



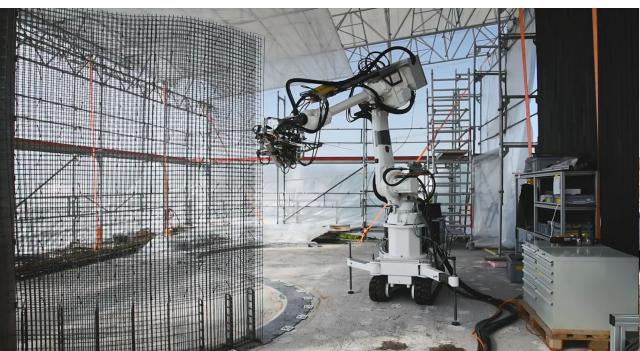
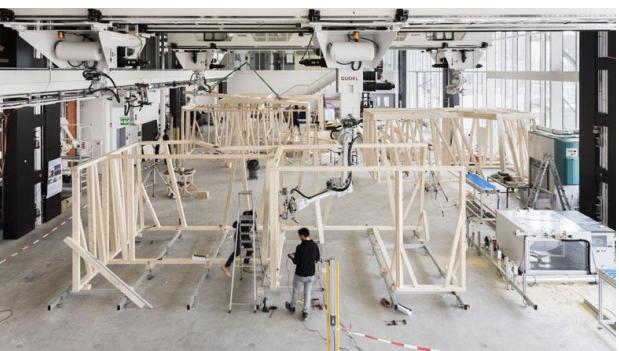
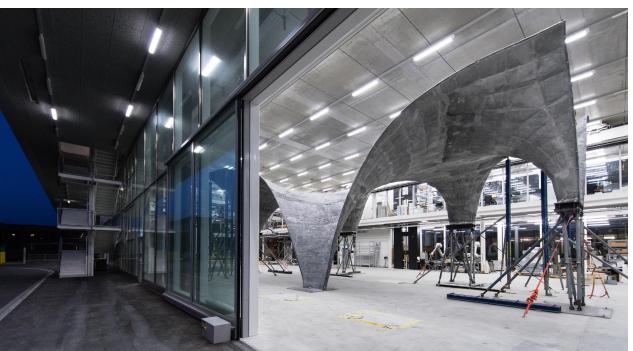
C O M P A S

```
    if key in mesh.vertices():
        if key in fixed:
            continue

        p = key_xyz[key]
        nbrs = mesh.vertex_neighbours(key, ordered=True)
        c = center_of_mass_polygon([[key_xyz[nbr] for nb
```

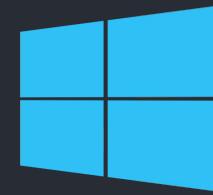
Computational Framework for Research and Collaboration in AE(F)C

- No more reinventing the wheel
- Generate reproducible results
- Shareable work
- Transparent collaboration
- Clear communication
- Interaction with industry
- ...



Installation

⌘ + SPACE



Terminal

Anaconda Prompt

vanmelet — -bash — 112x38

Last login: Mon Sep 30 23:03:58 on ttys001
(base) arch-hix-dock-453:- vanmelet\$ []

Anaconda Prompt

(base) C:\Users\user>■


```
conda config --add channels conda-forge  
conda create -n ita19 python=3.6 python.app COMPAS=0.8.1 --yes  
conda activate ita19  
python
```



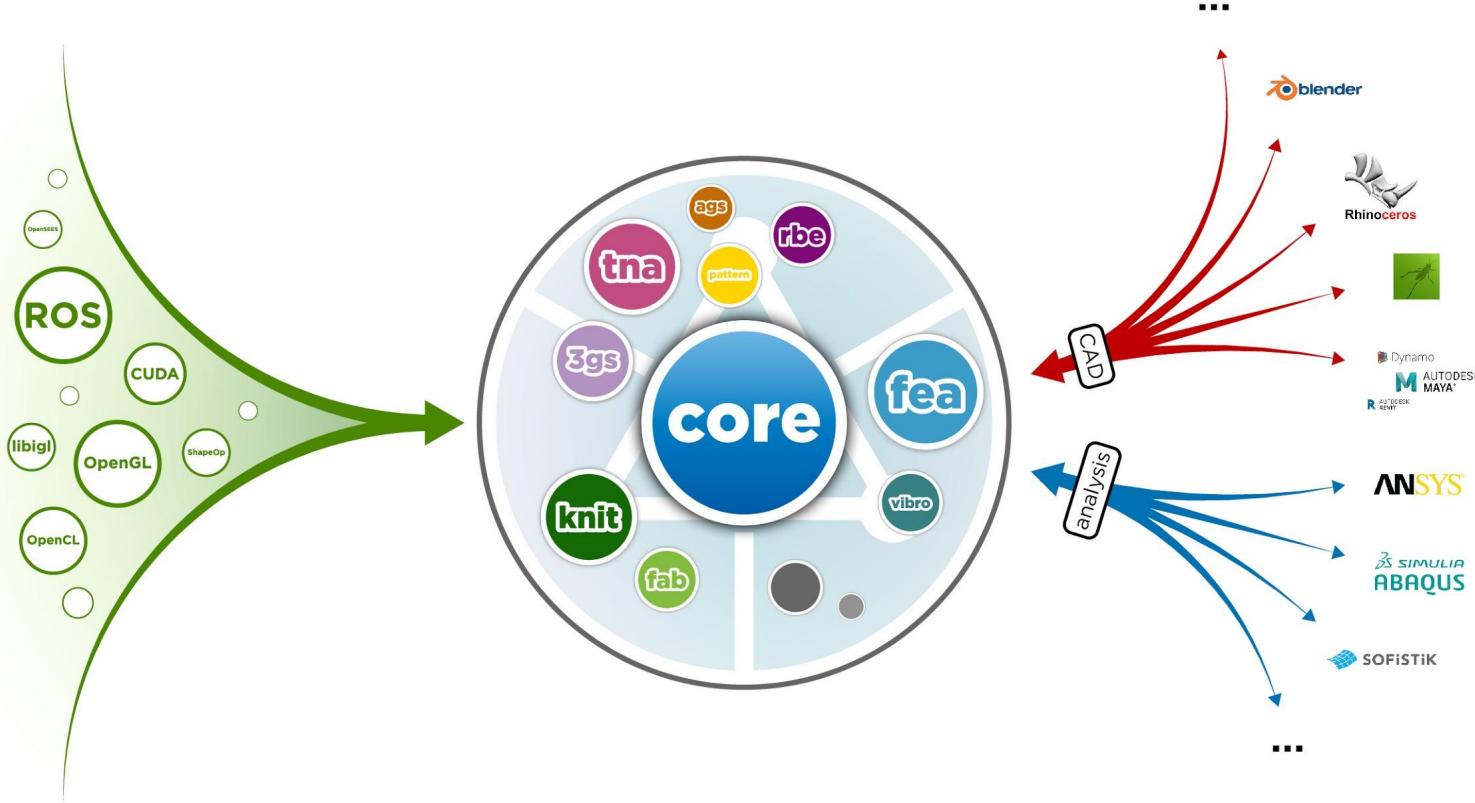
```
conda config --add channels conda-forge  
conda create -n ita19 python=3.6 python.app COMPAS=0.8.1 --yes  
conda activate ita19  
python
```

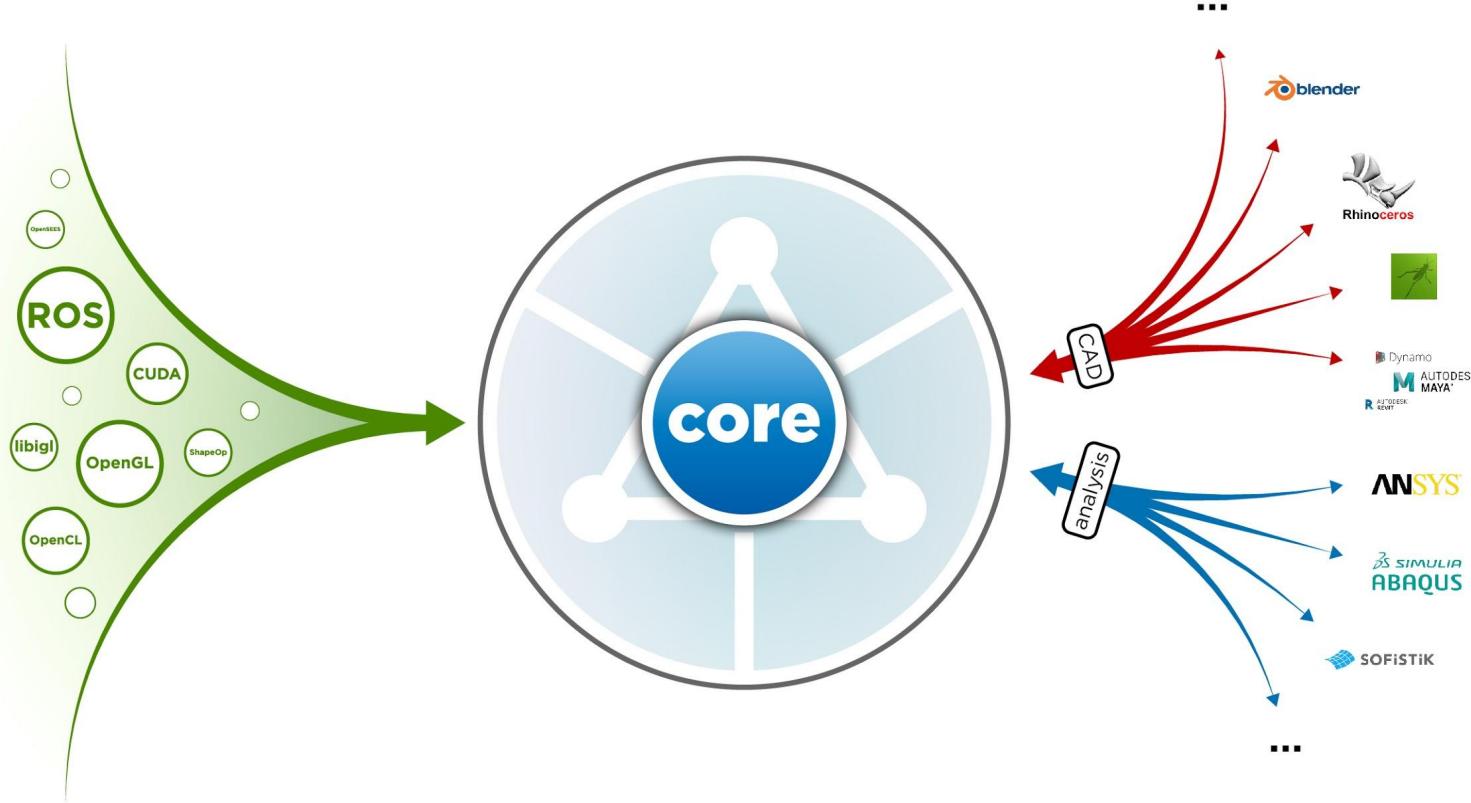
```
conda config --add channels conda-forge  
conda create -n ita19 python=3.6 python.app COMPAS=0.8.1 --yes  
conda activate ita19  
python
```

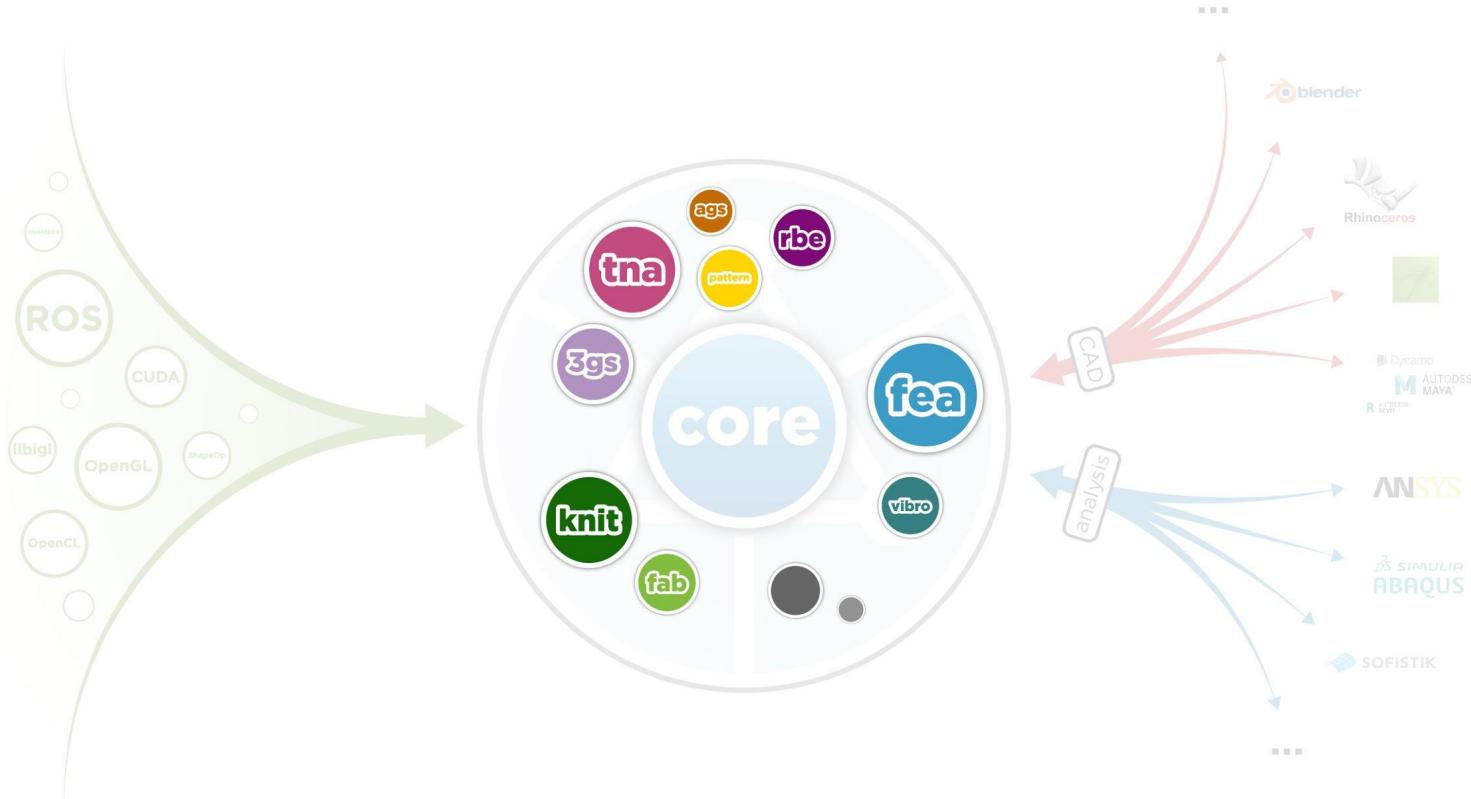
```
conda config --add channels conda-forge  
conda create -n ita19 python=3.6 python.app COMPAS=0.8.1 --yes  
conda activate ita19  
python
```

```
conda config --add channels conda-forge  
conda create -n ita19 python=3.6 python.app COMPAS=0.8.1 --yes  
conda activate ita19  
python
```

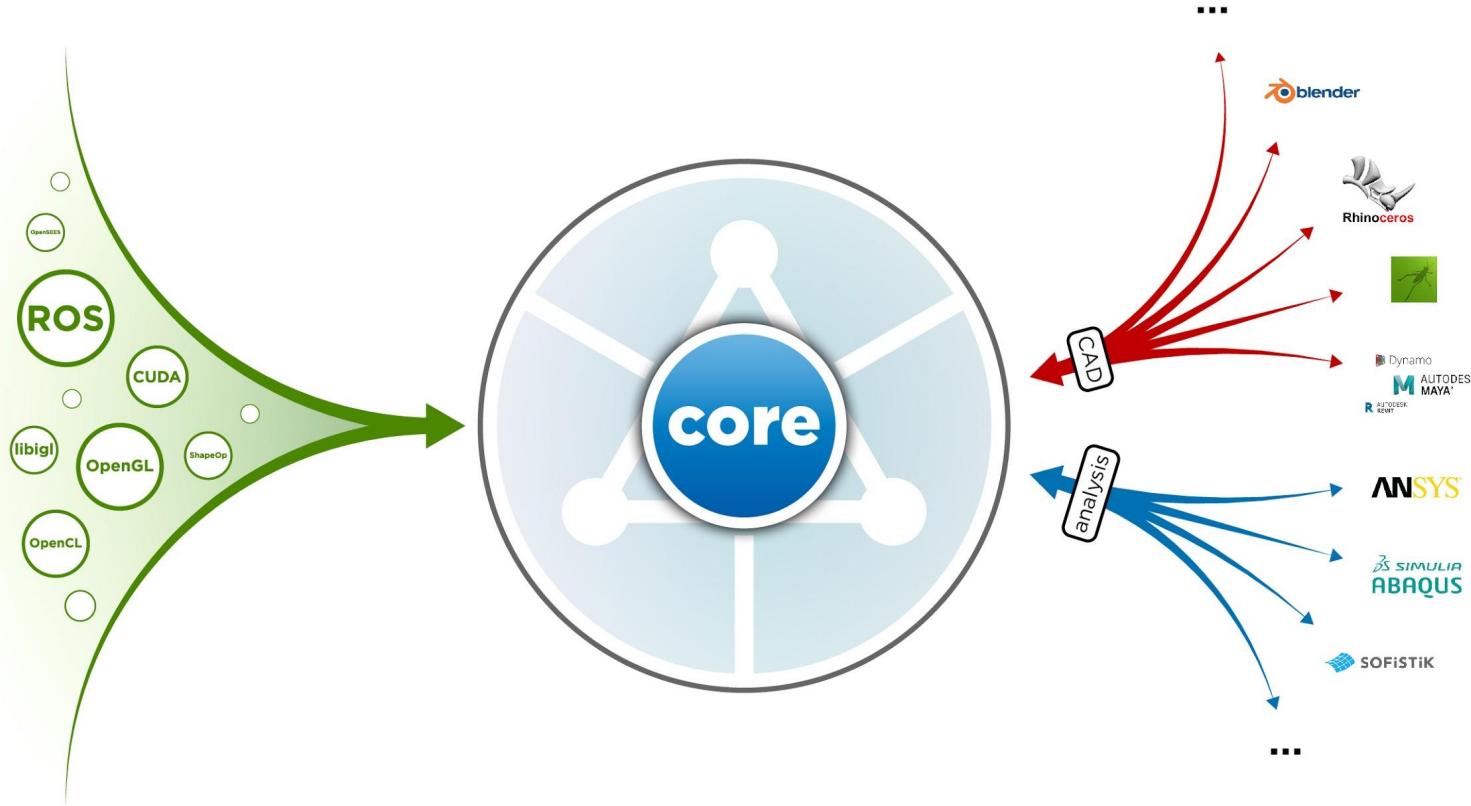
```
conda config --add channels conda-forge
conda create -n ita19 python=3.6 python.app COMPAS=0.8 --yes
conda activate ita19
python
```







COMPAS main



COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

Primitives & Shapes

- Point
- Vector
- Line
- Plane
- Frame
- Circle
- Box
- Cone
- Cylinder
- Torus

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

Basic functions

- Angles
- Distance
- Size
- Intersections
- Offsets
- ...

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

Algorithms

- Bounding boxes
- Convex hull
- Interpolation
- Planarisation
- Smoothing
- Transformations
- ...

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

Numerical solvers & algorithms

- Differential Evolution
- Dynamic Relaxation
- Force Density Method
- Genetic Algorithms
- Principal Component Analysis
- Topology Optimisation

