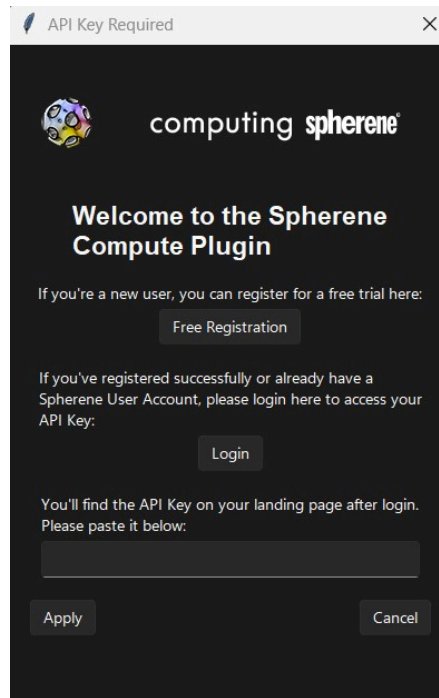


Quick start guide of SphereneNTOP

In nTop, Spherene is a **cloud-based** connector service that requires registration. When you activate Spherene computation for the first time, a request window will appear prompting you to enter your API key.



Installation & Uninstallation

You can use the connector installer `.msi` to install the Spherene connector:

- Execute the connector `.msi` file to install the Spherene. When asked, select a folder where you want to install the Spherene connector.
- Restart nTop, and you should see the Spherene block in the Connector section.

To uninstall the Spherene connector:

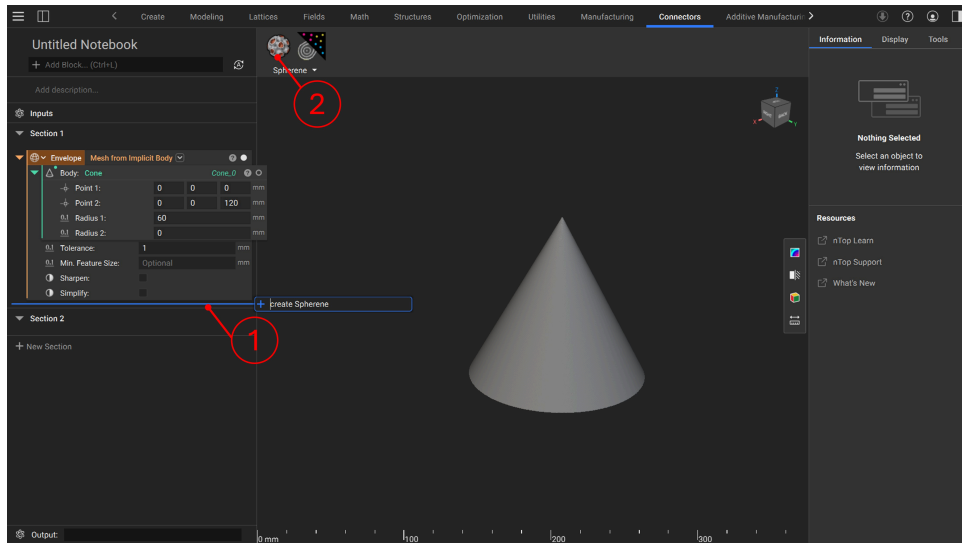
- Execute the same `.msi` file to begin the uninstallation process.
- Delete the Spherene connector folder, then restart your PC to complete a clean uninstallation.

UI of Spherene in nTop

In this section, we introduce the user interface of the Spherene connector and explain how to start the Spherene Compute block. **Advanced nTop users** can skip this section and proceed to the [Quick-start example](#) for the quick-start example.

