OpenFOAMマイクロ講習 : snappyHexMesh超入門

2013年3月23日

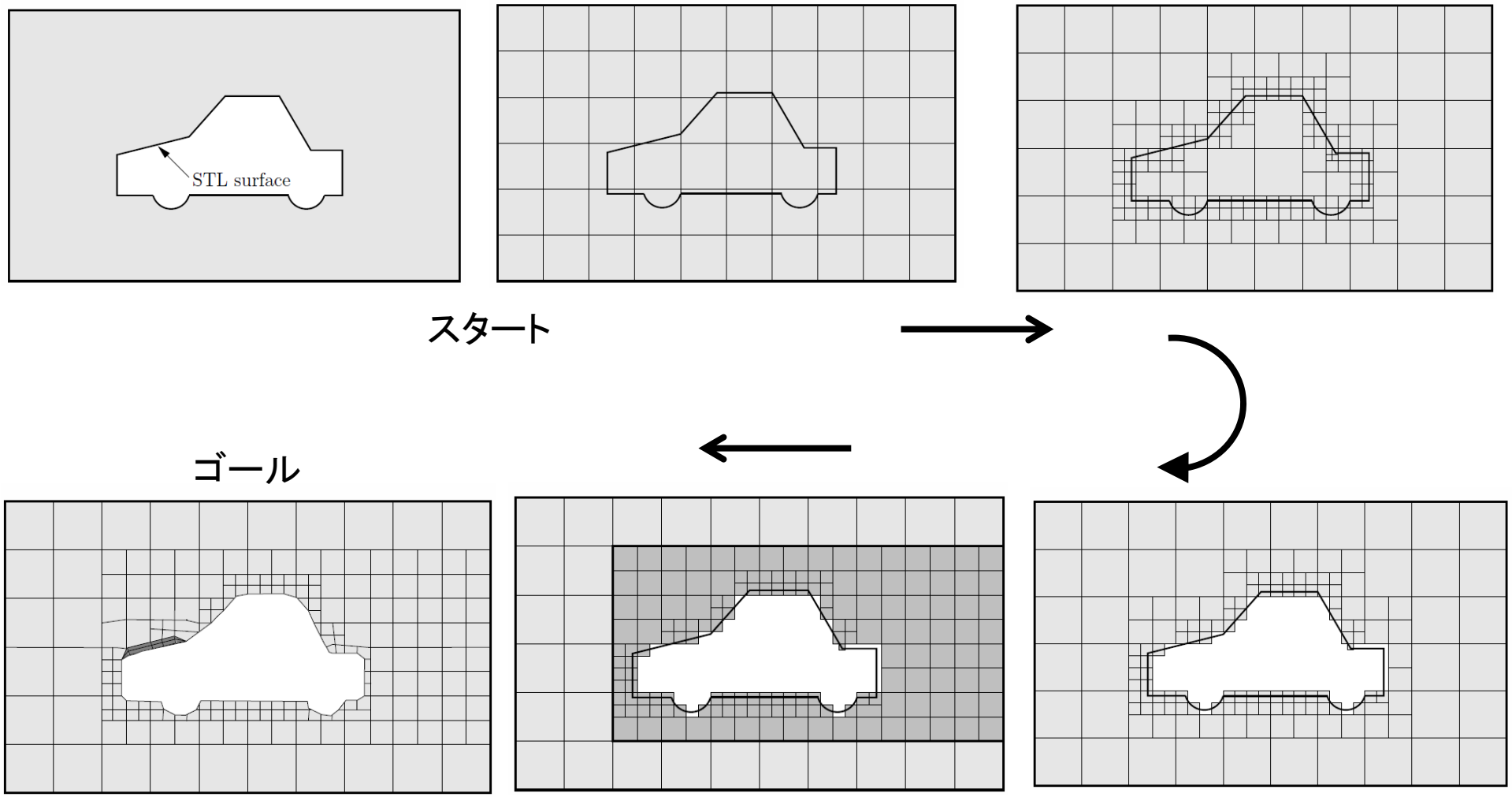
オープンCAE勉強会＠富山

中川慎二

参考文献

- OpneFOAM User Guide
  - Version 2.1.0 (15th December 2011)

