SolidsPy: Teaching Computational Modeling with Python

Nicolás Guarín-Zapata – @nicoguaro

February 9, 2018

UNIVERSIDAD EAFIT®

What am I sharing today?

Our teaching experience in computational modeling with Python using a finite element module/program.

Outline

- Introduction
- Finite Element Method
- SolidsPy
- Examples of use

Introduction

Teaching goals

- Computational modeling
- Computing with Python
- Numerical methods

Computational Modeling

- Second year, undergraduate course
- Use of computing to solve engineering problems
- Use SolidsPy building-blocks to perform mechanical simulations

Introduction to Finite Element Method

- First year, graduate-level course
- Mathematical details of the method
- Step-by-step construction of a Finite Element program

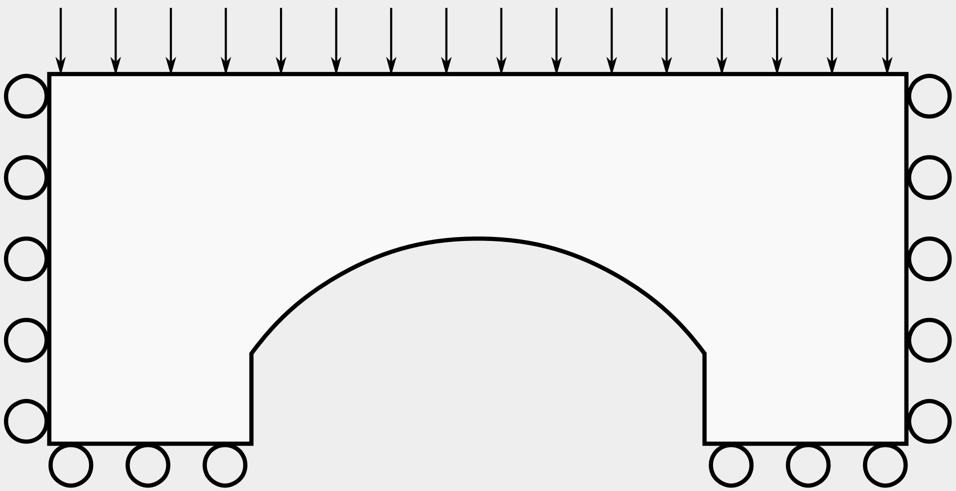
The Finite Element Method

What is the Finite Element Method?

A method for solving problems of engineering and mathematical physics that involve partial differential equations.