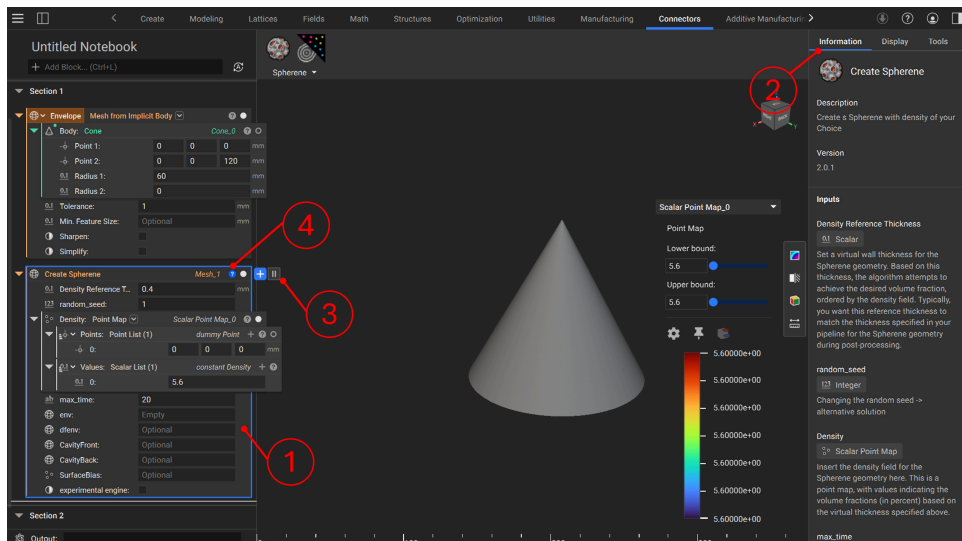
There are two ways to add the Spherene Compute block in nTop:

- **Solution 1:** Double-click at position 1, type `Create Spherene`, and press Enter.
- **Solution 2:** Click the `Create Spherene` button under `nTop > Connectors`, as shown in position 2 in the figure below.



The UI of the Spherene Compute block is shown at position 1 in the figure below. Details for each attribute are displayed in the `Information` section (position 2).

The first important step after adding the Spherene Compute block is to right-click on the block and set it to `Manual Run Mode`. This prevents nTop from automatically running the computation every time you change an attribute, saving both computational resources and time. You can find the pause/run button at position 3. To start the computation, please ensure all settings are properly configured. The `?` button at position 4 will provide hints if there are any issues with the configuration.



Quick-start example

The Spherene connector generates Adaptive Density Minimal Surfaces (ADMS) geometry based on a given envelope. In addition to the basic functionality of creating an ADMS with constant density, the connector offers **four advanced capabilities**:

- Generate an ADMS with varying densities, allowing precise control over material distribution, and therefore, material properties.
- Generate an ADMS with varying surface bias to introduce controlled variation in surface morphology.
- Generate ADMS in sharp edges or thin regions using a density field envelope (`dfenv`).
- Generate cavities in the ADMS geometry, which is useful for integrating inlets, outlets, or other functional features.

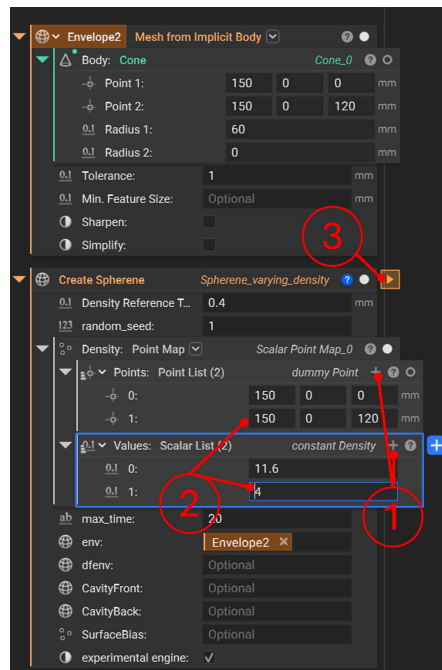
In this section, we use a simple example with a cone-shaped envelope to demonstrate how to use the Spherene connector and its advanced functions. For more complex design tasks, refer to the [Advanced Examples](#).

Example with varying density

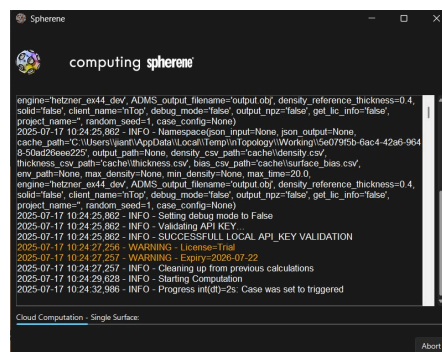
Density is defined as the percentage ratio of the generated Spherene volume to the envelope volume^[1]. The `Density` attribute in the Spherene Compute block controls the density distribution of the generated Spherene geometry. If a single point-value pair is provided, the density is treated as constant. If a point map with multiple point-value pairs is assigned, the resulting Spherene geometry will have varying densities.

