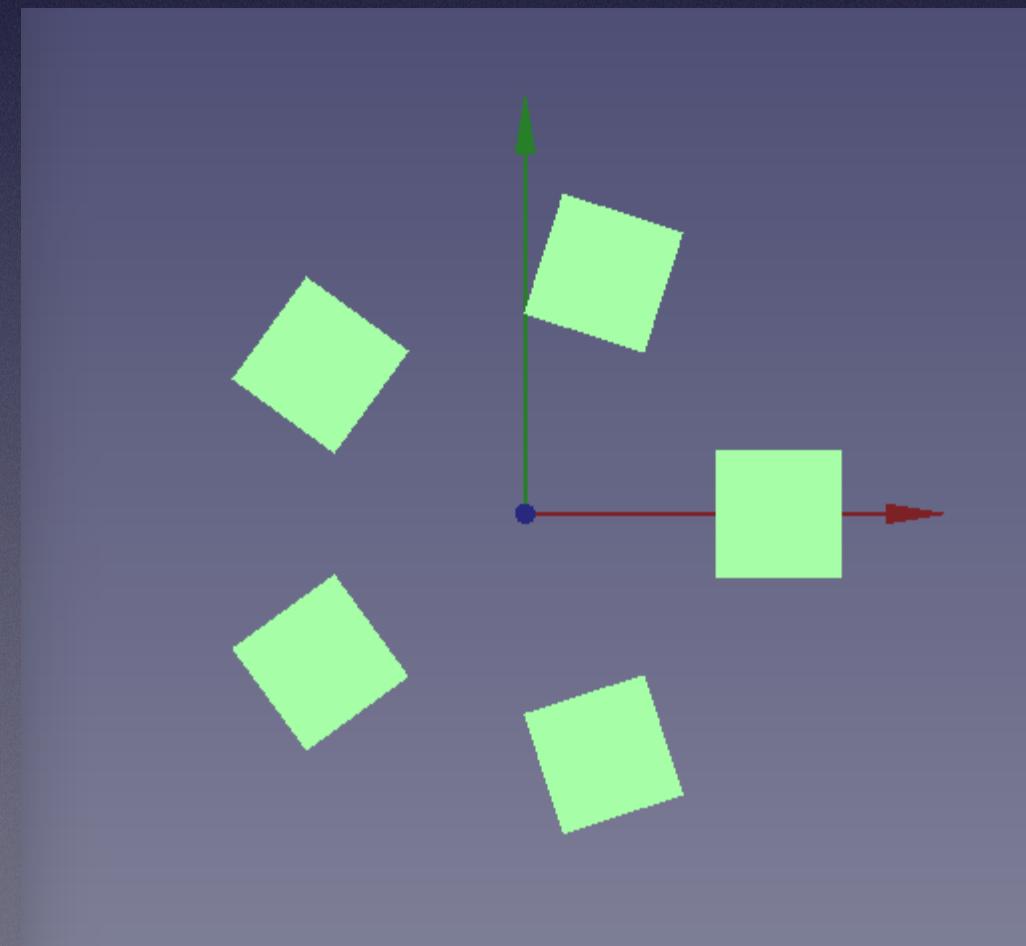


# Adding non-GDML solids to the GDML document

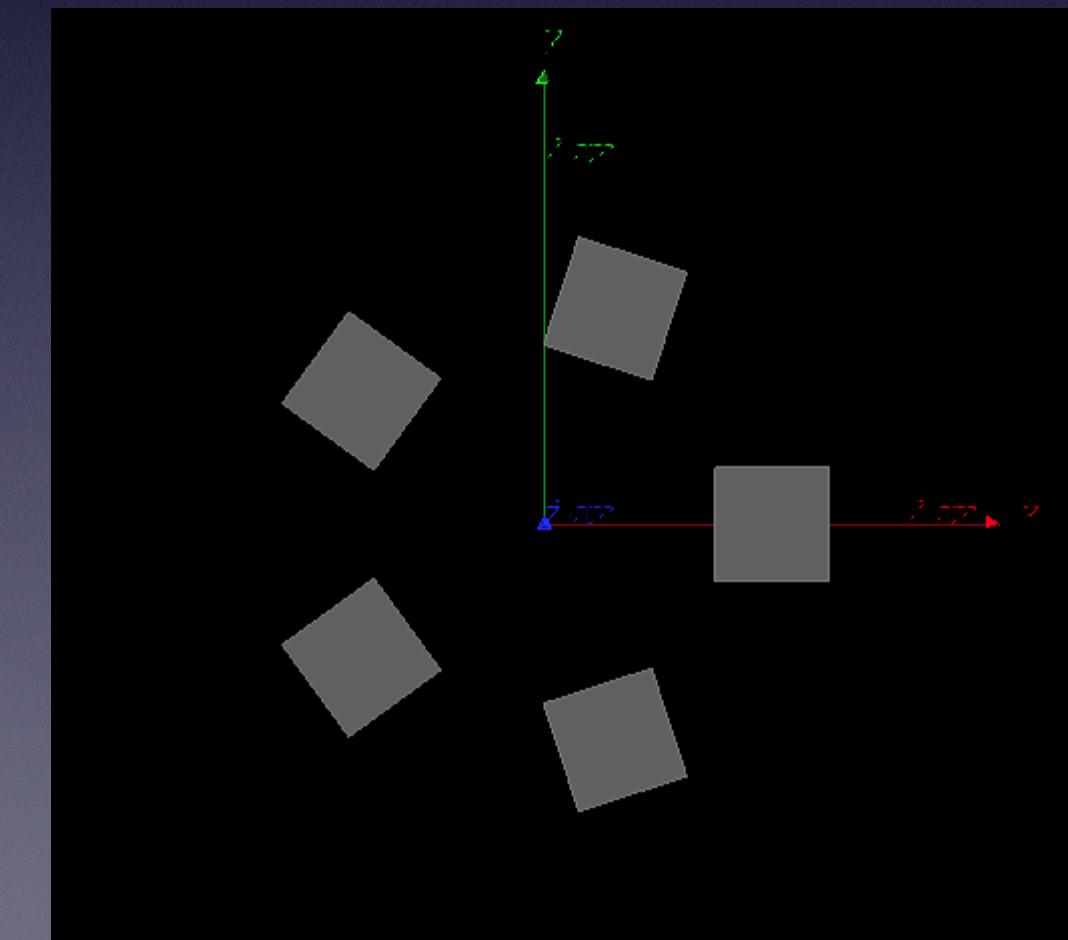
## 3. Arrays

### b. Polar Arrays

Polar arrays are added by selecting the object and executing the polar array command



