

Technische Universität München

## BGCE Project: CAD – Integrated Topology Optimization

BGCE Second Milestone Meeting

S. Joshi, J.C. Medina, F. Menhorn,  
S. Reiz, B. R  th, E. Wannerberg, A. Yurova

November 3, 2015



# Contents

## 1. Introduction

- 1.1 Motivation
- 1.2 Workflow Overview
- 1.3 Organization

## 2. Topology optimization

- 2.1 User experience and internal structure
- 2.2 The next steps MOVE TO LATER

## 3. Surface Extraction

- 3.1 Dual Contouring
- 3.2 Projection and Parametrization

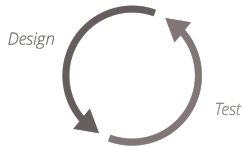
## 4. B-Spline Fitting

- 4.1 Peters' scheme
- 4.2 Fitting pipeline

## 5. Summary & Outlook

# Motivation

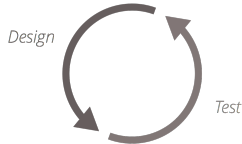
Current Design Process:



- Iterative and redundant
- Time consuming

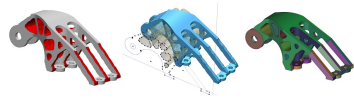
# Motivation

## Current Design Process:



- Iterative and redundant
- Time consuming

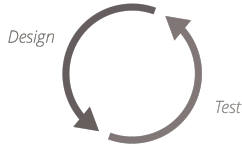
## Topology optimization



- Promoted by additive manufacturing

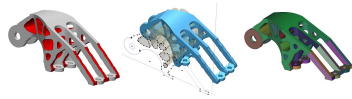
# Motivation

## Current Design Process:



- Iterative and redundant
- Time consuming

## Topology optimization



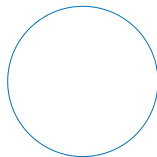
- Promoted by additive manufacturing

## Focus:

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

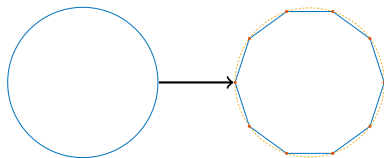
# Workflow Overview

CAD design



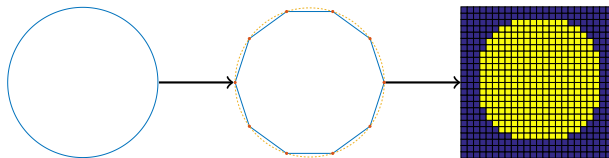
# Workflow Overview

## STL interface



# Workflow Overview

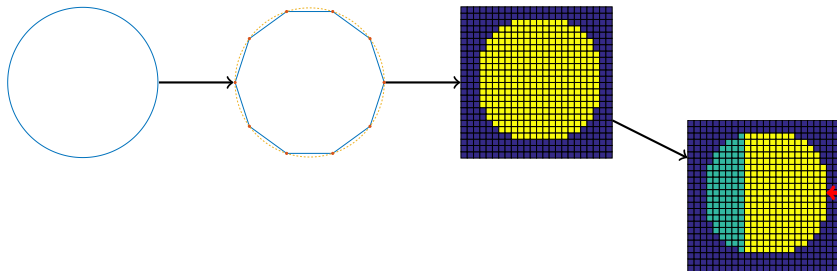
Voxelized topology





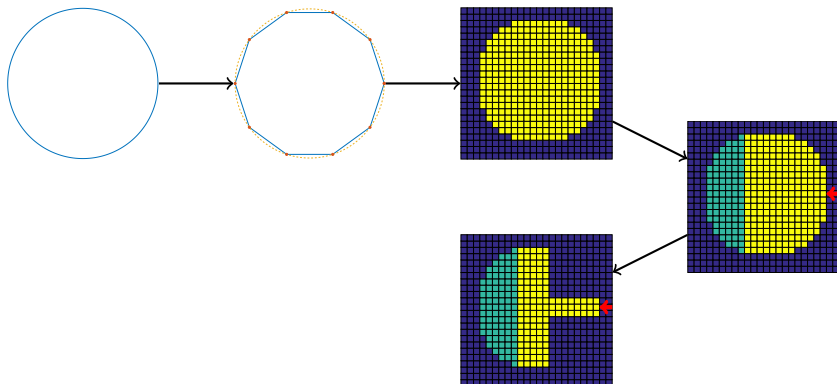
# Workflow Overview

Specification of loads and fixtures



