

Generation of SDFGeometry online



<https://hofk.de/SDFGeometry/>

Define the SDF in JSON format and drag the file into the marked area. Download the three.js BufferGeometry definition or GLTF. Examples below and <https://hofk.de/SDFGeometry/examples.zip>

Parameter	Defaults	
"Settings":		
"tsl":	0.05	triangle side length
"maxDistance":	1000.0	up to this distance is calculated
"startPoint":	{"x":0,"y":0,"z":-1000}	starting point outside the SDF
"direction":	{"x":0,"y":0,"z":1}	direction from starting point to hit SDF
"self-selected ID"	-	Use unique, meaningful Id. "Settings" is reserved
"type"	-	see available types below
"c"	{"x":0,"y":0,"z":0}	center
"whd"	{"x":1,"y":1,"z":1}	width, height, depth
"h"	1	height
"d"	1	depth, distance, displace (default only for depth)
"r"	1	radius
"R"	2	great radius (torus, link, Radius revolute)
"rb"	1	radius bottom
"rt"	0.5	radius top
"rxyz"	{"x":0.6,"y":1,"z":0.6}	radius x, y, z
"from", "to"	-	capsule (capsule default is vertical with r and h)
"bev"	0.25	bevel
"fx", "fy", "fz"	-	frame x , y, z (boxframe)
"f"	-	frame (cross), frequency (functions)
"a"	1.0	factor for SDFs defined with boundary functions
"mode"	"sin"	modus (distort)
"dxyz"	-	differences {"x": , "y": , "z": } (distort)
"fxyz"	-	frequencies {"x": , "y": , "z": } (distort)
"n"	3	number, (prism edges 3, .. 8), power exponent
"nnn"	{"x":3,"y":3,"z":3}	numbers (repeat)
"o"	0	angle in ° (DEG)
"rotx"	0	rotation x in °
"roty"	0	rotation y in °
"rotz"	0	rotation z in °
"order"	"XYZ"	order of Euler rotations
"quat"	-	rotation by quaternion, if used, Euler rotations are ignored
"bend"	0	bend factor in ° (DEG) x direction
"points"	-	array of points [{"h": , "r": } , ...] (rotate lines)
"planes"	-	array of section planes [{"normal": , "d": } , ...]
"normal"	-	plane normal {"x": , "y": , "z": }
"func"	-	displace and rotate functions (see available functions)
"shape"	-	2D shapes for types "extrude", "revolute"
"op"	-	operand
"op1"	-	operand 1
"op2"	-	operand 2
"b"	0.5	blending factor (0..1), factor for "cosparabol"
"s"	0.08	smooth (edge and operations)
"hollow"	true	for SDFs defined with boundary functions
"execute"	false	true to use this partial SDF, normally only the last part-SDF is executed

Parameters that do not have a default must be specified.

The parameters "c", "rotx", "roty", "rotz", "order", "quat", "bend", "s" are available for all SDF primitives.



Types and Parameters

Primitives		an example
		page
"type":	Parameter (for all: "c", "rotx", "roty", "rotz", "order", "quat", "bend", "s")	
"sphere"	"r"	5
"hemisphere"	"r" ($y \geq 0$)	9
"cutphere"	"r", "h" (factor h for $r: h > 0.11$)	9
"hollowsphere"	"r", "R", "h"	13
"concavesphere"	"r", "d" ($d < r$)	9
"cylinder"	"rb", "rt", "h"	5
"torus"	"r", "R"	5
"link"	"r", "R", "d" (also $d < 0$)	10
"cappedtorus"	"r", "R", "o"	9
"solidangle"	"r", "o"	10
"ellipsoid"	"rxyz"	5
"octahedron"	"r"	9
"pyramid"	"r" ($y \geq 0$)	9
"tetrahedron"	"r"	9
"capsule"	"r", "h" (total height h)	
"capsule"	"from", "to" (centers of the capsule's final hemispheres.)	9
"box"	"whd"	7
"bevelbox"	"whd", "bev"	6
"cutbox"	"whd", "d" (cut d)	10
"boxframe"	"whd", "fx", "fy", "fz"	8
"cross"	"r", "f", "d"	6
"prism"	"n", "r", "d"	5
"polyhedron"	"planes": [{"normal": {"x": , "y": , "z": }, "d": } , ...] (array of planes - normal, d must be specified) At least 4 planes must be specified to create a finite body.	12

Primitives generated from 2D SDFs.

"extrude"		6
"revolute"		8
with 2D SDF parameter	Parameter	
"shape":		
"parallelogram"	"whd", "o"	6
"vesica"	"whd"	6
"unevencapsule"	"rb", "rt", "h" ($h > rb+rt$)	8
"cutdisk"	"r", "h", "d" ($r < h < 2r$)	8



Primitives defined by boundary functions.

It only works with "balanced" parameter constellations.

Tip: Starting with a working example, make small, targeted changes to individual parameters.

Not all other parameters always work correctly in combination, especially "hollow" with the parameters for center, rotation, and bending.

an example

page

18

19

18

19

21

"paraboloid"	"a", "h", "hollow"
"cubic"	"a", "h", "hollow"
"powerparaboloid"	"a", "n", "h", "hollow"
"cosparabol"	"a", "n", "b", "f", "h", "hollow"
"hyperboloid"	"a", "r", "h", "hollow" ($r > -1$)

"rotate"

with function parameter Parameter (The function extends a base radius r)

"func":

"bevel"	"r", "h", "bev", "hollow"	19
"bevelcos"	"r", "h", "bev", "a", "f", "hollow"	19
"sin"	"r", "h", "a", "f"	19
"cos"	"r", "h", "a", "f",	19
"pow"	"r", "h", "a", "n", "hollow"	20
"root"	"r", "h", "a", "n", "hollow"	20
"exp"	"r", "h", "a", "hollow"	20
"log"	"r", "h", "a", "hollow"	20
"hyperbola"	"r", "h", "a", "hollow"	21
"lines"	"r", "h", "hollow", "points": [{"h": , "r": }, ...]	21
"curve"	"r", "h", "hollow", "points": [{"h": , "r": }, ...]	21
"linescos"	"r", "h", "a", "f", "hollow", "points": [{"h": , "r": }, ...]	20

The first point is usually close {"h": 0, "r": ..} the last (line) is extended to h, or h cuts off the function.



Operations

	Parameter	an example	page
"type":			
"union",	"op1", "op2", "s"		5
"union",	"op", "s" (op is array [...] of operands)		22
"subtraction"	"op1", "op2", "s" (op1 - op2)		7
"subtraction"	"op", "s" (op is array [...] of operands, subtraction from first entry)		22
"intersection"	"op1", "op2", "s"		8
"blending"	"op1", "op2", "b", "s" (b= 0...1)		8
"elongateX"	"op", "d" (distance)		13
"elongateY"	"op", "d" (distance)		13
"elongateZ"	"op", "d" (distance)		13
"rotation"	"op", "rotx", "roty", "rotz", "order", "quat", "c" (c is rotation pivot)		23
"translation"	"op", "c" (c is new center)		23
"mirrorXY"	"op"		
"mirrorXZ"	"op"		
"mirrorYZ"	"op"		16
"mirrorPlane"	"op", "normal", "d" (distance)		16
"scale"	"op", "whd" (scale factors)		13
"repeat"	"op", "d" (distance), "nnn" (odd number per axis, even numbers are rounded up)		14
"section"	"op", "planes", "s" "planes": [{"normal": , "d": }, ...] (array of section planes - normal:{"x": , "y": , "z": }, d must be specified)		11
"displace"	"op", "func", "d", "f" function "func": "sinX", "sinY", "sinZ", "sinXY", "sinXZ", "sinYZ", "sinXYZ", "FBM", "sinFBM", "sinRND"		15
"distort"	"op", "mode", "dxyz", "fxyz" (modus, differences, frequencies) modus "mode": "sin", "noise", "vnoise", "twist", "pinch"		17

See examples below with json and image. Last pages for possible errors.

Additional json files: <https://hofk.de/SDFGeometry/examples.zip>



Examples:

01_sphere.json

```
{
  "Sph": { "type": "sphere" }
}
```

A start point with direction must be specified here

02_spheres.json

because the default values result in a ray under the SDF.

```
{
  "Settings": { "tsl": 0.15, "maxDistance": 10,
    "startPoint": { "x": 0, "y": -10, "z": 0 }, "direction": { "x": 0, "y": 1, "z": 0 } },
  "SphA": { "type": "sphere", "c": { "x": 0, "y": 1.5, "z": 0 } },
  "SphB": { "type": "sphere", "r": 1.3, "c": { "x": 0, "y": 2.8, "z": 1 } },
  "Union1": { "type": "union", "op1": "SphA", "op2": "SphB", "s": 0.12 }
}
```

03_octPrism.json

for n, also take 3, 4, 5, 6, 8 =>

```
{
  "Settings": { "tsl": 0.1 },
  "octPrism": { "type": "prism", "n": 8, "r": 1, "s": 0.06 }
```

04_torusBended.json

```
{
  "Settings": { "tsl": 0.05 },
  "Torus": { "type": "torus", "r": 0.4, "bend": 28 }
```

05_ellipsoidBended.json

```
{
  "Settings": { "tsl": 0.1 },
  "Ellipsoid": { "type": "ellipsoid", "rxyz": { "x": 1.0, "y": 1.5, "z": 0.6 }, "bend": 50 }
```

xx_no picture Box - rotation using quaternion

// Use e.g. https://threejs.hofk.de/quaternion/quaternion_axisangle.html to achieve the desired rotation.

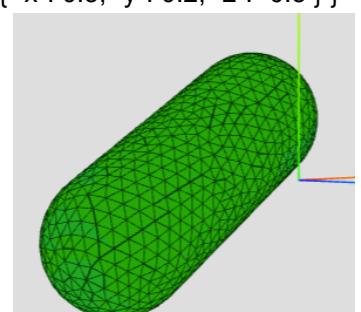
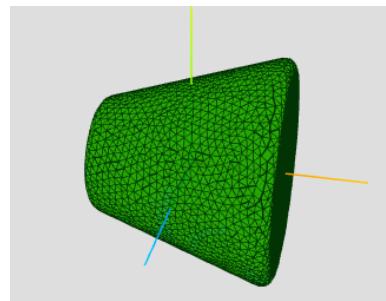
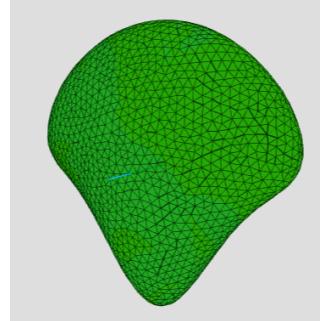
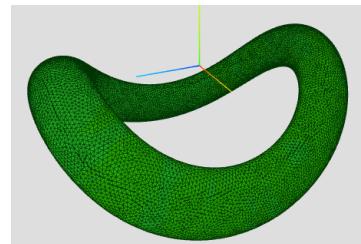
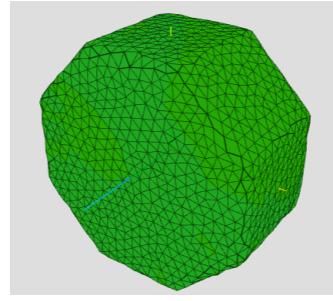
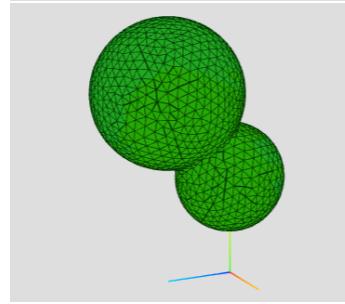
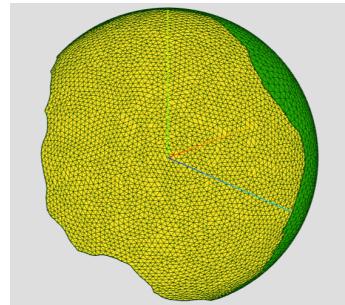
```
{
  "Settings": { "tsl": 0.11 },
  "BoxQuat": { "type": "box", "whd": { "x": 1.2, "y": 1.9, "z": 0.7 }, "quat": { "w": 0.914, "x": 0, "y": -0.266, "z": -0.307 }, "s": 0.08 }
```

06_capsuleFromTo.json

```
{
  "CapsuleFT": { "type": "capsule", "r": 0.4, "from": { "x": -0.3, "y": -0.2, "z": -0.5 }, "to": { "x": 0.3, "y": 0.2, "z": -0.5 } }
```

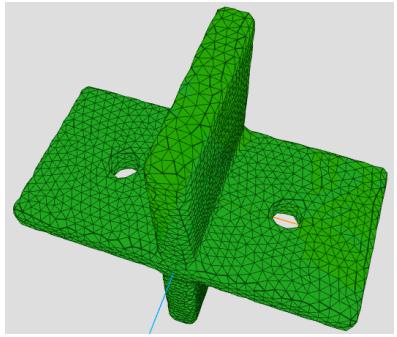
07_cylinder_cone.json

```
{
  "Cone": { "type": "cylinder",
    "rb": 0.4, "rt": 0.7, "h": 0.6, "rotz": 90, "s": 0.06 }
```