Let's see an example

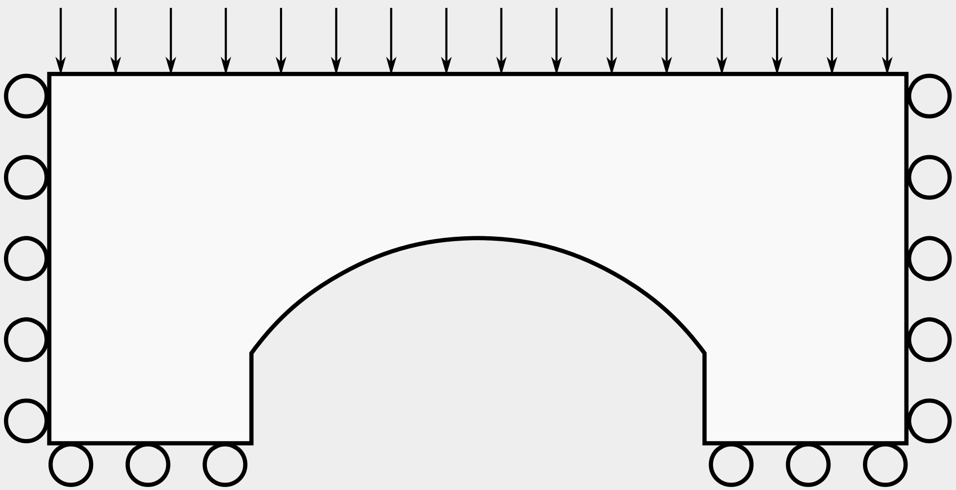
The differential equation

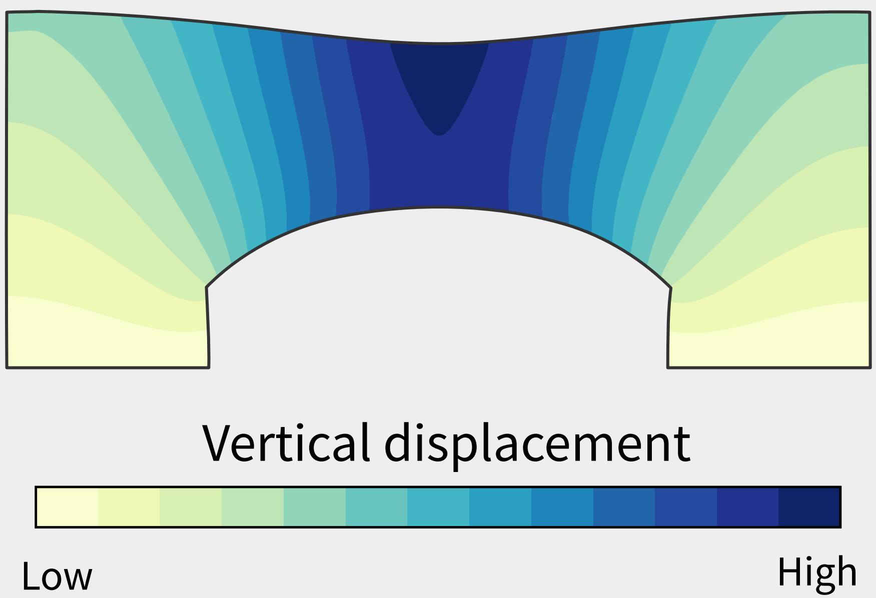


$$\mu \nabla^2 \boldsymbol{u} + (\mu + \lambda) \nabla (\nabla \cdot \boldsymbol{u}) + \boldsymbol{F} = 0$$