View of exported gdml file containing a polar array, using `load_gdml_color`

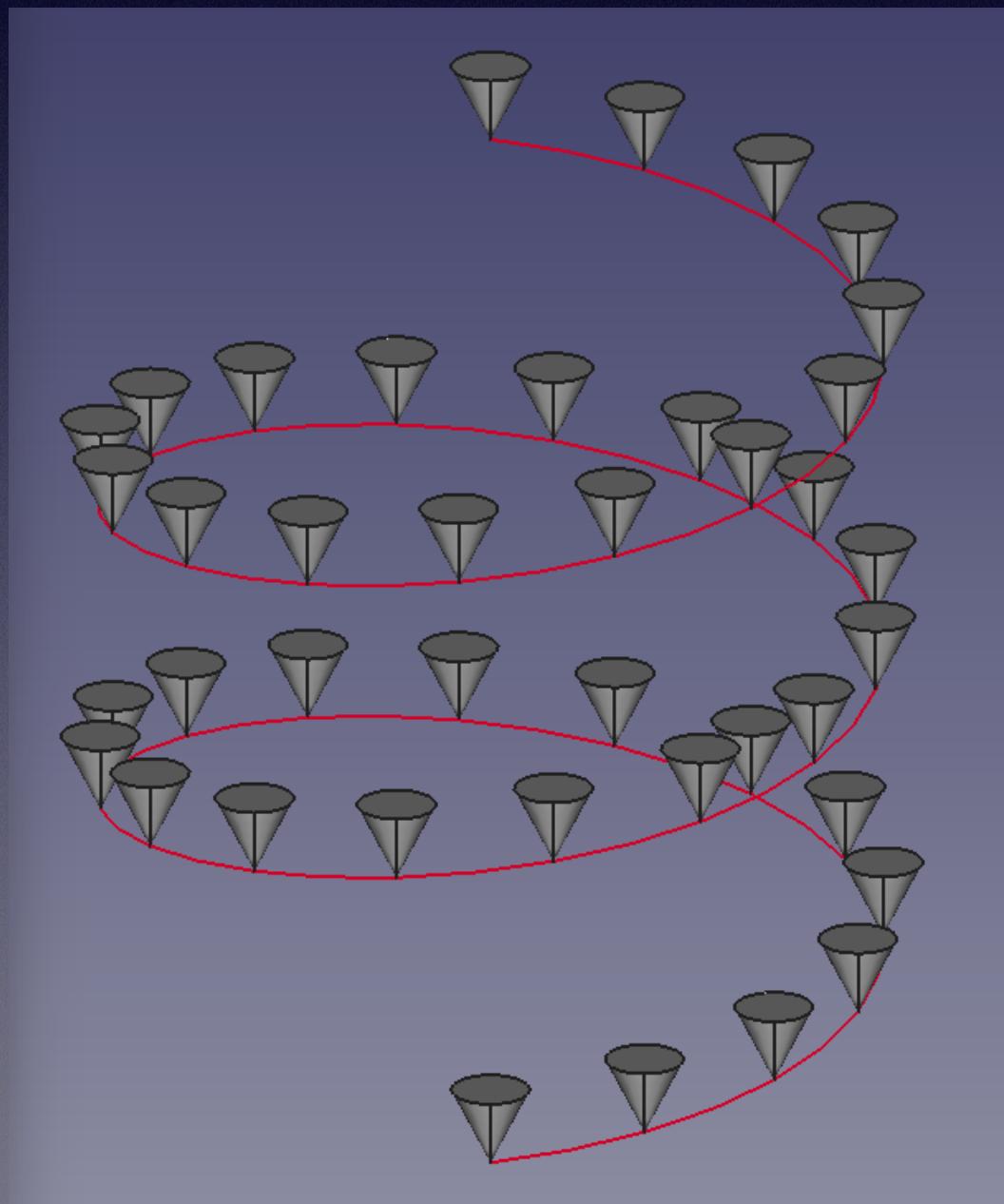


# Adding non-GDML solids to the GDML document

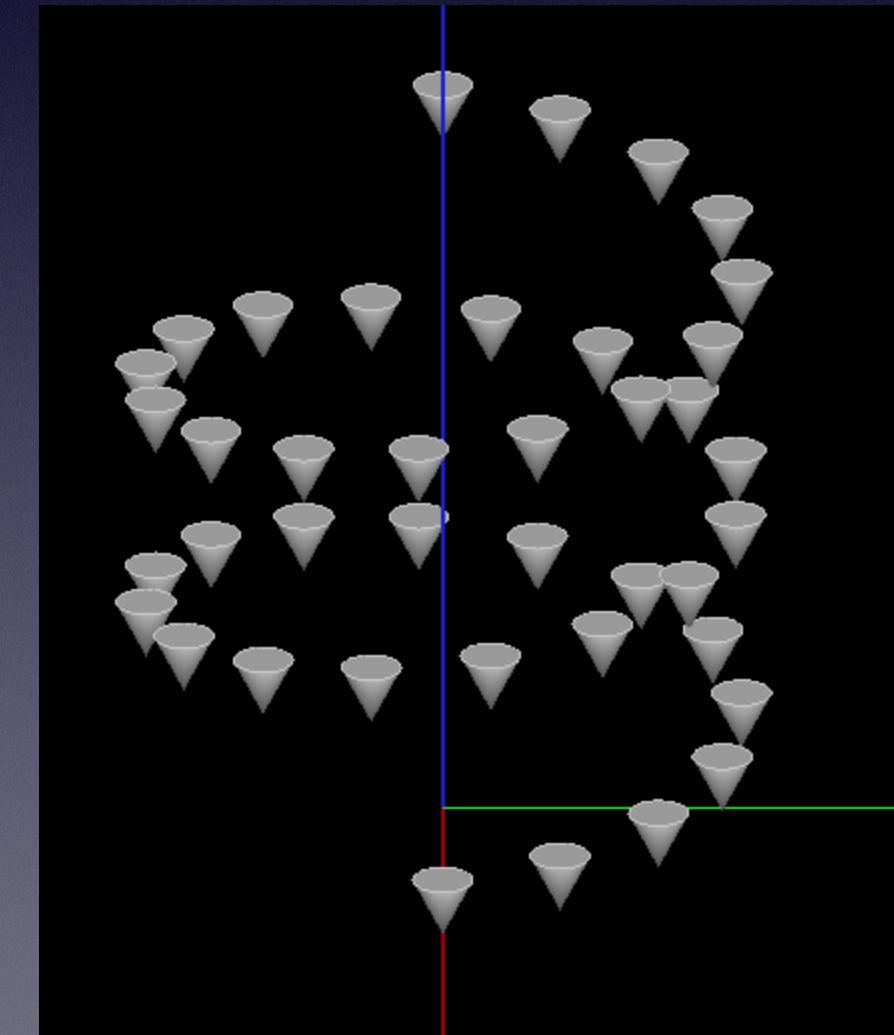
## 3. Arrays

### c. Arrays of objects distributed along a path

Path arrays are added by selecting the object, a curve (path) in 3D and executing the path array command