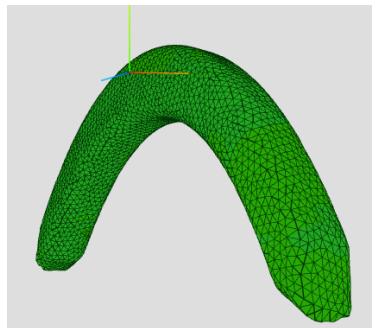
08_crossWithHoles.json (see also 13_cross.json)

```
{
  "Settings": { "tsl": 0.06 },
  "Box1": { "type": "box", "whd": { "x": 2, "y": 0.15, "z": 1 }, "s": 0.05 },
  "Box2": { "type": "box", "whd": { "x": 0.15, "y": 2, "z": 1 }, "s": 0.05 },
  "CapsuleA": { "type": "capsule", "r": 0.10,
    "from": { "x": 0.5, "y": -1, "z": 0 }, "to": { "x": 0.5, "y": 1, "z": 0 } },
  "CapsuleB": { "type": "capsule", "r": 0.10,
    "from": { "x": -0.5, "y": -1, "z": 0 }, "to": { "x": -0.5, "y": 1, "z": 0 } },
  "UnionBox": { "type": "union", "op1": "Box1", "op2": "Box2", "s": 0.05 },
  "UnionCaps": { "type": "union", "op1": "CapsuleA", "op2": "CapsuleB", "s": 0.0 },
  "Subtract1": { "type": "subtraction", "op1": "UnionBox", "op2": "UnionCaps", "s": 0.03 }
}
```



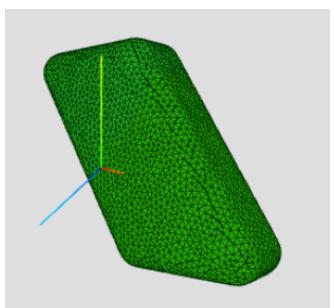
09_bevelboxBended.json "bev": default value

```
{
  "Settings": { "tsl": 0.08 },
  "bevelBox": { "type": "bevelbox", "whd": { "x": 3.4, "y": 0.6, "z": 0.6 },
    "c": { "x": 0.2, "y": -0.2, "z": -0.8 }, "rotY": 45, "rotX": 20, "bend": 18, "s": 0.2 }
}
```



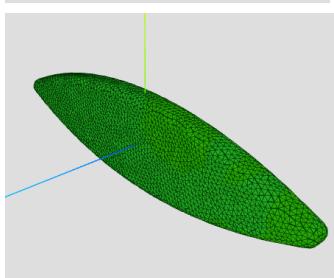
10_extrudeParallelogram.json extrusion of 2D SDF

```
{
  "Settings": { "tsl": 0.06 },
  "extrudeParall": { "type": "extrude", "shape": "parallelogram",
    "o": 60, "whd": { "x": 1.9, "y": 0.8, "z": 0.344 },
    "c": { "x": 0.3, "y": 0.04, "z": -0.4 },
    "rotZ": 45, "rotY": 25, "order": "ZXY", "s": 0.2 }
}
```



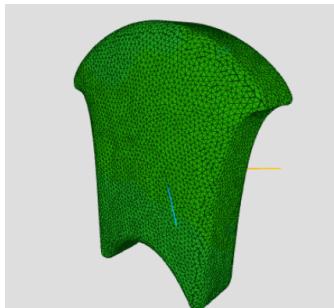
11_extrudeVesica.json extrusion of 2D SDF (x >= y if not, it is automatically swapped)

```
{
  "Settings": { "tsl": 0.03 },
  "extrudeVesica": { "type": "extrude", "shape": "vesica",
    "whd": { "x": 1.2, "y": 0.34, "z": 0.15 }, "s": 0.05 }
}
```



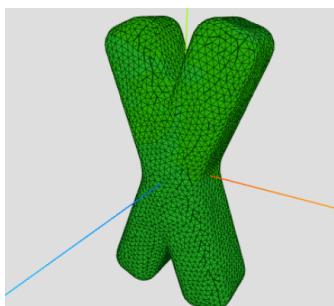
12_prismBended.json special effects through bending, "bend": < 90

```
{
  "Settings": { "tsl": 0.04 },
  "triPrism": { "type": "prism", "n": 3, "r": 0.9, "d": 0.33, "rotY": 45, "bend": 55, "s": 0.12 }
}
```



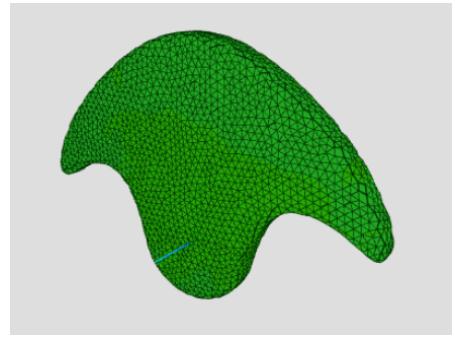
13_cross.json

```
{
  "Settings": { "tsl": 0.025 },
  "Cross": { "type": "cross", "r": 1, "f": 0.07, "d": 0.15, "o": 40, "s": 0.07 }
}
```



14_extrudeUnevenCapsuleBended.json

```
{  
  "Settings": { "tsl": 0.12 },  
  "extrUnevenCaps": { "type": "extrude", "shape": "unevencapsule",  
    "rb": 0.9, "rt": 0.4, "h": 3.9, "d": 0.22, "bend": 25 }  
}
```

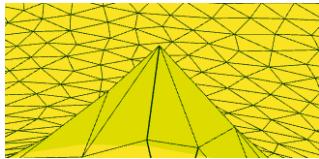


special effects through bending

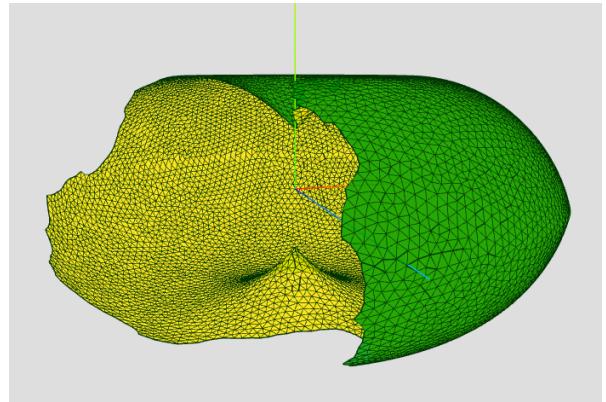
15_revoluteVesica.json Very sharp indentations can cause problems, algorithm can terminate.

```
{  
  "Settings": { "tsl": 0.03 },  
  "evoluteVesica": { "type": "revolute", "shape": "vesica", "whd": { "x": 0.4, "y": 0.5, "z": 0.3 }, "R": 0.4, "s": 0 }  
}
```

used: triangulation sequence >>>>



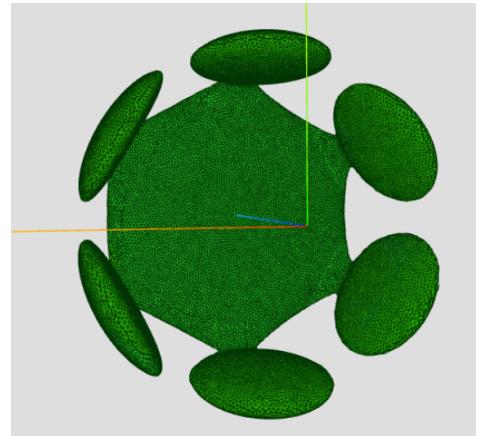
It's still working here



16_subtractionSpherePrism.json

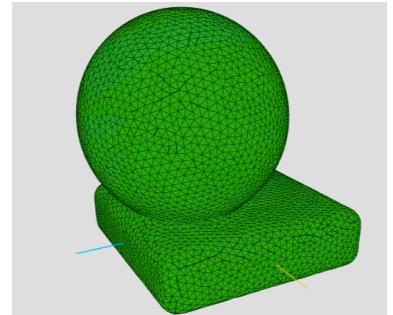
```
{  
  "Settings": { "tsl": 0.015 },  
  "Sph": { "type": "sphere", "r": 0.7, "c": { "x": 0, "y": 0, "z": 0.4 } },  
  "Prism": { "type": "prism", "n": 6, "r": 0.49, "d": 1.25 },  
  "Subtr": { "type": "subtraction", "op1": "Sph", "op2": "Prism", "s": 0.15 }  
}
```

With only slightly changed parameters,
it may not work because the narrow connections
cannot be triangulated.



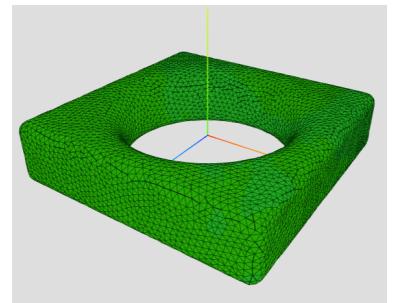
17_unionSphereBox.json

```
{  
  "Sph": { "type": "sphere", "r": 0.8, "c": { "x": 0.0, "y": 0.75, "z": 0.25 } },  
  "Box": { "type": "box", "whd": { "x": 1.4, "y": 0.4, "z": 1.4 }, "s": 0.11 },  
  "Union": { "type": "union", "op1": "Sph", "op2": "Box", "s": 0.08 }  
}
```



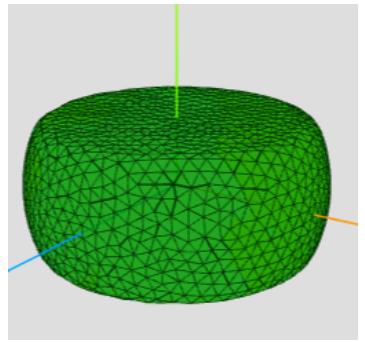
18_subtractionBoxSphere.json

```
{  
  "Box": { "type": "box", "whd": { "x": 2, "y": 0.5, "z": 2 } },  
  "Sph": { "type": "sphere", "r": 0.8, "c": { "x": 0, "y": 0.1, "z": 0 } },  
  "Sub": { "type": "subtraction", "op1": "Box", "op2": "Sph", "s": 0.3 }  
}
```



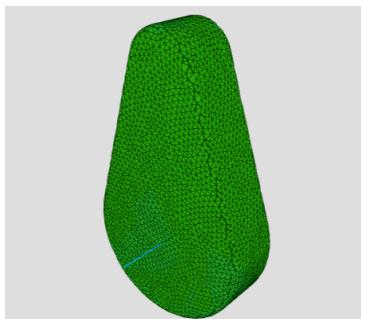
19_intersectionSphereBox.json

```
{  
  "Settings": { "tsl": 0.044 },  
  "Box": { "type": "box", "whd": { "x": 2.5, "y": 0.4, "z": 2.5 } },  
  "Sph": { "type": "sphere", "r": 0.6, "c": { "x": 0.0, "y": 0.0, "z": 0 } },  
  "IntersSphBox": { "type": "intersection", "op1": "Sph", "op2": "Box", "s": 0.04 }  
}
```



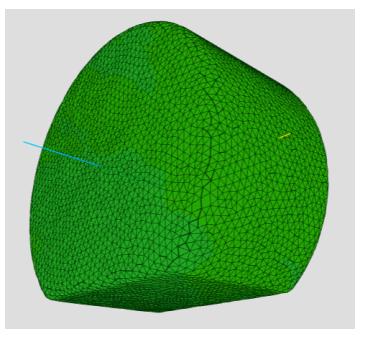
20_extrudeUnevenCapsule.json extrusion of 2D SDF

```
{  
  "Settings": { "tsl": 0.07 },  
  "extrudeUnevCaps": { "type": "extrude", "shape": "unevencapsule",  
                      "rb": 1.1, "rt": 0.5, "h": 3.5, "d": 0.75 }  
}
```



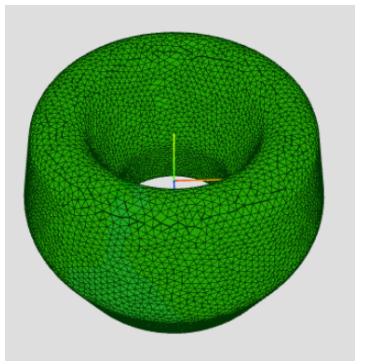
21_extrudeCutdisk.json extrusion of 2D SDF

```
{  
  "Settings": { "tsl": 0.05 },  
  "extrudeCutdisk": { "type": "extrude", "shape": "cutdisk",  
                     "r": 1, "h": 1.65, "d": 1, "s": 0.07 }  
}
```



22_revoluteUnevenCapsule.json revolution of 2D SDF

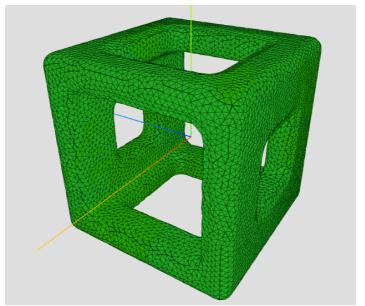
```
{  
  "Settings": { "tsl": 0.1 },  
  "revCaps": { "type": "revolute", "shape": "unevencapsule",  
              "R": 1.5, "rb": 0.6, "rt": 0.3, "h": 2.0, "s": 0.065 }  
}
```



23_boxframe.json

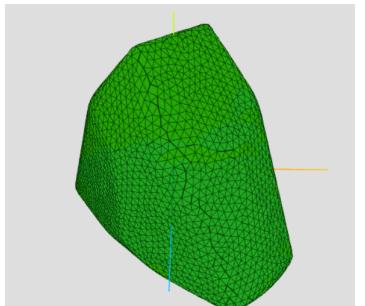
startPoint "y": 0.45, to hit the SDF

```
{  
  "Settings": { "tsl": 0.03, "startPoint": { "x": 0, "y": 0.45, "z": -100 } },  
  "BoxFrame": { "type": "boxframe", "fx": 0.15, "fy": 0.22, "fz": 0.32, "s": 0.06 }  
}
```