And boundary conditions

The solution

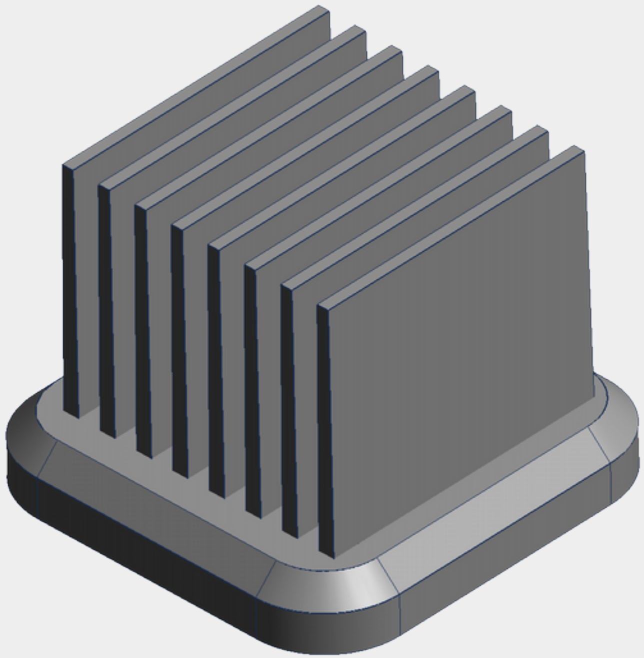


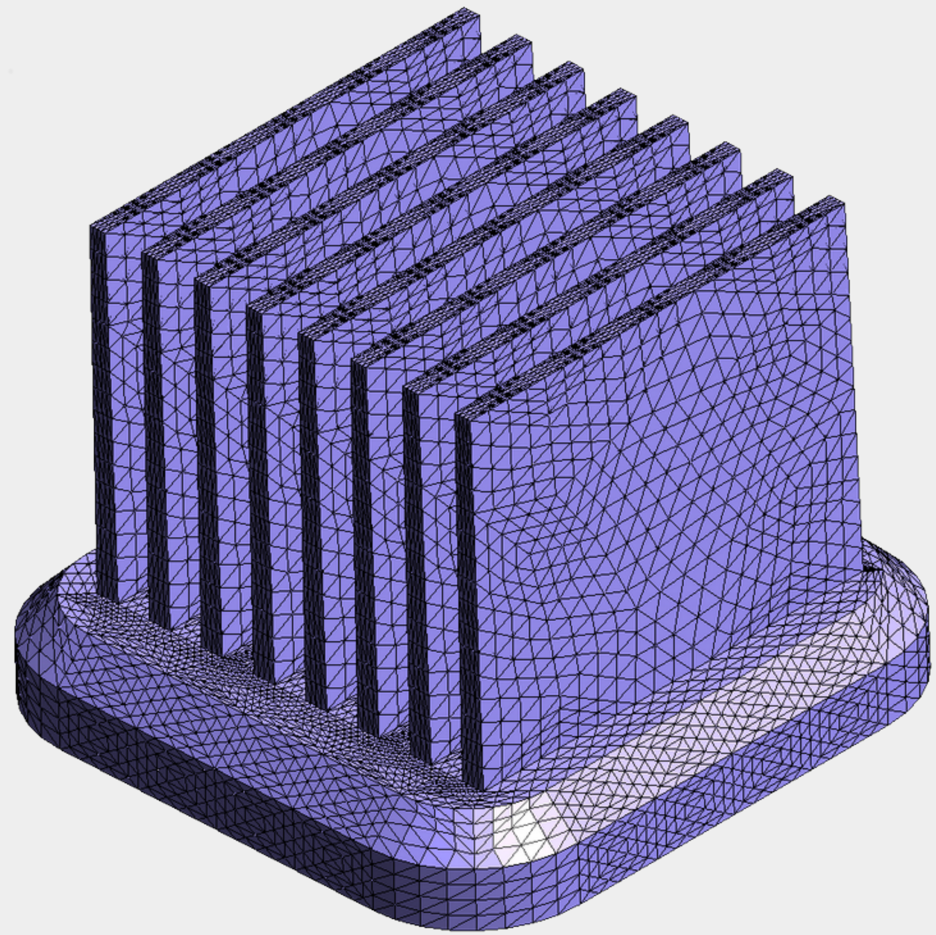


Different steps involved

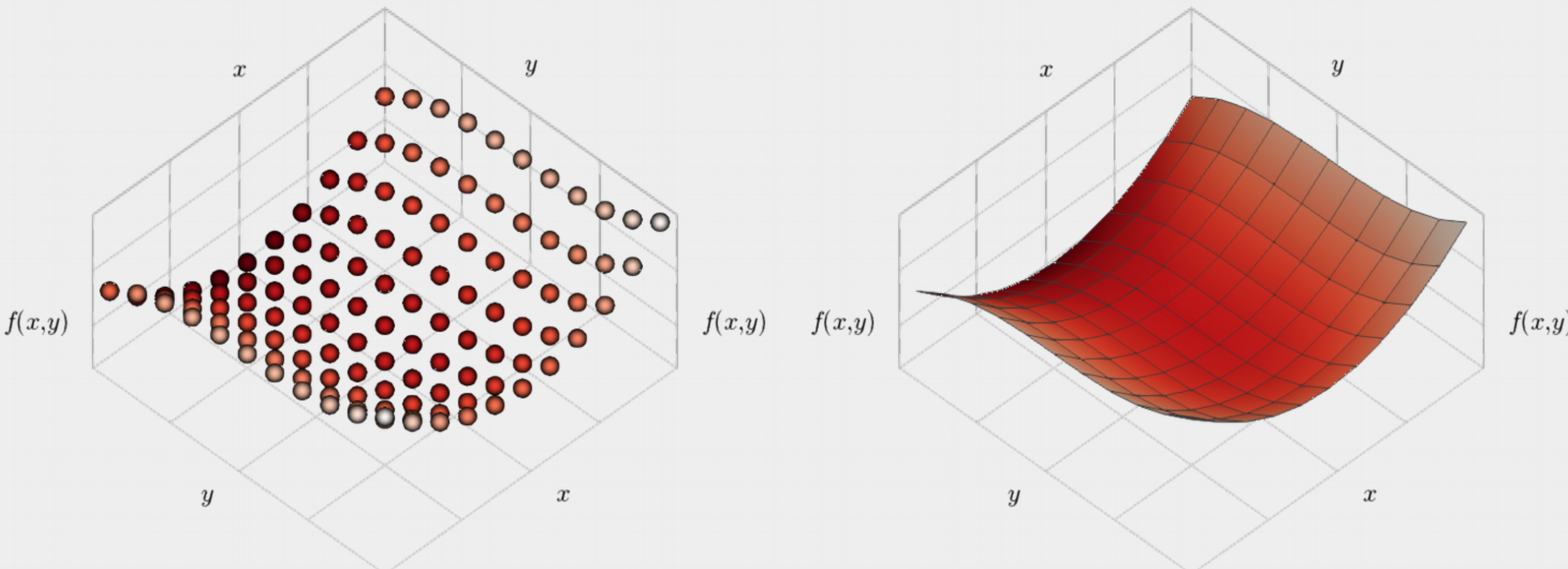
- Geometry discretization
- Interpolation
- Numerical integration