View of exported gdml file containing a path array, using load\_gdml\_color

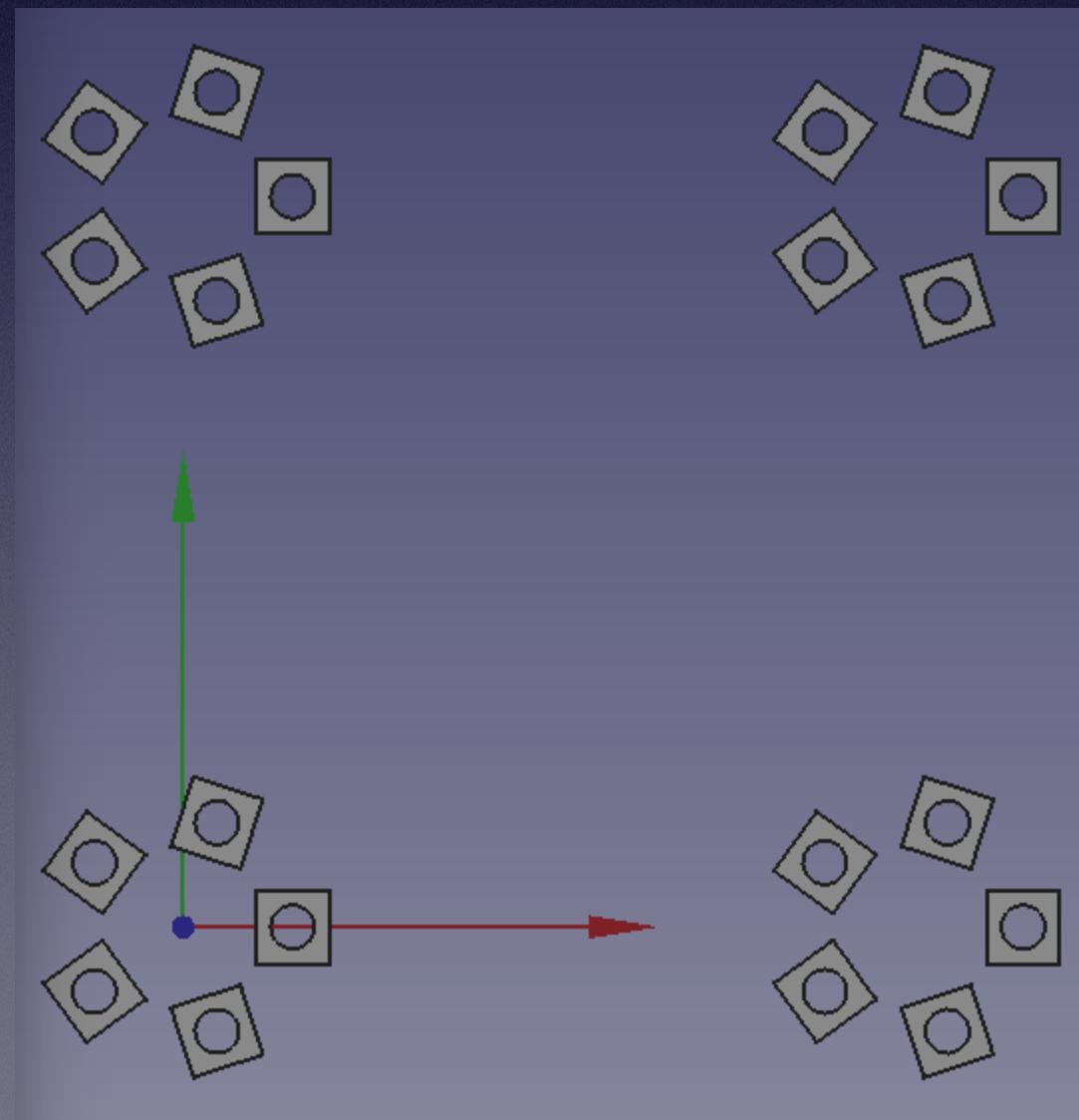


# Adding non-GDML solids to the GDML document

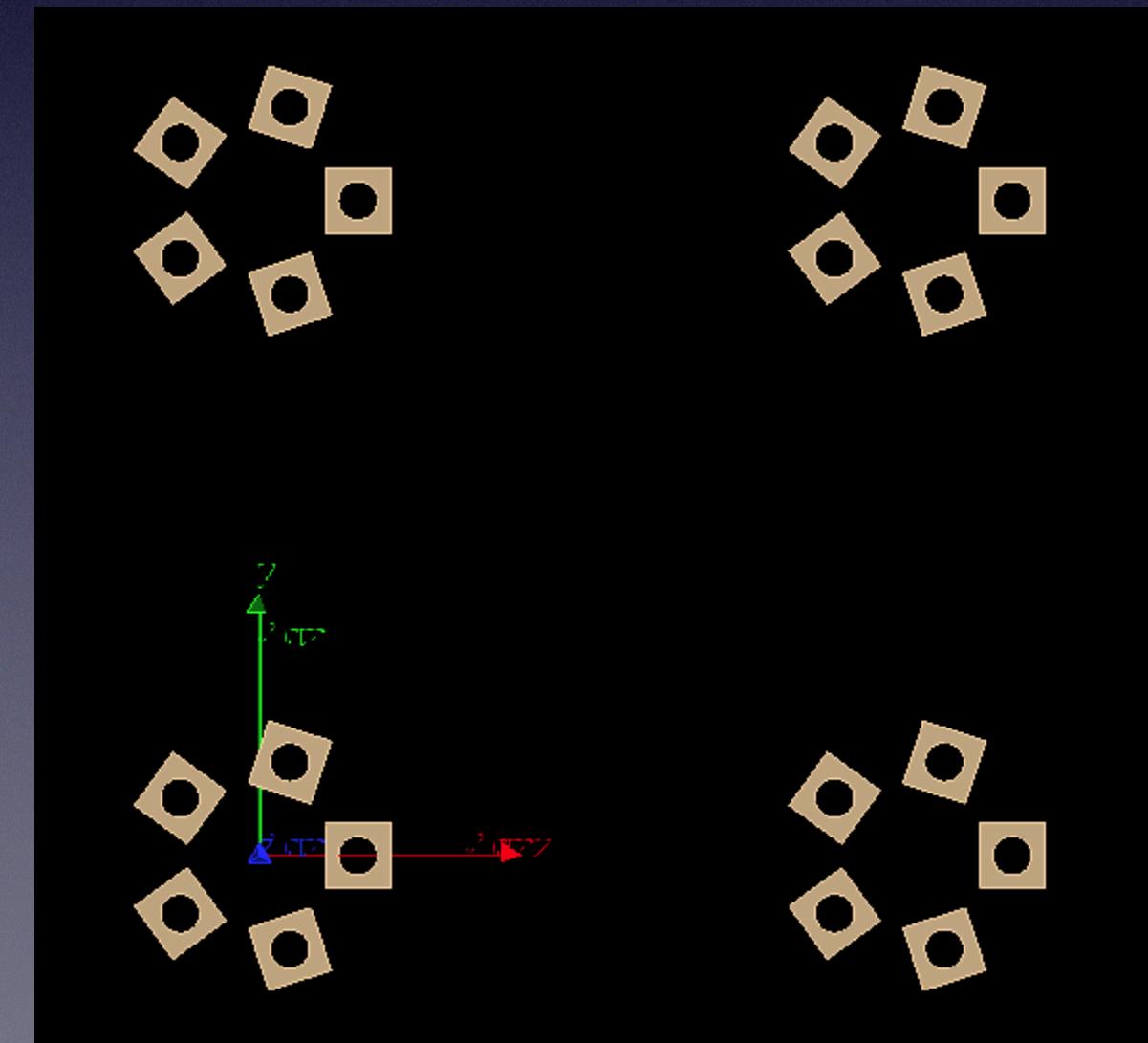
## 3. Arrays

### e. Arrays of arrays

Any object that can be exported to GDML by the Workbench can serve as the base of an array. Below shows an orthogonal array of a polar array of a Cut of two solids in FreeCAD.



