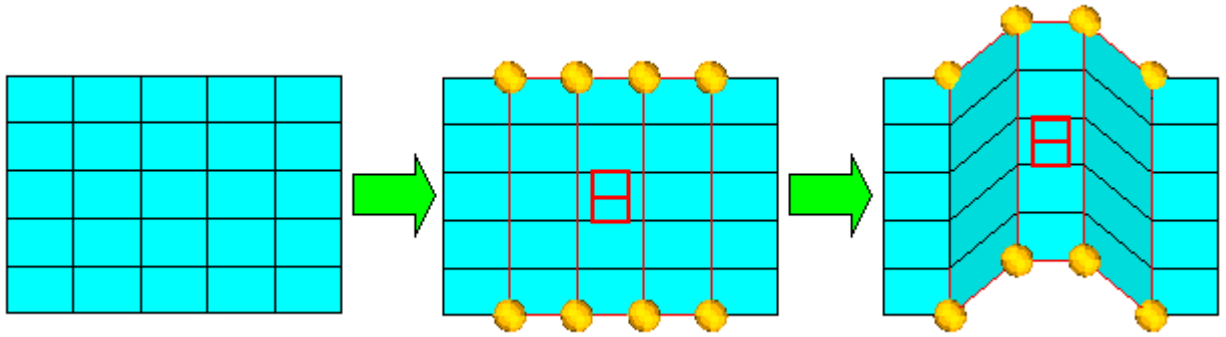
HyperMorph 10.0

Morphing

- Introduction to the Morphing
  - What is Morphing
    - Mesh morphing module in HyperMesh.
    - Allows you to morph an FE model in useful, logical, and intuitive ways which result in minimal element distortion.
  - Why use Morphing
    - Only nodal location is chanced . Node id, element id and any association such as contact groups remain unchanged. Allowing you to modify original mesh to meet new mesh design.
  - Where can I find Morphing module
    - The Morphing module is in HyperMesh -> Tool page -> HyperMoprh panel

Morphing

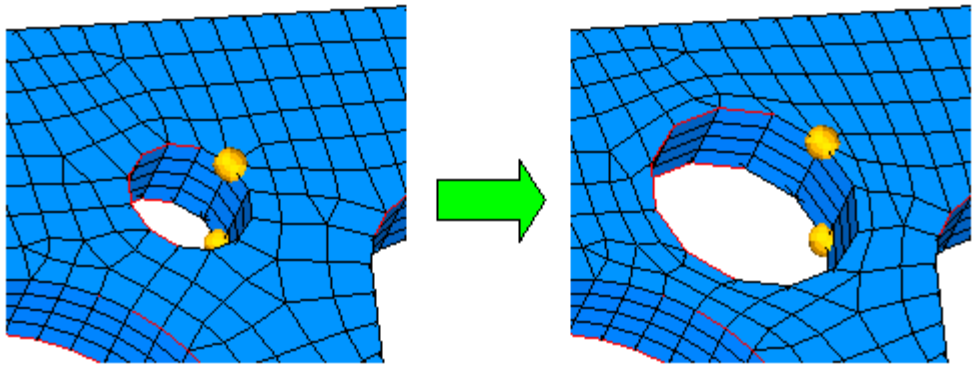
- How does Morphing works
  - Mesh Model is divided into domains
  - Handles are placed at domain boundaries
  - Domain shape is controlled by attached handles
  - Handle movements change domain shapes, which in turn move nodes within domains
  - Global handles affect entire model
  - Local handles only affect parent and neighboring domains
  - Map to geometry



Morphing

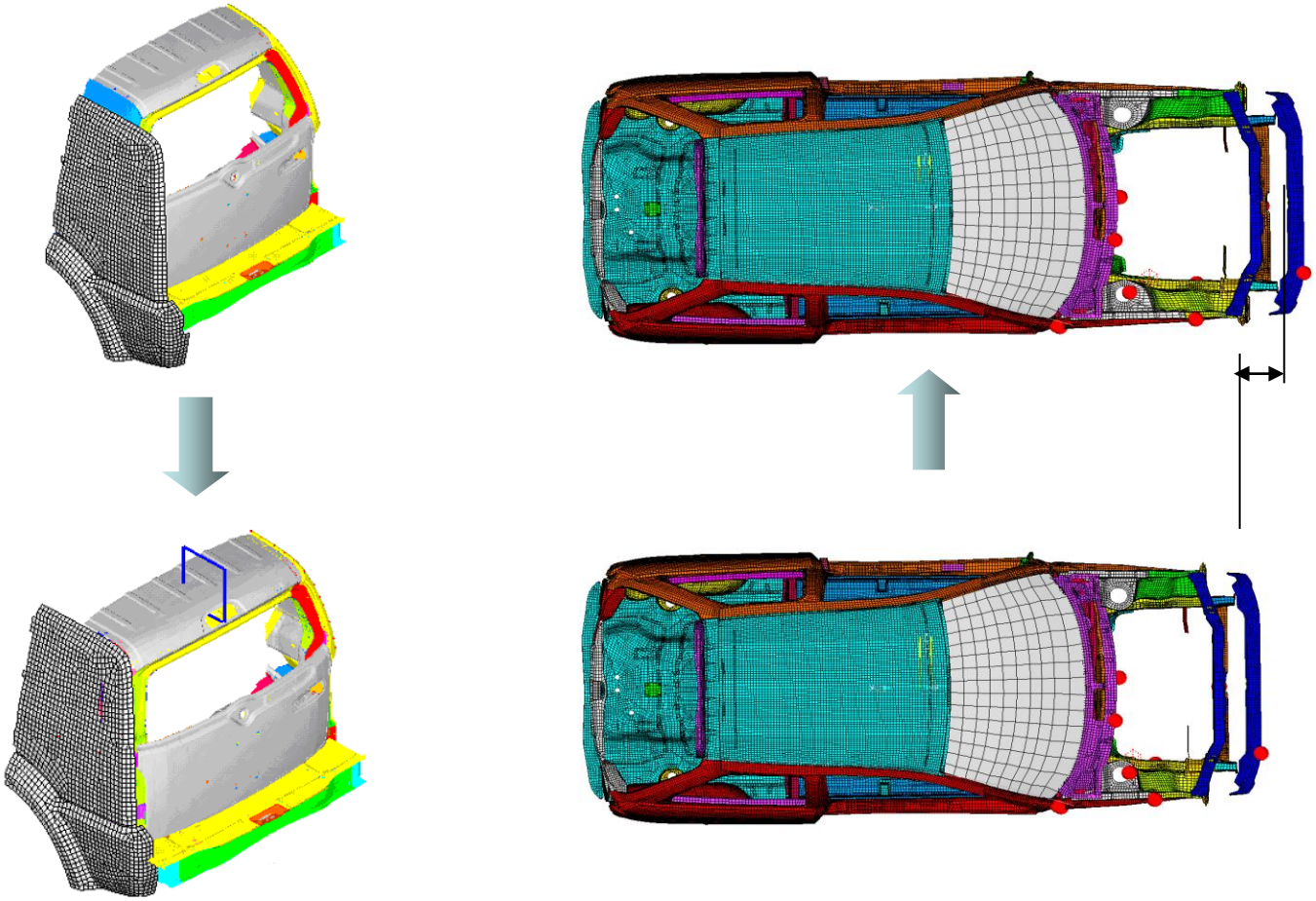
- What are Morphing applications
  - Rapidly change shape of existing model
  - Improve element quality by dragging handles or mapping edge domains
  - Fit old model to new design data
  - Map an existing mesh onto lines or surfaces
  - Generate NURB surfaces using FE -> Surf feature in HyperMesh
  - Generate and edit shape variables for optimization

Application A: Easily alter the diameter of holes for solid models



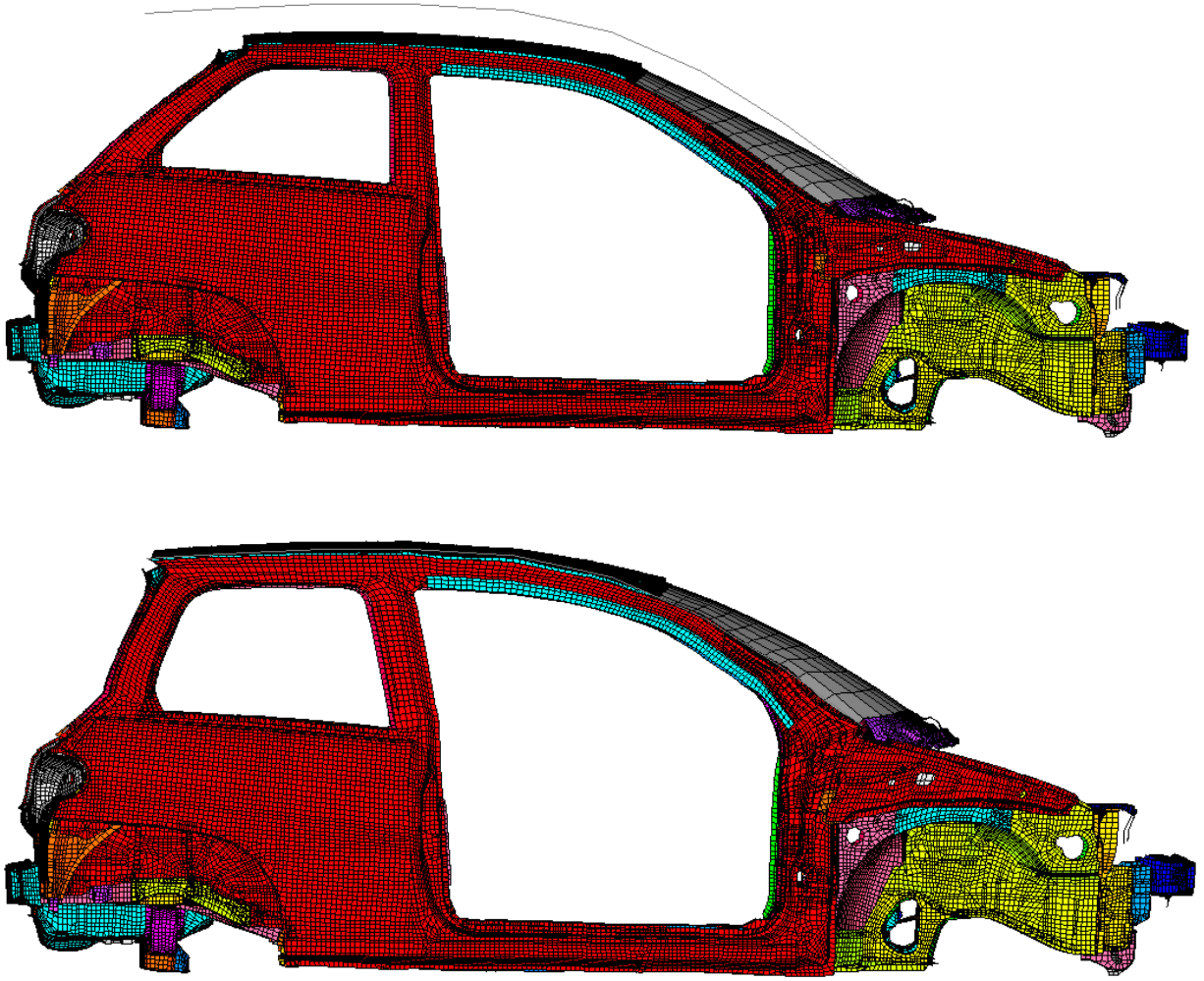
Morphing

- Application B: Rapidly stretch the full vehicle body



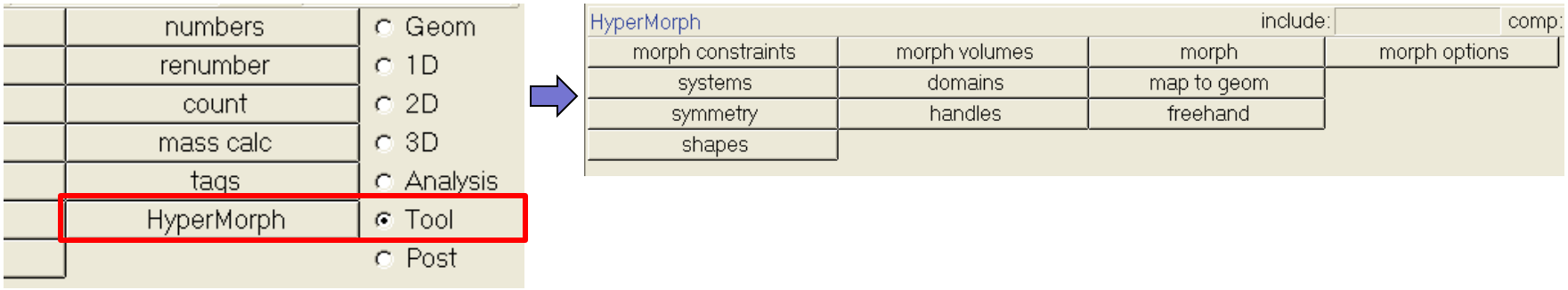
Morphing

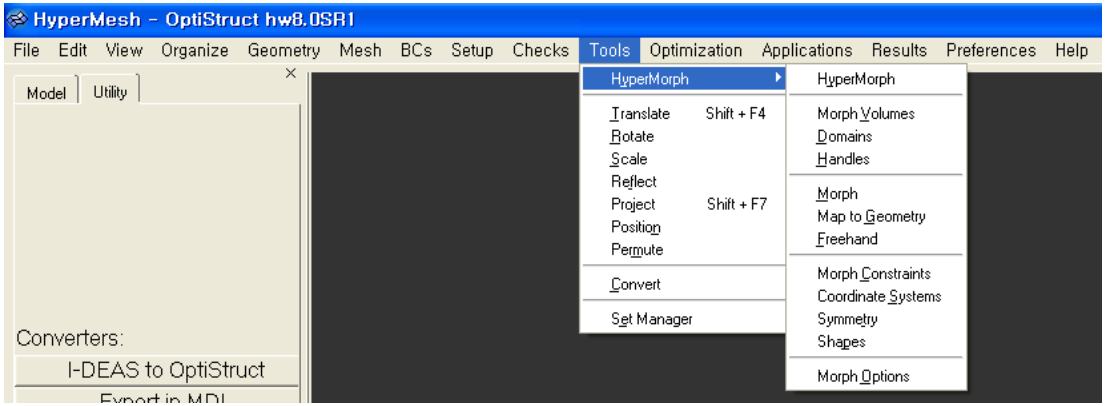
- Application C: map to geometry



Morphing

- Tools for Morphing





Morphing

- Tools for Morphing

  - Morph constraints: Create/update/release constraints to morph a CAE model.
  - Systems: Create, edit and update system
  - Symmetry: Create, edit and update symmetry. Update domains/handles to symmetry
  - Shapes: Create/apply/autoshape/convert shape into loads/save shape/apply the saved shape to another model.
  - Morph Volume: Create/edit & update/save & export-import/convert HEXA into morph volume
  - Domains: Create/edit/update domains and setup parameters
  - Handles: Create/edit/update handles and dependency or save-to load-from a file
  - Morph: morph the model and create shape entities.
  - Map to geom: map domains and handles to geometric data.
  - Freehand: Easy way of morphing. Good for quick change and bead creation.

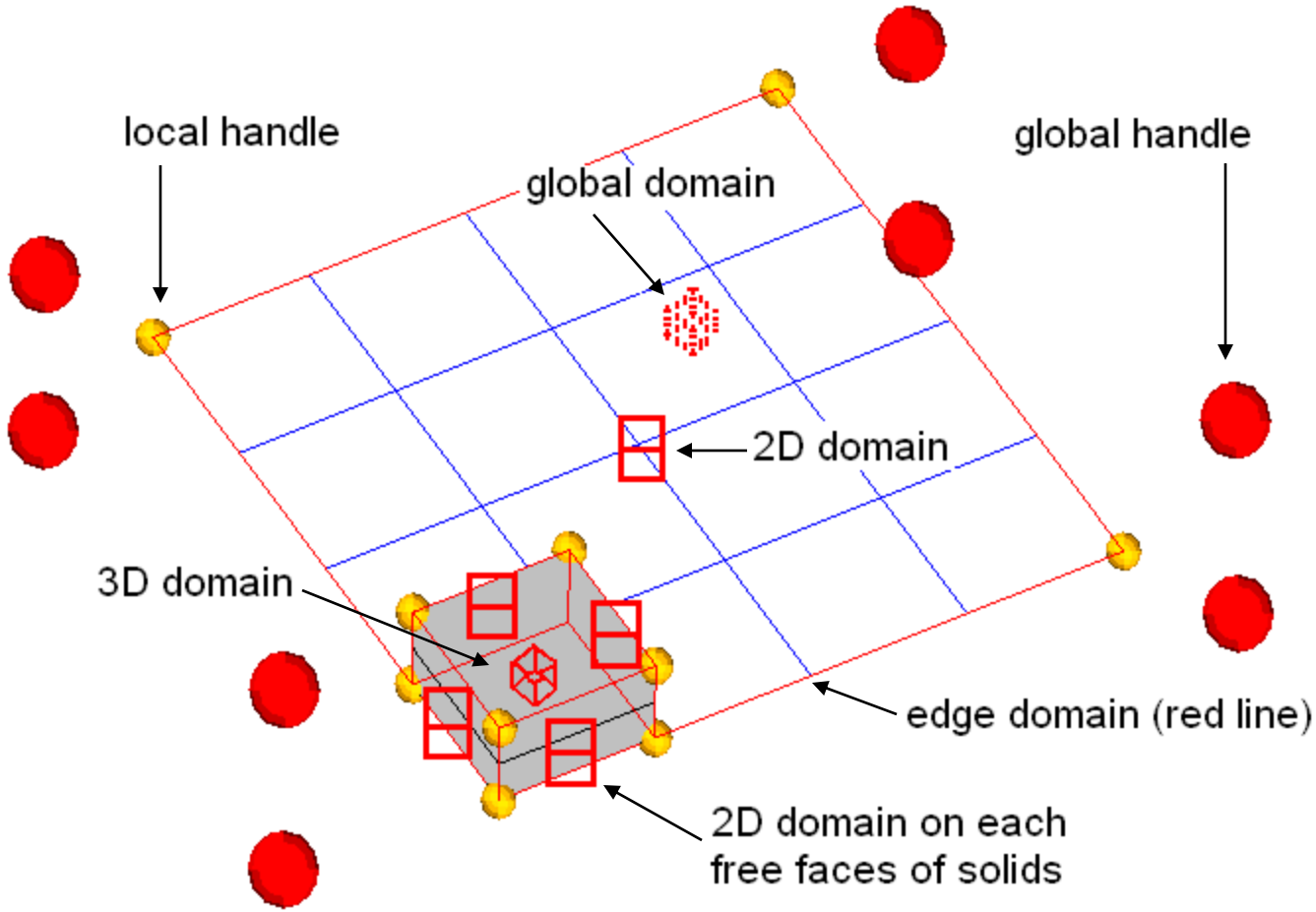
Morphing

- Agenda :
  - Introduction to HyperMorph terminology
  - HyperMorph features
  - Morphing process
  - Strategy and examples

Introduction to HyperMorph terminology

Introduction to HyperMorph terminology

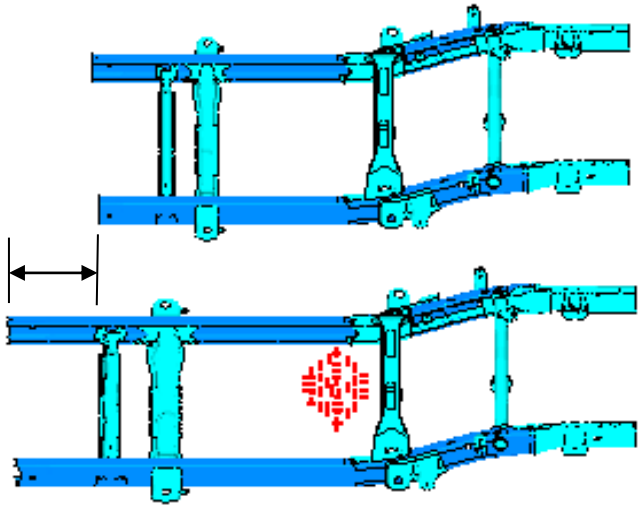
- What does HyperMorph look like?



