Acoustic Topology Optimization in openCFS

Felix Huber

Institute of Mechanics and Mechatronics - TU Wien

04.07.2025

Recap: Structural Topology Optimization

Solid Isotropic Material with Penalization (SIMP)

$$
\begin{aligned}
\text{given} \quad & \mathbf{f} \in \mathbb{C}^{N \cdot \mathrm{DOF}} \\
\text{find} \quad & \min_{\rho} J(\mathbf{u}) = \min_{\rho} \mathbf{u}^{\mathsf{T}} \mathbf{f} \in \mathbb{R}^{+} \\
\text{s.t.} \quad & \widetilde{\mathbf{K}} \mathbf{u} = \mathbf{f} \\
& \widetilde{\mathbf{K}} = \bigwedge_{e=1}^{E} \bar{\rho}_e^{p} \mathbf{K}_e \\
& \bar{\rho}_e := \frac{\sum_{\eta=1}^{E} w(\mathbf{x}_\eta - \mathbf{x}_e) \rho_\eta}{w(\mathbf{x}_\eta - \mathbf{x}_e)} && \forall\, 1 \leqslant e \leqslant E \\
& w(d) = \max(0, R - |d|) \\
& \rho_e \in [\rho_{\min}, 1] && \forall\, 1 \leqslant e \leqslant E
\end{aligned}
$$

Recap: Sensitivity and Constraints

For the general compliance problem, the sensitivity is directly computed with the adjoined method as

$$\frac{\partial J}{\partial \rho_e} = -\mathbf{u}^{\mathsf{T}} \frac{\partial \widetilde{\mathbf{K}}}{\partial \rho_e} \mathbf{u} \qquad \forall\, 1 \leqslant e \leqslant E\,.$$

Additionally, we can add constraints for the optimization process e.g.

$$\frac{1}{E} \sum_{e=1}^{E} \bar{\rho}_e(\rho) \leqslant V^*$$

as volume constraint.

Recap: Optimize

init Set initial pseudo densities $\rho_e^{(0)} \quad \forall\, 1 \leqslant e \leqslant E$

1. Apply filtering $\bar{\rho}_e := \frac{\sum_{\eta=1}^{E} w(r)\rho_\eta^{(i)}}{w(r)}$
2. Penalize $\widetilde{\mathbf{K}} = \bigwedge_{e=1}^{E} \bar{\rho}_e^p \mathbf{K}_e$
3. Solve FEM problem $\widetilde{\mathbf{K}}\mathbf{u} = \mathbf{f}$ to obtain $\mathbf{u}$
4. Check convergence (e.g. by evaluating cost function)
5. Optimizer computes new pseudo densities $\rho_e^{(i+1)}$ from sensitivities and constraints

We quit the iteration at specified convergence criteria (e.g. relative change in cost function).

Acoustic Cost Functions

Squared pressure at selected DOFs (`dynamicOutput`)

$$J = \mathbf{p}^\mathsf{T}\mathbf{L}\mathbf{p}^*$$

$$\frac{\partial J}{\partial \rho_e} = 2\,\mathrm{Re}\left\{\boldsymbol{\lambda}^\mathsf{T}\frac{\partial \widetilde{\mathbf{S}}}{\partial \rho_e}\mathbf{p}\right\}$$

$$\text{s.t.} \quad \widetilde{\mathbf{S}}^\mathsf{T}\boldsymbol{\lambda} = -\mathbf{L}\mathbf{p}^*$$

Squared reflected pressure at selected DOFs (`reflectedWave`)

$$J = (\mathbf{p} - p_I)^\mathsf{T}\mathbf{L}(\mathbf{p} - p_I)^*$$

$$\frac{\partial J}{\partial \rho_e} = 2\,\mathrm{Re}\left\{\boldsymbol{\lambda}^\mathsf{T}\frac{\partial \widetilde{\mathbf{S}}}{\partial \rho_e}(\mathbf{p} - p_I)\right\}$$

$$\text{s.t.} \quad \widetilde{\mathbf{S}}^\mathsf{T}\boldsymbol{\lambda} = -\mathbf{L}(\mathbf{p} - p_I)^*$$

- $\mathbf{p}$ complex acoustic pressure
- $\mathbf{L}$ selector matrix
- $\widetilde{\mathbf{S}}$ parameterized system matrix
- $\boldsymbol{\lambda}$ adjoined solution
- p_I complex incident wave amplitude (at selected DOFs)