24_blendingPrismBox.json

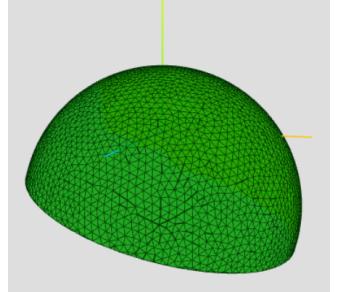
```
{  
  "Box": { "type": "box" },  
  "triPrism": { "type": "prism", "n": 3, "r": 0.9, "d": 0.66, "rotY": 45 },  
  "Blend": { "type": "blending", "op1": "triPrism", "op2": "Box", "b": 0.25, "s": 0.3 }  
}
```



25_hemisphere.json

```
{  
  "Hemisph": { "type": "hemisphere", "r": 1.05, "c": { "x": 0, "y": -0.5, "z": 0 }, "rotz": 15 }  
}
```

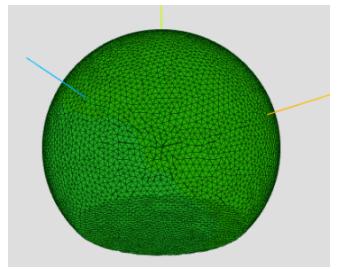
cut not rounded



26_cutsphere.json

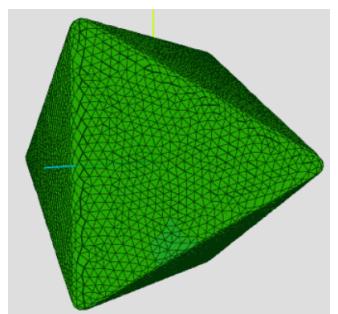
NOTE! Here h is factor for radius, h > 0.11, cut not rounded

```
{  
  "Settings": { "tsl": 0.03 },  
  "CutSph": { "type": "cutsphere", "r": 0.7, "c": { "x": 0, "y": 0.1, "z": 0 }, "h": 0.85 }  
}
```



27_octahedron.json

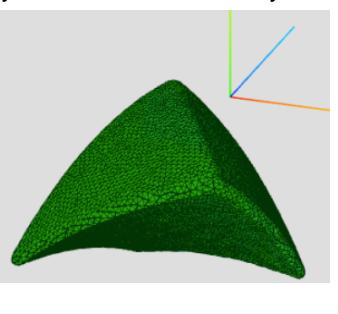
```
{  
  "Octahed": { "type": "octahedron", "r": 1.12, "rotx": 45, "s": 0.06 }  
}
```



28_pyramid.json

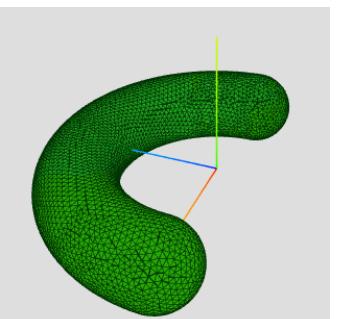
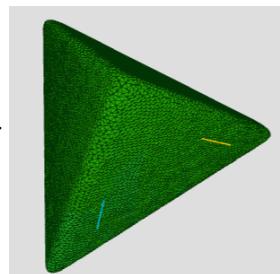
bended

```
{  
  "Pyramid": { "type": "pyramid", "r": 1.3, "c": { "x": 0, "y": -0.5, "z": -1.5 }, "bend": 15, "s": 0.06 }  
}
```



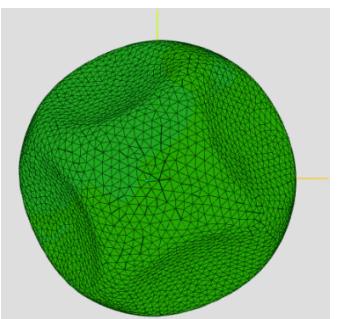
29_tetrahedron.json

```
{ "Settings": { "tsl": 0.05 } }  
"Tetra": { "type": "tetrahedron", "r": 2.0, "s": 0.16 }
```



30_cappedTorus.json

```
{  
  "Settings": { "tsl": 0.04 },  
  "CappedTorus": { "type": "cappedtorus", "r": 0.29, "R": 1.0, "o": 90 }  
}
```



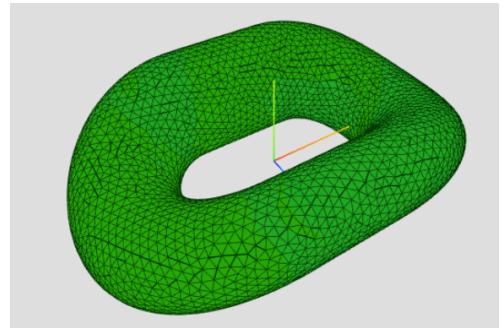
31_concaveSphere.json

```
{  
  "ConcaveSph": { "type": "concavesphere", "r": 0.8, "d": 0.8, "rotz": 20, "s": 0.07 }  
}
```



32_linkBended.json

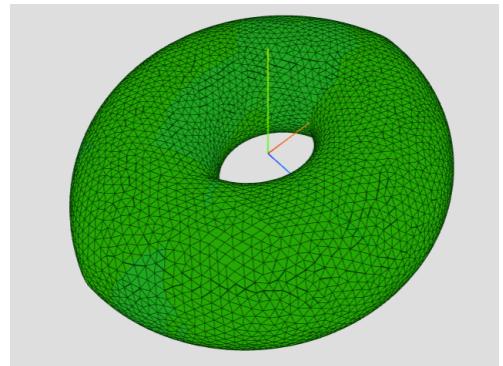
```
{  
  "Settings": { "tsl": 0.09 },  
  "Link": { "type": "link", "d": 2, "r": 0.5, "R": 1, "rotY": 90, "bend": -20 }  
}
```



33_linkNegative.json

```
{  
  "Settings": { "tsl": 0.06 },  
  "Link": { "type": "link", "d": -0.8, "r": 0.5, "R": 1 }  
}
```

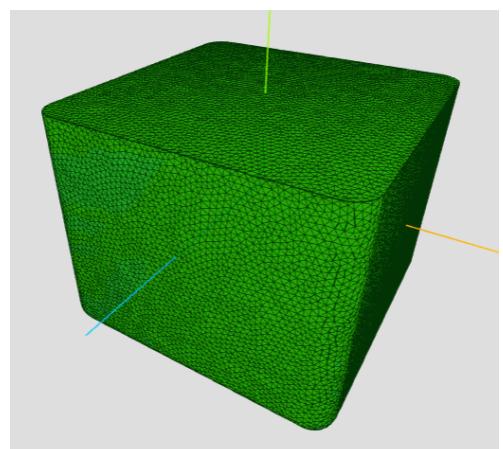
Note: d < 0



34_cutbox.json

```
{  
  "Settings": { "tsl": 0.025 },  
  "CutBox": { "type": "cutbox", "d": 0.2, "s": 0.09 }  
}
```

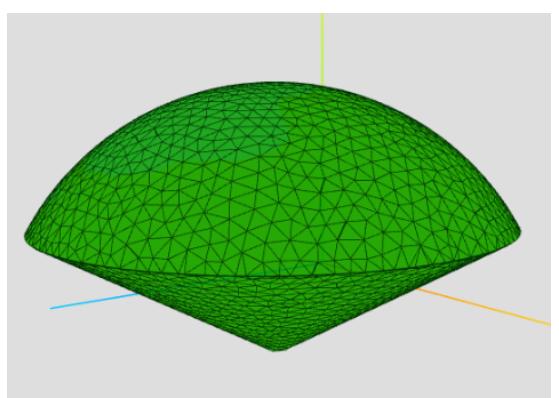
cut not rounded



35_solidangle.json

```
{  
  "Settings": { "tsl": 0.045 },  
  "solidangle": { "type": "solidangle",  
    "c": { "x": 0, "y": -0.3, "z": 0.2 }, "r": 1.1, "o": 60 }  
}
```

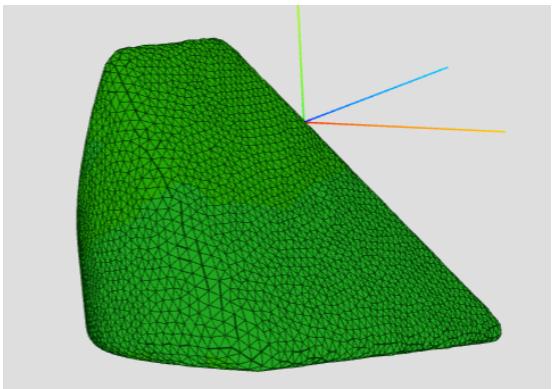
Cut not rounded



36_sectionCylinder.json

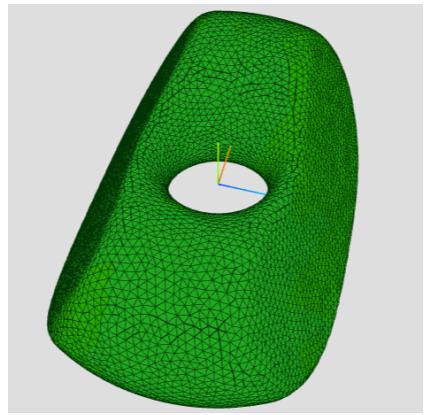
```
{  
  "Settings": { "tsl": 0.04, "startPoint": { "x": 0, "y": -0.25, "z": -1000 } },  
  "Cyl": { "type": "cylinder", "rt": 1.2, "s": 0.09 },  
  "Section": { "type": "section",  
    "op": "Cyl", "s": 0.09,  
    "planes": [ { "normal": { "x": 1, "y": 1, "z": 2 }, "d": 0.0 },  
      { "normal": { "x": 1, "y": 1, "z": -1 }, "d": 0.0 },  
      { "normal": { "x": -1, "y": 1, "z": -1 }, "d": 0.4 } ]  
  }
```

startPoint "y": -0.25, to hit the SDF



37_sectionTorus.json

```
{  
  "Settings": { "tsl": 0.12 },  
  "Torus": { "type": "torus"},  
  "Section": { "type": "section", "op": "Torus", "s": 0.24,  
    "planes": [ { "normal": { "x":0, "y":0.5, "z": 0.9}, "d": 1.3},  
      { "normal": { "x":0, "y":0.5, "z": -0.9}, "d": 1.3} ] }  
}
```

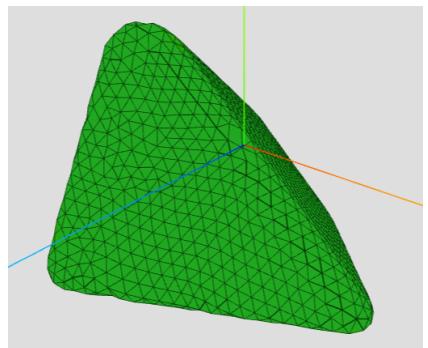
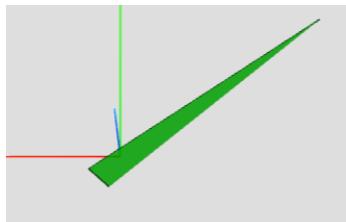


38_sectionBox.json

```
{  
  "Box": { "type": "box" },  
  "Sect": { "type": "section",  
    "op": "Box",  
    "planes": [ { "normal": { "x": 1, "y":1, "z": 2}, "d": 0.0 },  
      { "normal": { "x": 1, "y":1.1, "z": -1}, "d": 0.0 },  
      { "normal": { "x": -1, "y":1, "z": -1}, "d": 0.4 } ],  
    "s": 0.15  
  }  
}
```

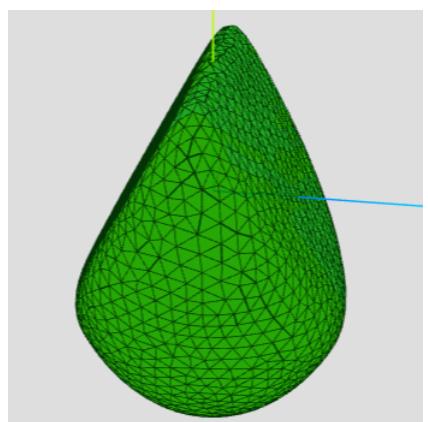
Risk, with the box default settings
the figure is hit very close.

triangulation sequence >>>



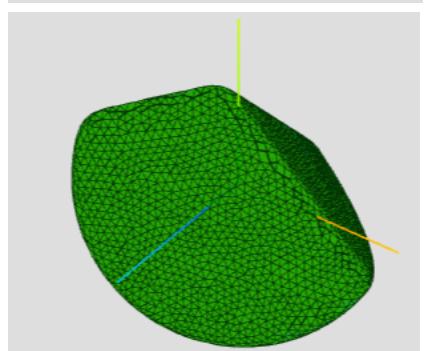
39_sectionEllipsoid.json

```
{  
  "Settings": { "tsl": 0.08 },  
  "Ellip": { "type": "ellipsoid" },  
  "Section": { "type": "section",  
    "op": "Ellip",  
    "planes": [  
      { "normal": { "x":0, "y":0.5, "z": 0.9}, "d": 0.3},  
      { "normal": { "x":0, "y":0.5, "z": -0.9}, "d": 0.3} ],  
    "s": 0.16  
  }  
}
```



