

# Free-size Optimization Manufacturability

A concern in free-size optimization is that the design concepts developed are very often not manufacturable. Another problem is that the solution of a free-size optimization problem can be mesh dependent, if the appropriate measure is not taken.

OptiStruct offers a number of different methods to account for manufacturability when performing free-size optimization.

## Member Size Control

Member size control allows you some control over the member size in the final free-size design and the resulting degree of simplicity therein.

This feature may be added one of the two ways.

### DOPTPRM Card

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
DOPTPRM	<i>MINDIM</i>	<i>VALUE</i>							

### DSIZE Card

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	<i>MEMBSIZ</i>	<i>MINDIM</i>							

Here, only the preferred minimum diameter (width in 2D) of members may be defined as the *VALUE* field, following the *MINDIM* keyword. A global minimum member size is defined in this way. Here, the preferred minimum, *MINDIM* member may be defined on the *MEMBSIZ* continuation line. Member size dimensions can be defined differently for each *DSIZE* entry in this way.

### Minimum Member Size Control

Although minimum member size control penalizes the formation of small members, results that contain members significantly under the specified minimum member size can still be obtained. This is because a small member in the structure can be very important to the load transmission and may not be removed by penalization. Minimum member size control functions more as a quality control than a quantity control.

A discrete solution is achieved in two iterative steps. The first step converges to a solution with a large number of semi-dense elements. The second step tries to refine this solution to a solution with fully dense members. Each step consists of a number of iterations. The first step consists of two entire convergence phases - the first run with the initial discreteness values (defined by *DISCRETE* and *DISCRT1D* parameters on the *DOPTPRM* Bulk Data Entry), followed by a run with the discreteness values increased by 1.0. This procedure is implemented in order to achieve a solution with clearly defined members. If this step does not create a solution with clearly defined members, the preferred

minimum member size will not be preserved in the second step. In which case, you will need to increase the discreteness parameters and/or reduce the convergence tolerance (defined by the *OBJTOL* parameter on the DOPTPRM Bulk Data Entry) to improve the solution of the first phase. The default discreteness is set to 1.0 for 1D elements, plates and shells, and 2.0 for 3D solids.

In general, once *MINDIM* is activated, checkerboarding is controlled by the methods applied for this feature, eliminating the need for the *CHECKER* parameter. In rare circumstances, checkerboards may still be introduced in the second phase described above for 3D solids. If this happens, an additional checkerboard control algorithm can be activated with the *MMCHECK* parameter. (The *CHECKER* and *MMCHECK* parameters are defined using the DOPTPRM Bulk Data Entry).

The use of this card will assure a checkerboard-free solution, although with the undesired side effect of achieving a solution that involves a large number of semi-dense elements, similar to the result of setting *CHECKER* equal to 1. Therefore, use this card only when it is necessary.

It is recommended that *MINDIM* be at least 3 times the average element size for all elements referenced by that *DSIZE* (or all designable elements when defined on DOPTPRM). The average element size for 2D elements is calculated as the average of the square root of the area of the elements, and for 3D elements, as the average of the cubic root of the volume of the elements.

This recommendation is enforced when combined with other manufacturing constraints, and if the defined *MINDIM* is less than this value, it will be reset to a default value equal to 3 times the average element size.

---

## Pattern Repetition

A technique where different structural components can be linked together so as to produce similar topological layouts.

To achieve this goal, a master *DSIZE* card needs to be defined, followed by any number of slave *DSIZE* cards which reference the master. The master and slave components are related to each other through local coordinate systems, which are required, and through scaling factors, which are optional.

Other manufacturing constraints, such as minimum or maximum member size, can be applied to the master *DSIZE* card. These constraints will then automatically be applied to the slave *DSIZE* card(s) as described in the sections below.

The following procedure should be followed to set up pattern repetition:

1. Create a master DTPG card.
2. Apply other manufacturing constraints as needed.
3. Define the local coordinate system associated to the master DTPG card.
4. Create a slave DTPG card.
5. Define the local coordinate systems associated to the slave DTPG card.
6. Apply scaling factors as needed.
7. Repeat steps 4-6 for any number of slave DTPG cards.

## Local Coordinates Systems

Local coordinates systems are generated by providing four points. These points can be defined either by entering explicit coordinates or by referencing existing grids, as follows:

- |             |  |
|-------------|--|
| <b>CAID</b> | Defines the anchor point for the local coordinates system.           |
| <b>CFID</b> | Defines the direction of the X-axis.                                 |
| <b>CSID</b> | Defines the XY plane and indicates the positive sense of the Y-axis. |
| <b>CTID</b> | Indicates the positive sense of the Z-axis.                          |

The definition of the fourth point allows for both right-handed and left-handed coordinate systems, which facilitates the creation of reflected patterns.

