

This chapter describes the automatic and manual procedure for creating prisms in TGrid. It also discusses the solution to some common problems that you may face while creating prisms.

This chapter comprises the following sections:

- Section 10.1: Overview
- Section 10.2: Procedure for Generating Prisms
- Section 10.3: Prism Meshing Options
- Section 10.4: Zones Created During Prism Generation
- Section 10.5: How TGrid Builds Prisms
- Section 10.6: Using Adjacent Zones as the Sides of Prisms
- Section 10.7: Direction Vectors
- Section 10.8: Offset Distances
- Section 10.9: Improving Prism Quality
- Section 10.10: The Prisms Panel
- Section 10.11: Text Interface for Generating Prisms
- Section 10.12: Prism Meshing Problems

### 10.1 Overview

In TGrid, you can create 3D prism cells starting from either quadrilateral or triangular boundary faces (or both). The 2D prism cells (quadrilateral) can be created starting from linear faces. Prisms can be used to resolve a boundary layer region of a triangular or tetrahedral mesh. They can be used to extend some portion of a domain for which the volume mesh already exists (e.g., increase the length of an inlet pipe), or to create a volume mesh by extrusion. To create prisms, you need to specify the boundary zone(s) from which the prisms will be built, their heights, and the direction in which you want to build them.

TGrid also allows you grow prisms on dangling walls. You can also specify different growth methods on adjacent walls in the domain. TGrid can automatically detect the collision between the prisms layers and avoid the collision by adjusting the height of the colliding prism layers.

Figure 10.1.1 shows the types of prisms that TGrid can generate.

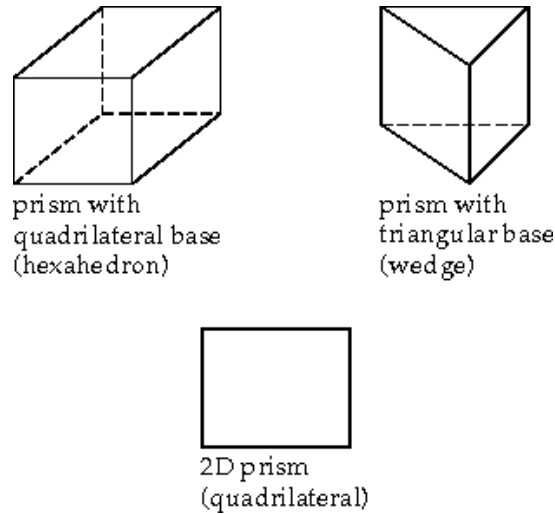


Figure 10.1.1: Prism Shapes that TGrid Creates

Instructions for building prisms are provided in Section 10.2: [Procedure for Generating Prisms](#). Details about how TGrid uses your inputs to generate the prisms are provided in Section 10.5: [How TGrid Builds Prisms](#). Problems related to prism generation are discussed in Section 10.12: [Prism Meshing Problems](#).

## 10.2 Procedure for Generating Prisms

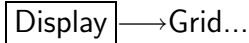
The procedure for creating prisms from boundary zones is as follows:

1. Check the boundary mesh to ensure that free nodes or faces with high skewness do not exist. See Sections 7.1 and 14.2 for details.
2. Select the boundary zone(s) from which you want to grow prisms in the **Boundary Zones** selection list in the **Prisms** panel.

Mesh → Prisms...



If you read in a mesh file created in a previous version of TGrid, it is recommended that you reset all the prism parameters using the command `/mesh/prism/reset-parameters` before proceeding with setting the prism parameters.

3. Specify prism growth parameters in the **Growth** tab.
  - (a) Select the appropriate **Offset Method** and **Growth Method** for growing prisms. See Sections 10.8 and 10.10 for details.
  - (b) Define the height of the first prism layer (**First Height/First Aspect Ratio**).
  - (c) Set the growth method related parameter (**Slope, Rate, or Exponent**, as applicable) and the **Number of Layers**.
  - (d) Specify options for proximity detection and ignoring prisms in regions of proximity and invalid normals in the **Prisms Growth Options** panel (opened using the **Growth Options...** button), if required. Refer to Section 10.3: [Prism Meshing Options](#) for details.
  - (e) Click the **Plot** or **Print** button to preview the height distribution.  
For information on setting different growth parameters for different zones, see Section 10.3.1: [Growing Prisms Simultaneously from Multiple Zones](#).
  
4. Specify the direction for prism growth (see Section 10.7: [Direction Vectors](#)).
  - (a) Enable the display of normals in the **Attributes** section of the **Display Grid** panel and click **Apply**.  

  - (b) Click the **Direction** tab in the **Prisms** panel to view the direction parameters.
  - (c) Click **Draw** and check if the normal vectors for each zone are pointing in the direction that you want to build the prisms.
    - If the vectors for the faces on each zone do not point in the same direction (i.e., if some point on one side of the zone and others point on the other side), click the **Orient Normals** button to reorient them.
    - If the vectors point in the wrong direction, click the **Flip Normals** button to reverse the direction.
  - (d) Display the zone(s) to verify that the normals now point in the correct direction.
  - (e) Specify the method for determining the direction of the prisms:
    - Select **Normal** in the **Method** list to compute normal direction vectors. This method takes into account the change in direction required for a curved region. Specify the values for the maximum angle change.
    - Select **Uniform** in the **Method** list to use a constant, uniform direction for flat prism regions. Specify the appropriate direction vector.
  
5. Set the parameters for prism improvement and prism projection available in the **Improve** and **Project** tabs, respectively, as required.

6. Click **Apply** to save your settings.
7. Save the mesh.

**File** → **Write** → Mesh...



It is a good practice to save the mesh at this point. If you are dissatisfied with the prisms generated, you can read in this mesh and modify the prism parameters to regenerate the prisms.

8. Click **Create** to generate the prisms.

Refer to Section [10.12: Prism Meshing Problems](#) for suggestions on solving problems occurring during prism generation.

9. Check the maximum face skewness value reported in the TGrid console when the final prism layer is created, to ensure that the value is acceptable.

This is only for 3D and is especially important if you are going to generate tetrahedral or pyramidal cells using this new boundary zone.

The new zone should not have highly skewed triangles or quadrilaterals. For more information about the quality of quadrilateral faces used for creating pyramids, see Section [9.3.1: Creating Pyramids](#).

If the reported maximum skewness is too high, read the mesh file you saved after setting the prism parameters and try again with different parameters. For details, see Section [10.12: Prism Meshing Problems](#).

10. A number of new zones are created when the prism generation is complete (see Section [10.4: Zones Created During Prism Generation](#)). If the prism-cap or prism-side face zones are *not* supposed to be walls (e.g., if they are supposed to be simple interior zones), change them to the appropriate boundary type.

**Boundary** → Manage...

For most cases, performing these steps will result in an acceptable mesh of prismatic cells, but for some complex geometries, you may have to modify the procedure. If you are not satisfied with the prisms generated by TGrid, you can read in the mesh you saved before generating prisms and modify the prism parameters. Refer to Sections [10.5—10.12](#) for details on modifying the prism generation parameters to create a better mesh.

## 10.3 Prism Meshing Options

This section describes various the options available in TGrid while creating a prism mesh for the domain.

### 10.3.1 Growing Prisms Simultaneously from Multiple Zones

You can select multiple face zones for simultaneously growing prism layers. If the face zones are connected, the prisms grown will also be connected. A single zone of prism cells and a single zone of cap faces will be created for each set of simultaneously grown layers. To retain the individual prism cell zones and cap face zones, enable **Grow Individually** in the **Prisms Growth Options** panel (opened by clicking the **Growth Options...** button).

TGrid allows you to grow prisms from multiple zones with the same or different growth parameters.

- To use the same growth parameters for all zones, follow the steps in Section 10.2: [Procedure for Generating Prisms](#). In this case, the same number of layers are grown from all zones, and all other parameters (growth method, offset method, direction method, etc.) will remain the same.
- To use different growth parameters for different zones, apply different growth methods, offset methods, first height, and other growth parameters (as required for the growth method selected) individually for each zone and click the **Apply** button in the **Zone Specific Growth** group box in the **Prisms** panel.

If you specify different growth parameters for different zones, all other parameters (direction method, offset method, etc.) can be set separately for each zone. In this case, the offset height of each node that is shared by multiple zones will be the average of the heights applied on the separate zones.

This produces a continuous transition between the zones (Figure 10.3.1). Offset smoothing (see Section 10.8: [Offset Smoothing](#)) is recommended in these cases to avoid sharp height changes at such edges.

To assign different growth controls and different offset types to different face zones, replace steps 2 and 3 in Section 10.2: [Procedure for Generating Prisms](#) with the following steps:

1. Select the zone(s) for which you want to specify a set of growth parameters in the **Boundary Zones** list in the **Prisms** panel.
2. Set the appropriate **Offset Method**, **Growth Method**, **First Height**, and the related growth parameters (as required) in the **Growth** tab.

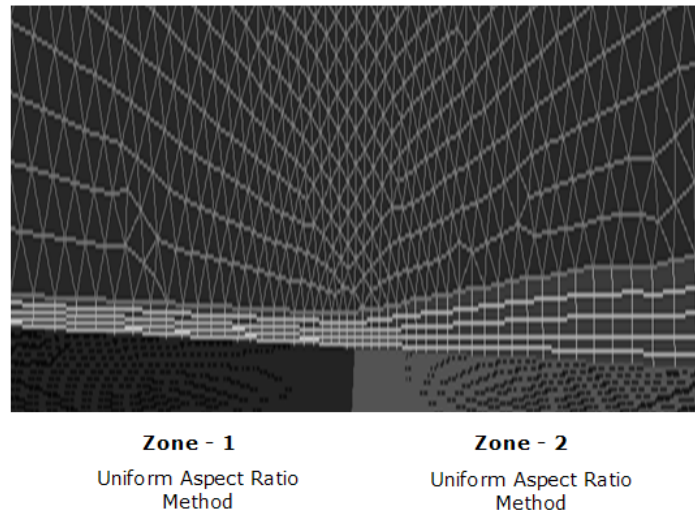


Figure 10.3.1: Different Growth Methods on Adjacent Zones

3. Specify parameters for proximity/collision detection in the Prisms Growth Options panel (see Section 10.3.3: [Detecting Proximity and Collision](#) for details).

**i** By default, TGrid generates a single zone of prism cells and cap faces for the simultaneously grown prism layers. If you want to retain individual prism cell zones and cap faces, enable **Grow Individually** in the **General Options** group box in the **Prisms Growth Options** panel.

4. Click **Apply** in the **Prisms** panel.
5. Repeat this procedure for other zones for which you want to apply different growth controls.
6. Click **Create**.



