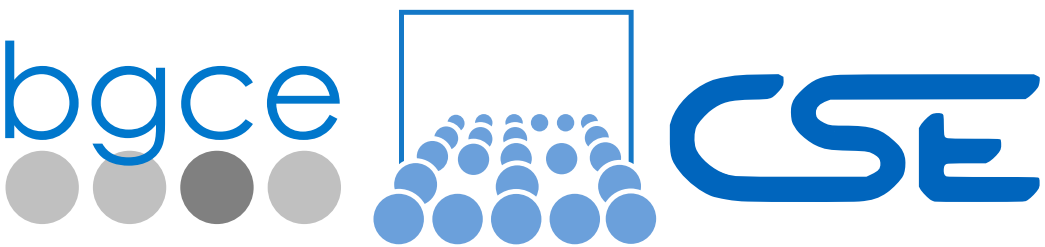
Technische Universität München

BGCE Project: CAD – Integrated Topology Optimization

BGCE Final Milestone Meeting

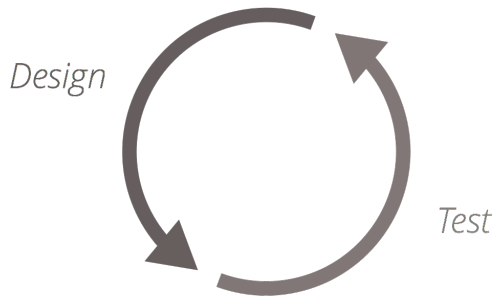
S. Joshi, J.C. Medina, F. Menhorn,
S. Reiz, B. Rüth, E. Wannerberg, A. Yurova

March 1, 2016



Motivation

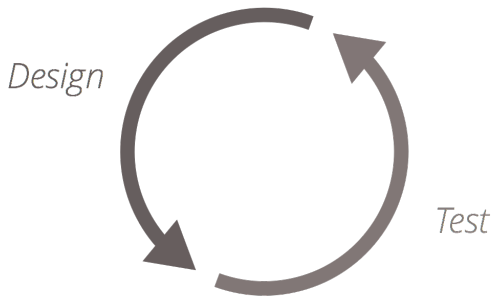
Current Design Process:



- Iterative and redundant
- Time consuming

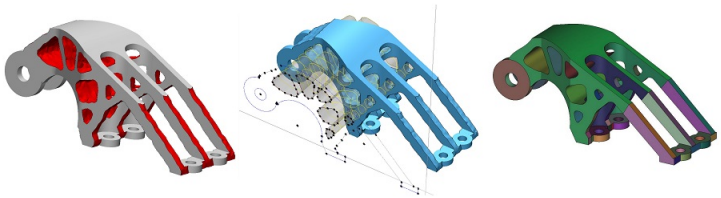
Motivation

Current Design Process:



- Iterative and redundant
- Time consuming

Topology optimization



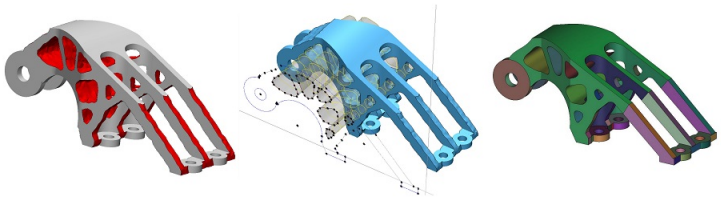
- Promoted by additive manufacturing

Motivation

Current Design Process:

- Iterative and redundant
- Time consuming

Topology optimization

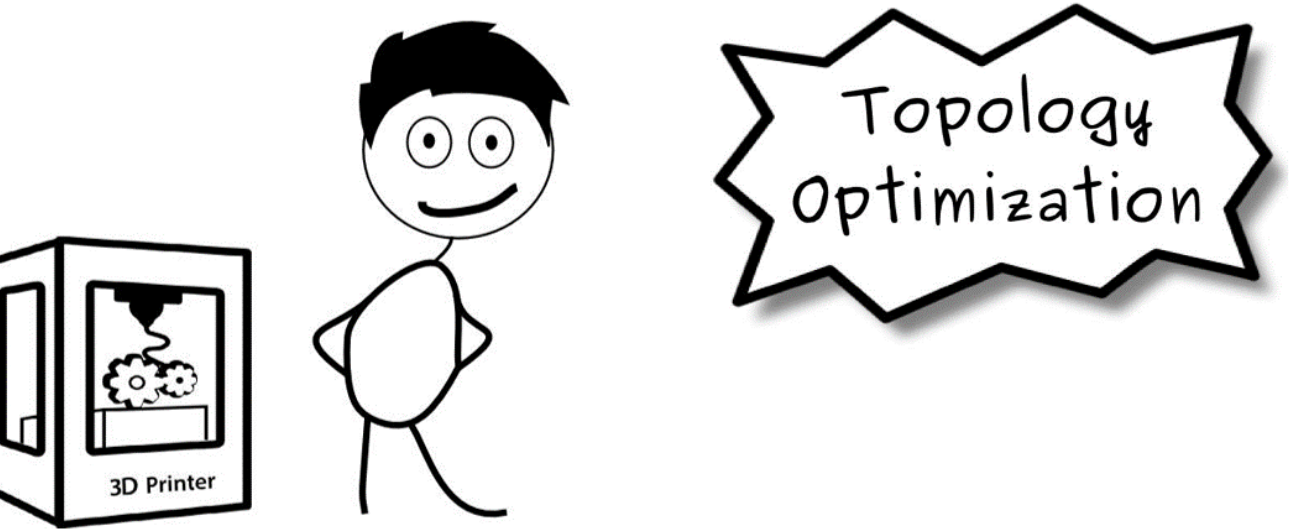


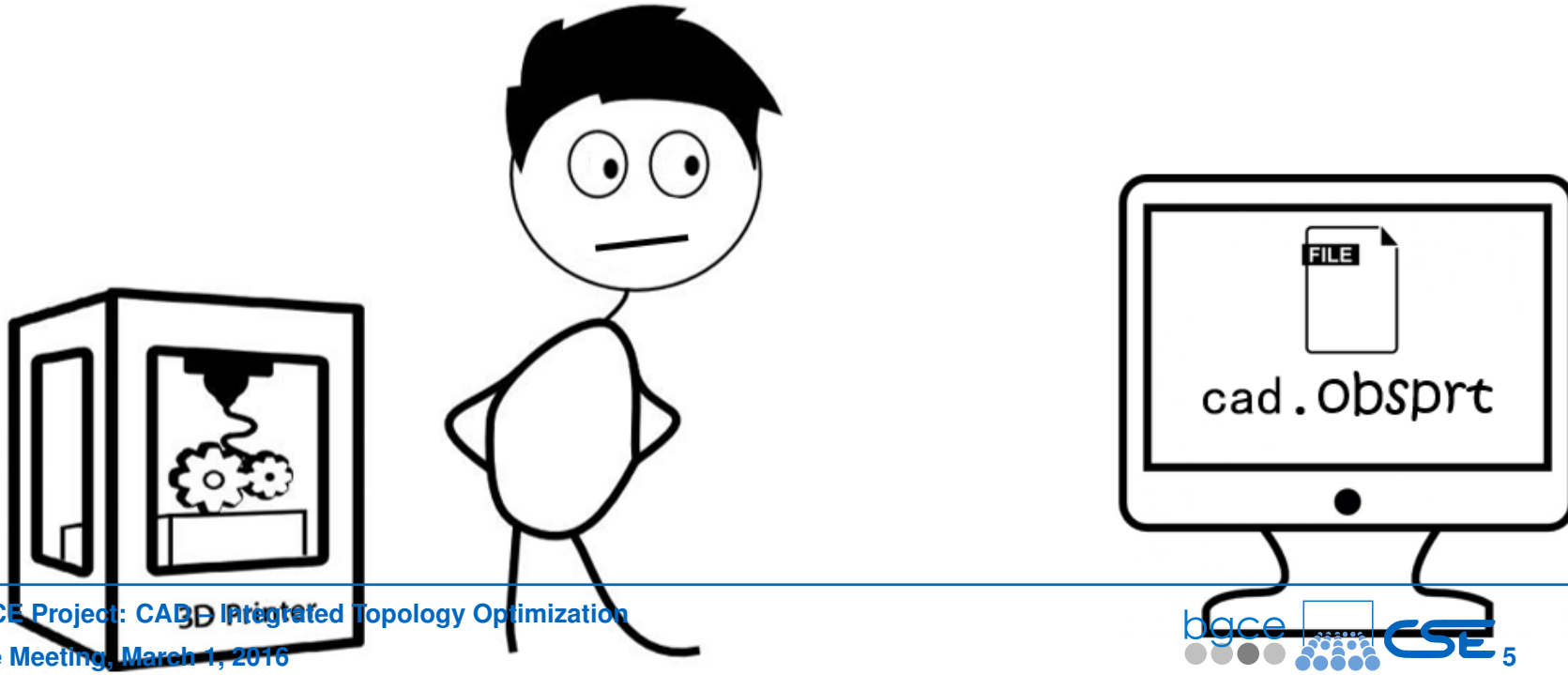
- Promoted by additive manufacturing

Focus:

Convert optimized geometry to **lightweight** and **scalable** CAD formats

Contents


Topology
Optimization
3D Printer


3D Printer
FILE
cad.obsprt

How hard is it to design a lamp?

Problem:

- The Engineer designer pendulum

Desired:

$\Rightarrow$ One click optimization

How hard is it to design a lamp?

Problem:

- The Engineer designer pendulum
- Top-Opt algorithms are a one way street

Desired:

⇒ One click optimization

⇒ A full circle optimization process

How hard is it to design a lamp?

Problem:	Desired:
• The Engineer designer pendulum	⇒ One click optimization
• Top-Opt algorithms are a one way street	⇒ A full circle optimization process
• Exotic input file types	⇒ Standardized input files

What they get

- One-step solution process
- Full 3-D optimization via Finite Elements
- Production-ready output geometry

DEMO

Features

Fully integrated design process

- CAD to CAD
- Turnkey
- Standardized I/O

Features

Fully integrated design process

- CAD to CAD
- Turnkey
- Standardized I/O

Control to the user

- Resolution
- Smoothness
- Localized Optimization

Features

Fully integrated design process

- CAD to CAD
- Turnkey
- Standardized I/O

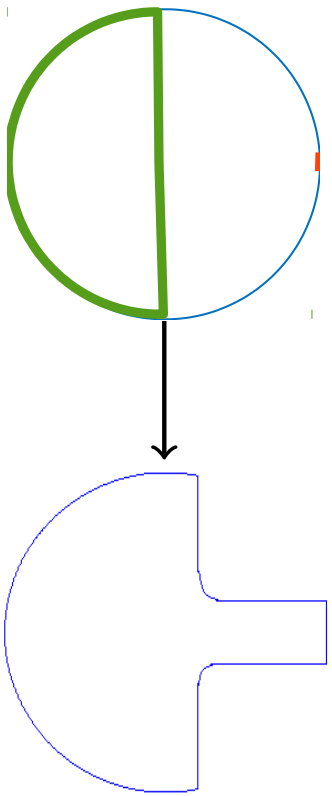
Control to the user

- Resolution
- Smoothness
- Localized Optimization

100% open source

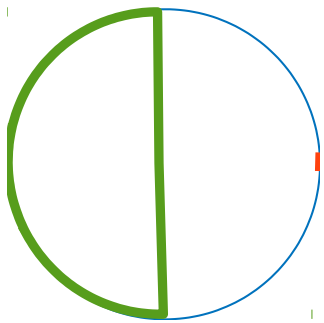
Overview: Workflow

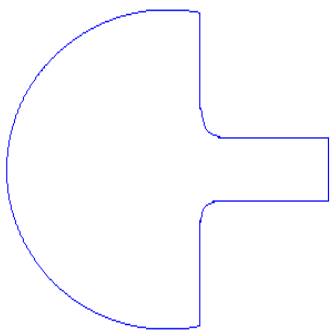
What the user sees



Overview: Workflow

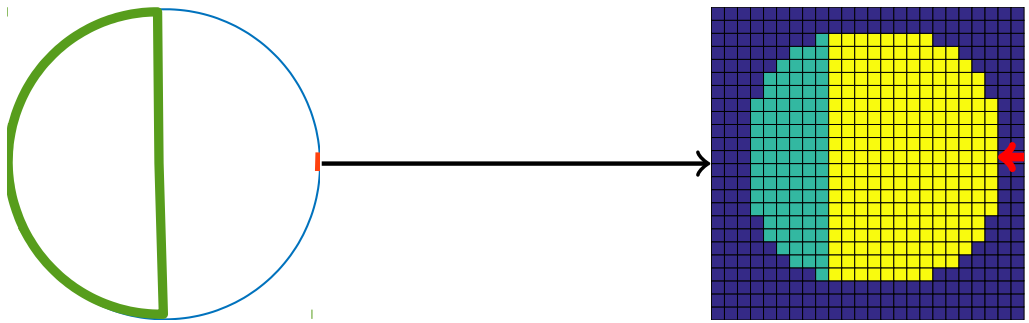
CAD design including specification of loads and fixtures