View of the exported gdml file containing an orthogonal array of a polar array, using `load_gdml_color`

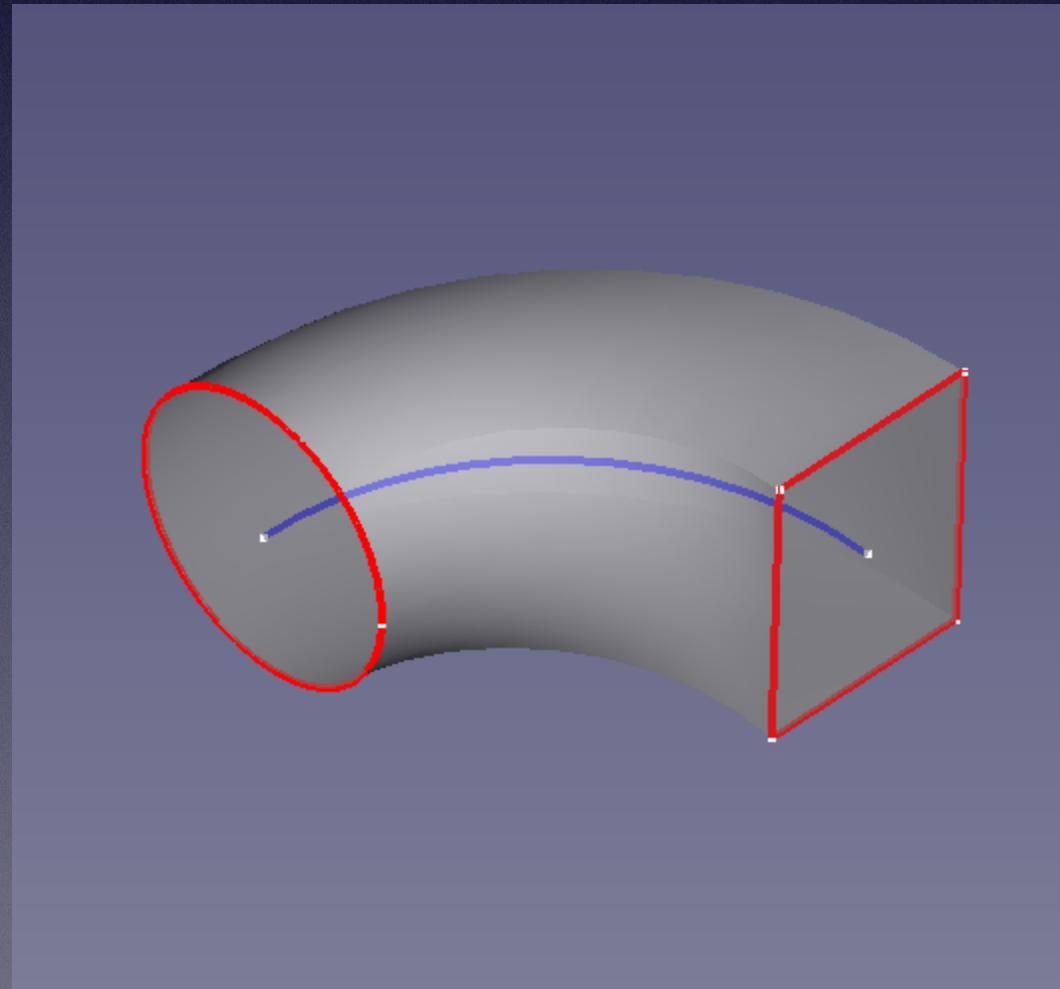


# Adding non-GDML solids to the GDML document

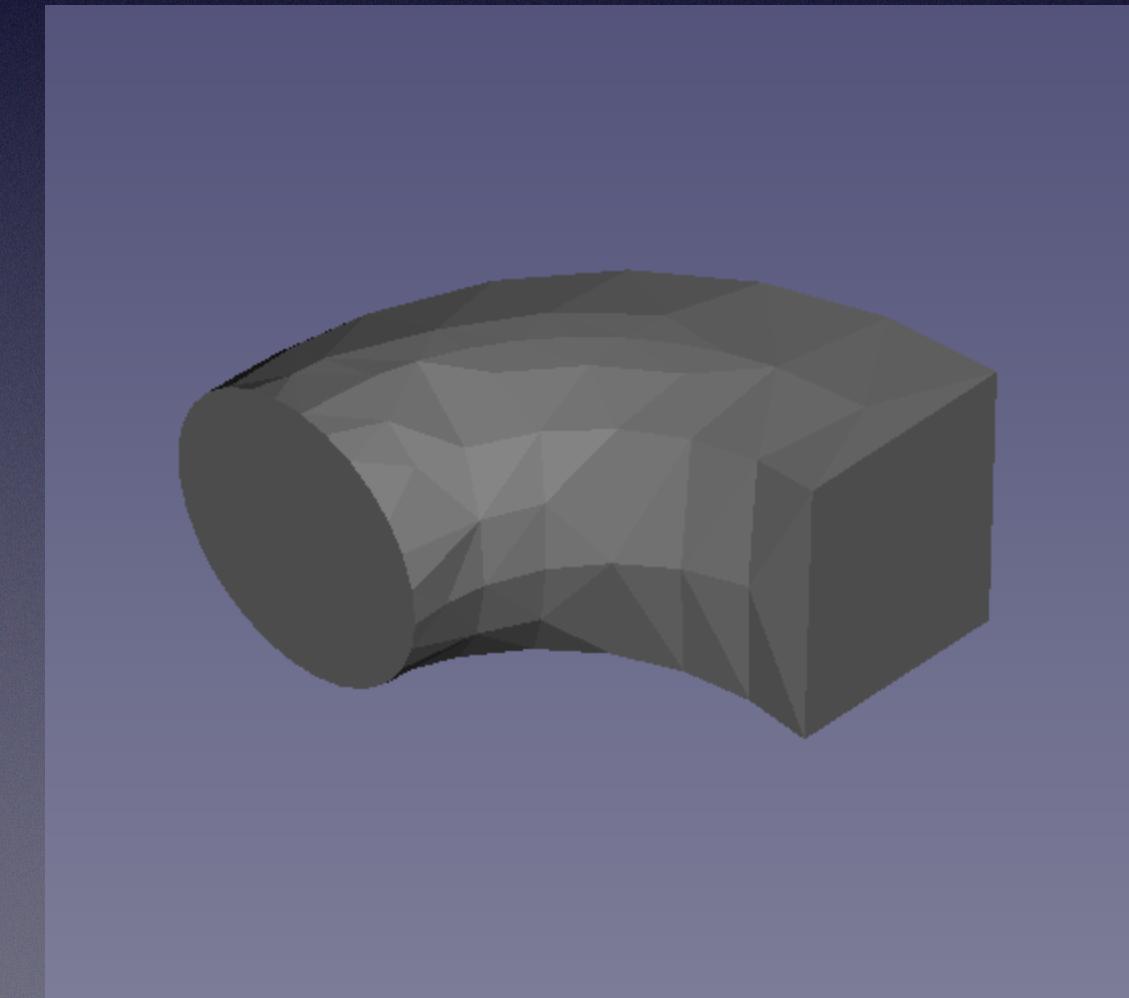
Any solid can be exported to a gdml file by meshing it then tessellating the mesh