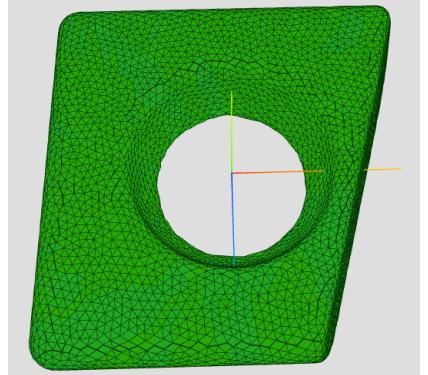
40_sectionSphere.json

```
{  
  "Sph": { "type": "sphere"},  
  "Section": { "type": "section",  
    "op": "Sph",  
    "planes": [ { "normal": { "x": 1, "y":1, "z": 2}, "d": 0.2 },  
      { "normal": { "x": 1, "y":1.1, "z": -1}, "d": 0.3 },  
      { "normal": { "x": -1, "y":1, "z": -1}, "d": 0.4 } ],  
    "s": 0.16 }  
}
```



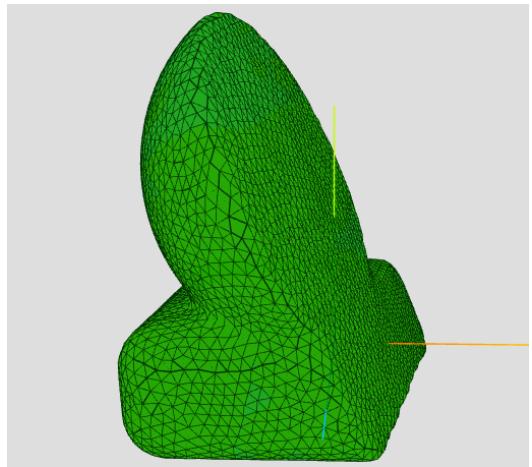
41_subtractionBoxSphereSection.json

```
{  
  "Box": { "type": "box", "whd": { "x": 2, "y": 0.3, "z": 2 } },  
  "Sph": { "type": "sphere", "r": 0.7, "c": { "x": 0, "y": 0.18, "z": 0 } },  
  "Sub": { "type": "subtraction", "op1": "Box", "op2": "Sph", "s": 0.1},  
  "Section": { "type": "section", "op": "Sub", "s": 0.06,  
    "planes": [ { "normal": { "x":1, "y": 0.3, "z": 0.3}, "d": 0.75} ] }  
}
```



42_unionBoxSphereSection.json

```
{  
  "Sph": { "type": "sphere", "r": 0.8, "c": { "x": 0.0, "y": 0.75, "z": 0.25 } },  
  "Box": { "type": "box", "whd": { "x": 1.4, "y": 0.4, "z": 1.4 }, "s": 0.11 },  
  "Union": { "type": "union", "op1": "Sph", "op2": "Box", "s": 0.11 },  
  "Section": { "type": "section", "op": "Union", "s": 0.06,  
    "planes": [ { "normal": { "x": 1.4, "y": 0.6, "z": 0.3}, "d": 0.2} ] }  
}
```



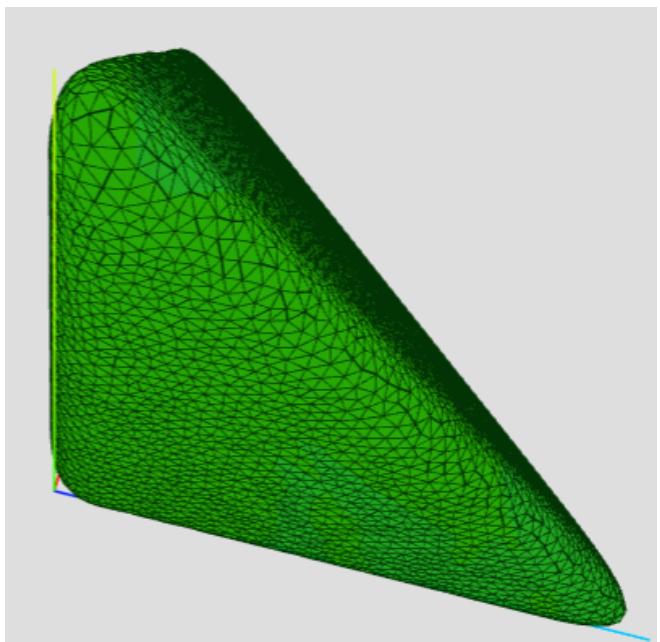
section or polyhedron $d = 1/(\sqrt{3}) = 0.57735\dots$

43_BoxSection.json

```
{ "Settings": { "tsl": 0.025 },  
  "Box": { "type": "box", "c": { "x": 0.5, "y": 0.5, "z": 0.5 } },  
  "Corner": { "type": "section", "op": "Box", "s": 0.12,  
    "planes": [  
      { "normal": { "x": -1, "y": 0, "z": 0 }, "d": 0.0 },  
      { "normal": { "x": 0, "y": -1, "z": 0 }, "d": 0.0 },  
      { "normal": { "x": 0, "y": 0, "z": -1 }, "d": 0.0 },  
      { "normal": { "x": 1, "y": 1, "z": 1 }, "d": 0.57735 }  
    ] }  
}
```

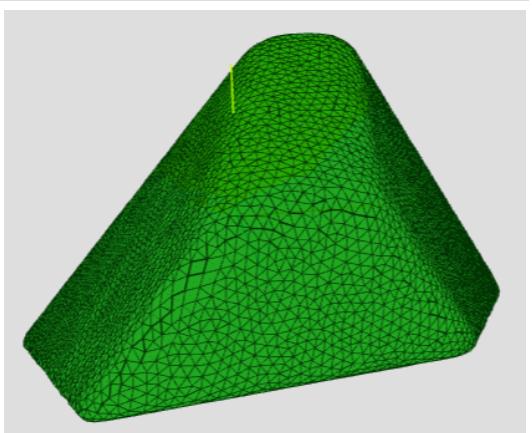
44_Polyhedron.json

```
{ "Settings": { "tsl": 0.025 },  
  "Polyh": { "type": "polyhedron", "s": 0.12,  
    "planes": [  
      { "normal": { "x": -1, "y": 0, "z": 0 }, "d": 0.0 },  
      { "normal": { "x": 0, "y": -1, "z": 0 }, "d": 0.0 },  
      { "normal": { "x": 0, "y": 0, "z": -1 }, "d": 0.0 },  
      { "normal": { "x": 1, "y": 1, "z": 1 }, "d": 0.57735 }  
    ] }  
}
```



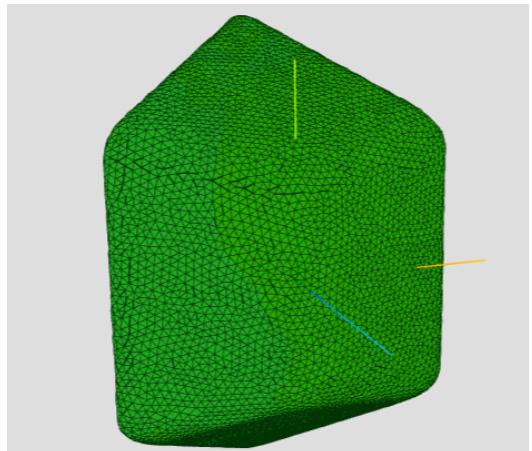
45_Polyhedron.json

```
{ "Settings": { "tsl": 0.08 },  
  "Polyhed": { "type": "polyhedron", "s": 0.18,  
    "planes": [  
      { "normal": { "x": -1, "y": 0.8, "z": 0 }, "d": 0.4 },  
      { "normal": { "x": 0, "y": -1, "z": 0.3 }, "d": 1 },  
      { "normal": { "x": 0, "y": 0.2, "z": -0.8 }, "d": 1 },  
      { "normal": { "x": 1, "y": 0.6, "z": 0.4 }, "d": 1 },  
      { "normal": { "x": 0, "y": 1, "z": 0.4 }, "d": 0.7 },  
      { "normal": { "x": 0.2, "y": 0, "z": 1 }, "d": 1 }  
    ] }  
}
```



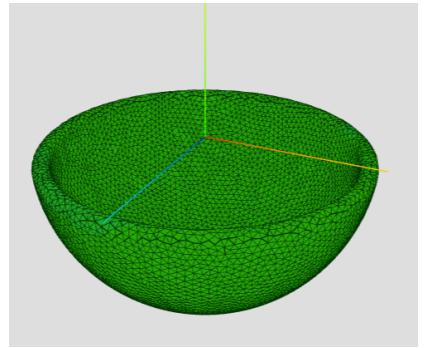
46_Polyhedron.json

```
{ "Settings": { "tsl": 0.04 },  
  "Polyh": { "type": "polyhedron", "s": 0.13,  
    "planes": [  
      { "normal": { "x": 0.23, "y": 0, "z": -0.23 }, "d": 0.64 },  
      { "normal": { "x": -0.23, "y": 0, "z": -0.23 }, "d": 0.64 },  
      { "normal": { "x": 0.07, "y": 0, "z": 0.20 }, "d": 0.17 },  
      { "normal": { "x": -0.07, "y": 0, "z": 0.20 }, "d": 0.17 },  
  
      { "normal": { "x": 0, "y": 1, "z": 0.7 }, "d": 0.5 },  
      { "normal": { "x": 0, "y": -1, "z": 0.7 }, "d": 0.5 }  
    ] }  
}
```



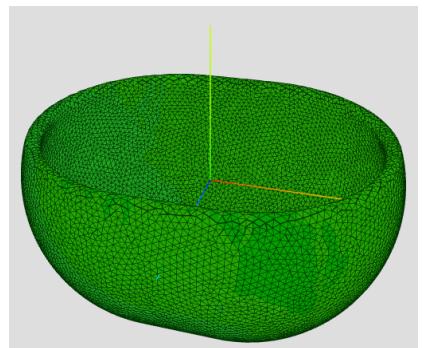
47_Hollowsphere.json

```
{  
  "Settings": { "tsl": 0.03, "startPoint": { "x": 0, "y": -0.8, "z": -100 } },  
  "HollowSph": { "type": "hollowsphere", "R": 1.0, "r": 0.9, "h": 1.0 }  
}
```



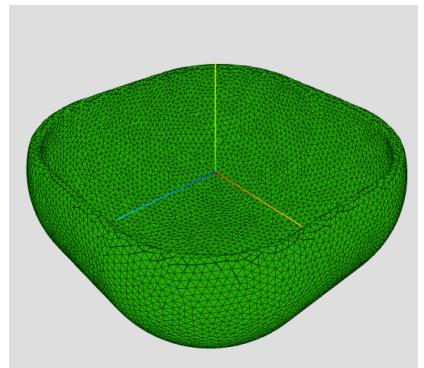
48_HollowsphereElongateX.json

```
{  
  "Settings": { "tsl": 0.035, "startPoint": { "x": 0, "y": -0.8, "z": -100 } },  
  "HollowSph": { "type": "hollowsphere", "R": 1.0, "r": 0.9, "h": 1.4 },  
  "Elong": { "type": "elongateX", "op": "HollowSph", "d": 0.4 }  
}
```



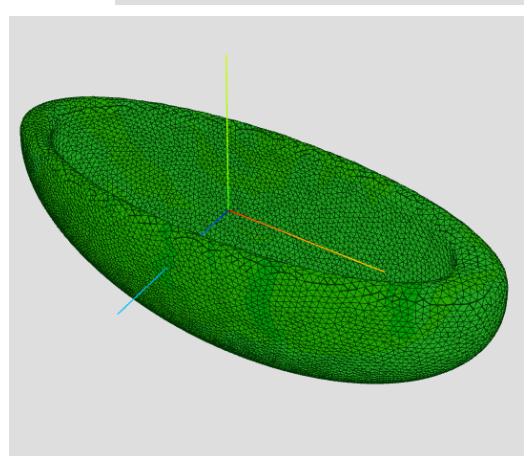
49_HollowsphereElongateXZ.json

```
{  
  "Settings": { "tsl": 0.039, "startPoint": { "x": 0, "y": -0.8, "z": -100 } },  
  "HollowSph": { "type": "hollowsphere", "R": 1.0, "r": 0.88, "h": 1.4 },  
  "ElongX": { "type": "elongateX", "op": "HollowSph", "d": 0.6 },  
  "ElongZ": { "type": "elongateZ", "op": "ElongX", "d": 0.6 }  
}
```



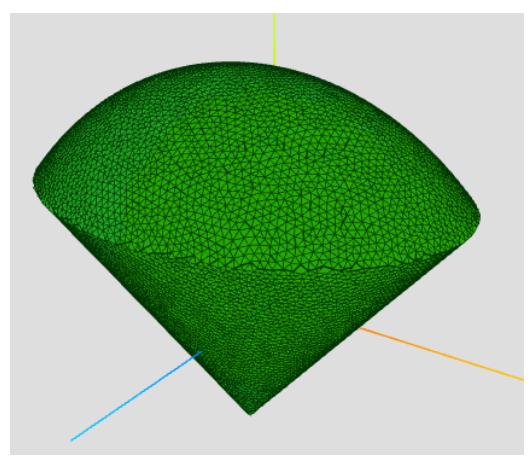
50_HollowsphereScale.json

```
{  
  "Settings": { "tsl": 0.04 },  
  "HSph": { "type": "hollowsphere", "R": 1, "r": 0.8, "h": 1.25 },  
  "Scale": { "type": "scale", "op": "HSph",  
            "whd": { "x": 1.9, "y": 1.2, "z": 0.8 } }  
}
```