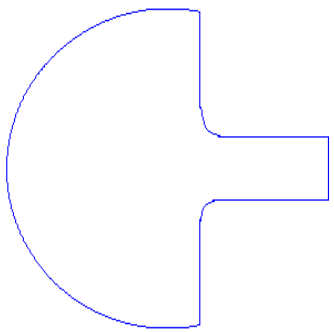




Overview: Workflow

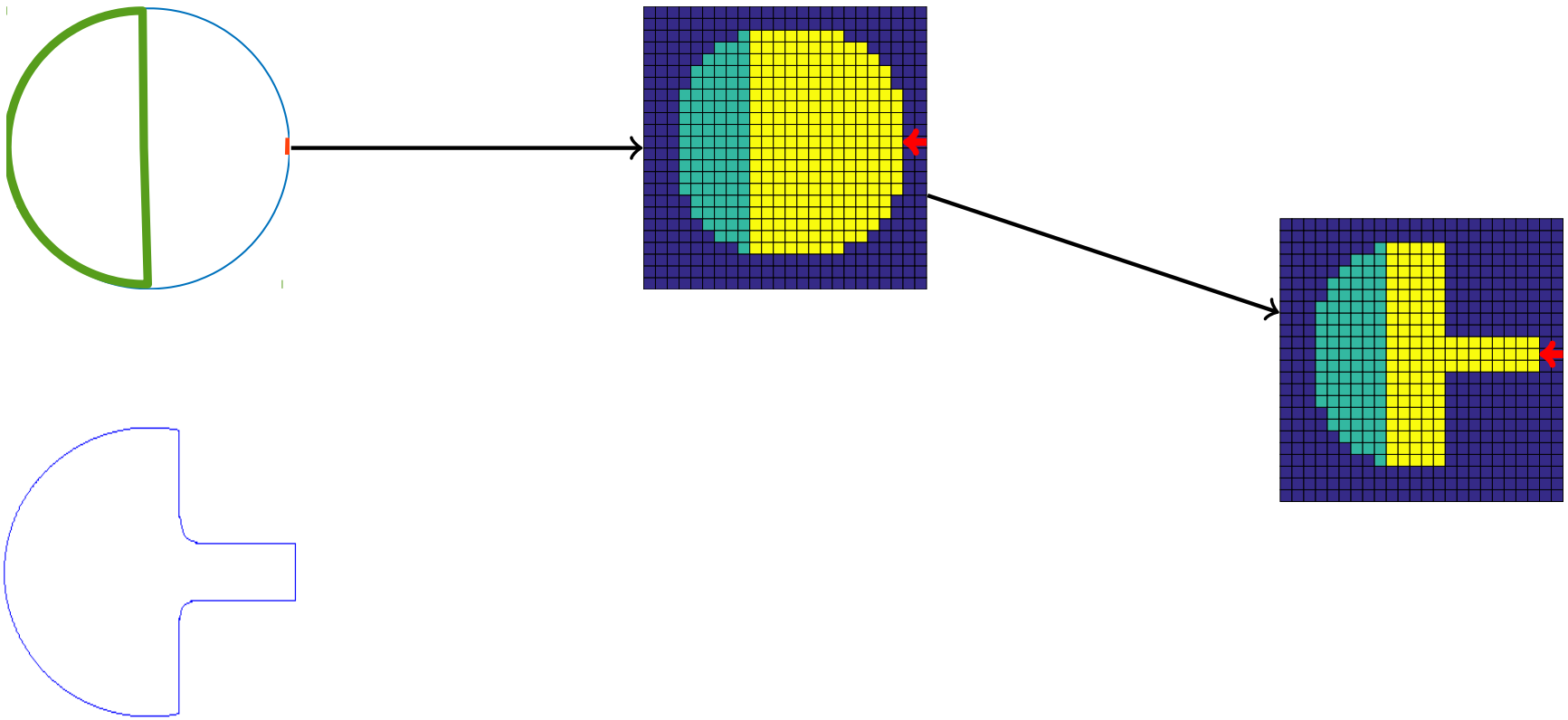
Voxelized topology





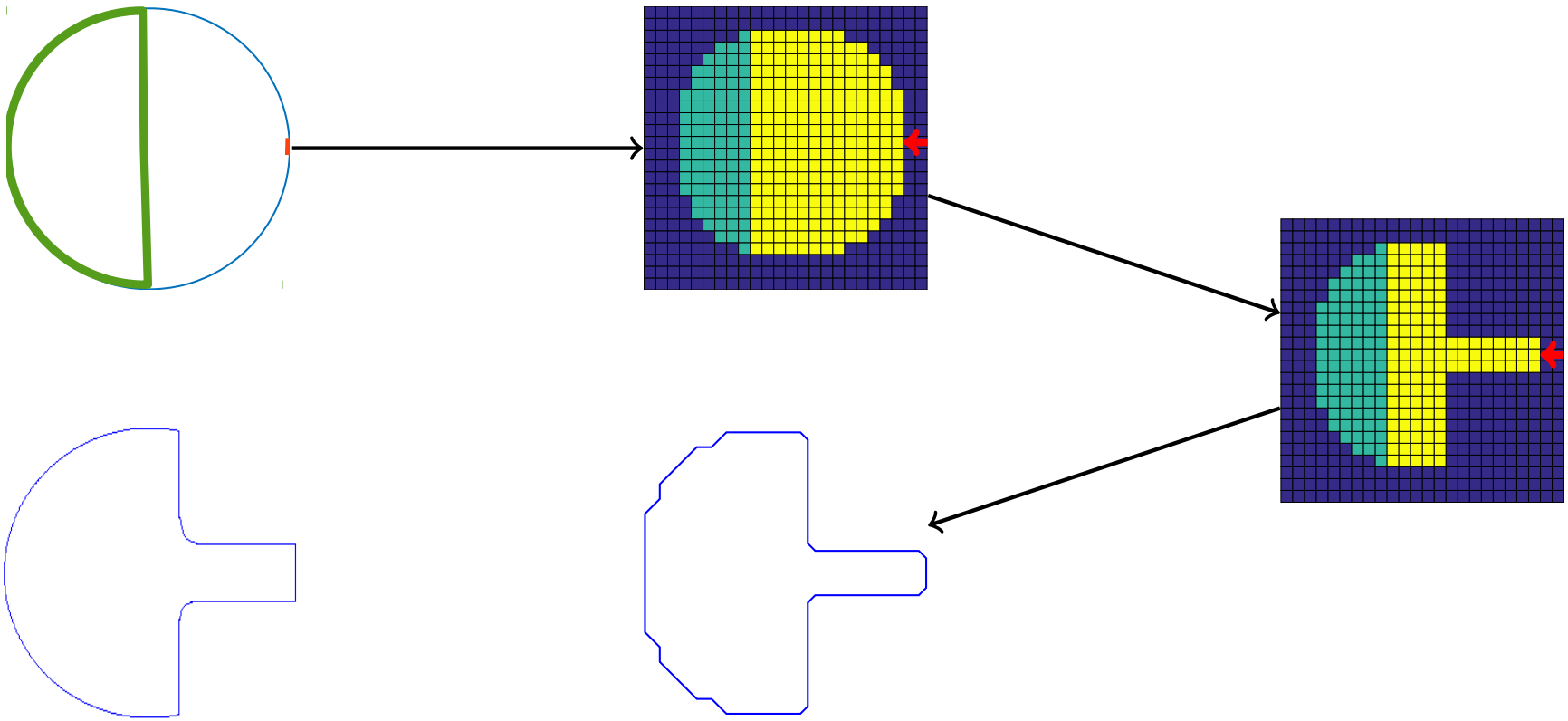
Overview: Workflow

Optimized topology



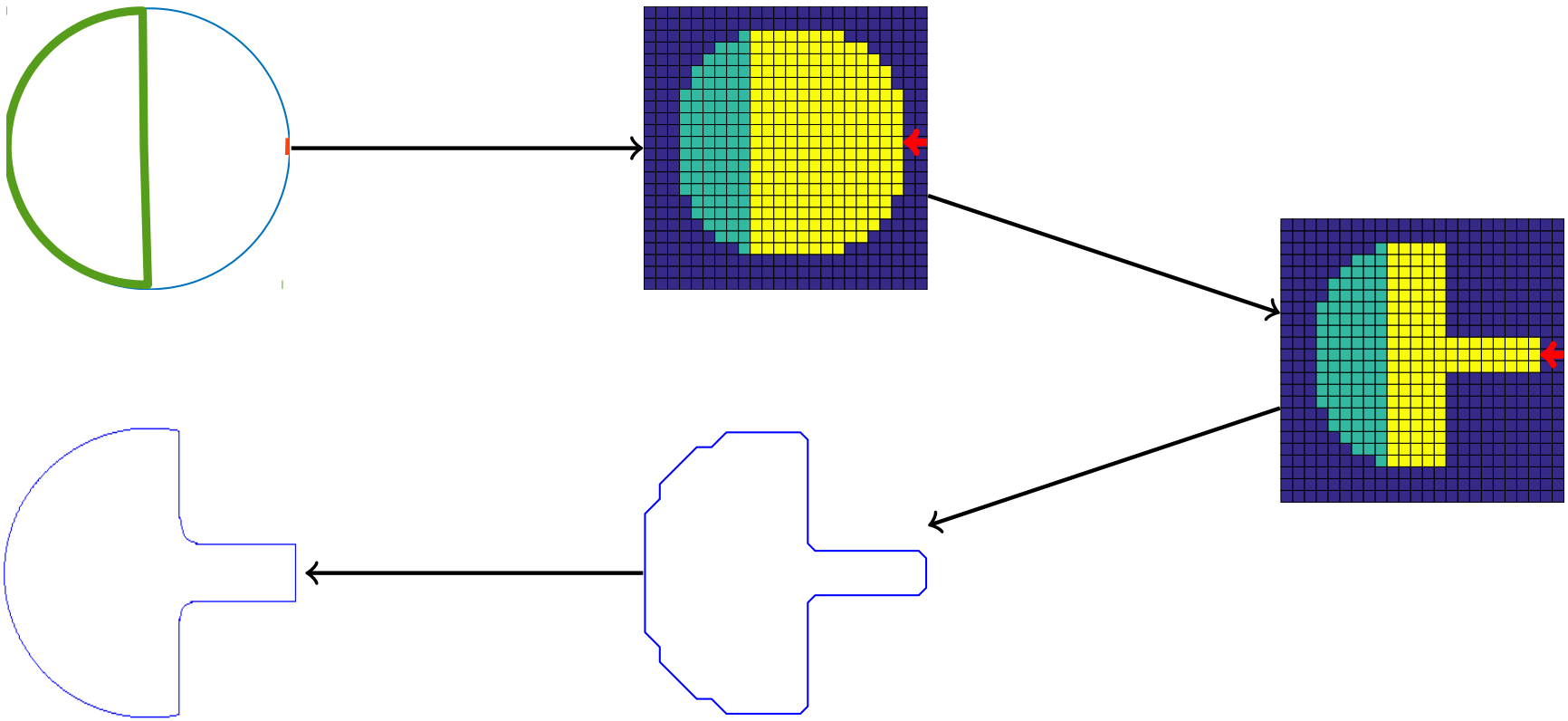
Overview: Workflow

Surface extraction



Overview: Workflow

Fit B-Spline surface



Contents

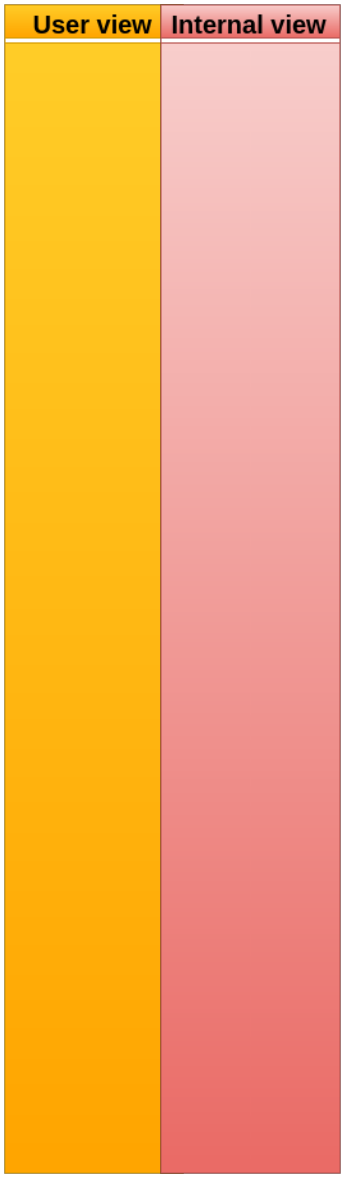
Status

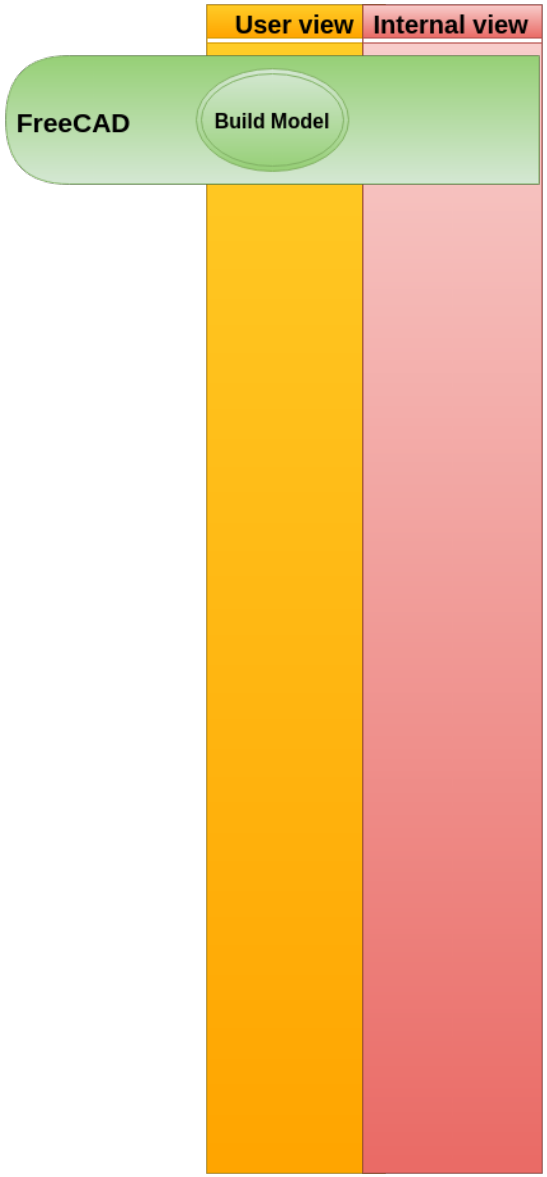
Last milestone

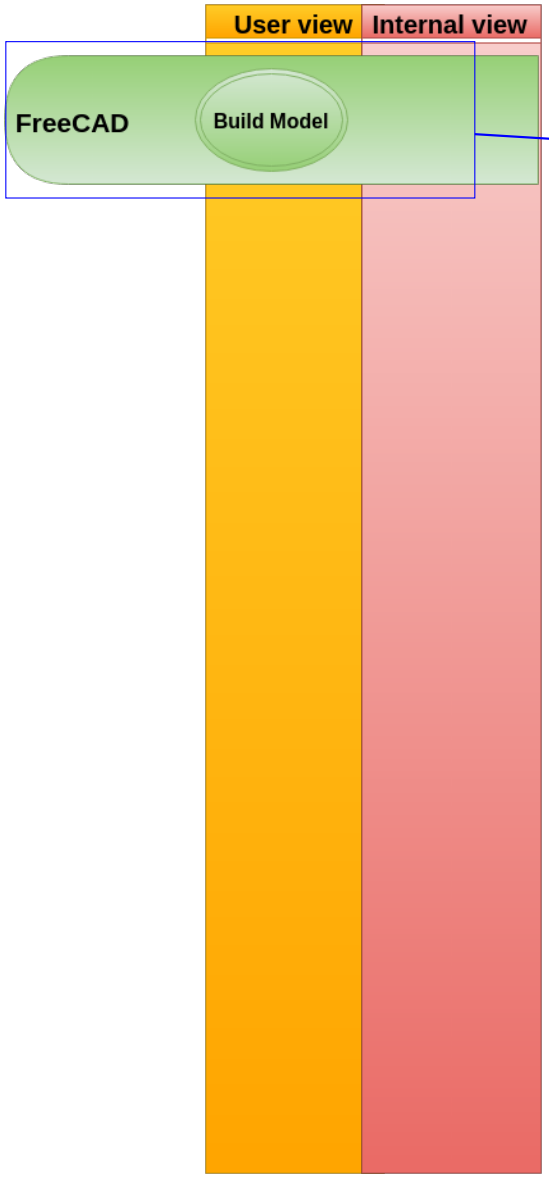
- ✓ Manual voxelization using CVMLCPP
- ✓ ”Hard coded” script for ToPy input
- ✓ Topology optimized geometry using ToPy
- ✗ Recognition of boundary conditions

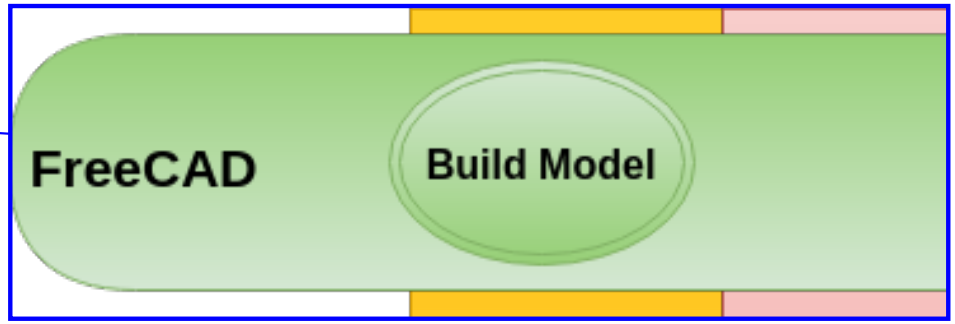
Today