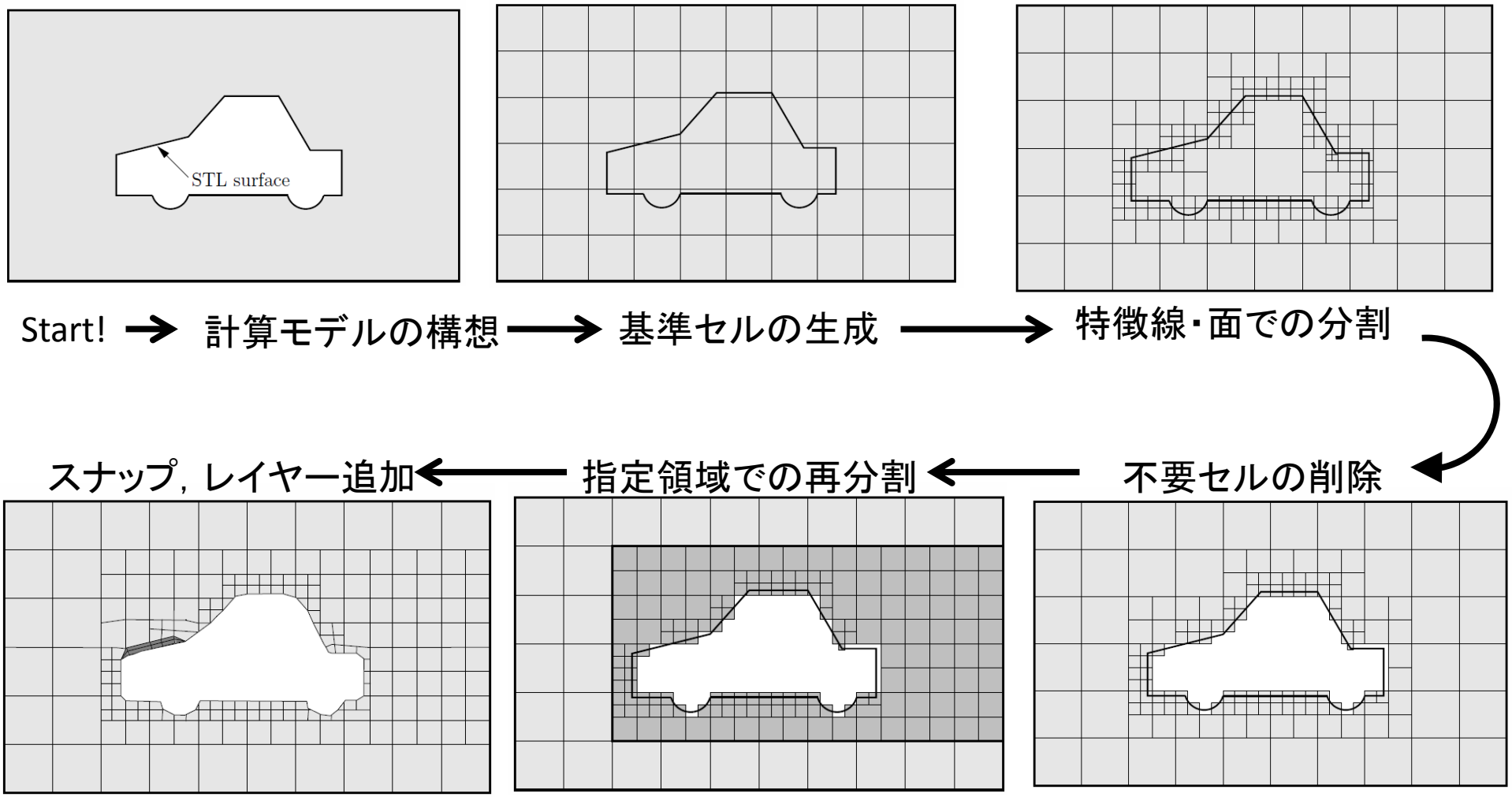
どんなセルができていくか？



作業の流れ

- 計算領域の構造を決定
  - 領域の広さ;境界面の分け方(境界条件の設定)
  - 必要なセルの大きさ(場所によって異なる？)
  - 許容されるセル数
- 基準となるセルの作成
- 特徴線・面でのセル分割
- 不要セルの削除
- セルの再分割
- 面へ沿わせたセルの変形(スナッピング)

どんなセルができていくか？



5.4.1 snappyHexMeshの実行前に

- 必要なら：STL形式の形状データを，ケースディレクトリ下のconstant/triSurfaceディレクトリに置く。
- 計算領域の大きさ および 基準となるメッシュの大きさ を決めるヘキサメッシュを作っておく
  - blockMeshユーティリティーを使うことが多い
- ケースディレクトリ下のsystemディレクトリに，snappyHexMeshDictファイルを作成し，設定を記述する。

snappyHexMeshの設定項目

Keyword	Description	Example
castellatedMesh	Create the castellated mesh?	true
snap	Do the surface snapping stage?	true
doLayers	Add surface layers?	true
mergeTolerance	Merge tolerance as fraction of bounding box of initial mesh	1e-06
debug	Controls writing of intermediate meshes and screen printing	
	— Write final mesh only	0
	— Write intermediate meshes	1
	— Write volScalarField with cellLevel for post-processing	2
	— Write current intersections as *.obj* files	4
geometry	Sub-dictionary of all surface geometry used	
castellatedMeshControls	Sub-dictionary of controls for castellated mesh	
snapControls	Sub-dictionary of controls for surface snapping	
addLayersControls	Sub-dictionary of controls for layer addition	
meshQualityControls	Sub-dictionary of controls for mesh quality	

Table 5.7: Keywords at the top level of *snappyHexMeshDict*.