Example:

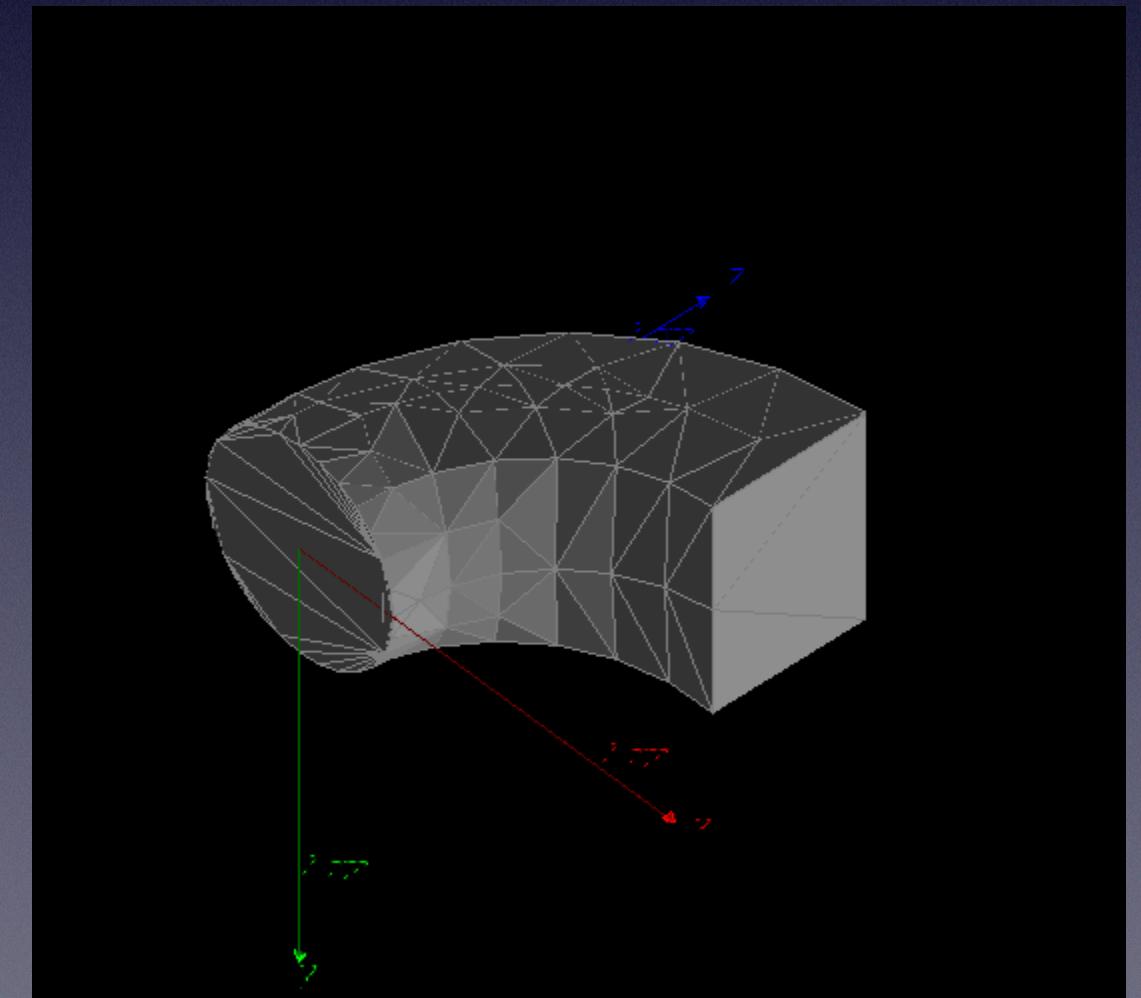
Part Workbench can create a solid by sweeping one sketch to another along a path. GDML Workbench currently cannot export a sweep.



Can convert the sweep shape to a tessellated solid using the Part to Mesh command



View of exported gdml file containing a tessellation, using load\_gdml\_color



# Opticals Support

- With GDML workbench activated, open new file
  - Creates GDML file structure
    - Matrix
    - Surfaces
- Import matrix.xml file see example  
freecad/gdml/Resources/matrix.xml
  - Matrix is a number of FreeCAD spreadsheets within the Opticals/Matrix Group
- Import optical.xml file see example  
freecad/gdml/Resources/optical.xml
  - Definition of a number of surfaces
- Icons/Commands
  - Set Skin Surface 
  - Set Border Surface 