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

Robotic fundamentals

- Robot Model
- Links
- Joints
- URDF & ROS package support
- Forward Kinematics

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

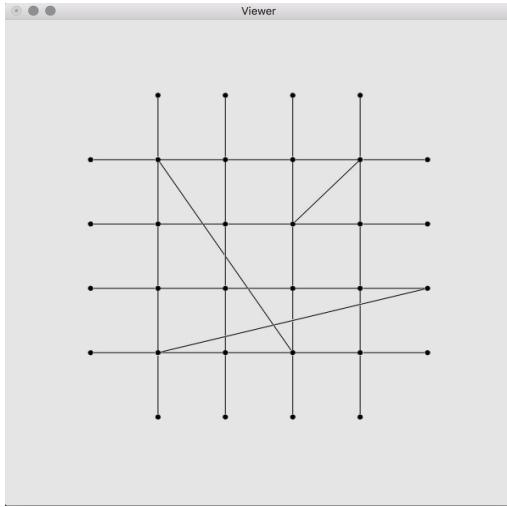
compas_plotters

compas_rhino

Flexible data structures

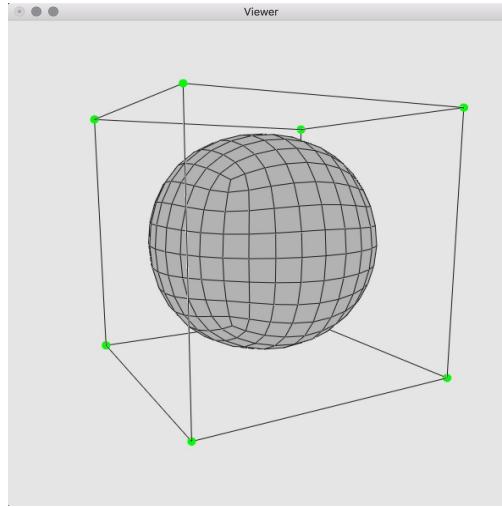
- Network
- Mesh
- VolMesh

Network



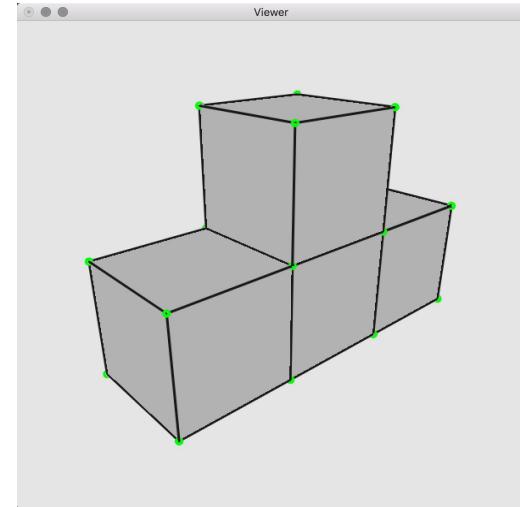
- General networks
- Edge graph

Mesh

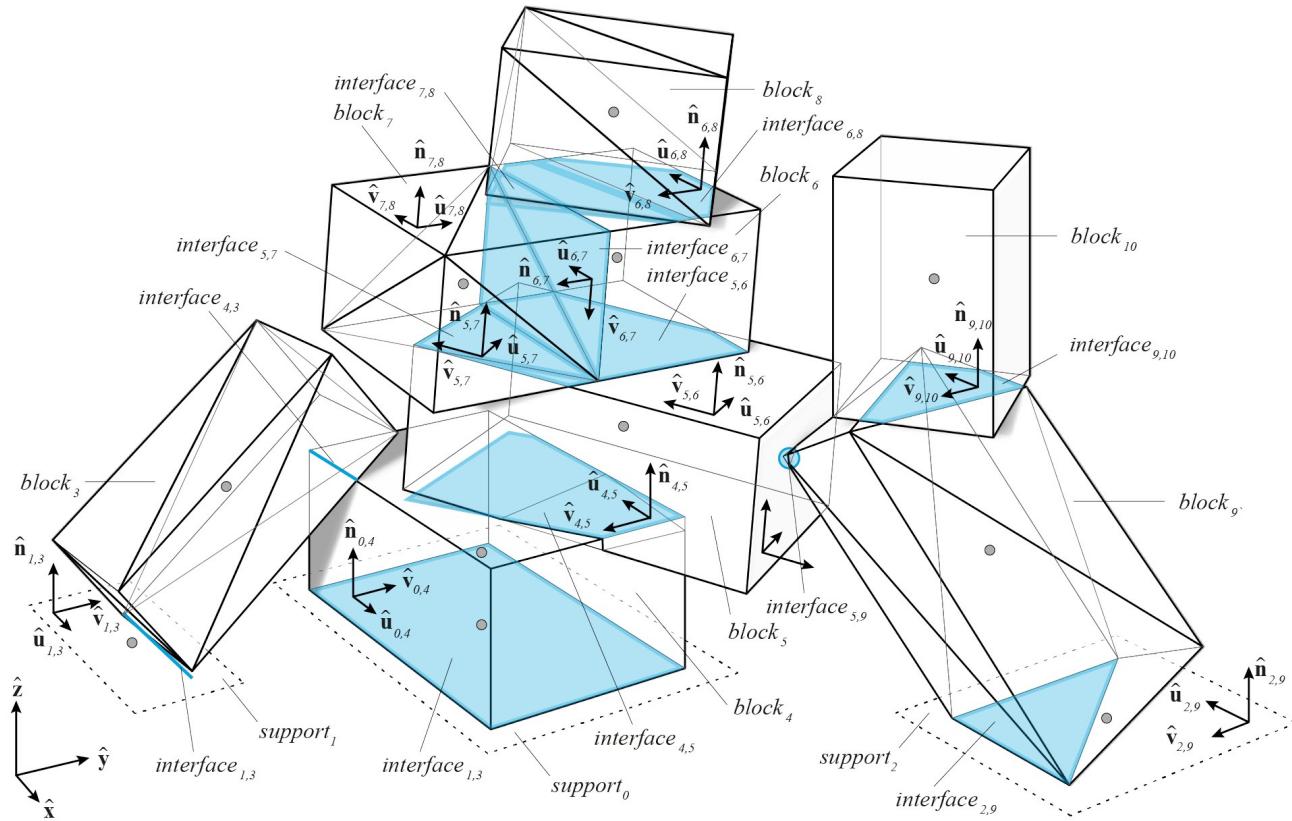


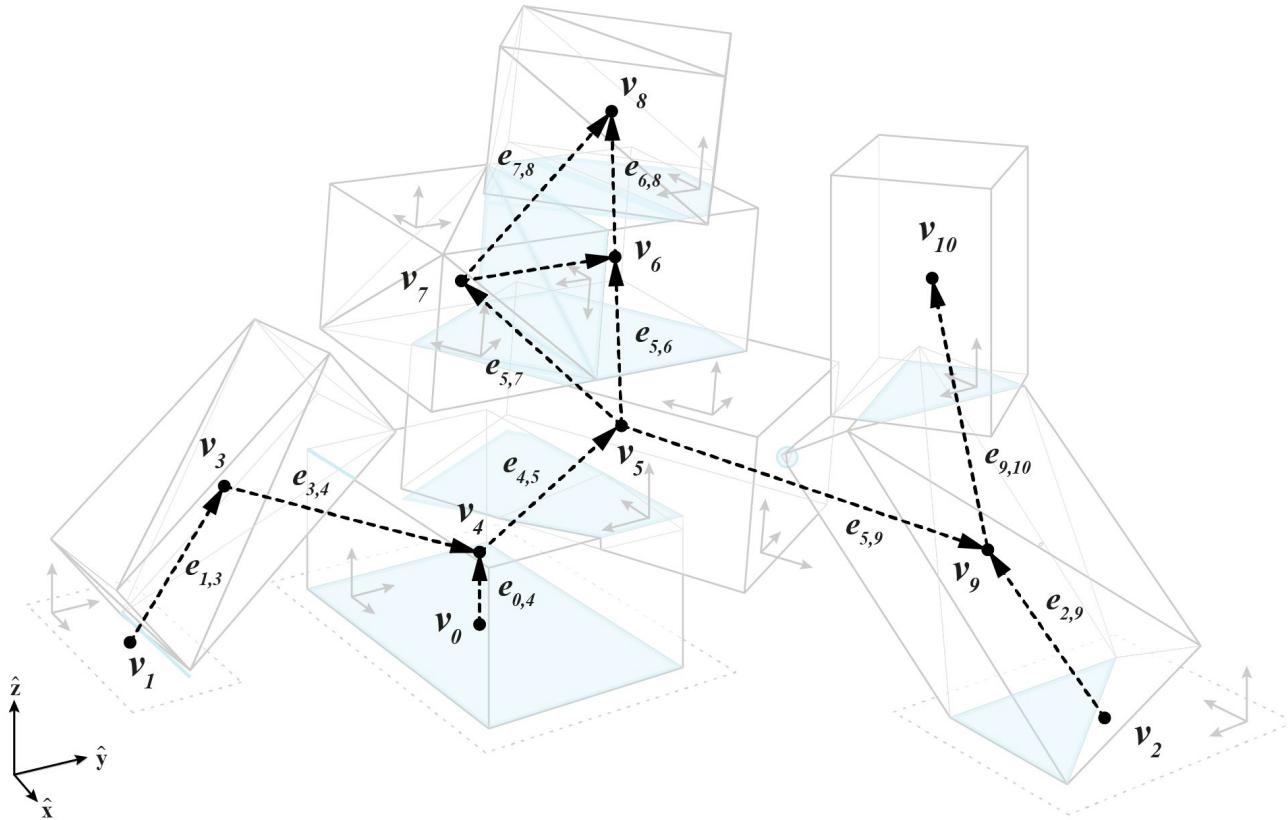
- Surface meshes
- Half-edge

VolMesh



- Cellular meshes
- Half-plane





COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

Basic visualisation tools

- 2D plots
- Dynamic plotting
- Interactive plots

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

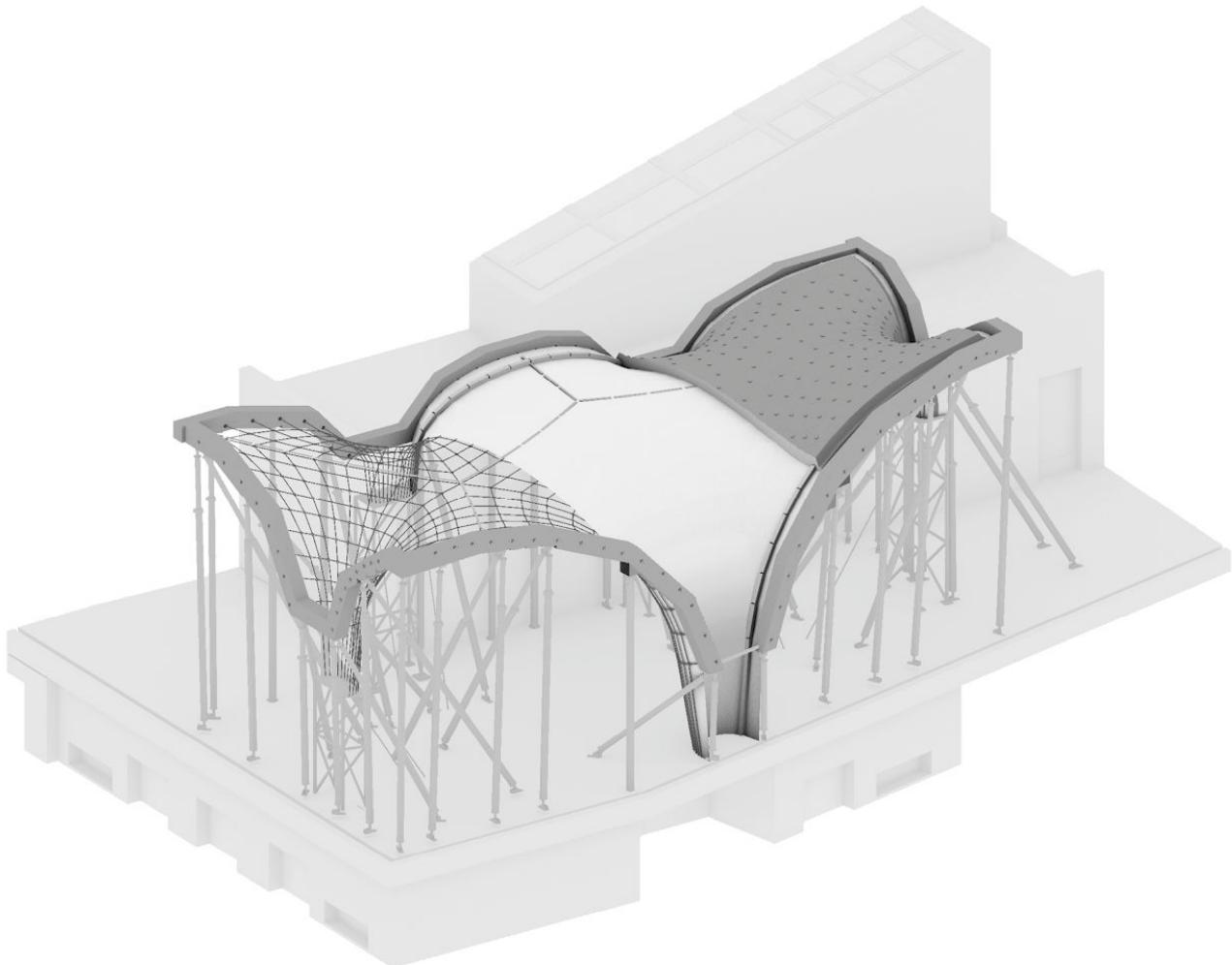
compas_ghpython

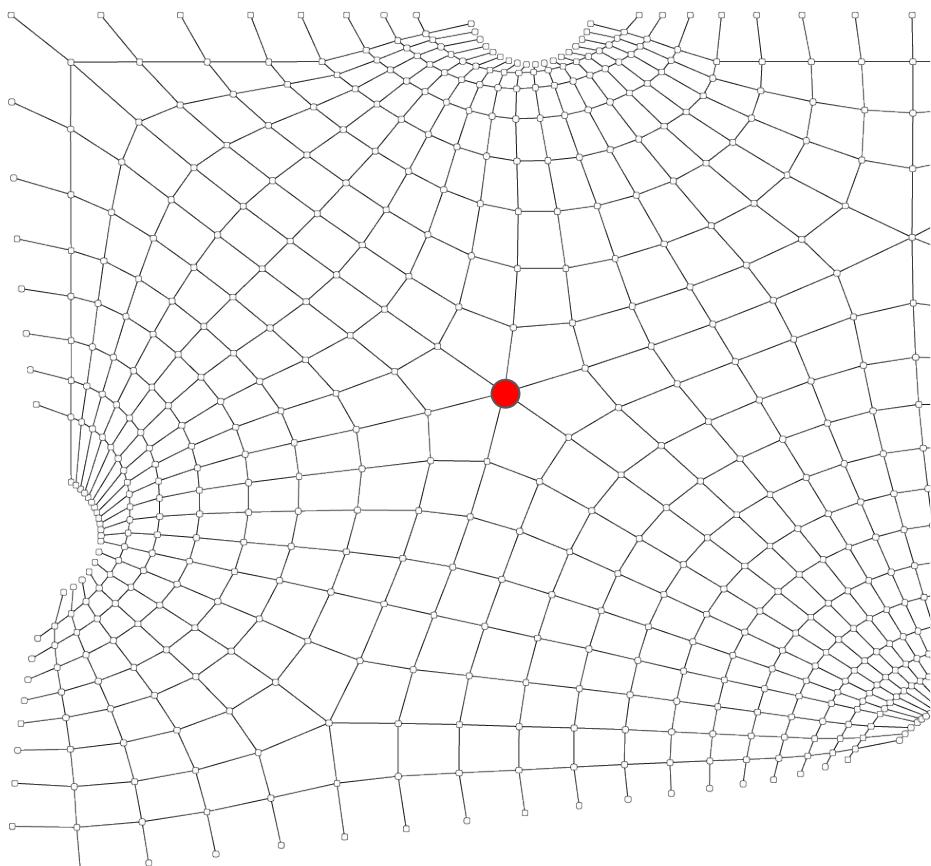
compas_plotters

compas_rhino

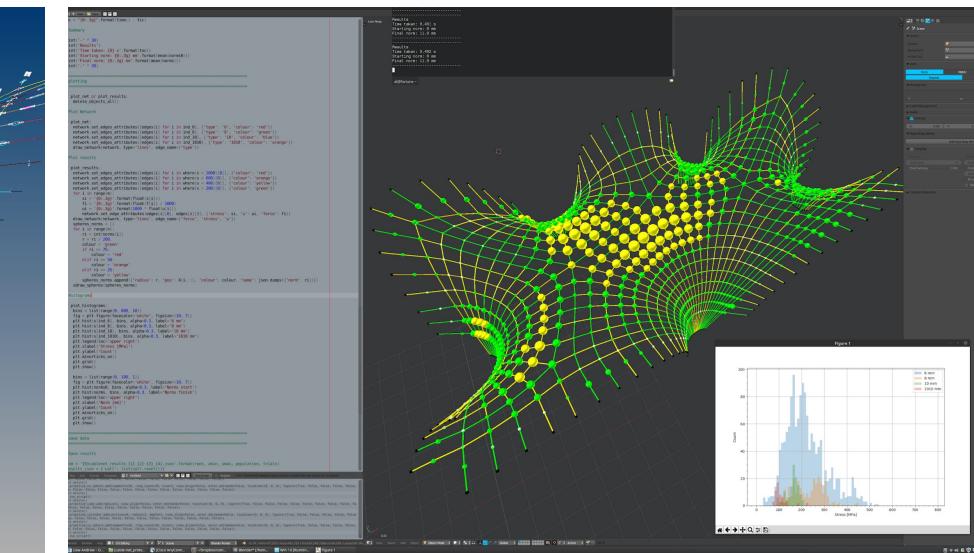
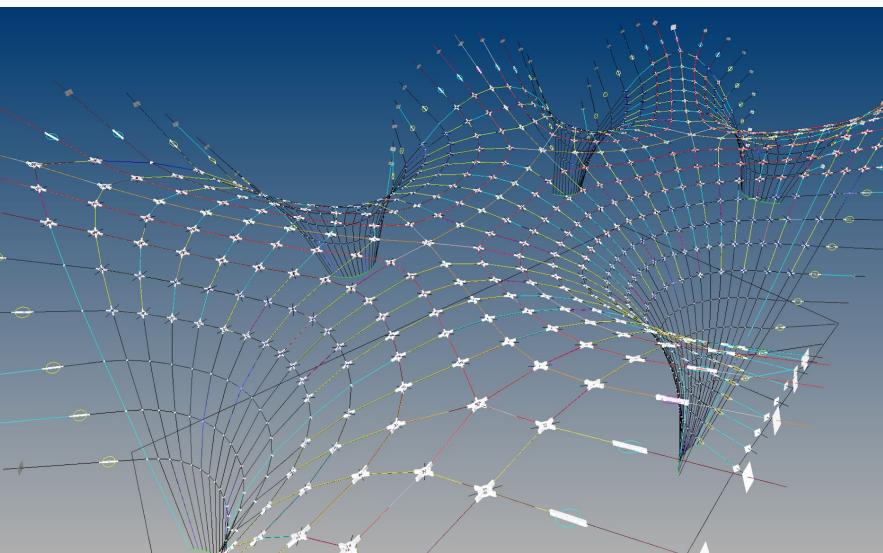
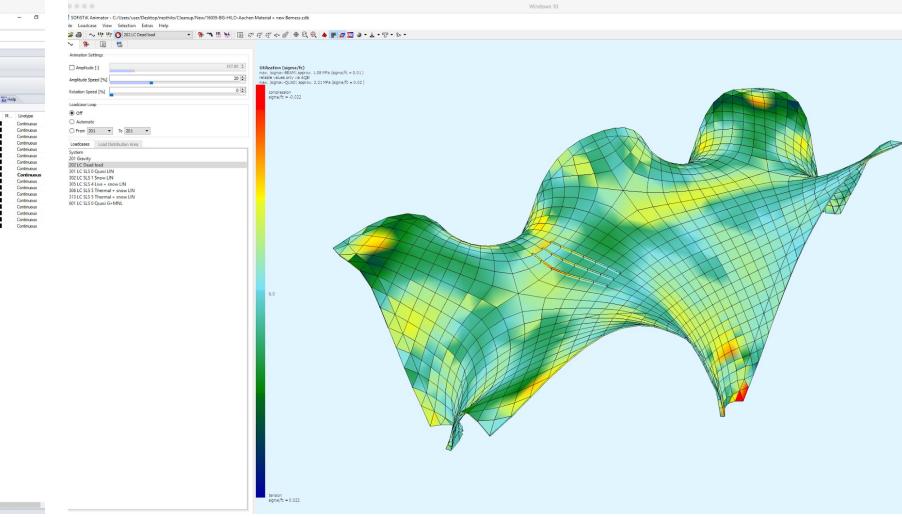
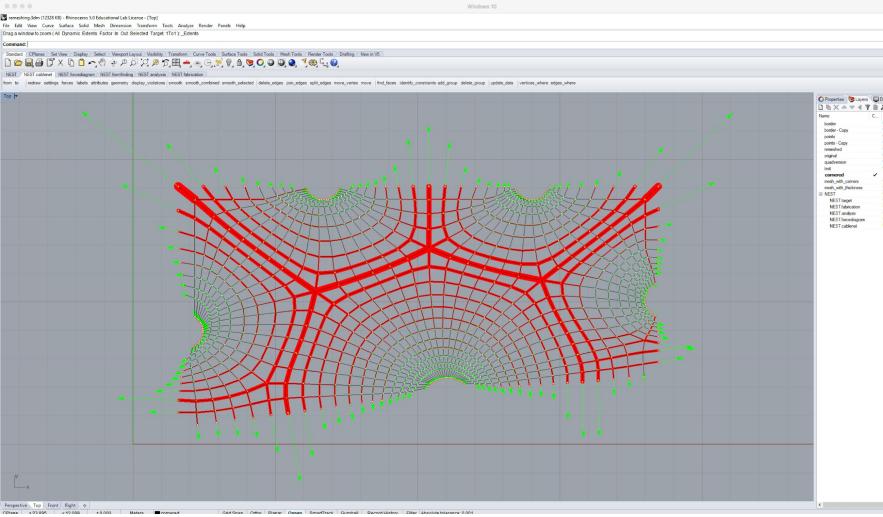
CAD support

- Integration in CAD software
- Unified scripting interface
- ...





```
cablenet.vertex[745] => {  
    'constraint_name': 'target',  
    'constraint_type': None,  
    'cx': 0.950,  
    'cy': 0.950,  
    'damping': 1,  
    'is_anchor': False,  
    'is_constrained': True,  
    'is_fixed': False,  
    'is_ridge': False,  
    'px': 0.000,  
    'py': 0.000,  
    'pz': 0.000,  
    'ring_type': 'XL',  
    'rx': 0.000,  
    'ry': 0.000,  
    'rz': 0.000,  
    'sag': 0.024,  
    'sw1': 0.672,  
    'sw2': 0.981,  
    'x': 6.405,  
    'y': 5.366,  
    'z': 5.653,
```



COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

```
# HiLo: unroll-fabric.py
```

```
from compas.utilities import pairwise
from compas.rpc import Proxy
```

```
from compas.numerical import dr
```

```
from compas.geometry import cross_vectors
from compas.geometry import normalize_vector
```

```
from compas.geometry import Point
from compas.geometry import Frame
from compas.geometry import Transformation
```

```
from compas.datastructures import Mesh
from compas.datastructures import mesh_quads_to_triangles
from compas.datastructures import mesh_transform
```

```
from compas_hilo.datastructures import Shell
from compas_hilo.rhino import ShellArtist
```

```
FILE_I = os.path.join(DATA, 'fabric.json')
FILE_O = os.path.join(DATA, 'fabric-flattened.json')
```

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

```
# HiLo: unroll-fabric.py
```

```
from compas.utilities import pairwise
from compas.rpc import Proxy

from compas.numerical import dr

from compas.geometry import cross_vectors
from compas.geometry import normalize_vector

from compas.geometry import Point
from compas.geometry import Frame
from compas.geometry import Transformation

from compas.datastructures import Mesh
from compas.datastructures import mesh_quads_to_triangles
from compas.datastructures import mesh_transform

from compas_hilo.datastructures import Shell
from compas_hilo.rhino import ShellArtist

FILE_I = os.path.join(DATA, 'fabric.json')
FILE_O = os.path.join(DATA, 'fabric-flattened.json')
```

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

```
# HiLo: unroll-fabric.py
```

```
from compas.utilities import pairwise
from compas.rpc import Proxy
```

```
from compas.numerical import dr
```

```
from compas.geometry import cross_vectors
from compas.geometry import normalize_vector
```

```
from compas.geometry import Point
from compas.geometry import Frame
from compas.geometry import Transformation
```

```
from compas.datastructures import Mesh
from compas.datastructures import mesh_quads_to_triangles
from compas.datastructures import mesh_transform
```

```
from compas_hilo.datastructures import Shell
from compas_hilo.rhino import ShellArtist
```

```
FILE_I = os.path.join(DATA, 'fabric.json')
```

```
FILE_O = os.path.join(DATA, 'fabric-flattened.json')
```

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

```
# HiLo: unroll-fabric.py
```

```
from compas.utilities import pairwise
from compas.rpc import Proxy
```

```
from compas.numerical import dr
```

```
from compas.geometry import cross_vectors
from compas.geometry import normalize_vector
```

```
from compas.geometry import Point
from compas.geometry import Frame
from compas.geometry import Transformation
```

```
from compas.datastructures import Mesh
from compas.datastructures import mesh_quads_to_triangles
from compas.datastructures import mesh_transform
```

```
from compas_hilo.datastructures import Shell
from compas_hilo.rhino import ShellArtist
```

```
FILE_I = os.path.join(DATA, 'fabric.json')
FILE_O = os.path.join(DATA, 'fabric-flattened.json')
```

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

Python

```
from compas.geometry import bounding_box  
from compas.geometry import convex_hull  
from compas.geometry import transform_points
```

```
from compas.numerical import dr
```

Numpy

```
from compas.geometry import bounding_box_numpy  
from compas.geometry import convex_hull_numpy  
from compas.geometry import transform_points_numpy
```

```
from compas.numerical import dr_numpy  
from compas.numerical import pca_numpy
```

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

Python

```
from compas.geometry import bounding_box  
from compas.geometry import convex_hull  
from compas.geometry import transform_points
```

```
from compas.numerical import dr
```

Numpy

```
from compas.geometry import bounding_box_numpy  
from compas.geometry import convex_hull_numpy  
from compas.geometry import transform_points_numpy
```

```
from compas.numerical import dr_numpy  
from compas.numerical import pca_numpy
```

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

Python

```
from compas.geometry import bounding_box  
from compas.geometry import convex_hull  
from compas.geometry import transform_points
```

```
from compas.numerical import dr
```

Numpy

```
from compas.geometry import bounding_box_numpy  
from compas.geometry import convex_hull_numpy  
from compas.geometry import transform_points_numpy
```

```
from compas.numerical import dr_numpy  
from compas.numerical import pca_numpy
```

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

Python

```
from compas.geometry import bounding_box  
from compas.geometry import convex_hull  
from compas.geometry import transform_points
```

```
from compas.numerical import dr
```

Numpy

~~```
from compas.geometry import bounding_box_numpy
from compas.geometry import convex_hull_numpy
from compas.geometry import transform_points_numpy
```~~~~```
from compas.numerical import dr_numpy  
from compas.numerical import pca_numpy
```~~

COMPAS main

compas

datastructures

files

geometry

numerical

robots

rpc

topology

utilities

compas_blender

compas_ghpython

compas_plotters

compas_rhino

Remote Procedure Calls

```
from compas.rpc import Proxy  
  
numerical = Proxy('compas.numerical')  
  
res = numerical.dr_numpy(...)
```

The screenshot shows a terminal window at the bottom left and a Jupyter Notebook interface at the top right.