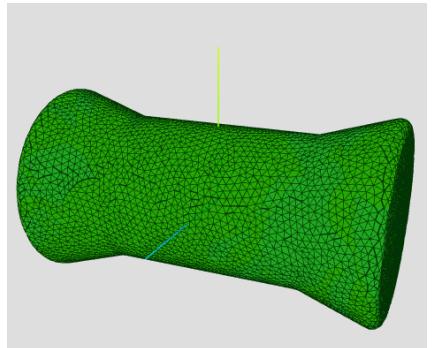
51_SolidangleScale.json

```
{  
  "Settings": { "tsl": 0.025 },  
  "solidangle": { "type": "solidangle",  
                 "c": { "x": 0, "y": -0.3, "z": 0.2 }, "r": 1.1, "o": 45 },  
  "Scale": { "type": "scale", "op": "solidangle",  
            "whd": { "x": 1.3, "y": 1.5, "z": 1.0 } }  
}
```



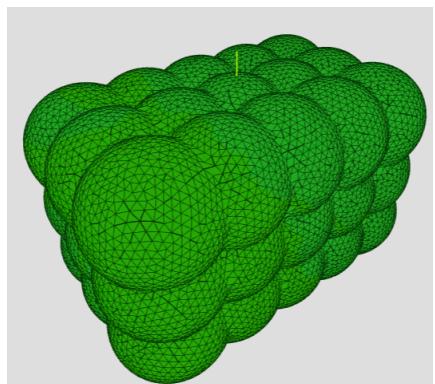
52_CylinderElongateX.json

```
{  
  "Cone": { "type": "cylinder", "rb": 0.4, "rt": 0.7, "h": 0.6, "rotz": 90, "s": 0.06 },  
  "Elong" : { "type": "elongateX", "op": "Cone", "d": 0.8 }  
}
```



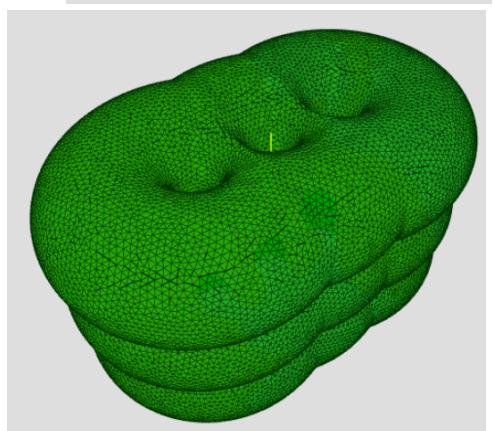
53_SphereRepeat.json

```
{  
  "Sph": { "type": "sphere", "r": 0.55 },  
  "Repeat": { "type": "repeat", "op": "Sph", "d": 0.55, "nnn": { "x": 3, "y": 2, "z": 4 } }  
}  
// "y": 2, "z": 4: even numbers are rounded up
```



54_TorusRepeat.json

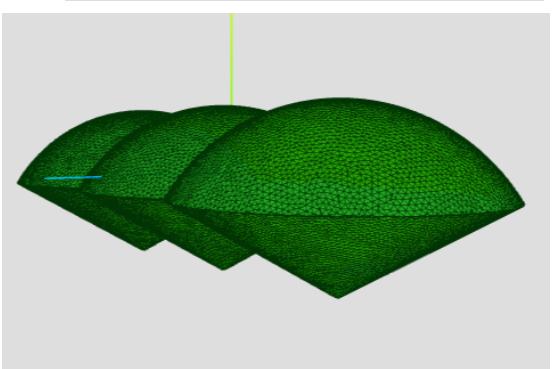
```
{  
  "Settings": { "tsl": 0.07 },  
  "Tor": { "type": "torus", "R": 0.8, "r": 0.6 },  
  "Rep": { "type": "repeat", "op": "Tor",  
           "d": 0.7, "nnn": { "x": 3, "y": 3, "z": 0 } }  
}
```



Very sensitive to parameters due to the holes and sharp points that are created.

55_SolidangleRepeat.json

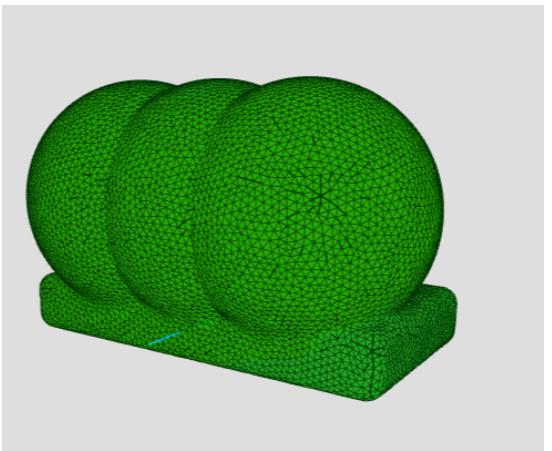
```
{  
  "Settings": { "tsl": 0.022 },  
  "solidangle": { "type": "solidangle",  
                 "c": { "x": 0.0, "y": -0.5, "z": 0.05 }, "o": 60 },  
  "Rep": { "type": "repeat",  
           "op": "solidangle", "d": 1.0, "nnn": { "x": 3, "y": 0, "z": 0 } }  
}
```



Very sensitive to parameters due to the sharp points that are created.

56_unionBoxSphereRepeat.json

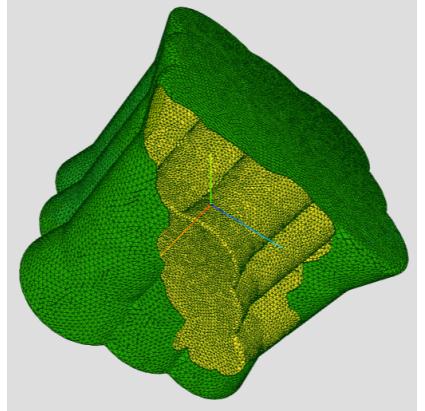
```
{  
  "Sph": { "type": "sphere", "r": 0.8, "c": { "x": 0.0, "y": 0.75, "z": 0.25 } },  
  "Box": { "type": "box", "whd": { "x": 1.4, "y": 0.4, "z": 1.4 }, "s": 0.11 },  
  "Union": { "type": "union", "op1": "Sph", "op2": "Box", "s": 0.08 },  
  "Rep": { "type": "repeat",  
           "op": "Union", "d": 0.8, "nnn": { "x": 3, "y": 0, "z": 0 } }  
}
```



57_ConeElongateRepeat.json

```
{  
  "Cone": { "type": "cylinder", "rb": 0.4, "rt": 0.7, "h": 0.6,  
            "rotz": 90, "s": 0.06 },  
  "Elong": { "type": "elongateX", "op": "Cone", "d": 0.8 },  
  "Rep": { "type": "repeat",  
            "op": "Elong", "d": 0.87, "nnn": { "x": 0, "y": 3, "z": 3 } }  
}
```

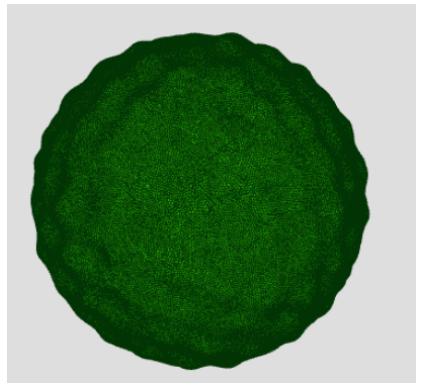
used: triangulation sequence >>>



58_SphereDisplaceSinXYZ.json

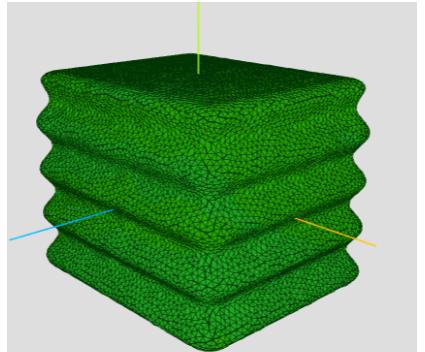
```
{  
  "Settings": { "tsl": 0.044 },  
  "Sph": { "type": "sphere", "r": 1.64 },  
  "Displ": { "type": "displace", "op": "Sph", "func": "sinXYZ", "d": 0.03, "f": 2.04 }  
}
```

// See error causes at the end of the help.



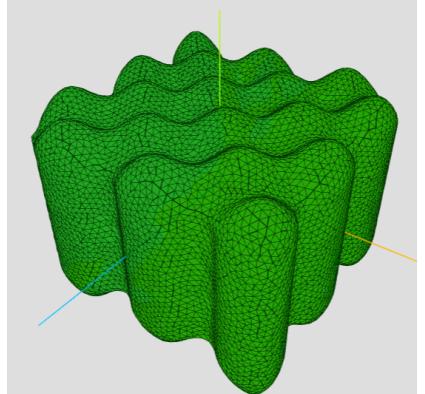
59_BoxDisplaceSinY.json

```
{  
  "Settings": { "tsl": 0.03 },  
  "Box": { "type": "box", "whd": { "x": 1.04, "y": 0.84, "z": 0.84 } },  
  "Displ": { "type": "displace", "op": "Box", "func": "sinY", "d": 0.035, "f": 4.34 }  
}
```



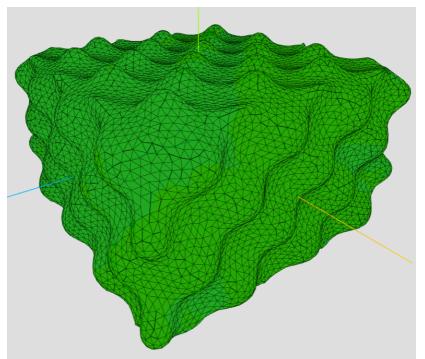
60_BoxDisplaceSinXZ.json

```
{  
  "Settings": { "tsl": 0.03 },  
  "Box": { "type": "box", "whd": { "x": 1.04, "y": 0.59, "z": 1.04 } },  
  "Displ": { "type": "displace", "op": "Box", "func": "sinXZ", "d": 0.06, "f": 2.34 }  
}
```



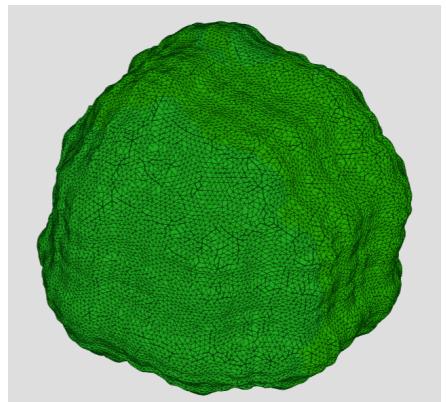
61_BoxDisplace.json

```
{  
  "Settings": { "tsl": 0.03 },  
  "Box": { "type": "box", "whd": { "x": 1.04, "y": 0.59, "z": 1.04 } },  
  "Displ": { "type": "displace", "op": "Box", "func": "sinXYZ", "d": 0.06, "f": 2.84 }  
}
```



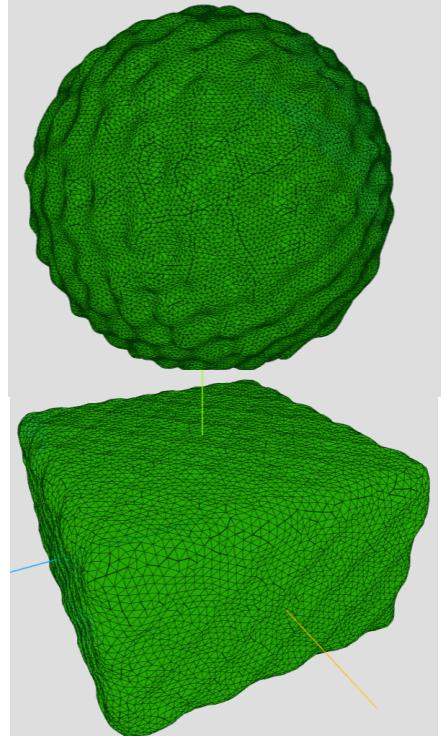
62_SphereDisplaceSinFBM.json

```
{  
  "Settings": { "tsl": 0.03 },  
  "Sph": { "type": "sphere", "r": 1.09 },  
  "Displ": { "type": "displace", "op": "Sph", "func": "sinFBM", "d": 0.14, "f": 4.34}  
}
```



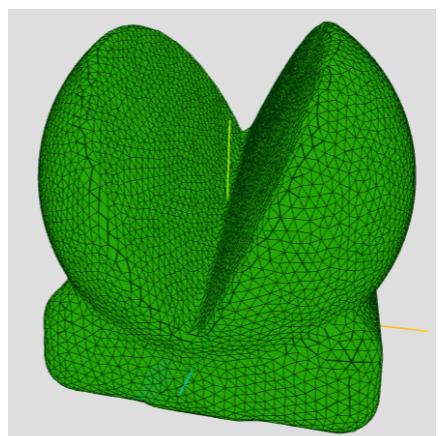
63_SphereDisplaceSinRND.json

```
{  
  "Settings": { "tsl": 0.03 },  
  "Sph": { "type": "sphere", "r": 1.09 },  
  "Displ": { "type": "displace", "op": "Sph", "func": "sinRND", "d": 0.02, "f": 4.34}  
}
```



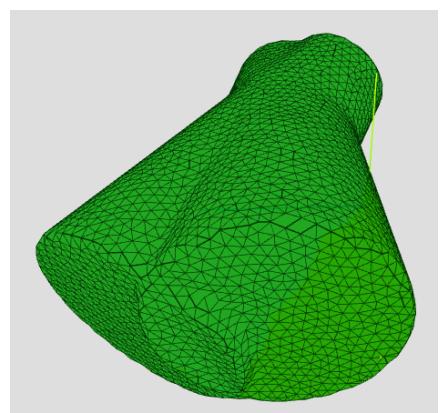
64_BoxDisplaceFBM.json

```
{  
  "Settings": { "tsl": 0.03 },  
  "Box": { "type": "box", "whd": { "x": 1.04, "y": 0.59, "z": 1.04 } },  
  "Displ": { "type": "displace", "op": "Box", "func": "FBM", "d": 0.01, "f": 3.84 }  
}
```