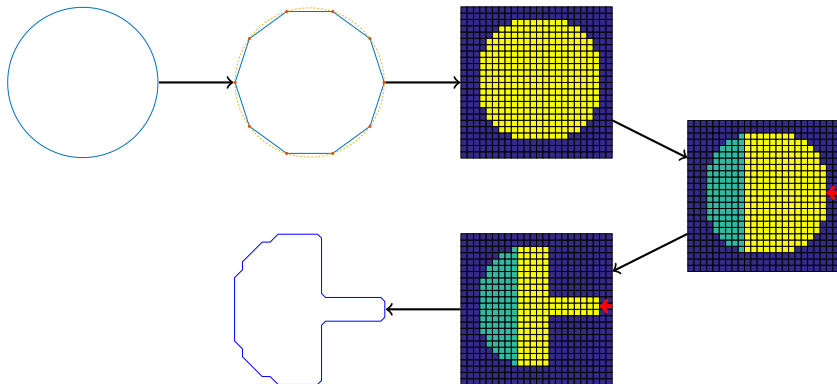
# Workflow Overview

Optimized topology



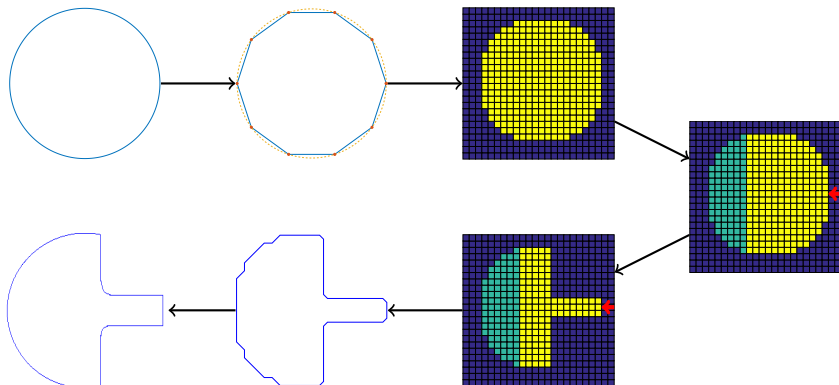
# Workflow Overview

## Surface extraction



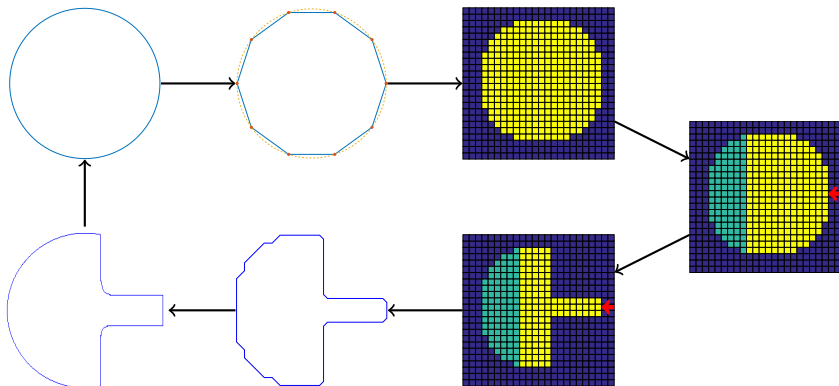
# Workflow Overview

## Parametrized CAD-geometries



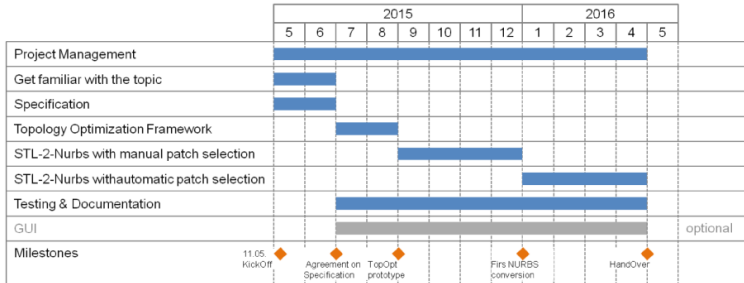
# Workflow Overview

Iterative design process



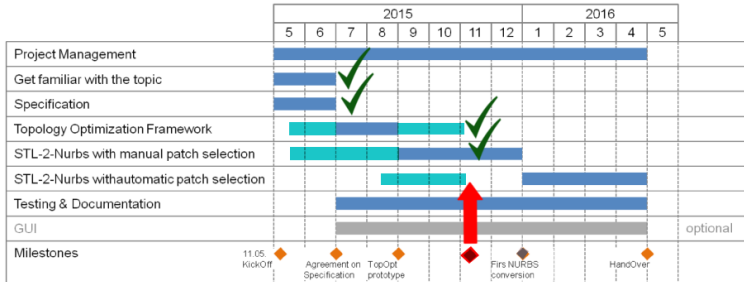
# Schedule & Milestones

## Schedule:



# Schedule & Milestones

## Schedule: (current)



# Divide and Conquer

**Project Manager**



Benjamin Rüth



Erik Wannerberg

**Team Leader**

Friedrich Menhorn Saumitra Joshi Severin Reiz Juan Carlos Medina Erik Wannerberg

**C++ Implementation**

Benjamin Rüth Anna Yurova

**Surface Fitting**

Friedrich Menhorn Saumitra Joshi Severin Reiz

**Topology Optimization**

Benjamin Rüth Juan Carlos Medina

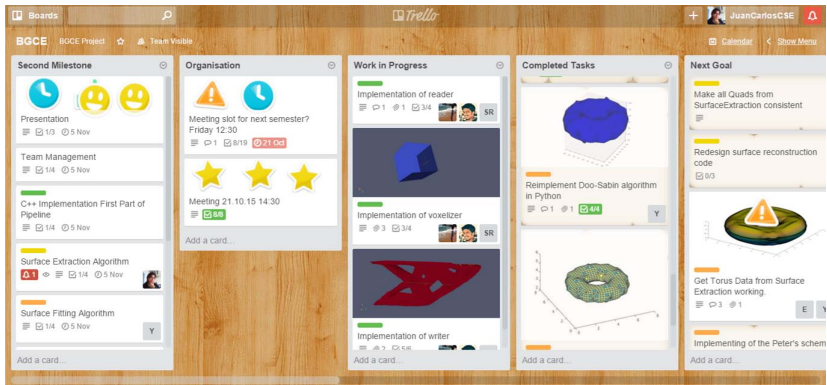
**Surface Extraction**

Erik Wannerberg Anna Yurova

**Surface Fitting**



# Project management



# Contents

## 1. Introduction

- 1.1 Motivation
- 1.2 Workflow Overview
- 1.3 Organization

## 2. Topology optimization

- 2.1 User experience and internal structure
- 2.2 The next steps MOVE TO LATER

## 3. Surface Extraction

- 3.1 Dual Contouring
- 3.2 Projection and Parametrization

## 4. B-Spline Fitting

- 4.1 Peters' scheme
- 4.2 Fitting pipeline

## 5. Summary & Outlook

# Status DRAFT

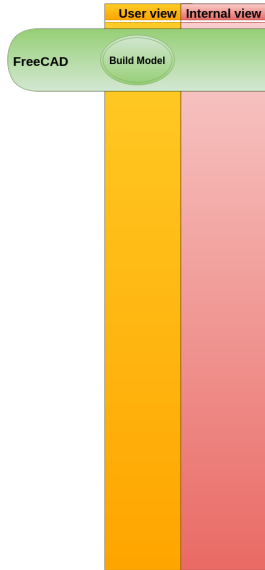
## Last milestone

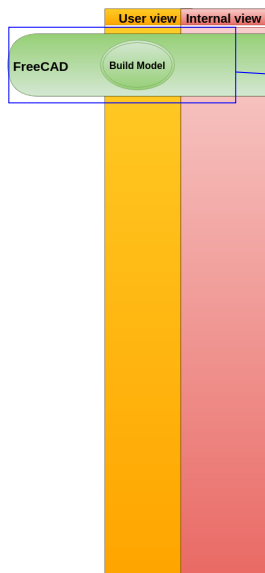
- ✓ Manual voxelization using CVMLCPP
- ✓ "Hard coded" script for ToPy input
- ✓ Topology optimized geometry using ToPy
- ✗ Recognition of boundary conditions