- ✓ Voxelization with OpenCascade
- ✓ Extraction of loads, fixtures and active elements through colouring
- ✓ Automatic ”one click” pipeline to surface reconstruction


User view
Internal view


User view
Internal view
FreeCAD
Build Model



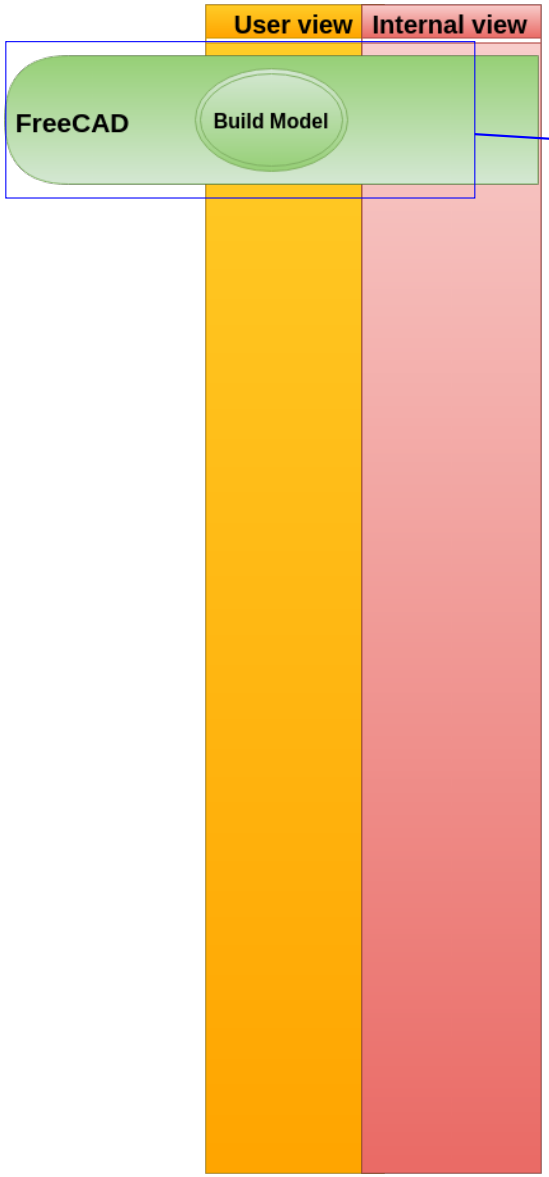


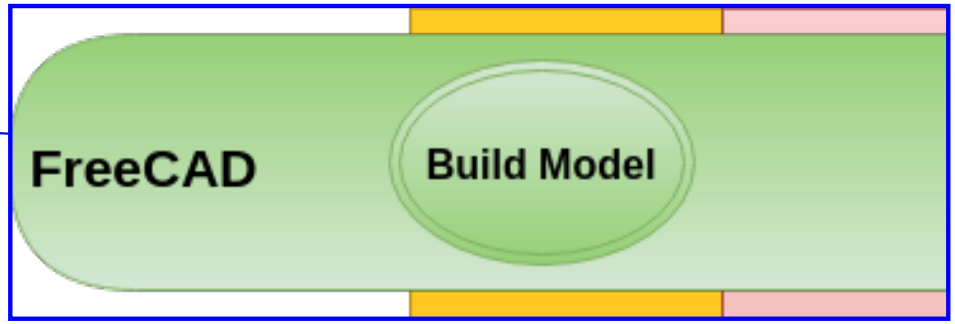
- Model geometry in your favorite CAD tool
- Colour faces for boundary conditions

  Red Fixture

  Green Active

  RGB RGB value in $[0 \leq R < 255, 0 \leq G < 255, 0 \leq B < 255]$ for load vector
- Save model as `STEP with Colours` and `IGES with Colours`





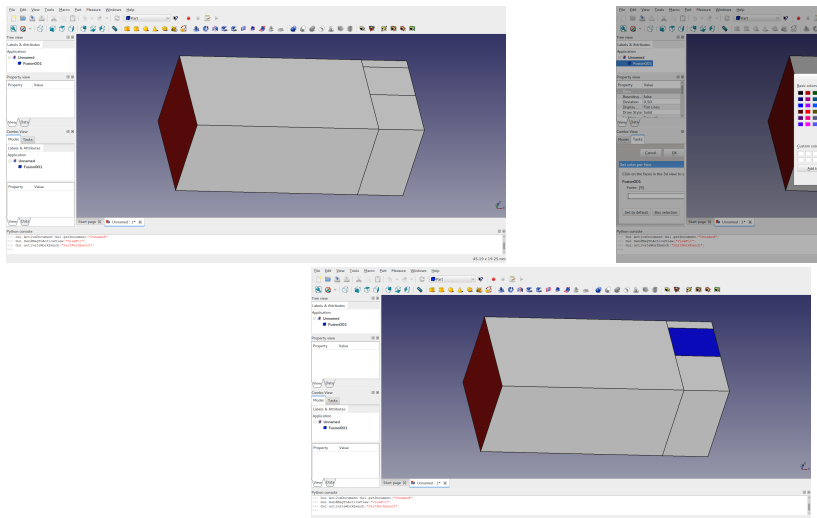
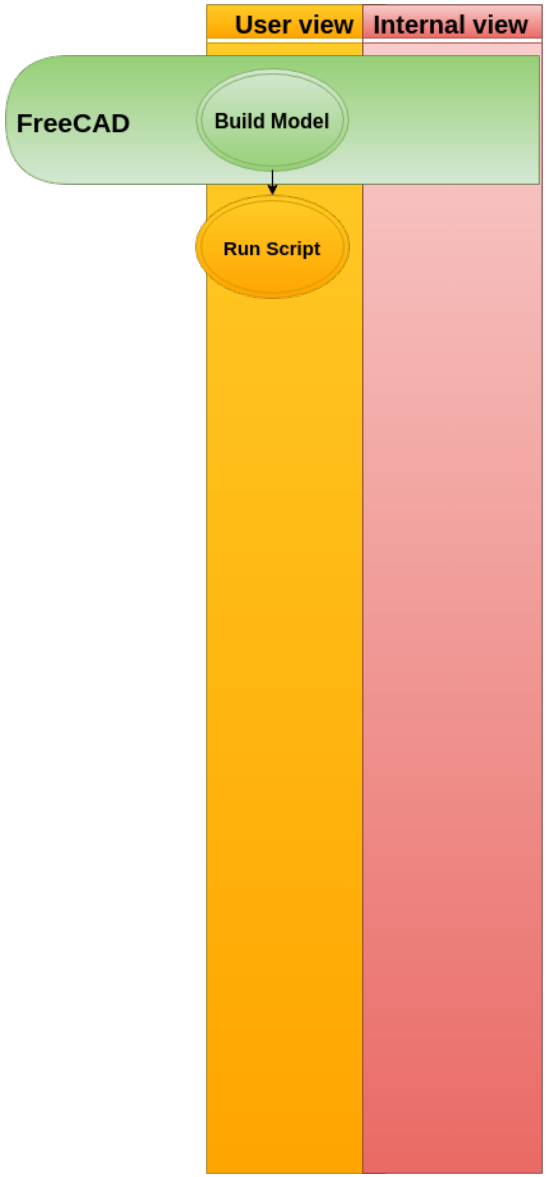
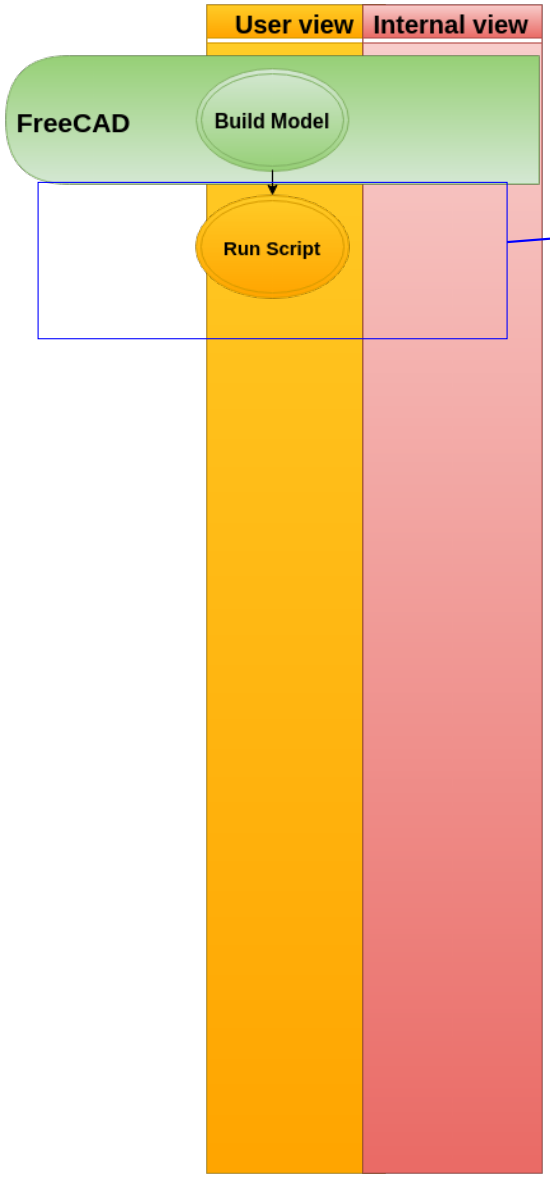