- Equations solution
- Post-processing / Visualization

Geometry discretization



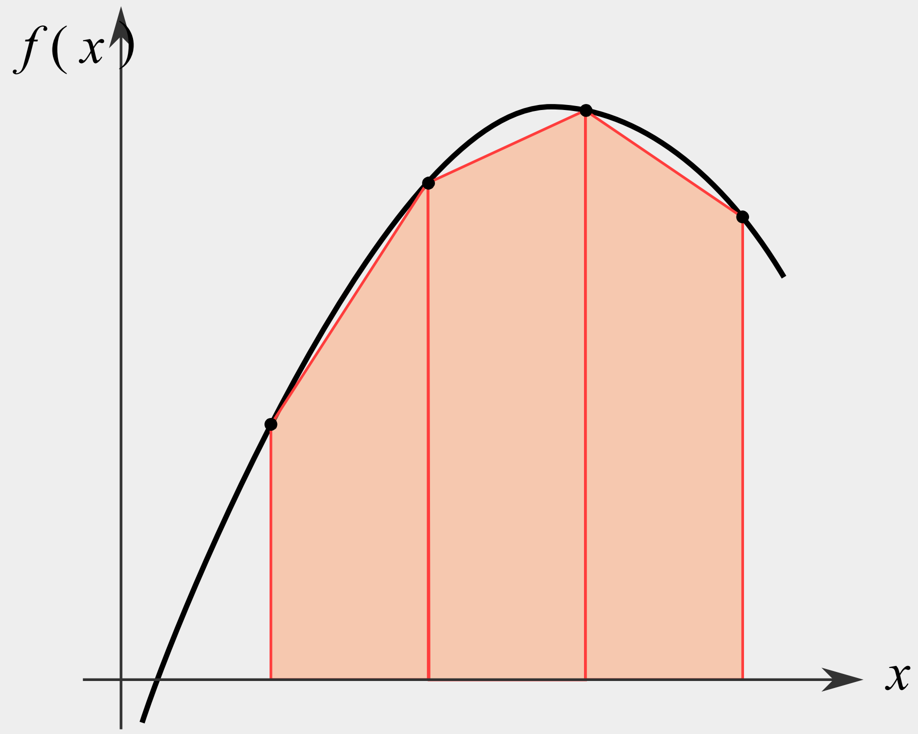


Interpolation



Numerical integration

Using **interpolation** and **numerical integration**, the problem translates into solving a linear system.