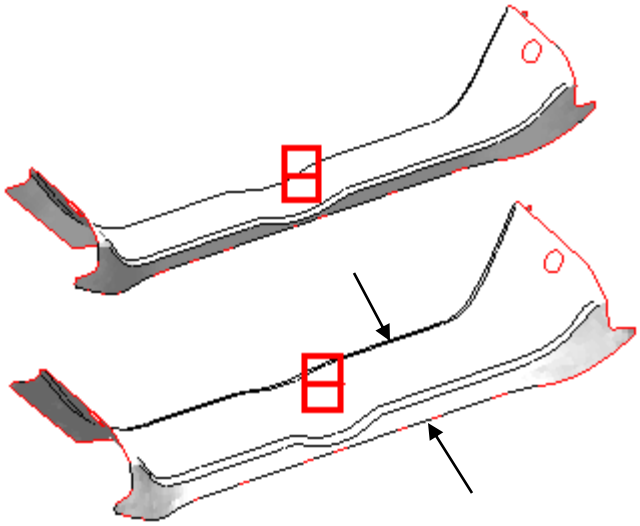
Introduction to HyperMorph terminology

- Domain : entity comprises elements and nodes as a part of morphing process.
- Global domain : a single domain which can influence every node in the model.
- Local domains : include1D domain, 2D domain, 3D domain and edge domain. A model can have multiple local domains for morphing different local areas.

example of global domain



example of local domain

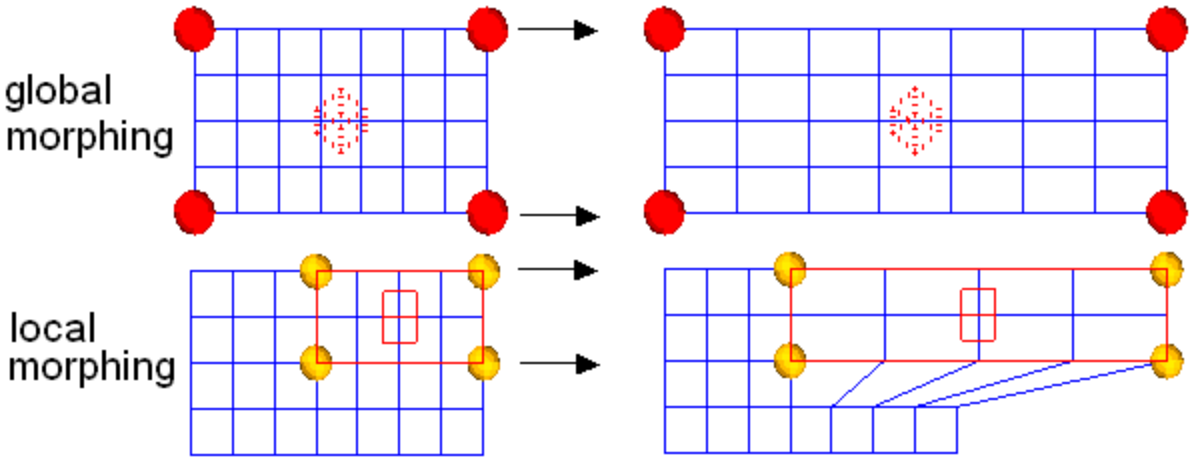


Introduction to HyperMorph terminology

Domain type	Content	Symbol in HM
1D domains	Contain a group of 1D elements such as bars and rigid elements.	
2D domains	Contain a group of shell elements	
3D domains	Contain a group of solid elements.	
edge domains	Contain a series of nodes and are commonly found along the edges of 2D and 3D domains.	Red lines around the edges of all 2D domains
global domain	Consists of the entire model.	

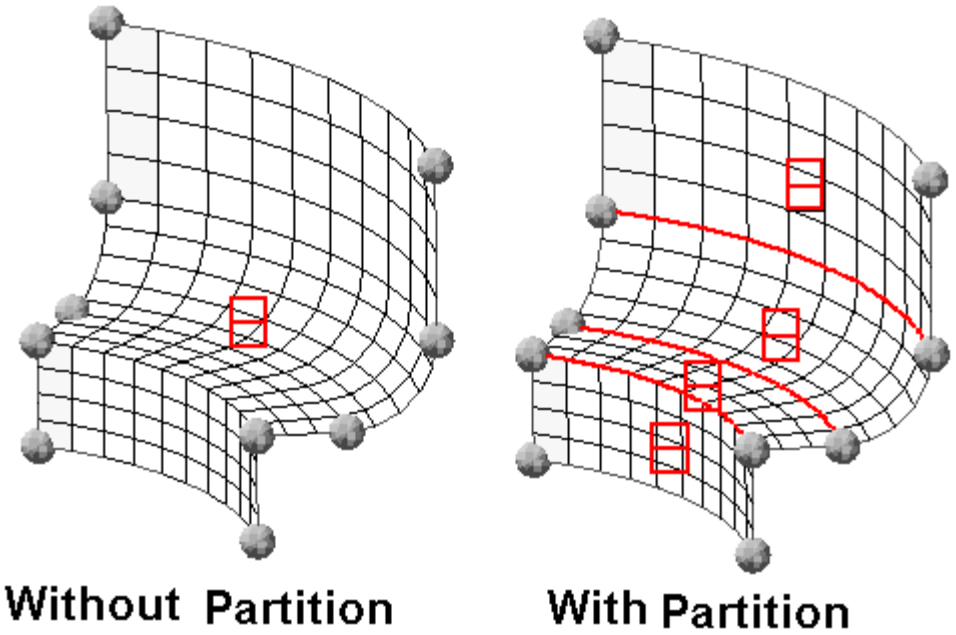
Introduction to HyperMorph terminology

- Handle : accompany each domain and provide the mechanism to modify shape of a mesh
- Handle influence : describe how a movement of a handle relate to nodal movements.
- Global handle : only exist in global domain. Movement of a global handle can affect every node within a model. It allows a large scale shape change.
- Local handle : only exist in local domains. Any local handle can only influence nodes contained in the local domains they are associated with. It is used for local shape changes.
- Global morphing : morphing using global domains and global handles.
- Local morphing : morphing using local domains and local handles.



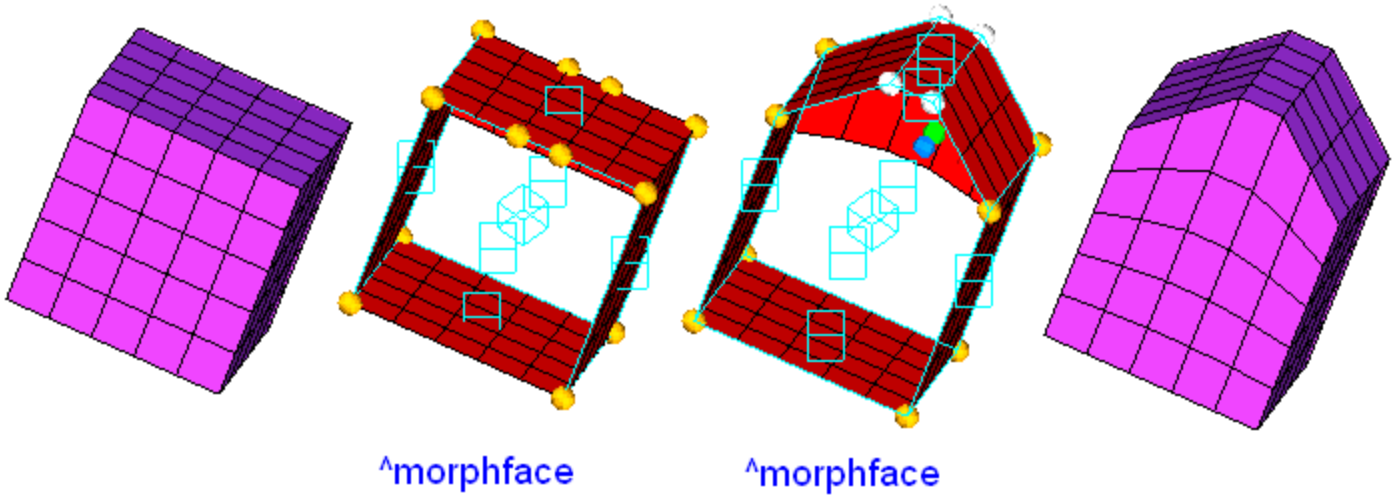
Introduction to HyperMorph terminology

- Domain angle : the angle between the normal vectors between two elements. When the value is exceeded, a partition break is confirmed and a new domain will be created with an edge running between the two elements.
- Curve tolerance : a parameter used to decide if a mesh geometric feature is straight or curve. Similar to domain angle, a partition is performed when the value is exceeded.
- Partition : a HyperMorph term to logically divides a 2D domain into smaller 2D domains based on the values of domain angle and curve tolerance.



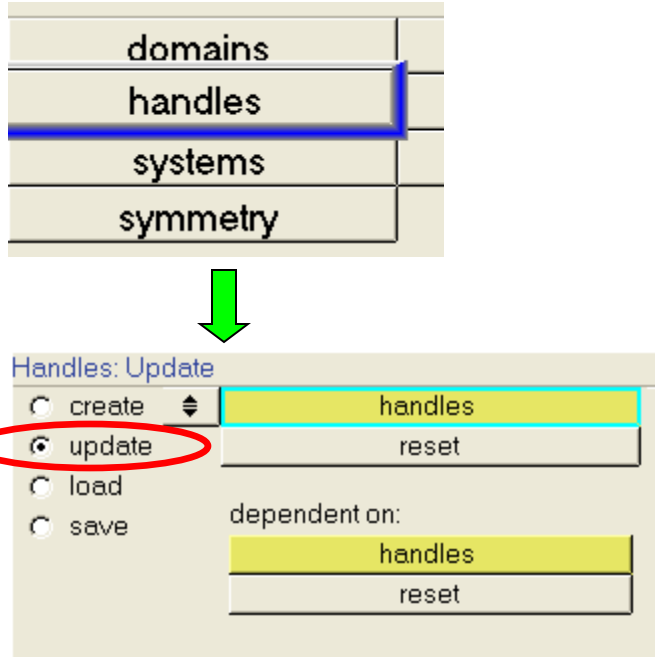
Introduction to HyperMorph terminology

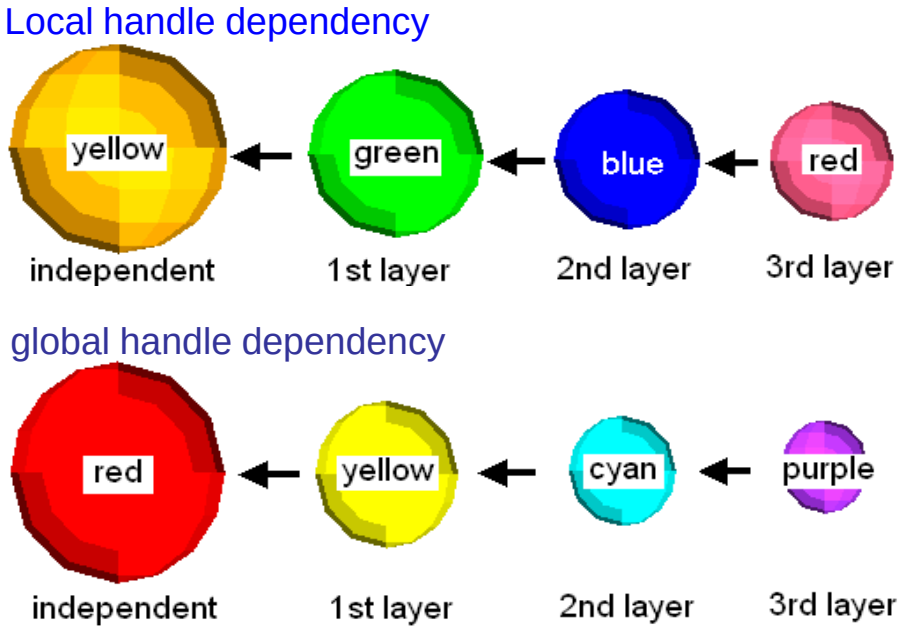
- ^morphface : 2D elements on the faces of each 3D domain and placed into a ^morphface component. Any morphing operation on those face elements within ^morphface influences underneath solid elements. Essentially, to morph solid elements is to morph elements within ^morphface.



Introduction to HyperMorph terminology

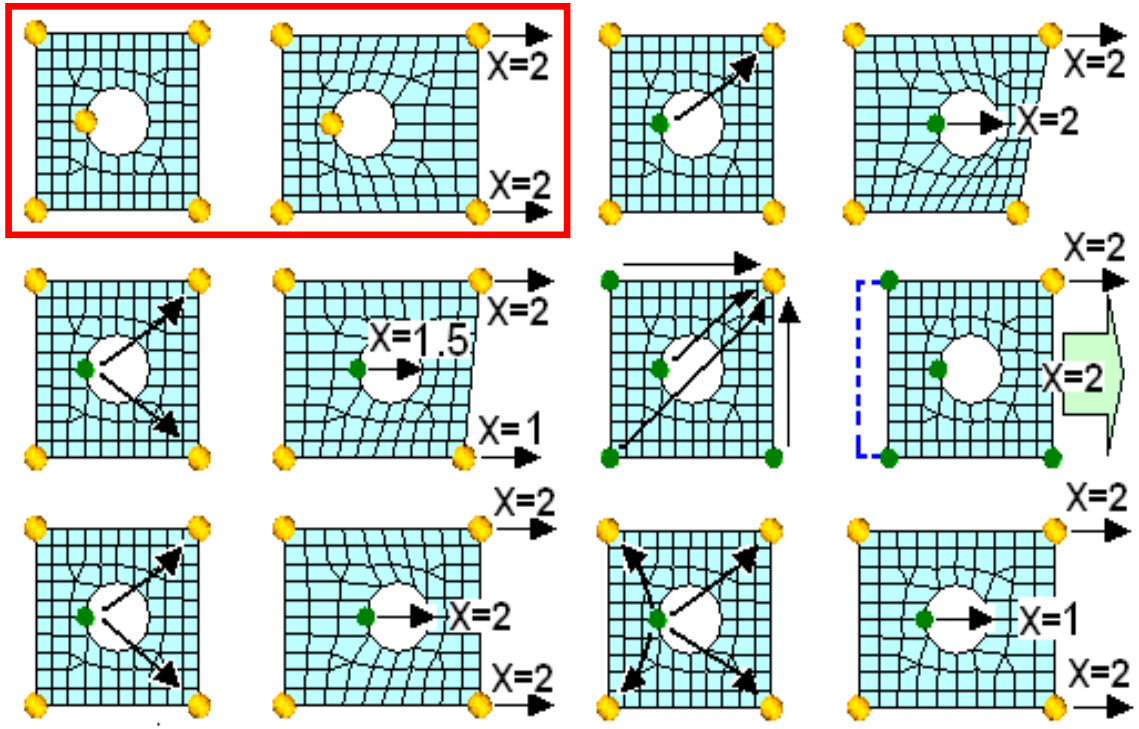
- Dependency : a HyperMorph feature which can be used to build relationships among handles. Multiple layers of dependency is supported.
- Independent handle : the handle is only morphed by its own movement and independent from other handle movement.
- Dependent handle : the handle is affected by the movement of its associated independent handle.