Before you click the **Create** button, ensure that *all* the zones for which you want to grow prisms are selected in the **Boundary Zones** list.

### 10.3.2 Growing Prisms on a Two-Sided Wall

Two-sided walls may be present in some geometries. To grow prisms on both sides of a two-sided wall (e.g., growing prisms on dangling walls), do the following:

1. Select the two-sided wall on which you want to grow prisms from the **Boundary Zones** selection list.

2. Specify the prism growth parameters as required.
3. Enable Grow on Two Sided Wall in the Growth tab of the Prisms panel.
4. Click Create. See Figure 10.3.2 for details.

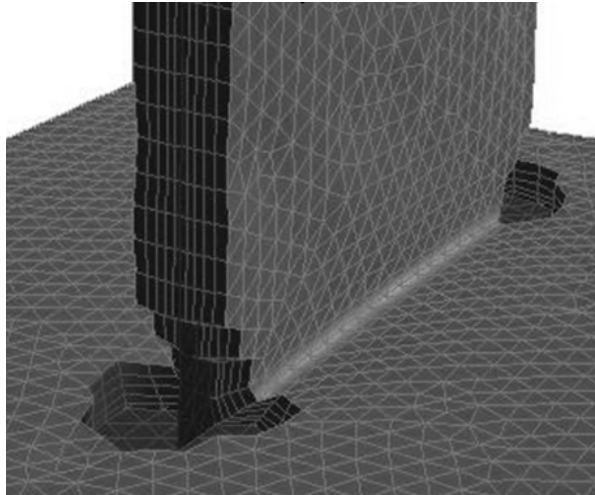


Figure 10.3.2: Prism Growth on a Dangling Wall

### 10.3.3 Detecting Proximity and Collision

If the zones on which you want to grow prisms are very close to each other, the prism layers from zones may intersect or collide with each other. This results in bad quality of prism layers.

For example, in Figure 10.3.3, the prism layers grown from proximal surfaces intersect each other. The **Allow Shrinkage** option in TGrid allows you to avoid the intersection of prism layers by adjusting the height of the prism layers in closely placed zones. Figure 10.3.4 shows the use of this option to avoid intersection of the prism layers by adjusting the prism layer height.

*A value of  $n$  for Gap Factor implies that a gap equal to  $n$  times the maximum base edge length at the node in question will be maintained. Hence, a value of 1 implies that the gap maintained is equal to the maximum base edge length at the node considered.*

**i** Offset heights are scaled only in the regions where the intersection of prism layers takes place. For the remaining regions, the prisms are created according to the specified parameters.

You can also ignore prisms regions of close proximity during prism generation. The area of proximity will be determined based on the shrink factor specified. Prism layers will be grown from the remaining geometry according to the specified parameters.

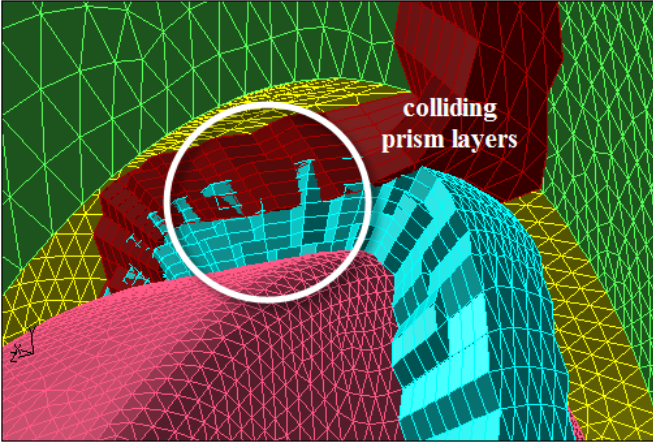


Figure 10.3.3: Collision of Prism Layers

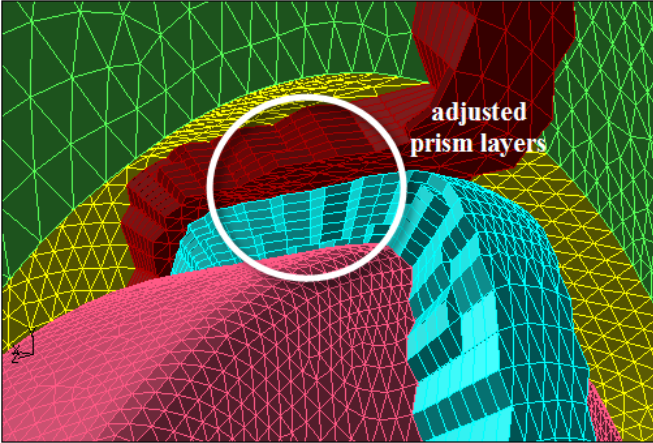


Figure 10.3.4: Prism Layers Shrunk to Avoid Collision



Figure 10.3.5 shows a portion of the geometry where the region of proximity in a sharp corner has been ignored while creating prisms.

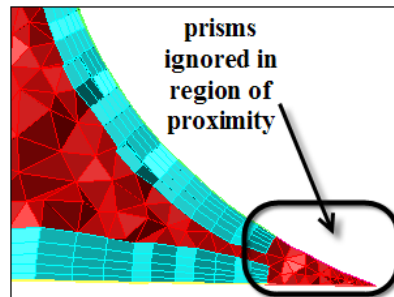


Figure 10.3.5: Ignoring Areas of Proximity

For automatic adjustment of intersecting/colliding prism layers while growing prisms, do the following:

1. Specify the prism growth parameters.
2. Click the **Growth Options...** button in the **Growth** tab to open the **Prisms Growth Options** panel (see Section 10.10.1: [The Prisms Growth Options Panel](#)).
3. Enable the appropriate options in the **General Options** group box.
4. Specify the parameters required for the proximity calculation and controlling the prism layer height.

Enable **Allow Shrinkage** to allow shrinkage of prism layers in areas of proximity. This option is enabled by default.

Enable **Keep First Layer Offsets**, if required.

*This option allows you to keep the original first offset height and scale the offset heights for the remaining layers in case of intersecting prism layers.*

Specify an appropriate value for **Gap Factor**.

Enable **Allow Ignore** to ignore the prism growth parameters in areas of proximity.

Enter an appropriate value for **Max Aspect Ratio** or **Max Shrink Factor** as required.

Click **Apply** and close the **Prism Growth Options** panel.

5. Click **Create** in the **Prisms** panel.

### 10.3.4 Ignoring Invalid Normals

Some geometries may contain regions where the normals considered while growing prisms may be invalid. The normal at a particular node may be nearly tangential to the surrounding faces in the geometry and as a result, the prism generation may fail. For such cases, you can choose to ignore regions of the geometry where invalid normals exist, during prism generation. Prism layers will be grown from the remaining geometry according to the specified parameters.

Figure 10.3.6 shows a portion of the geometry where the normal at the highlighted node will be nearly tangential to some of the surrounding faces. You can see that the region around this node has been ignored while creating prisms.

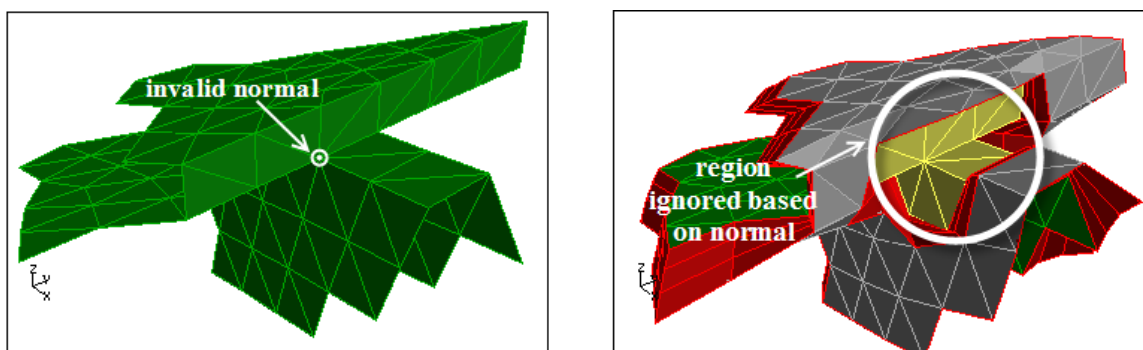


Figure 10.3.6: Ignoring Invalid Normals

You can ignore regions based on normals while creating prisms as follows:

1. Specify the prism growth parameters.
2. Select the **Boundary Zone** containing the invalid normal and click the **Growth Options...** button to open the **Prisms Growth Options** panel.
  - (a) Enable **Ignore Invalid Normals** in the **General Options** group box.
  - (b) Click **Apply** and close the **Prism Growth Options** panel.
3. Grow prisms as required.

### 10.3.5 Preserving Orthogonality

In some cases, it may be important for the prismatic mesh to be orthogonal near the original boundary. For such cases, there are two ways to preserve orthogonality:


- Decrease the value specified for the **Max. Angle Change** in the **Direction** tab of the **Prisms** panel to a lower value to limit the change in normal direction in each prism layer.

This parameter is set to 45 degrees by default. This means the normal direction at a node can change up to 45 degrees during normal, edge, and node smoothing. For example, if you reduce this parameter to 10 degrees, the change in normal direction will be more gradual, and the mesh near the original boundary will be nearly orthogonal.

- Specify an explicit number for **Orthogonal Layers** in the **Direction** tab of the **Prisms** panel.

When you specify orthogonal layers, you should decrease the **Max Angle Change** to approximately 10 degrees. This will prevent sudden sharp changes in normal direction at the first non-orthogonal layer.

For a non-orthogonal layer, TGrid will perform edge swapping normal, edge, or node smoothing so that the layer is orthogonal to the original boundary. For example, if you specify 5 orthogonal layers, no smoothing will be performed on the first 5 layers, resulting in direction vectors that are normal to the original surface mesh faces. Full smoothing is used on the sixth and subsequent layers.

 If you are preserving orthogonality near the original boundary, make sure that the prism layers do not grow too quickly. Use small layer heights, relative to the sizes of the faces on the original boundary. If prism layers grow quickly, the nearly-orthogonal direction vectors are likely to cross at sharp concave corners, causing the prism generation to fail. TGrid will stop the prism layer generation at this point. See Section [10.7: Normal Smoothing](#).

Preserving orthogonality can affect skewness and lead to left-handed faces and/or negative volumes. See Section [10.12: Negative Volumes/Left-Handed Faces/High Skewness](#) for details.

### 10.4 Zones Created During Prism Generation

When TGrid generates prisms, several new cell and face zones are created. You can identify the new zones by their default names.