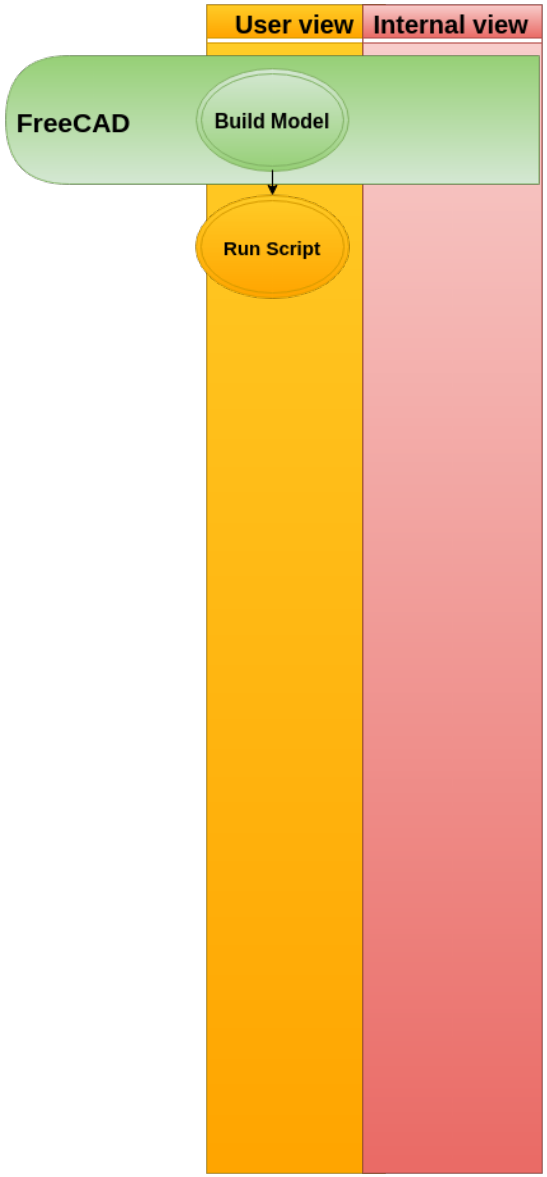
Figure: Color faces in FreeCAD

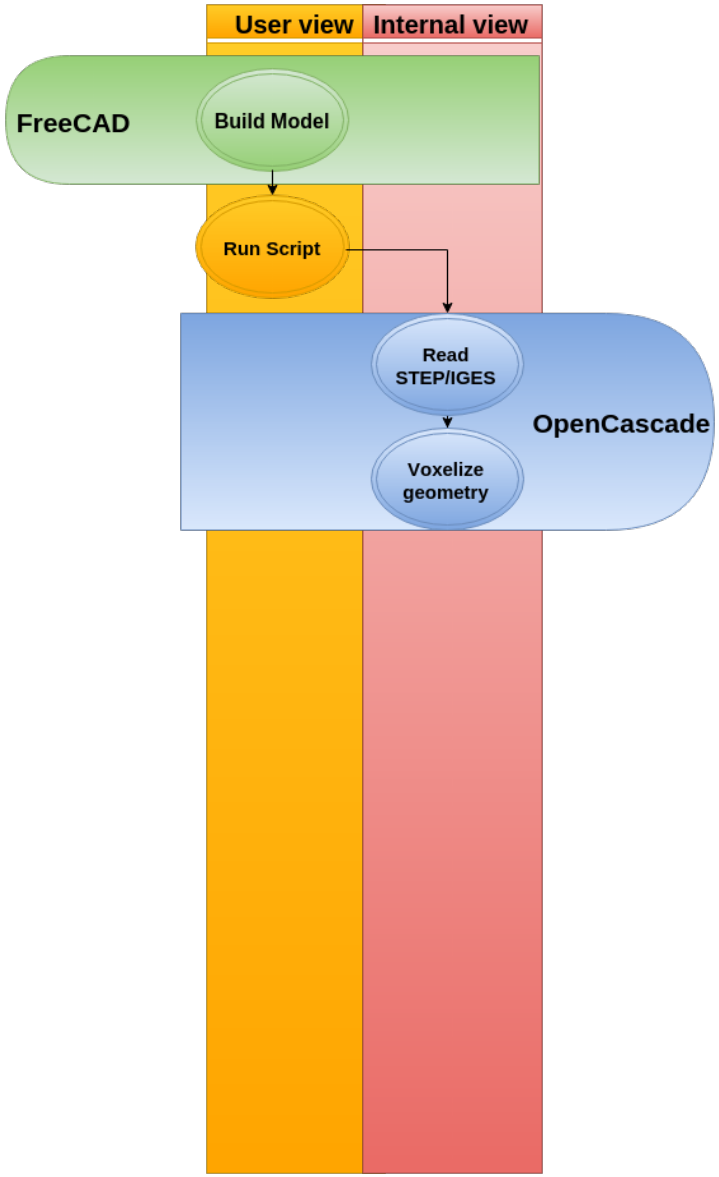

User view
Internal view
FreeCAD
Build Model
Run Script

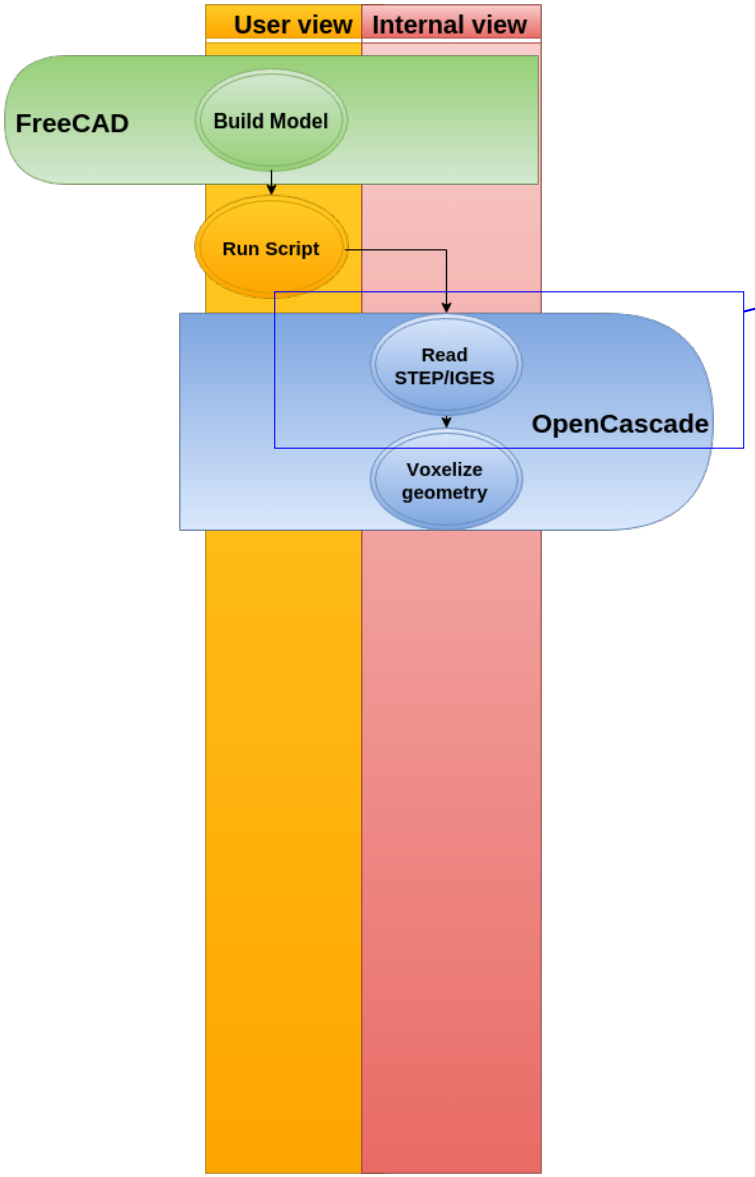


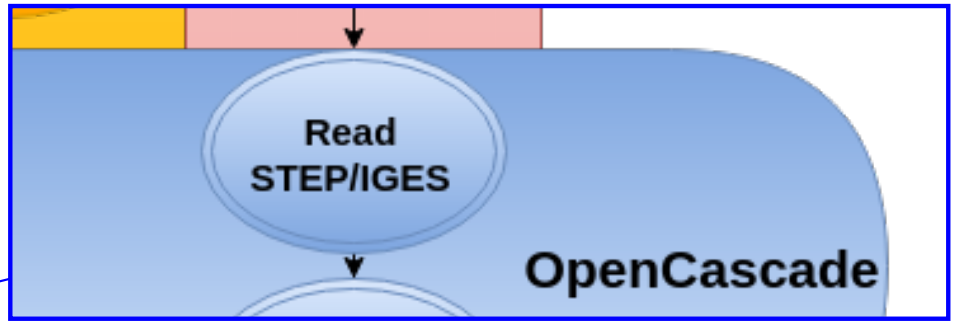
→ Run

```
CADTopOpt filedir filename force_scaling
refinement
```


User view
Internal view
FreeCAD
Build Model
Run Script

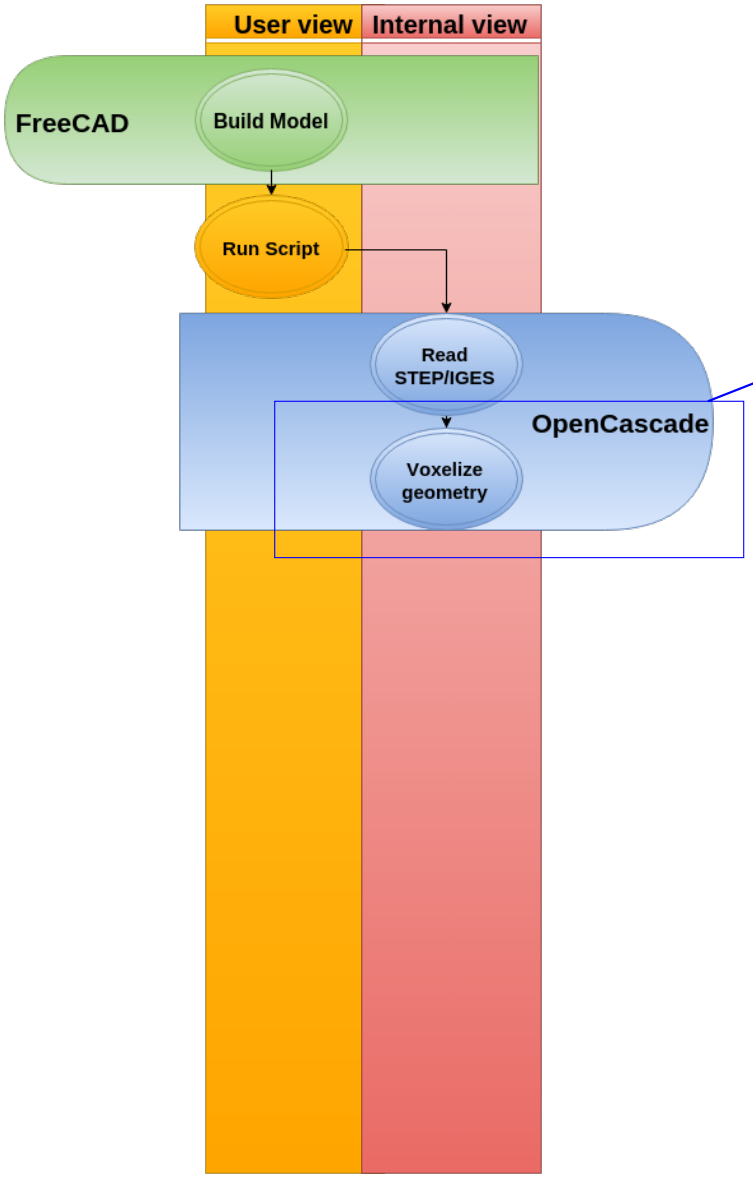

User view
Internal view
FreeCAD
Build Model
Run Script
Read STEP/IGES
Voxelize geometry
OpenCascade