Equations solution

Nodal displacements

$$[K]\{U\}=\{F\}$$

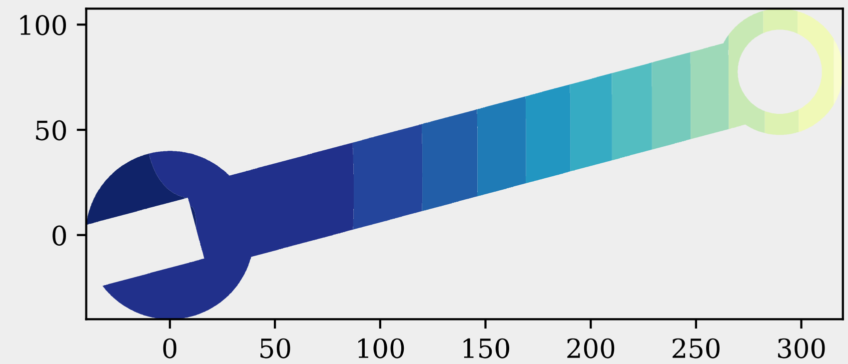
Stiffness matrix

Nodal forces

SciPy has several solvers that can be used for the solution of the system.

Visualization

Once we have the solution, we visualize the results to check that it makes sense.



SolidsPy

Open-source ecosystem

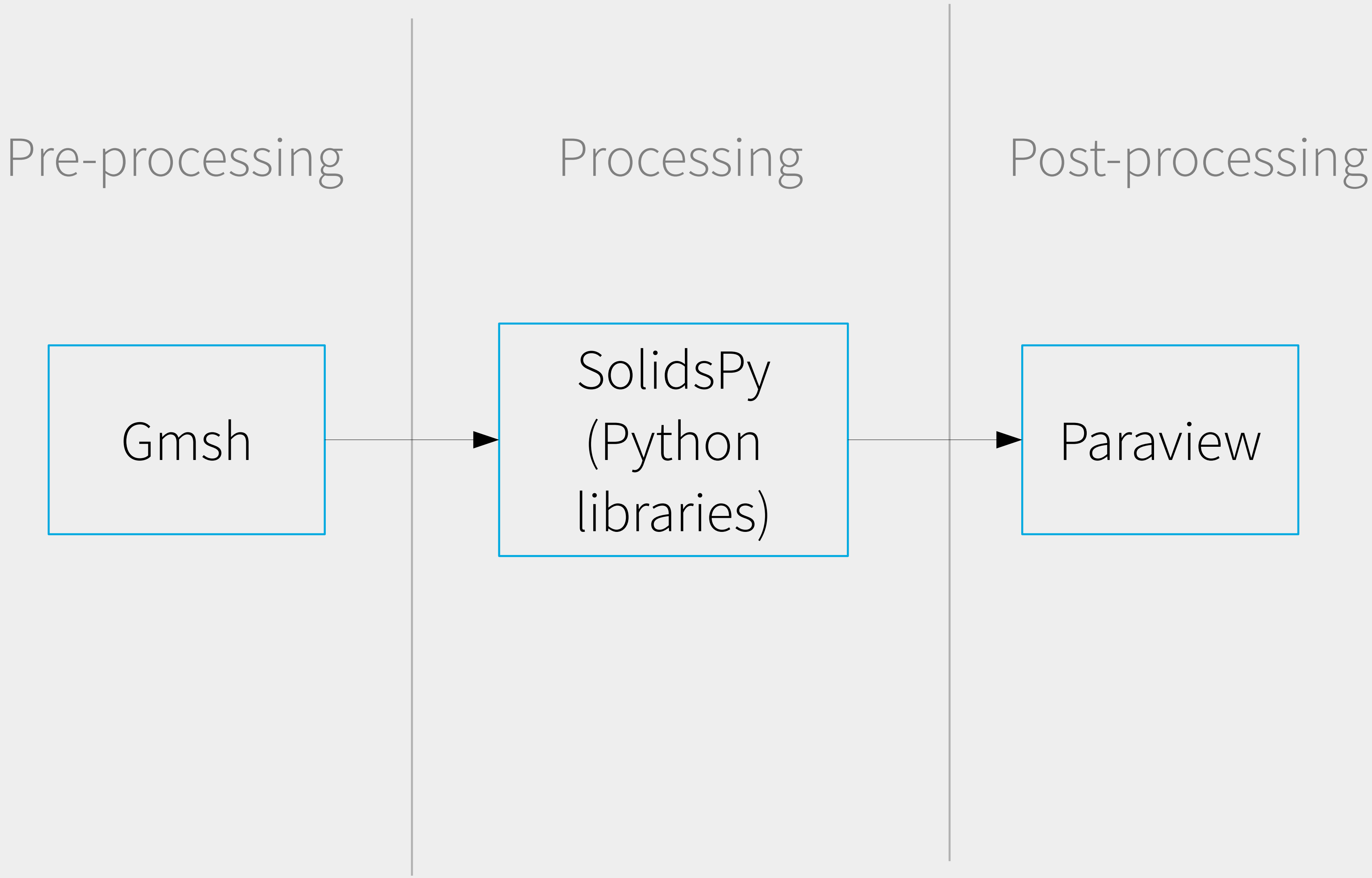
Python libraries:

- Numpy
- Scipy
- SymPy
- Matplotlib
- meshio

External software:

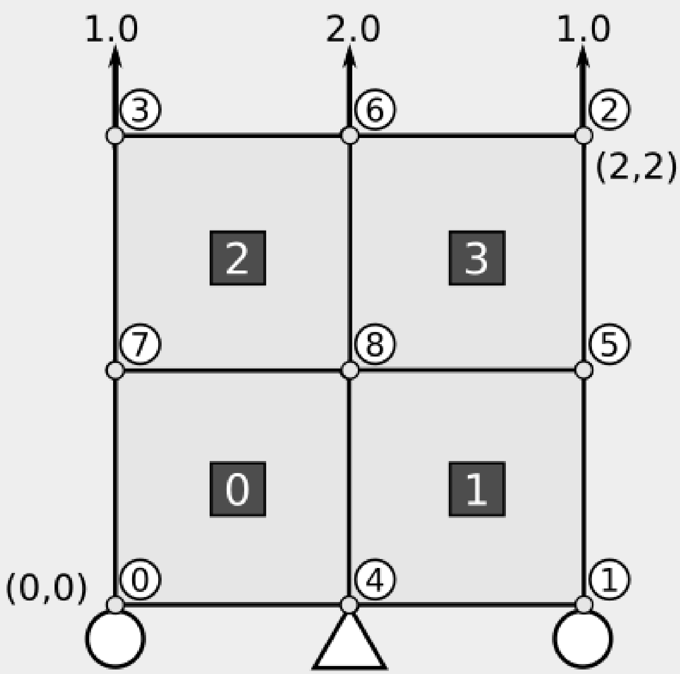
- Gmsh
- ParaView

Open-source ecosystem



Simple input files

Let's consider this simple model.