- The cell zone containing the prisms will be named `prism-cells-n`.
- The face zone created at the end of the last prism layer will be named `prism-cap-n`.
- If the prism layers are bounded by one or more existing triangular face zones, each of these will be retriangulated so that the faces of the zone that are adjacent to the prisms will become quadrilateral faces, while the rest of the faces in the zone remain triangles.

If an original triangular face zone is called `symmetry` the portion of it that still contains triangles will be retained as `symmetry`, while the portion containing quadrilateral faces will be named `symmetry-quad:n`.

- Any faces that were not projected to adjacent zones (see Section 10.6: [Using Adjacent Zones as the Sides of Prisms](#)) are collected into a zone named `prism-side-n`.
- For a zone `wall-n`, the zone created by ignoring prism growth in the region of proximity will be named as `wall-n:ignore`, where, `n` is the zone number assigned by TGrid.



TGrid will merge all the ignored threads related to a base thread into a single thread by default. You can however change this using the command `/mesh/prism/controls/merge-ignored-threads?` which will generate more than one thread per base thread.

To merge one or more prism cell zones with other cell zones (and merge the corresponding face zones), use the **Merge** option in the **Manage Cell Zones** panel (see Section 13.9.3: [The Manage Cell Zones Panel](#)).

By default, the `prism-cap-n` zones will be wall zones. If you do not want to include these walls in the model, change them to interior zones (or any other type) using the **Manage Face Zones** panel (see Section 7.13.1: [The Manage Face Zones Panel](#)).

### 10.5 How TGrid Builds Prisms

TGrid follows the following steps to build the prism and to create a high-quality mesh.

1. Boundary mesh analysis:

Analyze the initial surface mesh to determine if any adjacent zones should be projected to and retriangulated (if triangular), or if their faces should be used as the sides of new prism cells (if quadrilateral).

For details on how TGrid determines whether to use existing boundary zones or not see Section 10.6: [Using Adjacent Zones as the Sides of Prisms](#)

## 2. Prism layer growth.

Grow layers of prisms one by one, based on the layer height specification. This is a multi-step procedure, and hence you can modify or skip some of the steps:

- (a) Determine the initial direction vectors so that TGrid will know the direction in which to build the prisms. Several advancement direction methods are available, as discussed in Section 10.7: [Direction Vectors](#).
- (b) Smooth the initial direction vectors. See Section 10.7: [Normal Smoothing](#) for details.
- (c) Determine the initial offset distances so that TGrid will know how far from the corresponding nodes on the previous layer to place the new nodes that define the prisms. See Section 10.8: [Offset Distances](#) for details.
- (d) Smooth the offset distances are smoothed to eliminate spikes and dips in the new prism layer. See Section 10.8: [Offset Smoothing](#) for details.
- (e) Project new nodes along the outer edges of the prism layer to adjacent zones, based on the analysis in step 1.
- (f) Create prism faces and cells using the new nodes. The new faces at the top of the prism layer are referred to as cap faces.
- (g) The new cap faces of skewness value greater than the specified threshold are improved by edge swapping or edge node smoothing. See Section 10.9.1: [Edge Swapping and Smoothing](#) for details.
- (h) Apply global smoothing across the new surface. This is also skewness-driven, but is based on the nodes not the faces. See Section 10.9.2: [Node Smoothing](#) for details.
- (i) Check the quality of all new cells and faces.
- (j) If quality problems are detected, prism layer creation is stopped. If more layers are to be grown, the process is repeated starting from substep (a).

3. Zone clean up: After all the prism layers have been grown, the new faces along the outer sides of the layers are moved to zones of the same types as the zones to which their nodes are projected. Unprojected faces are collected in a single `prism-side` zone.

4. Retriangulation: Adjacent triangular zones that have been projected to are automatically retriangulated. Triangles that are overlapped by new quadrilateral faces from the prism layers are removed. Most of the original nodes are used in the new triangulation. The result is a conformal interface between the sides of the prism layers and the adjacent triangular zones. An analogous procedure is used in 2D.

## 10.6 Using Adjacent Zones as the Sides of Prisms

For each zone adjacent to the zone(s) from which you grow prisms, TGrid will check the angle between the prism growth direction and the adjacent zone. Depending on the size of this angle, TGrid will decide whether to use the adjacent zone for the prism sides, or create a new zone.

### Using Adjacent Quadrilateral Face Zones

If an adjacent zone has quadrilateral faces, it will be used without modification as the prism-side boundary, and it satisfies the following requirements:

- It must share nodes with the boundary zone from which you are building the prisms i.e., there must be no free nodes (see Section 7.1.1: [Free and Isolated Nodes](#)) where the zones touch.
- The angle between the adjacent zone and the prism growth direction must be less than the specified threshold, **Max. Adjacent Zone Angle** in the **Project** tab of the **Prisms** panel. The default threshold value is 45 degrees.

For example, if you are growing prisms from the bottom surface in Figure 10.6.1 using the default maximum adjacent zone angle of 45 degrees, the zone on the left will be excluded, but the zone on the right will be used as it is. A new zone will be created for the left sides of the prisms.

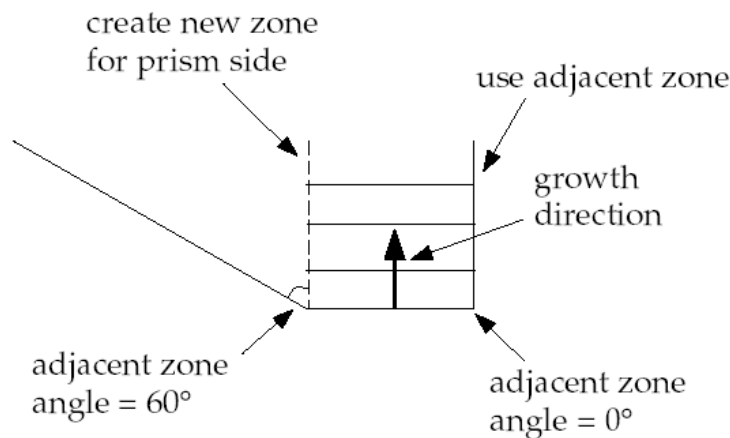


Figure 10.6.1: Effect of Adjacent Zone Angle

To fill in the gap on the left side of Figure 10.6.1, (in 2D) create a domain enclosing the gap and generate triangular cells, or (in 3D) create pyramids using the prism side faces, create a domain, and generate tetrahedral cells.

## Projecting to Adjacent Triangular Face Zones

If an adjacent zone has triangular faces, TGrid will perform two additional steps to incorporate the zone into the prism layers: projection and retriangulation. TGrid will project the outer nodes of the prisms onto the triangular faces of the adjacent zone, provided it satisfies the following requirements:

- It must share nodes with the boundary zone from which you are building the prisms. That is, there must be no free nodes (see Section 7.1.1: [Free and Isolated Nodes](#)) where the zones touch.
- The angle between the adjacent zone and the prism growth direction must be less than the specified threshold, **Max. Adjacent Zone Angle** (in the **Project** section of the **Prisms** panel). The default threshold value is 45 degrees.

In Figure 10.6.1, the zone on the right will be projected to and retriangulated, while the zone on the left will be excluded.

## Retriangulation

As shown in Figure 10.1.1, the sides of the prism will always be quadrilateral faces, regardless of the type of face from which a 3D prism is built (triangular or quadrilateral). If the prism-side nodes are projected to an adjacent triangular boundary face zone:

1. The prism-side faces on the shared boundary will be overlaid on the triangular faces of the existing boundary zone.
2. TGrid will retriangulate the triangular boundary zone so that the triangular portion of the zone ends at the border of the portion now filled with quadrilateral faces. This is done to obtain a conformal mesh where all the nodes match up at face edges.
3. The triangles that were overlapping the quadrilateral prism-side faces will be deleted.

Figure 10.6.2 shows the initial boundary mesh for a car and the symmetry boundary beside it. If the prisms are generated from the car body *without* retriangulation of the triangular symmetry boundary, the resulting boundary mesh will be as shown in Figure 10.6.3. With retriangulation, the boundary mesh will appear as in Figure 10.6.4.

Though it is possible to disable retriangulation (by disabling the **Retriangulate Adjacent Zones** option in the **Project** tab of the **Prisms** panel), it is not always recommended. Expert users experimenting with different settings may temporarily turn off retriangulation.

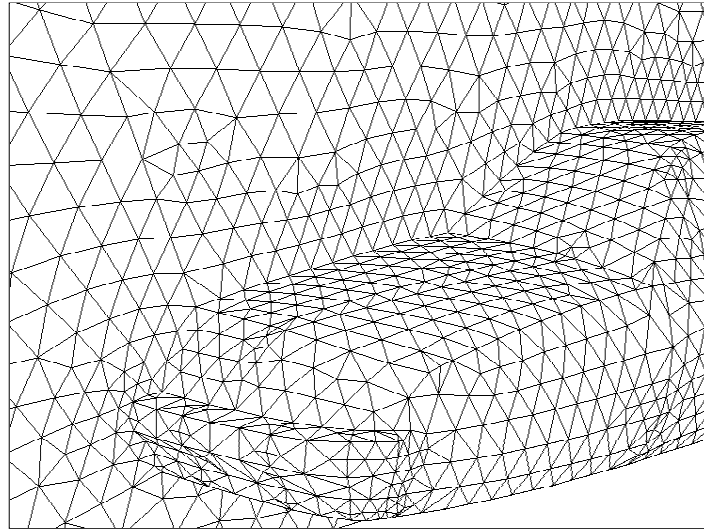


Figure 10.6.2: Symmetry Zone and Car Wall Before Prism Generation

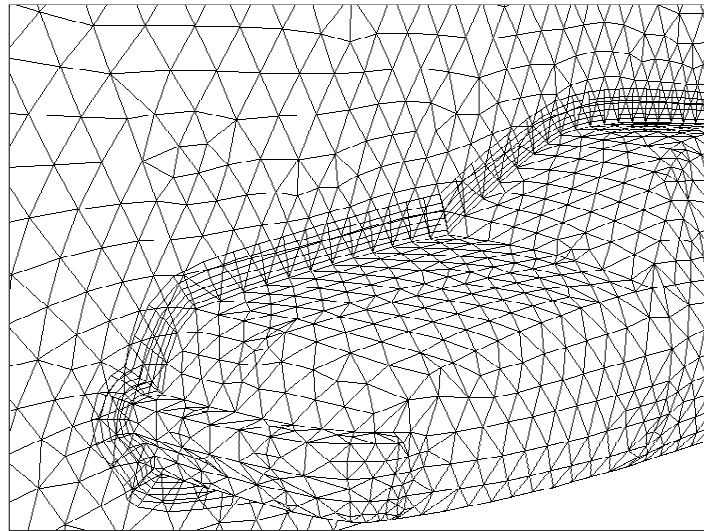


Figure 10.6.3: Symmetry Zone and Car Wall After Prism Generation *Without* Retriangulation