Introduction to HyperMorph terminology

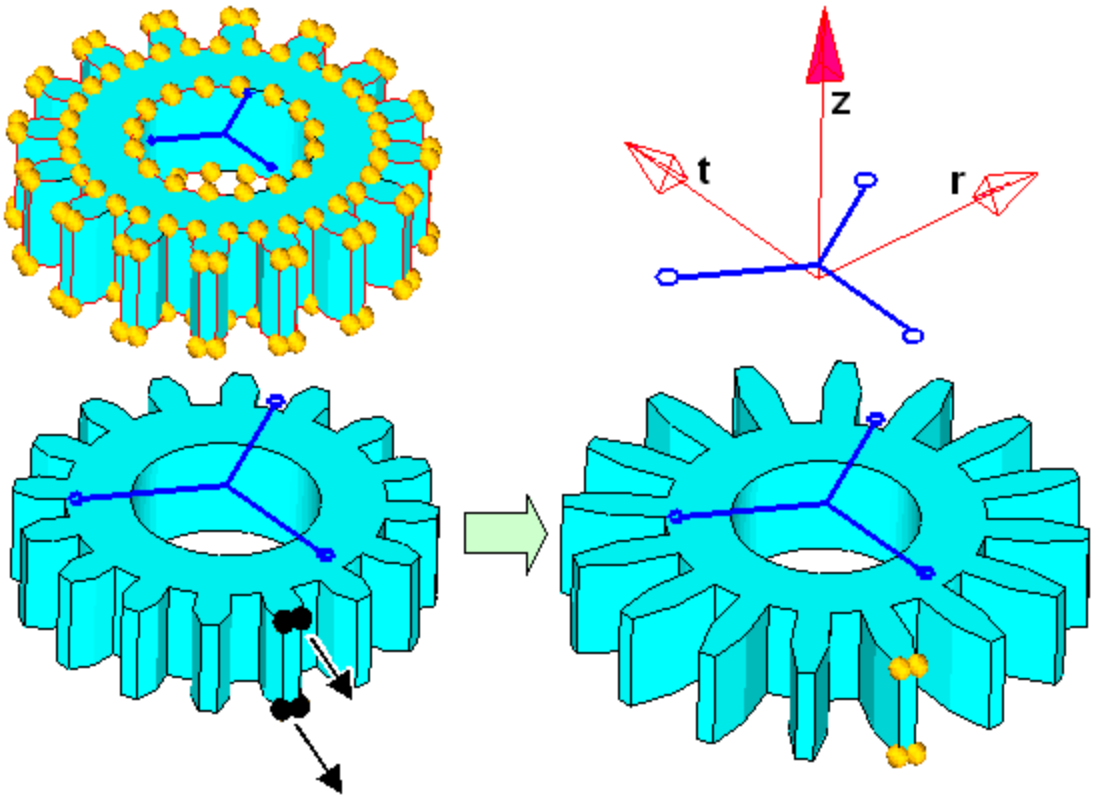
No dependency



Introduction to HyperMorph terminology

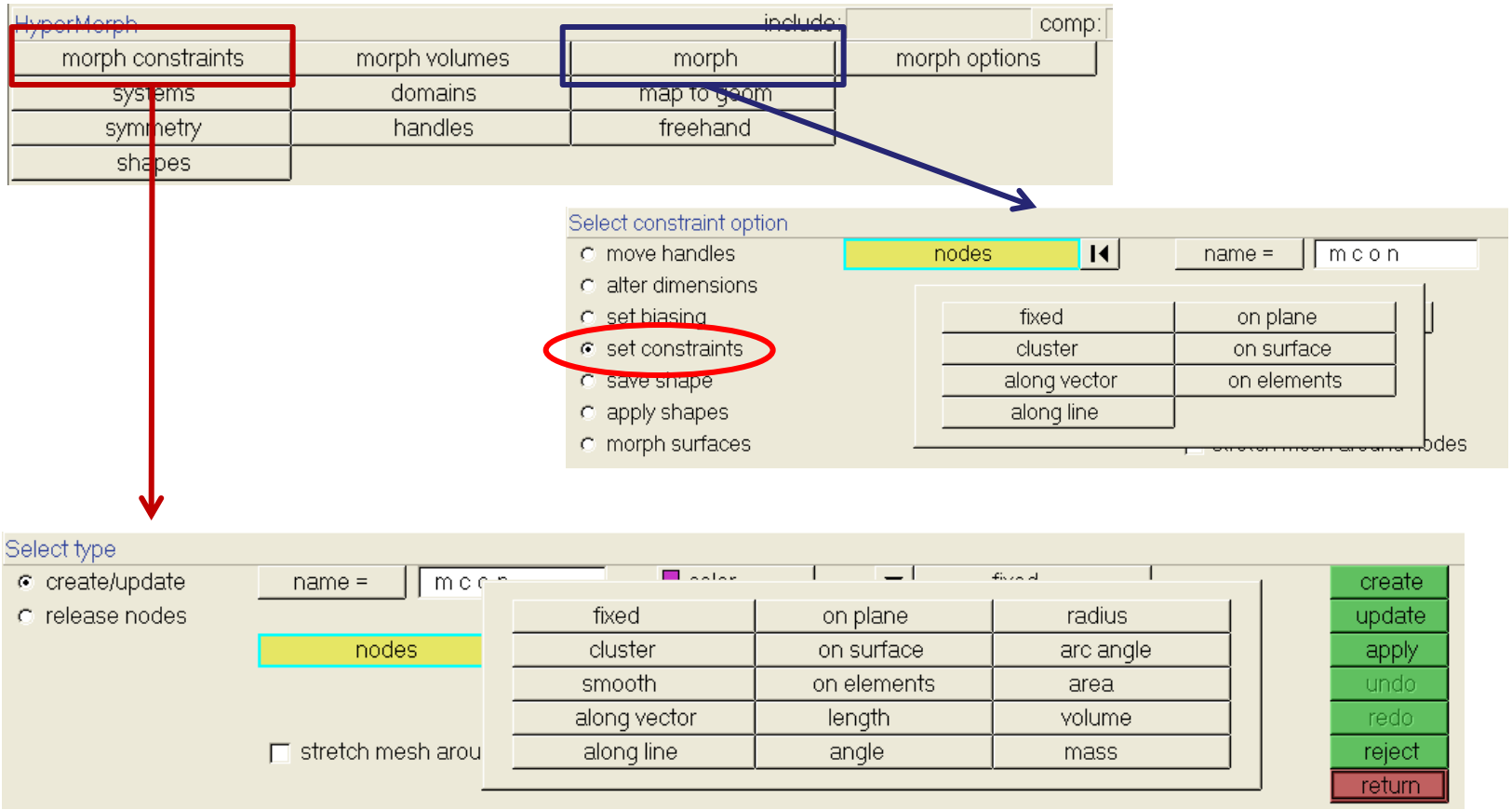
- Symmetry : a HyperMoprh entity allows users to link handles in a symmetric fashion The movements of one handle will be reflected and applied to the symmetric handles.

Cyclical symmetry



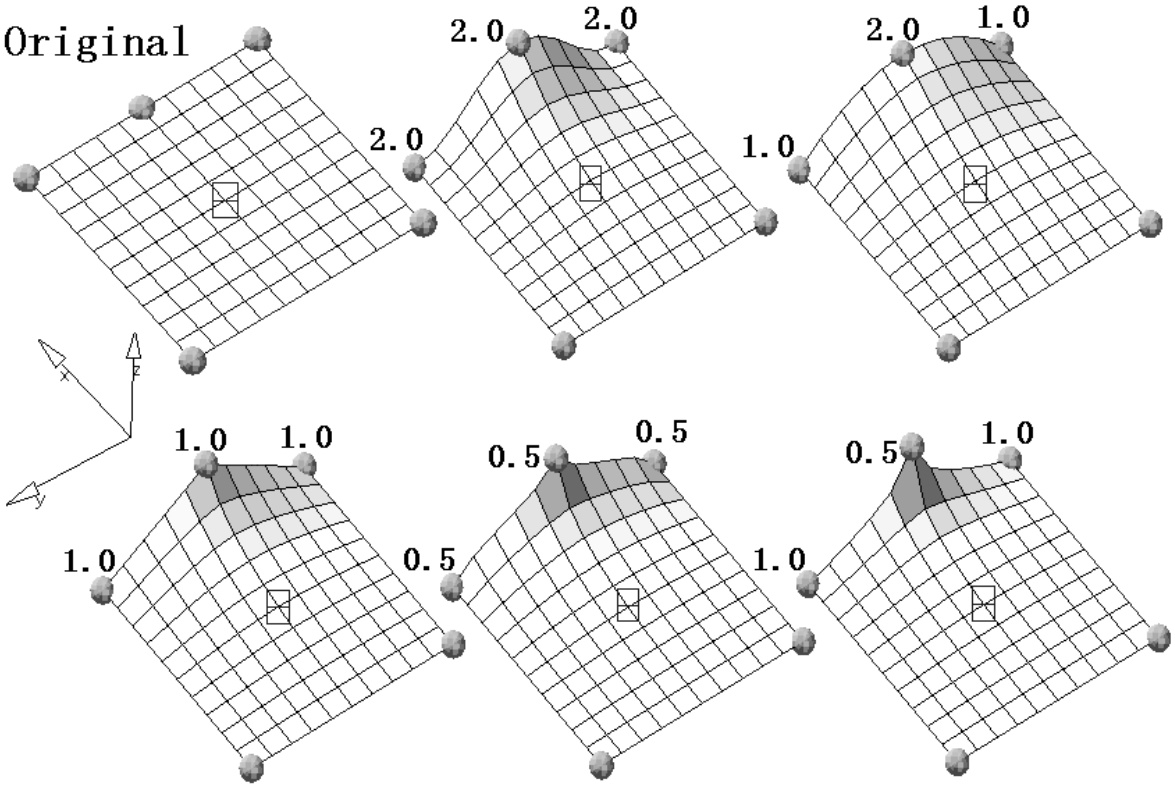
Introduction to HyperMorph terminology

- **Constraints** : a HyperMorph feature to restrict the movement of nodes during morphing operations.



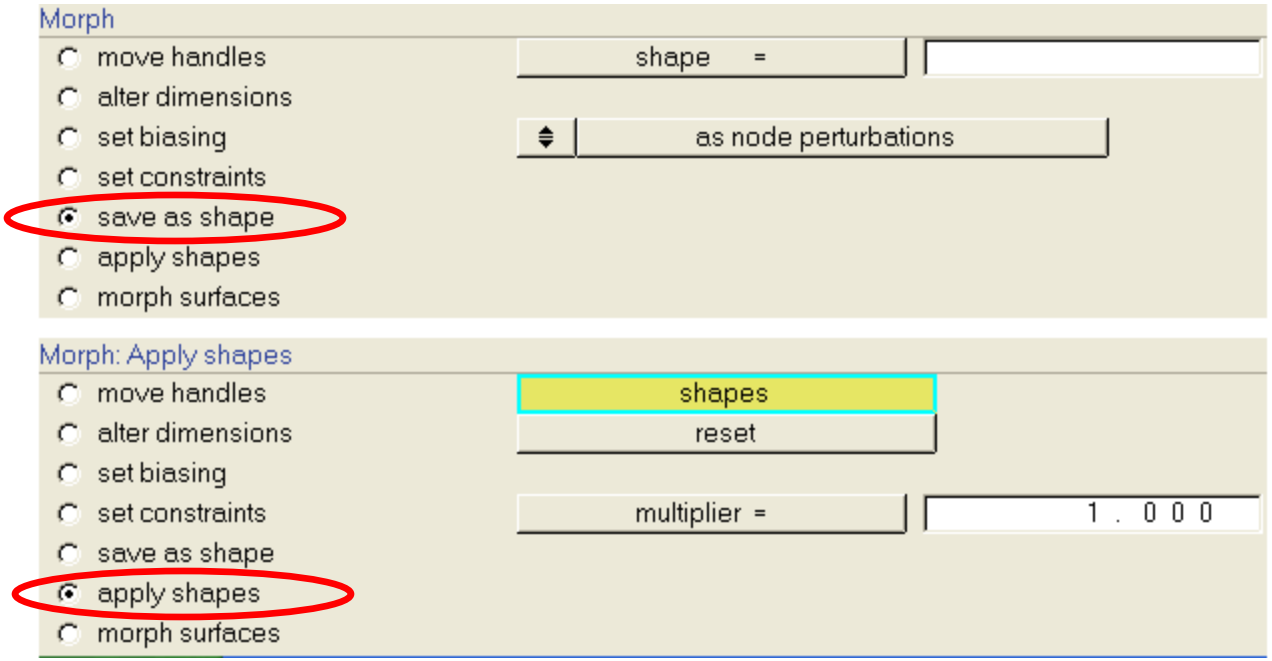
Introduction to HyperMorph terminology

- Biasing : a HyperMoprh feature to modify the influence of a handle over the nearby nodes. A biasing factor can be assigned to a handle. Higher bias value increase the influence of a handle over nodes. Lower bias value decrease the influence. The default value of each handle is 1.0 with linear influence.



Introduction to HyperMorph terminology

- Shape：a HyperMoprh entity records the difference between the initial state of the model and the current state of the model. It can be used for storing, re-applying and combining multiple mesh changes. It can also be linked to optimization code to perform shape optimization.

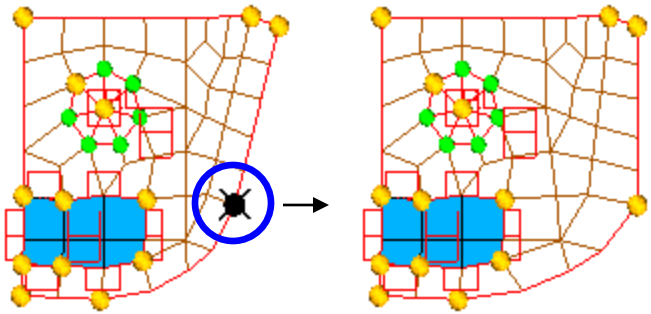


HyperMorph Features

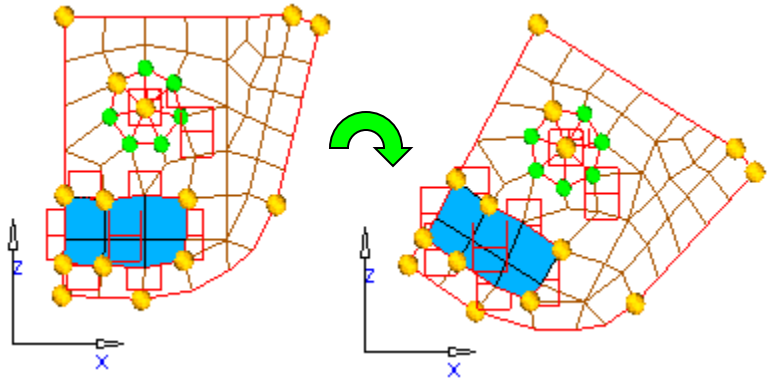
HyperMorph features

- Perform morphing operation by move handles

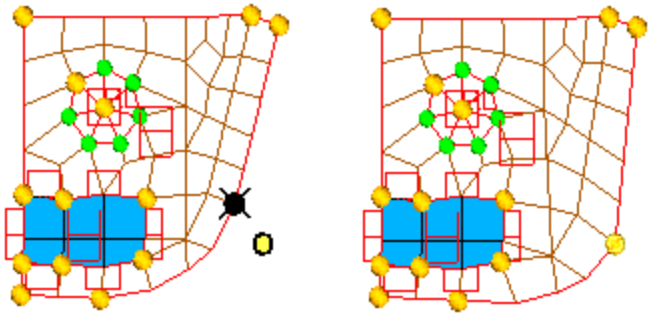
Morph interactively by dragging handles across graphics area



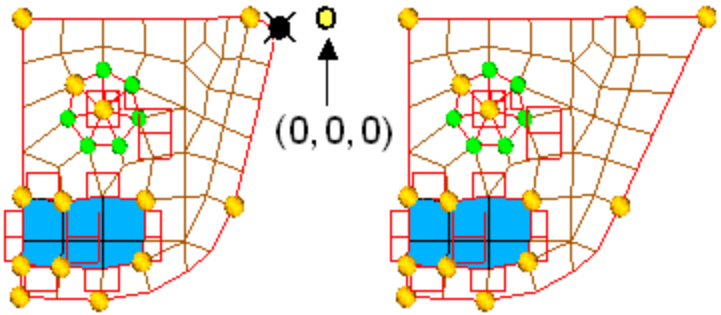
Rotate a mesh



Translate a handle to a node



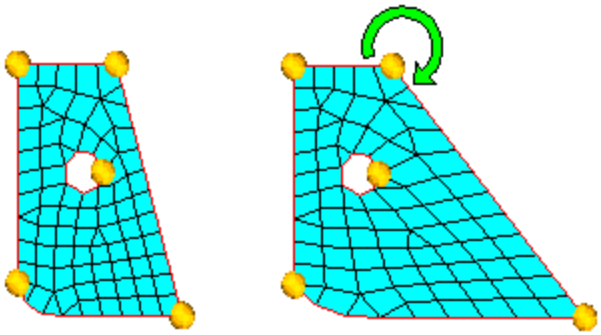
Translate a handle to a coordinate



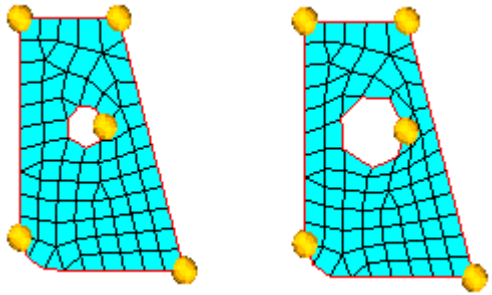
HyperMorph features

- Perform morphing operation by alter dimension

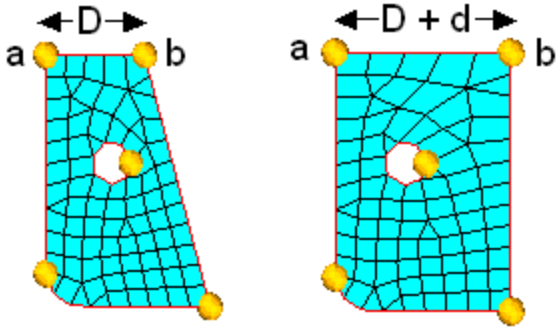
alter dimension (angle)