65_mirrorUnionSection.json

```
{  
  "Sph": { "type": "sphere", "r": 0.8, "c": { "x": 0.0, "y": 0.75, "z": 0.25 } },  
  "Box": { "type": "box", "whd": { "x": 1.4, "y": 0.4, "z": 1.4 }, "s": 0.11 },  
  "UnionA": { "type": "union", "op1": "Sph", "op2": "Box", "s": 0.07 },  
  "Section": { "type": "section", "op": "UnionA", "s": 0.06,  
    "planes": [ { "normal": { "x": 1.4, "y": 0.6, "z": 0.3}, "d": 0.2 } ] },  
  "MirrorYZ": { "type": "mirrorYZ", "op": "Section"},  
  "UnionB": { "type": "union", "op1": "Section", "op2": "MirrorYZ", "s": 0.07 }  
}
```



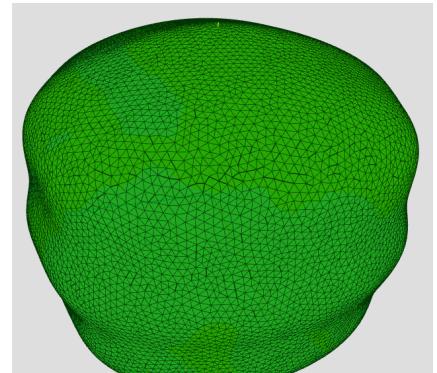
66_mirrorPlaneUnion.json

```
{  
  "Settings": { "tsl": 0.07 },  
  "Cone": { "type": "cylinder", "c": { "x": -0.2, "y": 0.05, "z": 0.5 },  
    "rb": 0.4, "rt": 0.7, "h": 1.2, "rotz": 80, "s": 0.04 },  
  "Mirrored": {  
    "type": "mirrorPlane",  
    "op": "Cone",  
    "normal": { "x": -0.2, "y": -0.3, "z": 0.86 },  
    "d": 0.75 },  
  "Union": { "type": "union", "op1": "Cone", "op2": "Mirrored", "s": 0.04 }  
}
```



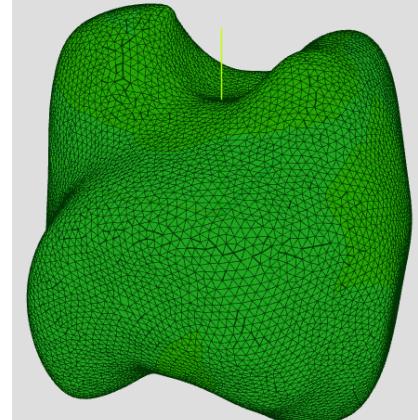
67_distortSinSphere.json

```
{  
  "Settings": { "tsl":0.04 },  
  "Sph": { "type":"sphere", "r": 1.1 },  
  "Distort": {  
    "type": "distort",  
    "op": "Sph",  
    "mode": "sin",  
    "dxyz": { "x":0.05, "y":0.12, "z":0.0 },  
    "fxyz": { "x":7.0, "y":8.0, "z":6.0 }  
  }  
}
```



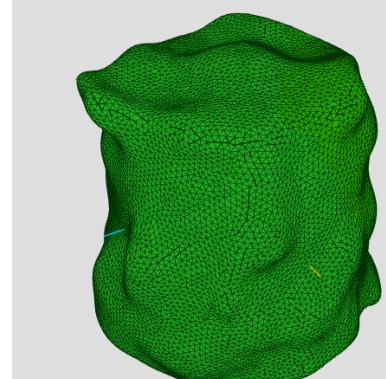
68_distortNoiseSphere.json

```
{  
  "Settings": { "tsl":0.04 },  
  "Sph": { "type":"sphere" },  
  "Dist": {  
    "type": "distort",  
    "op": "Sph",  
    "mode": "noise",  
    "dxyz": { "x":0.2, "y": 0.5, "z":0.4},  
    "fxyz": { "x": null , "y": 1.4, "z": null }  
  }  
}
```



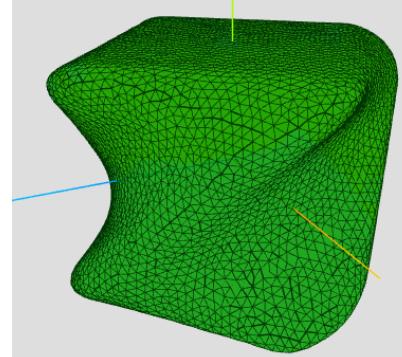
69_distortVnoiseCylinder.json

```
{  
  "Settings": { "tsl":0.04 },  
  "Cyl": { "type":"cylinder", "rb": 1.2, "rt": 0.95, "h": 1.1, "s": 0.125 },  
  "Distort": {  
    "type": "distort",  
    "op": "Cyl",  
    "mode": "vnoise",  
    "dxyz": { "x":0.1, "y":0.12, "z":0.15 },  
    "fxyz": { "x":3.0, "y":1.9, "z":2.0 }  
  }  
}
```



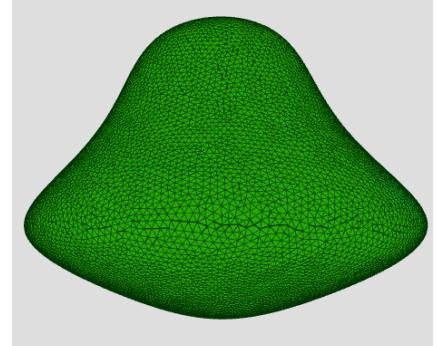
70_distortTwistBoxr.json

```
{  
  "Settings": { "tsl":0.02 },  
  "Box": { "type":"box", "s":0.2 },  
  "Dist": {  
    "type": "distort",  
    "op": "Box",  
    "mode": "twist",  
    "dxyz": { "x": null, "y": 0.7, "z": null},  
    "fxyz": { "x": null, "y": 1.5, "z": null}  
  }  
}
```



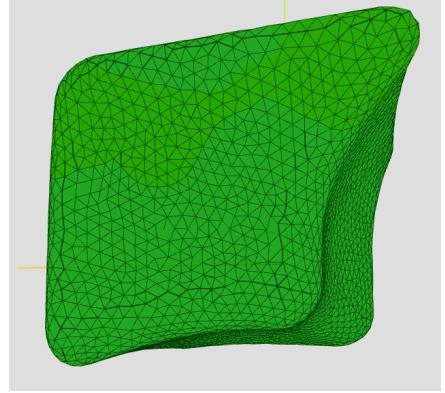
71_distortPinchSphere.json

```
{  
  "Settings": { "tsl":0.04 },  
  "Sph": { "type":"sphere" },  
  "Dist": { "type": "distort",  
    "op": "Sph",  
    "mode": "pinch",  
    "dxyz": { "x": null, "y": 0.3, "z": null },  
    "fxyz": { "x": null, "y": 3.0, "z": null }  
  }  
}
```



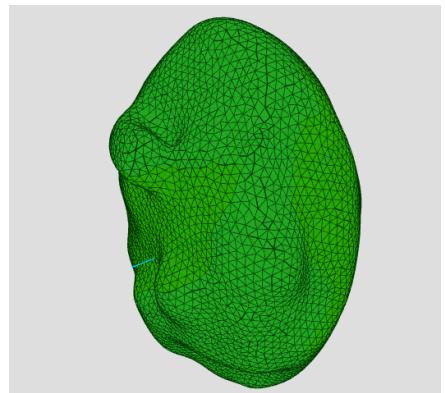
72_distortNoiseBox.json

```
{  
  "Settings": { "tsl":0.04 },  
  "Box": { "type":"box", "s":0.15 },  
  "Dist": {  
    "type": "distort",  
    "op": "Box",  
    "mode": "noise",  
    "dxyz": { "x":0.4, "y": 0.3, "z":0.4},  
    "fxyz": { "x":null, "y": 0.8, "z":null }  
  }  
}
```



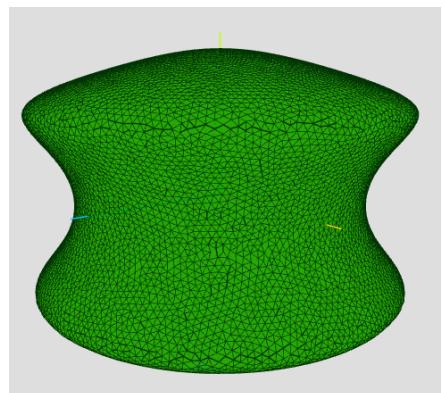
72_distortNoiseEllipsoid.json

```
{  
  "Settings": { "tsl": 0.07},  
  "Ellipsoid": { "type": "ellipsoid", "rxyz": { "x": 1.2, "y": 1.7, "z": 0.9 } },  
  "Dis": {  
    "type": "distort",  
    "op": "Ellipsoid",  
    "mode": "noise",  
    "dxyz": { "x":0.10, "y":0.12, "z":0.2 },  
    "fxyz": { "x":null, "y":2.1, "z":null }  
  }  
}
```



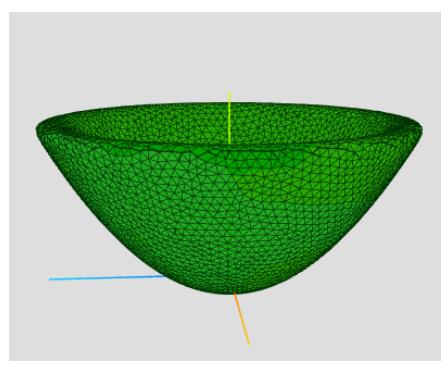
73_distortTwistSphere.json

```
{  
  "Settings": { "tsl":0.04 },  
  "Sph": { "type":"sphere", "r": 0.9 },  
  "Distort": {  
    "type": "distort",  
    "op": "Sph",  
    "mode": "twist",  
    "dxyz": { "x": null, "y": 0.2, "z": null },  
    "fxyz": { "x": null, "y": 5.0, "z": null }  
  }  
}
```



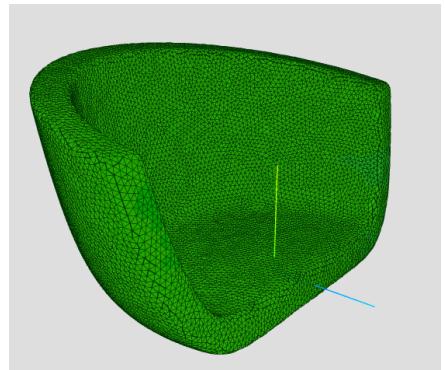
74_ParaboloidHollow.json

```
{  
  "Parab": { "type": "paraboloid", "hollow": true }  
}
```



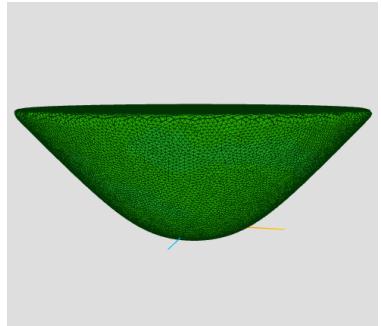
75_PowerparaboloidHollowSection.json

```
{  
  "Power": { "type": "powerparaboloid",  
            "a": 0.01, "n": 9, "h": 1.5, "s": 0.06, "hollow": true},  
  "Sect": { "type": "section", "op": "Power",  
            "planes": [{ "normal": { "x":0, "y":0.2, "z":1 }, "d": 0.4, "s": 0.04 }]  
          }  
}
```



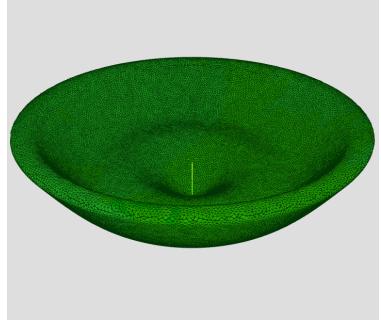
76_Cubic.json

```
{  
  "Settings": { "tsl": 0.04 },  
  "Cubic": { "type": "cubic", "a": 1.0, "h": 1.5, "s": 0.12, "hollow": false}  
}
```



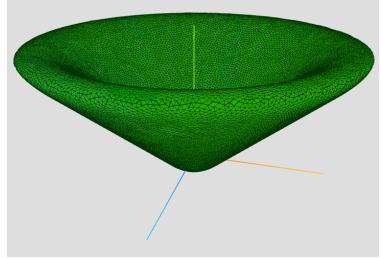
77_CosparabolHollow.json

```
{  
  "CosParabol": { "type": "cosparabol",  
                 "a": 0.19, "n": 2, "b": 0.20, "f": 2.7, "h": 2.1 , "s": 0.16, "hollow":  
true}  
}
```



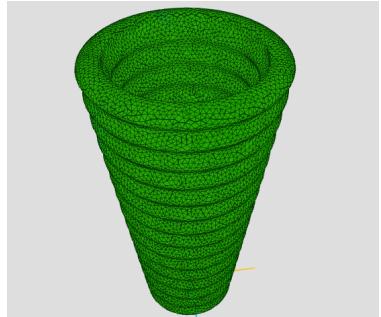
78_RotateBevelHollow.json

```
{  
  "Settings": {"tsl": 0.025},  
  "RotateBevel": { "type": "rotate", "func": "bevel",  
                 "bev": 0.95, "r": 0.09, "h": 1.3, "hollow": true, "s": 0.14}  
}
```