Simple input files

nodes.txt

```
0  0.00  0.00   0  -1
1  2.00  0.00   0  -1
2  2.00  2.00   0   0
3  0.00  2.00   0   0
4  1.00  0.00  -1  -1
5  2.00  1.00   0   0
6  1.00  2.00   0   0
7  0.00  1.00   0   0
8  1.00  1.00   0   0
```

mater.txt

```
1.0  0.3
```

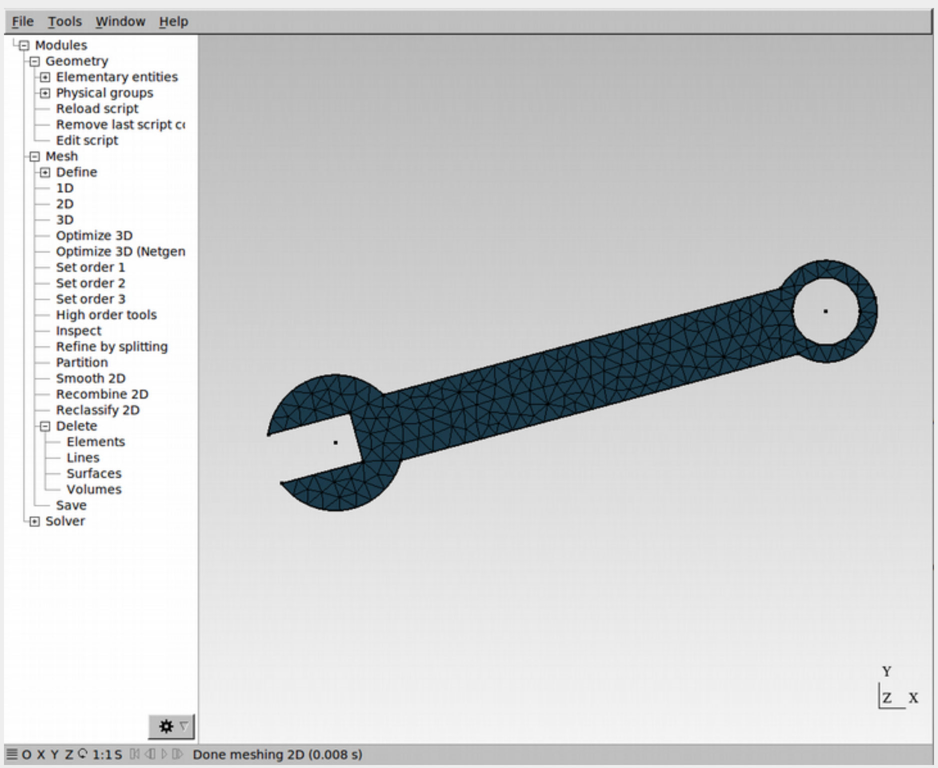
eles.txt

```
0   1   1   0   4   8   7
1   1   1   4   1   5   8
2   1   1   7   8   6   3
3   1   1   8   5   2   6
```