Terminal:

```
arch-helix-dock-453:~ vanmele@t: python
Python 3.6.3 |Anaconda custom (64-bit)| (default, Oct  6 2017, 12:04:38)
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)] on darwin
Type "help", "copyright", "credits" or "license" for more information.

>>> import compas
>>> from compas.datastructures import Mesh
>>> mesh = Mesh.from_obj('compas.obj')
>>> mesh.summary()

=====
Mesh summary
=====

- name: Mesh
- vertices: 36
- edges: 68
- faces: 25
- vertex degree: 2/4
- face degree: 2/4

=====

```

Jupyter Notebook:

The notebook cell contains the following Python code:

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import compas

try:
    from numpy import array
    from numpy import pi
    from numpy import isnan
    from numpy import ones
    from numpy import zeros
    from scipy.linalg import norm
    from scipy.sparse import diags
except ImportError:
    compas.raise_if_not_installed()

from compas.numerical import connectivity_matrix
from compas.numerical import normrow
from compas.numerical import _all_
_all_ = ['dr_numpy']

K = [0.0,
     [0.0, 0.5],
     [0.5, 0.0, 0.5],
     [0.5, 0.5, 0.5],
     [1.0, 0.0, 0.0, 1.0],
     ]

class Conff():
    def __init__(self, c):
        self.c = c
        self.a = (1 - c * 0.5) / (1 + c * 0.5)
        self.b = 0.5 * (1 + self.a)

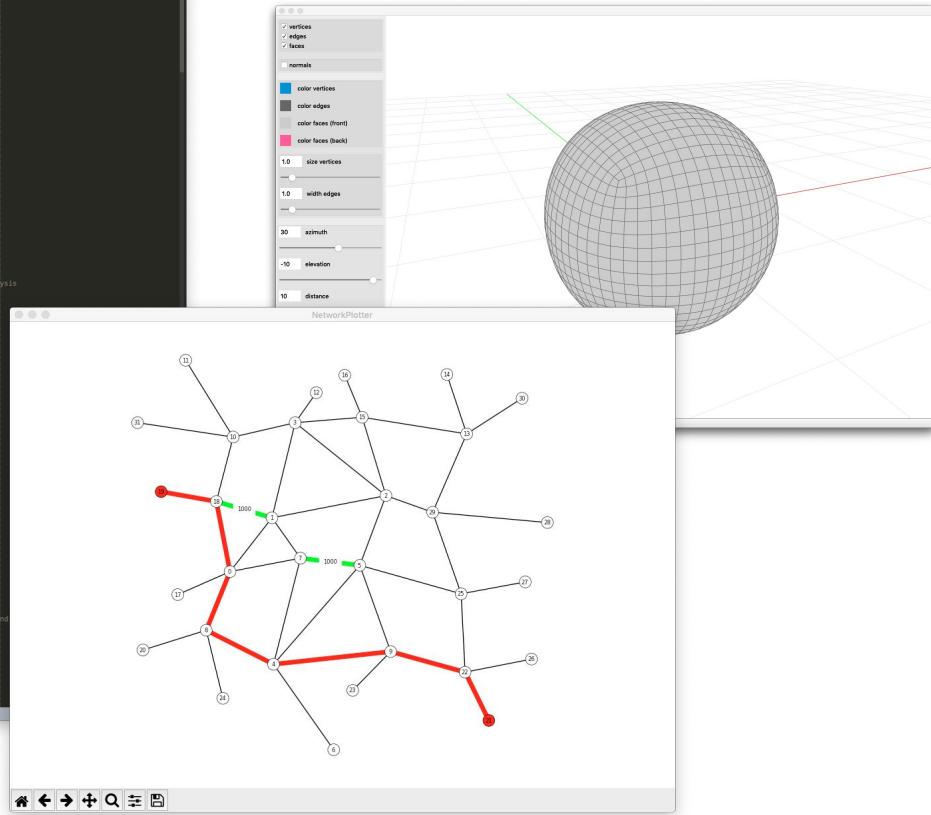
    def dr_numpy(vertices, edges, fixed, loads, qpre, fpre, lpre, limit, E, radius,
               callback=None, callback_kwargs=None, maxiter=1000000):
        """Implementation of the symplectic relaxation method for form finding and analysis
        of articulated networks of axial-force members.

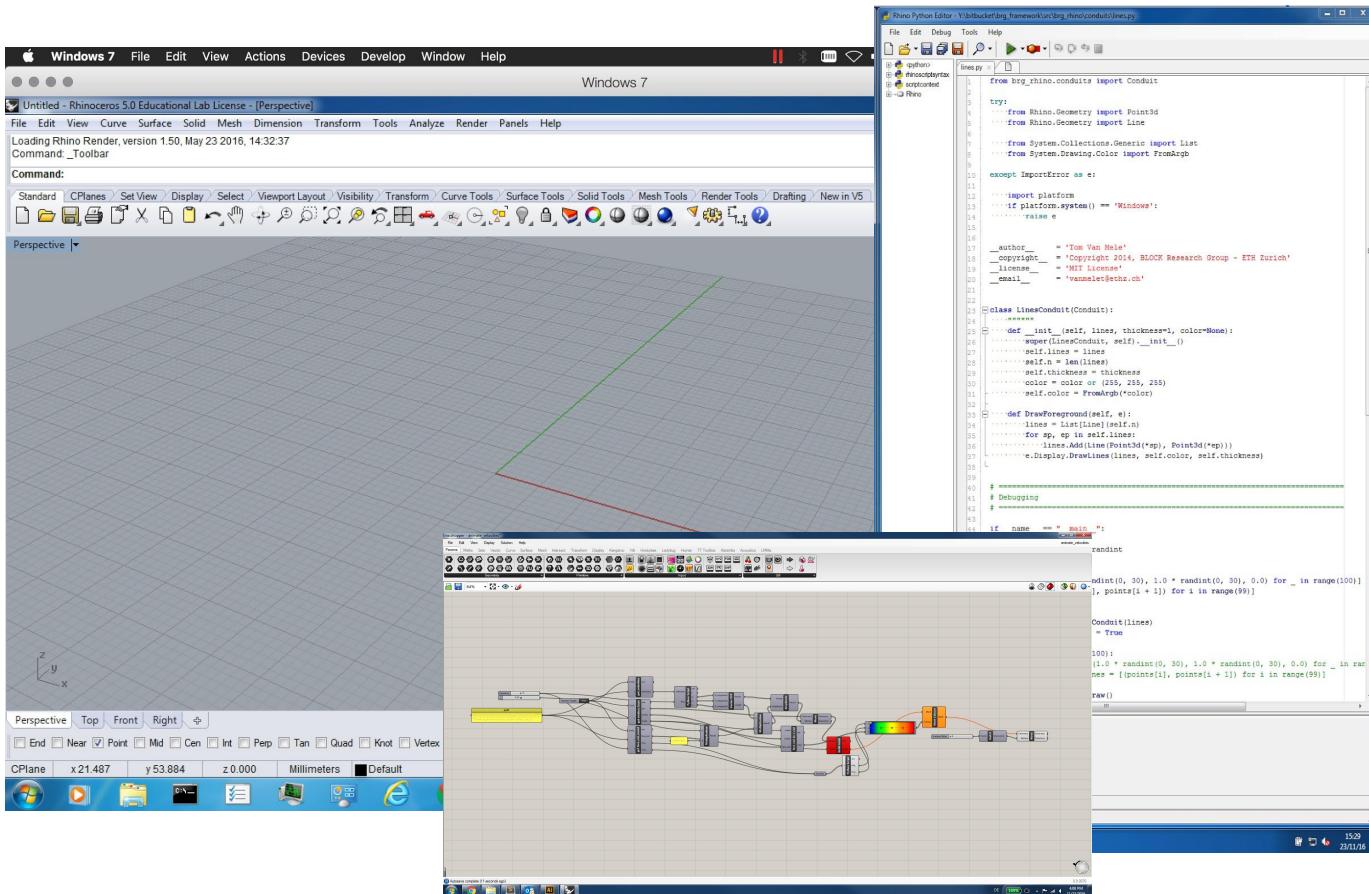
        Parameters
        ==========
        vertices : list
            XYZ coordinates of the vertices.
        edges : list
            Connectivity of the vertices.
        fixed : list
            Indices of the fixed vertices.
        limit : float
            Maximum displacement limit for the vertices.
        radius : float
            Radius of the sphere used for the callback.

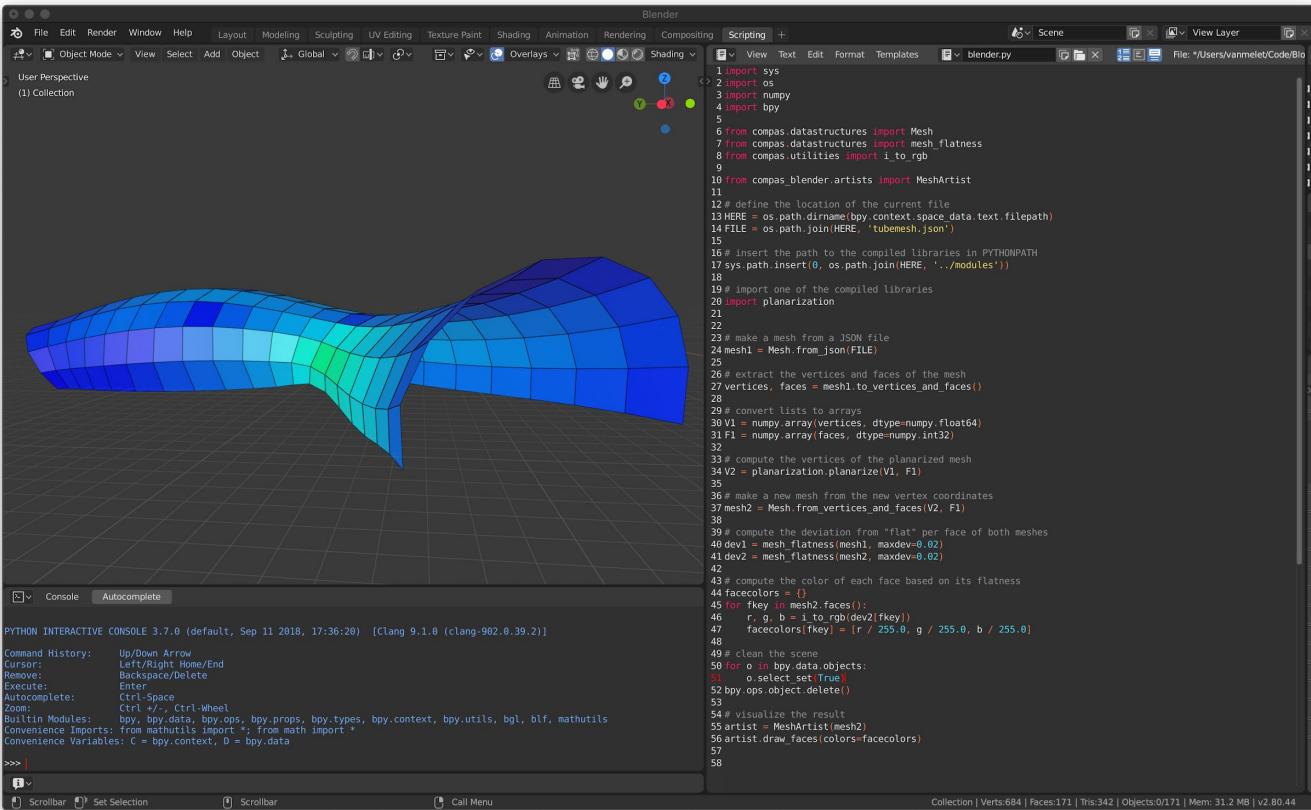
        Returns
        ======
        vertices : list
            XYZ coordinates of the vertices.
        edges : list
            Connectivity of the vertices.
        fixed : list
            Indices of the fixed vertices.
        limit : float
            Maximum displacement limit for the vertices.
        radius : float
            Radius of the sphere used for the callback.

        Notes
        =====
        This function uses a symplectic relaxation method to find the equilibrium configuration of an articulated network of axial-force members. The method iterates until the maximum displacement limit is reached or the maximum number of iterations is exceeded. The function also provides a callback mechanism to monitor the progress of the optimization.

        References
        ==========
        Van Mele T., Mollaert M. and Block P.,
        Designing tension structures by integrating bending and stretching energy, in: Proceedings of the International Conference on Numerical Methods in Civil Engineering 2013, Istanbul, Turkey, 2013.
```







Rhino

```
from compas.datastructures import Mesh
from compas.datastructures import mesh_subdivide

from compas_rhino.artists import MeshArtist

mesh = Mesh.from_polyhedron(6)
subd = mesh_subdivide(mesh, k=3)

artist = MeshArtist(None)

artist.mesh = mesh
artist.draw_vertices()
artist.draw_edges()

artist.mesh = subd
artist.draw_faces()
```

Blender

```
from compas.datastructures import Mesh
from compas.datastructures import mesh_subdivide

from compas_blender.artists import MeshArtist

mesh = Mesh.from_polyhedron(6)
subd = mesh_subdivide(mesh, k=3)

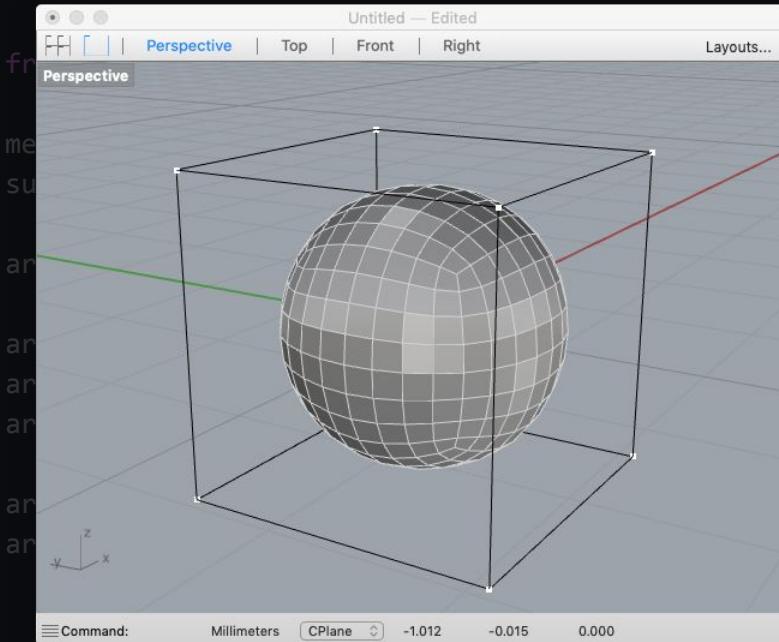
artist = MeshArtist(None)

artist.mesh = mesh
artist.draw_vertices()
artist.draw_edges()

artist.mesh = subd
artist.draw_faces()
```

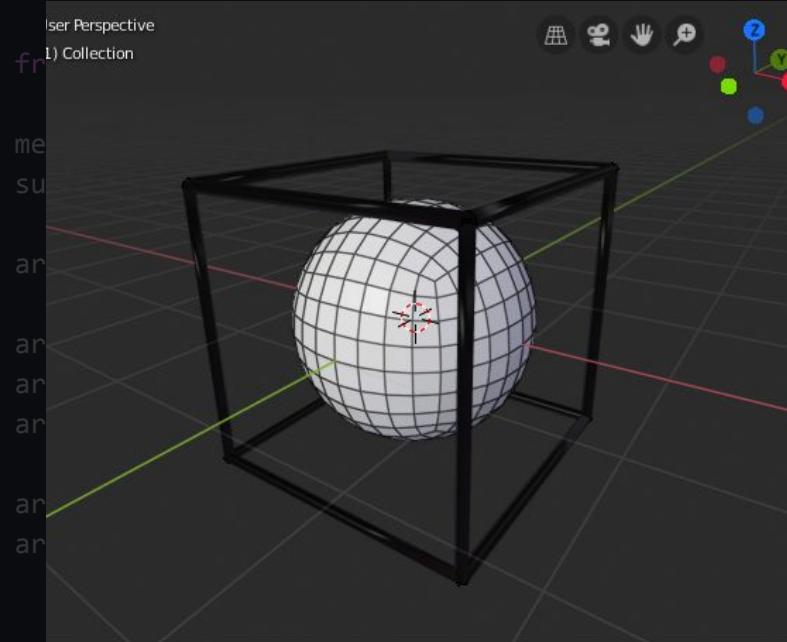
Rhino

```
from compas.datastructures import Mesh  
from compas.datastructures import mesh_subdivide
```



Blender

```
from compas.datastructures import Mesh  
from compas.datastructures import mesh_subdivide
```



Rhino

```
from compas.datastructures import Mesh
from compas.datastructures import mesh_subdivide

from compas_rhino.artists import MeshArtist

mesh = Mesh.from_polyhedron(6)
subd = mesh_subdivide(mesh, k=3)

artist = MeshArtist(None)

artist.mesh = mesh
artist.draw_vertices()
artist.draw_edges()

artist.mesh = subd
artist.draw_faces()
```

Blender

```
from compas.datastructures import Mesh
from compas.datastructures import mesh_subdivide

from compas_blender.artists import MeshArtist

mesh = Mesh.from_polyhedron(6)
subd = mesh_subdivide(mesh, k=3)

artist = MeshArtist(None)

artist.mesh = mesh
artist.draw_vertices()
artist.draw_edges()

artist.mesh = subd
artist.draw_faces()
```

Rhino

```
from compas.datastructures import Mesh
from compas.datastructures import mesh_subdivide

from compas_rhino.artists import MeshArtist

mesh = Mesh.from_polyhedron(6)
subd = mesh_subdivide(mesh, k=3)

artist = MeshArtist(None)

artist.mesh = mesh
artist.draw_vertices()
artist.draw_edges()

artist.mesh = subd
artist.draw_faces()
```

Blender

```
from compas.datastructures import Mesh
from compas.datastructures import mesh_subdivide

from compas_blender.artists import MeshArtist

mesh = Mesh.from_polyhedron(6)
subd = mesh_subdivide(mesh, k=3)

artist = MeshArtist(None)

artist.mesh = mesh
artist.draw_vertices()
artist.draw_edges()

artist.mesh = subd
artist.draw_faces()
```

COMPAS main — examples

Installation

Python 3.6.7 | packaged by conda-forge | (default, Jul 2 2019, 02:07:37)

[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)] on darwin

Type "help", "copyright", "credits" or "license" for more information.

>>>

Python 3.6.7 | packaged by conda-forge | (default, Jul 2 2019, 02:07:37)

[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)] on darwin

Type "help", "copyright", "credits" or "license" for more information.

>>>