In this example, we define two points in the `Density` attribute, creating a density gradient between them. The workflow is as follows:

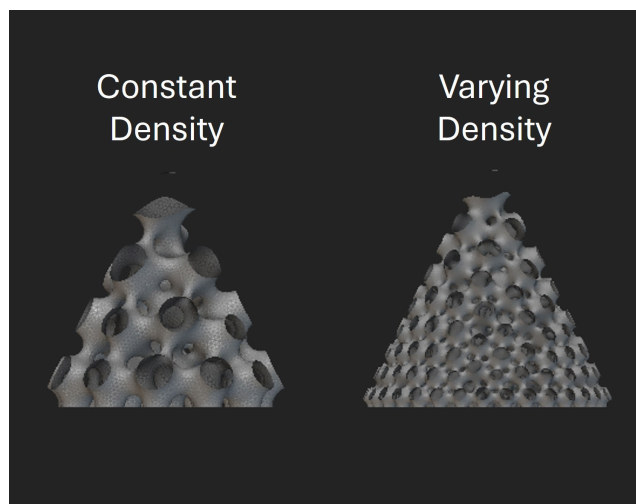
- Click the `+` button (position 1) in the point map under the `Density` attribute to add a new point–value pair.
- Enter the coordinates and corresponding value for the new point–value pair (as shown in position 2).
- Click the `Run` button (position 3) to start the computation.




Once the computation starts, a computing window will appear. You can stop the process at any time by clicking **Abort**. When the computation is complete, the window will close automatically and deliver the computed spherene geometry.



The following figure compares Spherene geometries with constant density and with a density gradient.

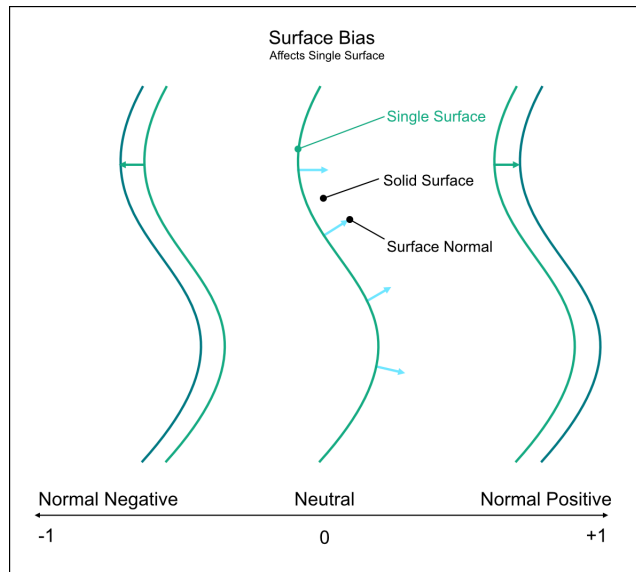


A more complex point map can be assigned to the **Density** attribute for advanced design tasks (see [Advanced Examples](#)). The Spherene connector also provides a helpful tool,

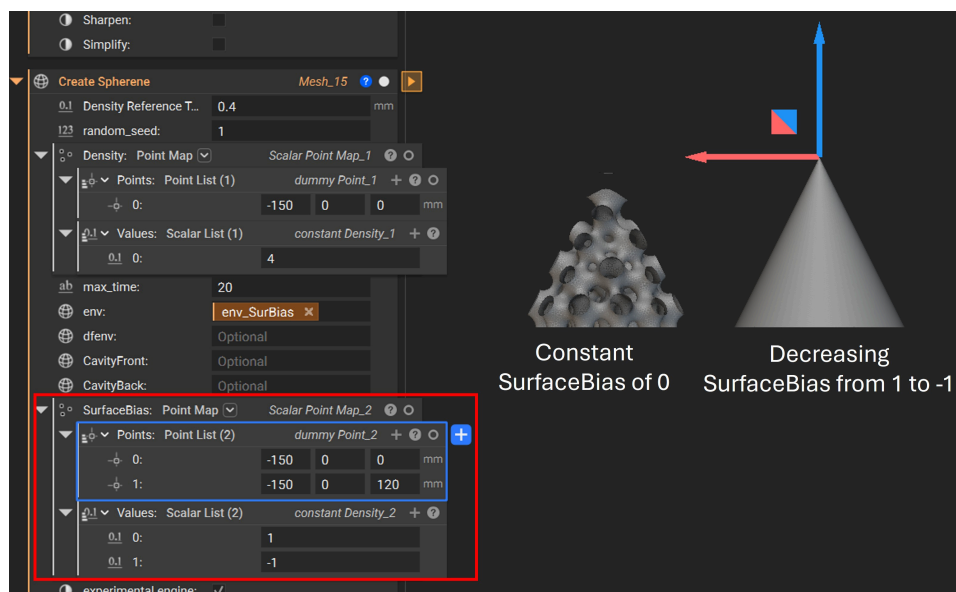
Scalarfield to Point Map , which generates a point map from a scalar field by sampling points and scaling the values to a specified range. For usage examples of **Scalarfield to Point Map**, refer to the [Advanced Examples](#).