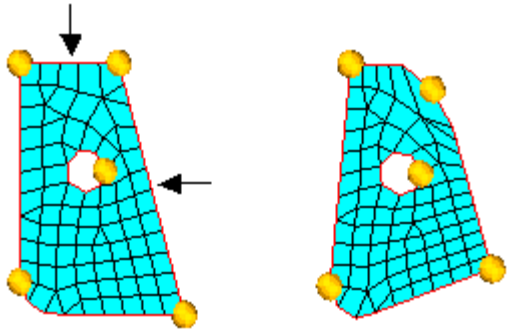
alter dimension (radius)



alter dimension (distance)

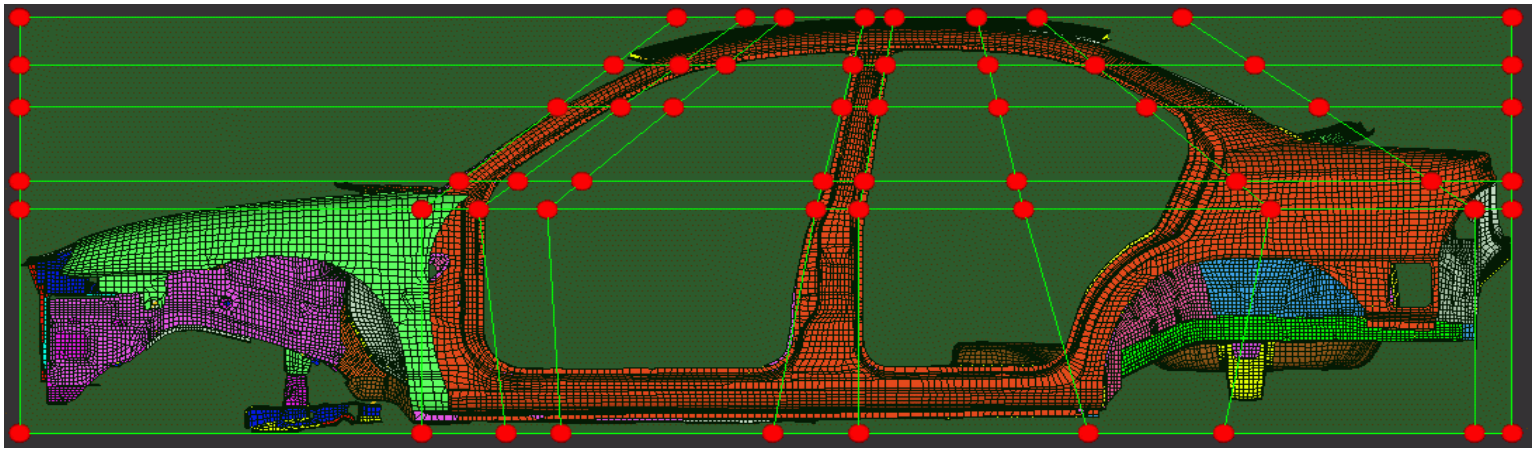


alter dimension (curvature)



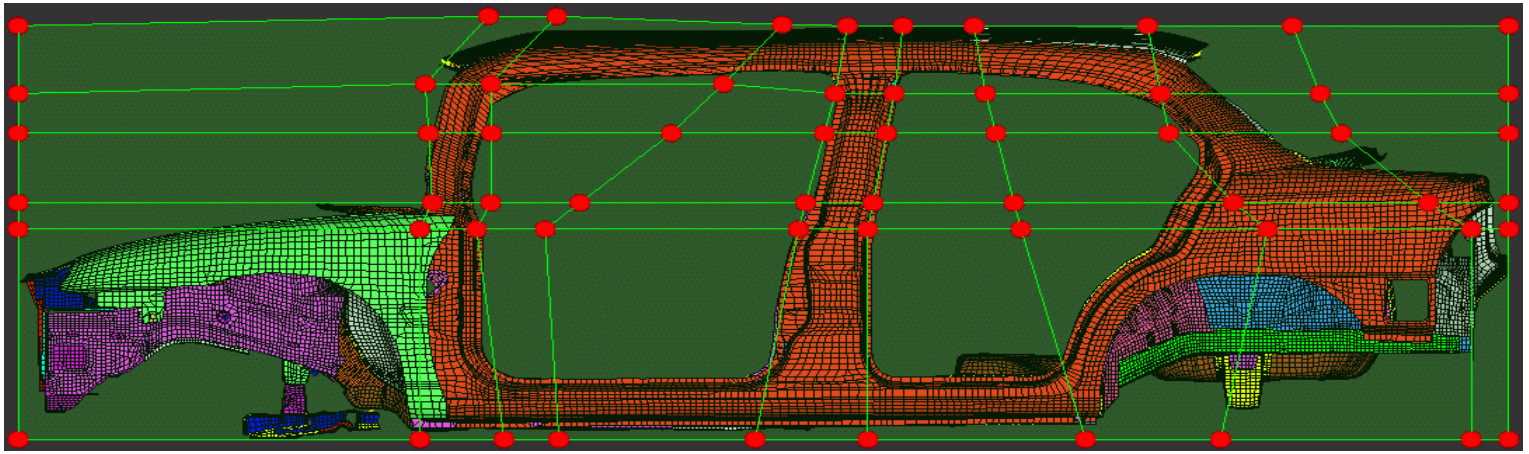
HyperMorph features

- Perform morphing operation by VolumeMorph



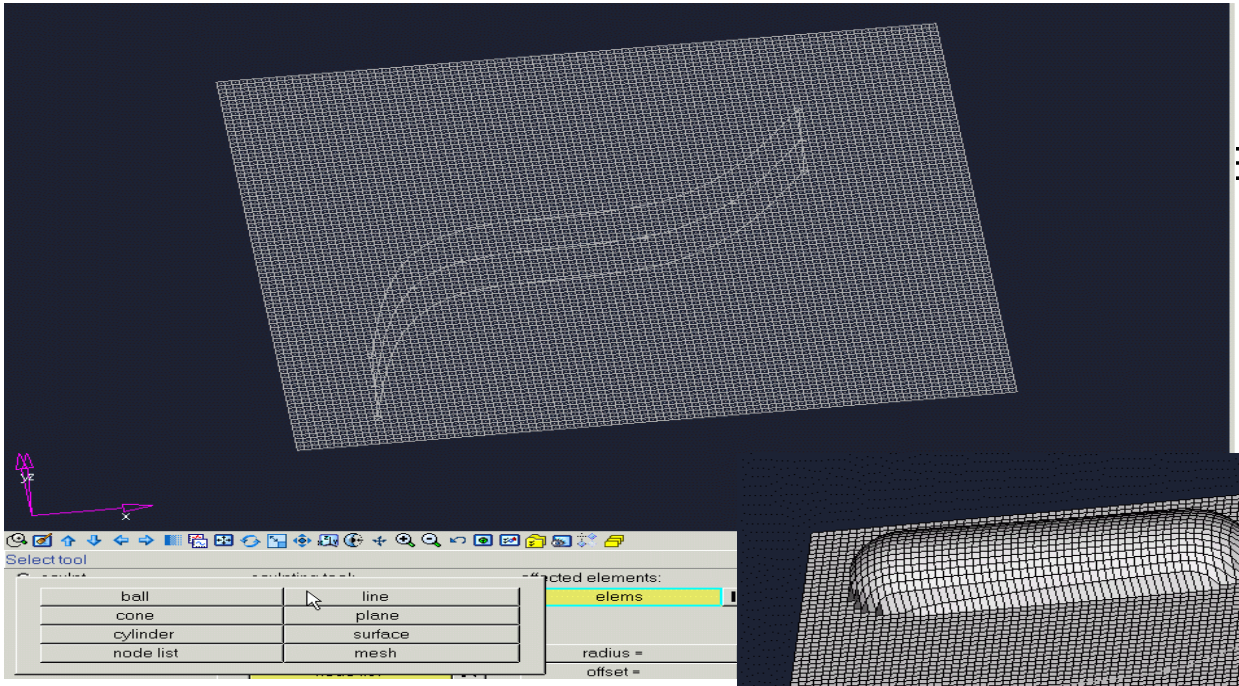
Before

After



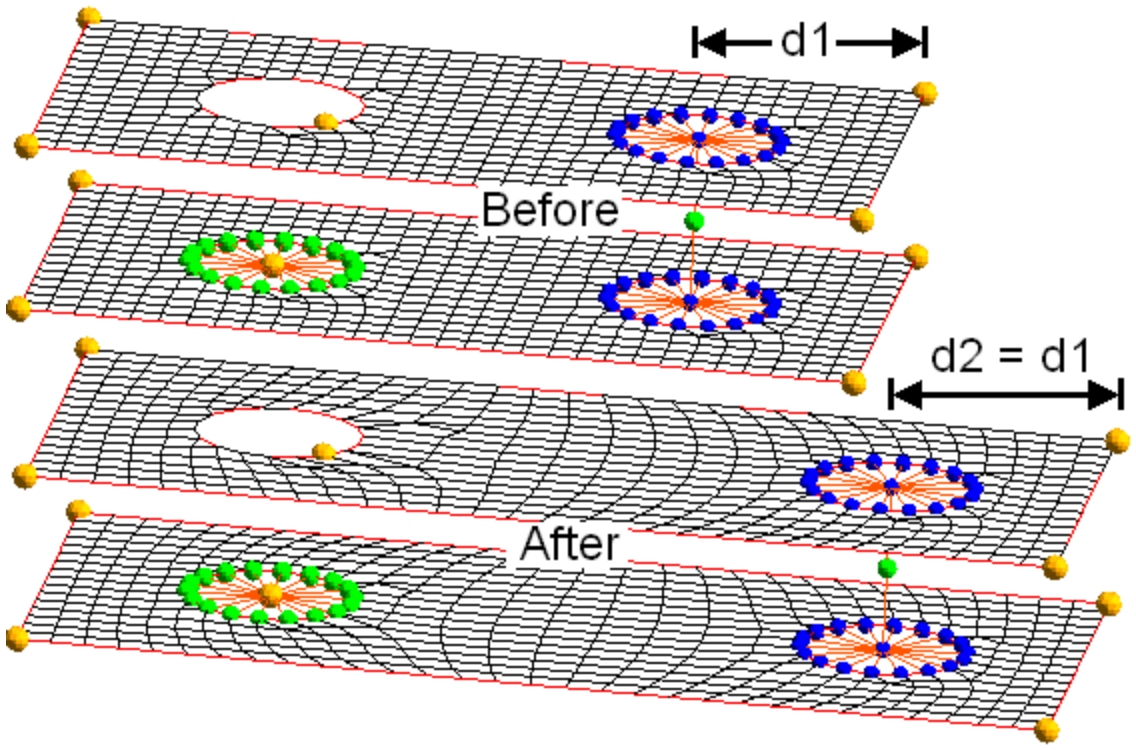
HyperMorph features

- Perform morphing operation by Bead insertion with Freehand



HyperMorph features

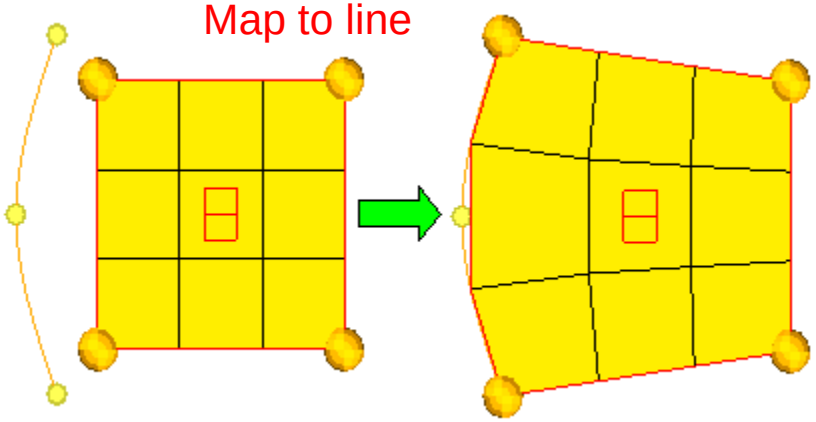
- Perform morphing operation with 1d elements and dependency