Labels & Attributes

	Labels	Attributes	Description
1	Glass	7.74901*eV	1.31
2	LYSO	7.75415*eV	1.31
3	G4_CESIUM_IODIDE : 1.000	7.75929*eV	1.31
4	Geant4	7.76444*eV	1.31
5	Opticals	7.7696*eV	1.31
6	Matrix	7.77477*eV	1.31
7	SY	7.77994*eV	1.31
8	RS	7.78512*eV	1.31
9	FTC	7.7903*eV	1.31
10	STC	7.79549*eV	1.32
11	RY1	7.80069*eV	1.32
12	RY2	7.8059*eV	1.32
13	YR	7.81111*eV	1.32
14	SCINTLAR	7.81633*eV	1.32
15	RINDEXLAR	7.82156*eV	1.32
16	ABSLNGTHLAR	7.82679*eV	1.32
17	RAYLEIGHLAR	7.83203*eV	1.32
18	REFLECTIVITY	7.83728*eV	1.32
19	EFFICIENCY	7.84253*eV	1.32
20	AirRINDEX	7.8478*eV	1.32
21	Surfaces	7.85307*eV	1.32
22	worldVOL	7.85834*eV	1.32
23	Origin	7.86362*eV	1.32
24	GDMLBox_WorldBox	7.86891*eV	1.32
25		7.87421*eV	1.32
26		7.87952*eV	1.32
27		7.88483*eV	1.32
28		7.89015*eV	1.32
29		7.89547*eV	1.32
30		7.90081*eV	1.32
31		7.90615*eV	1.32
32		7.91149*eV	1.32
33		7.91685*eV	1.32
34		7.92221*eV	1.32
35		7.92758*eV	1.32
36		7.93296*eV	1.32

Properties

density	1.00
expand	3.00
name	LYSO
specific	4.00

Properties

ABSLNGTH	ABSLNGTHLAR
FASTCOMPONENT	SCINTLAR
FASTTIMECONSTANT	FTC
RAYLEIGH	RAYLEIGHLAR
REEMISSIONPROB	SCINTLAR
RESOLUTIONSCALE	RS
RINDEX	RINDEXLAR
SCINTILLATIONYIELD	SY
SLOWCOMPONENT	SCINTLAR
SLOWTIMECONSTANT	STC
YIELDRATIO	YR

View Data

## Other Information

- Information on the GDML workbench can be found on Keith's github page:  
<https://github.com/KeithSloan/GDML>
- Installation - see README in github
  - Gmsh shared library needs to be installed in FreeCAD Resources/lib see README
- Use instructions and example files are on the wiki page  
<https://github.com/KeithSloan/GDML/wiki>

# Future

- Command line scripts
- User defined materials in FreeCAD

User-defined materials can be imported into the Workbench, but not defined in it
- GUI creation of more solids

Although all solids defined under GDML are implemented, the lesser common ones do not have corresponding commands/icons to create them in the GDML Workbench. Currently can import all solids, so one without an icon can be imported into a document from a gdml file
- Check for overlapping solids in FreeCAD itself
- Closer match between exported and imported GDML files
- Need Help! Both Keith and I are in our early 70's and probably cannot continue to maintain the project far into the future. Would like to get younger people who are competent in python involved in the project with a view to eventually taking over the maintenance. We would be willing to mentor anyone who is interested!

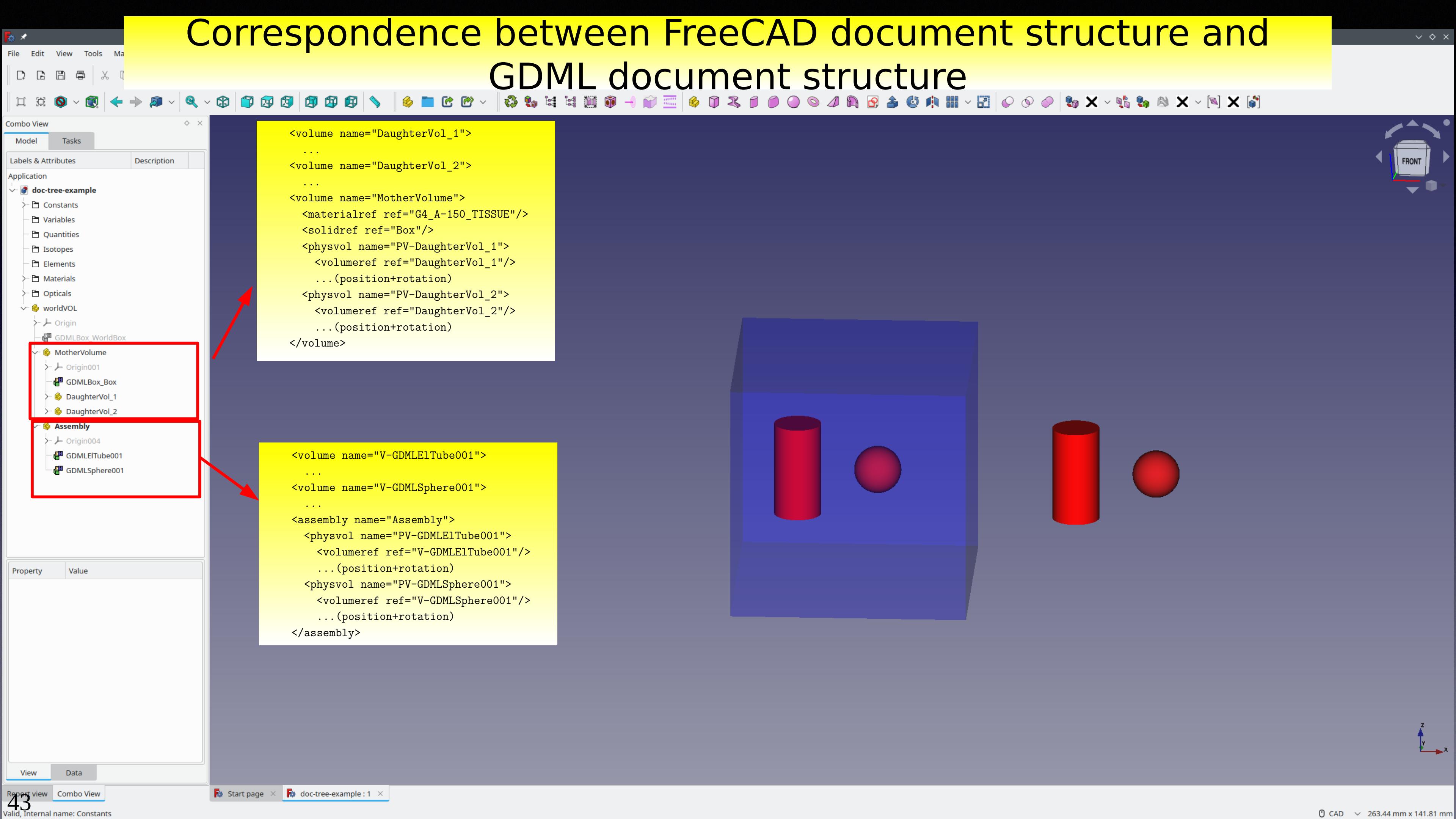


Thanks

- Jim Austin (icon designs)
- Emmanuel Delage, Wouter Deconnick, Louis Helary
- Damian Lambert, Paulo Dondero, Ami Hashemi
- John Watts, Frederico-Carminati
- Hilden Timo, Atanu Quant, Masaki Morita
- Large number from the FreeCAD community  
Please see github README
- See also GitHub Contributors