OpenFOAM UserGuide U-146

例：計算領域

- 灰色部分を計算領域とし，格子を生成したい。
- できるだけ正六面体に近いセルを作りたい。（精度UP）

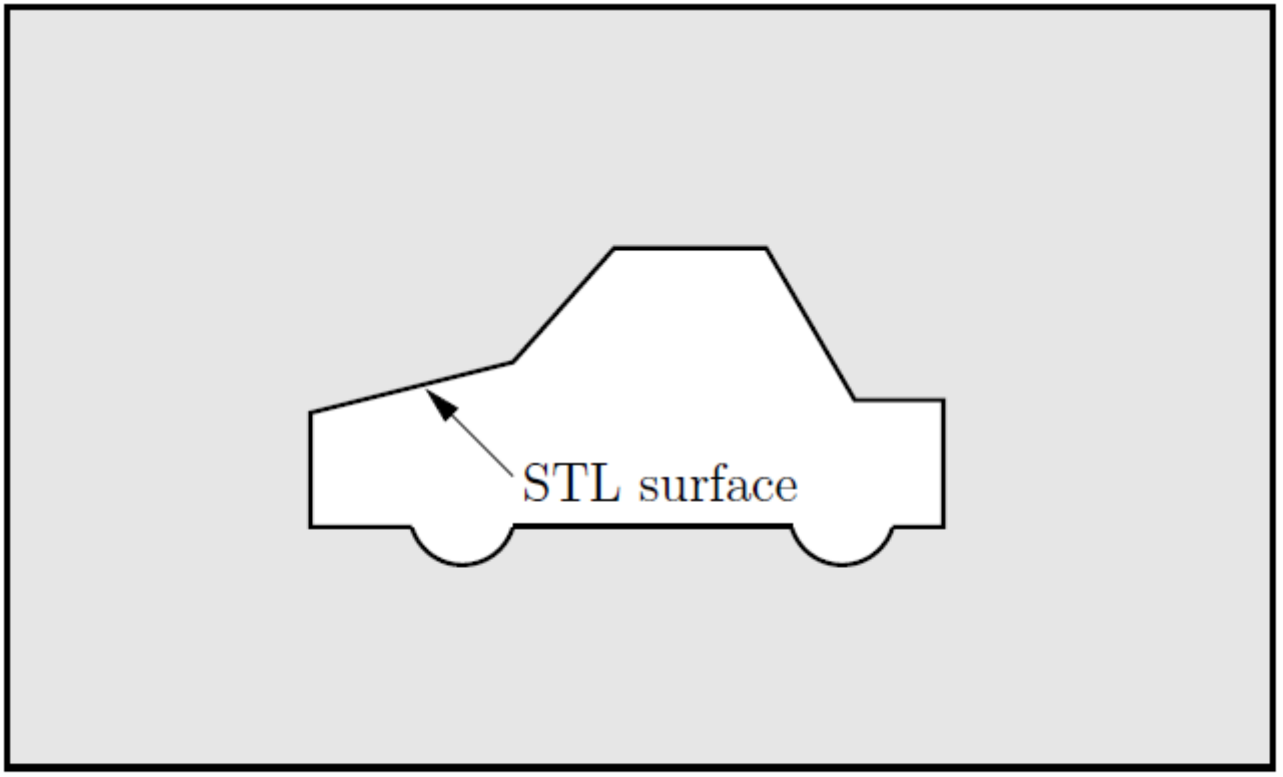


Figure 5.8: Schematic 2D meshing problem for snappyHexMesh

5.4.2 Creating the background hex mesh

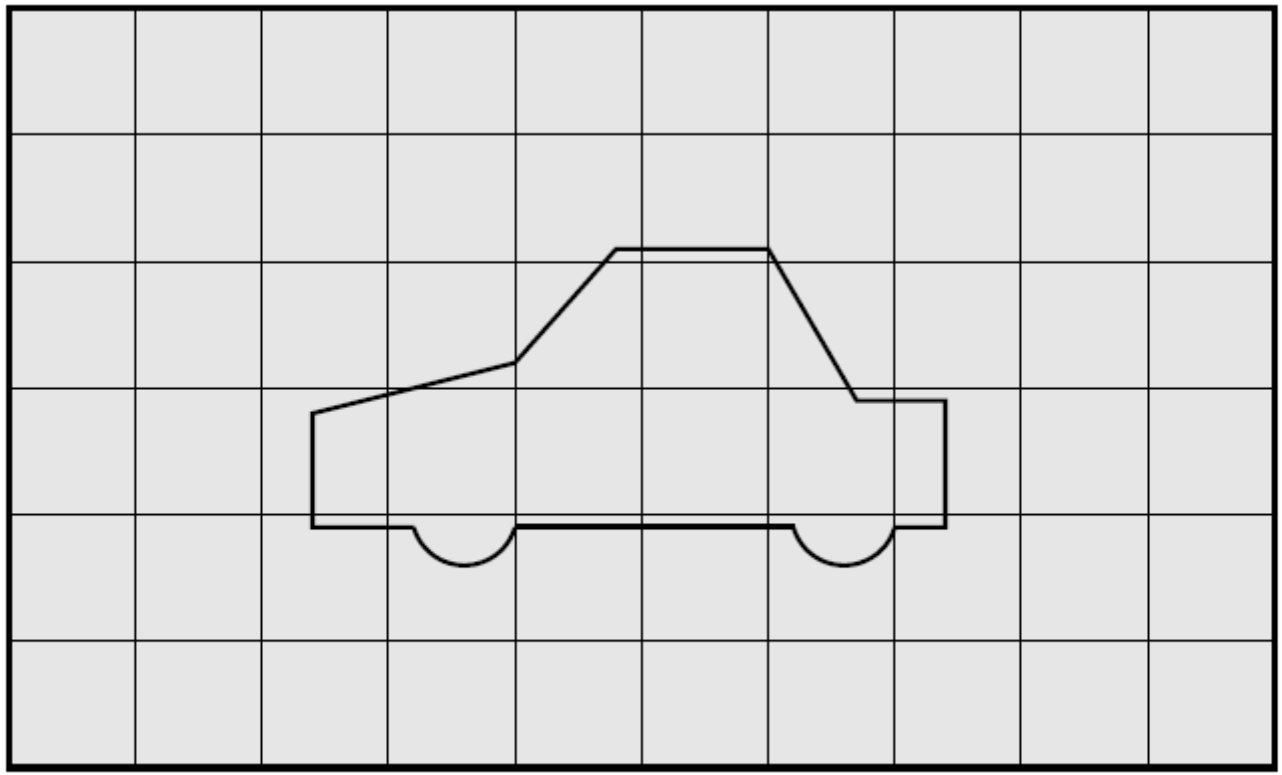


Figure 5.9: Initial mesh generation in snappyHexMesh meshing process

Creating the background hex mesh

- the mesh must consist purely of hexes;
- the cell aspect ratio should be approximately 1, at least near surfaces at which the subsequent snapping procedure is applied, otherwise the convergence of the snapping procedure is slow, possibly to the point of failure;

- there must be at least one intersection of a cell edge with the STL surface, i.e. a mesh of one cell will not work.