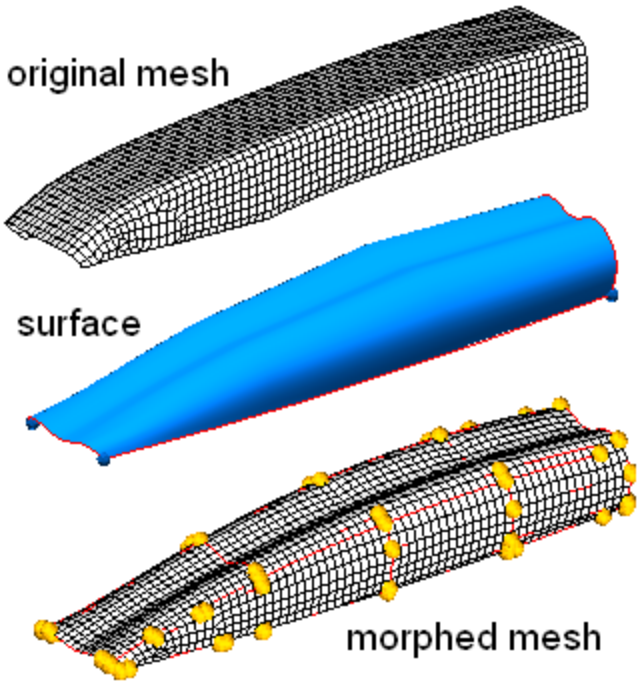
HyperMorph features

- Map to geometry

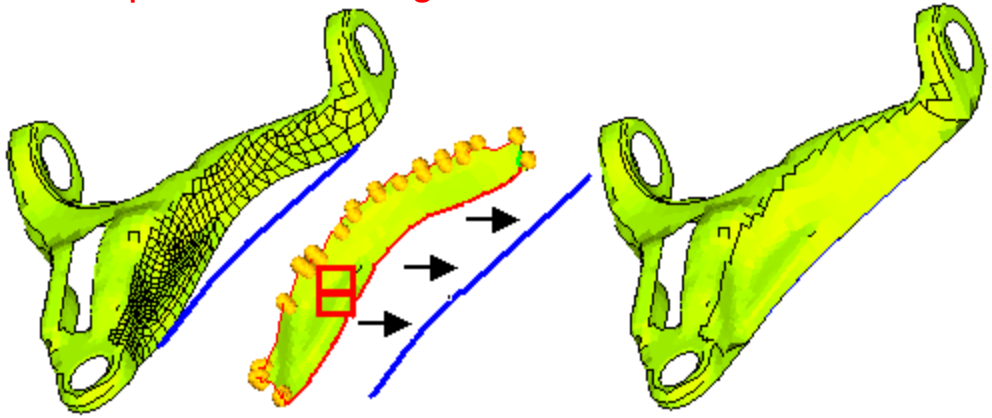
Map to line



Map to surface

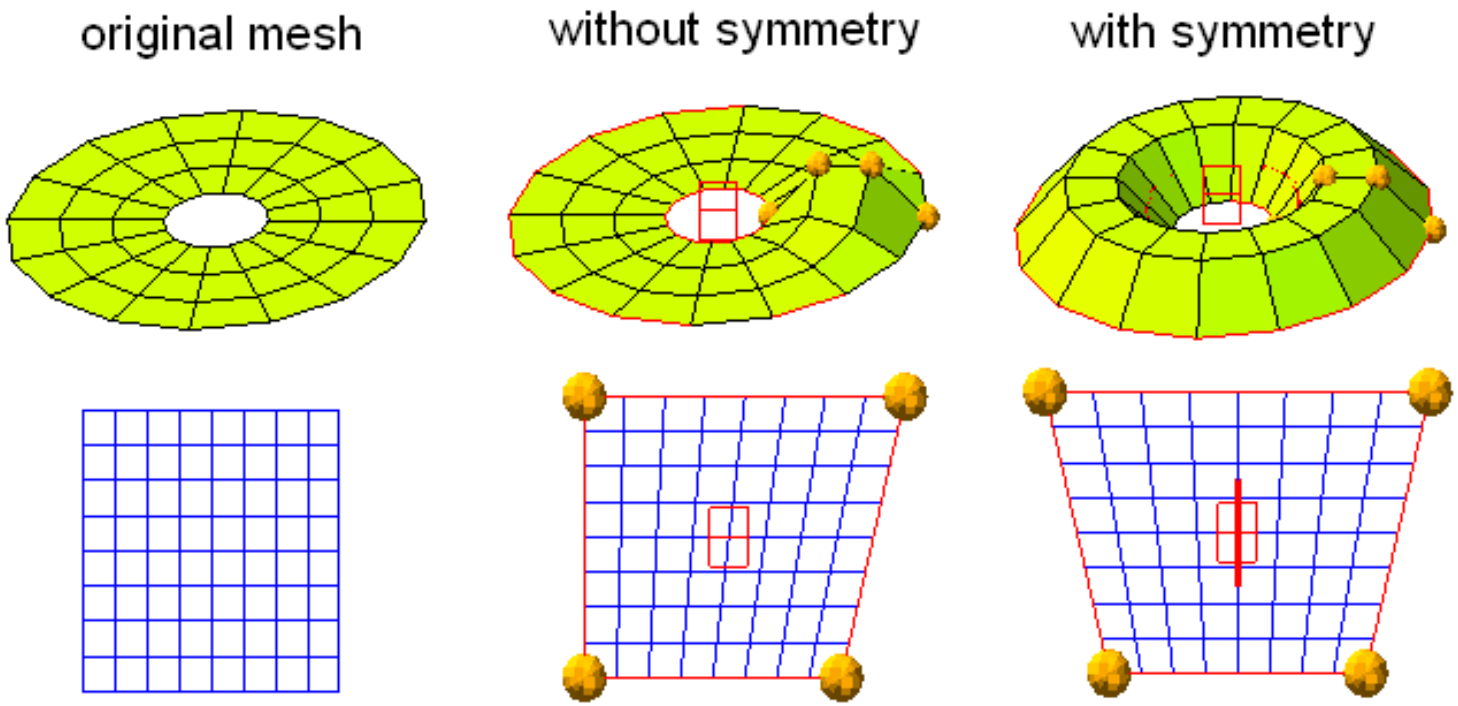


Map to surface edge



HyperMorph features

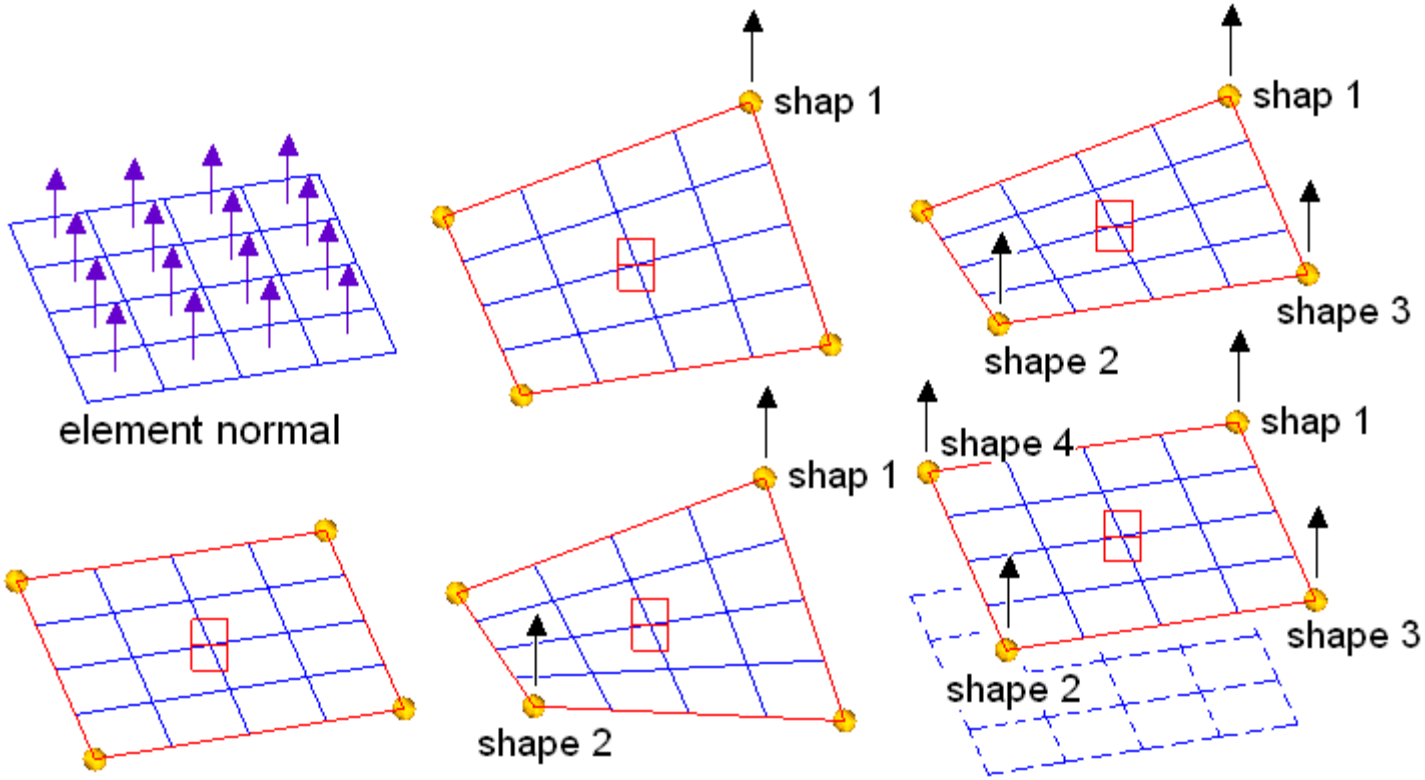
- Perform morphing with symmetry



HyperMorph features

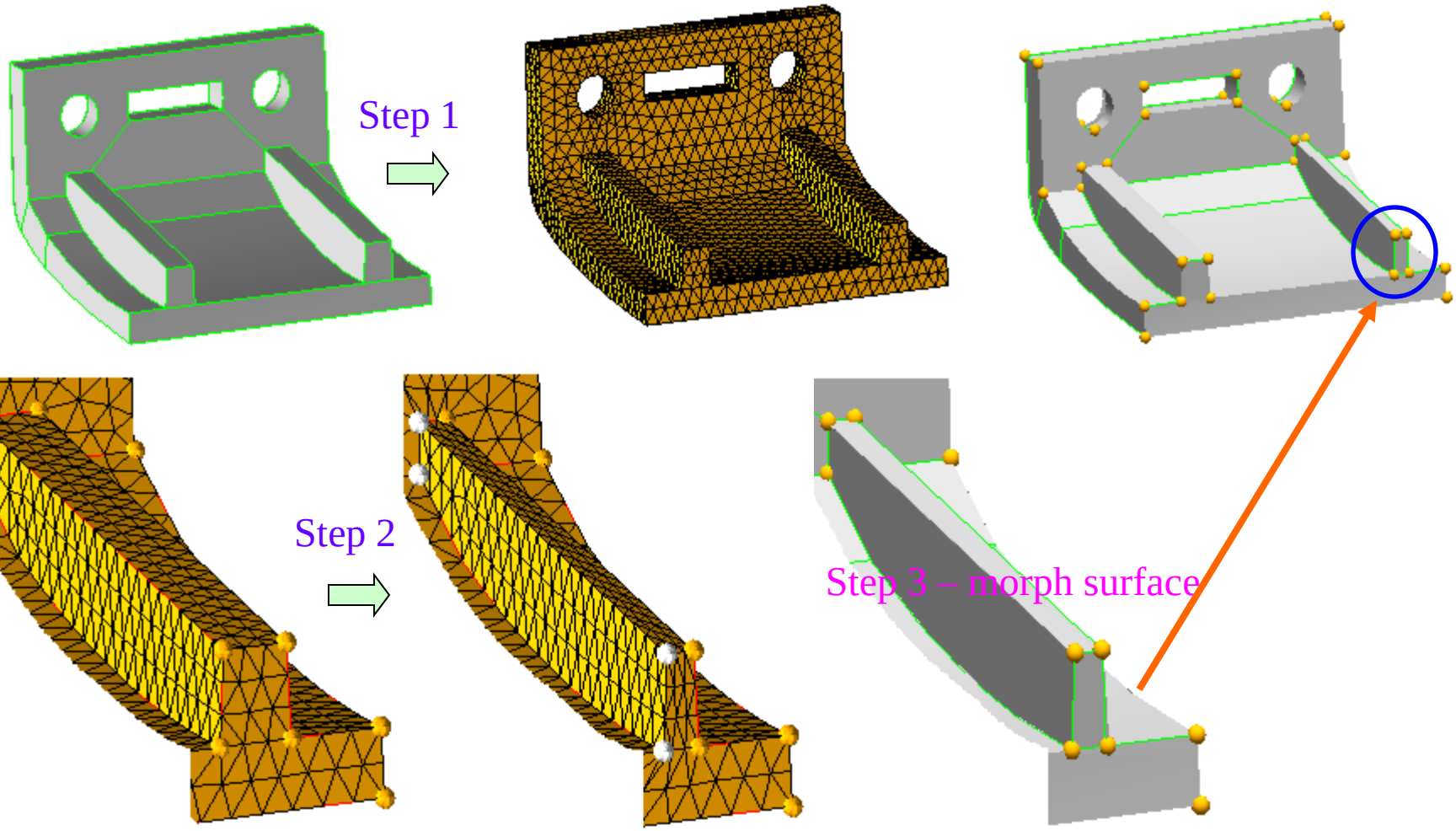
- AutoShape

Morph handle according to element normal or vector
Generate shape variables for optimization
Store or combine multiple shapes



HyperMorph features

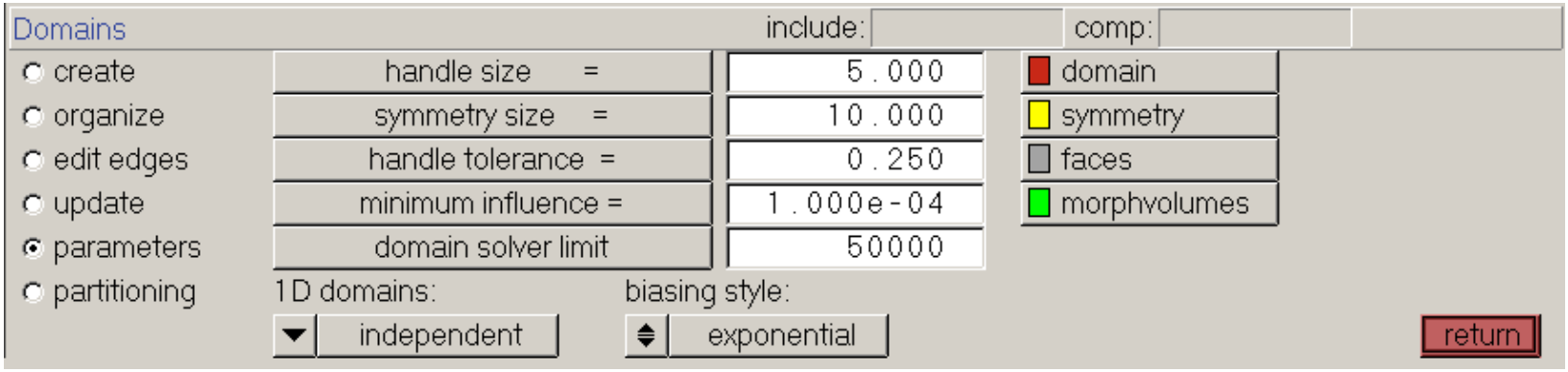
- Morph surface



Morphing process

Morphing process

- Outline of the process
  - Step 1 – Load a mesh model
  - Step 2 – setup parameters



Morphing process

If apply global morphing : If you wish to preserve the local geometry, the hierarchical method should be selected. If you wish to do a large scale change with a tolerance to bend and distort the local geometry, choose the direct method.

* Global Domains and Handles

If apply local morphing : 1d domains

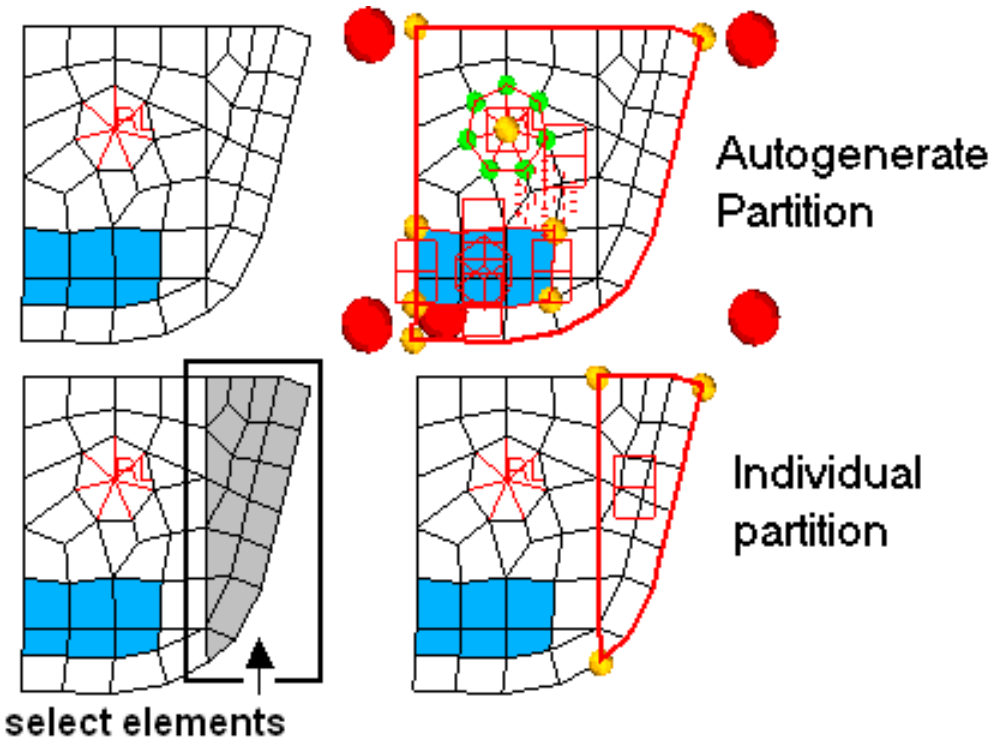
2d domains

3d domains

Edge domains

Morphing process

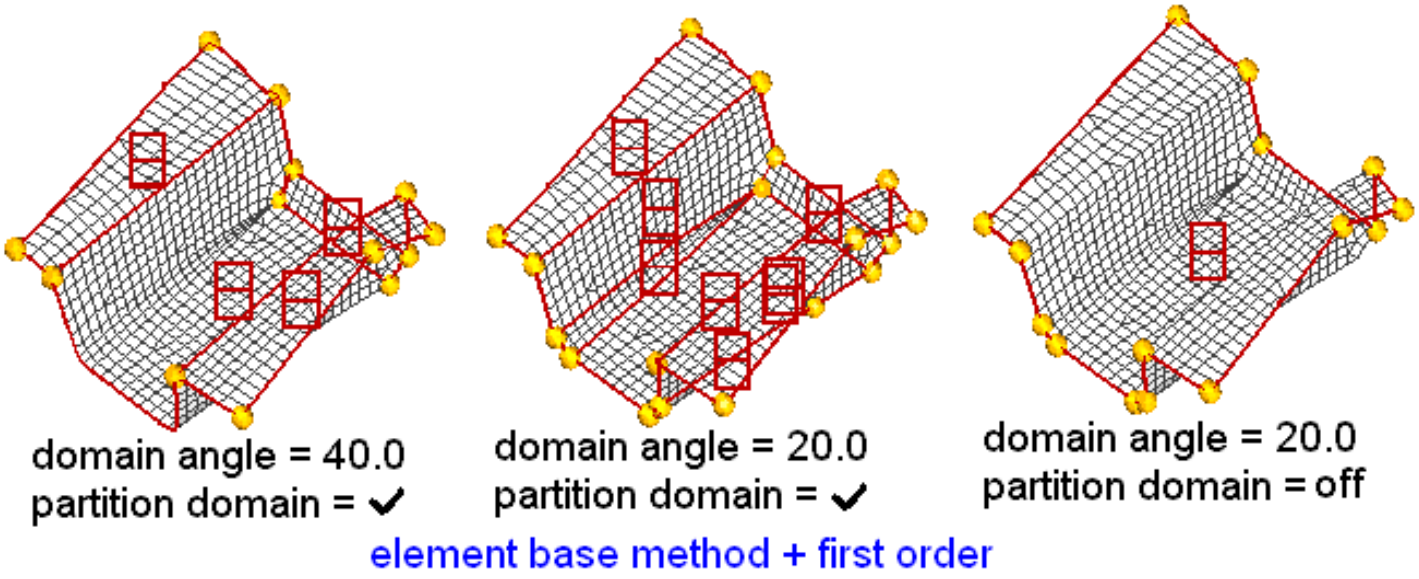
- Step 3 – create domains and handles
  - autogenerate : automatically create all domains and handles. Good for simple geometry.
  - individual partitions : select only local areas for partitioning. Usually generate fewer domains and handles. Recommended for experienced users.



Morphing process

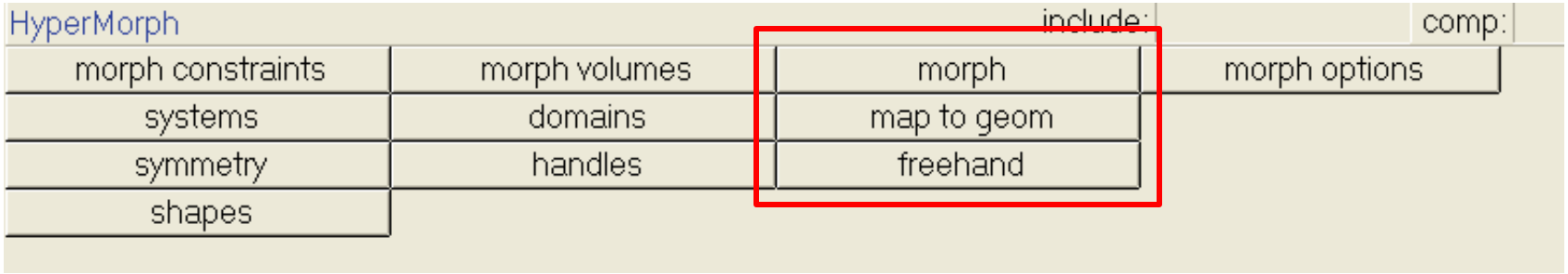
- Step 4(optional) – refine partition

  To re-create, edit, merge, or delete domains and handles. Using different parameters to re-partitioning domains to be able to build desired handles and domains.