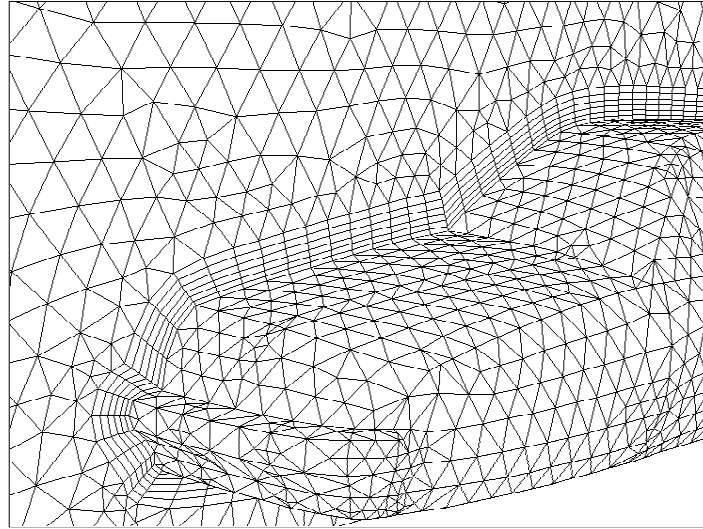


Figure 10.6.4: Symmetry Zone and Car Wall After Prism Generation and Retriangulation

## 10.7 Direction Vectors

As TGrid builds each layer of prisms, it needs to know which direction to build in. There are two methods available for determining the direction vectors:

- **Extrusion Method:** For creating straight-sided prism regions without any curvature, you can specify a uniform direction vector for all prism layers. Select **Uniform** in the **Method** list in the **Direction** tab of the **Prisms** panel.

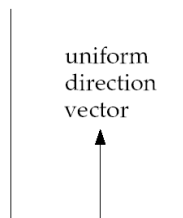


Figure 10.7.1: Uniform Direction Vector for a Straight-Sided Prism Region

Specify the uniform direction vector or click **Compute** in the **Vector** group box. TGrid will use this constant vector for all prism layers, instead of computing a new direction for each layer (see Figure 10.7.1).

- **Normal Method:** For regions with curvature, TGrid determines the appropriate normal direction vector at each node, since it may be different for each node and each layer, see Figure 10.7.2. Select **Normal** in the **Method** group box in the **Direct** tab of the **Prisms** panel.

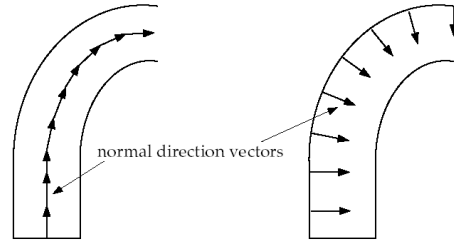


Figure 10.7.2: Normal Direction Vectors for a Curved Prism Region

### Normal Smoothing

The normal direction vectors obtained using the methods described in Section 10.7: [Direction Vectors](#) are smoothed so that there is a gradual change in direction from one node to the next. This will reduce the chance of direction vector intersection, which causes the prism generation to fail. This step is not necessary when a uniform direction vector is used, since there is no change in direction from one node to the next.

Figure 10.7.3 shows normal direction vectors in the vicinity of a sharp (90 degree) corner.

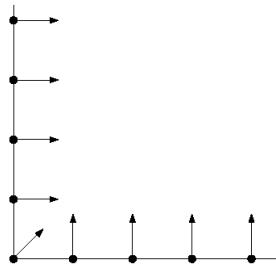


Figure 10.7.3: Normal Direction Vectors Before Smoothing

The direction vector is at an angle of 45 degree at the corner, while elsewhere it is 0 or 90 degrees. These direction vectors will intersect and prisms cannot be generated. Figure 10.7.4 shows the normal direction vectors near the corner after they have been smoothed. After smoothing, the normal direction changes more gradually from node to node.

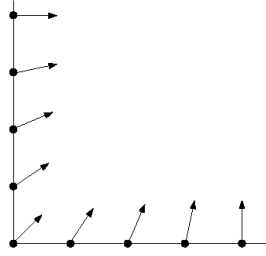


Figure 10.7.4: Normal Direction Vectors After Smoothing

It is possible to create very thin prism layers using the direction vectors shown in Figure 10.7.3, as long as the last layer lies below the point where the vectors cross.

## 10.8 Offset Distances

The offset distance for a given node is the distance between adjacent layers at that node. This distance is based on the prism layer height computed from the specified prism growth parameters. The new node for a prism layer is placed at this distance along the direction vector.

There are four methods available for determining the offset distances:

- **Uniform Offset Distance Method:** In this method, every new node (child) is initially the same distance away from its parent node (i.e., the corresponding node on the previous layer, from which the direction vector is pointing).

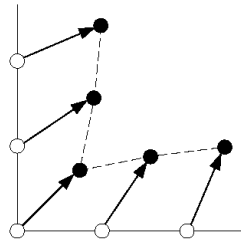


Figure 10.8.1: Uniform Offset Distance Method

Uniform distances are fine for planar surface meshes, but they result in crevices at sharp corners, for example, the 90 degree corner in Figure 10.8.1. The direction vector at the corner node will be at an angle of 45 degrees, while the vectors at

the adjacent nodes will be slightly more orthogonal to their faces. For uniform distances, this places the nodes for the first layer as shown in Figure 10.8.1.

The dashed line connecting the new nodes begins to pinch inward at the corner. The distances between parent and child nodes are still all the same, but the angles of the direction vectors introduce this pinching effect. Without some sort of correction such crevices can eventually collapse.

- Minimum Height Offset Distance Method:** In this method, child nodes are guaranteed to be *at least* as far from their parent nodes as the distance computed for the current layer from the growth inputs. TGrid will also try to retain the shape of the original boundary.

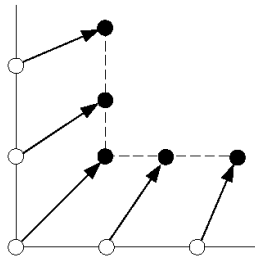


Figure 10.8.2: Minimum-Height Offset Distance Method

For a perfect 90 degree corner, this results in a perfect 90 degree corner for the next layer (see Figure 10.8.2). Thus, in this method the shape is better preserved if no additional smoothing is applied.

The minimum height method have some limitations. If the angle of the direction vector at the corner is not 45 degrees (i.e., if it does not exactly bisect the angle between the faces), the minimum-height method can increase skewness.

- Aspect Ratio Method:** This method allows you to control the aspect ratio of the prism cells that are extruded from the base boundary zone. The *aspect ratio* is defined as the ratio of the prism base length to the prism layer height. The various growth methods (constant, linear, geometric, and exponential) can be used to specify the aspect ratio of the first layer.

When a non-constant growth method is used, the aspect ratio for subsequent layers will change accordingly. The heights of the prism cells will vary according to the local face sizes in the surface mesh, providing a convenient way to automatically vary prism heights across a zone.

- Last Ratio Method:** This method also allows you to control the aspect ratio of the prism cells that are extruded from the base boundary zone. You can specify *First Height* for the first prism layer. If you select this method, the various

growth methods will not be accessible. The last aspect ratio method is explained in Figure 10.8.3.

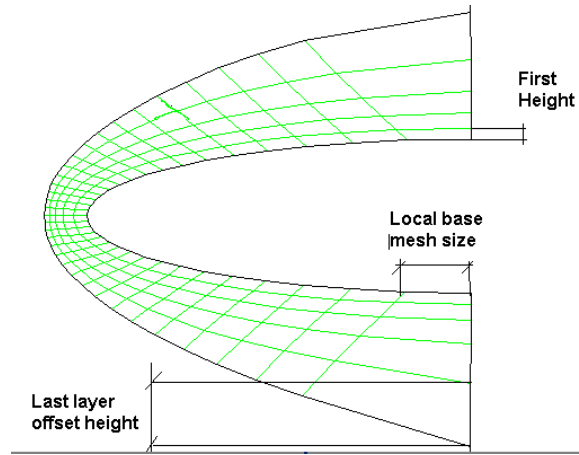


Figure 10.8.3: Last Ratio Method

Local base mesh size is used to find out the offset height for the last layer. For example, if you specify 80 as the **Last Percent** value the offset height of the last layer will be 0.8 times the local base mesh size. Local growth rate is used to calculate the other intermediate offset heights exponentially.

### Offset Smoothing

The purpose of offset smoothing is to eliminate spikes and dips in the new surface layer. Smoothing is applied iteratively. Figure 10.8.4 shows how the nodes are moved during offset smoothing.

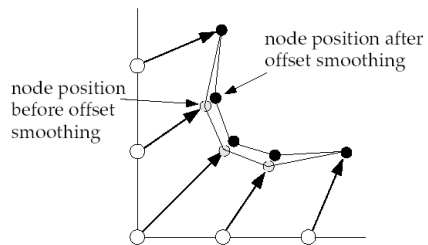


Figure 10.8.4: Effect of Offset Smoothing

### 10.9 Improving Prism Quality

TGrid performs the following to improve the skewness of prism faces:

- Edge swapping and smoothing (Section 10.9.1: [Edge Swapping and Smoothing](#))
- Node smoothing (Section 10.9.2: [Node Smoothing](#))

#### 10.9.1 Edge Swapping and Smoothing

After the new triangular cap faces are created for the new layer, the skewness is compared with the specified skewness threshold (Skewness in the **Swapping and Smoothing** group box in the **Improve** tab of the **Prisms** panel). If the skewness of a face is too high, TGrid performs the following improvement procedure:

1. Swap the longest edges of highly skewed faces.

When edge swapping occurs on a cap face, swapping propagates downward. Swapping must be performed on the corresponding faces on all prism layers, including the face on the original boundary mesh, so that the grid lines within the prism layers do not cross.

Swapping is *not* performed if it will significantly alter the geometry defined by the original boundary mesh.

2. If swapping is not performed, TGrid will try to improve the skewness by smoothing the nodes of the skewed faces using one of the following operations.
  - If the sharpest angle at the node (i.e., the most acute angle between adjacent faces that use the node) is nearly 180 degrees, TGrid will perform smoothing (as described in Section 10.9.2: [Node Smoothing](#)) to move the nodes.
  - If the sharpest angle is far from 180 degrees, TGrid will limit the smoothing so that the nodes are only allowed to move along the face's longest edge. This method will prevent the collapse of sharp corners in the mesh.