5.4.3 Cell splitting at feature edges and surfaces

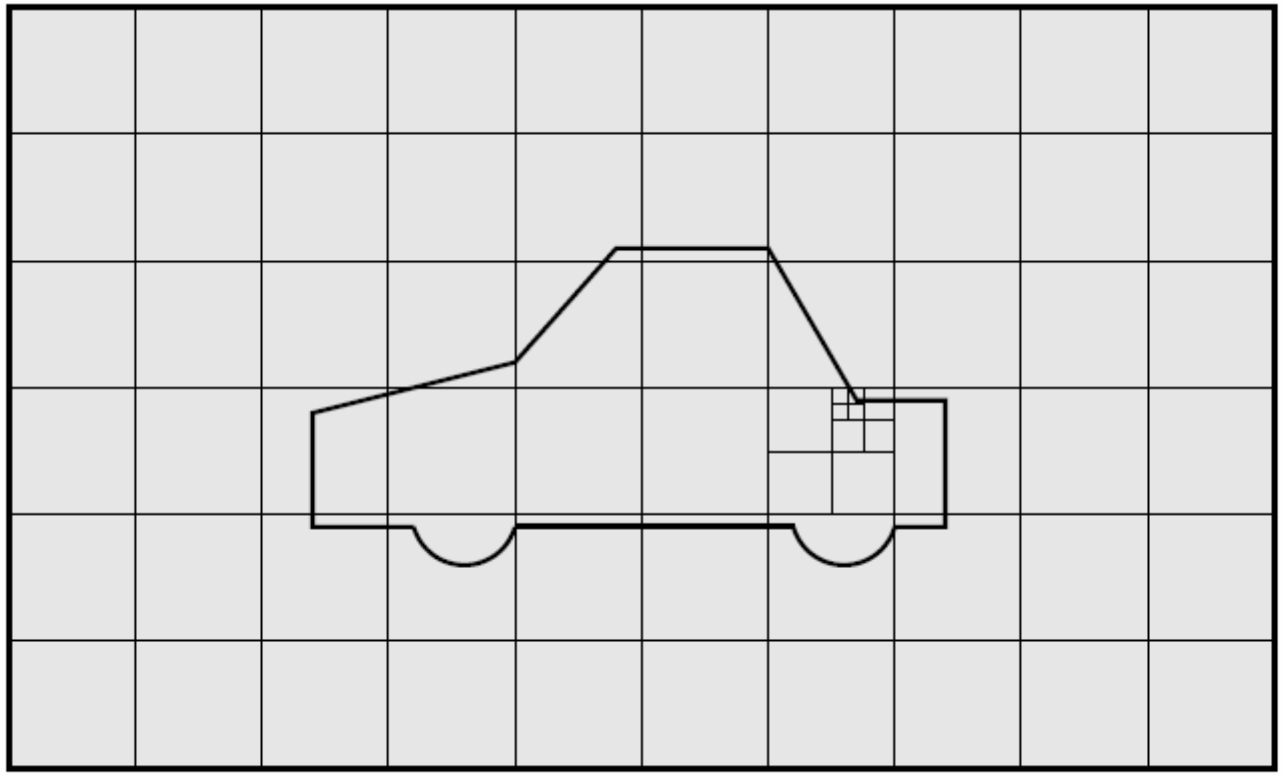


Figure 5.10: Cell splitting by feature edge in snappyHexMesh meshing process

Cell splitting at feature edges and surfaces

Keyword	Description	Example
`locationInMesh`	Location vector inside the region to be meshed *N.B.* vector must not coincide with a cell face either before or during refinement	`(5 0 0)`
`maxLocalCells`	Max number of cells per processor during refinement	`1e+06`
`maxGlobalCells`	Overall cell limit during refinement (*i.e.* before removal)	`2e+06`
`minRefinementCells`	If $\geq$ number of cells to be refined, surface refinement stops	`0`
`nCellsBetweenLevels`	Number of buffer layers of cells between different levels of refinement	`1`
`resolveFeatureAngle`	Applies maximum level of refinement to cells that can see intersections whose angle exceeds this	`30`
`features`	List of features for refinement	
`refinementSurfaces`	Dictionary of surfaces for refinement	
`refinementRegions`	Dictionary of regions for refinement	

Table 5.8: Keywords in the *castellatedMeshControls* sub-dictionary of *snappyHexMeshDict*.