Morphing process

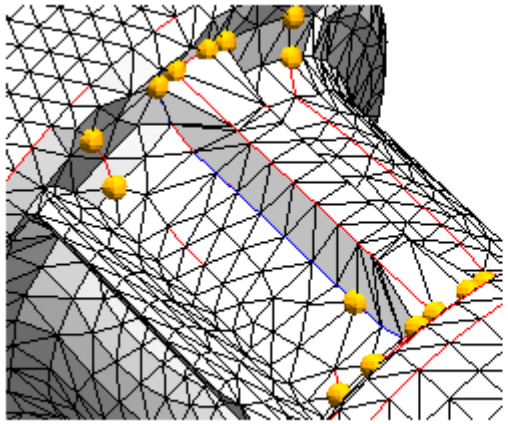
- Step 5 – Morph
- Move handles : move handles to morph a mesh.
- Alter dimension : select a dimension to change its value. This allows a precise modification of a dimension
- Map to geom : map nodes or domain to existing geometry
- Freehand : Easy way of morphing. Good for quick change and bead creation.



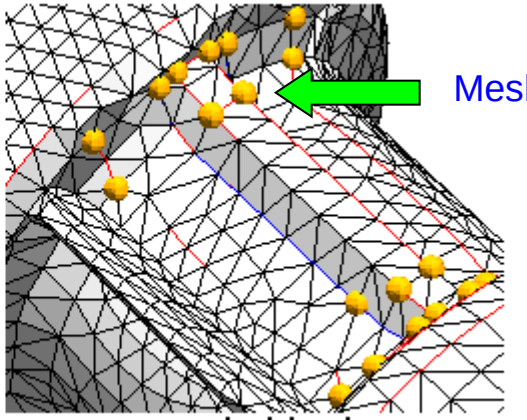
Morphing process

- Step 6(optional) – impose additional features to improve morphed mesh quality

  Add biasing, handles dependency, constraint, extra handles, symmetry or reference geometry for mapping



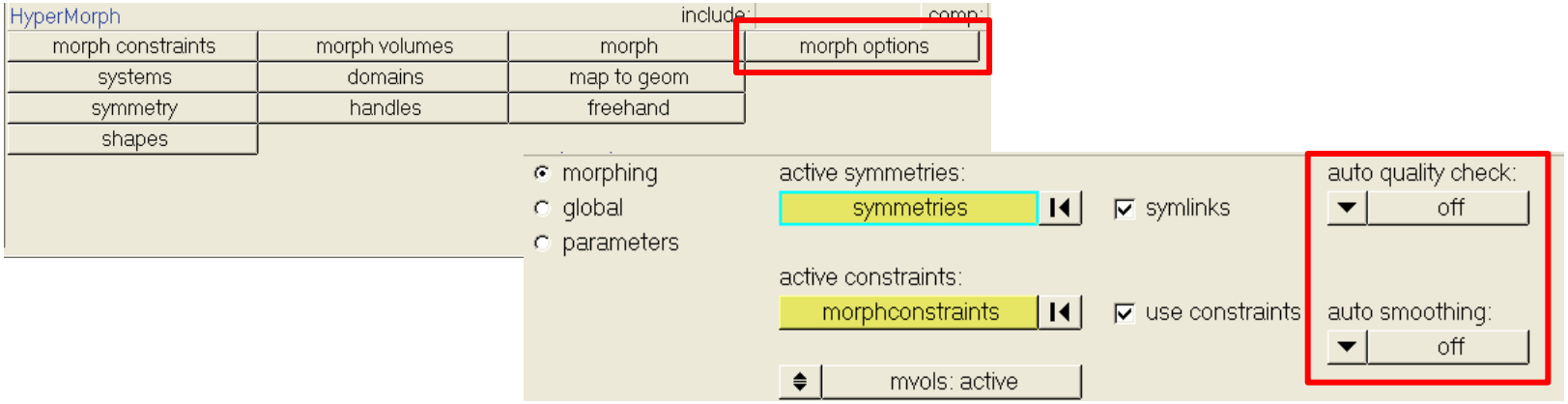
morph curvature



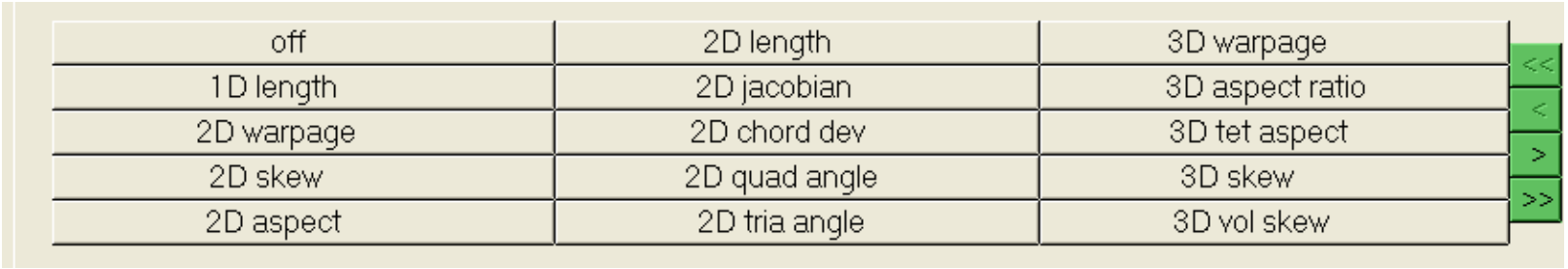
apply biasing

Morphing process

- Step 6(optional) – impose additional features to improve morphed mesh quality

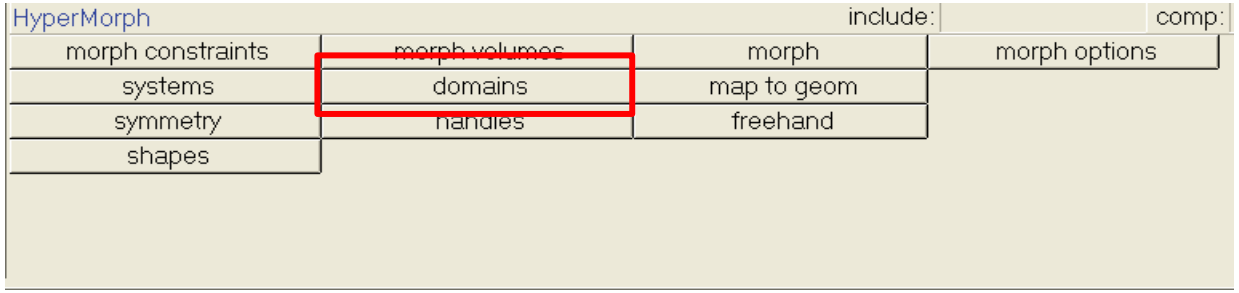


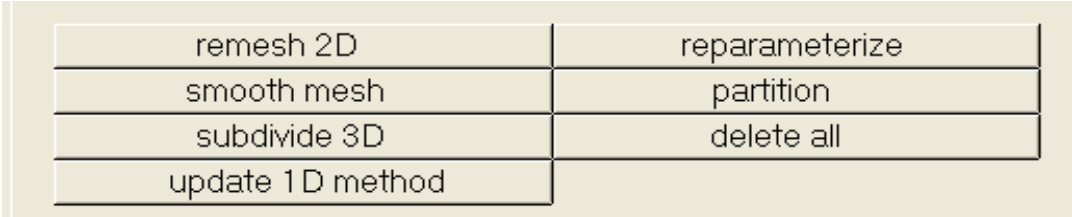
Auto quality check



Morphing process

- Step 6(optional) – impose additional features to improve morphed mesh quality



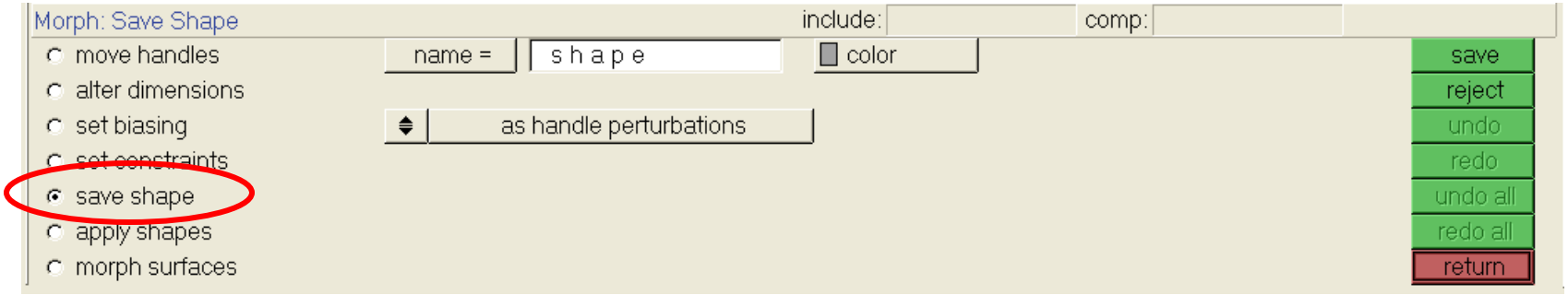


Morphing process

- Step 7(optional) – Export a solver file

  HyperMorph supports any solver which is supported by HyperMesh. HyperMorph entities will not get exported into a solver deck. (Altair Optistruct is exceptional)

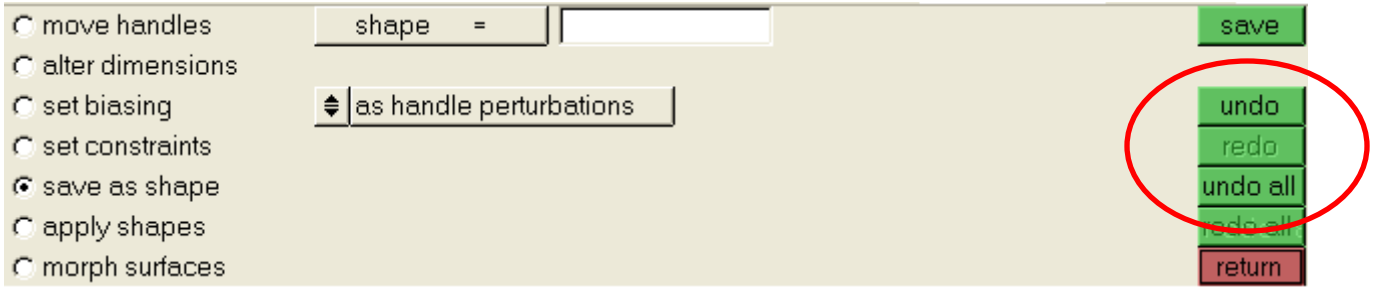
- Step 8 – save morphed mesh as shape entities
  - Storing different mesh-shape changes in one model
  - Re-applying a shape change to the mesh at a later stage
  - Combining multiple shape changes simultaneously
  - Recovering the original model
  - Completing analysis, optimization, or parametric studies using OptiStruct or HyperStudy



Morphing process

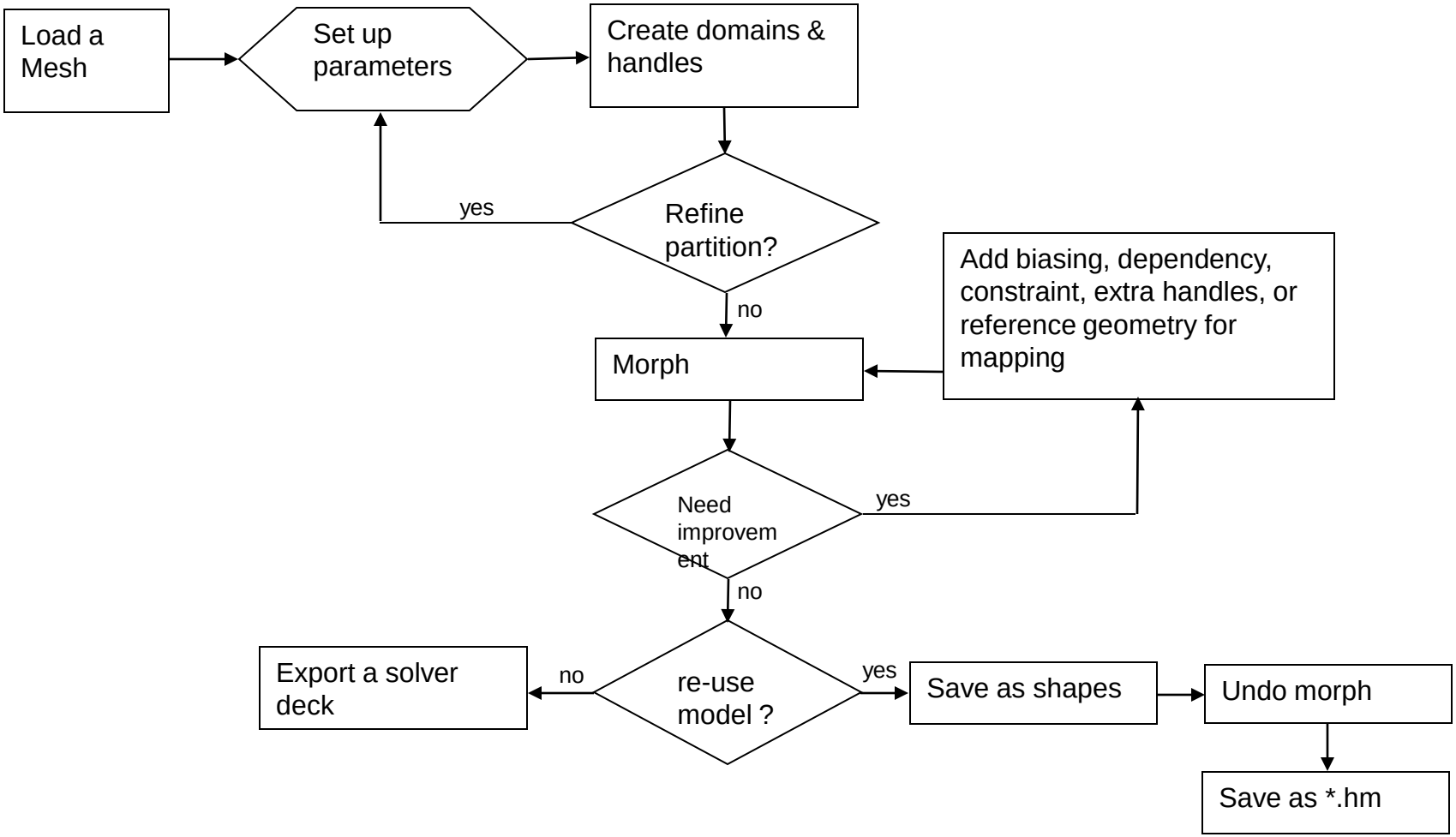
- Step 9 – Undo morph

  Use undo or undo all to get back the original mesh before saving.



- Step 10 – Save as a HyperMesh binary data file (*.hm)

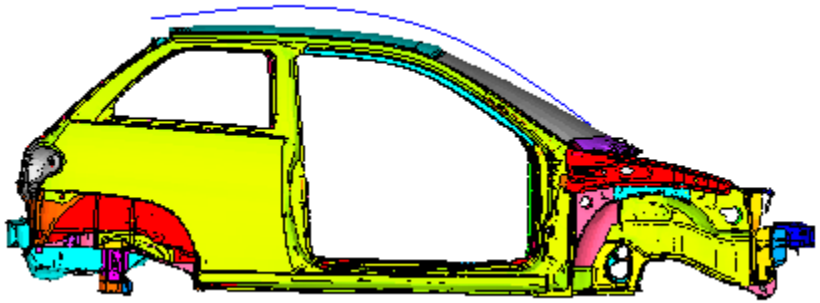
Morphing process



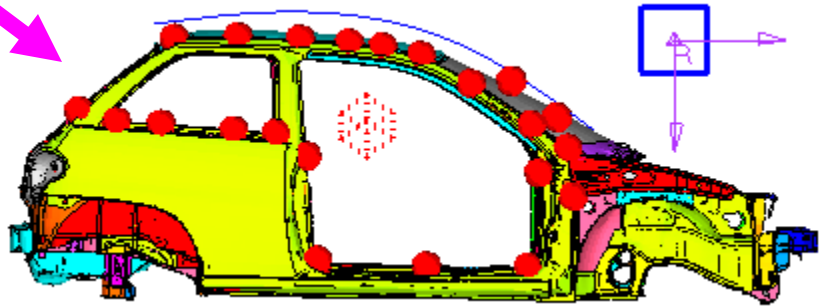
Strategy and examples

Strategy and examples

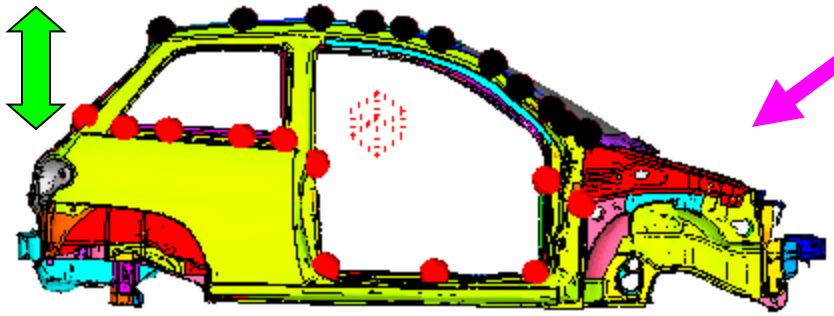
- *Raise the roof*



Create global domain
Create 1 plane symmetry
Create global handles
Constraint fixed nodes on target mesh