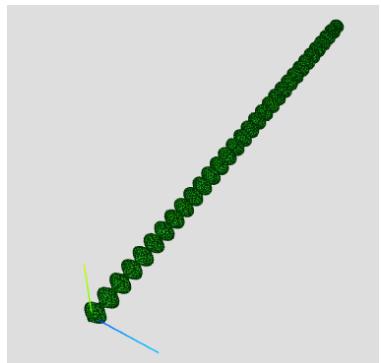
79_RotateBevelcosHollow.json

```
{  
  "RotateBevelCos": { "type": "rotate", "func": "bevelcos",  
                     "bev": 8.5, "r": 0.45, "a": 0.03, "f": 24.8, "h": 4.0, "hollow": true, "s": 0.12 }  
}
```



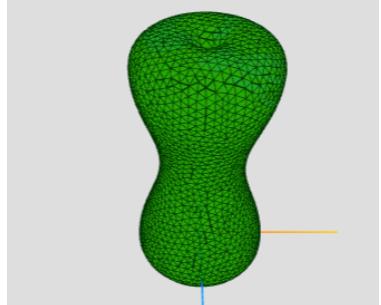
80_RotateSin.json

```
{  
  "RotateSin": { "type": "rotate", "func": "sin",  
                "a": 0.06, "f": 23.8, "r": 0.12, "h": 8.43, "rotz": 90, "s": 0.09 }  
}
```



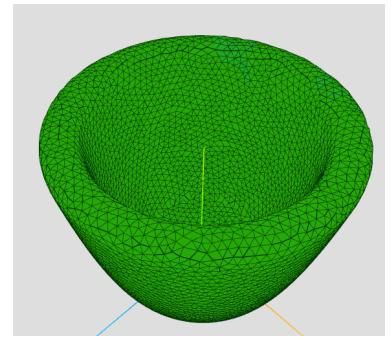
81_RotateCos.json

```
{  
  "RotateCos": { "type": "rotate", "func": "cos",  
                "a": 0.08, "f": 6.28, "r": 0.45, "h": 1.5, "s": 0.16 }  
}
```



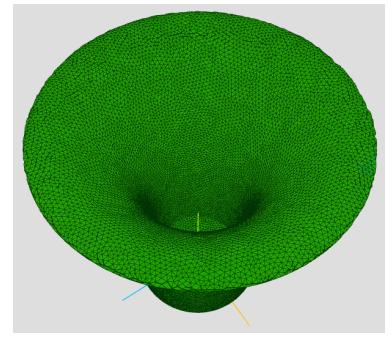
82_RotatePow5.json

```
{  
  "RotatePow5": { "type": "rotate", "func": "pow",  
                 "a": 0.6, "n": 5, "r": 0.0, "h": 1.5, "hollow": true, "s": 0.1 }  
}
```



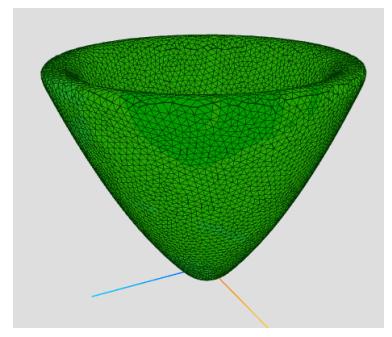
83_RotateRoot5.json

```
{  
  "Settings": {"tsl": 0.03},  
  "RotateRoot": { "type": "rotate", "func": "root",  
                 "a": 0.008, "n": 5, "hollow": true, "r": 0.68, "h": 2.7, "s": 0.18 }  
}
```



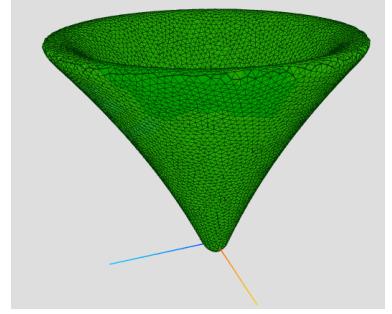
84_RotateExpHollow.json

```
{  
  "RotateExp": { "type": "rotate", "func": "exp",  
                "a": 0.9, "r": 0.1, "h": 1.99, "hollow": true, "s": 0.11 }  
}
```



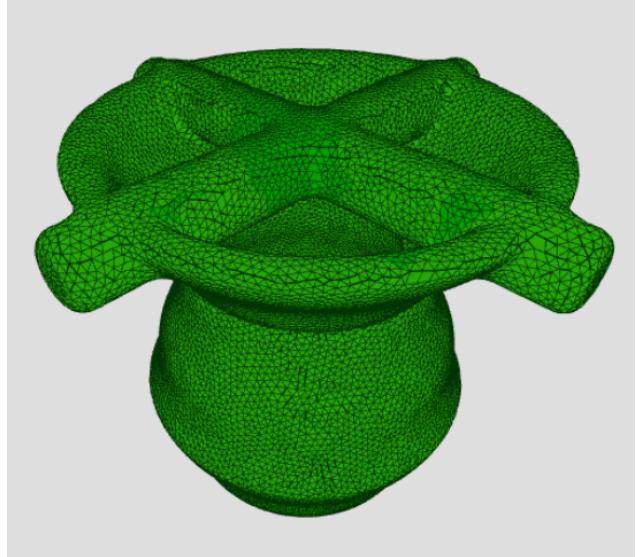
85_RotateLogHollow.json

```
{  
  "RotateLog": { "type": "rotate", "func": "log",  
                "a": 0.4, "r": 0.0, "h": 2.22, "hollow": true, "s": 0.12 }  
}
```



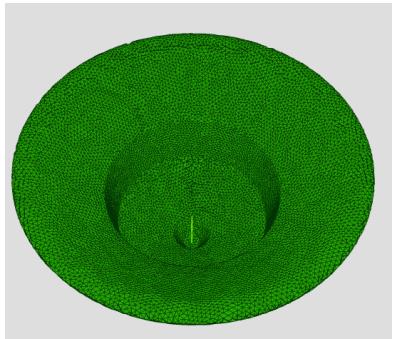
86_RotateLinescosHollowUnion.json

```
{ "Settings": { "tsl": 0.055 },  
  "RotateLcos": { "type": "rotate", "func": "linescos", "r": 0.5,  
                 "a": 0.02, "f": 9.43, "h": 3.0, "hollow": true, "s": 0.12 },  
  "points": [  
    { "h": 0.0, "r": 0.5 },  
    { "h": 0.8, "r": 0.8 },  
    { "h": 2.0, "r": 0.4 },  
    { "h": 2.8, "r": 1.0 }  
  ],  
  "Cyl_1": { "type": "cylinder", "c": { "x": 0.0, "y": 2.7, "z": 0.0 },  
             "rb": 0.12, "rt": 0.12, "h": 1.7, "rotz": 90, "s": 0.1 },  
  "Cyl_2": { "type": "cylinder", "c": { "x": 0.0, "y": 2.7, "z": 0.0 },  
             "rb": 0.12, "rt": 0.12, "h": 1.7, "rotx": 90, "s": 0.1 },  
  "Union": { "type": "union",  
            "op1": "Cyl_1", "op2": "Cyl_2", "s": 0.1 },  
  "OP": { "type": "union",  
          "op1": "RotateLcos", "op2": "Union", "s": 0.1 }  
}
```



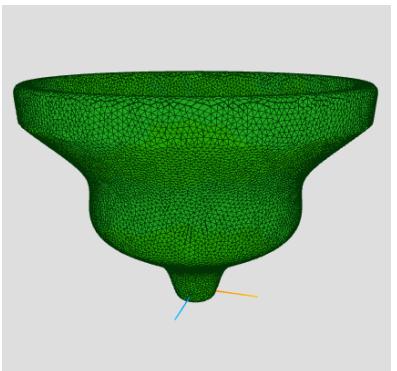
87_RotateLinesHollow.json

```
{ "Settings": { "tsl": 0.06 },
  "RotataLines": {
    "type": "rotate", "func": "lines", "r": 0.1, "h": 3.1, "hollow": true, "s": 0.12,
    "points": [
      { "h": 0.0, "r": 0.1 },
      { "h": 0.5, "r": 0.33},
      { "h": 1.1, "r": 1.2 },
      { "h": 1.9, "r": 1.3 },
      { "h": 2.6, "r": 2.0 }
    ] }
}
```



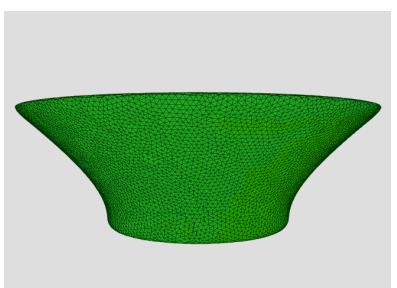
88_RotateCurveHollow.json

```
{ "Settings": { "tsl": 0.06},
  "RotataLines": {
    "type": "rotate", "func": "curve", "r": 0.1, "h": 3.1, "hollow": true, "s": 0.12,
    "points": [
      { "h": 0.0, "r": 0.1 },
      { "h": 0.5, "r": 0.33},
      { "h": 1.1, "r": 1.2 },
      { "h": 1.9, "r": 1.3 },
      { "h": 2.6, "r": 2.0 }
    ] }
}
```



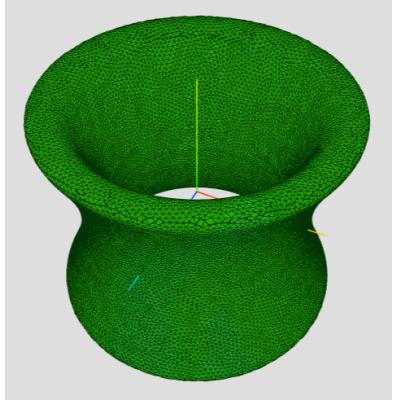
89_RotateHyperbolaHollow.json

```
{ "Settings": { "tsl": 0.05 },
  "Hyp": { "type": "rotate", "func": "hyperbola",
            "a": 0.75, "r": 0.0, "h": 1.6, "s": 0.12, "hollow": true}
}
```



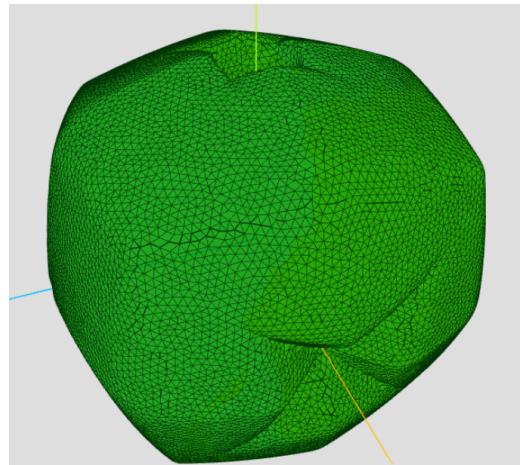
89_HyperboloidHollow.json

```
{ "Settings": { "tsl": 0.035 },
  "Hyperboloid": { "type": "hyperboloid",
                  "a": 1.11, "r": -0.25, "h": 1.2, "s": 0.12, "hollow": true }
}
```



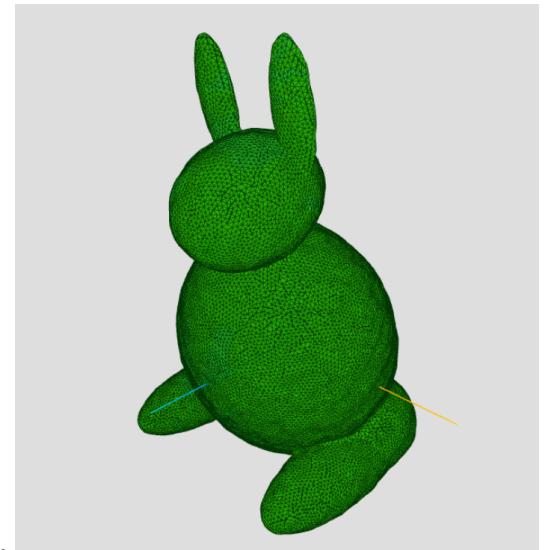
90_BlendingSubtraction.json

```
{ "Settings": { "tsl": 0.025 },
  "Octahed": { "type": "octahedron", "r": 1.12 },
  "Sph": { "type": "sphere" },
  "Blend_A": { "type": "blending",
               "op1": "Sph", "op2": "Octahed", "b": 0.5, "s": 0.01 },
  "BoxFrame": { "type": "boxframe",
                "fx": 0.15, "fy": 0.22, "fz": 0.32, "s": 0.06 },
  "Blend_B": { "type": "blending",
               "op1": "Blend_A", "op2": "BoxFrame", "b": 0.5, "s": 0.01 },
  "Cone": { "type": "cylinder",
            "rb": 0.012, "rt": 0.15, "h": 1.5, "s": 0.02 },
  "Sub": { "type": "subtraction",
            "op1": "Blend_B", "op2": "Cone", "s": 0.04 }
}
```