Smoothing does *not* propagate downward, it is performed only on the skewed cap faces themselves.

This edge swapping/smoothing procedure is not performed on quadrilateral cap faces.

#### 10.9.2 Node Smoothing

If the edge swapping and smoothing are not sufficient to reduce the skewness of some faces, include an additional smoothing operation. In this method, if any of the faces surrounding a new node have a skewness greater than the specified threshold, the node is smoothed.

Instead of moving only the node, this method will also move the surrounding nodes to make space for the node to move.

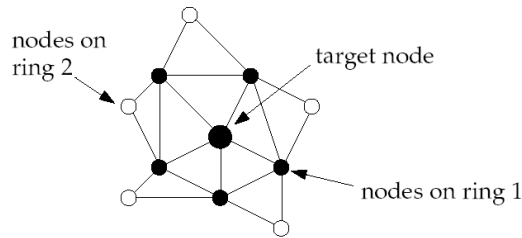


Figure 10.9.1: Node Smoothing in Rings

The surrounding nodes are grouped and smoothed in rings (see Figure 10.9.1). The nodes on the outermost ring are smoothed first, etc., until the target node (on ring zero) is finally smoothed.

This type of smoothing is helpful in concave areas where the advancing layers continually decrease the available surface area, causing squashing of triangular faces. By smoothing outer rings of nodes first, more space is provided for the nodes that need to be moved in order to reduce the skewness. Thus, this method performs regional smoothing around faces of high skewness. The node smoothing procedure can be performed on both tri and quad faces.

### 10.9.3 Improving Warp

Improving warp in prisms is a postprocessing step which is carried out after all the required prism layers are created. TGrid allows you to improve the warp in the generated prism faces by moving the nodes of the face to make it planar.

When the **Improve** option is enabled in the **Warp** group box in the **Improve** tab of the **Prisms** panel (see Section 10.10: [The Prisms Panel](#)), TGrid identifies the prism layer(s) having faces with warp  $\geq 0.5$ . The nodes on the identified faces are moved and the new location of the node is updated only when:

1. The overall maximum warp of the faces connected to the node has decreased.
2. The overall maximum skewness of the cells connected to the node has decreased or remained the same.

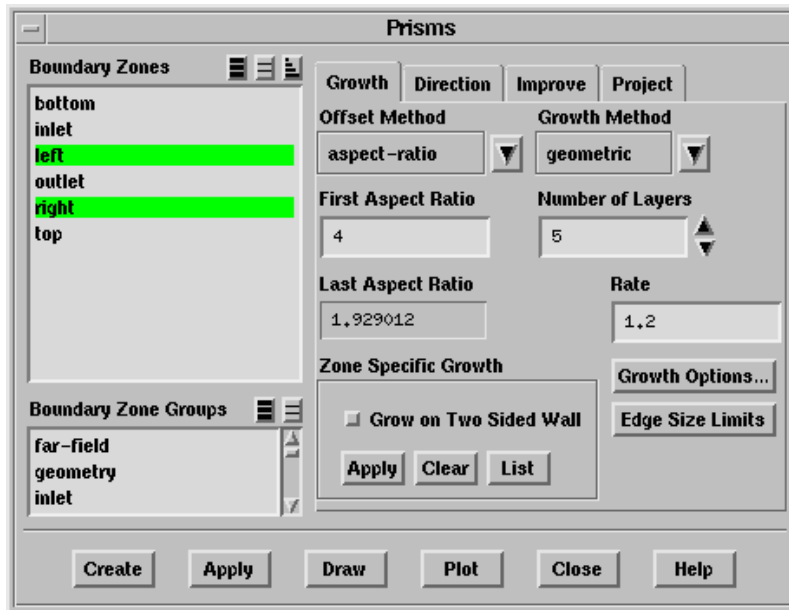
This is useful in regions having complex sharp corners, where the nodes of a prism cell may have drastic normal change.

## 10.10 The Prisms Panel

The Prisms panel allows you to specify parameters needed for prism cell creation, and then create the prisms. When you press any of the tab buttons along the top of the panel, the panel contents change to show the related controls. The controls for all inactive tabs are hidden.

The Prisms panel consists of a series of tabbed frames, within the main panel, listed horizontally across the top of the panel. Each frame has a name and a set of related controls.

Mesh → Prisms...



### Controls

**Boundary Zones** contains a selection list of all of the boundary zones in your mesh. You will select the zone(s) from which you want the prisms to be built.

**Boundary Zone Groups** contains a list of boundary zone types and user-defined groups. If you select a boundary type (or group) from this list (e.g., inlet), all boundary zones of that type (or group) will be selected in the **Boundary Zones** list. This shortcut allows you to easily select all boundary zones of a certain type without having to select each zone individually. You can select multiple boundary types in the **Boundary Zone Groups** list to select all zones of several different types (e.g., inlet and outlet).

You can also select **tri** or **quad** in this list to select all triangular or quadrilateral boundary zones.



**Growth** displays the controls for computing the height of each layer of prisms. This is the default tab of the **Prisms** panel.

**Offset Method** contains a drop-down list of the methods available for computing the offset distance. See Section 10.8: [Offset Distances](#).

**uniform** specifies the uniform offset method.

**aspect-ratio** specifies the aspect-ratio offset method.

**last-ratio** specifies the last-ratio offset method.

**minimum-height** specifies the minimum-height offset method.

**Growth Method** contains a drop-down list of the methods available for computing the prism height. Figure 10.10.1 shows layer heights for all four methods, using a first height of 1 and a slope/rate/exponent of 1.2.

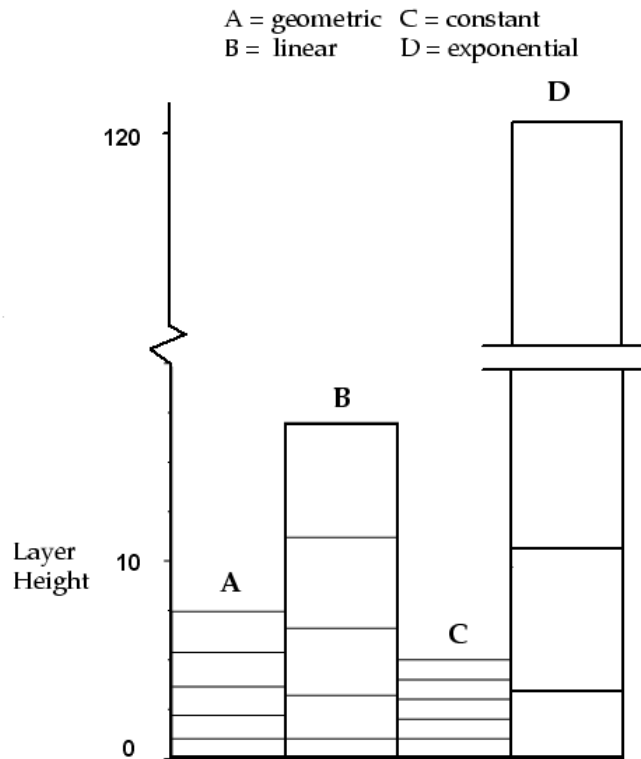


Figure 10.10.1: Layer Heights Computed Using the Four Growth Methods

**constant** enables the specification of a constant height for all prism layers. When this method is selected, you will need to specify the **First Height** and the **Number of Layers**. The height of all prisms will be the **First Height** ( $h_1$ ).

**linear** enables the specification of a variable height that decreases or increases linearly for each layer of prisms:

$$h_n = h_1 + s(n - 1) \quad (10.10-1)$$

When this method is selected, you will need to specify the **First Height** ( $h_1$ ), the **Number of Layers**, and the **Slope** ( $s$ ).

**geometric** enables the specification of a variable height that decreases or increases geometrically for each layer of prisms:

$$h_n = r^{n-1}h_1 \quad (10.10-2)$$

When this method is selected, you will need to specify the **First Height** ( $h_1$ ), the **Number of Layers**, and the **Rate** ( $r$ ).

**exponential** enables the specification of a variable height that decreases or increases exponentially for each layer of prisms:

$$h_n = h_1e^{(n-1)p} \quad (10.10-3)$$

When this method is selected, you will need to specify the **First Height** ( $h_1$ ), the **Number of Layers**, and the **Exponent** ( $p$ ).

**First Height** specifies the height of the first prism layer ( $h_1$ ). The computed heights of subsequent prism layers will be based on this height, depending on the **Growth Method** selected.

**Number of Layers** sets the total number of prism layers to be created.

**Total Height** reports the total height of the prisms to be grown. You can use this information to check if the specified prism heights will yield a mesh of the required height.

If you want to see the updated **Total Height**, press the <Enter> key after entering a new value for the other growth parameters. If you do not press <Enter> after entering the value, the **Total Height** reported will not reflect the changed value.

**First Aspect Ratio** specifies the first aspect ratio to be used when the **aspect-ratio** offset method is selected.

**Last Aspect Ratio** reports the last aspect ratio based on the specified first aspect ratio for the **aspect-ratio** offset method.

**Last Percent** specifies the percentage to be used along with the **First Height** when the **last-ratio** offset method is selected.

**Slope** sets the slope ( $s$ ) of the linear function used to compute the heights of the prism layers when the **linear** growth method is selected.

**Rate** sets the rate of growth ( $r$ ) of the geometric function used to compute the heights of the prism layers when the **geometric** growth method is selected.

**Exponent** sets the exponent ( $p$ ) for the exponential function used to compute the heights of the prism layers when the **exponential** growth method is selected.

**Growth Options...** opens the Prisms Growth Options panel (see Section 10.10.1: [The Prisms Growth Options Panel](#)).

**Edge Size Limits** displays (in an **Information** dialog box and in the console) the minimum, maximum, and average edge lengths for all selected **Boundary Zones**. The **First Height** should usually be smaller than the minimum length reported.

**Zone Specific Growth** allows you to simultaneously grow prisms from multiple zones using *different* growth parameters.

By default, this option will be enabled if you read in a mesh for which separate growths were specified for different zones and then subsequently saved. It will be disabled if you read in a mesh that does not have saved prism parameters.

You need not use any of these controls if you are building prisms from just one zone, building them from multiple zones using the same growth parameters, or growing layers for multiple zones at different times.