Morph handles to new positions

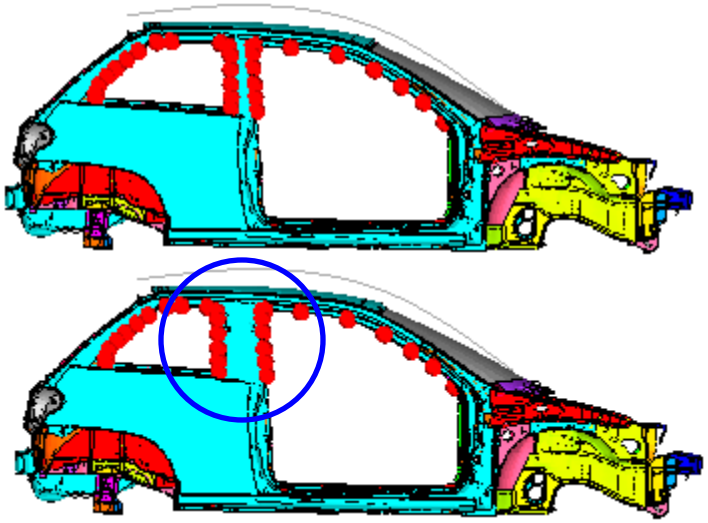


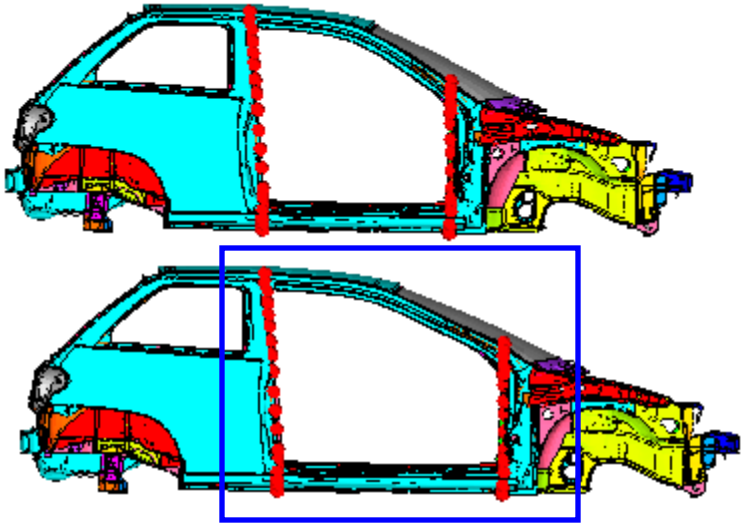
Strategy and examples

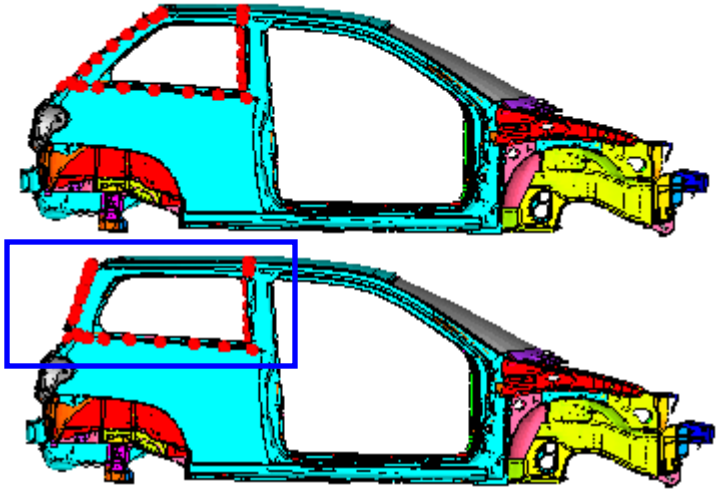
1. Change B - pillar

2. Change vehicle back shape

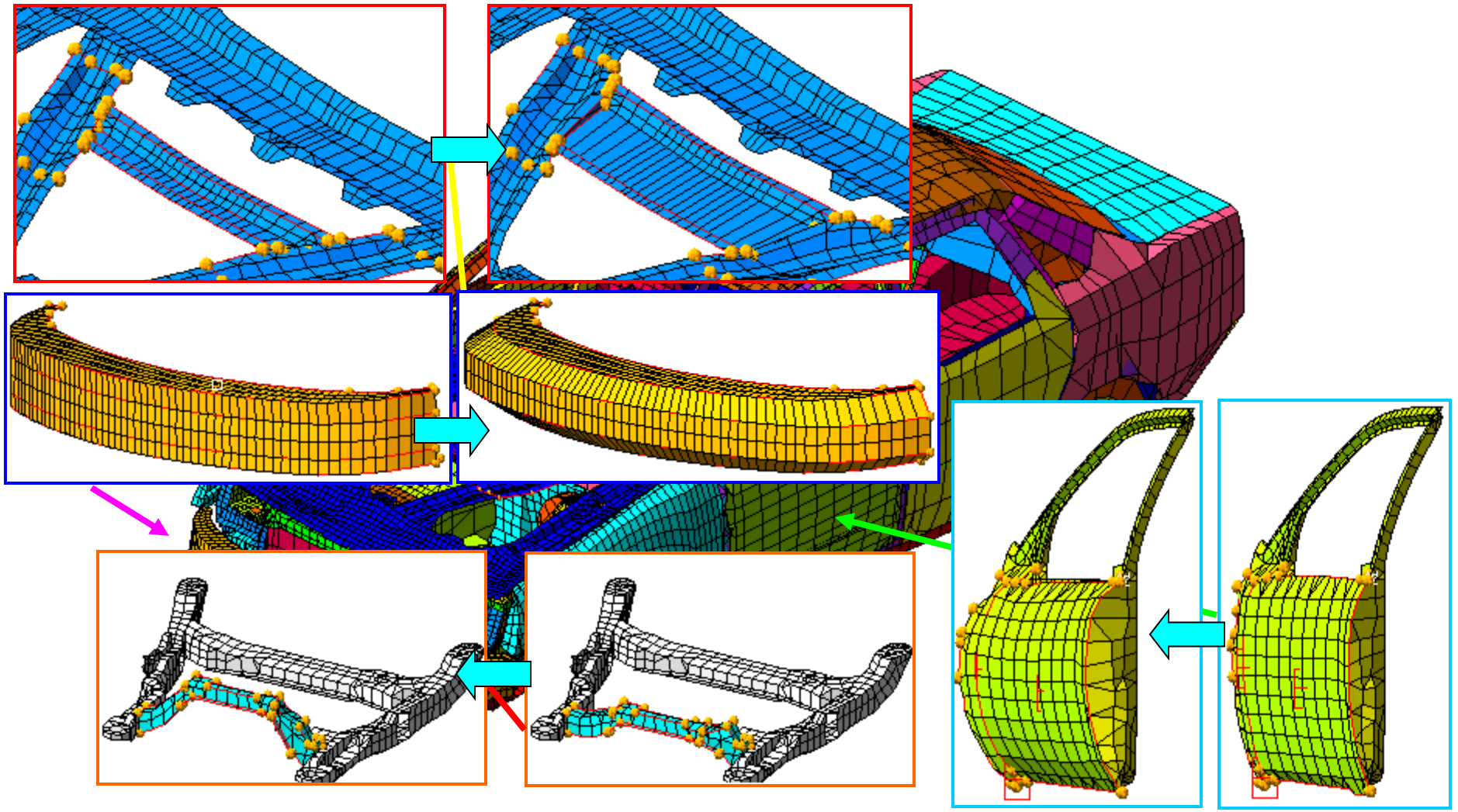
3. Change front occupancy



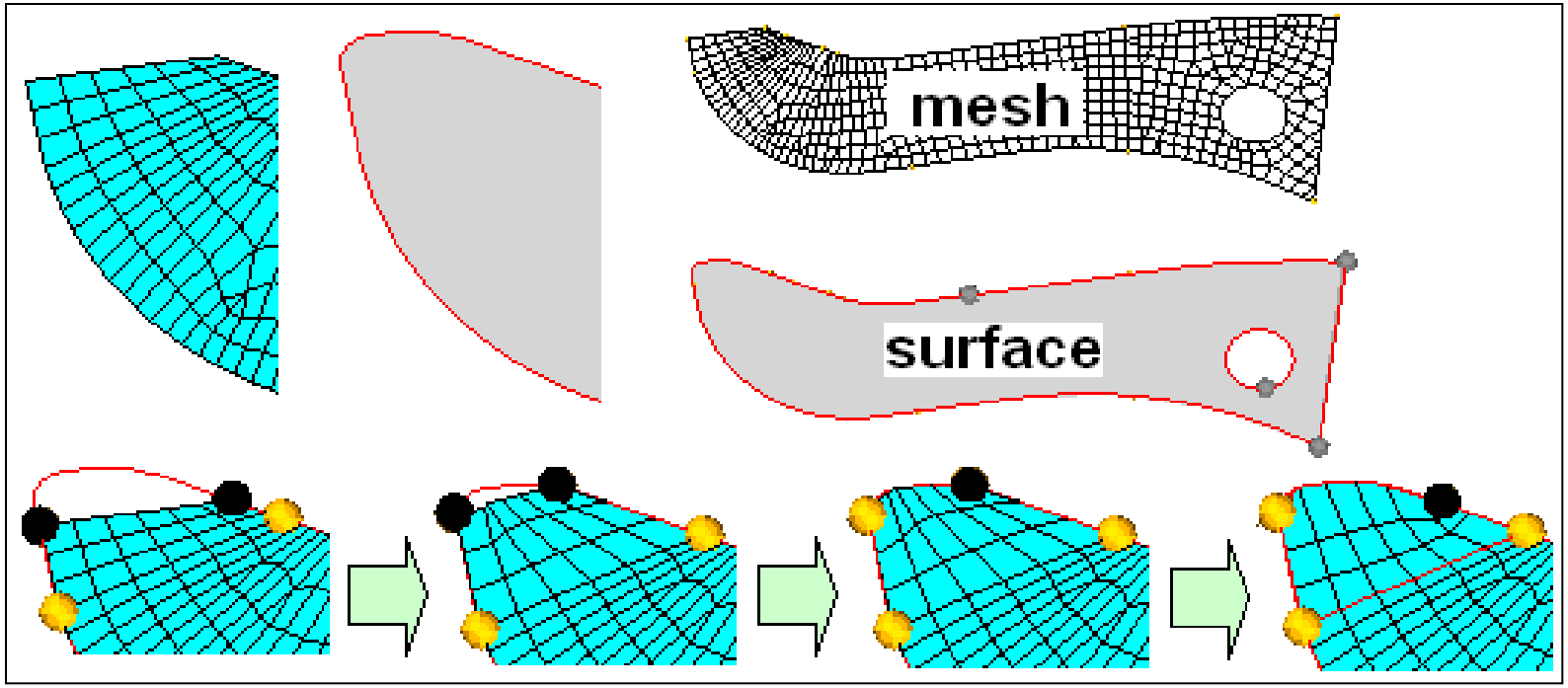




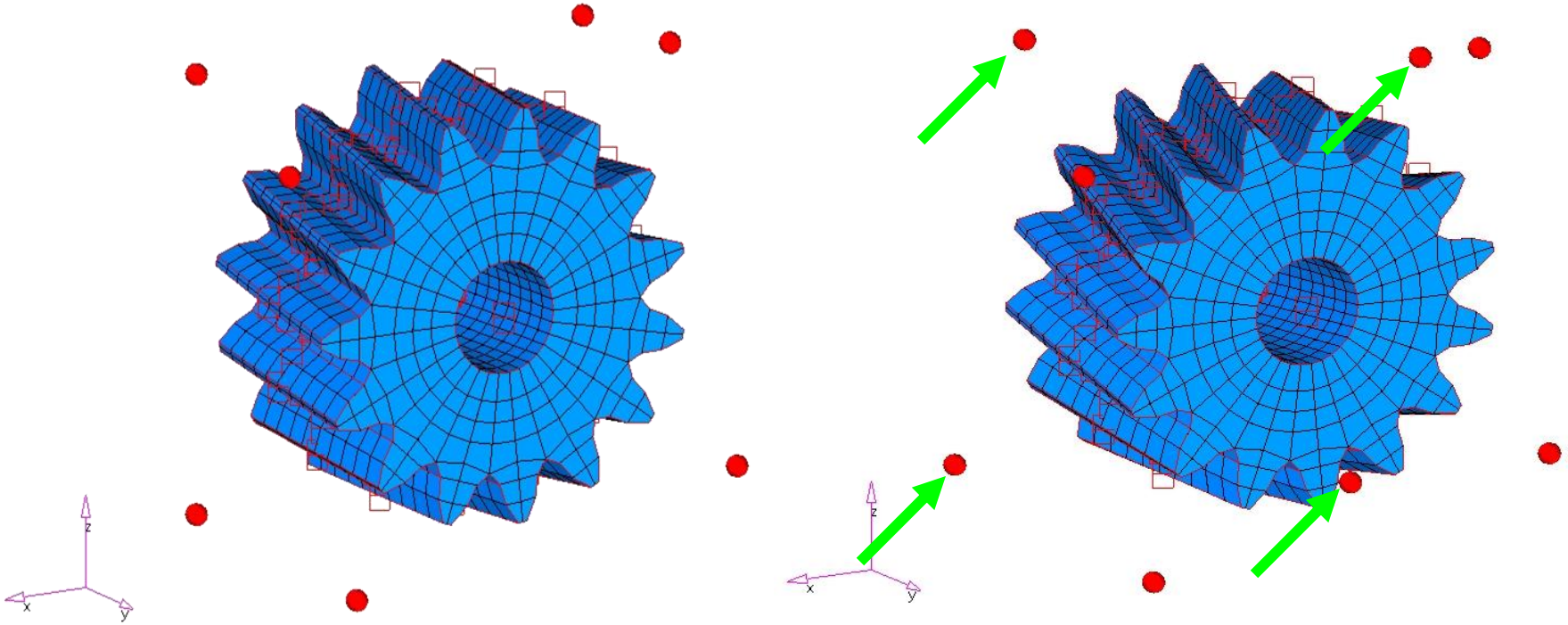
Strategy and examples



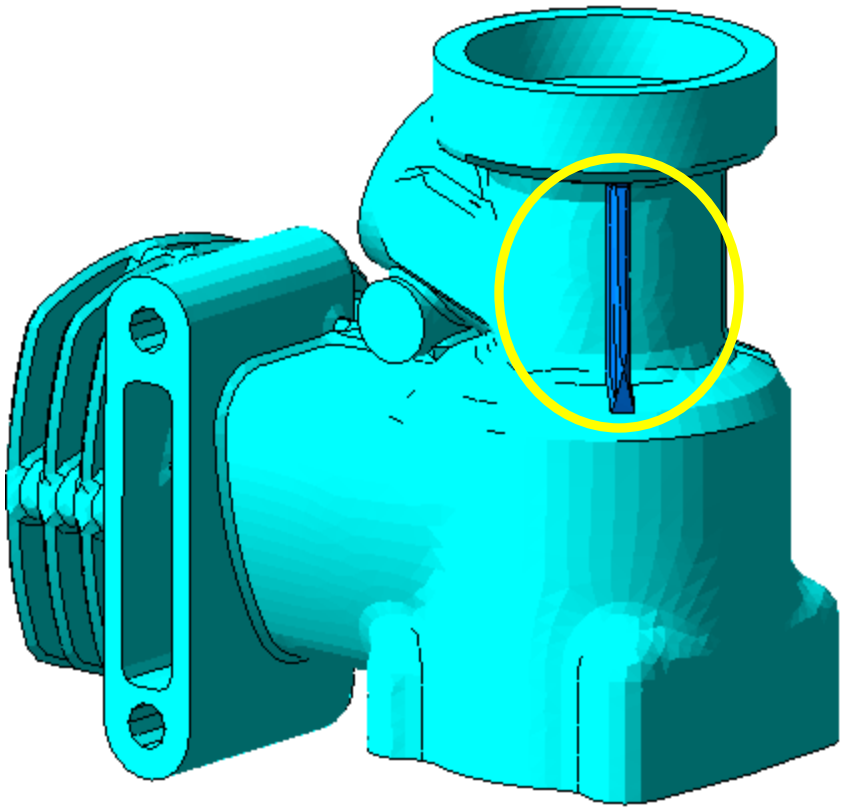
Strategy and examples

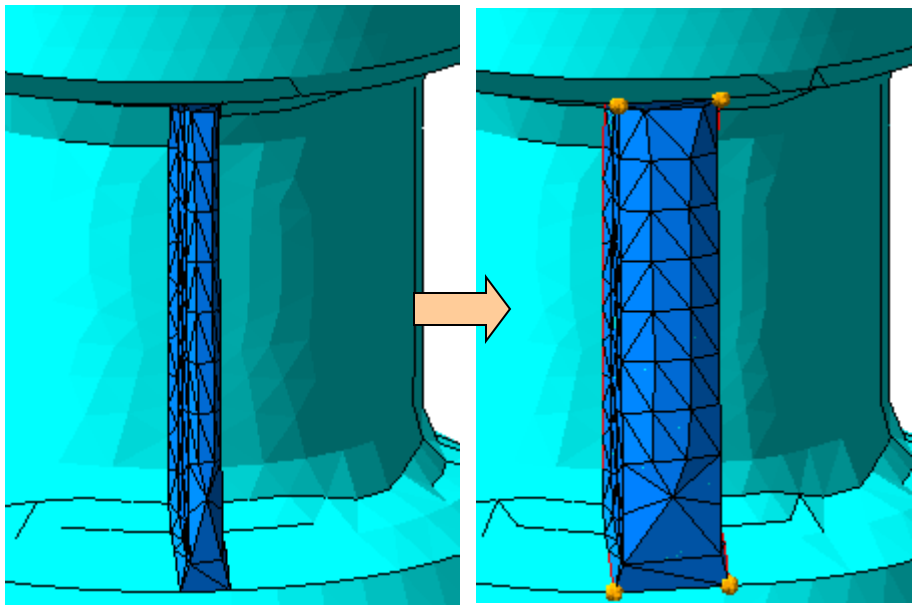


Strategy and examples



Strategy and examples

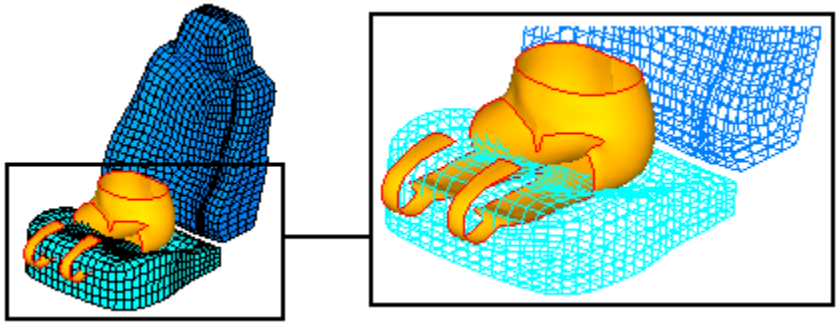




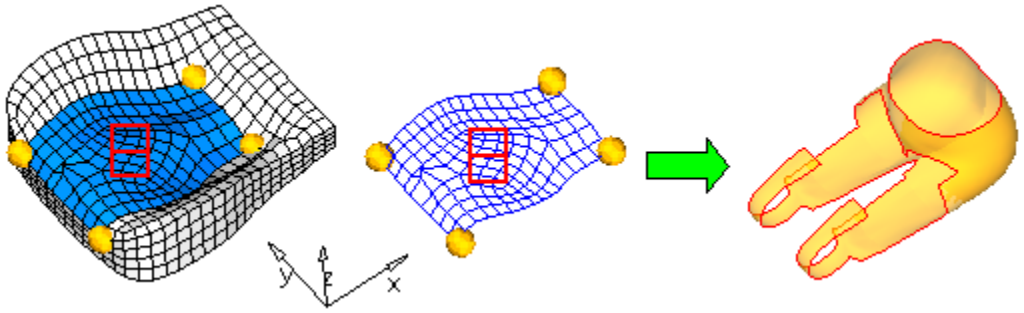
Strategy and examples

- Dummy de-penetration :Combine morphing with geometry cleanup and map to surface