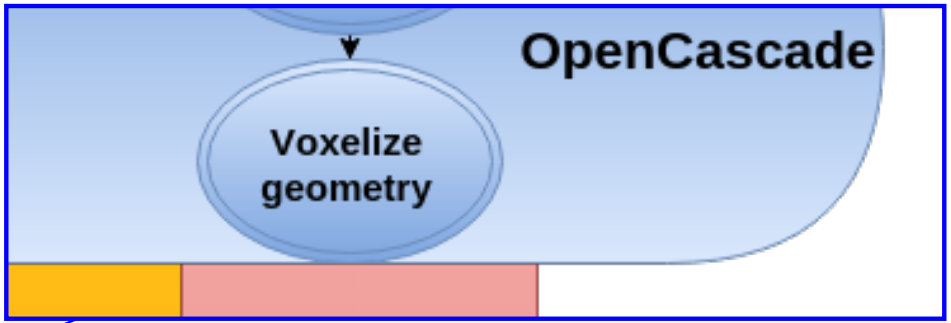




Read STEP and IGES file, extract colours and faces

- STEP holds the colours
- IGES holds the structure





Voxelize faces using OpenCascade

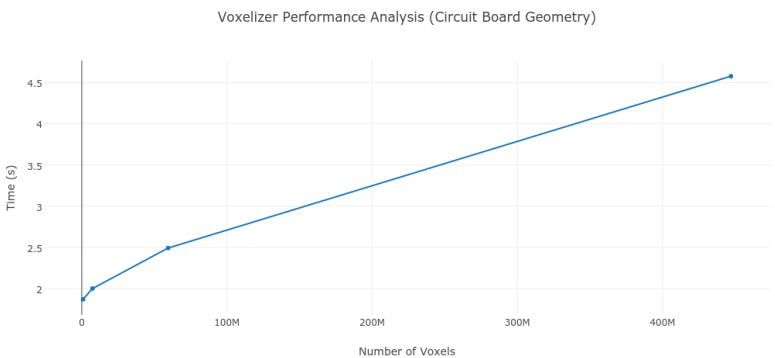
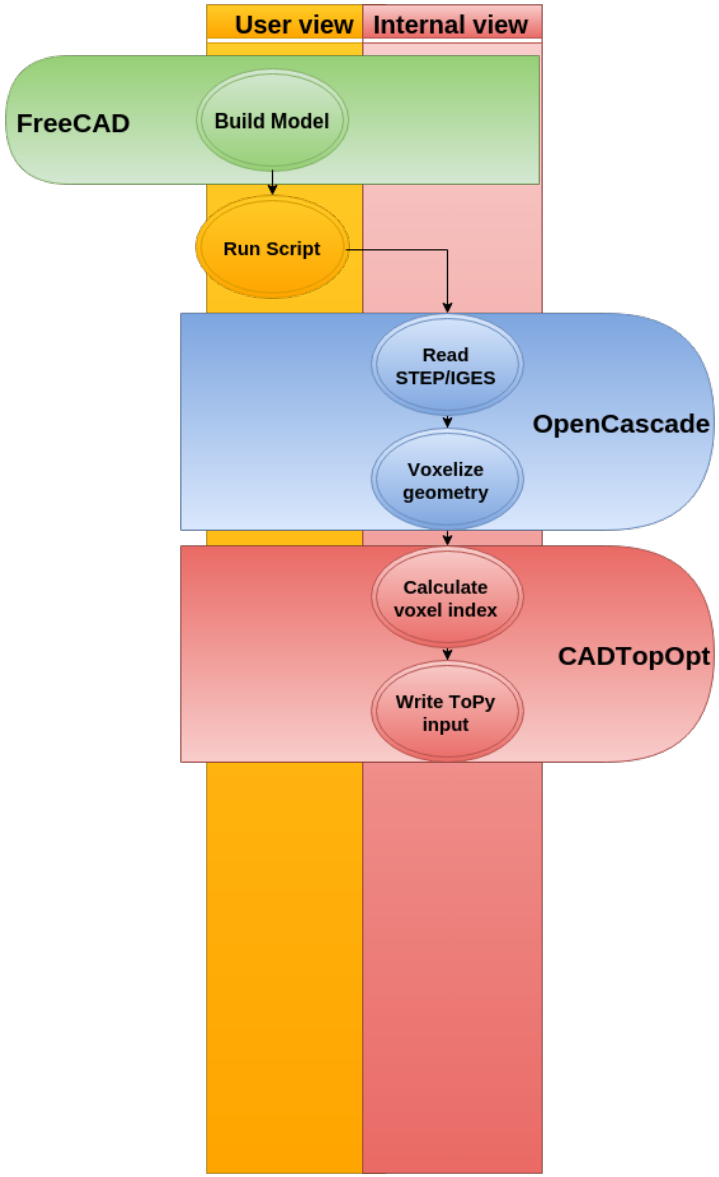
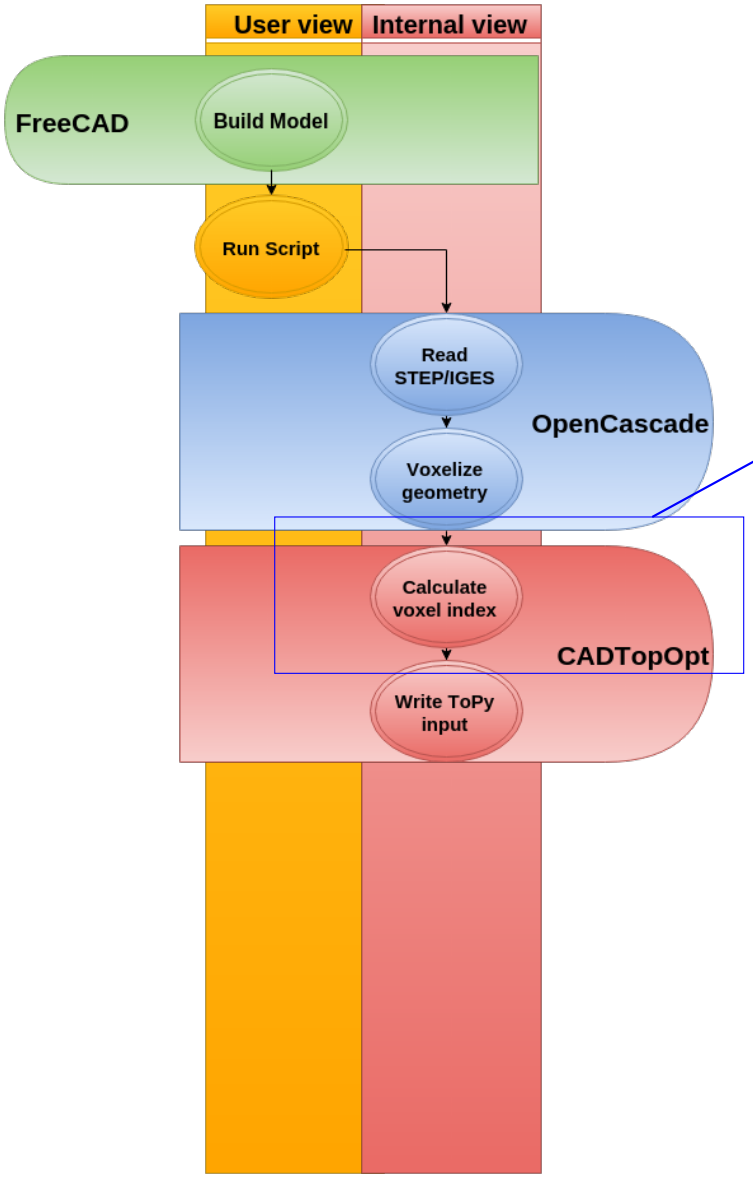
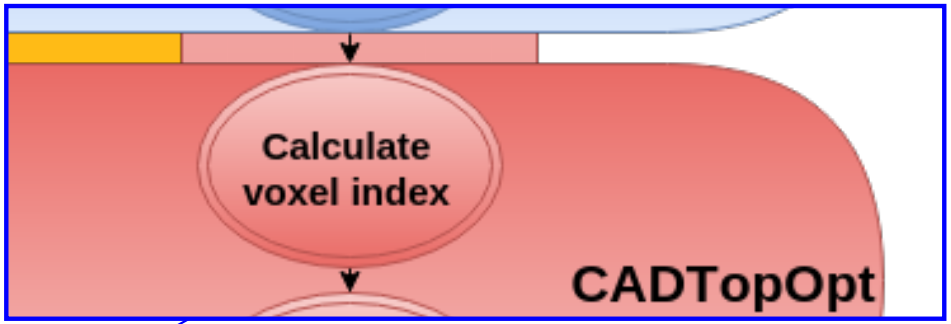


Figure: Scaling of voxelizer


User view
Internal view
FreeCAD
Build Model
Run Script
Read STEP/IGES
OpenCascade
Voxelize geometry
Calculate voxel index
CADTopOpt
Write ToPy input

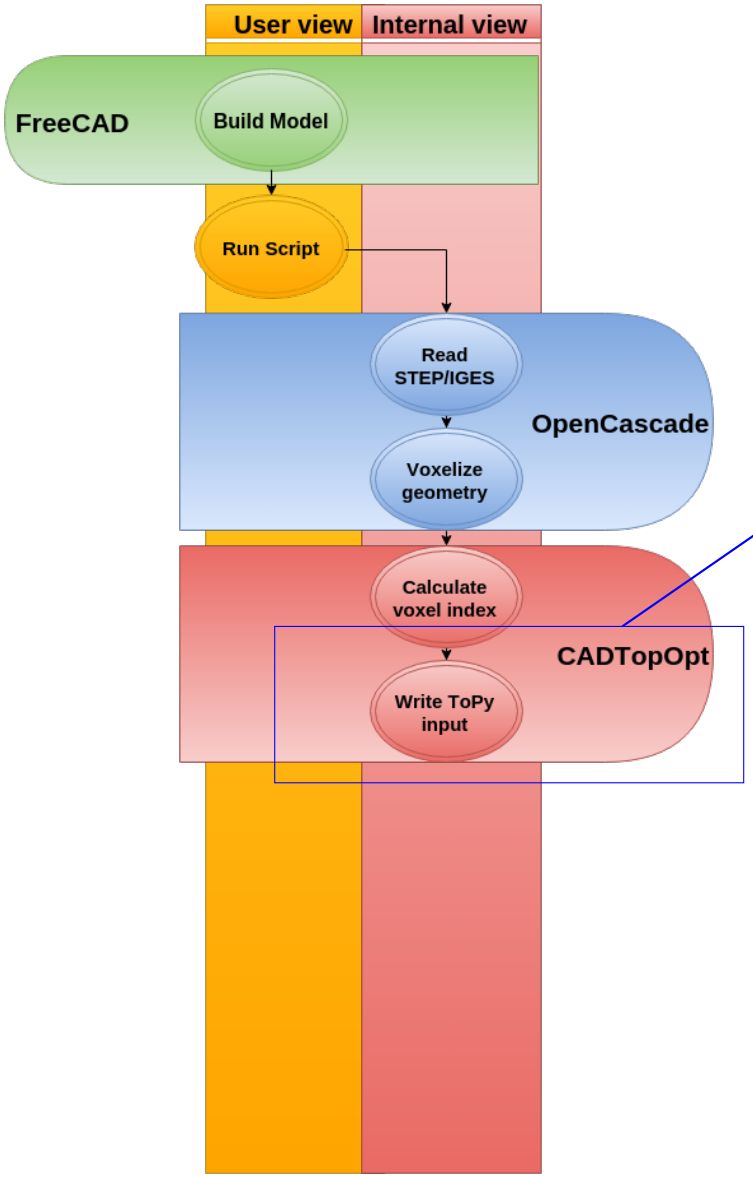


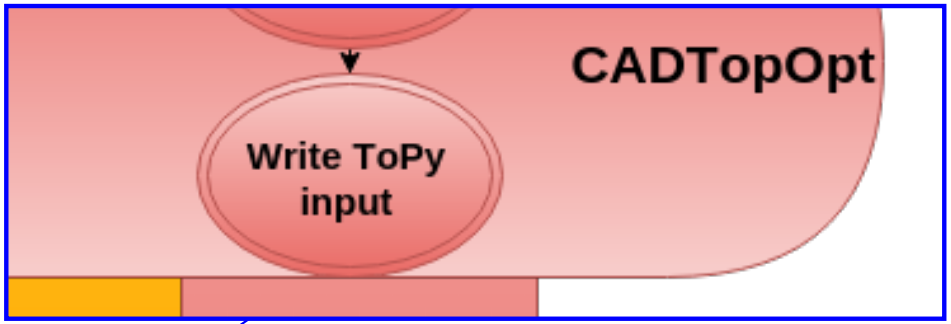


Different indexing for elements and nodes in ToPy

```
# ========================================
# === Discretisation of the design domain ===
# ========================================
# 2D: Y              3D: Y
#     |                  |
#     +---X              +---X
#                       /
#                      Z
#
# 1---5---9
# | 1 | 5 |
# 2---6---10
# | 2 | 6 |
# 3---7---11
# | 3 | 7 |
# 4---8---12
#
```

Figure: Indexing in ToPy [1]