Acoustic Topology Optimization

Dynamic SIM(P) for the acoustic pressure formulation (for single frequency ω)

$$
\begin{aligned}
\text{given} \quad & \mathbf{f} \in \mathbb{C}^N \\
\text{find} \quad & \min_{\rho} J(\mathbf{p}) \in \mathbb{R}^+ \\
\text{s.t.} \quad & \widetilde{\mathbf{S}}\mathbf{p} = \mathbf{f} \\
& \widetilde{\mathbf{S}} = \bigwedge_{e=1}^{E} \bar{\rho}_e \left(\mathbf{K}_e - \frac{\omega^2}{c_0^2}\mathbf{M}_e\right) \\
& \bar{\rho}_e := \frac{\sum_{\eta=1}^{E} w(\mathbf{x}_\eta - \mathbf{x}_e)\rho_\eta}{w(\mathbf{x}_\eta - \mathbf{x}_e)} && \forall\, 1 \leqslant e \leqslant E \\
& w(d) = \max(0, R - |d|) \\
& \rho_e \in [\rho_{\min}, 1] && \forall\, 1 \leqslant e \leqslant E
\end{aligned}
$$

Multifrequency

In openCFS we have two ways to optimize for multiple frequencies.

Superposition (weighted average)

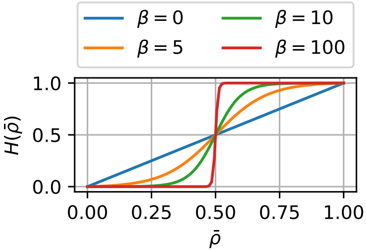
$$
\begin{aligned}
\text{given} \quad & \mathbf{f}_{\omega 1}, \mathbf{f}_{\omega 2} \\
\text{find} \quad & \min_{\rho} \frac{1}{2} J(\mathbf{p}_{\omega 1}) + \frac{1}{2} J(\mathbf{p}_{\omega 2}) \\
\text{s.t.} \quad & \widetilde{\mathbf{S}}_{\omega 1} \mathbf{p}_{\omega 1} = \mathbf{f}_{\omega 1} \\
& \widetilde{\mathbf{S}}_{\omega 2} \mathbf{p}_{\omega 2} = \mathbf{f}_{\omega 2} \\
& \dots
\end{aligned}
$$

min/max problem (slack formulation)

$$
\begin{aligned}
\text{given} \quad & \mathbf{f}_{\omega 1}, \mathbf{f}_{\omega 2} \\
\text{find} \quad & \min_{\rho} \alpha \\
\text{s.t.} \quad & \widetilde{\mathbf{S}}_{\omega 1} \mathbf{p}_{\omega 1} = \mathbf{f}_{\omega 1} \\
& \widetilde{\mathbf{S}}_{\omega 2} \mathbf{p}_{\omega 2} = \mathbf{f}_{\omega 2} \\
& J(\mathbf{p}_{\omega 1}) \leqslant \alpha \\
& J(\mathbf{p}_{\omega 2}) \leqslant \alpha \\
& \dots \\
& \alpha \in [0, \infty)
\end{aligned}
$$

Advanced Filtering

Projection: Apply a continuous Heaviside function to the density filtered field $H_\beta(\bar{\rho})$ prior penalization.



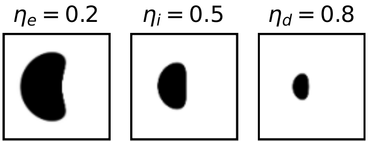
$$H_\beta(\bar{\rho}) = \frac{\tanh(\beta\eta) + \tanh(\beta(\bar{\rho} - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1 - \eta))}$$

Robust: Solve three systems per iteration, with three Heaviside filter settings.

$\eta_e < 0.5$ erode

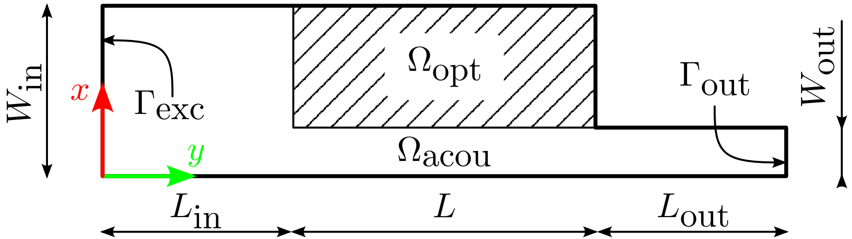
$\eta_d > 0.5$ dilate

$\eta_i = 0.5$ intermediate



Shape Mapping: Parameterized curve shapes that are mapped to the density field.