**Grow on Two Sided Wall** allows you to grow prisms on both sides of a two sided wall.

**Apply** applies the specified growth parameters to the selected **Boundary Zones**.

**Note:** *When setting prism parameters for use with the Auto Mesh option, it is necessary to apply the prism growth parameters on specific zones for TGrid to retain the growth parameters in memory. The Prisms option in the Auto Mesh panel will be visible only after applying zone-specific growth (by clicking Apply in the Zone Specific Growth group box).*

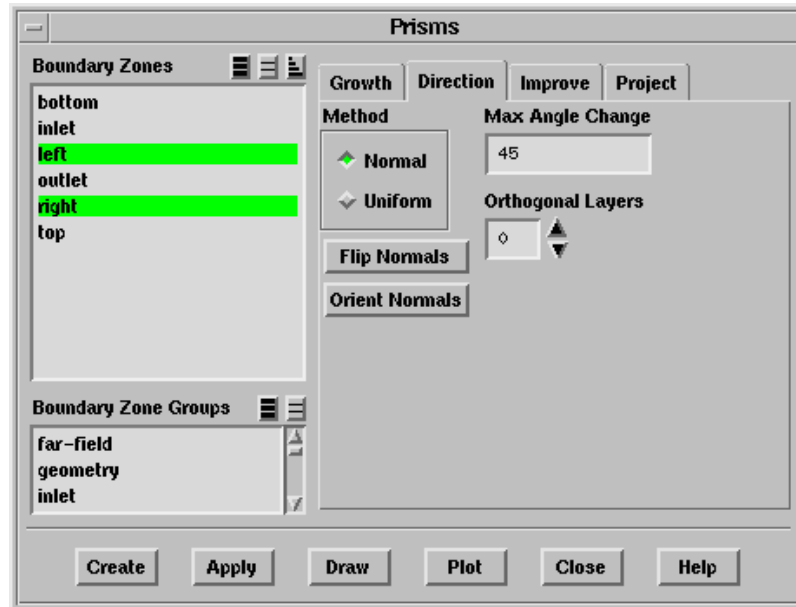
**Clear** clears the zone-specific growth settings attached to the zones selected in the **Boundary Zones** list.

**List** lists the zone-specific growth settings for the zones selected in the **Boundary Zones** list in the console.

**Plot** displays (in the active graphics window) an illustration of the prism layers, allowing you to preview the layer heights before actually creating the prisms. The plot will show the layer heights for all selected boundary zones, and (if different) the heights for the current settings in the panel.

**Print** reports the growth parameters, prism height for each layer, and total height in the TGrid console. The report will show the data for all selected zones, and (if different) the data for the current settings in the panel.

Direction displays the controls for computing the growth directions for the nodes on each layer of prisms.



Method lists the available methods for specifying the prism growth direction. See Section 10.7: [Direction Vectors](#) for details about these methods.

**Normal** enables the computation of direction vectors normal to the boundary zones from which you are growing.

**Uniform** enables a uniform direction specification for all nodes on all prism layers. When you select this method, you need to specify the direction Vector.

TGrid will use this constant vector for all nodes on all prism layers, instead of computing a new direction for each node on each layer. Therefore, this option is most useful for creating straight-sided prism regions without any curvature.

**Flip Normals** reverses the normal directions on the faces of the selected boundary zone(s). The normal directions will determine the side of the boundary zone on which prisms will be built.

**Orient Normals** orients contiguous faces (i.e., faces that touch each other) in a selected zone so that their normals are all consistently pointing on the same side of the zone (see Figure 7.13.1).

**Max Angle Change** specifies the maximum angle by which the normal direction at a node can change during smoothing. See Sections 10.3.5 and 10.7 for details about this parameter. This option is available only when the **Normal** method is selected.

**Vector** specifies the uniform direction for all layers of prisms. You can either enter values in the X, Y, Z fields, or click the **Compute** button to get the average normal direction of the selected zone(s). This option is available only when the **Uniform** method is selected.

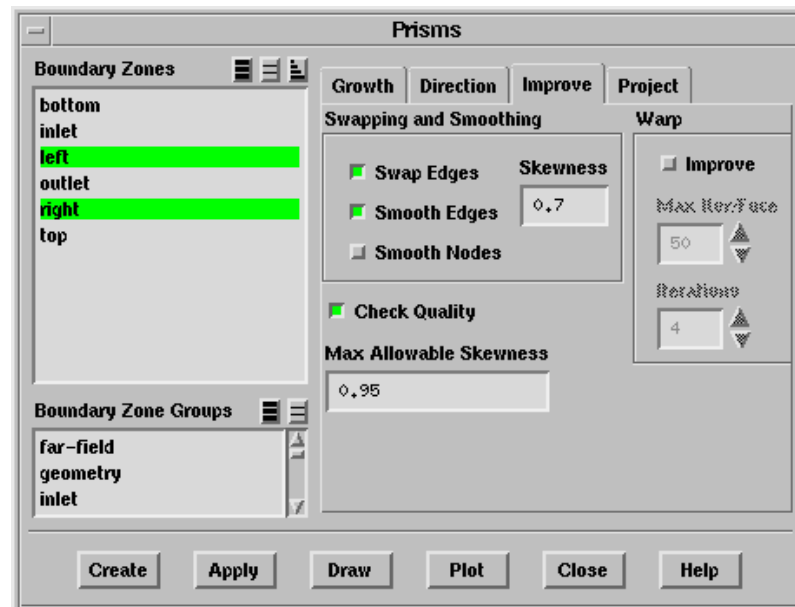
**Compute** computes the average normal direction to the selected zone or zones (i.e., the average of the unit normals of all the faces in the zones). This direction is usually a good choice for the uniform direction, especially if the zone is planar. This option is available only when the **Uniform** method is selected.



If you use the **Compute** button to calculate the average normal direction, do not change the sign of the displayed **Vector**. It is important for the uniform direction vector to have the same sign as the normal direction for the face zone.

**Orthogonal Layers** specifies the number of layers that should remain orthogonal to the original boundary. For an orthogonal layer, TGrid will perform no edge swapping or normal, edge, or node smoothing. See Section 10.3.5: [Preserving Orthogonality](#) for details. This option is available only when the **Normal** method is selected and it is set to zero by default.

**Improve** contains the controls for various procedures used to improve the quality of the prisms. These controls are not used if the **Uniform** direction method is selected.



**Swapping and Smoothing** contains controls for the edge swapping and smoothing, and node smoothing.

**Swap Edges** enables/disables edge swapping (see Section [10.9.1: Edge Swapping and Smoothing](#)) to decrease the skewness of highly skewed faces. This option is enabled by default.

**Smooth Edges** enables/disables edge smoothing (see Section [10.9.1: Edge Swapping and Smoothing](#)) to decrease the skewness of highly skewed faces. This option is enabled by default.

**Smooth Nodes** enables/disables node smoothing (see Section [10.9.2: Node Smoothing](#)) to decrease the skewness of highly skewed faces. This option is disabled by default.

**Skewness** specifies the face skewness above which edge swapping and smoothing are triggered. Only faces having skewness greater than this value are evaluated for these operations. The default value is 0.7.

**Warp** contains options for improving face warp.

**Improve** allows you to improve face warp (see Section [10.9.3: Improving Warp](#)) during prism generation. This option is disabled by default.

**Max Iter/Face** specifies the maximum number of attempts for improving a particular face.

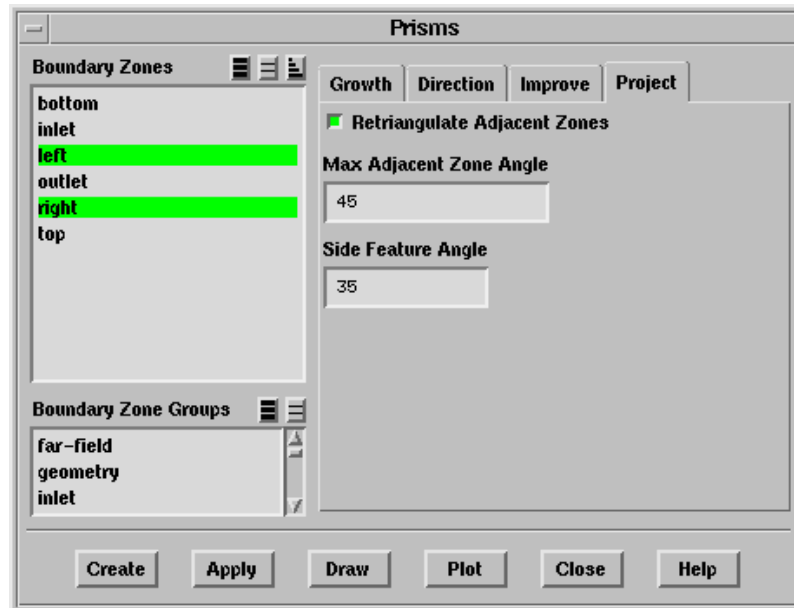
**Iterations** specifies the number of iterations through the warp improvement procedure.

**Check Quality** enables/disables the checking of face and cell quality after each prism layer is created. When this option is enabled (the default setting), TGrid will check the cell volume of every new cell, the face-handedness of every face on every new cell, the skewness of all new cap faces (3D only), and the skewness of all new cells (2D only).

If any new cell has zero or negative volume or any left-handed faces, or if the skewness exceeds the specified **Max. Allowable Skewness**, prism layer creation will be stopped. See Section [10.12: Negative Volumes/Left-Handed Faces/High Skewness](#) for details.

**Max. Allowable Skewness** specifies the maximum skewness allowed for a prism cap face (in 3D) or a prism cell (in 2D). If the skewness exceeds the specified value, TGrid will stop the meshing process and warn you that the skewness for that layer is too high. You can then examine the skewed faces or cells, make appropriate corrections, and recreate the prisms. See Section [10.12: Negative Volumes/Left-Handed Faces/High Skewness](#) for details.

Project contains controls for projection and retriangulation.



**Retriangulate Adjacent Zones** specifies whether or not the triangular face zones adjacent to which outer nodes have been projected will be automatically retriangulated. This option is enabled by default. See Section 10.6: [Retriangulation](#) for details about this procedure.

**Max Adjacent Zone Angle** sets the tolerance used to determine whether or not to use an adjacent zone. If a zone shares outer nodes with any of the zones from which the layers are being grown (the “base zones”), its angle with respect to the growth direction is compared with the specified value. If the angle is less than or equal to this value, then the zone will be used. See Section 10.6: [Using Adjacent Zones as the Sides of Prisms](#) for details. The default is 45 degrees.

