



This lecture will be recorded



C O M P A S

064-0026-00L: COMPAS II

Introduction to Computational Methods for Digital
Fabrication in Architecture

```
def smooth_mesh_length(mesh, lmin, lmax, fixed=None, kmax=100):
    """Smooth the mesh length by iteratively adjusting the vertex positions.
    The function iterates up to kmax times, adjusting the vertex positions
    until the mesh length is within the specified range [lmin, lmax].
    If a callback function is provided, it will be called after each iteration.
    """
    # Initialize the mesh length
    l = mesh.length()

    # Iterate until the mesh length is within the specified range
    for k in range(kmax):
        # Update the mesh length
        l = mesh.length()

        # Check if the mesh length is within the specified range
        if lmin <= l <= lmax:
            break

        # If the mesh length is not within the specified range,
        # adjust the vertex positions
        for key in mesh.vertices():
            if key not in fixed:
                # Calculate the distance from the center of mass to the vertex
                c = center_of_mass_polygon([key_xyz[ nbr ] for nbr in mesh.vertex_neighbours(key, ordered=True)])
                d = (c[0] - key_xyz[key][0])**2 + (c[1] - key_xyz[key][1])**2 + (c[2] - key_xyz[key][2])**2
                d = d**0.5

                # Calculate the adjustment factor
                attr = mesh.vertex[key]
                attr['x'] += d * (c[0] - key_xyz[key][0])
                attr['y'] += d * (c[1] - key_xyz[key][1])
                attr['z'] += d * (c[2] - key_xyz[key][2])

        # Update the mesh length
        l = mesh.length()

        # Call the callback function
        if callable(callback):
            callback(mesh, k, callback_args)

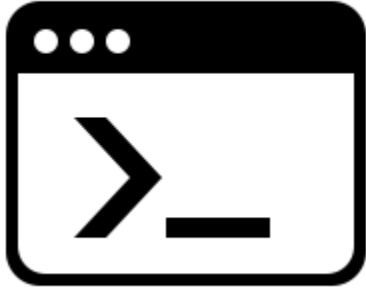
    # Return the smoothed mesh
    return mesh
```

slides + code

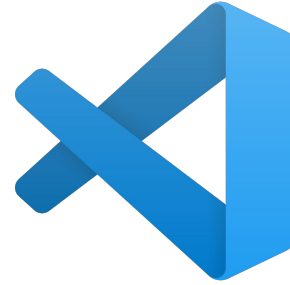
<https://dfab.link/fs2022>

Today's goal

Recap **design to fabrication workflows**



`docker-compose up -d`



Right-click → Compose Up

`docker/moveit-rrc-noetic-update`

`docker/moveit-rrc-noetic`

`docker/gofa-rrc-noetic`

Design

Input

Process

Output

Fabrication

Geometry

Discrete (eg. assembly)

Configurations

Machine Instructions

Robot model

Continuous (eg. print)

Frames

Planning scene

Hands on!

Thanks!