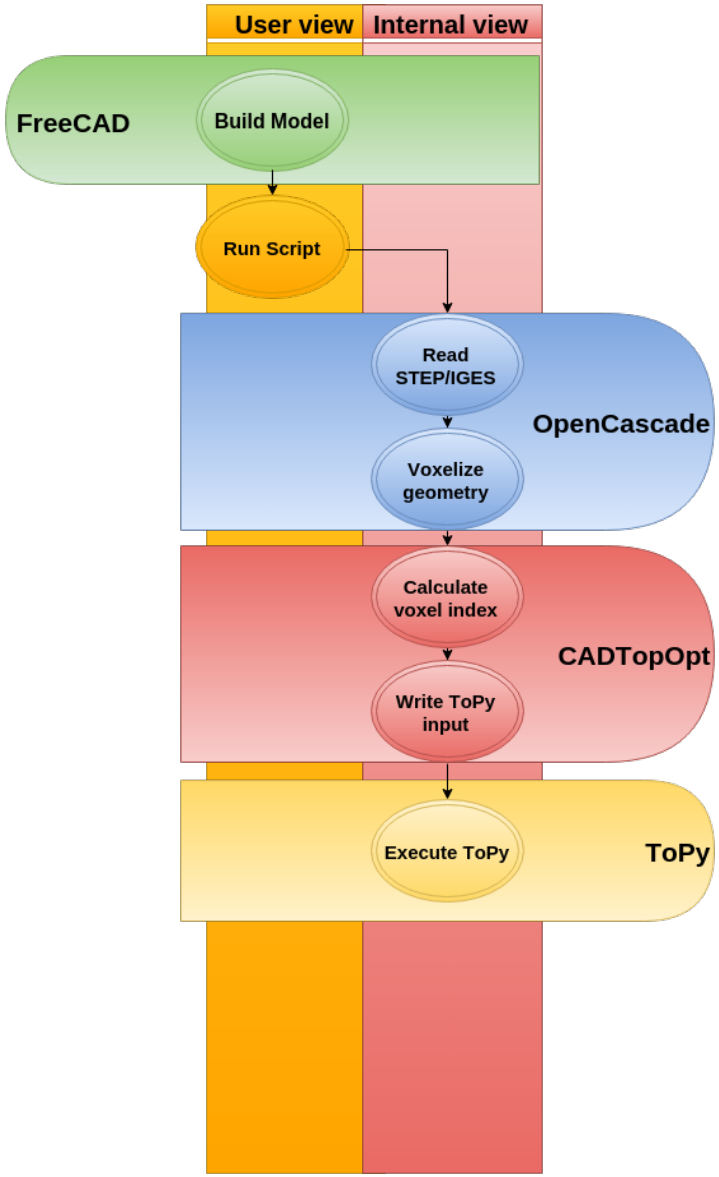


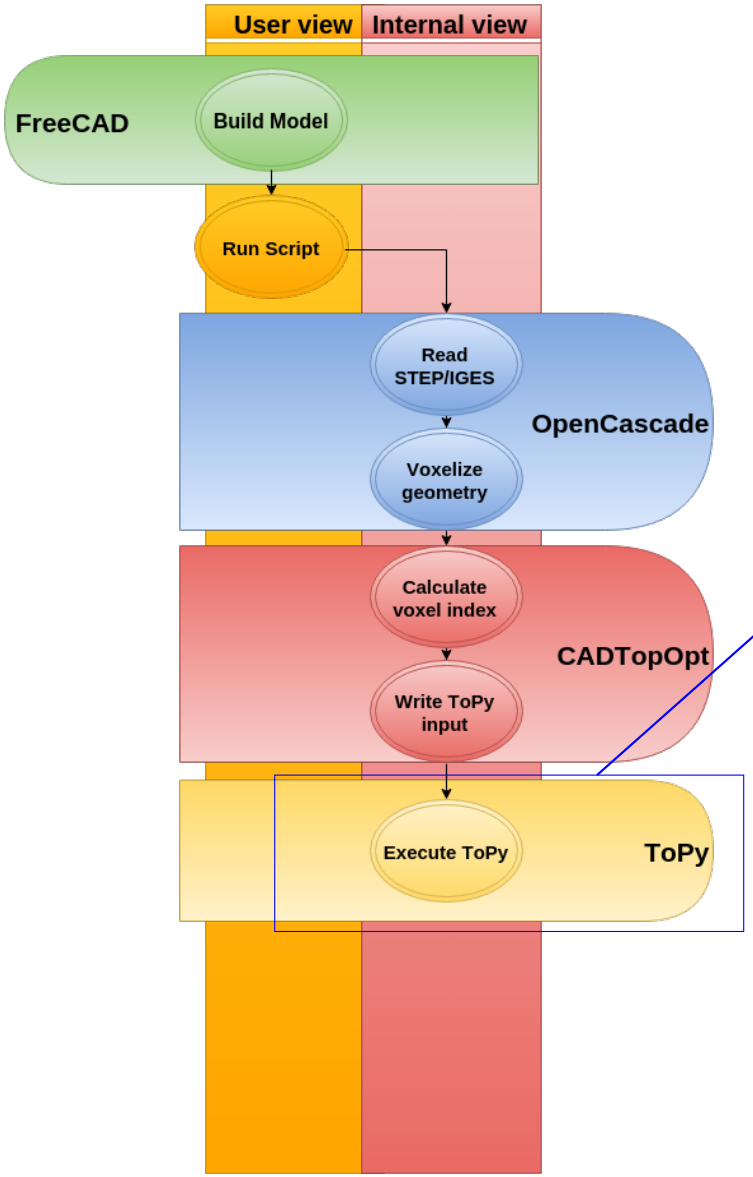


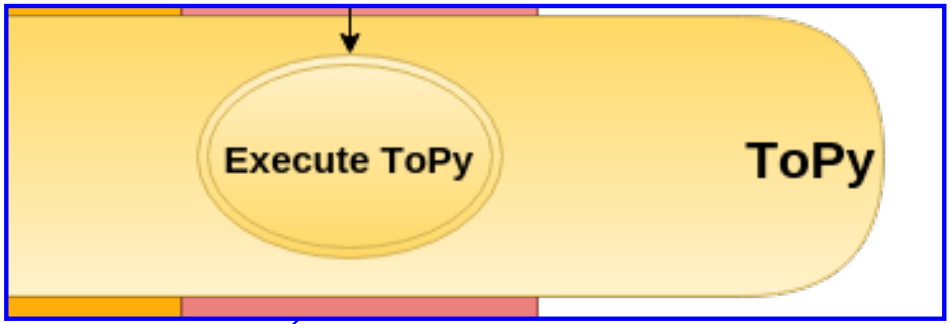
Each voxel index is specifically written

```
[ToPy Problem Definition File v2007]

PROB_TYPE:    comp
PROB_NAME:    topy_CantiLeverWithLoadAtEndSmallerMovedLoad
ETA:          0.4
DOF_PN:       3
VOL_FRAC:     0.15
FILT_RAD:     1.5
ELEM_K:       H8
NUM_ITER:     100
NUM_ELEM_X:   50
NUM_ELEM_Y:   20
NUM_ELEM_Z:   20

# Grey-scale filter (GSF)
P_FAC       : 1
P_HOLD      : 15
P_INCR      : 0.2
P_CON       : 1
P_MAX       : 3

Q_FAC       : 1
Q_HOLD      : 15
Q_INCR      : 0.05
Q_CON       : 1
Q_MAX       : 5
ACTV_ELEM: 18810; 18809; 18808; 18807; 18806; 18805; 18804; 18803; 18830; 18829; 18828; 18827; 18826; 18825; 18824; 18823; 18
18866; 18865; 18864; 18863; 18890; 18889; 18888; 18887; 18886; 18885; 18884; 18883; 18910; 18909; 18908; 18907; 18906; 18905;
18948; 18947; 18946; 18945; 18944; 18943; 18970; 18969; 18968; 18967; 18966; 18965; 18964; 18963; 18990; 18989; 18988; 18987;
19830; 19829; 19828; 19827; 19826; 19825; 19824; 19823; 19850; 19849; 19848; 19847; 19846; 19845; 19844; 19843; 19870; 19869;
19884; 19883; 19910; 19909; 19908; 19907; 19906; 19905; 19904; 19903; 19930; 19929; 19928; 19927; 19926; 19925; 19924; 19923;
19966; 19965; 19964; 19963; 19990; 19989; 19988; 19987; 19986; 19985; 19984; 19983
PASV_ELEM:
FXTR_NODE_X: 21; 20; 19; 18; 17; 16; 15; 14; 13; 12; 11; 10; 9; 8; 7; 6; 5; 4; 3; 2; 42; 41; 40; 39; 38; 37; 36; 35; 34; 33;
```

Figure: Script for ToPy


User view
Internal view
FreeCAD
Build Model
Run Script
Read STEP/IGES
OpenCascade
Voxelize geometry
Calculate voxel index
CADTopOpt
Write ToPy input
Execute ToPy
ToPy





Execute ToPy on the input file

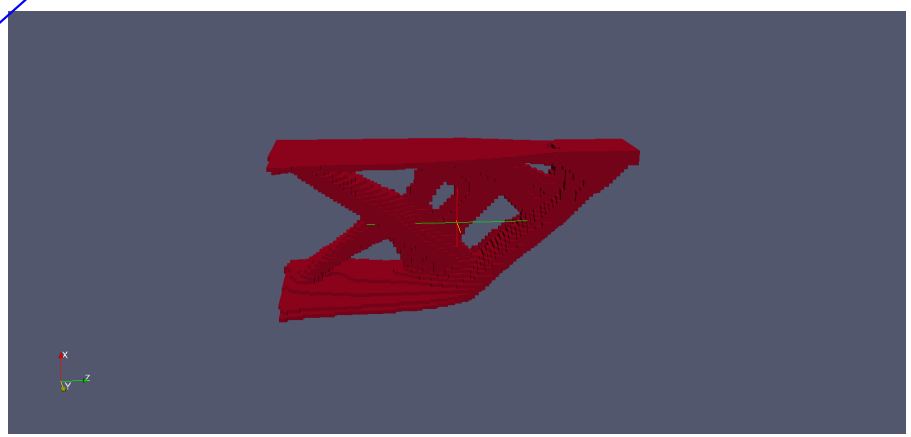
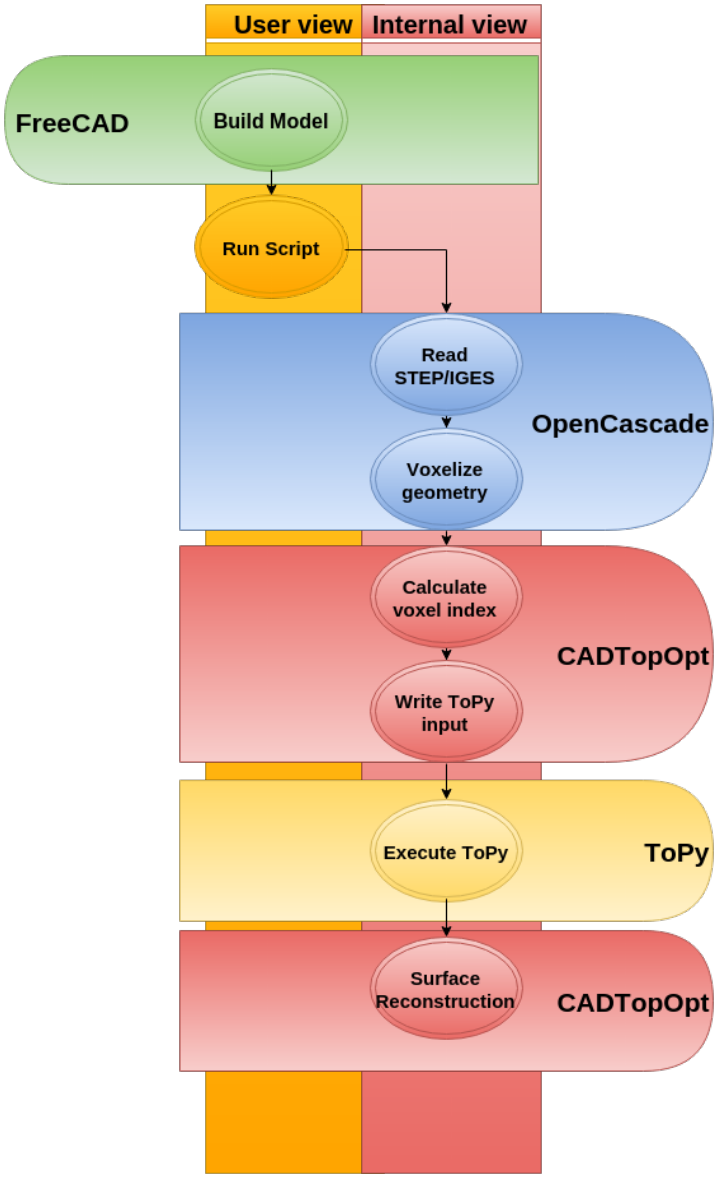
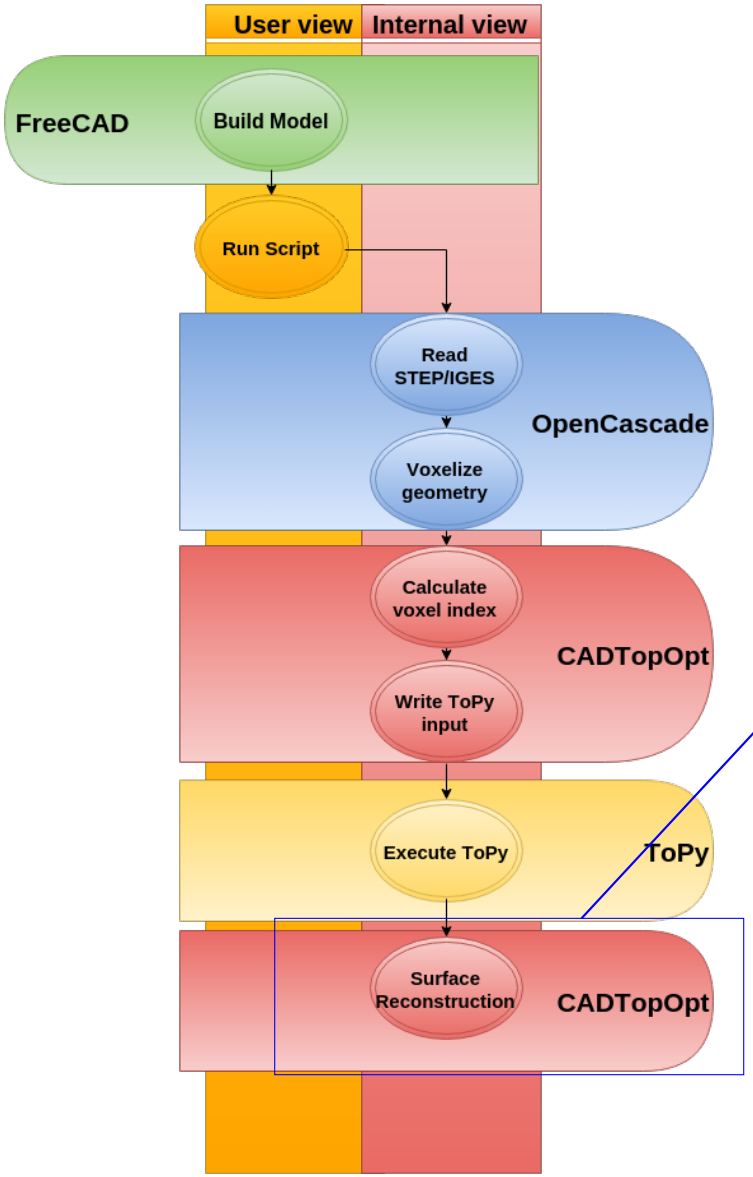
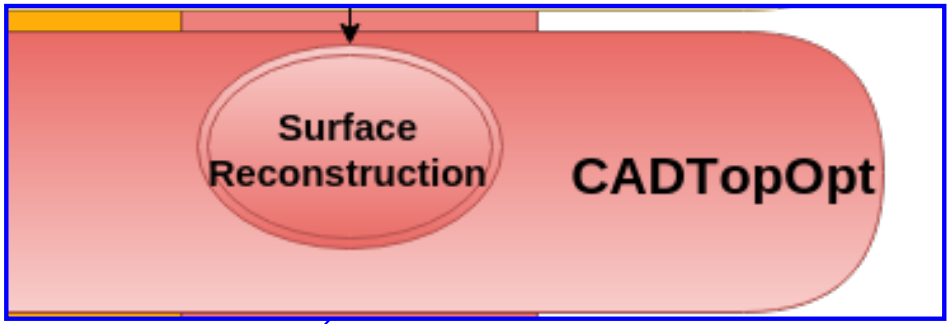