**Side Feature Angle** specifies the angle used for computing the feature normals.

**Create** creates the prisms using the settings specified in the panel.

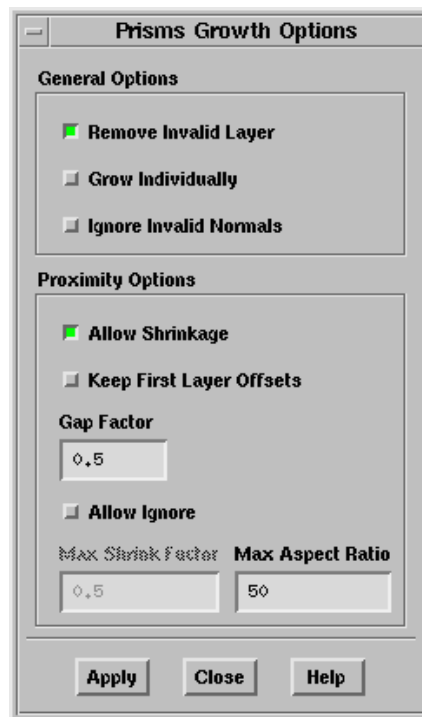
**Apply** applies all current settings for prism growth, without creating the prisms. If you wish to set the parameters in advance and return later to generate the prisms, click **Apply** before closing the Prisms panel. There is no need to click **Apply** before moving from one tab in the panel to another (e.g., from the **Growth** tab to the **Direct** tab).

**Note:** *Clicking Apply only applies the parameters set in the Prisms panel. When setting prism parameters for use with the Auto Mesh option, click Apply in the Zone Specific Growth group box for TGrid to retain the growth parameters on specific zones in memory. The Prisms option in the Auto Mesh panel will be visible only after applying zone-specific growth.*

Draw displays the boundary zone(s) selected in the **Boundary Zones** list. All current settings in the **Display Grid** panel (see Section 14.1.3: [The Display Grid Panel](#)) will be reflected in the resulting grid display. You may find it useful to enable the display of normal vectors so that you can easily determine the normal direction(s) of the selected boundary zone(s).

### 10.10.1 The Prisms Growth Options Panel

The Prisms Growth Options panel allows you to specify the parameters needed for detecting proximity/collision, retaining the first layer offset, and growing prisms individually from multiple zones.



#### Controls

General Options contains additional options for prism growth.

**Remove Invalid Layer** allows you to retain the valid prism layers grown before TGrid reports prism layer failure.

**Grow Individually** indicates whether individual prism cell zones and prism cap face zones should be retained or merged with like zones. By default this option is disabled, so a single zone of prism cells and a single zone of cap faces will be created for each set of simultaneously grown layers. See Section 10.3.1: [Growing Prisms Simultaneously from Multiple Zones](#) for details.



**Ignore Invalid Normals** allows you to ignore regions of invalid normals while creating prisms.

**Proximity Options** contains options for proximity detection.

**Allow Shrinkage** enables the shrinking of prism layers in areas of proximity to avoid prism collision.

**Keep First Layer Offsets** allows you to retain the original first offset height while scaling offset heights in case of intersecting prism layers.

**Gap Factor** controls the gap between the intersecting prisms layers with respect to the cell size of the prisms.

**Allow Ignore** allows you to ignore prism growth parameters in areas of proximity to avoid prism collision.

**Max Shrink Factor** determines the shrinkage of prism layers to avoid collision. This option is available only when the **Allow Ignore** option is enabled.

**Max Aspect Ratio** specifies the maximum aspect ratio to determine the limit for the shrinkage of prism layers. This option is available only when the **Allow Ignore** option is disabled.



The value specified for **Max Aspect Ratio** is only used as a guideline for limiting the shrinkage of the prism layers. The specified value may not always be adhered to in the final mesh as the value may be relaxed during the prism improvement operations such as smoothing, etc.

## 10.11 Text Interface for Generating Prisms

Text commands for generating prisms are listed below. Some of them allow additional control that is not available in the **Prisms** panel.

`/mesh/prism/create` creates prism layers on the specified face zone(s) according to the specified parameters.

`/mesh/prism/controls/adjacent-zone/project-adjacent-angle` sets the tolerance used to determine whether or not to use an adjacent zone. If a zone shares outer nodes with any of the zones from which the layers are being grown (the “base zones”), its angle with respect to the growth direction is compared with this value. If the angle is less than or equal to this value, then the zone will be used. See [Section 10.6: Using Adjacent Zones as the Sides of Prisms](#) for details.

`/mesh/prism/controls/adjacent-zone/project-converged` sets the convergence criterion for iterative projection. This is non-dimensionalized by the offset height at each local node.

`/mesh/prism/controls/adjacent-zone/project-iter` sets the maximum number of iterations to perform when projecting to multiple zones. Most projections converge in only a few iterations. There is normally no need to change this value.

`mesh/prism/controls/adjacent-zone/project?` enables/disables projection of outer nodes to adjacent zones.

`mesh/prism/controls/adjacent-zone/retri-feature-angle` allows you to specify the feature angle that should be prevented while generating prisms.

`mesh/prism/controls/adjacent-zone/retriangulate-adjacent?` specifies whether or not adjacent triangular face zones to which outer nodes have been projected will be automatically retriangulated. This option is on by default.

`/mesh/prism/controls/adjacent-zone/side-feature-align-angle` specifies the angle used for aligning projected normals along a feature edge.

`/mesh/prism/controls/adjacent-zone/side-feature-angle` specifies the angle used for computing the feature normals.

`/mesh/prism/controls/adjacent-zone/side-topology-align-angle` specifies the angle used for aligning projected normals along a particular feature edge based on the topology. This is particularly useful when the `side-feature-angle` specified is not sufficient to decide the feature edge to align the projected normals.

`mesh/prism/controls/auto-separate-cells?` toggles automatic movement of the cells extruded from different face zones to different cell zones.

`/mesh/prism/controls/check-quality?` enables/disables the checking of volume, skewness, and handedness of each new cell and face.

`/mesh/prism/controls/grow-individually?` specifies whether TGrid should grow prisms from multiple zones individually so that the individual prism cell zones and the individual cap face zones are retained.

`/mesh/prism/controls/improve/check-allowable-skew?` allows you to check the skewness of the prism cap for every layer.

`/mesh/prism/controls/improve/check-size?` enables the checking of cell size during the generation of each prism layer. TGrid will report an error if zero-area prism cells are generated and prism layer growth will be stopped.

*When the `check-size?` option is disabled, the zero-area prism cells generated can be removed during the improvement smoothing operations performed at the end of the prism generation procedure.*

`/mesh/prism/controls/improve/edge-smooth?` enables/disables local smoothing of nodes of the longest edges of skewed faces.

`/mesh/prism/controls/improve/edge-smooth-angle` specifies the maximum allowable angle between the normals of adjacent cap faces for skewness-driven edge smoothing.

`/mesh/prism/controls/improve/edge-swap?` enables/disables edge swapping to decrease the skewness of highly skewed faces.

`/mesh/prism/controls/improve/edge-swap-base-angle` specifies the maximum allowable angle between a pair of base face normals for skewness-driven edge swapping.

`/mesh/prism/controls/improve/edge-swap-cap-angle` specifies the maximum allowable angle between a pair of cap face normals for skewness-driven edge swapping.

`/mesh/prism/controls/improve/face-smooth?` allows you to enable face-driven smoothing to improve skewness.

`/mesh/prism/controls/improve/face-smooth-converged` specifies the convergence criteria for cap face smoothing.

`/mesh/prism/controls/improve/face-smooth-rings` specifies the number of rings to propagate skewness during smoothing of cap faces.

`/mesh/prism/controls/improve/face-smooth-skew` specifies the minimum skewness to smooth cap faces.

`/mesh/prism/improve/improve-warp?` enables/disables improving of face warp during prism generation. This option is disabled by default.

`/mesh/prism/controls/improve/left-hand-check` specifies checking for left-handedness of faces. A value of 0 implies face handedness will not be checked, 1 implies only cap faces will be checked, while 2 implies faces of all cells in current layer will be checked.

`/mesh/prism/controls/improve/max-allowable-skew` specifies the maximum allowable skewness for cap faces after smoothing. Prism layer growth will be stopped if any cap face has skewness greater the specified value.

`/mesh/prism/controls/improve/node-smooth?` allows you to enable node-driven smoothing to improve skewness.

`/mesh/prism/controls/improve/node-smooth-angle` refers to the maximum deviation of a node's sharpest angle (i.e., the most acute angle between adjacent faces that use the node) from 180 degrees. The node will be smoothed only if its sharpest angle falls within this range. For example, if the specified value is 45 degrees, nodes with sharpest angles between 135 and 225 degrees can be smoothed.

`/mesh/prism/controls/improve/node-smooth-converged` sets the convergence criterion for node smoothing. If the node positions are changing by less than this value, smoothing iterations will stop.

`/mesh/prism/controls/improve/node-smooth-iter` specifies the maximum number of node smoothing iterations to be performed for the nodes on each layer. These iterations will be performed until the convergence criterion is reached.

`/mesh/prism/controls/improve/node-smooth-local?` allows you to enable node smoothing to converge locally. This is useful for large geometries.

`/mesh/prism/controls/improve/node-smooth-rings` controls the locality of node smoothing by setting the number of rings around each node to be smoothed.

- If zero, only a node itself is smoothed. If one, the node and all of its neighbors are smoothed.
- If two, the neighbors of the neighbors are also smoothed, and so on.

`/mesh/prism/controls/improve/shrink-left-handed-cap?` allows you to enable shrinking of prism layers to get rid of left handed faces.

`/mesh/prism/controls/improve/swap-smooth-skew` specifies the skewness threshold for edge swapping and edge and node smoothing. The faces with skewness greater than or equal to the specified value will be swapped and/or smoothed.

`/mesh/prism/controls/merge-ignored-threads?` allows you to automatically merge all ignored zones related to a base thread into a single thread. This option is enabled by default. When this option is disabled, TGrid will generate more than one thread per base thread.

`mesh/prism/normal/bisect-angle` is required for growing prisms out of sharp interior corners. When the value of this angle is set, the normals are automatically projected onto the plane bisecting the angle between faces having an interior angle less than this angle.

`/mesh/prism/normal/converge-locally?` specifies whether or not the normal smoothing at each node is frozen once convergence is satisfied at that node. If not, all normals are continuously smoothed until all of them have converged.

`/mesh/prism/controls/normal/direction-method` specifies whether the prism layers should be grown normal to surfaces or along a specified direction vector.

`/mesh/prism/controls/normal/ignore-invalid-normals?` allows you to ignore nodes which have poor normals.

`/mesh/prism/controls/normal/intersect-detect?` enables the adjustment of normals to avoid their intersection.

