Material Definition (file: `1D_base_mesh_fixed.hm`)
The material to build the frame will be “steel” with the following (default) properties: linear elastic, isotropic and temperature independent. This material behavior is defined as (and through) the Card Image MAT1.

Units are: N, mm$^2$ and t (tons)

Frame Property Definition & Trouble Shooting
For simplicity reasons the frame will be modeled with CBAR elements referencing a “tube” cross-section with constant properties thoughout. The CBAR element properties can be best created with HyperBeam (integrated in HyperMesh). Once the cross section is defined, the cross section information is then stored as a beamsection within a beamsection collector.

The workflow is as:
- Define the required tube cross-section with HyperBeam and store the respective cross section data (area, moment of inertia etc.) in a beamsection collector
- Generate a property collector with Card Image PBARL. Inside this property collector the previously defined beamsection and a material collector is referenced. Eventually, the property collector is then assigned to all CBAR elements
These steps are explained in some detail next:

I. Defining the tube cross-section with HyperBeam

HyperBeam can be accessed through the main menu ➔ Properties ➔ HyperBeam

Alternatively, HyperBeam may be started from Page 1 ➔ HyperBeam.

Note: As we are using OptiStruct (Radioss bulk) as FEM solver, our life’s becomes easier if we make use of the 1D section library of OptiStruct. Therefore, the “standard section library:” should be OPTISTRUCT, the “standard section type:” should be “Tube”. Other section do exist of course

<table>
<thead>
<tr>
<th>Box</th>
<th>Rod</th>
<th>Chan2</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box1</td>
<td>I</td>
<td>L</td>
<td>Z</td>
</tr>
<tr>
<td>Hat</td>
<td>II</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Tube</td>
<td>Chan</td>
<td>T1</td>
<td></td>
</tr>
<tr>
<td>Bar</td>
<td>Chan1</td>
<td>T2</td>
<td></td>
</tr>
</tbody>
</table>

In the graphic editor of HyperBeam, the chosen tube section is displayed.

For the base design (it is actually just a “guess”) the tubes radii are assumed to be
\( r_{\text{out}} = 12.5 \text{ mm}, r_{\text{inside}} = 10.5 \text{ mm} \).

Of course, the tubes inner (DIM2) and outer (DIM1) radii can be easily changed and adjusted in HyperBeam.

Based on the cross-section the additional values such as area, moments of inertia etc. are automatically determined and listed in the side bar of HyperBeam. All this information is now readily available as a beamsection stored in a beamsection collector (Note, that different beamsections with different attributes, names and ID’s may reside inside a beamsection collector).

Once the definition of the cross section is completed you can return to the HyperMesh GUI by clicking the icon beneath “Model”

II. Defining a property collector
As before with the material collector, make a right mouse button click in the “Model Browser”
and specify the Card Image PBARL as well as the previously defined material.

The Card image of the property collector (PBARL) includes references to:

- Beamsection (here the ID of the beamsection is 8 (not to be mistaken with beamsection collector) of cross section type (CType) TUBE. Inner and outer radii are DIM2 and DIM1
- Material (MID=1)

In the next step, this property collector is assigned to all 1D elements. Hereto, the respective functionality available in the Model Browser is used: Right mouse button click on the before defined property collector, then select, Assign.
To better visualize the 1D elements we switch from the “Traditional Element Representation” to “3D Element Representation”. In addition the element shrink option is activated.

Figure: Car frame meshed with 1D elements (CBARS) with “shrink view” on. The x-symbols outside the frame refer to geometry points. These points will be used to model the wheel suspension.
Figure: Detailed view of the meshed frame (shrink view mode is active). A uniform tube cross section is used for simplicity reasons ($r_o=12.5$ mm, $r_i=10.5$ mm).

**Trouble Shooting**

However, it may happen that some 1D elements are not properly displayed. What is the problem?

Figure: The depicted “line” is actually a CBAR element.

Likely causes & ways to solve this issue

1. Make sure that a “property” is assigned to ALL the 1D elements. This can be checked visually in many ways, for instance, by activating the element display mode:

   ![Property Display](image)

   If a property collector is assigned to the 1D elements they will be displayed with the same color as the property collector.

2. Check the properties of the 1D elements again i.e. review the property collector (in the Model Browser make a right mouse button click on the corresponding property collector --&gt; Card Edit)

3. **Check the element orientation.**
   The element orientation is defined through its local xy-plane. The local element xy-plane in turn is defined through its x-axis (longitudinal direction from node a to b) and a vector v. Basically, one needs to take care of the v vector in order to solve (determine) the element orientation!
To control (or update) the local element xy-plane and thus the v vector go to the bars panel (main menu ➔ Mesh ➔ Edit ➔ 1D Elements ➔ Bars ➔ Update)

Select any of the displayed 1D elements (in the image below the “3D Element Representation” is active)

As shown above this will display not only the local element coordinate system but also the components of the vector v (with respect to the global coordinate system). Here, the vector v parallels the global y-direction (y comp = 1). The element xy-plane thus parallels the global xy-plane. In the next step, we are reviewing the element orientation of a 1D element which is not properly displayed (cross-sections are not shown). Just repeat the “review steps” from before ...
This time the location of node a and b is shown only. In the panel shown below it becomes apparent that the components of the v vector are all zero.

The good thing: the components of vector v can be corrected/updated at any time. Select a single (or multiple elements), then specify the component of the vector v with respect to the global (basic) coordinate system. In case the y-component of vector v is set to y=1 (global y), both, vector v and local y-direction of the element coincide. In the image below the local y direction and v vector are superimposed on each other.

In the image below, the vector v doesn’t coincide with any of the global components. Now it is quite apparent, that vector v and local x-axis together specify the local xy-plane of the element.
The same element is viewed from a different direction