## Today

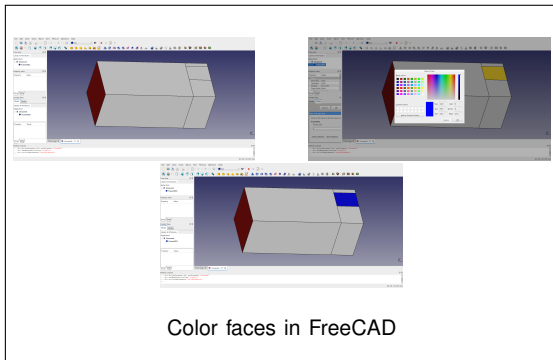
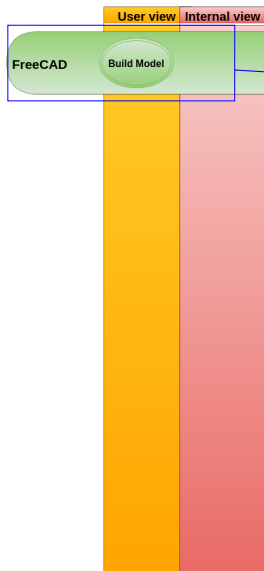
- ✓ Voxelization with OpenCascade
- ✓ Extraction of loads, fixtures and active elements through colouring
- ✓ Automatic "one click" pipeline to surface reconstruction

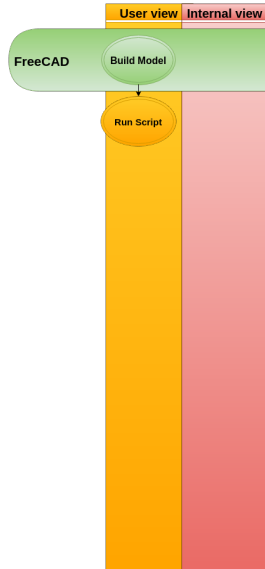
User view	Internal view



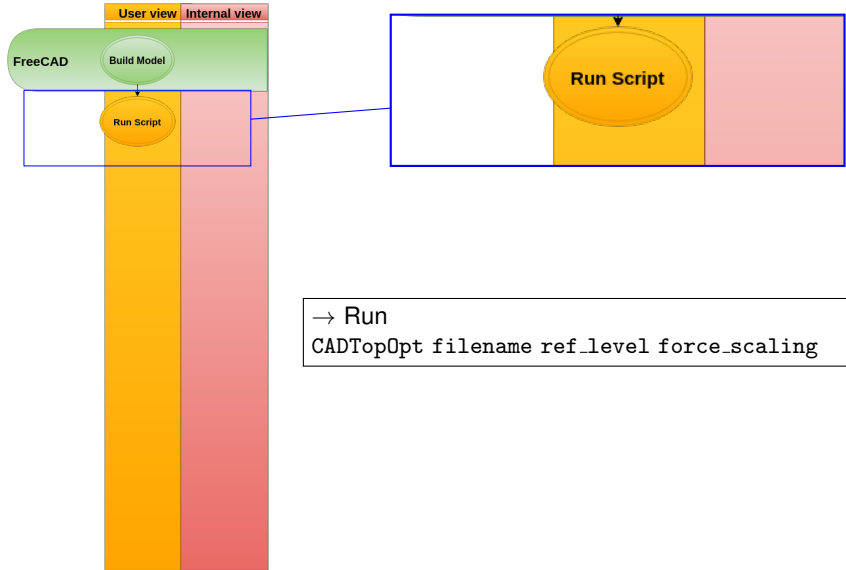


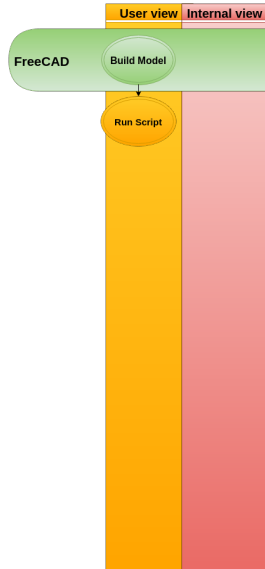
- Model geometry in favorite CAD tool
- Colour faces for boundary conditions
  - Red** Fixture
  - Green** Active
  - RGB** RGB value in  $[0 \leq R < 255, 0 \leq G < 255, 0 \leq B < 255]$  for load vector
- Save model as STEP with Colours and IGES with Colours