Example with varying Surface Bias

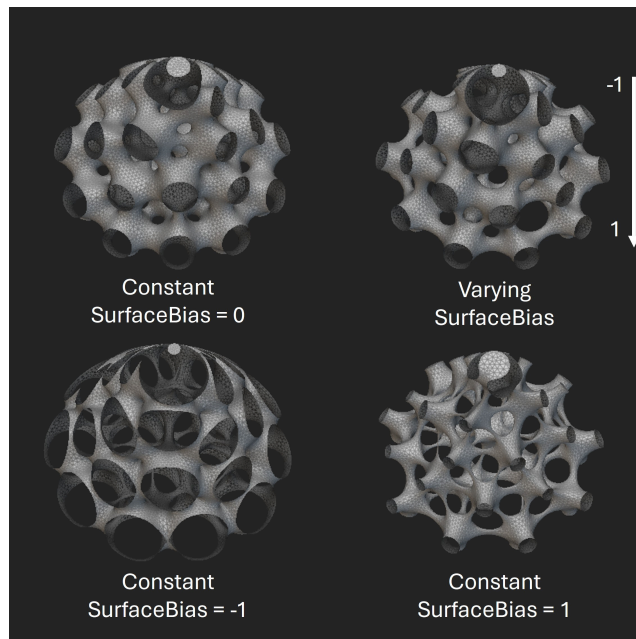
Similar to **Density**, you can control the surface bias of the generated Spherene geometry by assigning a point map to the **SurfaceBias** attribute. Surface bias determines how the minimal surface shifts from its ideal position toward the negative or positive direction along its normal.



In the following example, the surface bias is varied from 1 at the bottom to -1 at the top tip of the cone.



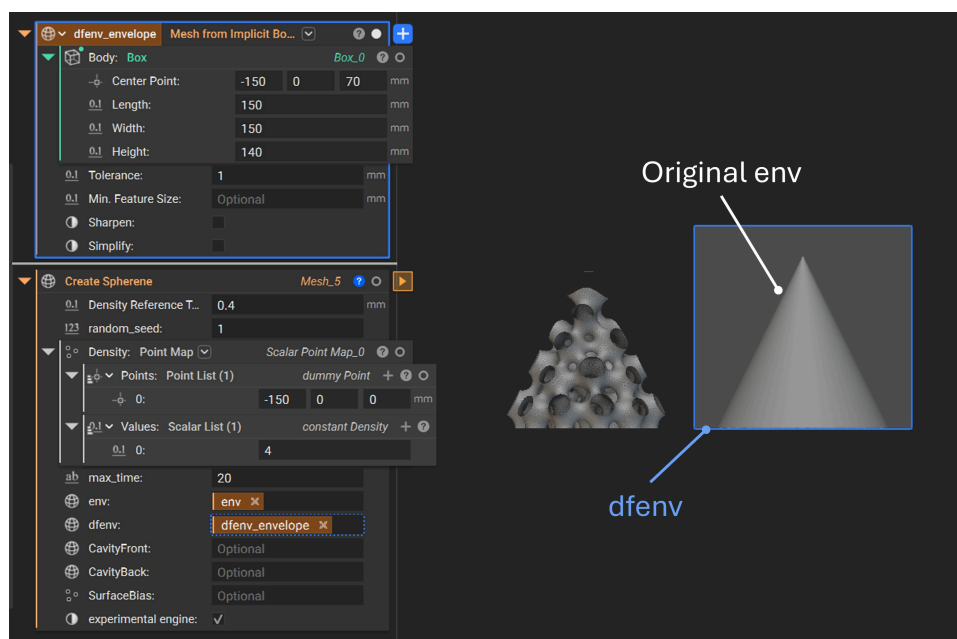
The figure below shows a comparison of Spherene geometries generated with different surface bias settings. Noticeable differences in surface morphology can be observed as the surface bias is adjusted.