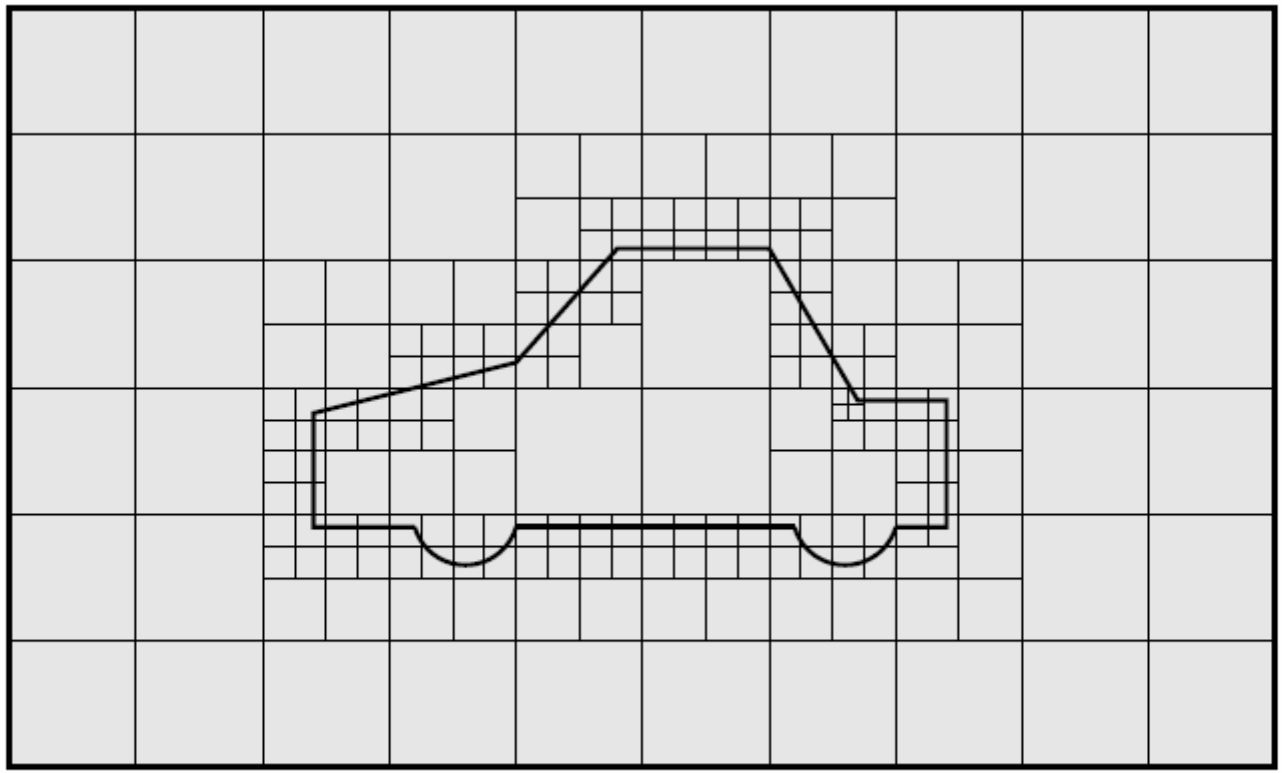


Figure 5.11: Cell splitting by surface in snappyHexMesh meshing process

file containing edge mesh

- The edgeMesh containing the features can be extracted from the STL geometry file using surfaceFeatureExtract, e.g.

```
surfaceFeatureExtract -includedAngle 150 surface.stl features
```