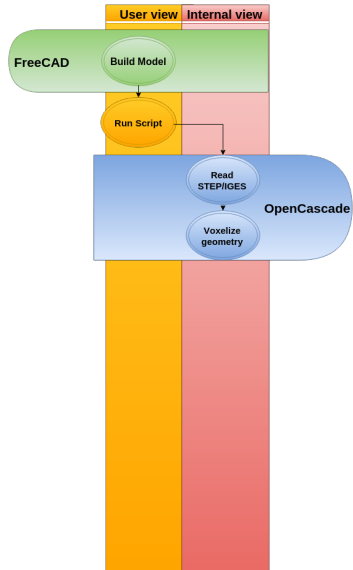


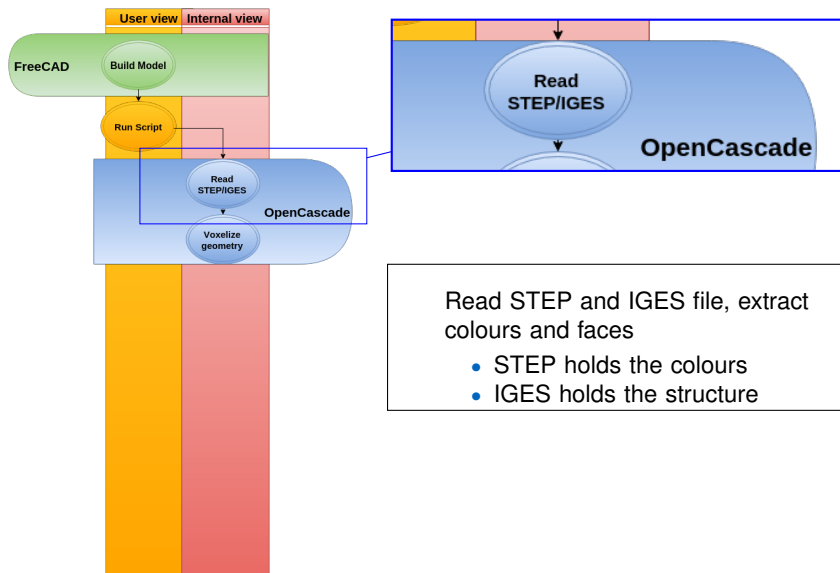


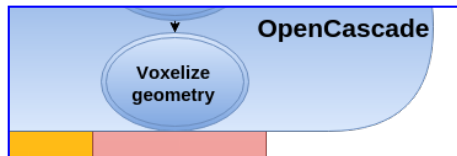
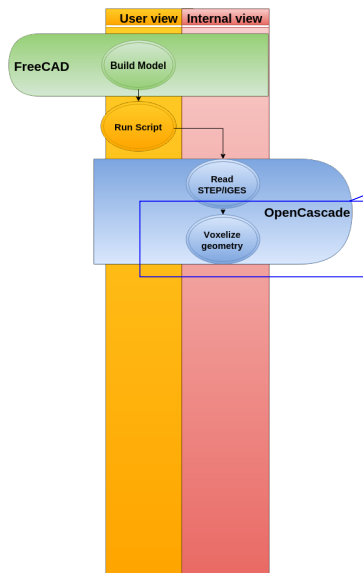






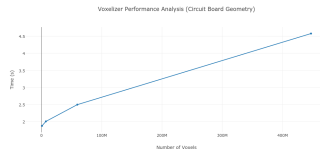




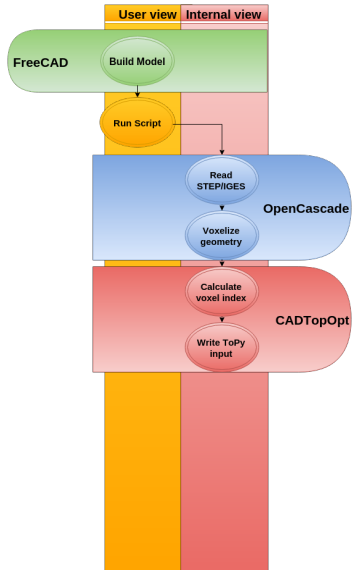


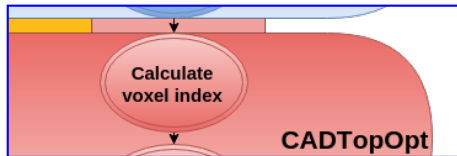
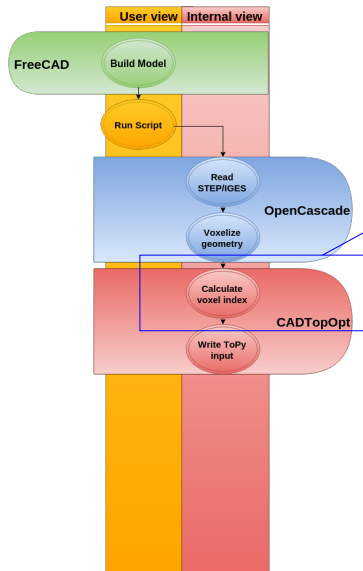
Voxelize faces using OpenCascade

- Included open cascade voxelizer



Scaling of voxelizer



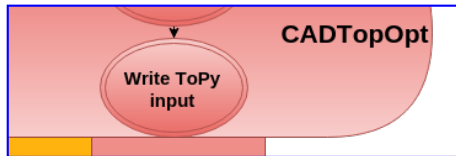


Different indexing for elements and nodes in ToPy

```
#####
# *** 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
#
```