loads.txt

```
3  0.0  1.0
6  0.0  2.0
2  0.0  1.0
```

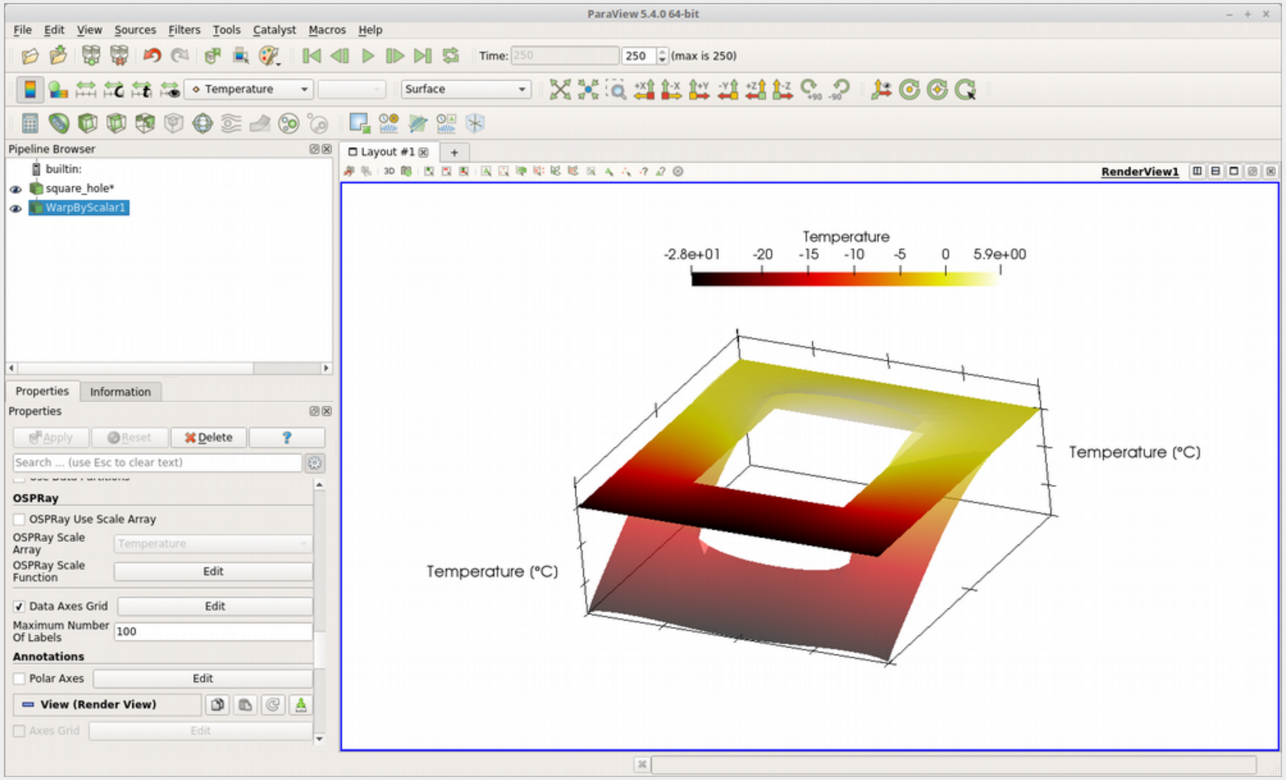
Gmsh for more complicated models



We use **Gmsh** for complicated geometries and then convert them to SolidsPy's format using **meshio**.

Paraview for complex visualizations

We can use **meshio** to export results to Paraview



Examples of use

SolidsPy as a standalone FE program

Just requires the path to input files to complete an analysis

Let’s see an example

```python
import matplotlib.pyplot as plt
from solidspy import solids_GUI
disp = solids_GUI()
plt.show()
```

SolidsPy as a standalone FE program

Several input files for different models in examples in the following repo:

https://github.com/AppliedMechanics-EAFIT/SolidsPy-meshes

Building-blocks for a Finite Element Analysis

- Mesh import/export
- Elements library
- Assembler
- Solver

Let’s see an example

```python
import solidspy as solids

# Reading the data
nodes, mats, elements, loads = solids.preprocesor.readin()

# Forming the system of equations
DME, IBC, neq = solids.assemutil.DME(nodes, elements)
KG = solids.assemutil.assembler(elements, mats, nodes, neq, DME)
RHSG = solids.assemutil.loadasem(loads, IBC, neq)

# System solution
UG = solids.solutil.static_sol(KG, RHSG)

# Post-processing
UC = solids.postprocesor.complete_disp(IBC, nodes, UG)
E_nodes, S_nodes = solids.postprocesor.strain_nodes(nodes,
    elements, mats, UC)
solids.postprocesor.fields_plot(elements, nodes, UC,
    E_nodes=E_nodes, S_nodes=S_nodes)
```

Summary

SolidsPy is ...

- FEM code in an open-source ecosystem
- Easy to use
- Used to teach
  - Computational modeling for undergrads
  - Finite element method for grads

SolidsPy

https://github.com/AppliedMechanics-EAFIT/SolidsPy