# Extra Slides

# Correspondence between FreeCAD document structure and GDML document structure



# Adding non-GDML solids to the GDML document

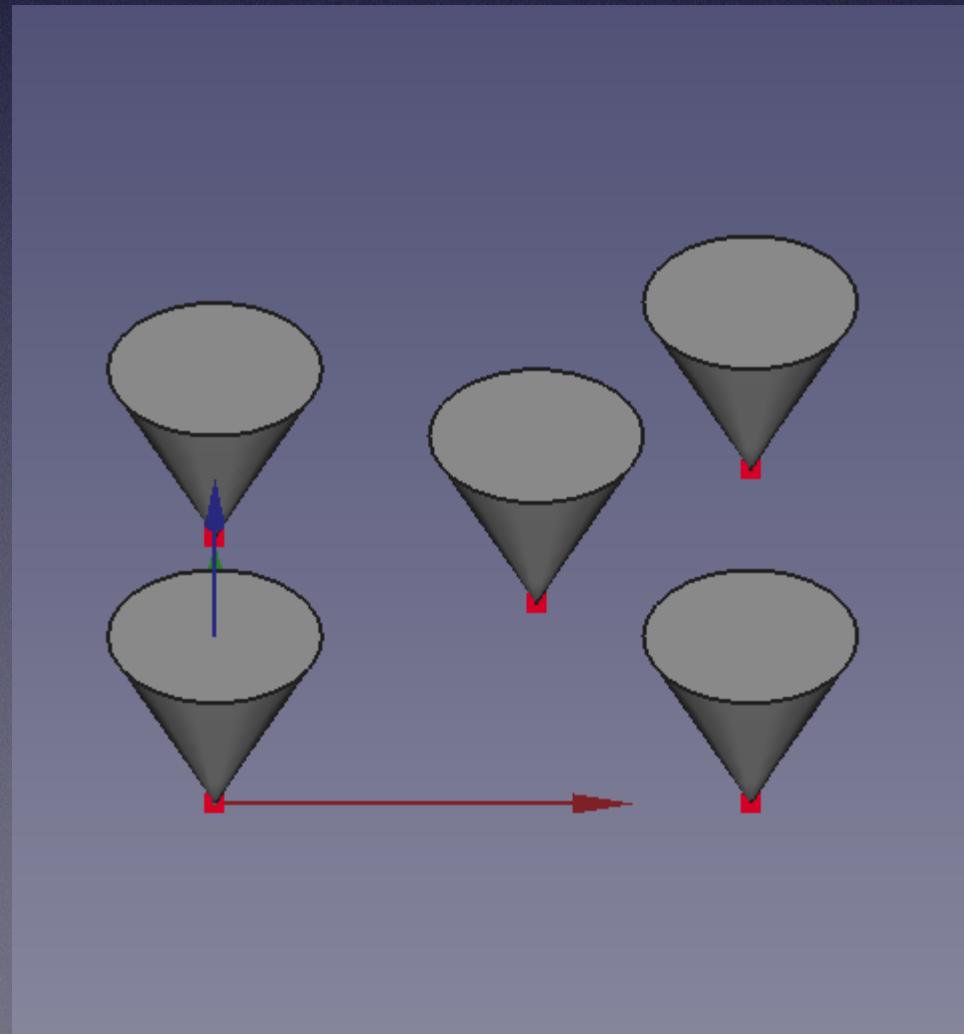
## 3. Arrays

### d. Arrays of objects placed at arbitrary points

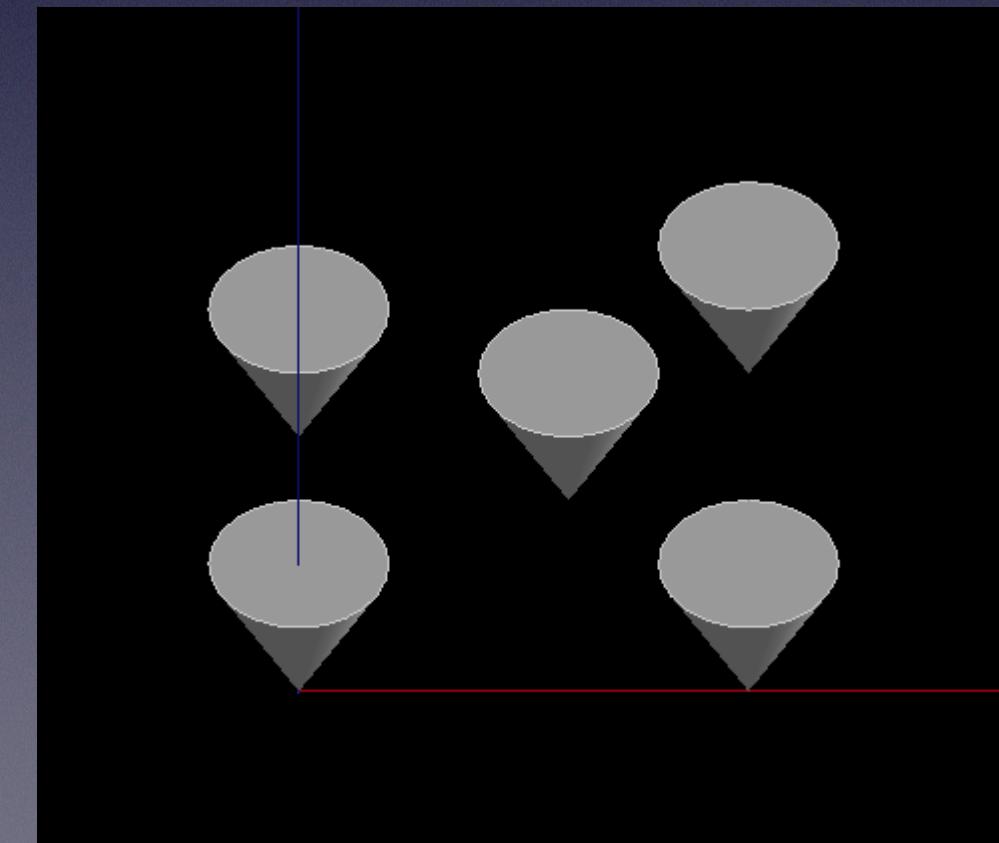
Point arrays are added by selecting the object, a set of points in 3D and executing the point array command.



Points can be entered in a variety of ways, including using the Points Workbench to read a variety of file formats.