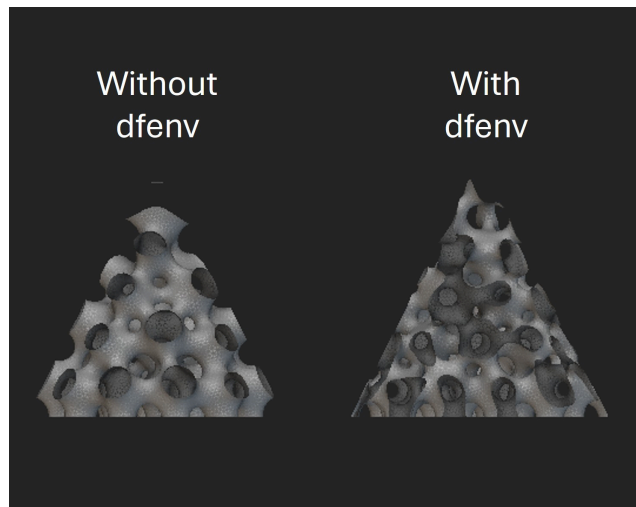
Example with density field envelope dfenv

The `dfenv` attribute allows you to input a mesh as the density field envelope. When a `dfenv` is defined, the Spherene geometry is first generated using `dfenv` as the envelope, then trimmed by the original envelope `env` to produce the final geometry. This enables the generation of Spherene structures at sharp edges or in thin regions where material might otherwise be insufficient.

In the following example, we aim to generate Spherene structures at the sharp tip and edges of a cone. To achieve this, we define a box-shaped `dfenv` (shown in blue) that is larger than the original cone envelope `env` (shown in gray).



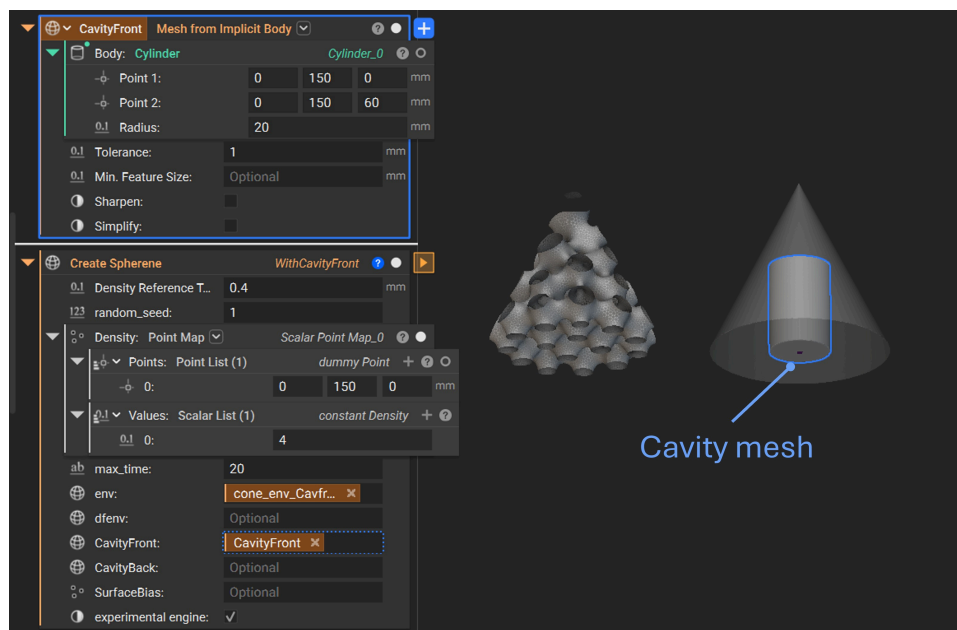
From the result below, you can see that the Spherene structure is successfully generated even at the sharp edges and the top tip with the `dfenv`.