- このコマンドによって，surface.stl から surface.eMesh ファイルが作られる。

5.4.4 Cell removal

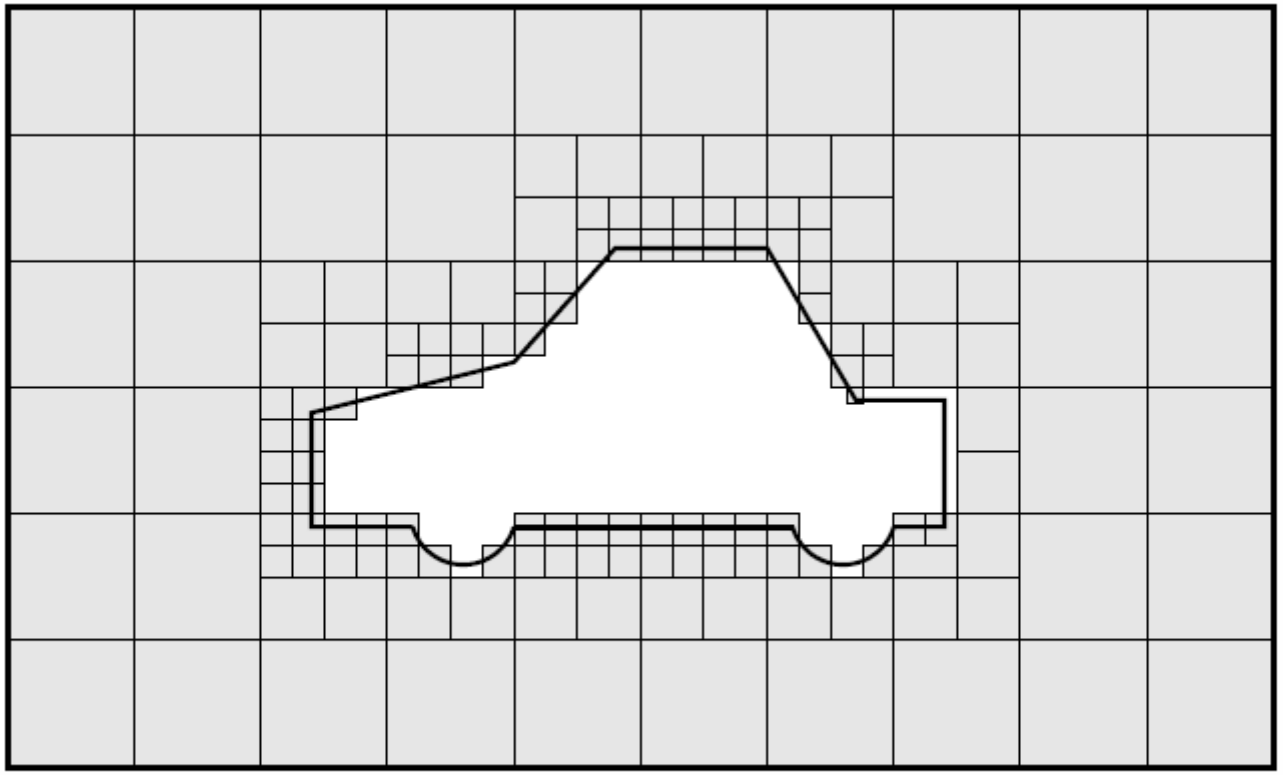


Figure 5.12: Cell removal in snappyHexMesh meshing process

- The region in which cells are retained are simply identified by a location vector within that region, specified by the **locationInMesh** keyword in castellatedMeshControls.
- Cells are retained if, approximately speaking, 50% or more of their volume lies within the region.

5.4.5 Cell splitting in specified regions

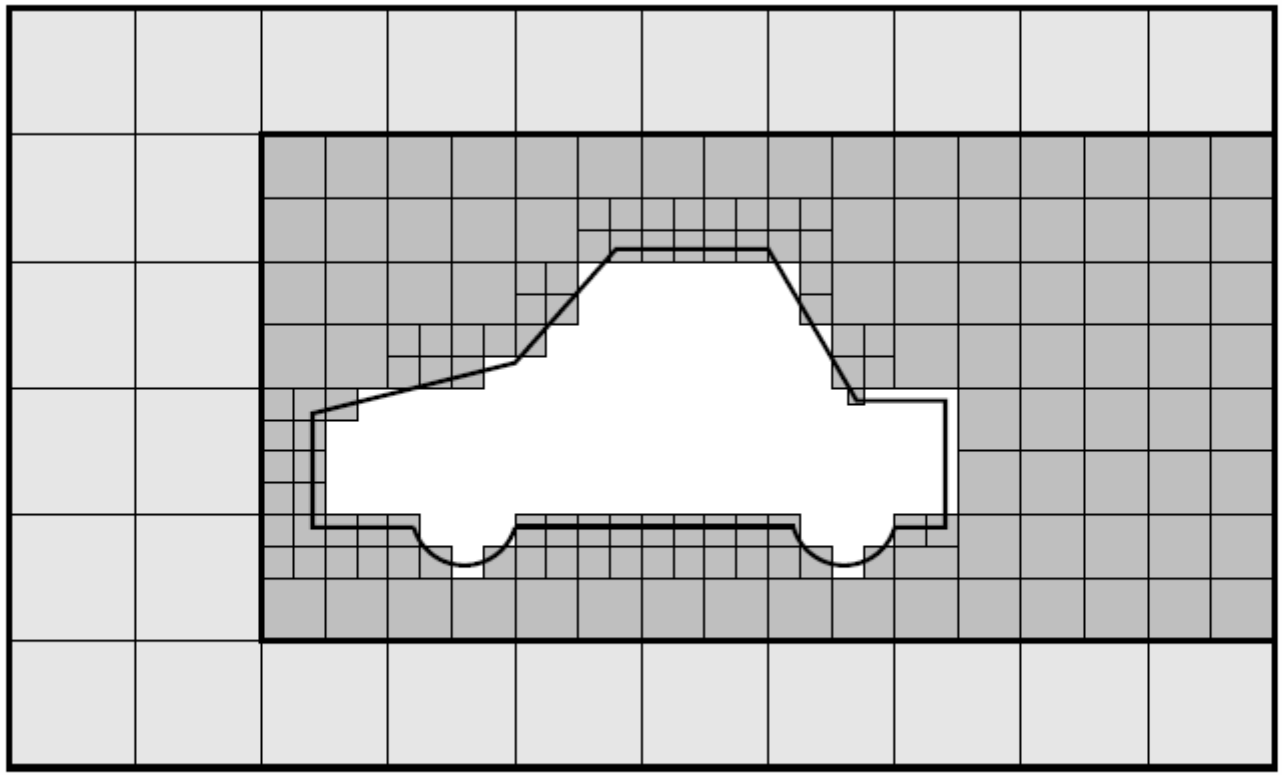


Figure 5.13: Cell splitting by region in snappyHexMesh meshing process

5.4.5 Cell splitting in specified regions

- The refinement- Regions sub-dictionary in castellatedMeshControls contains entries for refinement of the volume regions specified in the geometry sub-dictionary. A refinement mode is applied to each region which can be:
  - inside refines inside the volume region;
  - outside refines outside the volume region;
  - distance refines according to distance to the surface; and can accommodate different levels at multiple distances with the levels keyword.