Figure: ToPy Output


User view
Internal view
FreeCAD
Build Model
Run Script
Read STEP/IGES
Voxelize geometry
OpenCascade
Calculate voxel index
Write ToPy input
CADTopOpt
Execute ToPy
ToPy
Surface Reconstruction
CADTopOpt





Run dual contouring algorithm

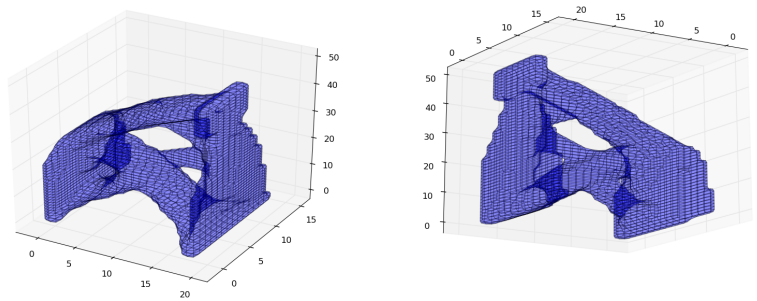
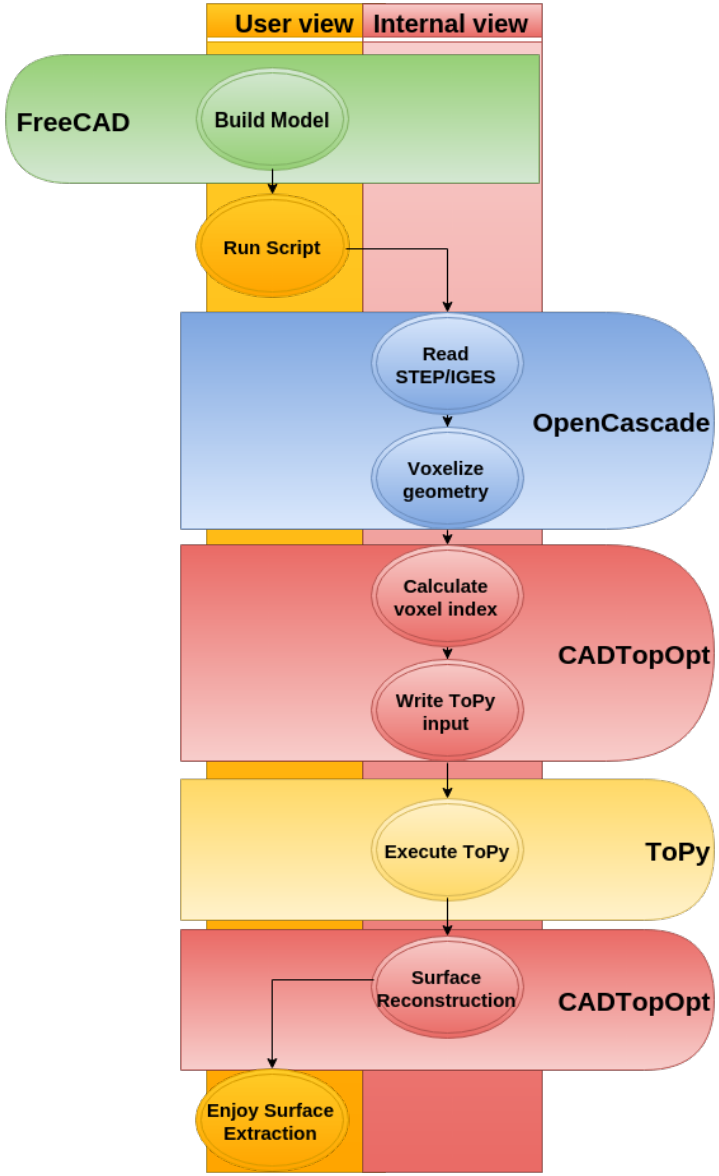
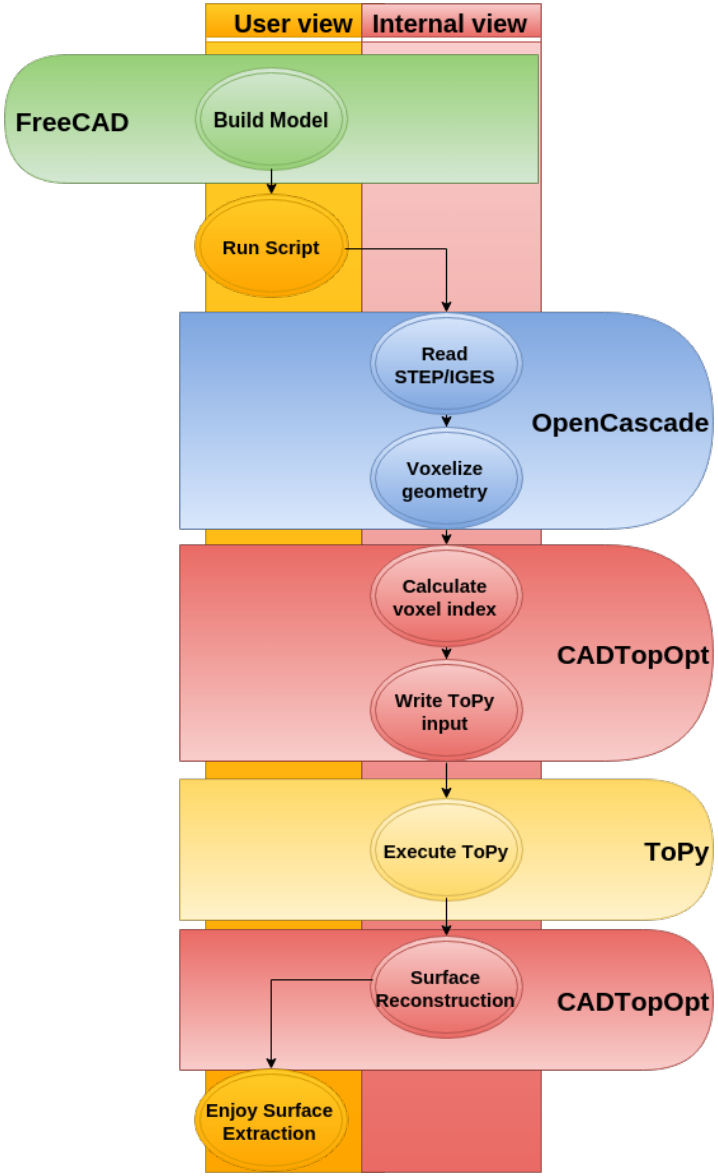
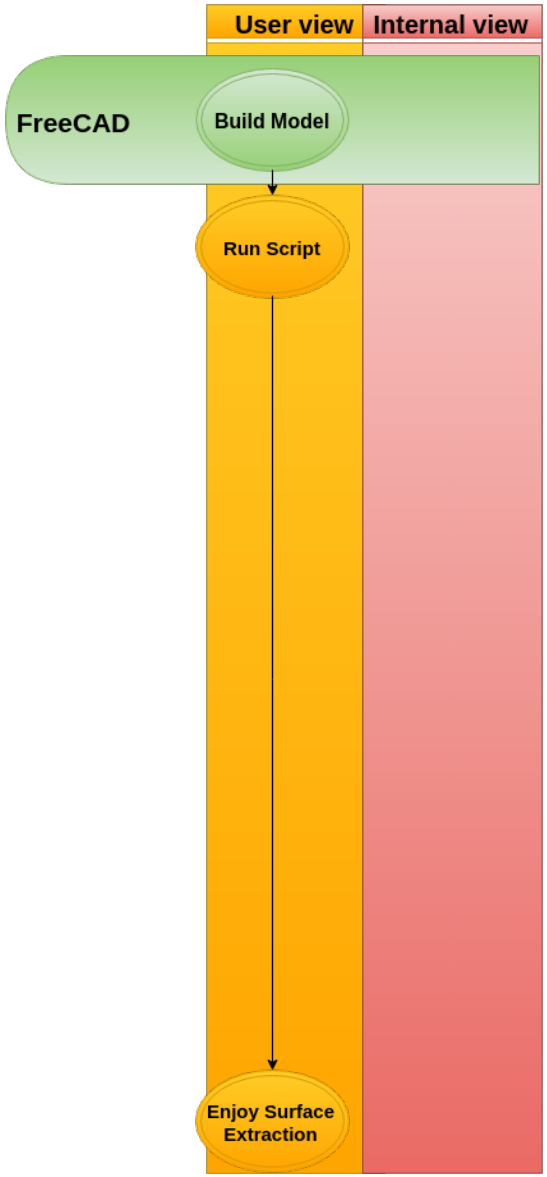


Figure: Surface extraction for Cantilever


User view
Internal view
FreeCAD
Build Model
Run Script
Read STEP/IGES
Voxelize geometry
OpenCascade
Calculate voxel index
Write ToPy input
CADTopOpt
Execute ToPy
ToPy
Surface Reconstruction
CADTopOpt
Enjoy Surface Extraction



But what does the user see?



But what does the user see?