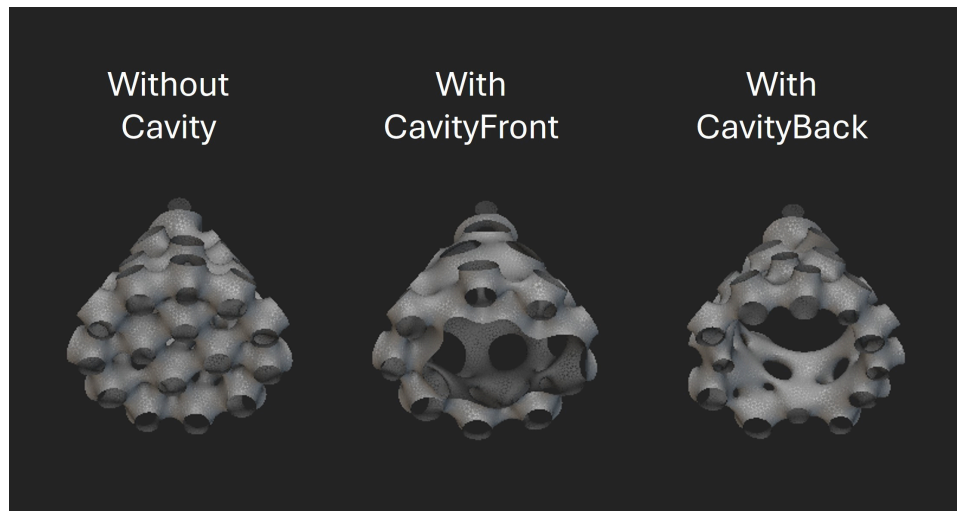
Example with cavity

You can introduce a cavity into the Spherene geometry by simply assigning a cavity mesh to one of the cavity attributes: `CavityFront` or `CavityBack`. This is especially useful for integrating inlets, outlets, or other functional features.

In the following example, a cylindrical cavity is introduced at the bottom of the cone envelope by assigning a cylinder mesh to either the `CavityFront` or `CavityBack` attribute.



By selecting either `CavityFront` or `CavityBack`, the cavity surface will be landed on different spaces of the ADMS structure. The following figure shows a comparison between a Spherene geometry without a cavity, with `CavityFront`, and with `CavityBack`.



Advanced Examples

The previous toy example is intended to demonstrate the basic usage of the Spherene connector. However, Spherene is capable of much more and can significantly enhance real-world design applications!

Below are several examples showcasing **the integration of Spherene in advanced design scenarios**. Detailed nTop settings and results can be explored in the accompanying example .ntop files.

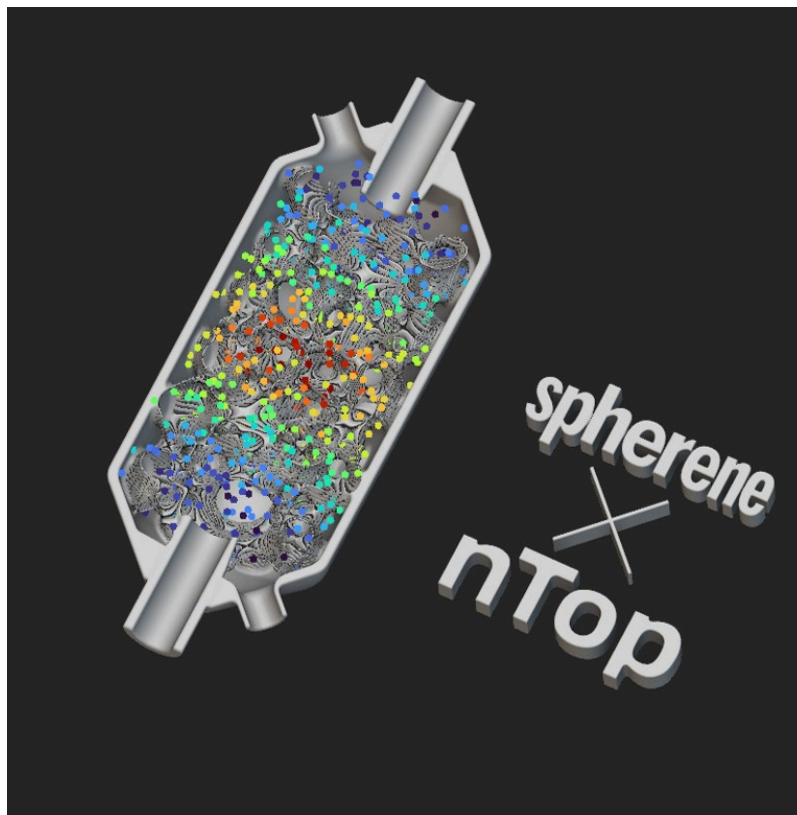
Lightweight bike seat using a Spherene lattice infill driven by mechanical modeling results



Topology optimization of a bracket with Spherene structure for various loading cases



Heat exchanger with Spherene geometry



Lightweight shoe sole design with Spherene geometry



-
1. Density = $\frac{\text{Area of Spherene} \times \text{Density reference thickness}}{\text{Envelope volume}} \times 100\%$, where the density reference thickness is an input parameter in the Spherene compute block. ↩