5.4.6 Snapping to surfaces

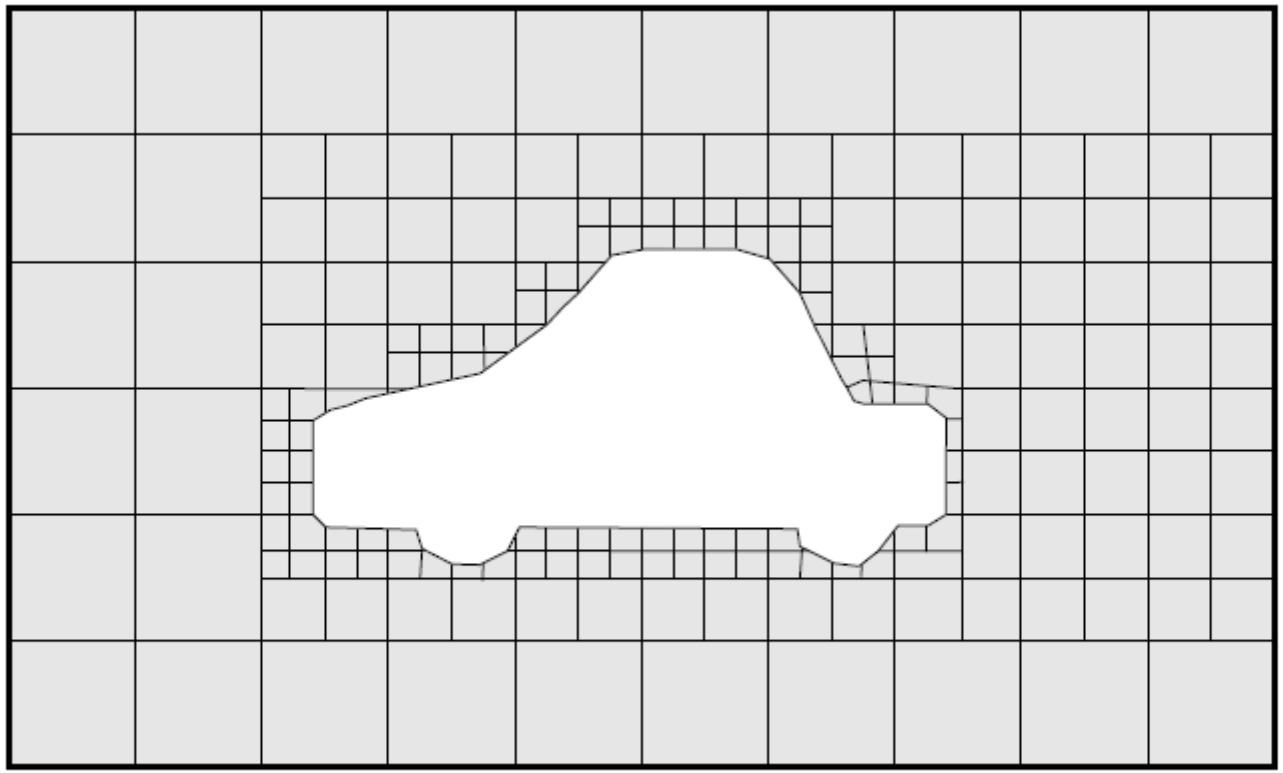


Figure 5.14: Surface snapping in snappyHexMesh meshing process

The next stage of the meshing process involves moving cell vertex points onto surface geometry to remove the jagged castellated surface from the mesh. The process is:

1. displace the vertices in the castellated boundary onto the STL surface;
2. solve for relaxation of the internal mesh with the latest displaced boundary vertices;
3. find the vertices that cause mesh quality parameters to be violated;
4. reduce the displacement of those vertices from their initial value (at 1) and repeat from 2 until mesh quality is satisfied.

5.4.6 Snapping to surfaces

Keyword	Description	Example
`nSmoothPatch`	Number of patch smoothing iterations before finding correspondence to surface	3
`tolerance`	Ratio of distance for points to be attracted by surface feature point or edge, to local maximum edge length	4.0
`nSolveIter`	Number of mesh displacement relaxation iterations	30
`nRelaxIter`	Maximum number of snapping relaxation iterations	5

Table 5.9: Keywords in the *snapControls* dictionary of *snappyHexMeshDict*.