Problem Statement



Goal: Maximize transmission between Γ_{in} and Γ_{out}.

$$L_{\text{in}} = 0.1\,\text{m} \qquad L = 0.05\,\text{m} \qquad L_{\text{out}} = 0.1\,\text{m}$$
$$W_{\text{in}} = 0.025\,\text{m} \qquad W_{\text{out}} = 0.004\,\text{m} \qquad \omega \in [1000\,\text{Hz}, 2000\,\text{Hz}]$$

Mesh Generation

Uniform grid with rectangular quadrilateral elements, created using the Python `mesh_tool` module.

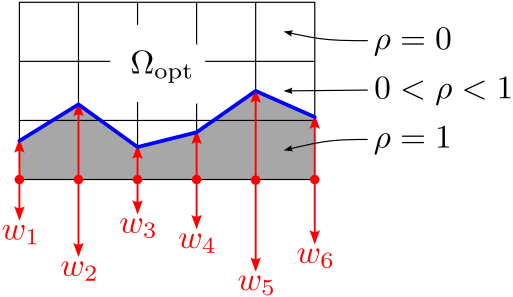
1. Create uniform grid with `mesh_tool.create_2d_mesh()`
2. Assign elements to regions
3. Create line elements and assign to surface regions
4. Assign nodes to nodesets
5. Export mesh file with `mesh_tool.write_ansys_mesh()`

Optimization Setup XML

```xml
<optimization>
    <costFunction type="reflectedWave" task="minimize" multiple_excitation="true">
        <stopping queue="5" value="0.001" type="relativeCostChange"/>
        <output>
            <acoustic name="L_exc" value="1" surfRegion="L_exc_surf" volumeNeighbour="S_acou"/>
        </output>
    </costFunction>

    <constraint type="volume" access="plain" mode="constraint" bound="upperBound" value="0.8"/>

    <optimizer type="snopt" maxIterations="100"/>

    <ersatzMaterial region="S_opt" material="acoustic" method="simp">
        <designSpace local_element_cache="false"/>
        <!-- Filter section -->
        <filters>
            <filter neighborhood="maxEdge" value="1.6" type="density"/>
        </filters>
        <design name="density" initial="0.5" physical_lower="1e-9" upper="1.0"/>
        <transferFunction type="simp" application="acoustic" param="1"/>
        <export save="last" write="iteration"/>
    </ersatzMaterial>
    <commit mode="forward" stride="10"/>
</optimization>
```

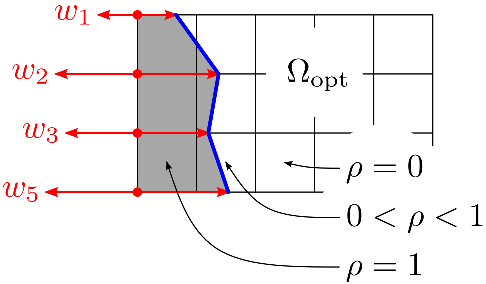
Shape Mapping XML

Shape mapping DOFs oriented in x-direction.




```xml
<!-- Filter section -->
<shapeMap beta="384" shape="tanh" ...>
    <node fixed="0" dof="x"/>
    <profile lower="0" upper="0.1" initial=".015"/>
</shapeMap>
```

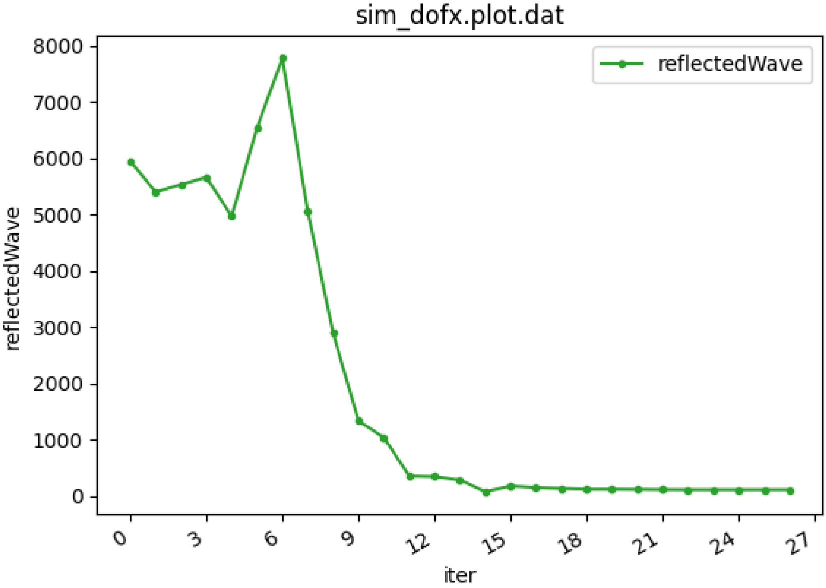
Shape mapping DOFs oriented in y-direction.




```xml
<!-- Filter section -->
<shapeMap beta="384" shape="tanh" ...>
    <node dof="y" fixed="0.09"/>
    <profile lower="0" upper="0.14" initial=".02"/>
</shapeMap>
```