Contents

Status

Last milestone

- 🕐 Surface reconstruction with the VTK Toolbox

Today

- ✓ Extraction of voxel data from Topy
- ✓ 3D Dual Contouring implementation
- ✓ Coarsening and non-manifold edge treatment
- ✓ Projection of datapoints onto quads and respective parametrization
- 🕐 Interface to NURBS

From Voxel to Mesh Geometry

- Extract isosurface from voxel information
- Algorithms: Marching Cubes, Dual Contouring, Extended Models
- Problems with VTK's Marching Cube implementation

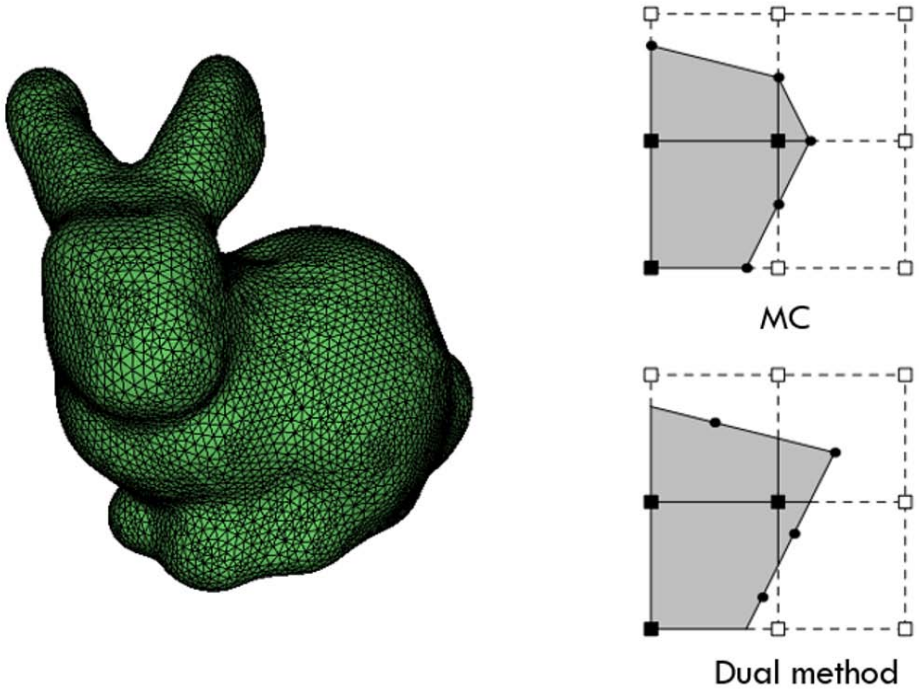
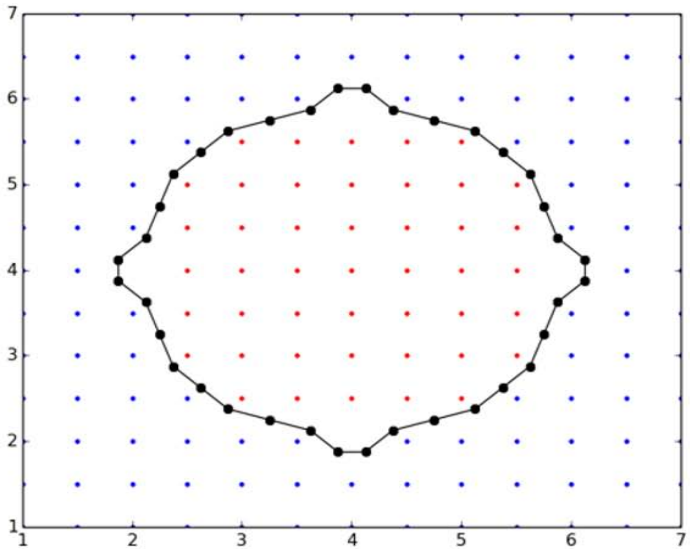
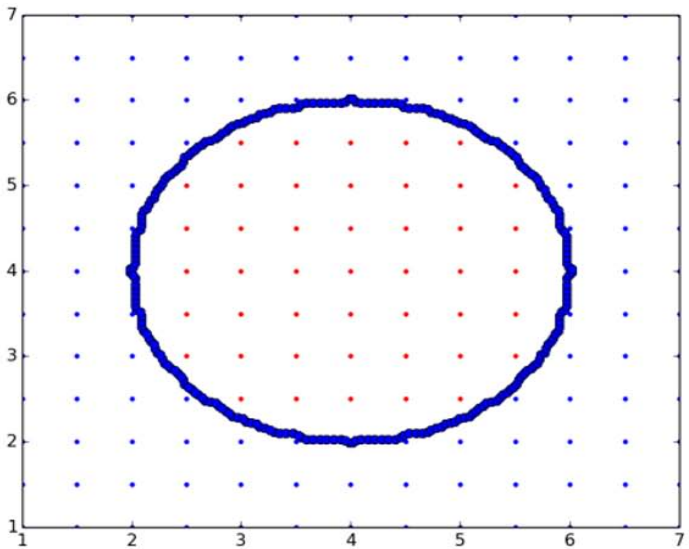


Figure: From [4],[5]

Dual Contouring

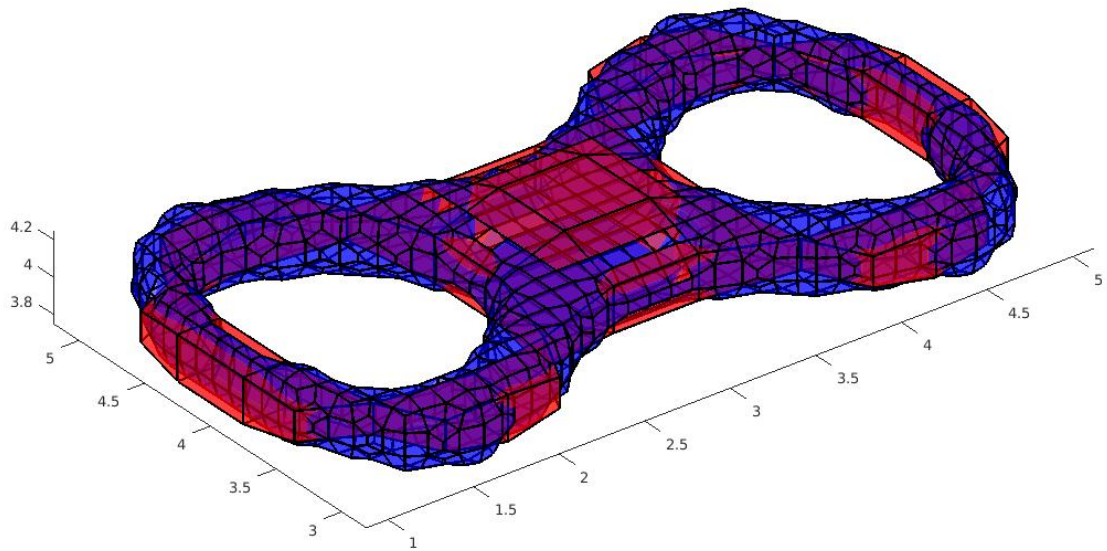
- Python implementation — Use of powerful libraries, including VTK
- Output: Closed surface made out of *quads*
- Coarsening is needed for surface fitting algorithms





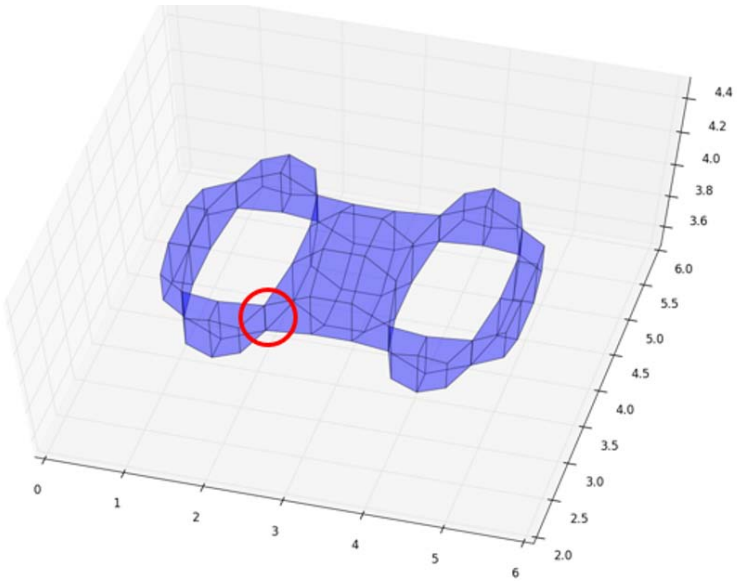
Dual Contouring

- Python implementation — Use of powerful libraries, including VTK
- Output: Closed surface made out of *quads*
- Coarsening is needed for surface fitting algorithms



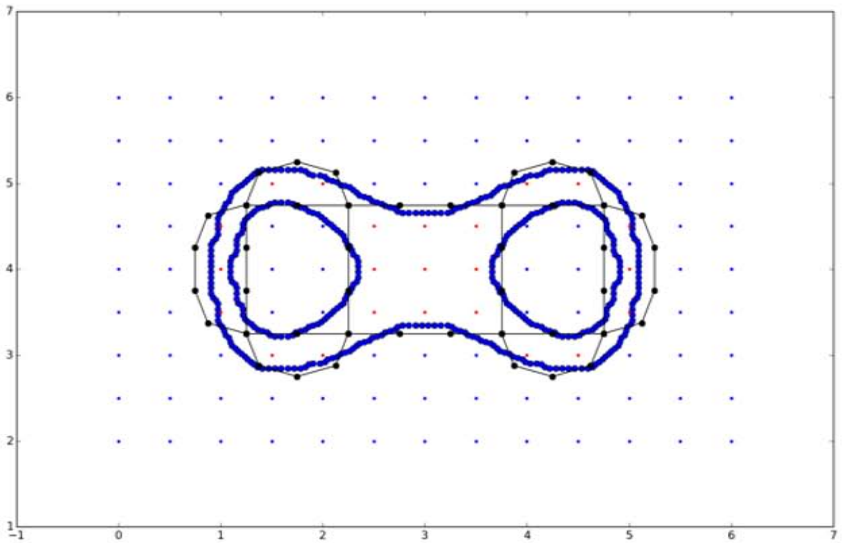
Dual Contouring — Problems

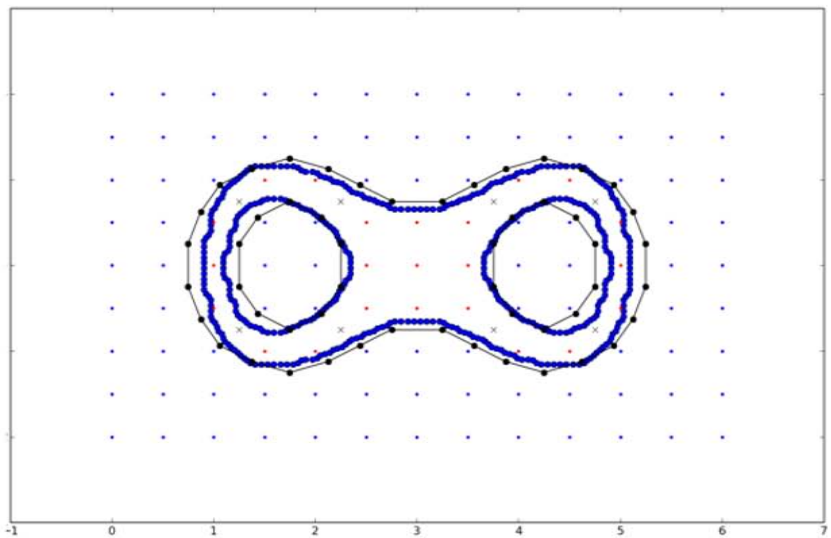
- **Non–manifold edges** appear
- One edge can only belong to two quads for the surface to be closed
- Special treatments in the implementation to avoid them



Dual Contouring — Problems

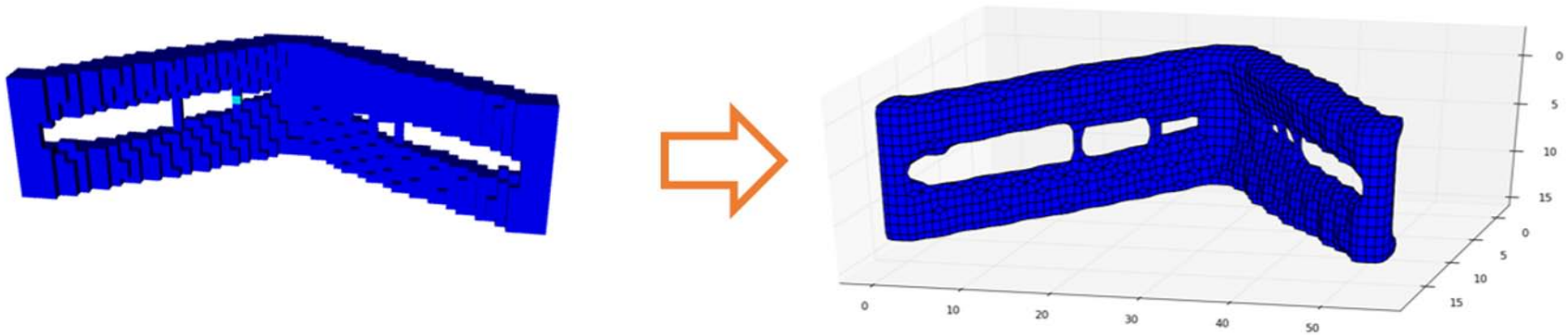
- **Non–manifold edges** appear
- One edge can only belong to two quads for the surface to be closed
- Special treatments in the implementation to avoid them





Dual Contouring — Input

- Interface between Topology Optimization and Surface Extraction
- Special implementation to use voxel data from ToPy as input



Demo

Projection and Parametrization

- Points from finer grid are projected to quads of the coarser grid
- Parameters *u* and *v* are found for each quad
- This information is needed for the algorithms in the last part of the pipeline