```
>>> import compas
```

```
>>>
```

```
>>> import compas  
>>> compas.__version__  
'0.8.1'  
>>>
```

```
>>> import compas  
>>> compas.__version__  
'0.8.1'  
>>> exit()
```



download examples

<https://tiny.cc/ita-files>

BlockResearchGroup/ITA19-examples

GitHub, Inc. (US) https://github.com/BlockResearchGroup/ITA19-examples

Getting Started Most Visited algos Plotly pyTools Jupyter Sublime Text Rhino Sphinx Git(Hub) Python CVX... Funding Tools

Search or jump to... Pull requests Issues Marketplace Explore

BlockResearchGroup / ITA19-examples

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

Examples for the first session of the ITA PhD course Edit

Manage topics

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

brgcode initial examples Latest commit 65d8273 6 minutes ago

dynamic-relaxation initial examples 6 minutes ago

smoothing-on-surface initial examples 6 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

© 2019 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

https://github.com/BlockResearchGroup/ITA19-examples/releases

BlockResearchGroup/ITA19-examples

GitHub, Inc. (US) https://github.com/BlockResearchGroup/ITA19-examples

Getting Started Most Visited algos Plotly pyTools Jupyter Sublime Text Rhino Sphinx GitHub Python CVX... Funding Tools