Indexing in ToPy

## 2.1. User experience and internal structure

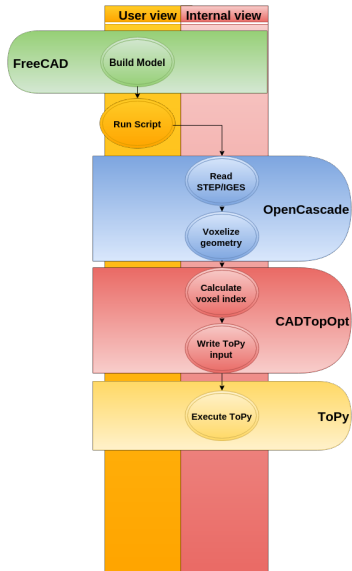


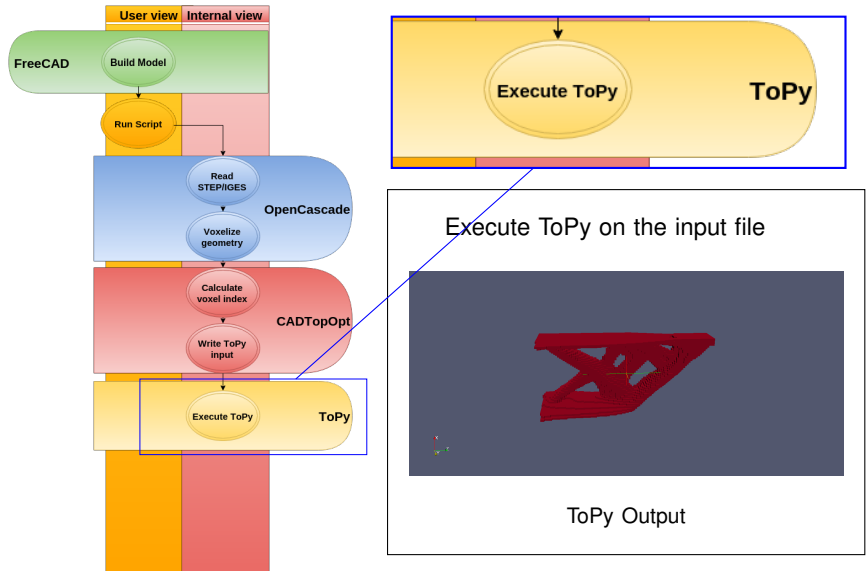
[Tafy Problem Definition File v2887]

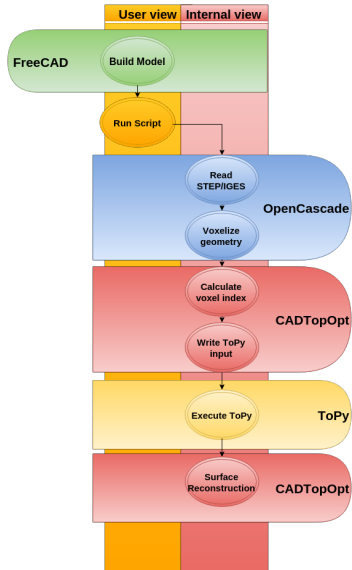
[illegible]

## Script for ToPy



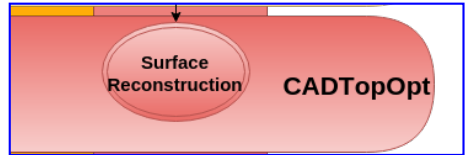
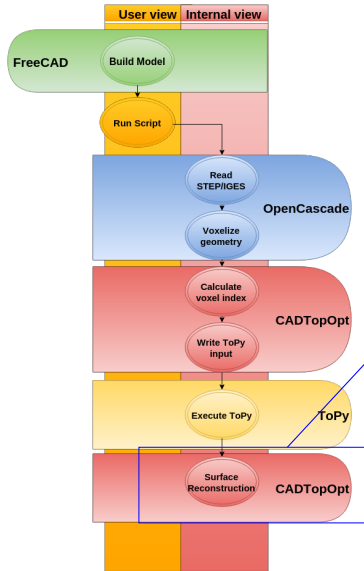




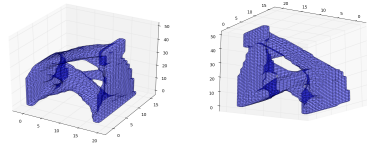


## 2. Topology optimization

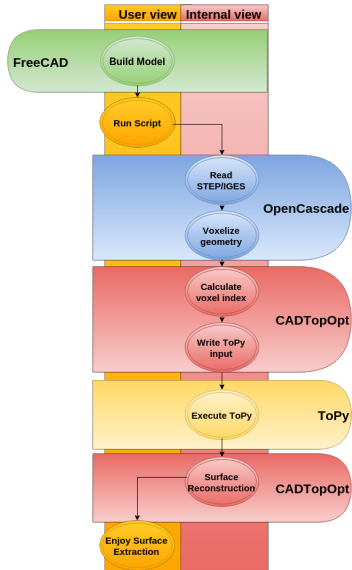
### 2.1. User experience and internal structure

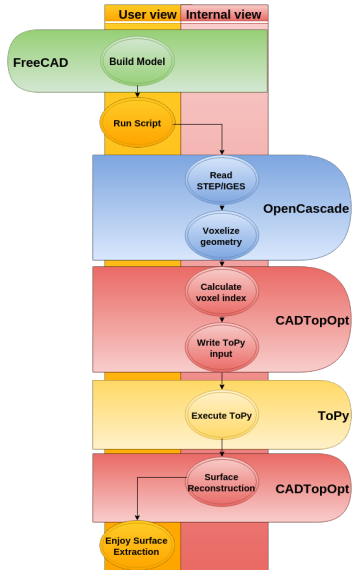


Running dual contouring algorithm

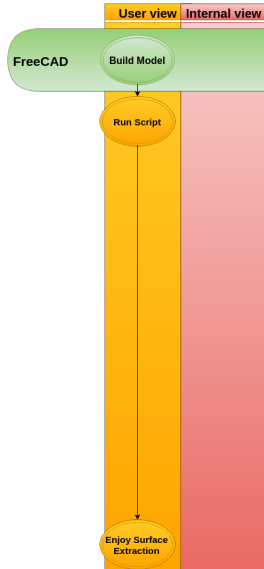


Surface extraction for Cantilever





But what does the user see?



But what does the  
user see?  
This!

## The next steps MOVE TO LATER

- Speed up ToPY
- Usage of different optimizers
- GUI for input



# Contents

## 1. Introduction

- 1.1 Motivation
- 1.2 Workflow Overview
- 1.3 Organization

## 2. Topology optimization

- 2.1 User experience and internal structure
- 2.2 The next steps MOVE TO LATER

## 3. Surface Extraction

- 3.1 Dual Contouring
- 3.2 Projection and Parametrization

## 4. B-Spline Fitting

- 4.1 Peters' scheme
- 4.2 Fitting pipeline

## 5. Summary & Outlook

# Status

## Last milestone

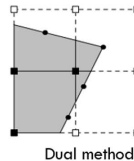
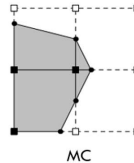
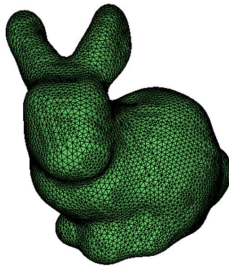
- 🕒 Surface reconstruction with the VTK Toolbox