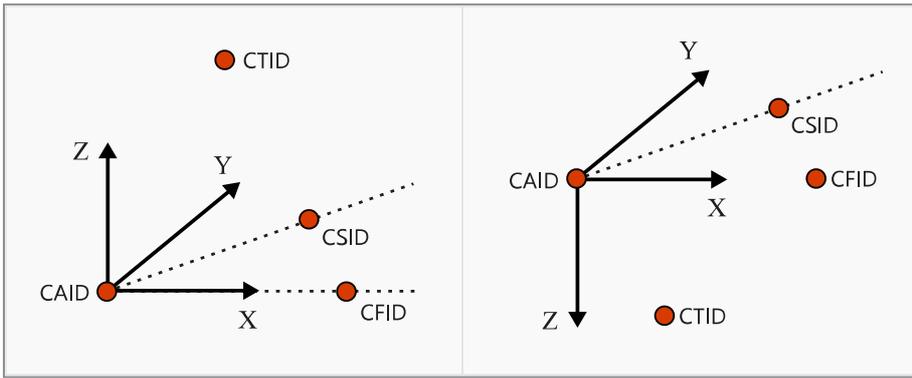


Figure 1. Local Coordinate Systems

Alternatively, local coordinate systems can be defined by referencing an existing rectangular coordinate system in the *CID* field, and by defining an anchor point in the *CAID* field.

**Note:** If the fields defining *CFID*, *CSID*, *CTID*, and *CID* are left blank, then the global coordinates system is used by default. The anchor point *CAID*, however, is always required.

### Scaling Factors

Scaling factors in the X, Y, and Z directions can be defined for each slave *DSIZE* card. These factors are always related to the local coordinate system. By playing with the local coordinate systems and the scaling factors, a wide range of effects can be obtained as illustrated in Figure 2.

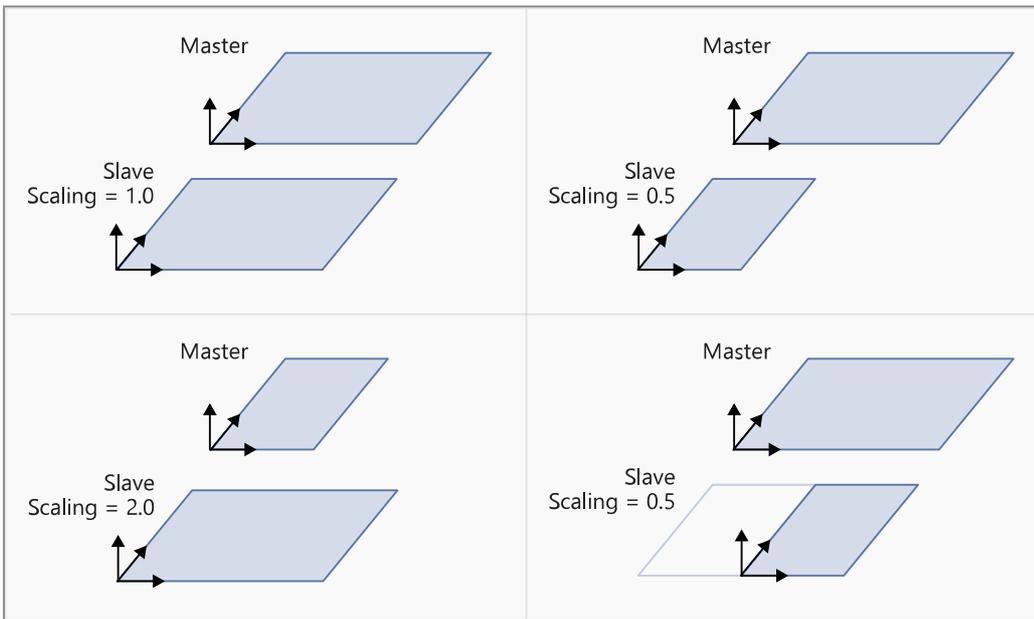


Figure 2. Scaling Factors

### Pattern Grouping

Pattern grouping is a feature where you can define a single part of the domain that should be designed in a certain pattern. pattern grouping - free-size optimization.

### Planar Symmetry

It is often desirable to produce a design that has symmetry. Unfortunately, even if the design space and boundary conditions are symmetric, conventional free-size optimization methods do not guarantee a perfectly symmetric design.

By using symmetry constraints in free-size optimization, symmetric designs can be attained regardless of the initial mesh, boundary conditions, or loads. Symmetry can be enforced across one plane, two orthogonal planes, or three orthogonal planes. A symmetric mesh is not necessary, as OptiStruct will create variables that are very close to identical across the plane(s) of symmetry.