View of exported gdml file containing a point array, using `load_gdml_color`



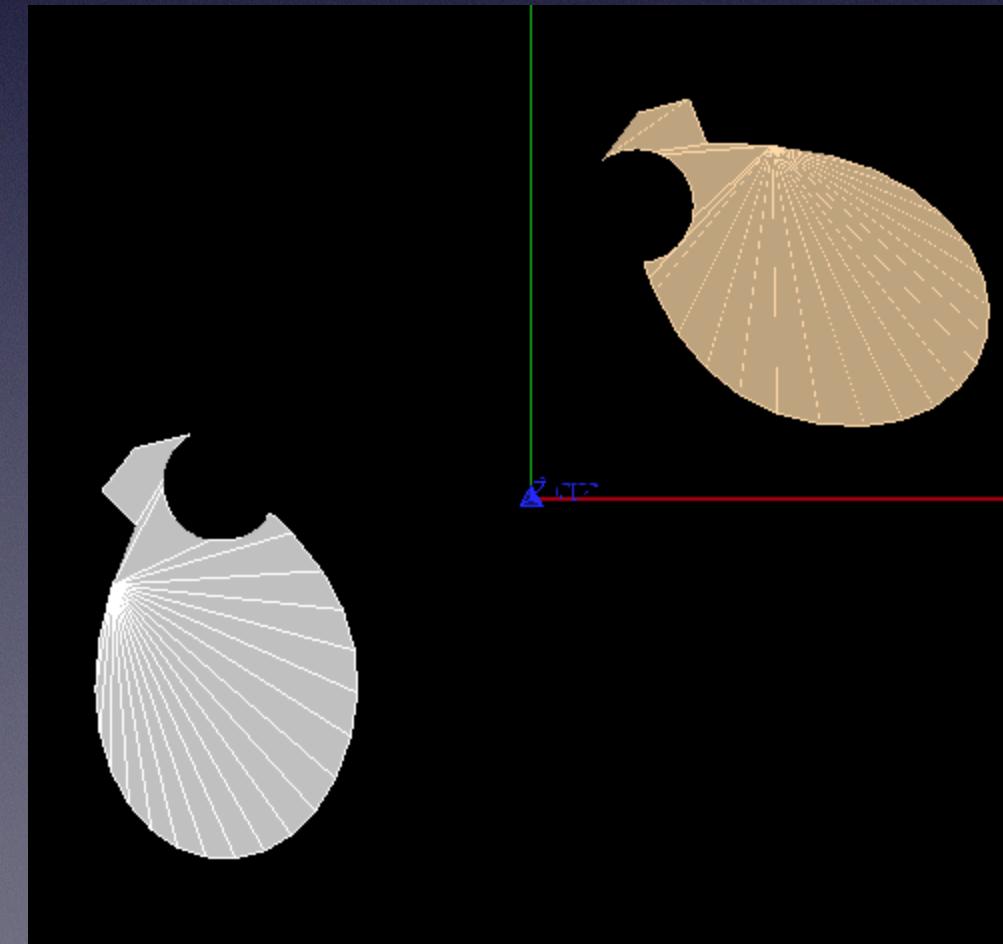
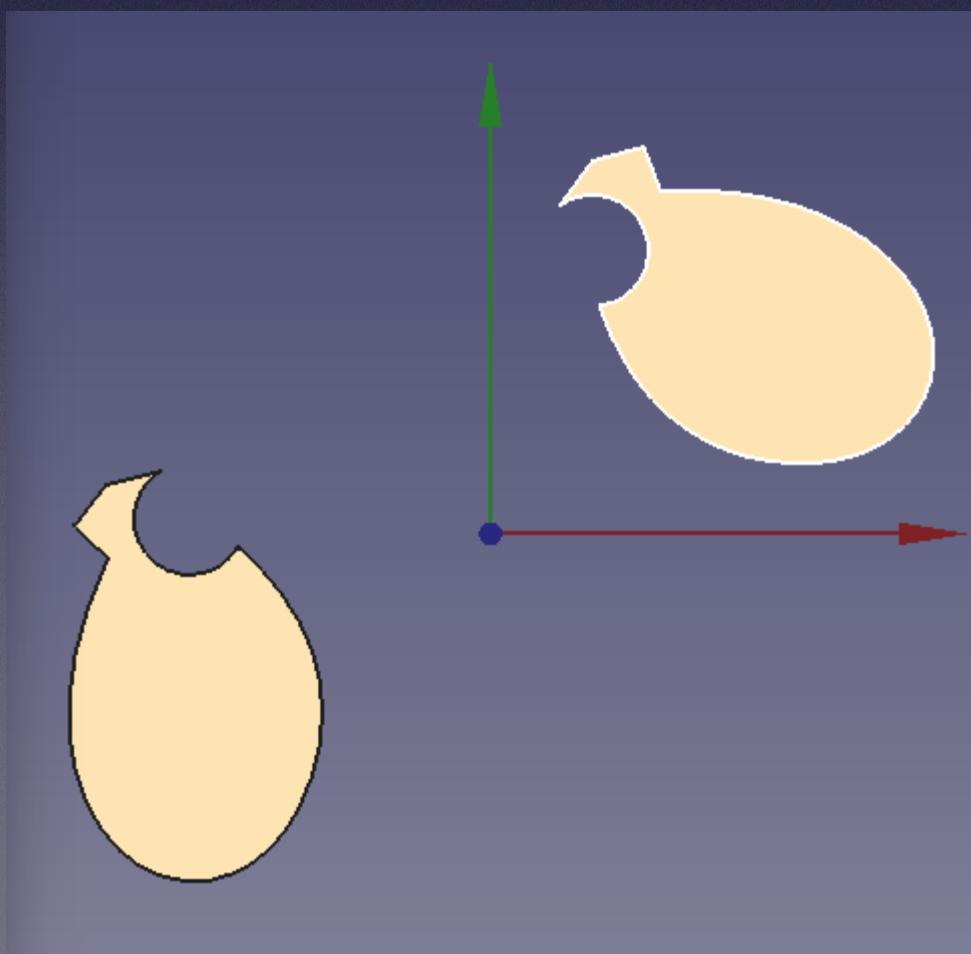
# Adding non-GDML solids to the GDML document

## 4. Mirrored object

Mirrored objects are added by selecting the object, a reflection plane and executing the mirror command.



View of exported gdml file containing a path array, using load\_gdml\_color



# GDML Workbench History

- CAD-GDML - Meeting at SLAC Sept 2011
  - C++ Workbench started by Emmanuel Delage
- Initial Workbench FreeCAD\_Python\_GDML
  - Geant4 python library which at the time was still under development
  - Now GitHub retired / archived
- GDML workbench - Created and maintained by **Keith Sloan**  
<https://github.com/KeithSloan/GDML>
  - Different approach
  - GDML Solids implemented as FreeCAD FeaturePython Objects
  - Recent enhancements - Thanks to Munther Hindi (joined the project November 2021)
    - Arrays, Mirrors
    - Extruded and Revolved Sketches