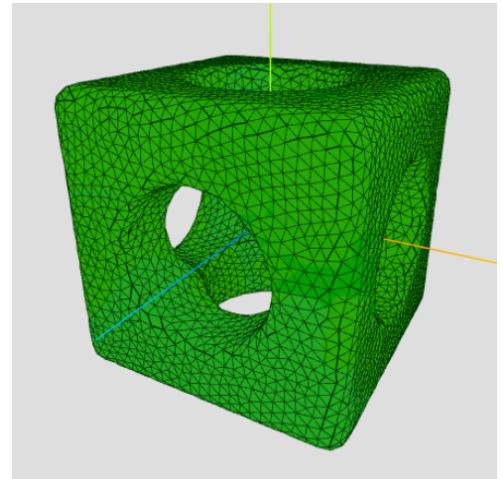
91_Rabbit.json

```
{ "Settings": { "tsl": 0.022 },
  "Body": { "type": "ellipsoid",
    "rxyz": { "x": 0.6, "y": 0.8, "z": 0.6 }, "rotx": -30 },
  "Head": { "type": "ellipsoid",
    "rxyz": { "x": 0.3, "y": 0.3, "z": 0.4 },
    "c": { "x": 0.0, "y": 0.9, "z": 0.4 }, "rotx": -10 },
  "EarRight": { "type": "ellipsoid", "c": { "x": -0.21, "y": 1.3, "z": 0.4 },
    "rxyz": { "x": 0.1, "y": 0.29, "z": 0.05 }, "rotx": -20 },
  "EarLeft": { "type": "mirrorYZ", "op": "EarRight" },
  "HipRight": { "type": "sphere", "r": 0.3,
    "c": { "x": -0.42, "y": -0.45, "z": -0.25 } },
  "HipLeft": { "type": "mirrorYZ", "op": "HipRight" },
  "LegRight": { "type": "ellipsoid",
    "c": { "x": -0.51, "y": -0.55, "z": 0.10 },
    "rxyz": { "x": 0.2, "y": 0.2, "z": 0.6 } },
  "LegLeft": { "type": "mirrorYZ", "op": "LegRight" },
  "Tail": { "type": "sphere", "r": 0.15, "c": { "x": 0.0, "y": -0.55, "z": -0.6 } },
  "Rabbit": { "type": "union",
    "op": [ "Body", "Head", "EarRight", "EarLeft", "HipRight", "HipLeft", "LegRight", "LegLeft", "Tail" ], "s": 0.025 },
  "RabbitDisp": { "type": "displace", "op": "Rabbit", "func": "FBM", "d": 0.0048, "f": 3.85 } }
```



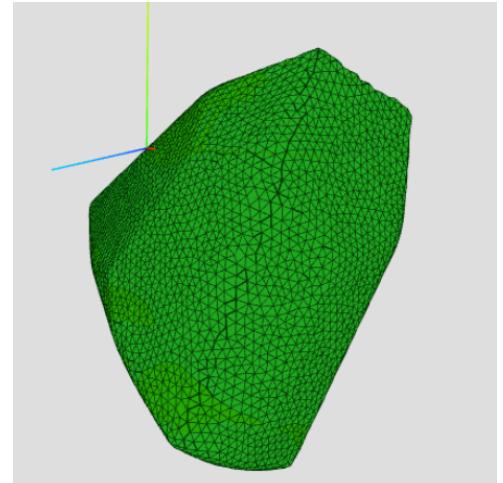
92_BoxSubCapsules.json

```
{ "Settings": { "startPoint": { "x": 0, "y": -0.44, "z": -10.0 } },
  "Box": { "type": "box" },
  "CapsX": { "type": "capsule", "r": 0.3, "from": { "x": -0.7, "y": 0.0, "z": 0.0 }, "to": { "x": 0.7, "y": 0.0, "z": 0.0 } },
  "CapsY": { "type": "capsule", "r": 0.3, "from": { "x": 0.0, "y": -0.7, "z": 0.0 }, "to": { "x": 0.0, "y": 0.7, "z": 0.0 } },
  "CapsZ": { "type": "capsule", "r": 0.3, "from": { "x": 0.0, "y": 0.0, "z": -0.7 }, "to": { "x": 0.0, "y": 0.0, "z": 0.7 } },
  "Sub": { "type": "subtraction", "op": [ "Box", "CapsX", "CapsY", "CapsZ" ], "s": 0.06 } }
```



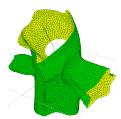
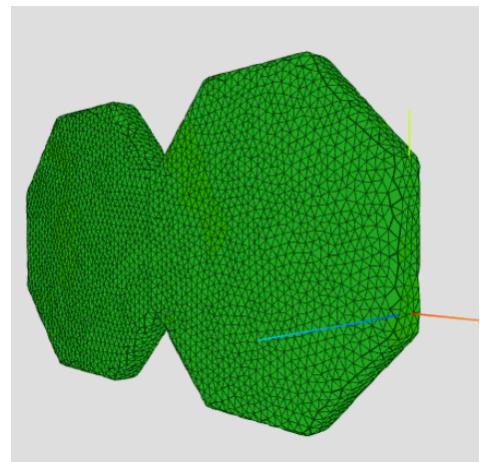
93_BlandingRotation.json

```
{ "Box": { "type": "box" },
  "triPrism": { "type": "prism",
    "n": 3, "r": 0.9, "d": 0.66, "roty": 45 },
  "Blend": { "type": "blending",
    "op1": "triPrism", "op2": "Box", "b": 0.25, "s": 0.3 },
  "Rotation": { "type": "rotation",
    "op": "Blend", "c": { "x": 0.5, "y": -0.5, "z": 0.0 }, "rotz": 35 } }
```



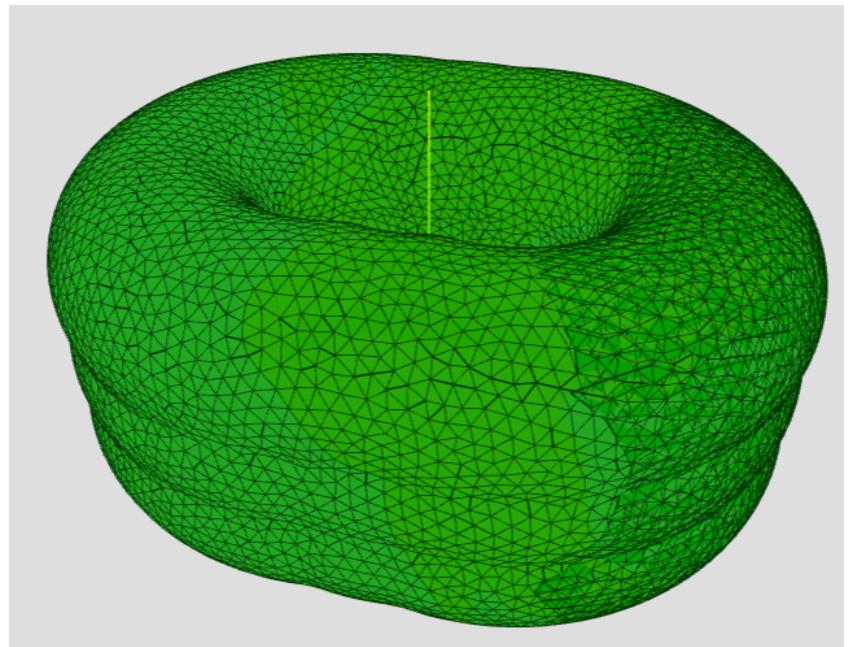
94_Translation.json

```
{ "Settings": { "tsl": 0.066 },
  "octPrism": { "type": "prism",
    "n": 8, "r": 1.0, "d": 0.12, "c": { "x": 1.0, "y": 0.0, "z": 0.0 }, "s": 0.06 },
  "Mirrored": { "type": "mirrorYZ", "op": "octPrism" },
  "Prism2": { "type": "union", "op": ["octPrism", "Mirrored"], "s": 0.04 },
  "Translation": { "type": "translation",
    "op": "Prism2", "c": { "x": -2.05, "y": 0.33, "z": 0.0 } } }
```



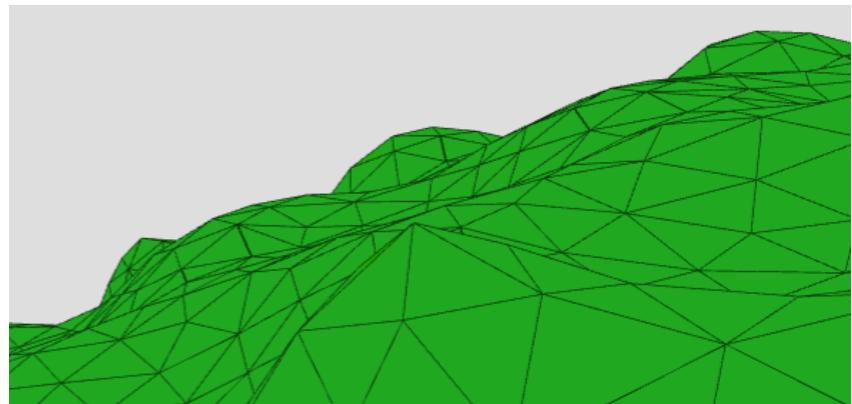
Errors:

In very unfavorable cases, the triangulation overflows several times and breaks off, or the result is unusable, as shown here in the picture.

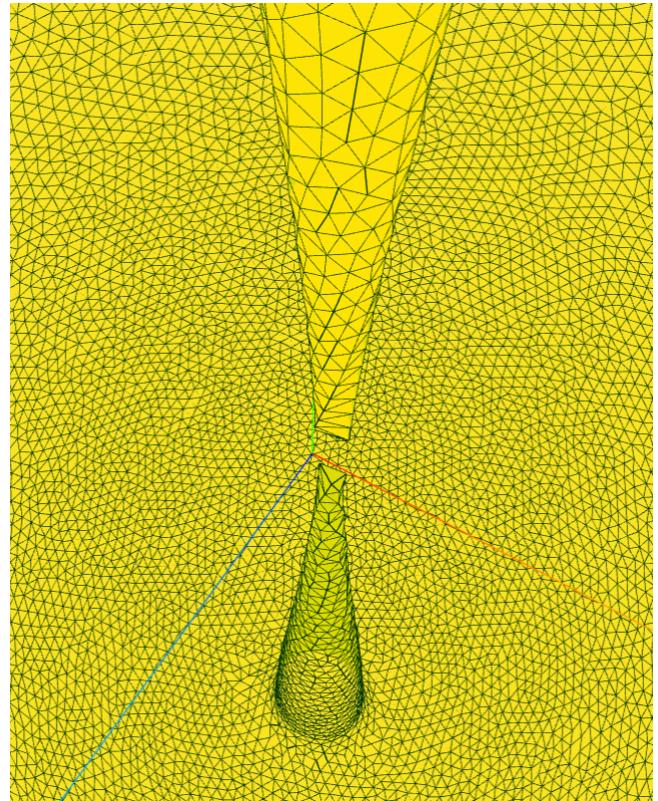


Displacement in several directions often results in sharp peaks or fractures. This causes the triangulation to fail. Try again with slightly different parameters.

It is still working here

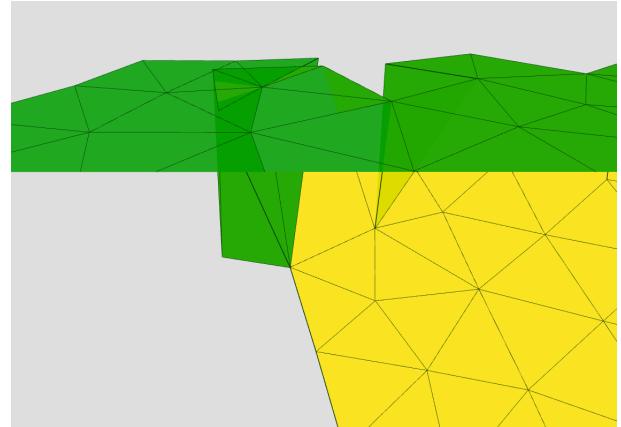
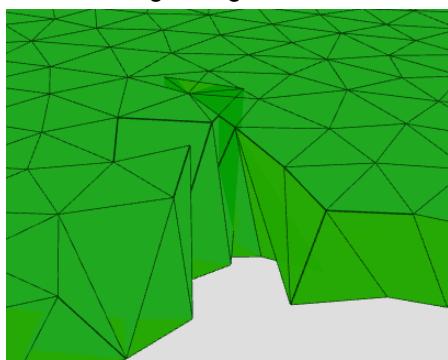


Sharp tips are problematic, here it just worked.
Interior view.



Here the triangulation algorithm fails under extreme conditions.
A significant reduction in the triangle length can sometimes
solve the problem.

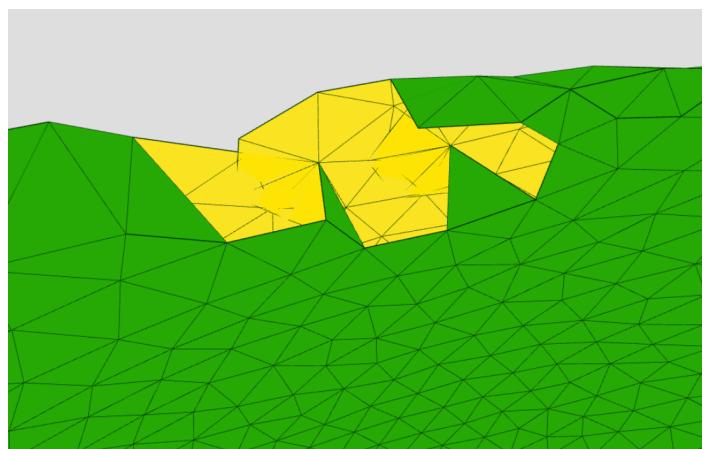
There is a result,
but it is unusable.



There are errors in the calculation
of the correct angles in the triangulation algorithm.

The two outer surfaces are only separated
by a distance of approximately the
length of the triangle ("Settings": { "tsl": ... }).

The geometry is unusable.



"type": "hyperboloid"
This deviation from the exact surface can only be found
by following the formation of the triangles
with the "triangulation sequence >>>>".