To define symmetry across one plane, it is necessary to provide an anchor grid and a reference grid. The first vector runs from the anchor grid to the reference grid. The plane of symmetry is normal to that vector and passes through the anchor grid.

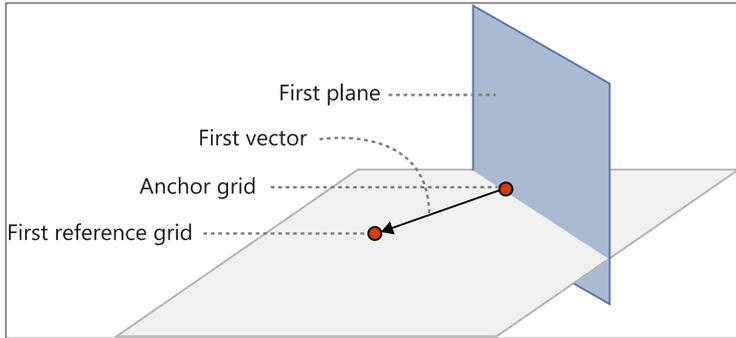


Figure 3. Symmetry Across One Plane

To define symmetry across two planes, a second reference grid needs to be provided. The second vector runs from the anchor grid to the projection of the second reference grid onto the first plane of symmetry. The second plane of symmetry is normal to that vector and passes through the anchor grid.

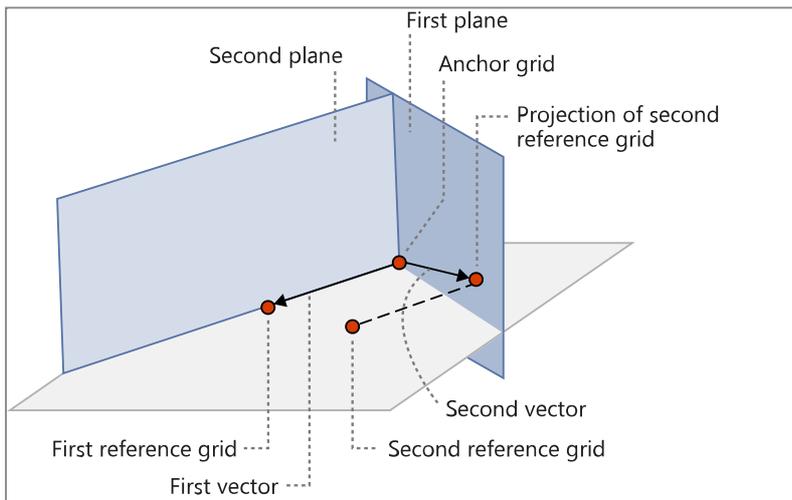


Figure 4. Symmetry Across Two Planes

To define symmetry across three planes, no additional information is required, other than to indicate that a third plane of symmetry is to be used. The third plane of symmetry is perpendicular to the first two planes of symmetry, and also passes through the anchor grid.

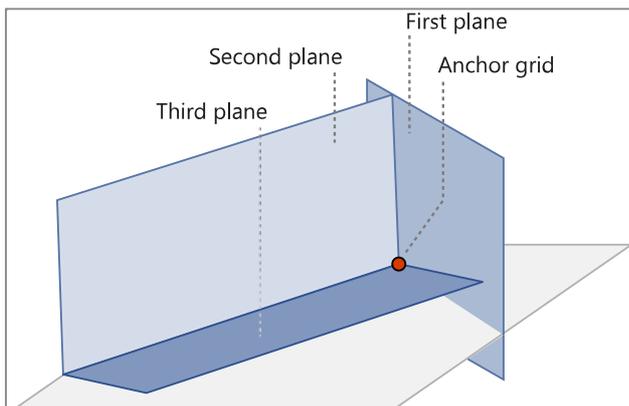


Figure 5. Symmetry Across Three Planes

## Uniform Element Thickness

Pattern grouping also provides the possibility to request a uniform element thickness throughout selected components.

This pattern group ensures that all elements of selected components maintain the same element thickness with respect to one another.

## Cyclical Symmetry

Cyclical symmetry can also be defined through the use of pattern grouping.

With cyclical pattern grouping, the design is repeated about a central axis a number of times determined by you. Furthermore, the cyclical repetitions can be symmetric within themselves. If that option is selected, OptiStruct will force each wedge to be symmetric about its centerline.

To define cyclical symmetry, it is necessary to provide an anchor grid and a reference grid. The axis of symmetry runs from the anchor grid to the reference grid. It is also necessary to specify the number of cycles; the repetition angle will be automatically computed.