Search or jump to... Pull requests Issues Marketplace Explore

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

Examples for the first session of the ITA PhD course Edit

Manage topics

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

brgcode initial examples Latest commit 65d8273 6 minutes ago

dynamic-relaxation initial examples 6 minutes ago

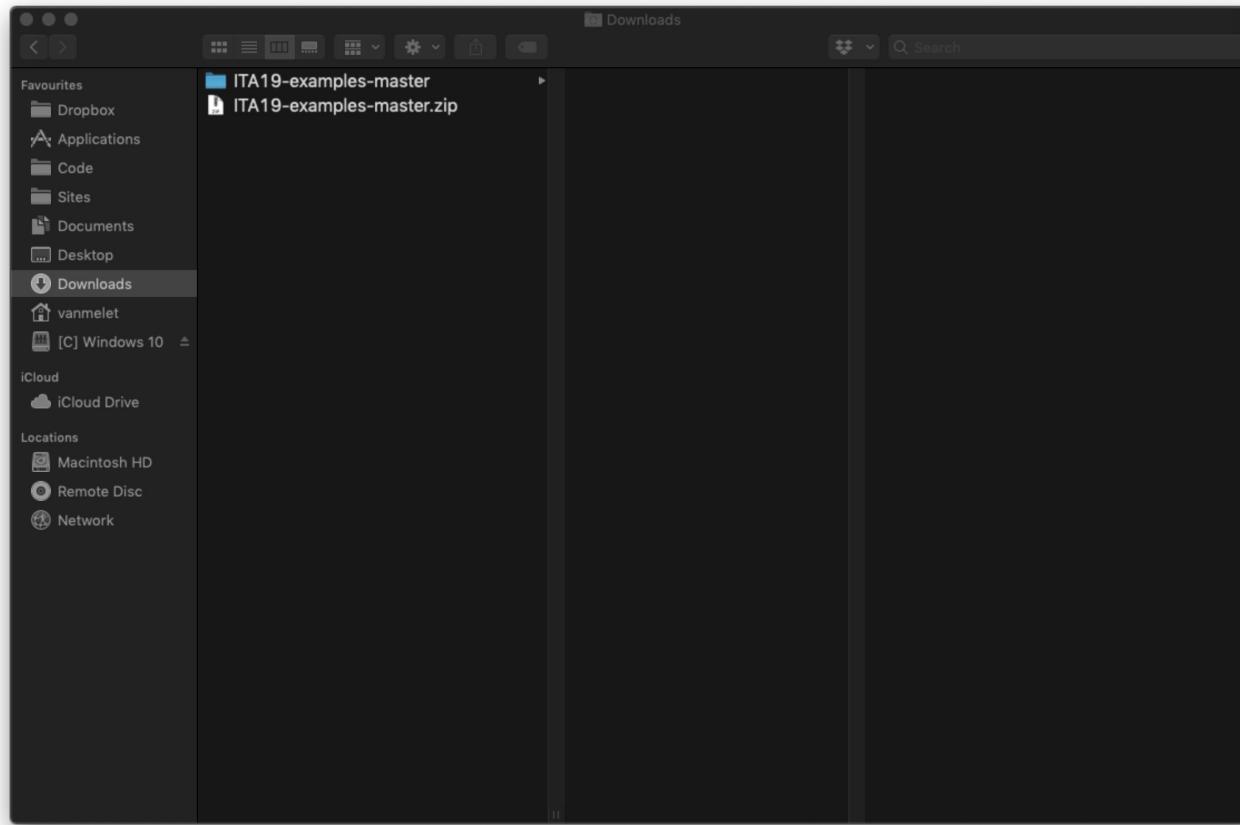
smoothing-on-surface initial examples 6 minutes ago

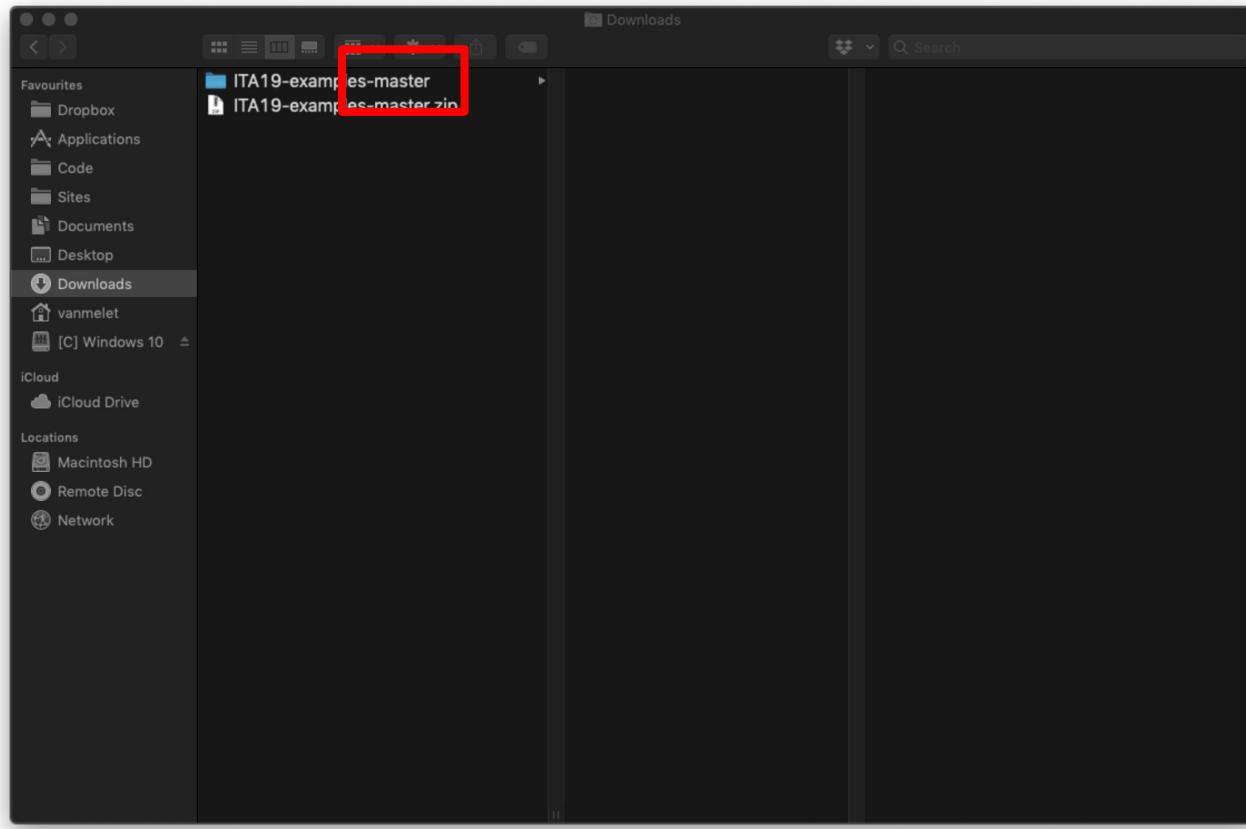
Help people interested in this repository understand your project by adding a README. Add a README

© 2019 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

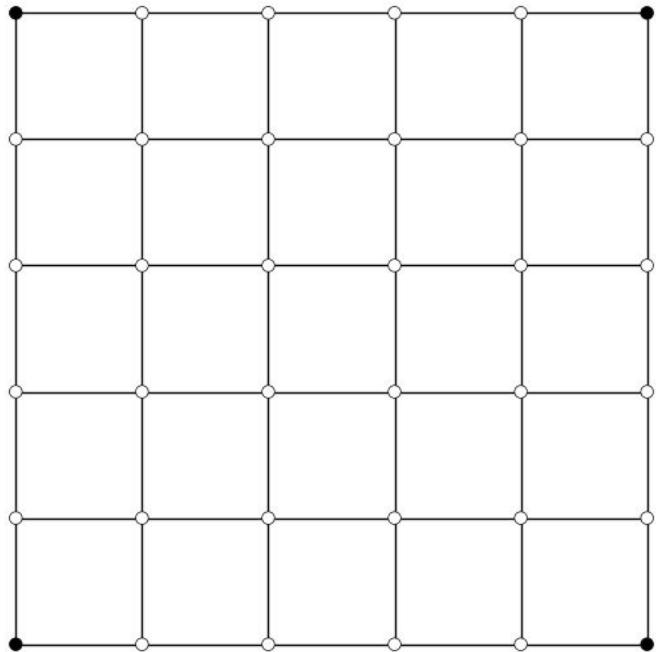
https://github.com/BlockResearchGroup/ITA19-examples/releases

The screenshot shows a GitHub repository page for 'BlockResearchGroup/ITA19-examples'. The repository has 2 commits, 1 branch, 0 releases, and 1 contributor. The 'Clone or download' button is highlighted with a red box. The page includes sections for managing topics, adding a README, and links to GitHub's terms of service, privacy policy, and other resources.

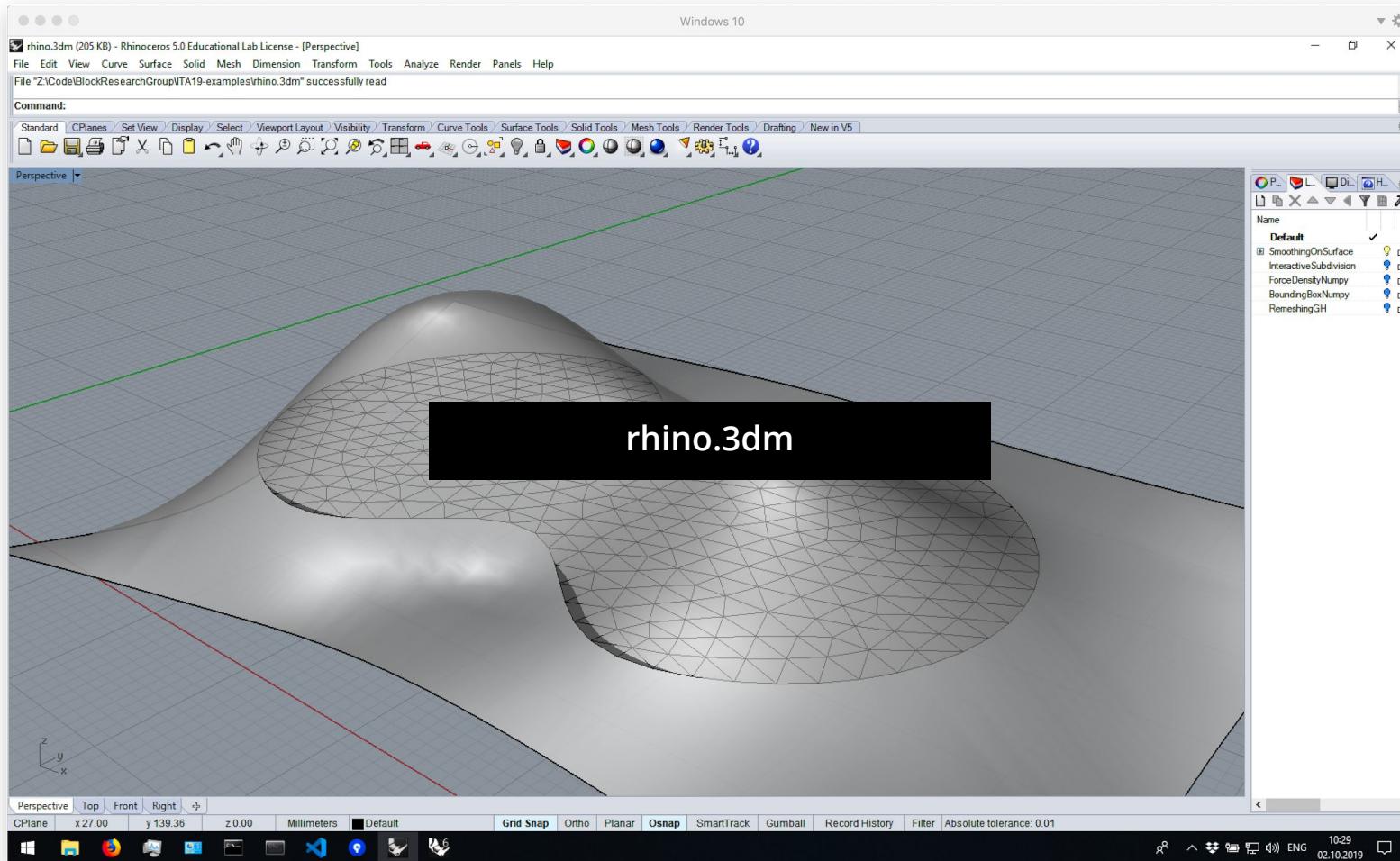


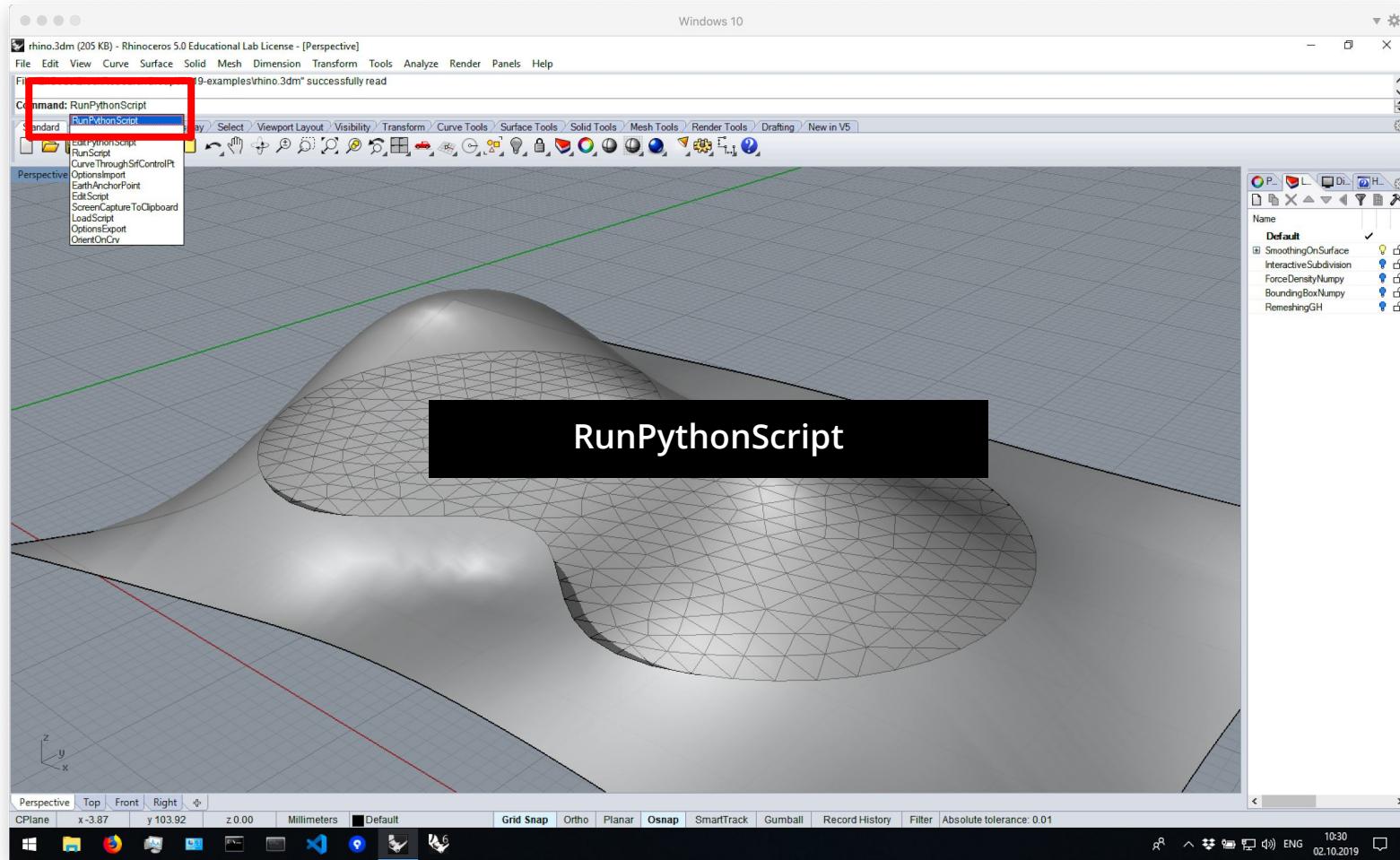


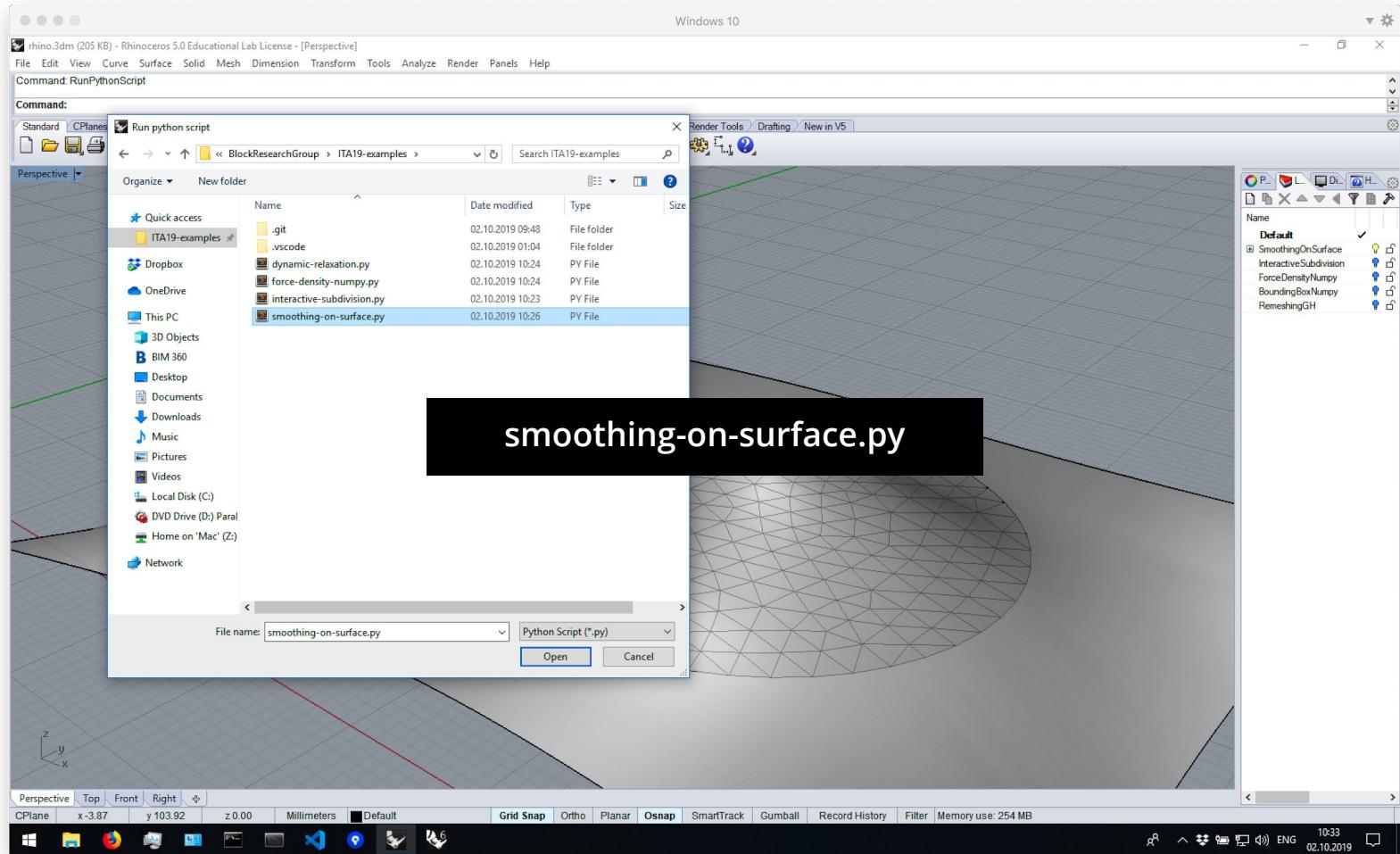
dynamic relaxation

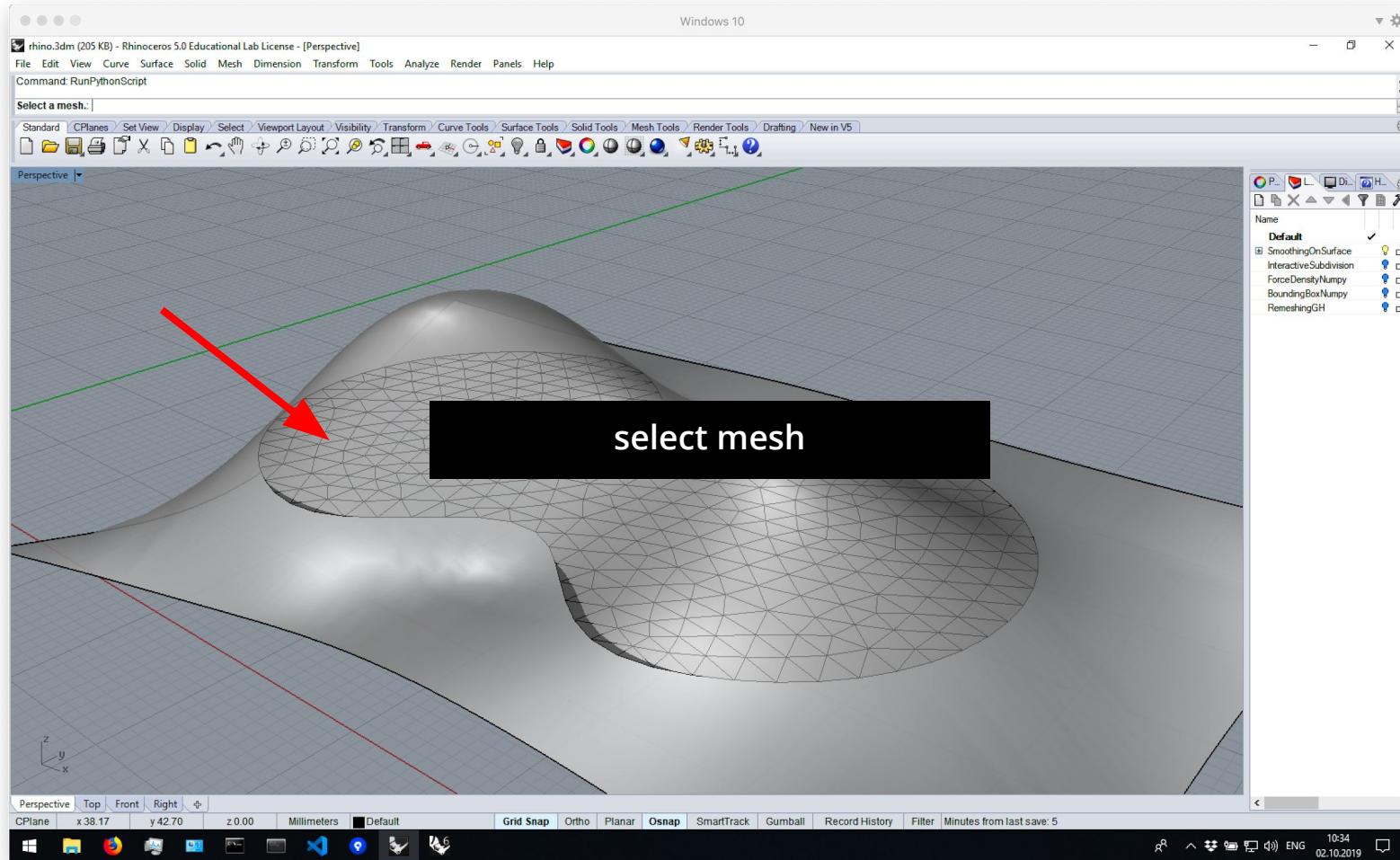


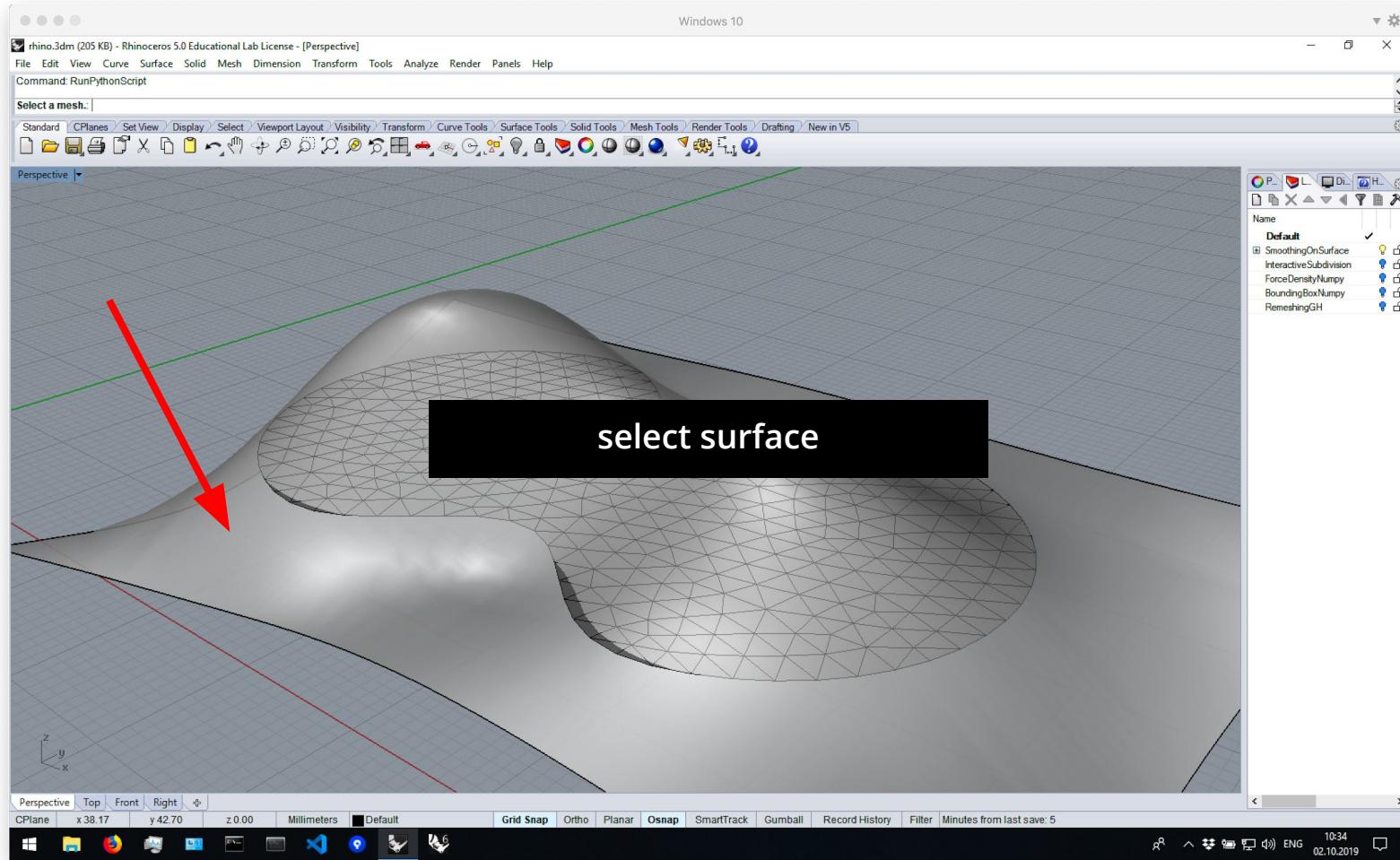