## 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

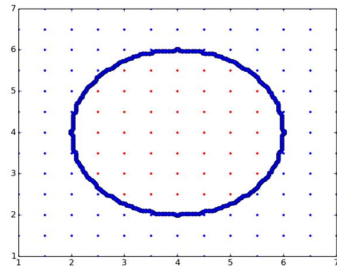
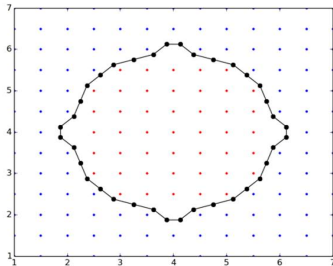
## From Voxel to Mesh Geometry

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



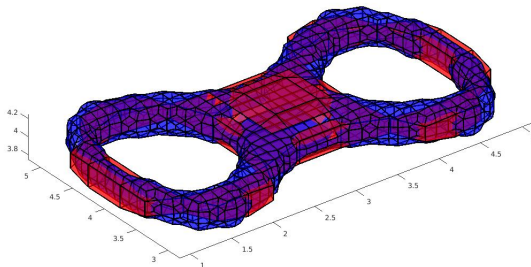
## Dual Contouring

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



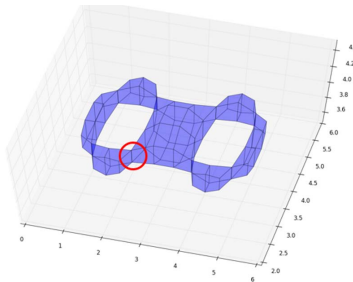
## Dual Contouring

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



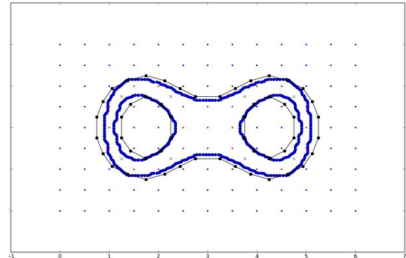
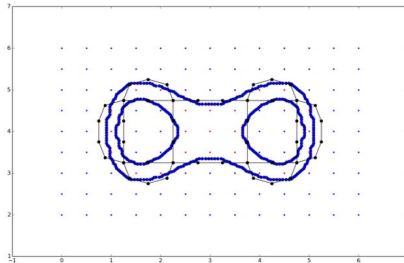
## Dual Contouring — Problems

- **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

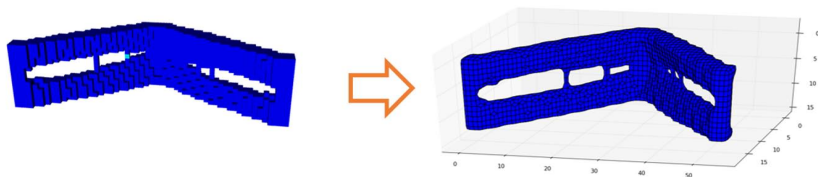


## Dual Contouring — Problems

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



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

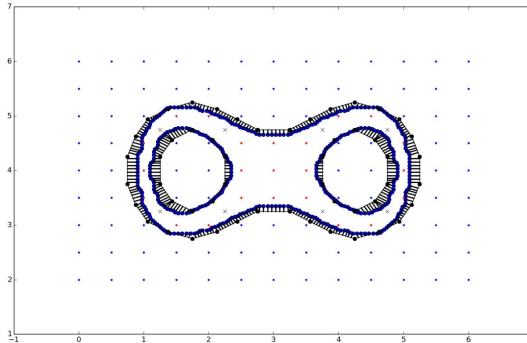




# Demo

## 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



# Contents

## 1. Introduction

- 1.1 Motivation
- 1.2 Workflow Overview
- 1.3 Organization

## 2. Topology optimization

- 2.1 User experience and internal structure
- 2.2 The next steps MOVE TO LATER

## 3. Surface Extraction

- 3.1 Dual Contouring
- 3.2 Projection and Parametrization

## 4. B-Spline Fitting

- 4.1 Peters' scheme
- 4.2 Fitting pipeline

## 5. Summary & Outlook

# B-Spline

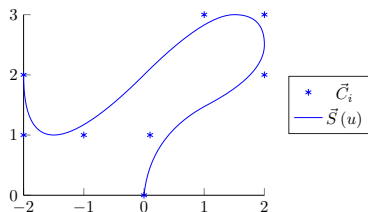
$$\vec{S}(u, v) = \sum_{i,j=1}^{n,m} \vec{C}_{i,j} N_i^p(u) N_j^p(v),$$

where  $p$  – degree of the B-Spline surface and  $n, m$  – number of control points in each direction.

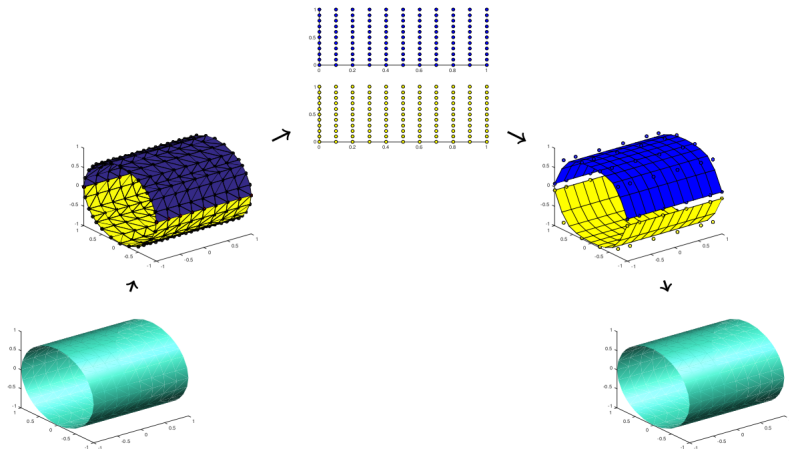
## B-Splines

- offer great flexibility for handling arbitrary shapes
- are CAD-standard

**Engineers are working with CAD**



# B-Spline Fitting Pipeline [Becker, Schäfer, Jameson]



# Status

## Last milestone

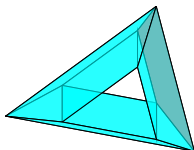
- ✗ Automatic patch selection
- ✗ Parametrization of obtained patches
- ✓ B-spline fitting using least squares
- 🕒 Smooth connection of patches
- ✗ Conversion back to CAD