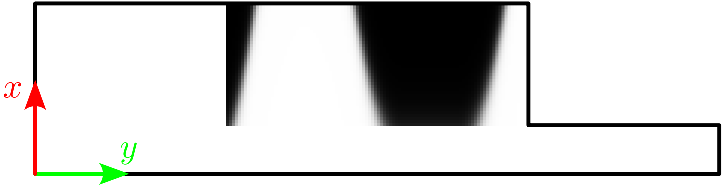
Monitoring: `plotviz.py`

Display the contents of the `*.plot.dat` file e.g to check convergence.



Monitoring: `show_density.py`

Display the contents of the `*.density.xml` file to visualize design.



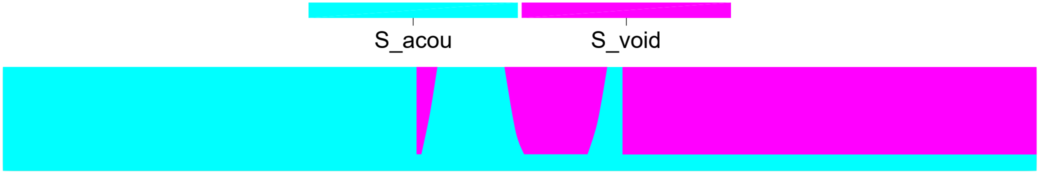
Monitoring

Monitoring tools can be used in parallel to the optimization process, this allows the user to spot problems early.

Post Processing: `evaluate`, `threshhold.py`

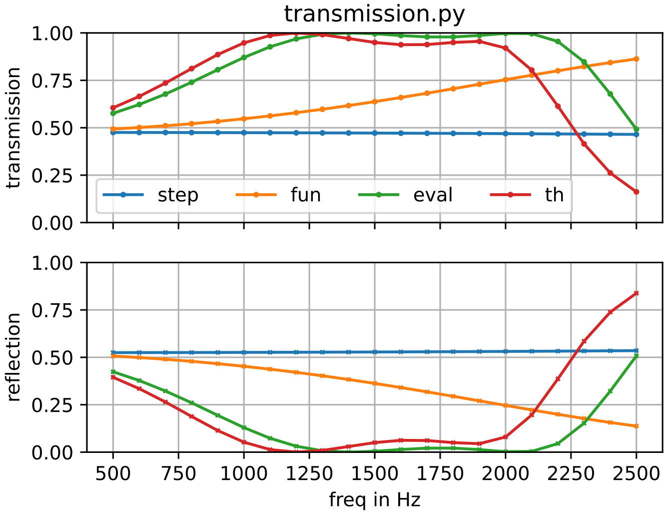
Evaluate: Instead of selecting an optimizer, `evaluate` can be used to solve the FEM problem with loaded pseudo densities.

Threshhold: To ensure the elements with $0 < \rho_e < 1$ don't cause problems, threshholding is used to simulate a pure FEM model, without pseudo densities. Helper program `threshhold.py` (not published yet).



Post Processing: `transmission.py`

Transmission: Finally, we compute the transmission/reflection coefficients. Helper program `transmission.py` (not published yet).



Summary

Good News

Topology optimization workflow in openCFS now supports the acoustic pressure formulation for harmonic problems.

Features:

- Dynamic SIMP for `acouPressure` PDE
- `dynamicOut` and `reflectedWave` cost functions
- Multi frequency optimization in two formulations
- Several filter options working and tested
- Shape mapping support (not published yet)
- New post processing Python modules (not published yet)

Any Questions?