rhino - smoothing on surface

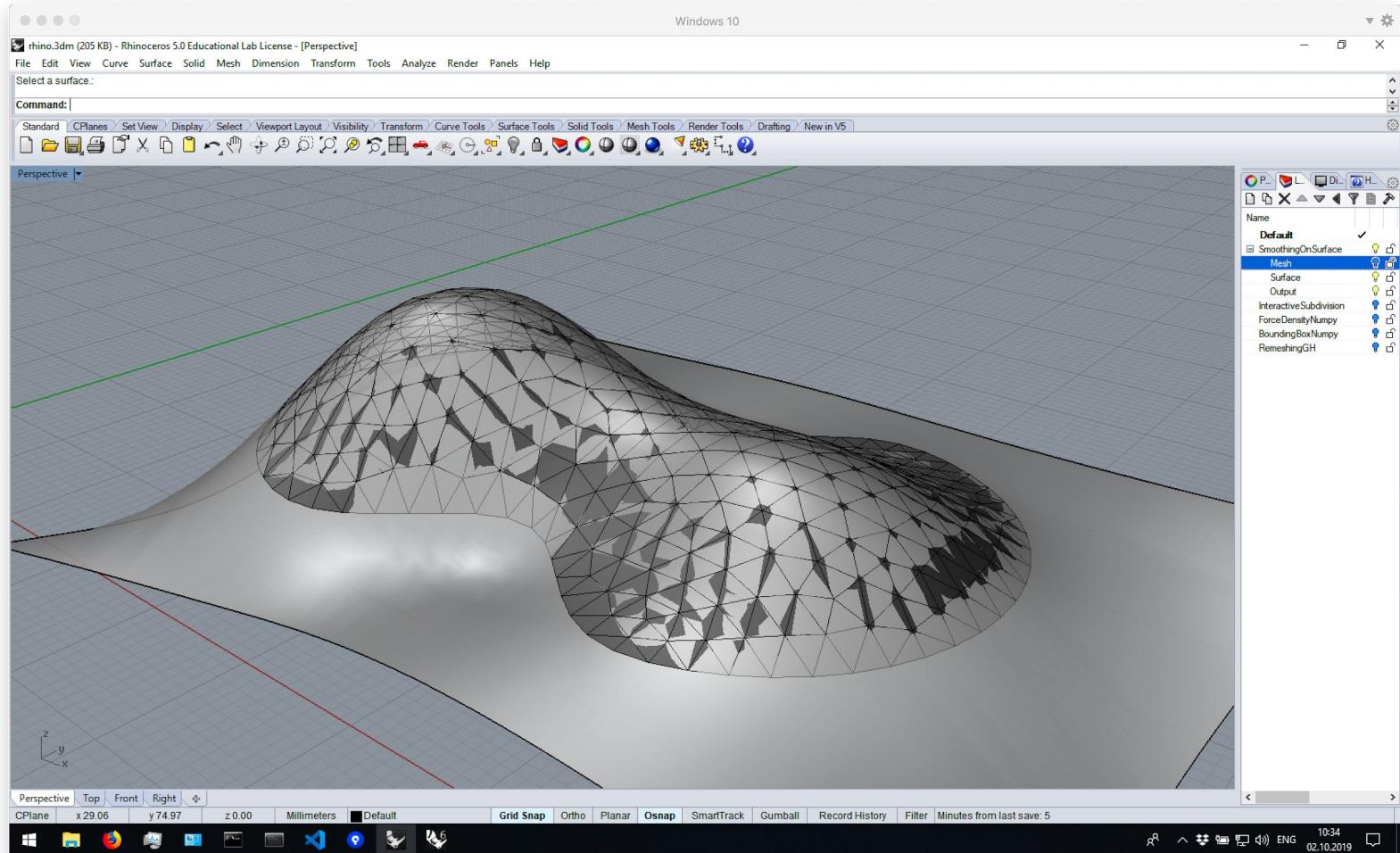




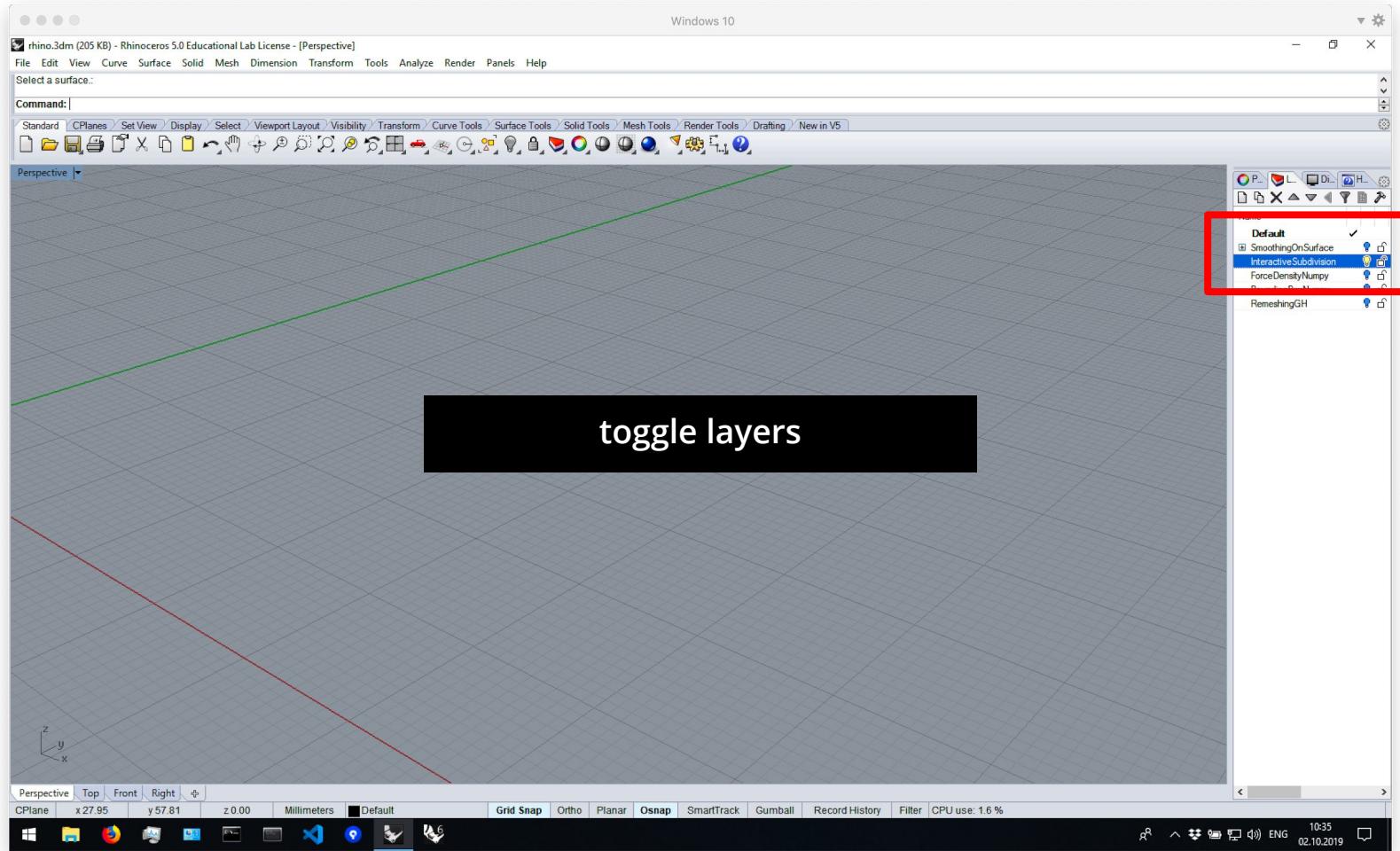


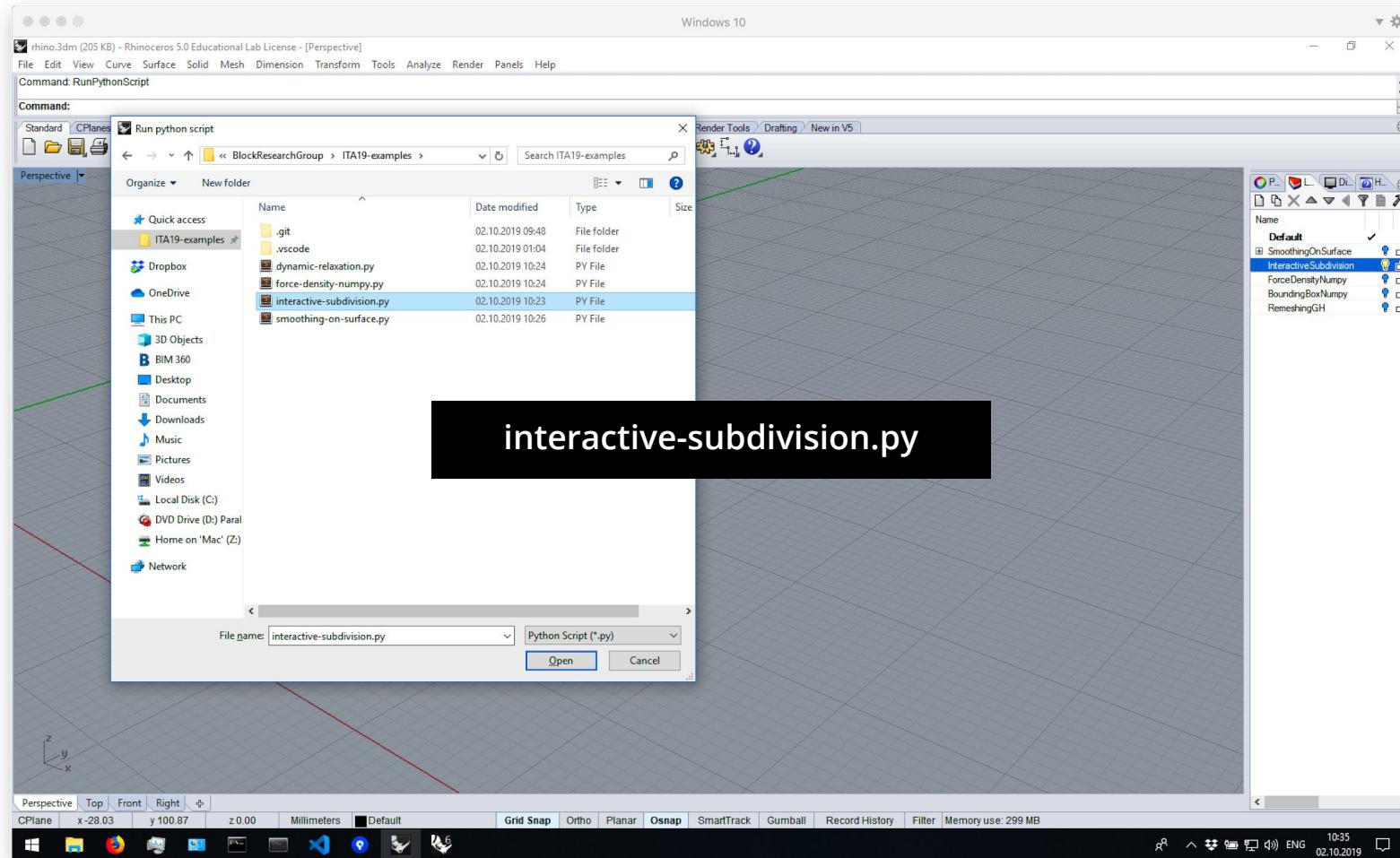


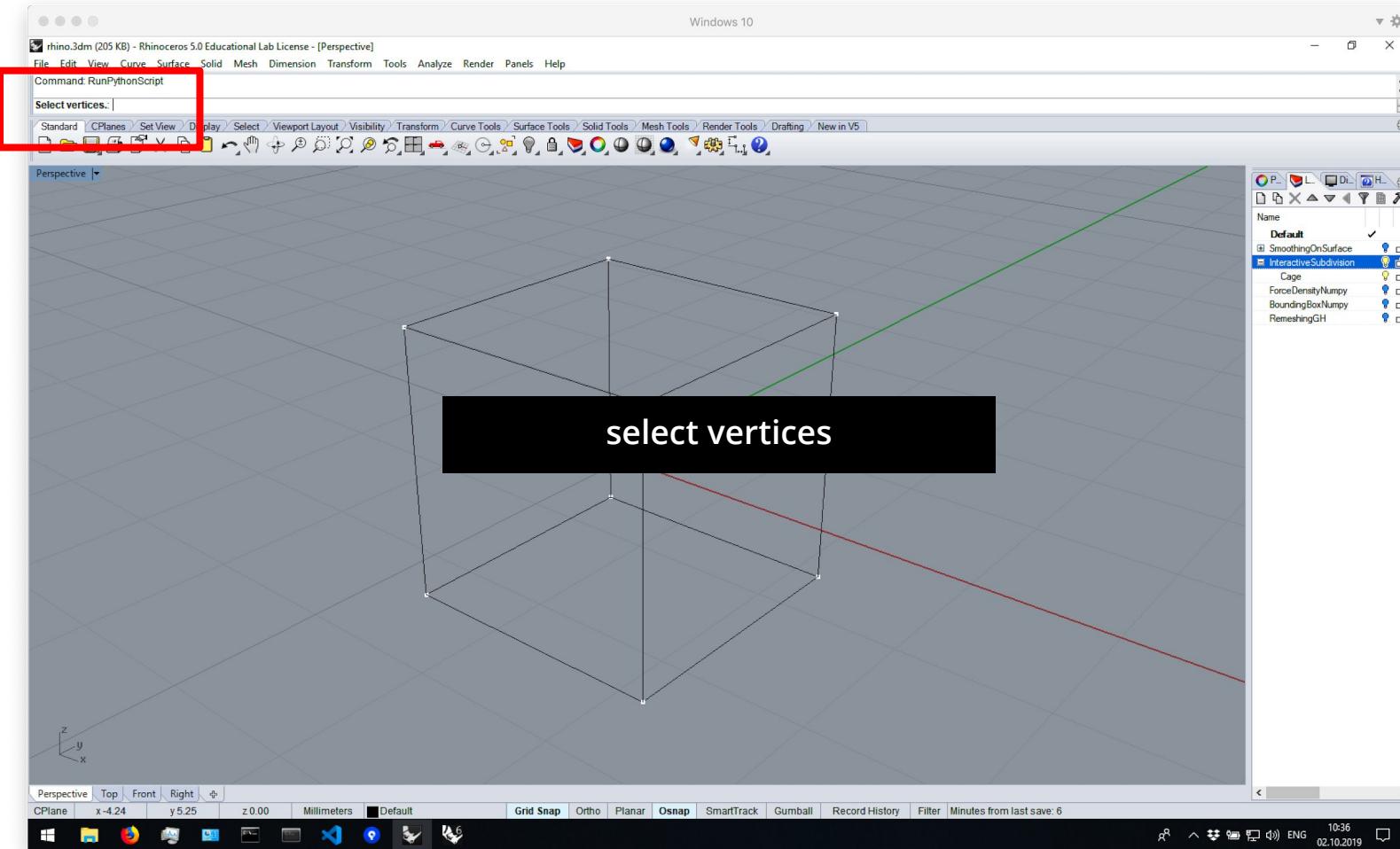


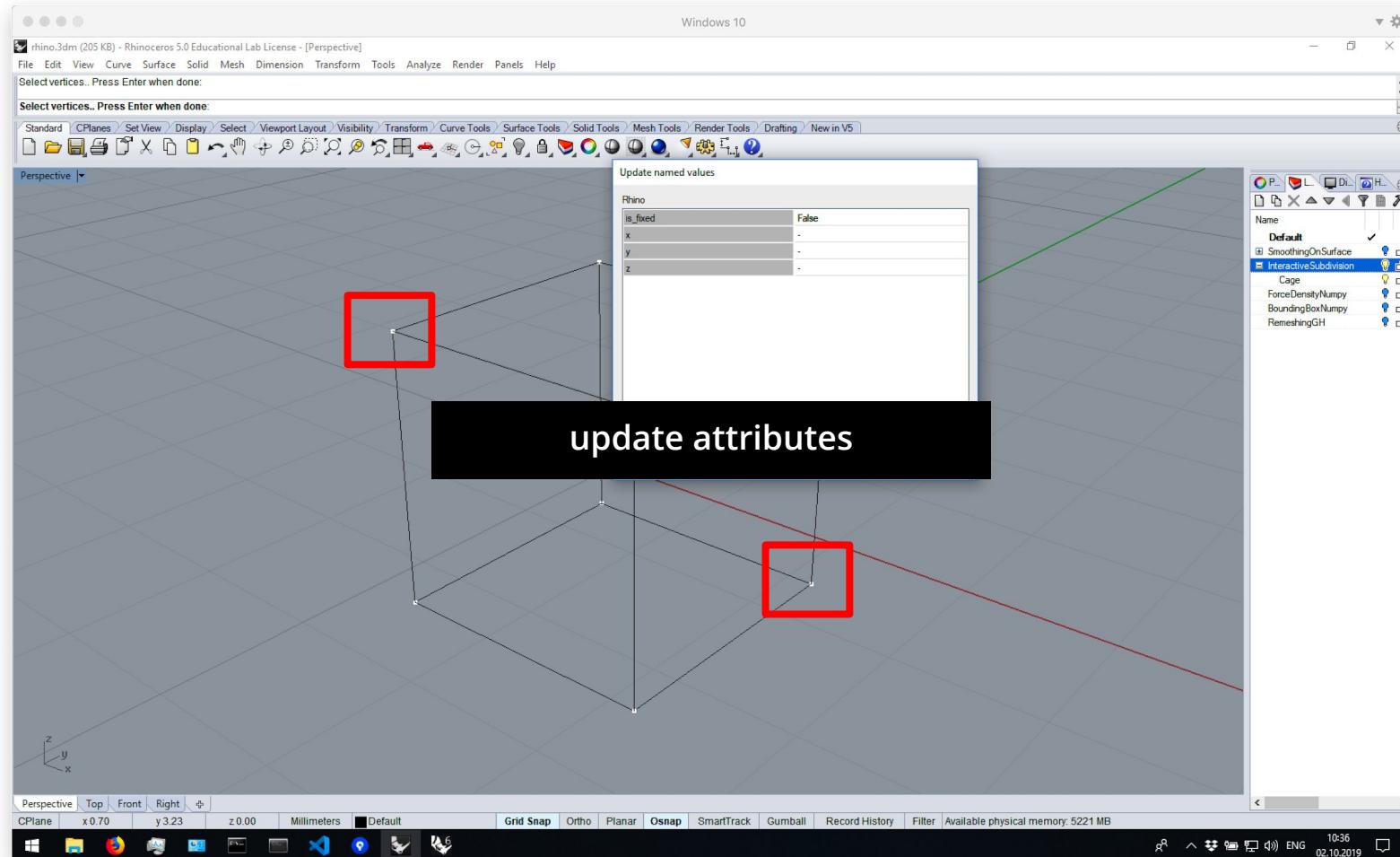


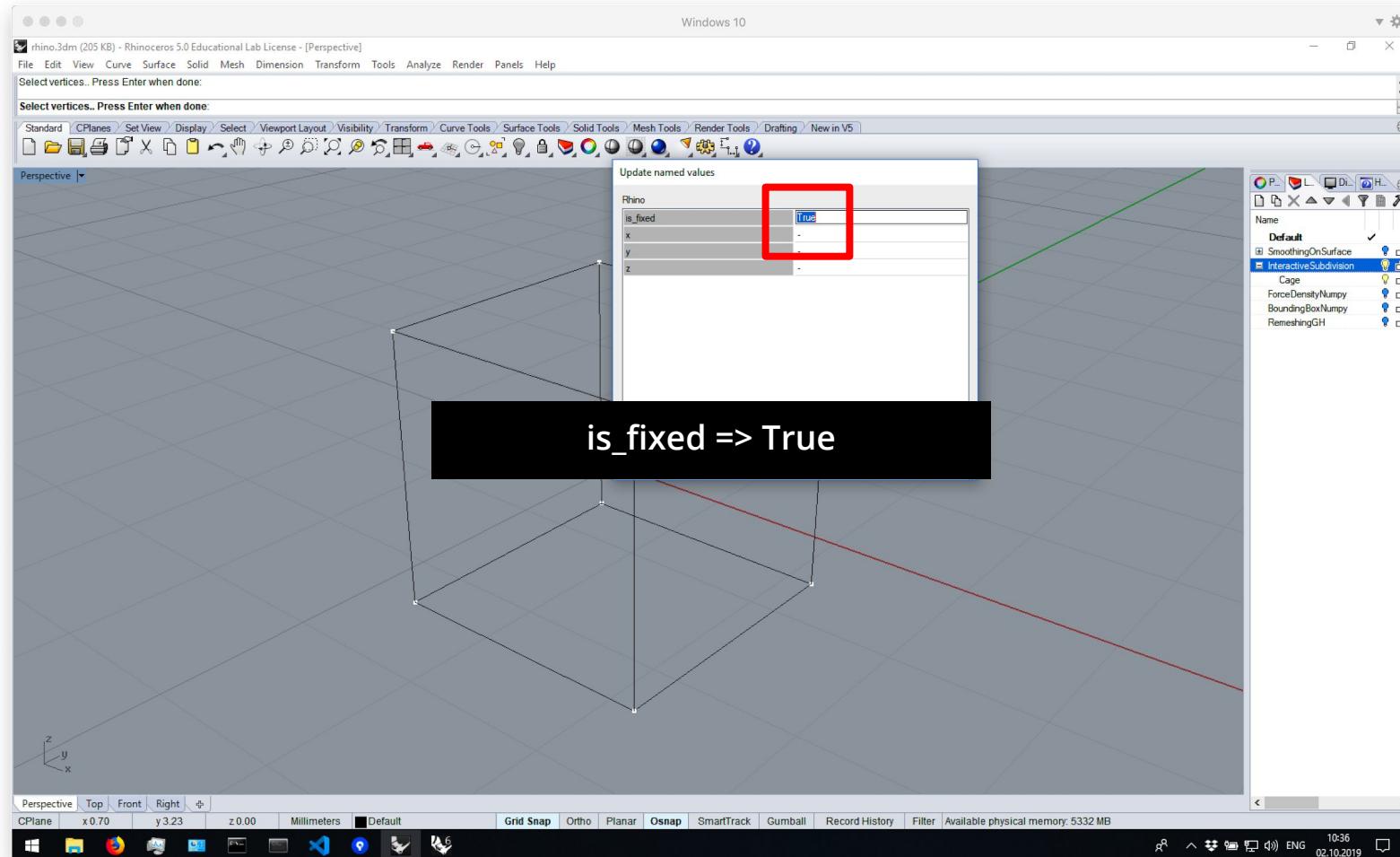
rhino - interactive subdivision

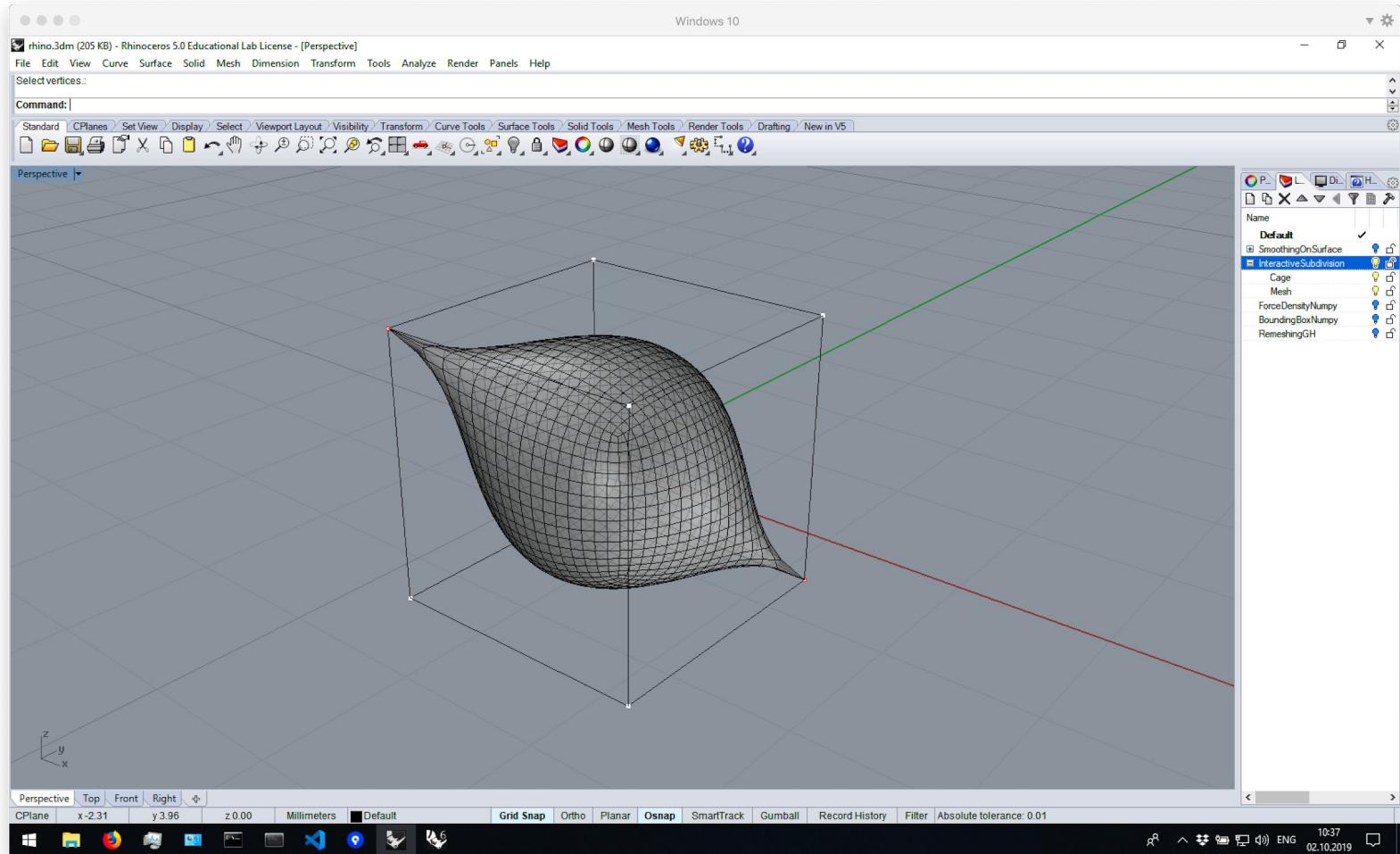




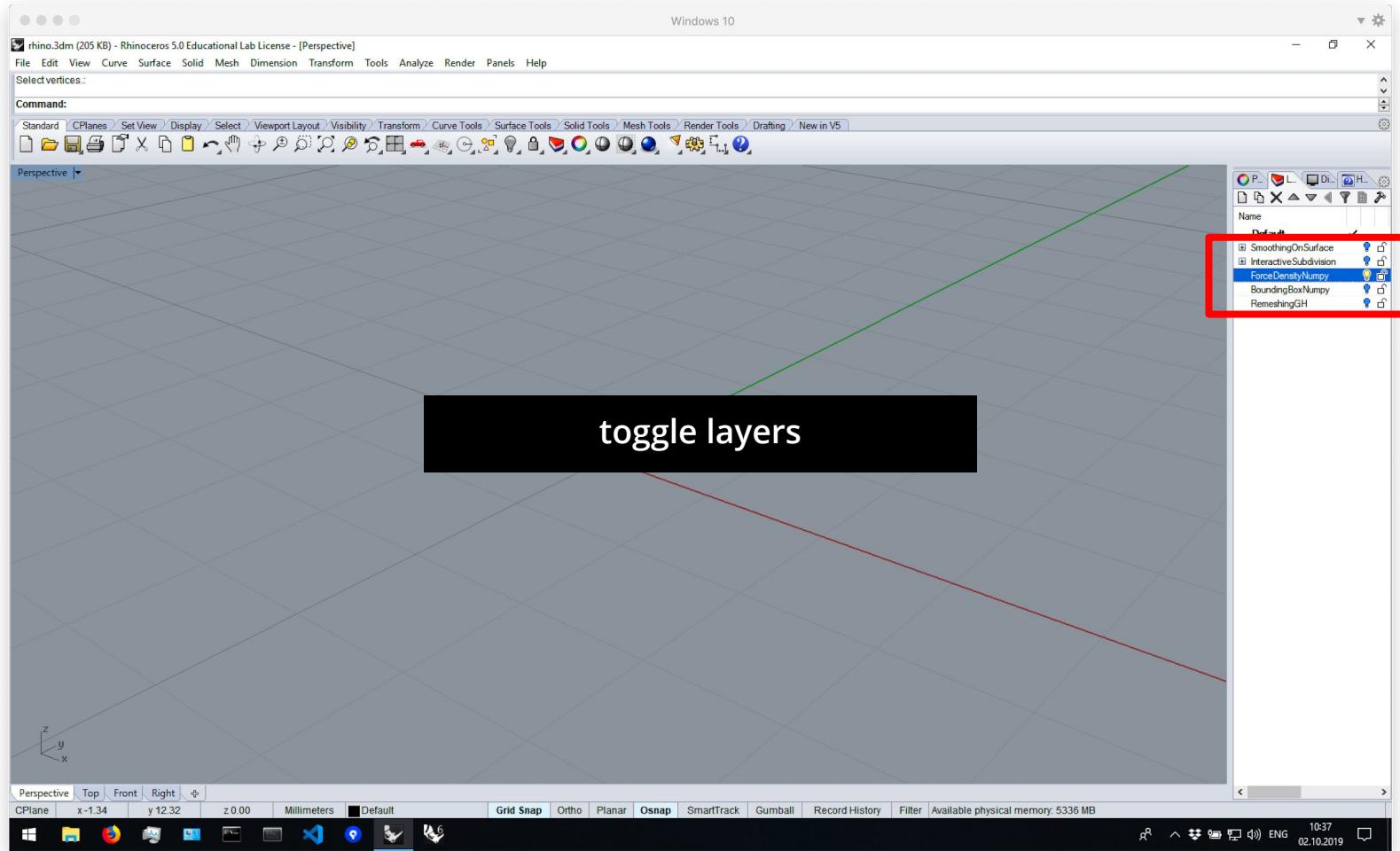


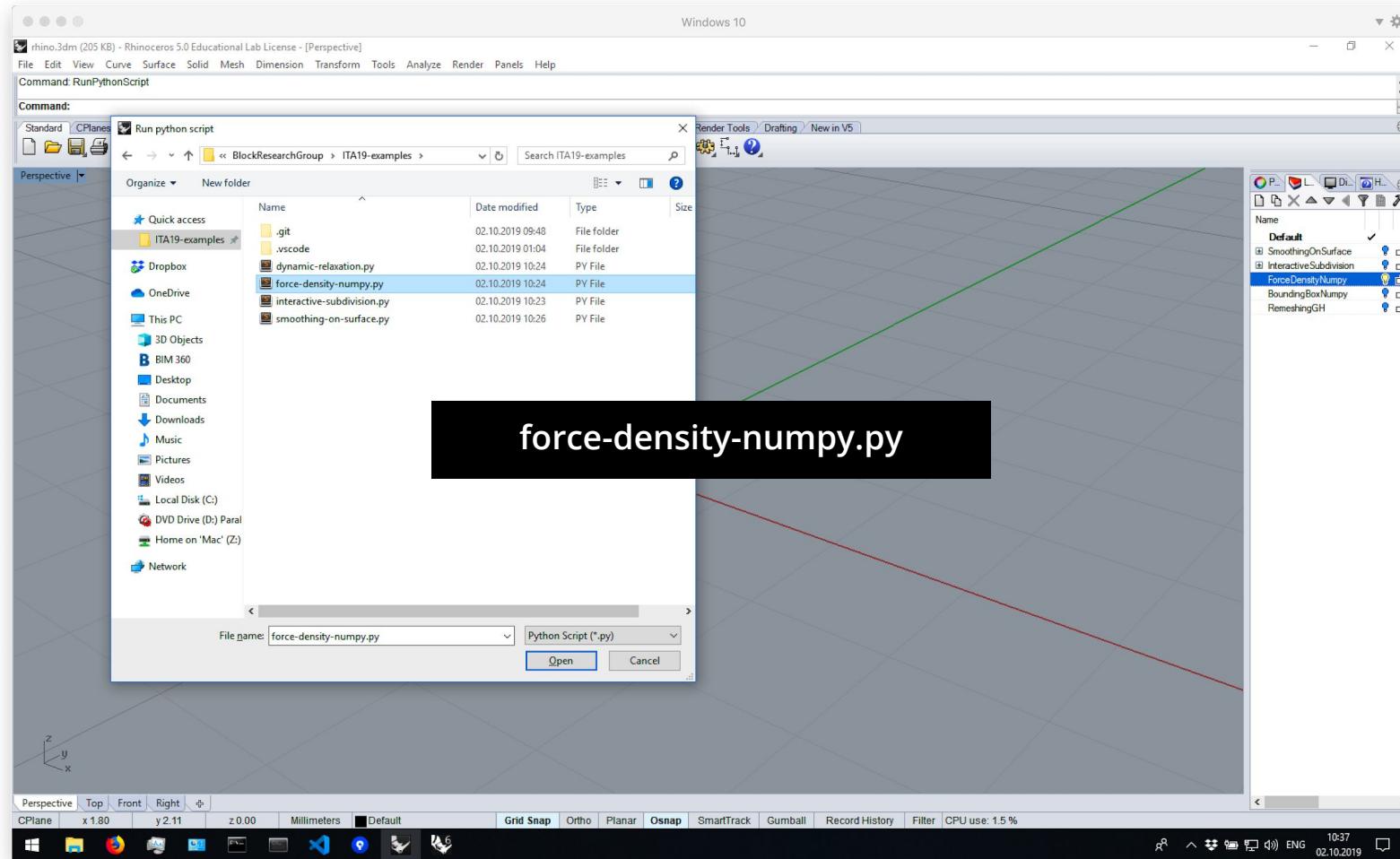


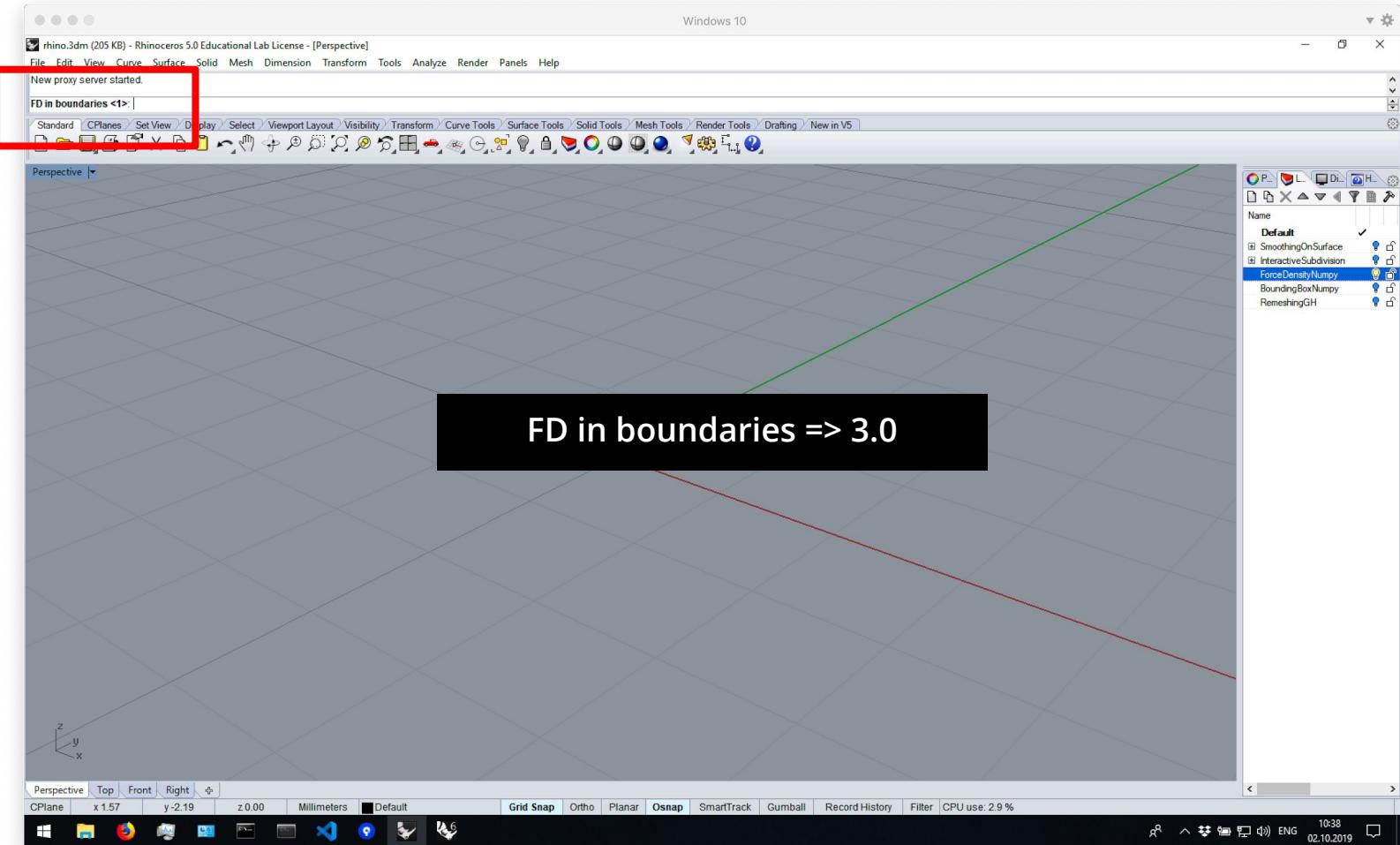


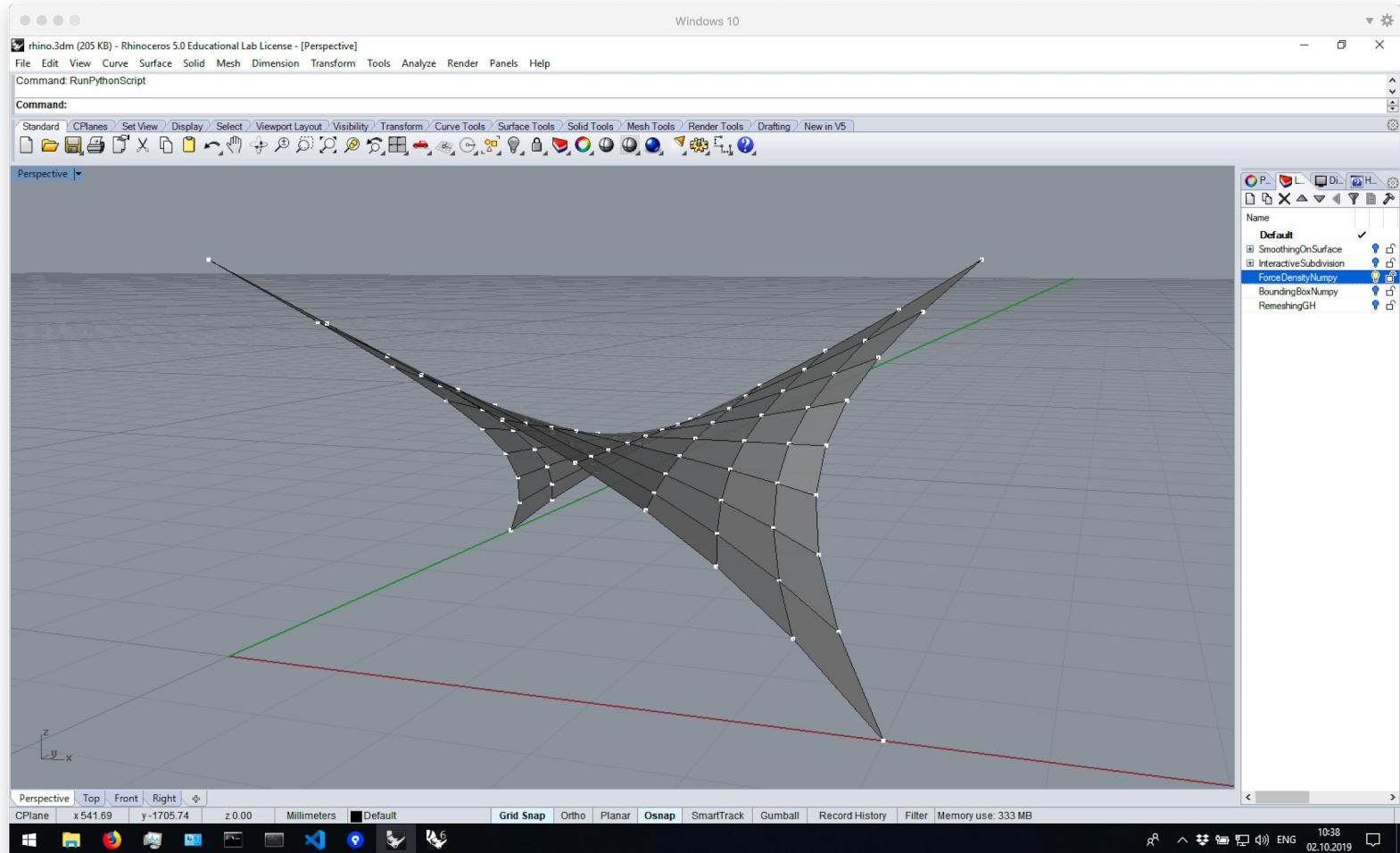


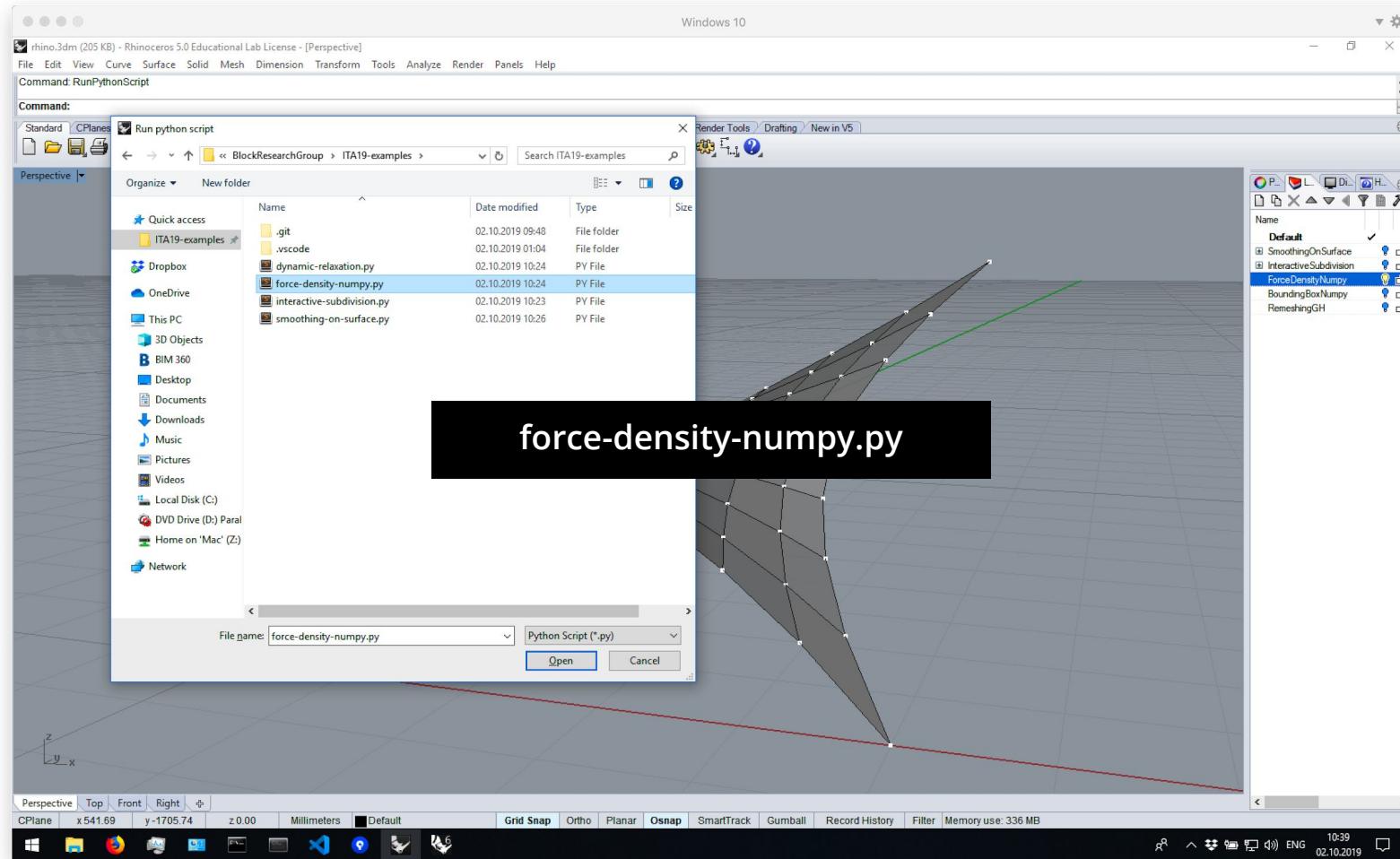
rhino - force density numpy

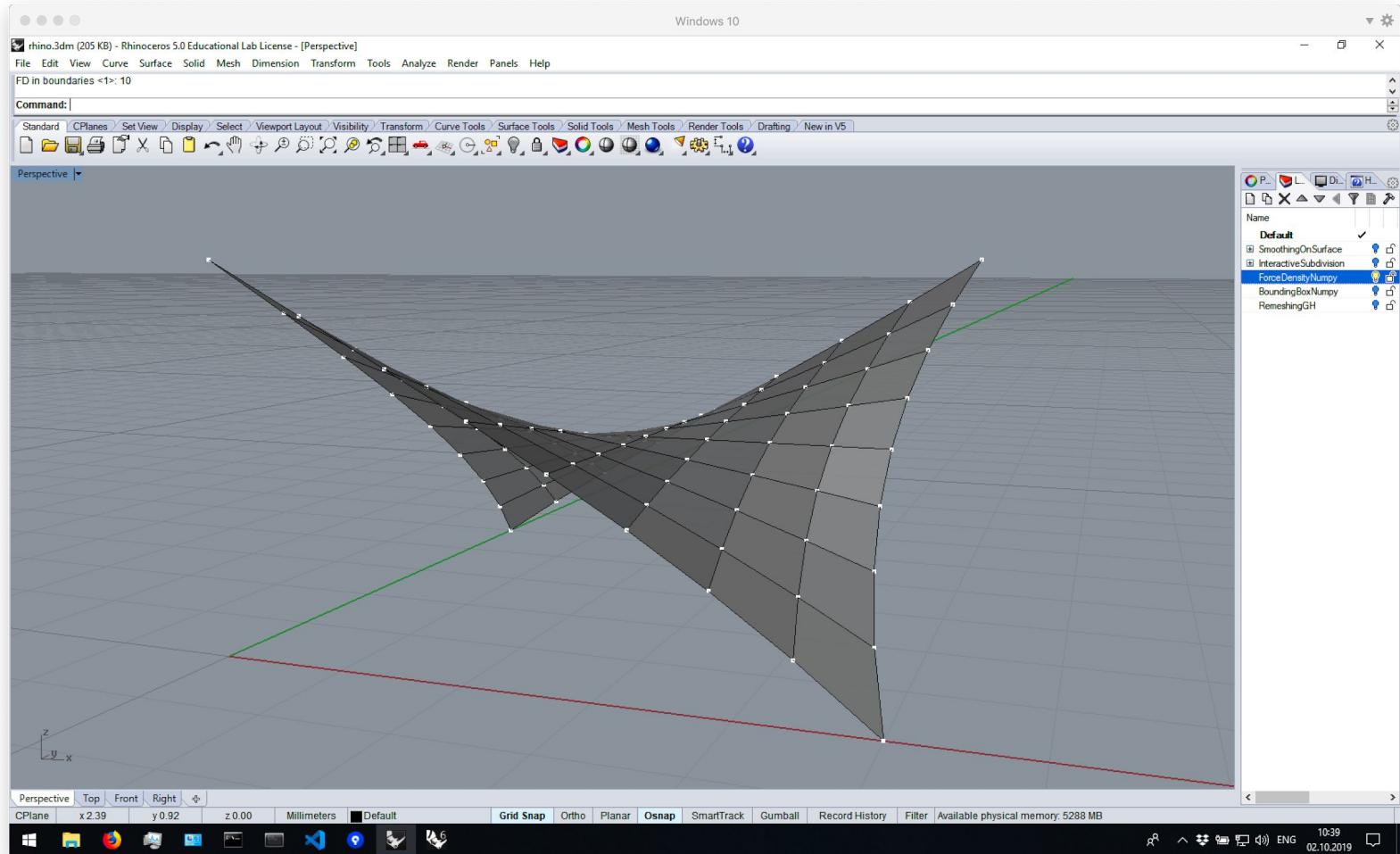


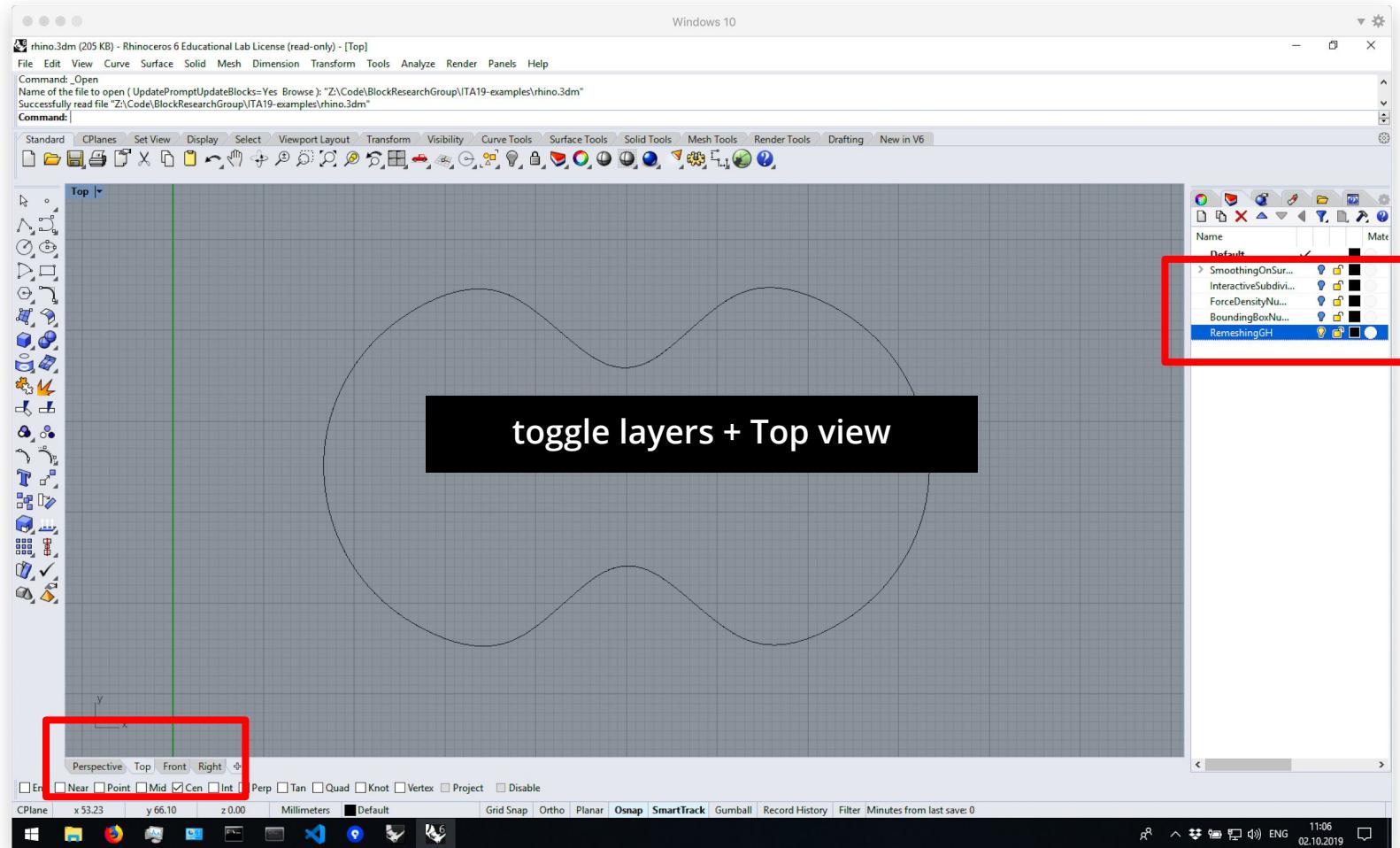


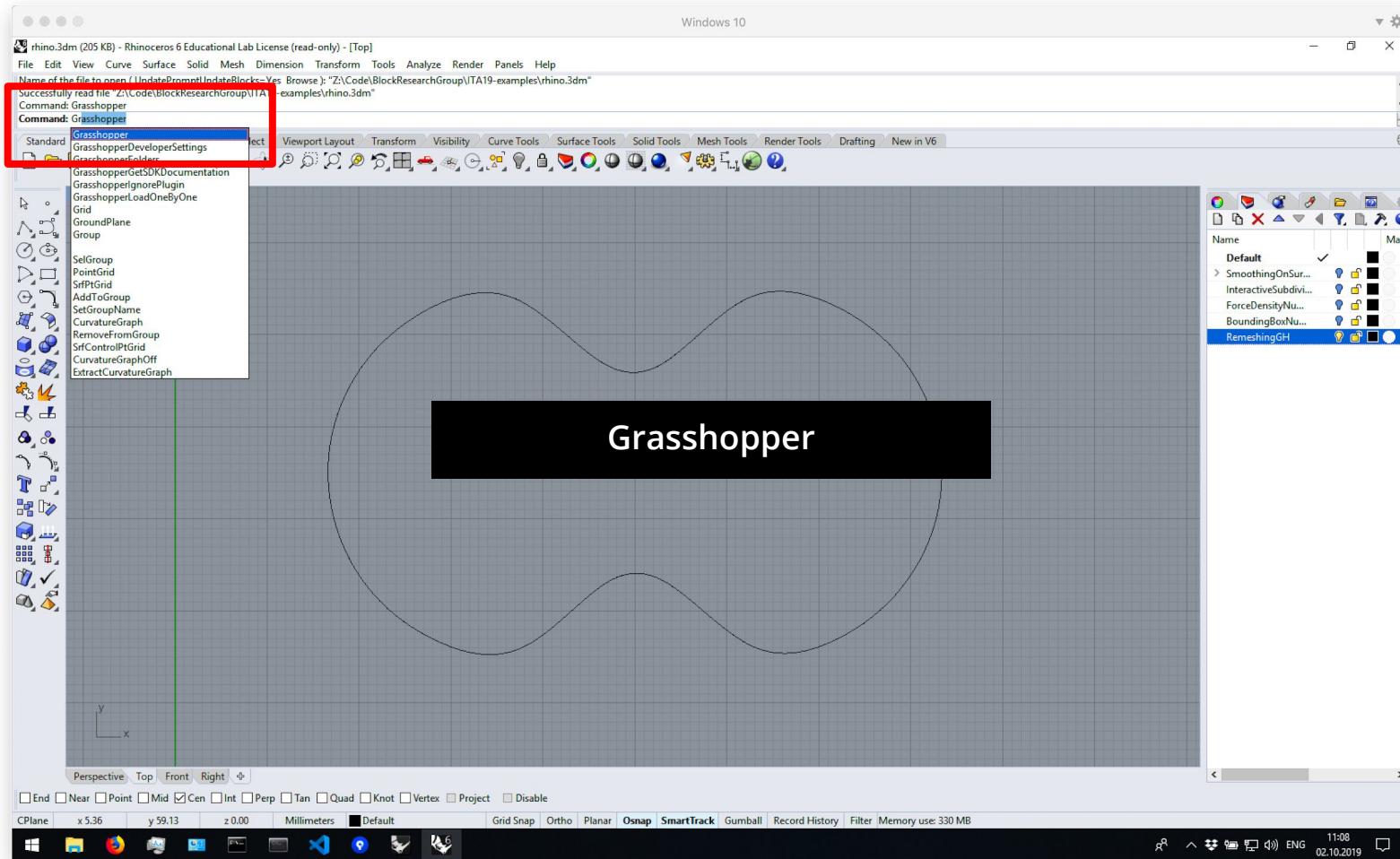


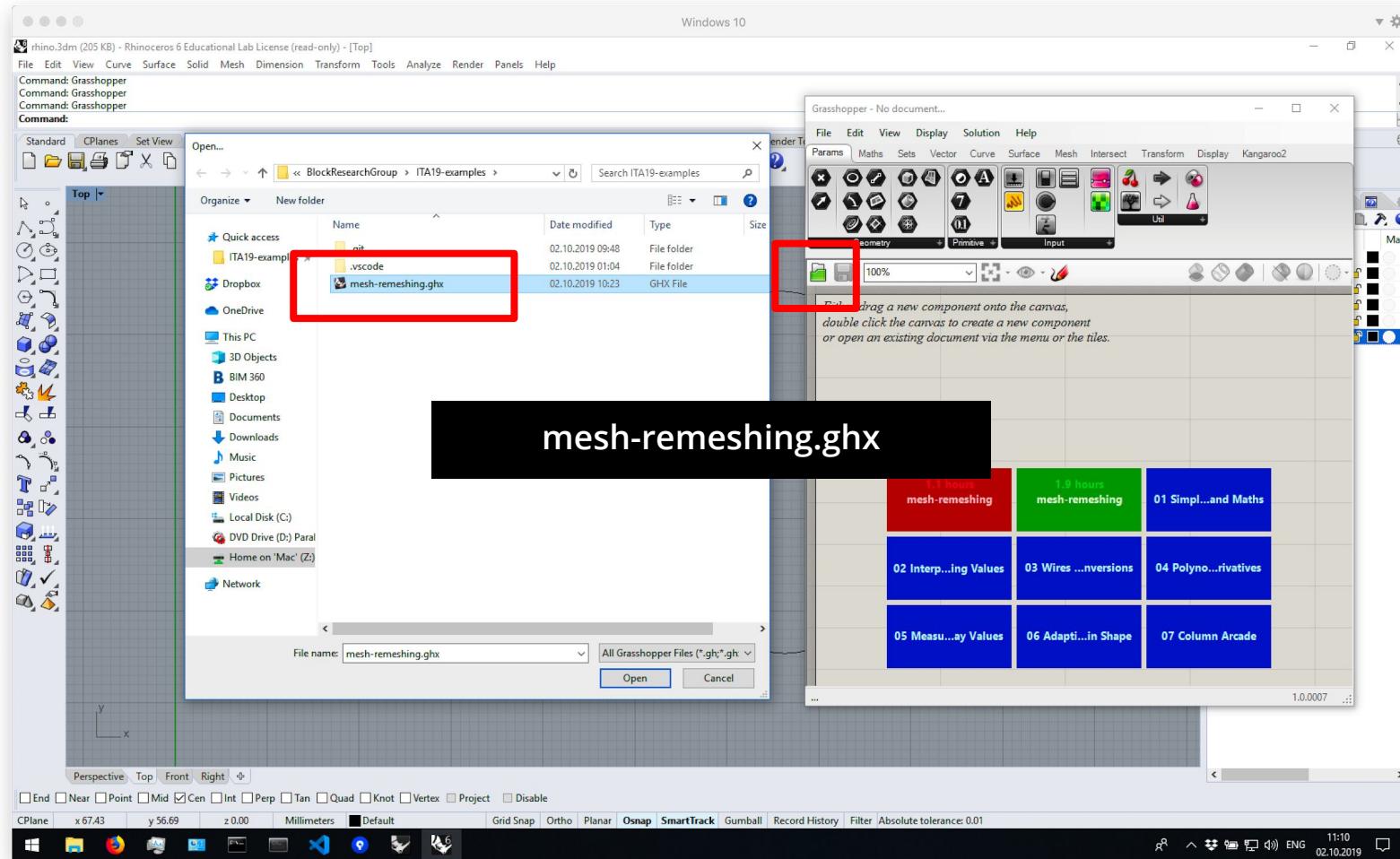


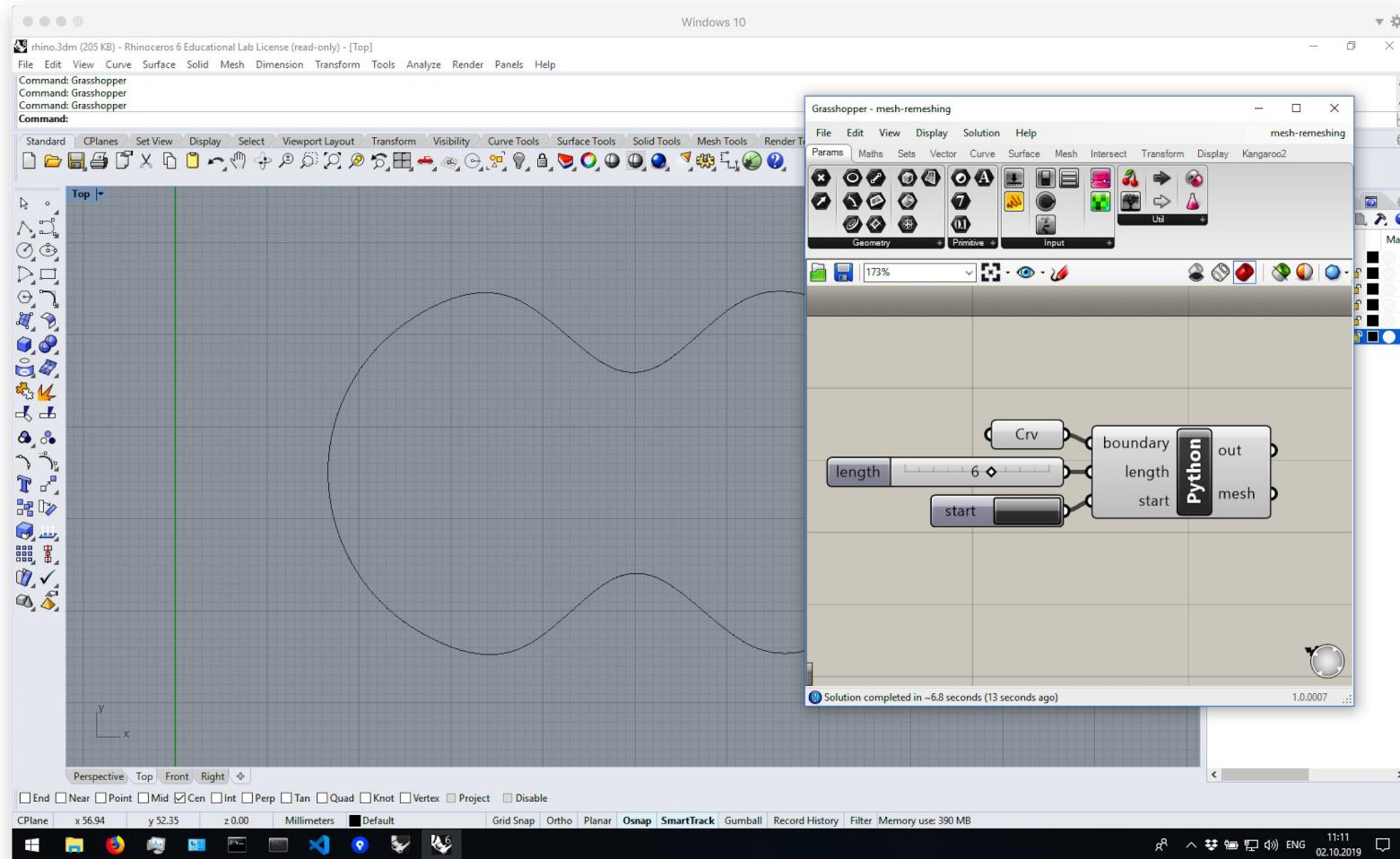


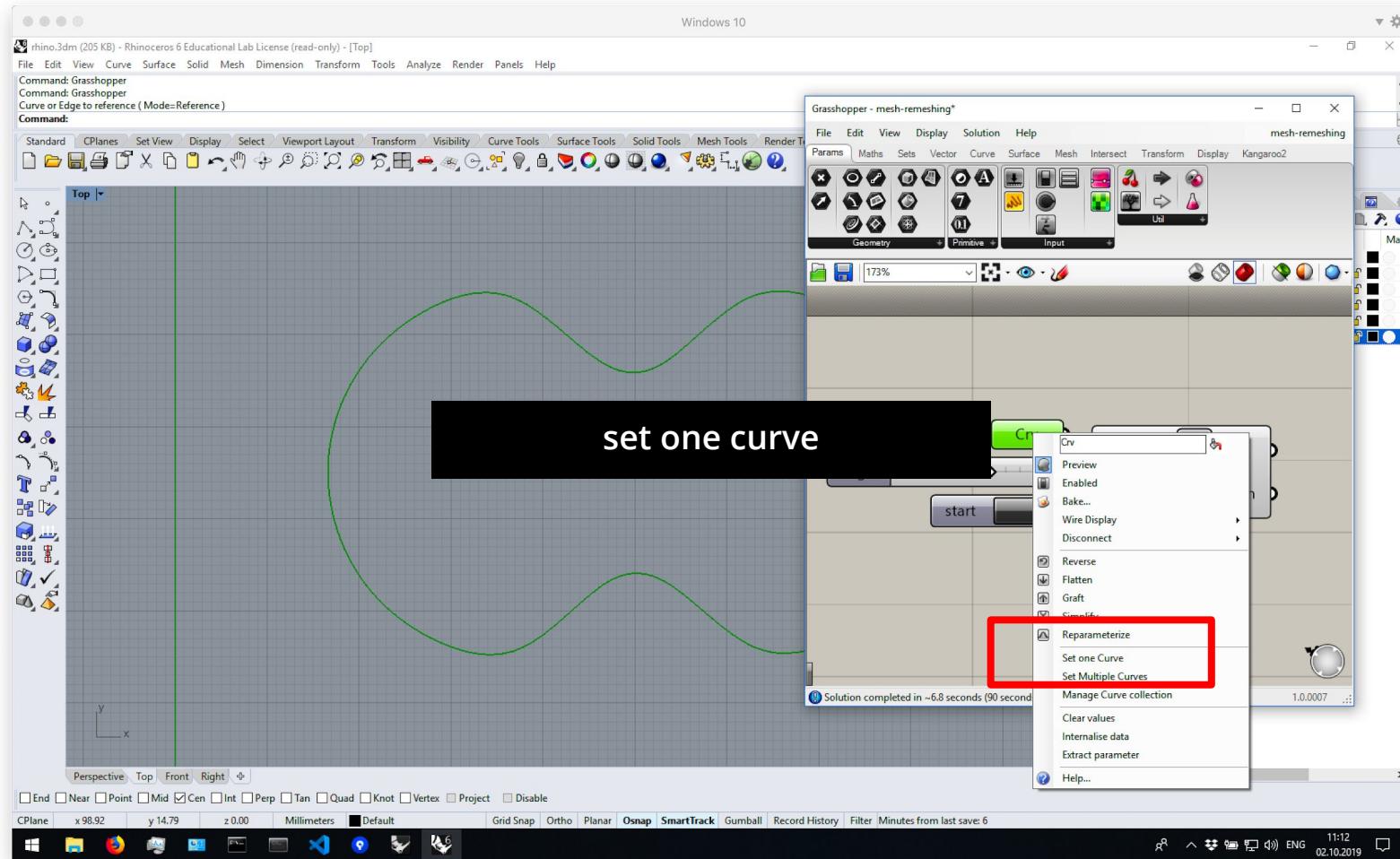


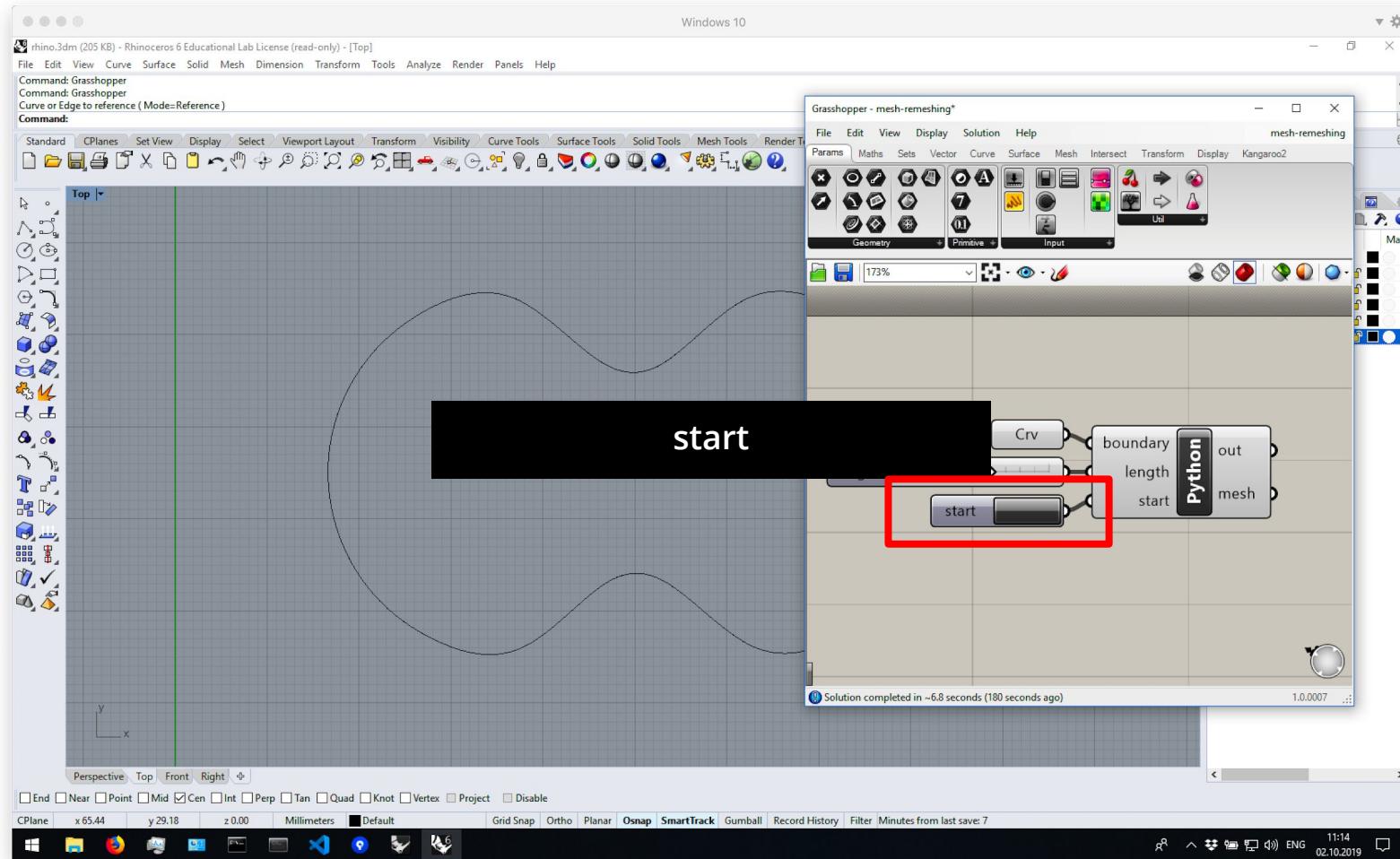


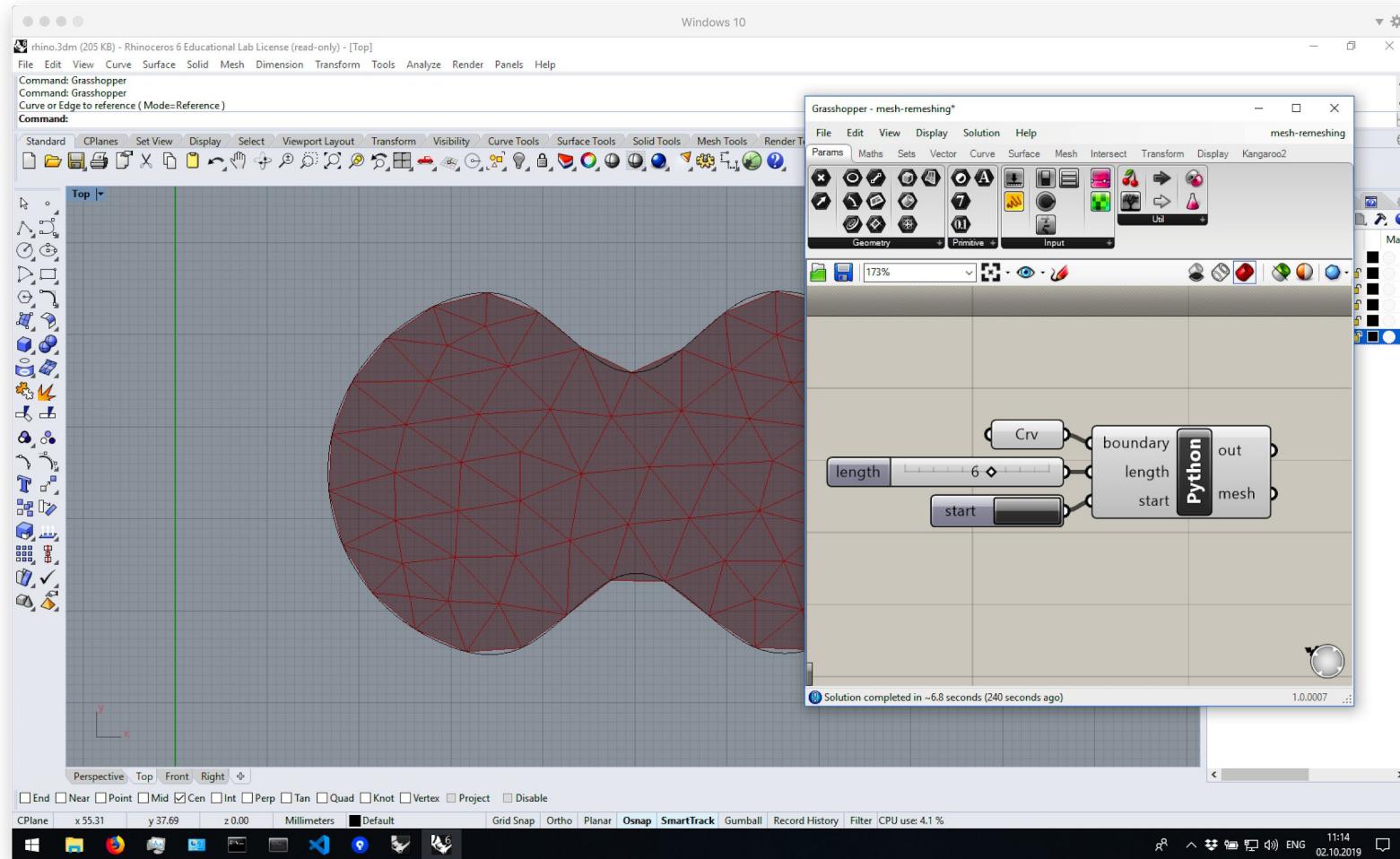


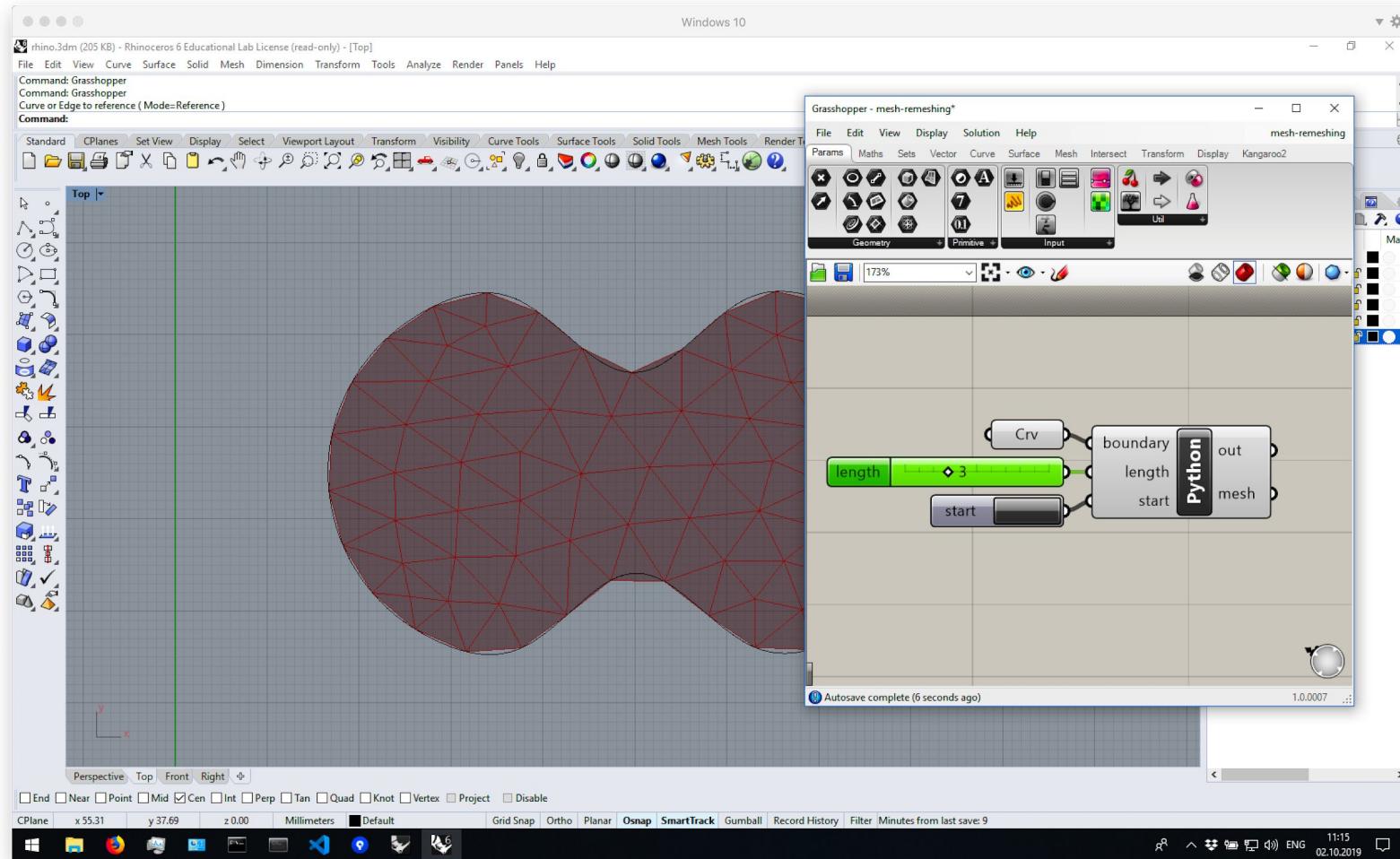


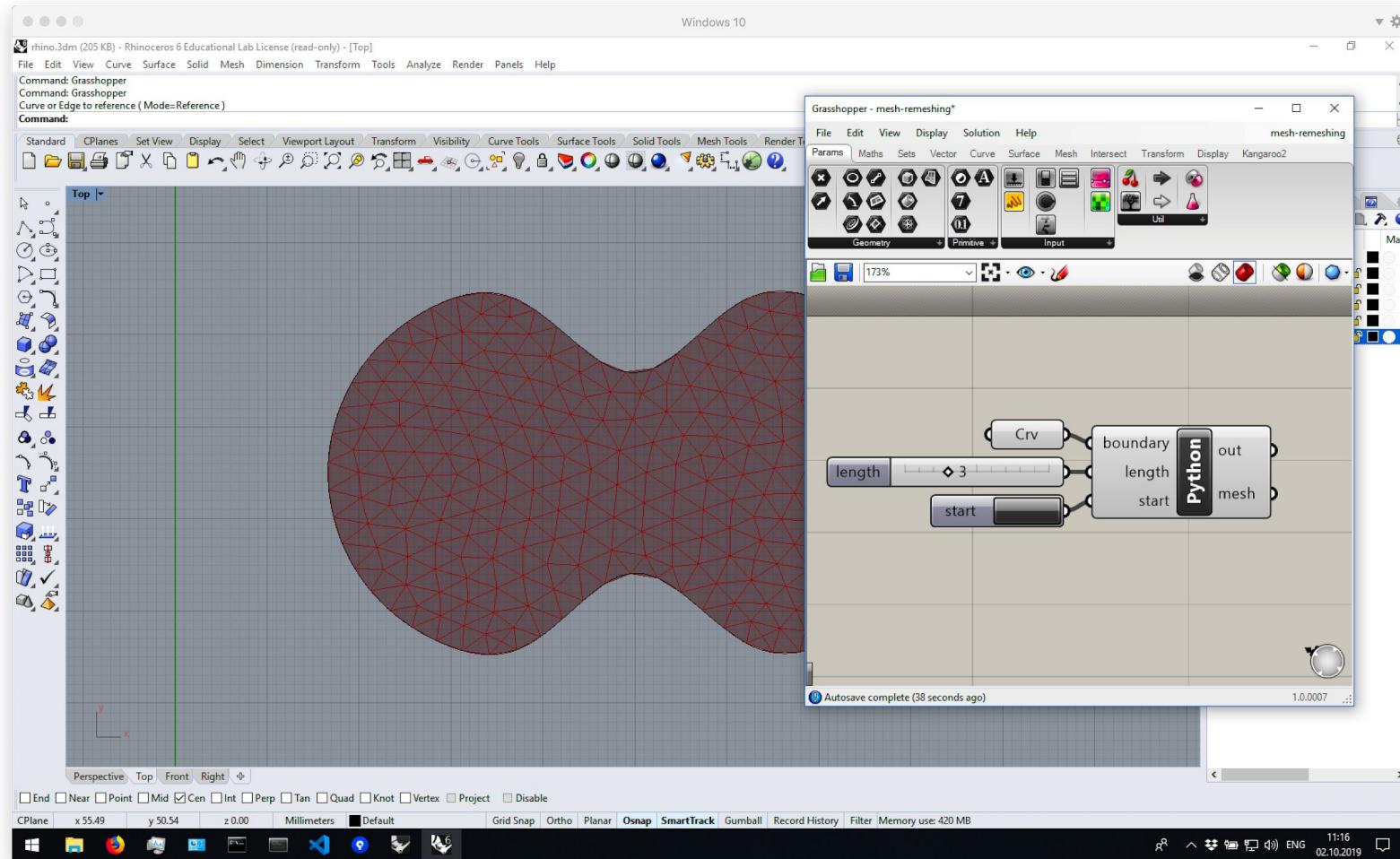




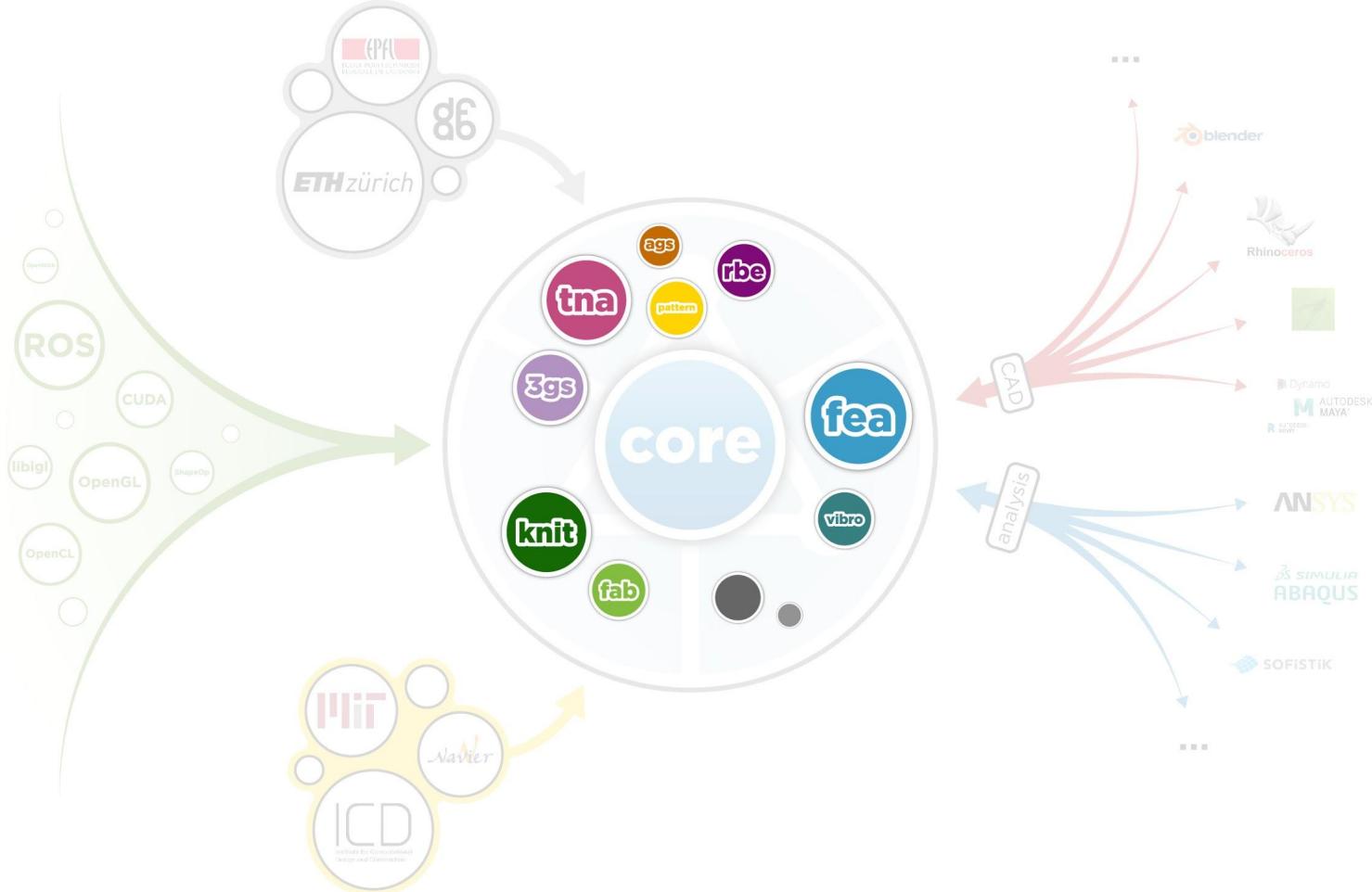








COMPAS packages



compas_agS

Algebraic Graph
Statics

compas_tna

Thrust Network
Analysis

compas_3gs

3D Graphic Statics

compas_fea

Finite Element
Analysis

compas_fab

Robotic Fabrication

compas_vol

Volumetric modelling