`/mesh/prism/controls/normal/intersect-detect-angle` specifies the angle based on which normal intersection is determined. The normal intersection is determined in cases when the feature angle of the manifold of a node is less than the value specified.

`/mesh/prism/controls/normal/max-angle-change` specifies the maximum angle by which the normal direction at a node can change during smoothing.

`mesh/prism/controls/normal/normal-method` specifies the method to use for normal direction computation (see Section 10.7: [Direction Vectors](#)).

`acute-bisection` is a variation of that proposed by Kallinderis et al. [?],[?][?, ?].

The faces surrounding a node are analyzed to find those whose planes form the most acute angle. The resulting normal lies on the plane bisecting those two planes.

This method is computationally more expensive than surface averaging, but is less robust at singular points where convex edges meet smooth surfaces. For example, if a box is sitting on the table, such a point would be where one of the box's corners meets the table top.

`face-average` computes the normal at a node by averaging the normals of the surrounding faces. This is the only method available in 2D.

`hybrid-normal` computes the normal by all the methods available and uses the best quality normal obtained. This is the default method in TGrid.

`minimized-angle` is a variation of that proposed by Pirzadeh [?],[?]. An initial normal at a node is determined by averaging the normals of the surrounding faces (the face-average method). Iterations are then performed to minimize the largest angle between the node normal and the normals of the surrounding faces.

This method is a computationally more expensive than surface averaging, and can produce slightly better normals when the uniform offset method is used and the surface mesh does not contain sharp edges. When the minimum-height offset method is used, this direction method may introduce spikes and pits in the advancing layer surfaces.

`surface-average` is similar to `face-average`, but handles sharp edges better. This is the default method in 3D.

`/mesh/prism/controls/normal/orthogonal-layers` specifies the number of layers to preserve orthogonality. All smoothing is deferred until after these layers have been generated.

`/mesh/prism/normal-smooth?` enables or disables smoothing of normal direction vectors.

`mesh/prism/normal-smooth-converged` sets the convergence criterion (in degrees) for normal smoothing. If the normal directions are changing by less than this value, smoothing iterations will stop.

`/mesh/prism/normal-smooth-iter` specifies the maximum number of normal smoothing iterations to be performed for the normal vectors on each layer. These iterations will be performed until the convergence criterion is reached.

`/mesh/prism/controls/offset/smooth?` enables/disables offset distance smoothing.

`/mesh/prism/controls/offset/smooth-converged` sets the convergence criterion for offset smoothing. If the offset heights change by less than this value, smoothing iterations will stop.

`/mesh/prism/controls/offset/smooth-iter` sets the maximum number of offset smoothing iterations to be performed. These iterations will be performed until the convergence criterion is reached.

`/mesh/prism/controls/proximity/allow-ignore?` allows you to ignore nodes where specified maximum shrink factor cannot be maintained.

`/mesh/prism/controls/proximity/allow-shrinkage?` allows you to enable shrinkage while growing prism layers.

`/mesh/prism/controls/proximity/gap-factor` specifies the gap between prism layers in the proximity region.

`/mesh/prism/controls/proximity/keep-first-layer-offsets?` allows you to retain first layer offsets while performing proximity detection.

`/mesh/prism/controls/proximity/max-aspect-ratio` specifies the minimum offset to avoid degenerate cells. This option is available only when the `allow-ignore?` option is disabled.

`/mesh/prism/controls/proximity/max-shrink-factor` specifies the shrink factor determining the maximum shrinkage of the prism layers. This option is available only when the `allow-ignore?` option is enabled.

`/mesh/prism/controls/proximity/smoothing-rate` specifies the rate at which shrinkage is propagated in lateral direction.

`/mesh/prism/controls/remove-invalid-layer?` removes the last prism layer if it fails in the quality check.

`/mesh/prism/controls/zone-specific-growth/apply-growth` applies the zone-specific growth parameters specified.

`/mesh/prism/controls/zone-specific-growth/clear-growth` clears the zone-specific growth specified.

`/mesh/prism/controls/zone-specific-growth/list-growth` lists the zone-specific growth parameters specified for individual zones in the console.

`/mesh/prism/improve/improve-prism-cells` smoothes cells by considering rings of cells around the prism cells.

`/mesh/prism/improve/smooth-brute-force?` forcibly smooth cells if cell skewness is still high after regular smoothing.

`/mesh/prism/improve/smooth-cell-rings` specifies the number of cell rings to be considered around the skewed cells (to be used by `improve-prism-cells`).

`/mesh/prism/improve/smooth-prism-cells` specifies optimization based smoothing of prism cells.

`/mesh/prism/improve/smooth-sliver-skew` specifies the skewness above which prism cells will be smoothed.

`/mesh/prism/list-parameters` lists all prism mesh parameters in the console.

`/mesh/prism/reset-parameters` resets the prism parameters to their defaults.



If you read in a mesh file created in a previous version of TGrid, you can reset all the prism parameters using this command before proceeding with setting the prism parameters.

`/mesh/revolve-face-zone` generates prisms by revolving a specified face zone through a specified angle about a specified rotation origin and axis to produce a polar mesh.

## 10.12 Prism Meshing Problems

This section discusses a number of common problems that you may encounter when generating prismatic meshes. A appropriate solution is also recommended for each problem. If the prism generation fails or is unsatisfactory, read in the mesh you saved before starting the prism generation (in step 6 of Section 10.2: Procedure for Generating Prisms) and repeat the process, incorporating the appropriate corrections.

The most common prism meshing problems are:

- Orientation
- Retriangulation failures
- Too many or too few nodes, or unknown face combinations

- Negative volumes, left-handed faces, or high skewness
- Growing too far
- Large jumps in prism height at the edges of layers

### Orientation

If the faces of the boundary zone from which you are building prisms are not all oriented in the same direction (Figure 7.13.1), TGrid will stop the prism generation process immediately and will alert you that it has reoriented the faces.

**Solution:** In this case, do the following:

1. Check the normal direction vectors on the boundary zone, to ensure that they have not been oriented to face in the wrong direction.
2. Follow the instructions in step 4 in Section 10.2: [Procedure for Generating Prisms](#) to check and, if necessary, correct (flip) the normal direction.
3. Click **Create** to restart the prism generation process.

In this case, there is no need to reread the mesh because TGrid stops before it actually creates any prism layers.

### Retriangulation Failures

If nodes, where adjacent zones meet the zones (from which prisms are growing) are duplicated, TGrid will be unaware that the zones are connected. It will ignore such zones when considering nodes for projection, and hence retriangulation is likely to fail.

**Solution:** To verify this problem check the messages printed before the first prism layer. It should contain a list of all the zones connected to the zones from which you are growing the prisms.

1. Make sure there are no free nodes in the surface mesh before growing any prisms.
2. Use the **Count Free Nodes** button in the **Merge Boundary Nodes** panel (see Section 7.1.2: [The Merge Boundary Nodes Panel](#)) to check for free nodes.
3. Turn on the **Free** option in the **Faces** section of the **Display Grid** panel, to see where the free nodes are located.



### Too Many/Few Nodes or Unknown Face Combinations

If you receive the following type of messages, then you are most likely growing prisms into existing cells.

```
Warning: wedge cell c887 has too many nodes
Warning: wedge cell c1928 has too few nodes
Warning: base_mask = 7 (0x7), unknown face combinations
```

Either you have already grown prisms from the selected zones, or you are growing them in the wrong direction into cells on the other sides of the selected zones. In the latter case, click the **Flip Normals** button in the **Direct** section of the **Prisms** panel before growing the layers. See step 4 in Section [10.2: Procedure for Generating Prisms](#).

### Negative Volumes/Left-Handed Faces/High Skewness

TGrid will automatically stop the prism layer creation process if it detects negative cell volumes, left-handed faces, or high skewness.

Left-handed faces are faces that have collapsed into their cells. You can view them by turning on the **Left Handed** option in the **Faces** section of the **Display Grid** panel. Very high skewness occurs when the value of skewness is greater than the specified **Max. Allowable Skewness** in the **Improve** section of the **Prisms** panel). These problems can occur where prism layer fronts are advancing too quickly in areas of high curvature.

For example, the first layer height may be greater than the minimum edge length of your original surface mesh. You should compare the **First Height** with the results of clicking the **Edge Size Limits** button in the **Growth** section of the **Prisms** panel. The **First Height** should usually be smaller than the minimum edge size.

**Solution:** If the **First Height** is reasonable but TGrid is unable to successfully generate your requested **Total Height**, try one of the following solutions:

- Reduce the **First Height** or the parameter used for the growth **Method** (i.e., **Slope**, **Rate**, or **Exponent**), and increase the **Number of Layers**. Extra layers allow smoothing to compensate better for sharp edges and corners.
- If you have specified a nonzero value for **Orthogonal Layers** (in the **Direct** section of the **Prisms** panel), change it to zero. Increasing orthogonality can reduce robustness.
- If you have reduced the **Max. Angle Change** (in the **Direct** section of the **Prisms** panel) from its default of 70 degrees to improve orthogonality, reset it to 70.
- If you have disabled edge swapping or normal, offset, or edge smoothing, turn them back on.

- If you are using the minimum-height offset method (enabled with the `mesh/controls/prism/offset-method` text command), switch back to the uniform method.
- Lower the **Skewness for Swapping and Smoothing** in the **Improve** section of the **Prisms** panel.
- Increase the number of **Layers** in the **Improve** section of the **Prisms** panel if it is smaller than the number of layers being grown.
- Enable **Smooth Nodes** in the **Improve** section of the **Prisms** panel. You can also increase the number of smoothing rings, using the following TUI commands:  
`mesh/controls/prism/node-smooth-rings`
- If you are growing simultaneously from multiple zones, use zone-specific growth controls. Use smaller initial heights and/or growth rates for zones that are having problems.

### Growing Too Far

The problem here is growing too many layers. Ideally, to provide for a smooth transition to the nearly-equilateral tetrahedra, the volume of the tetrahedra should be the local growth rate times the volume of the last layer prism cells.

**Solution:** To ensure this does not happen, verify that the prism parameters satisfy the above condition.

### Large Jumps in Prism Height

If the prisms use quadrilateral faces of adjacent zones, and the node spacings on these faces are very different from the layer heights, the prism heights jump at the outer edges.

**Solution:** To fix this problem, replace the adjacent quadrilateral face zones with triangular face zones, using the `boundary/create-tri-surface` text command. When you reattempt the prism generation, TGrid will project to and retriangulate the zone.

Since TGrid projects the outer nodes of the prisms to the adjacent boundary zone, and discards the original nodes on that portion of the zone, the different node spacing on the adjacent zone will not affect the prism heights at the outer edges.