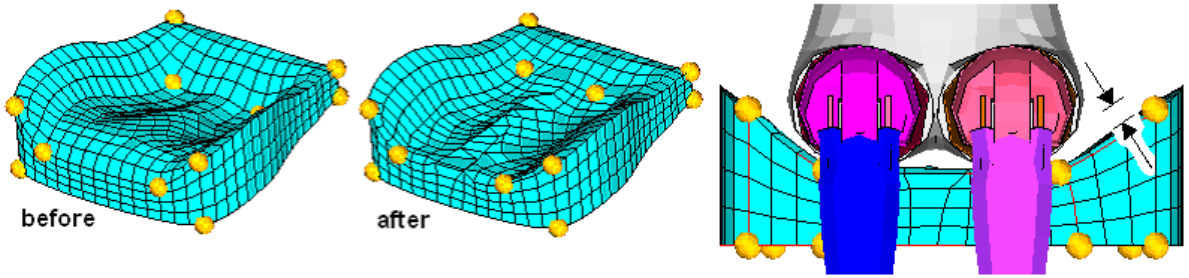
Penetration problem



Map to Geometry



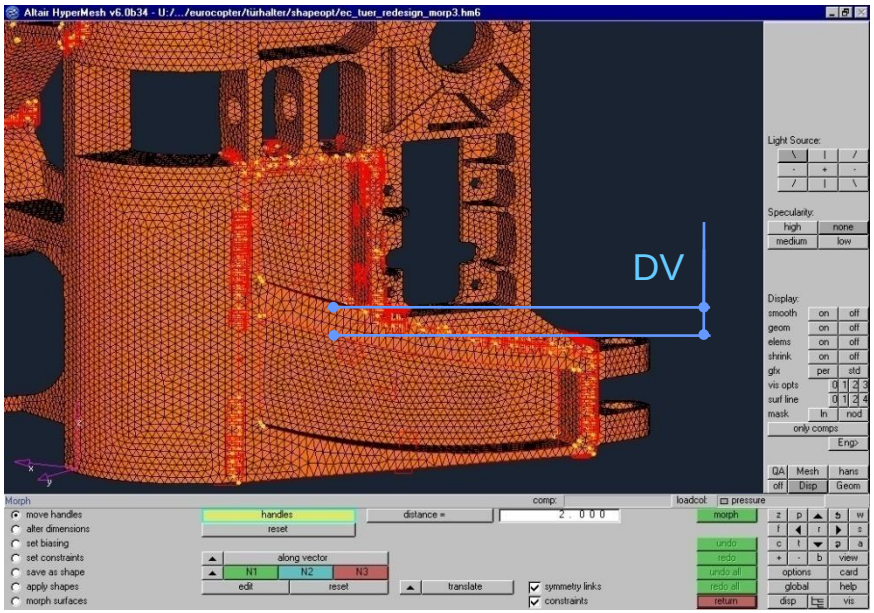
de-penetration

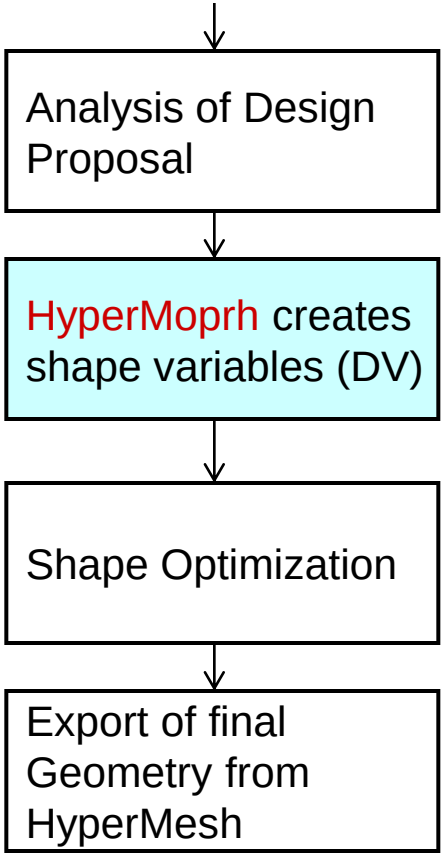


Optimization

Optimization

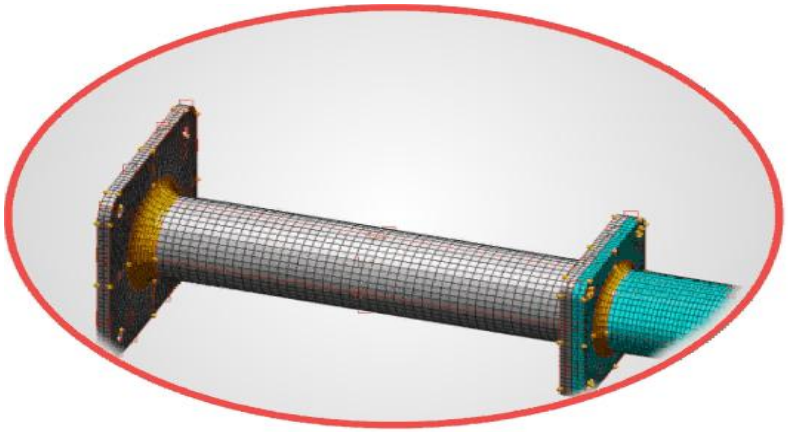
- Optimization – general approach

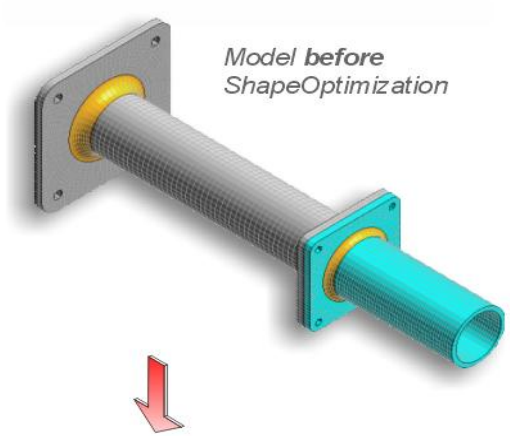


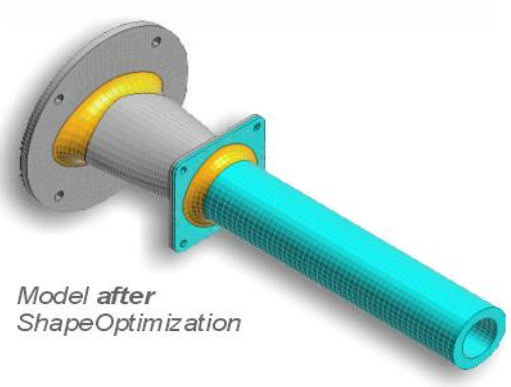


Optimization

- Optimization : using Altair Optistruct with HyperMorph
  - Shape Optimization
    - Fine tune designs
      - Find true dimensions
      - Reduce stresses
    - Control geometry for manufacturability
    - Easy to use: HyperMorph



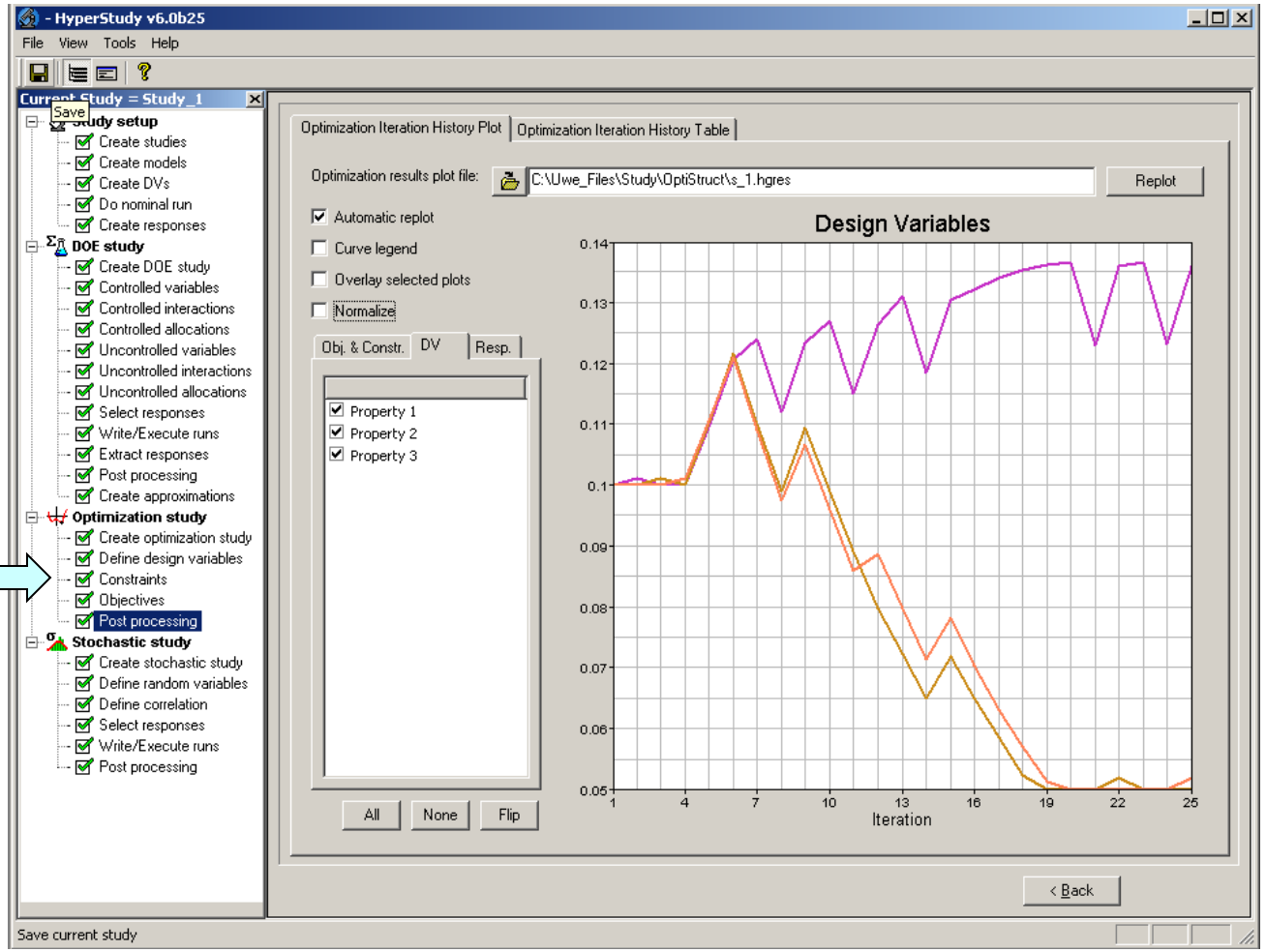




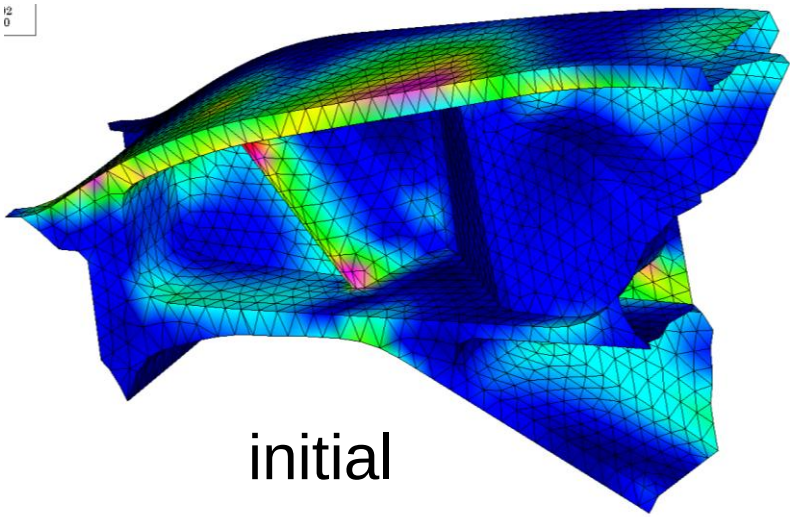
Optimization

- Optimization – using Altair HyperStudy with HyperMoprh

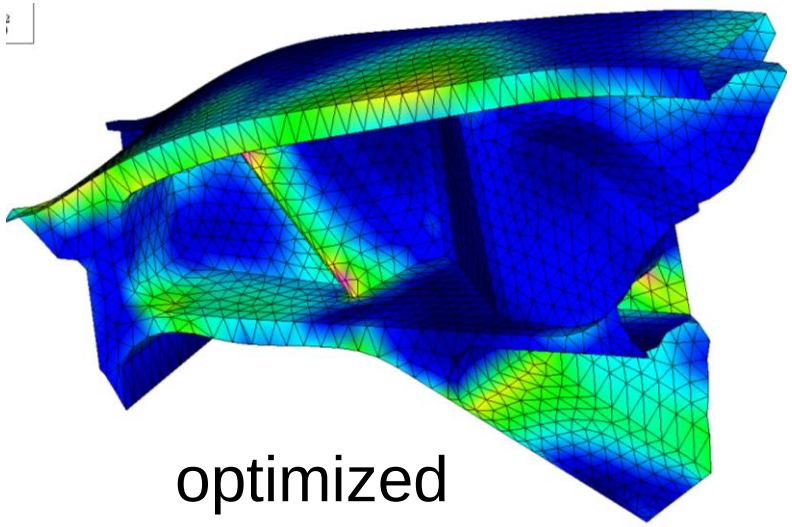
Many solver interfaces such as Abaqus, LS-Dyna, etc. for multi-attribute studies



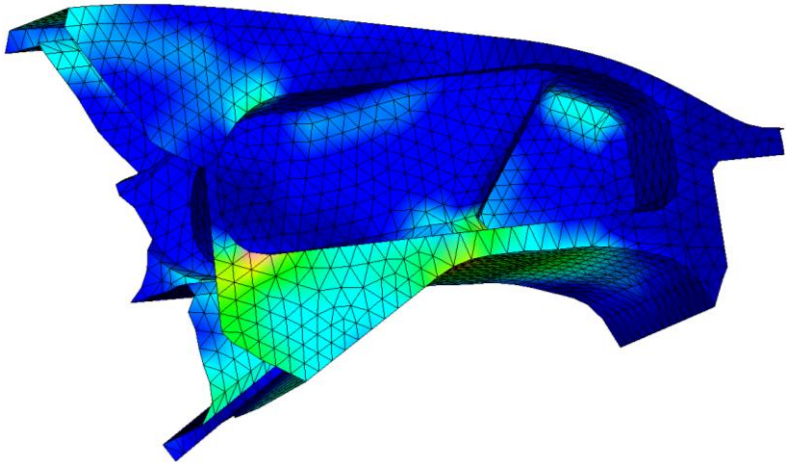
Optimization – comparison result

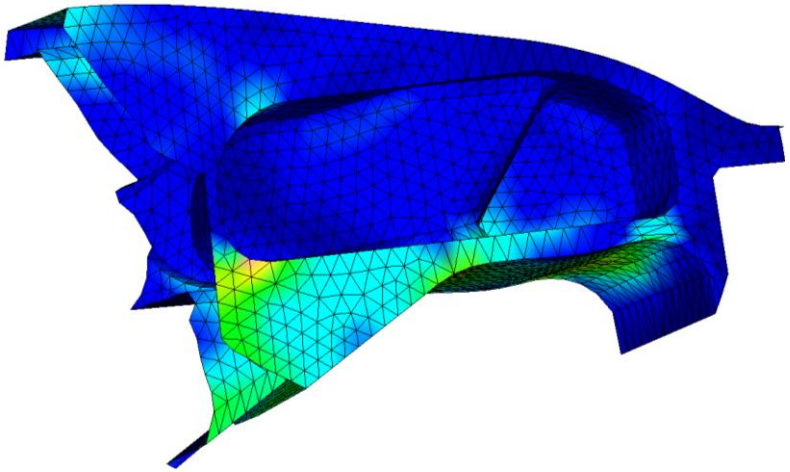


initial



optimized





Thank you