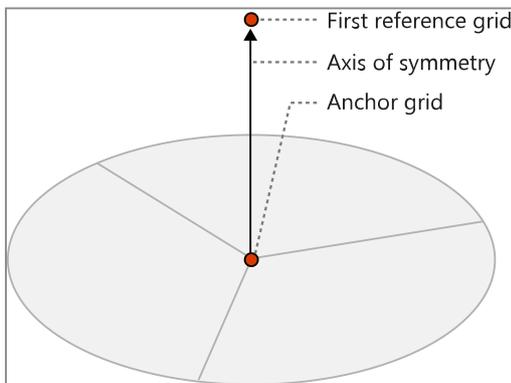


Figure 6. Cyclical Symmetry

To add planar symmetry within each wedge, a second reference grid needs to be provided. The plane of symmetry is determined by the anchor grid and the two reference grids.

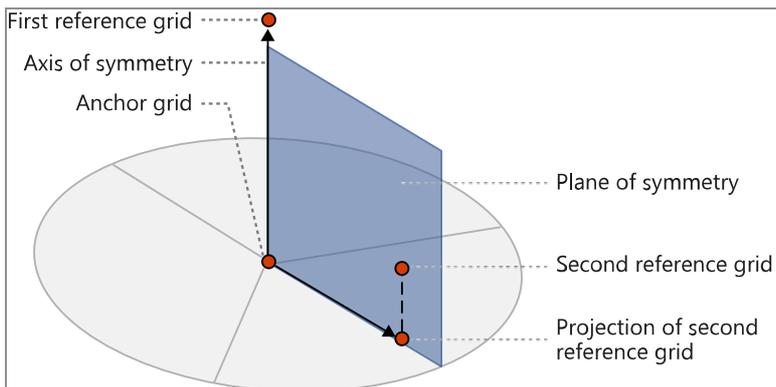


Figure 7. Planar Symmetry

## Linear and Planar Pattern Grouping (TYP=20 and TYP=21)

Linear pattern grouping requires that the anchor point and first point be defined. A vector from the anchor point to the first point defines the direction in which the thickness is set to be constant. Linear pattern grouping is typically designed to handle models with minimal or no curvature in the specified vector direction (which is typically orthogonal to the rolling direction in rolling applications). For models with low curvature in the vector direction, appropriate projections to the surface are used to determine the direction on the surface. For models with high curvature in the vector direction, depending on the direction of the specified vector, the direction may become

orthogonal to the surface whereby the pattern grouping direction cannot be determined. In such cases, Planar Pattern Grouping (TYP=21) is recommended.

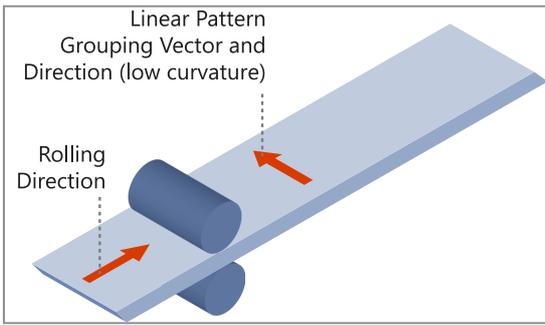


Figure 8. Planar Pattern Grouping

Planar pattern grouping requires that the anchor point and first point be defined. A vector from the anchor point to the first point is defined and thickness of the model in the various orthogonal planes to this vector is set to be constant. Planar pattern grouping is designed to handle models with high curvature in the orthogonal planes of the defined vector, and with minimal or no curvature in the direction of the defined vector. The vector defined in planar pattern grouping should typically lie in the rolling direction in rolling applications. This feature can handle large curvature in the slicing plane orthogonal to the defined vector. Planar pattern grouping cannot be used if large curvature exists in the rolling direction.

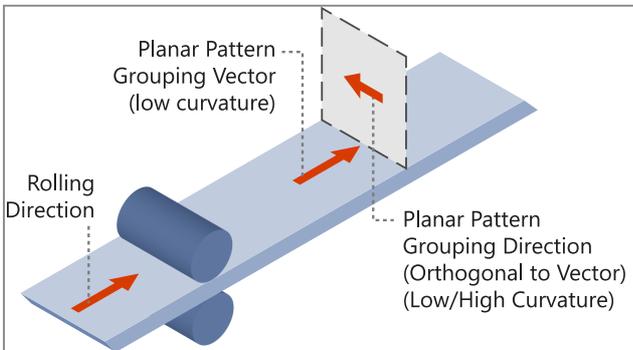


Figure 9.

#### See Also

[Free-size Optimization](#)

[Multi-Model Optimization](#)