5.4.7 Mesh layers

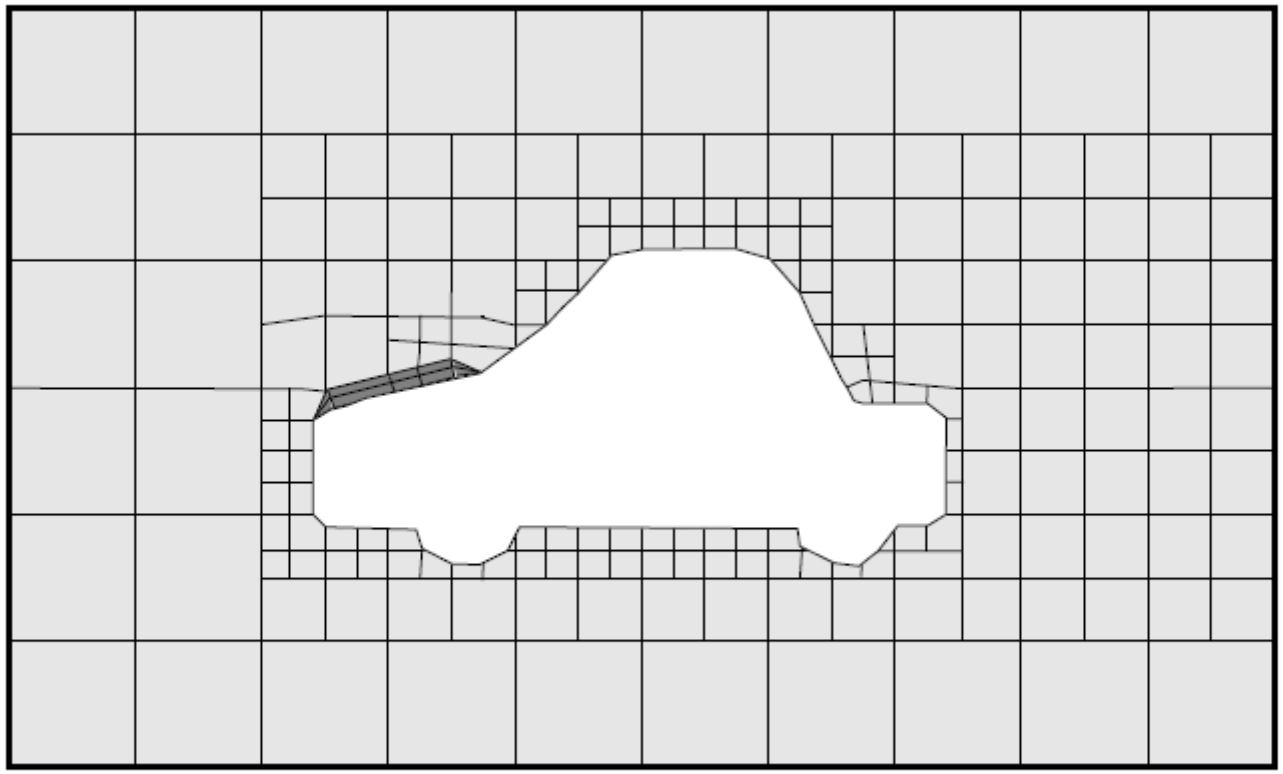


Figure 5.15: Layer addition in snappyHexMesh meshing process

The process of mesh layer addition involves shrinking the existing mesh from the boundary and inserting layers of cells, broadly as follows:

1. the mesh is projected back from the surface by a specified thickness in the direction normal to the surface;
2. solve for relaxation of the internal mesh with the latest projected boundary vertices;
3. check if validation criteria are satisfied otherwise reduce the projected thickness and return to 2; if validation cannot be satisfied for any thickness, do not insert layers;
4. if the validation criteria can be satisfied, insert mesh layers;
5. the mesh is checked again; if the checks fail, layers are removed and we return to 2.

Keyword	Description	Example
`layers`	Dictionary of layers	
`relativeSizes`	Are layer thicknesses relative to undistorted cell size outside layer or absolute?	`true/false`
`expansionRatio`	Expansion factor for layer mesh	`1.0`
`finalLayerRatio`	Thickness of layer furthest from the wall, either relative or absolute according to the `relativeSizes` entry	`0.3`
`minThickness`	Minimum thickness of cell layer, either relative or absolute (as above)	`0.25`
`nGrow`	Number of layers of connected faces that are not grown if points get not extruded; helps convergence of layer addition close to features	`1`
`featureAngle`	Angle above which surface is not extruded	`60`
`nRelaxIter`	Maximum number of snapping relaxation iterations	`5`
`nSmoothSurfaceNormals`	Number of smoothing iterations of surface normals	`1`
`nSmoothNormals`	Number of smoothing iterations of interior mesh movement direction	`3`
`nSmoothThickness`	Smooth layer thickness over surface patches	`10`
`maxFaceThicknessRatio`	Stop layer growth on highly warped cells	`0.5`
`maxThicknessToMedialRatio`	Reduce layer growth where ratio thickness to medial distance is large	`0.3`
`minMedianAxisAngle`	Angle used to pick up medial axis points	`130`
`nBufferCellsNoExtrude`	Create buffer region for new layer terminations	`0`
`nLayerIter`	Overall max number of layer addition iterations	`50`
`nRelaxedIter`	Max number of iterations after which the controls in the *relaxed* sub dictionary of `meshQuality` are used	`20`

Table 5.10: Keywords in the *addLayersControls* sub dictionary of *snappyHexMeshDict*.

5.4.8 Mesh quality controls

Keyword	Description	Example
`maxNonOrtho`	Maximum non-orthogonality allowed; `180` disables	`65`
`maxBoundarySkewness`	Max boundary face skewness allowed; `<0` disables	`20`
`maxInternalSkewness`	Max internal face skewness allowed; `<0` disables	`4`
`maxConcave`	Max concaveness allowed; `180` disables	`80`
`minFlatness`	Ratio of minimum projected area to actual area; `-1` disables	`0.5`
`minVol`	Minimum pyramid volume; large negative number, *e.g.*`-1e30` disables	`1e-13`
`minArea`	Minimum face area; `<0` disables	`-1`
`minTwist`	Minimum face twist; `<-1` disables	`0.05`
`minDeterminant`	Minimum normalised cell determinant; `1` = hex; $\leq$ 0 illegal cell	`0.001`
`minFaceWeight`	`0`→`0.5`	`0.05`
`minVolRatio`	`0`→`1.0`	`0.01`
`minTriangleTwist`	`>0` for Fluent compatability	`-1`
`nSmoothScale`	Number of error distribution iterations	`4`
`errorReduction`	Amount to scale back displacement at error points	`0.75`
`relaxed`	Sub-dictionary that can include modified values for the above keyword entries to be used when `nRelaxedIter` is exceeded in the layer addition process	`relaxed { ... }`

Table 5.11: Keywords in the *meshQualityControls* sub-dictionary of *snappyHexMeshDict*.