## Today

- ✓ Automatic patch selection – moved to the surface extraction part
- ✓ Parametrization of obtained patches
- ✓ B-spline fitting using least squares – modified
- ✓ Smooth connection of patches
- ✗ Conversion back to CAD

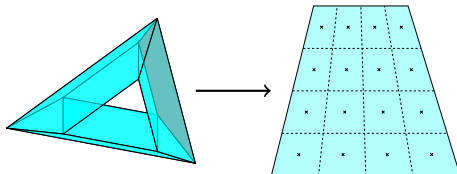
## Long way to smoothness. Peter's scheme

Control mesh



## Long way to smoothness. Peter's scheme

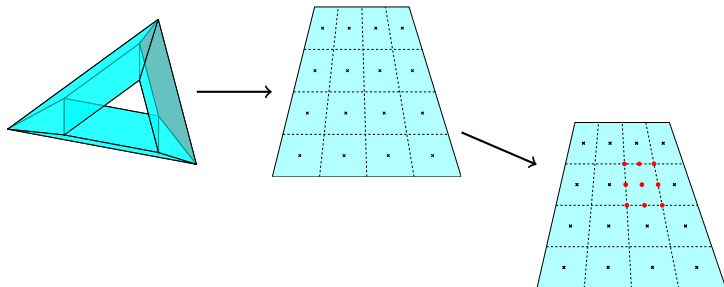
Refined control mesh





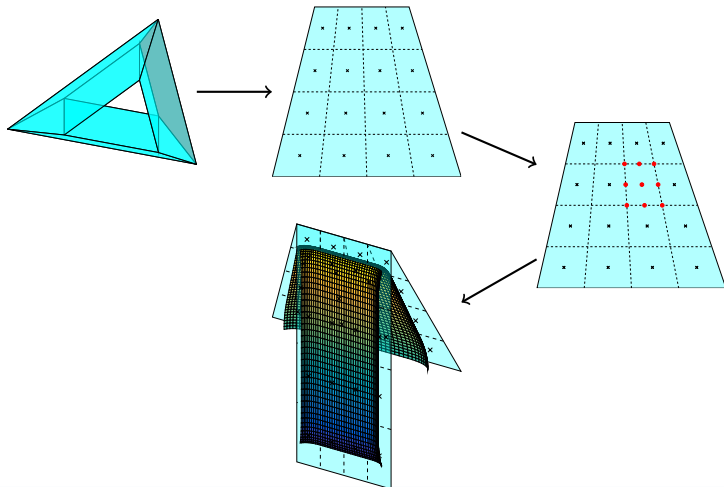
## Long way to smoothness. Peter's scheme

Bezier control points



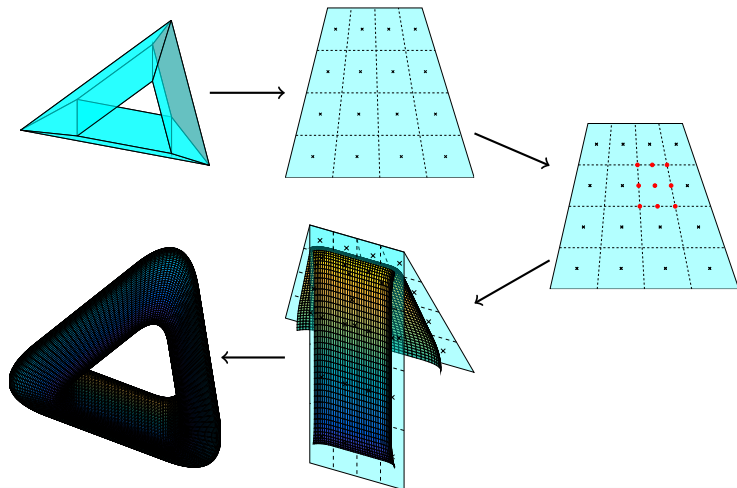
## Long way to smoothness. Peter's scheme

B-Spline patch



## Long way to smoothness. Peter's scheme

Peters' surface



# Long way to smoothness

## Main ideas

- Use the mesh obtained from Dual Contouring as a *control mesh*
- Modify the fitting step to take advantage of the **Peters' scheme**

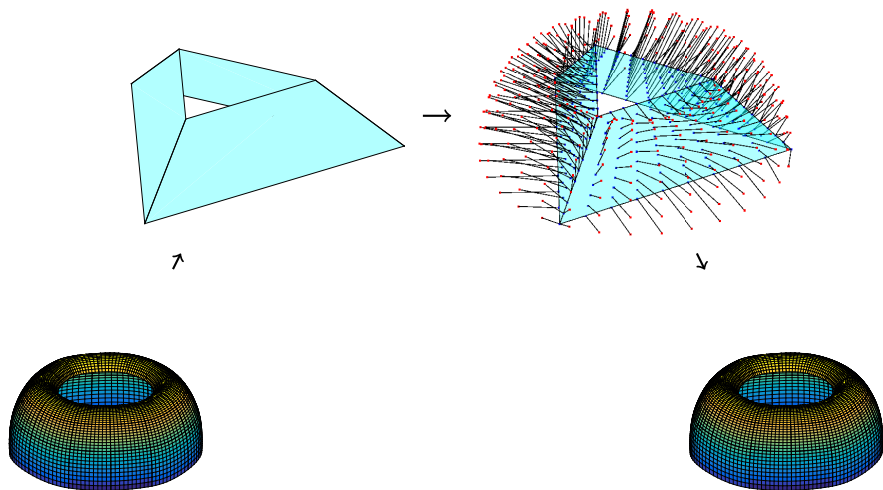
$$\downarrow$$
$$E_{dist}(V_x) = \sum_{i=1}^N \| P_i - y_i V_x \|_2^2 \rightarrow \min,$$

$y_i$  - coefficients obtained from the Peters' scheme theory.