compas_knit

Spatial Knitting

compas_dem

Discrete Element
Modeling (3DEC)

compas_dr6

6DOF Dynamic
Relaxation

compas_rbe

Rigid Block
Equilibrium

compas_ml

Machine Learning

compas_fofin

Form Finding

compas_rocking

Collapse Mechanisms

compas_assembly

Discrete element
assemblies

compas_pattern

Mesh Topology
Finding

compas_ag

Algebraic Graph
Statics

compas_tna

Thrust Network
Analysis

compas_3gs

3D Graphic Statics

compas_fea

Finite Element
Analysis

compas_fab

Robotic Fabrication

compas_vol

Volumetric modelling

compas_knit

Spatial Knitting

compas_dem

Discrete Element
Modeling (3DEC)

compas_dr6

6DOF Dynamic
Relaxation

compas_rbe

Rigid Block
Equilibrium

compas_ml

Machine Learning

compas_fofin

Form Finding

compas_rocking

Collapse Mechanisms

compas_ctrl

Active On-site Control

compas_pattern

Mesh Topology
Finding

compas_ag

Algebraic Graph
Statics

compas_tna

Thrust Network
Analysis

compas_3gs

3D Graphic Statics

compas_fea

Finite Element
Analysis

compas_fab

Robotic Fabrication

compas_vol

Volumetric modelling

compas_knit

Spatial Knitting

compas_dem

Discrete Element
Modeling (3DEC)

compas_dr6

6DOF Dynamic
Relaxation

compas_rbe

Rigid Block
Equilibrium

compas_ml

Machine Learning

compas_fofin

Form Finding

compas_rocking

Collapse Mechanisms

compas_ctrl

Active On-site Control

compas_pattern

Mesh Topology
Finding

compas_agc

Algebraic Graph
Statics

compas_tna

Thrust Network
Analysis

compas_3gs

3D Graphic Statics

compas_fea

Finite Element
Analysis

compas_fab

Robotic Fabrication

compas_vol

Volumetric modelling

compas_knit

Spatial Knitting

compas_dem

Discrete Element
Modeling (3DEC)

compas_dr6

6DOF Dynamic
Relaxation

compas_rbe

Rigid Block
Equilibrium

compas_ml

Machine Learning

compas_fofin

Form Finding

compas_rocking

Collapse Mechanisms

compas_ctrl

Active On-site Control

compas_pattern

Mesh Topology
Finding

compas_agS

Algebraic Graph
Statics

compas_tna

Thrust Network
Analysis

compas_3gs

3D Graphic Statics

compas_fea

Finite Element
Analysis

compas_fab

Robotic Fabrication

compas_vol

Volumetric modelling

compas_knit

Spatial Knitting

compas_dem

Discrete Element
Modeling (3DEC)

compas_dr6

6DOF Dynamic
Relaxation

compas_rbe

Rigid Block
Equilibrium

compas_ml

Machine Learning

compas_fofin

Form Finding

compas_rocking

Collapse Mechanisms

compas_assembly

Discrete element
assemblies

compas_pattern

Mesh Topology
Finding

Next week

- Make sure examples run on your system
- Get help if necessary
 - Forum
 - Issue tracker
 - Friday 1-4pm
 -
- Getting started
 - Dev tools 101
 - Python 101
 - COMPAS 101