## What is achieved?

- Smoothness of the fitted surface is now guaranteed by construction
- Fitting is possible for more complex shapes achieved by using an information from the Dual Contouring algorithm

## Improved pipeline



## Before and after Peters

## What is next?

### Further steps:

- Full integration with Surface Extraction part
- Exporting the results back to CAD

### Possible optimizations

- Introducing of the *fairness functional* in order to deal with more complex shapes
- Implementation of the *adaptive refinement* in order to control a maximum error tolerance
- Implementation of the *parameter correction* for the improved pipeline

# Contents

## 1. Introduction

- 1.1 Motivation
- 1.2 Workflow Overview
- 1.3 Organization

## 2. Topology optimization

- 2.1 User experience and internal structure
- 2.2 The next steps MOVE TO LATER

## 3. Surface Extraction

- 3.1 Dual Contouring
- 3.2 Projection and Parametrization

## 4. B-Spline Fitting

- 4.1 Peters' scheme
- 4.2 Fitting pipeline

## 5. Summary & Outlook



## What is done? What is next?

- Topology Optimization
  - ✓ Pipeline from CAD model to optimized voxel model
  - ✓ Support user input of boundary conditions
  - ✗ GUI for user interaction

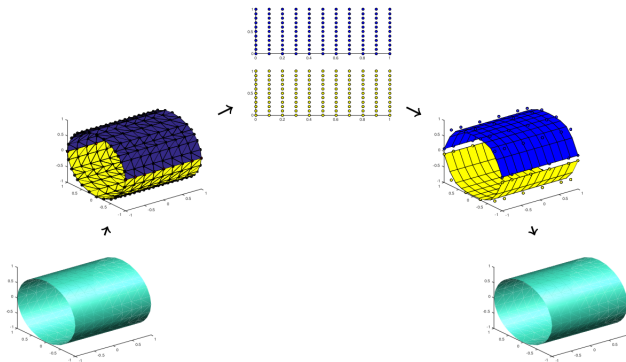
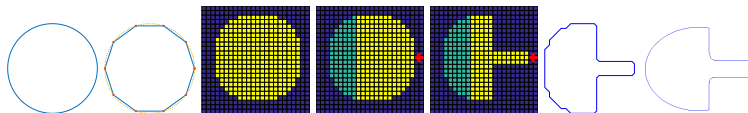
## What is done? What is next?

- Topology Optimization
  - ✓ Pipeline from CAD model to optimized voxel model
  - ✓ Support user input of boundary conditions
  - ✗ GUI for user interaction
- Surface Extraction
  - ✓ Dual Contouring for simple geometries
  - ✓ Provide necessary data for Surface Fitting
  - 🕒 Automatic patch distribution
  - ✗ Adaptive and topology safe Dual Contouring

## What is done? What is next?

- Topology Optimization
  - ✓ Pipeline from CAD model to optimized voxel model
  - ✓ Support user input of boundary conditions
  - ✗ GUI for user interaction
- Surface Extraction
  - ✓ Dual Contouring for simple geometries
  - ✓ Provide necessary data for Surface Fitting
  - ⌚ Automatic patch distribution
  - ✗ Adaptive and topology safe Dual Contouring
- Surface Fitting
  - ✓ B-spline fitting using least squares
  - ✓ Smooth connection of patches using Peters' scheme
  - ✗ Conversion back to CAD

# Thank you for your attention!

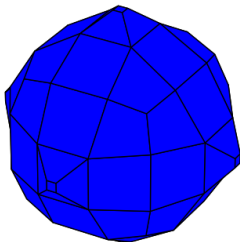


## Literature

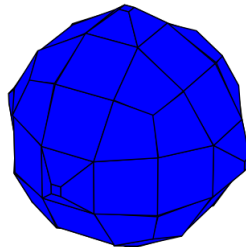
- **William Hunter.** "Predominantly solid-void three-dimensional topology optimisation using open source software"
- **Gerrit Becker, Michael Schäfer, Antony Jameson.** "An advanced NURBS fitting procedure for post-processing of grid-based shape optimizations"
- **Matthias Eck, Hugues Hoppe.** "Automatic Reconstruction of B-Spline Surfaces of Arbitrary Topological Type"
- **Tao Ju, Frank Losasso, Scott Schaefer, Joe Warren.** "Dual contouring of hermite data"

## Projection and Parametrization on arbitrary quads

1. find least squares plane approximating quad



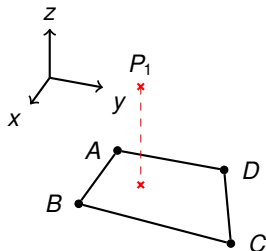
DC sphere



with plane quads

## Projection and Parametrization on arbitrary quads

1. find least squares plane approximating quad
2. projection of datapoint onto plane



### Coordinate transformation

system with basis

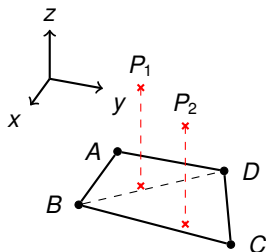
$$B_{BAD} = \begin{pmatrix} \vec{n} & \vec{AB} & \vec{AD} \end{pmatrix}$$

yields

$$(B_{BAD})^{-1} P_1 = \begin{pmatrix} d & u & v \end{pmatrix}^T$$

## Projection and Parametrization on arbitrary quads

1. find least squares plane approximating quad
2. projection of datapoint onto plane
3. find corresponding parameters  $[u, v] \in [0, 1]^2$



### Problem:

- ✓ for  $P_1: (u, v) = (0.5, 0.4)$
- ✗ for  $P_2: (u, v) = (1, 1)$

### Solution:

1. if we get  $u + v > 1$
2. use  $B_{BCD}$  instead of  $B_{BAD}$
3. set  $u = 1 - u, v = 1 - v$