ただで始める流体解析

初心者向け
自動車の外部流れ計算
（その1）

本日の流れ

1. はじめに
2. 公開されているモデル
3. 計算モデル作成
4. 計算設定
5. 計算実施
6. まとめ

1．はじめに

OpenFOAMを使い始めた方，または何とかものにしたい方へ（偉そうに書いていますが，流体解析については初心者です。）

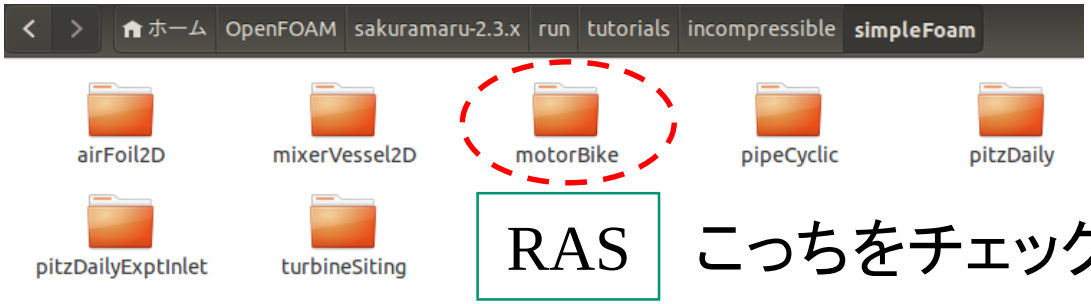
今回紹介する事例は，OpenFOAMを始めるに当たりちょうど良いモデルです。OpenFOAMのチュートリアルにあるmotorBikeの自動車版といった感じです。
後のスライドで示しますが，公開されてる自動車のモデルをダウンロードして計算をやってみましょう。

最も簡単は方法は，DEXCS for OpenFOAMを使えば全ての計算環境が全て用意されています。商用の流体解析ソフトのように何百万，何千万もしません。全て無料で使えます。

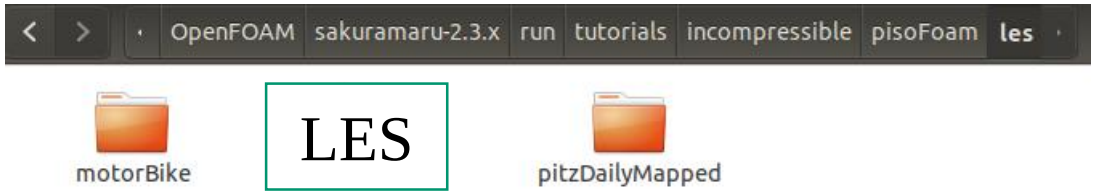
自分の経験ですが，基本的な流れ計算などは商用ソフトを使わなくても，OpenFOAMで計算ができ，精度も十分です。

今回の計算は，OpenFOAM 2.3.xで行っています。

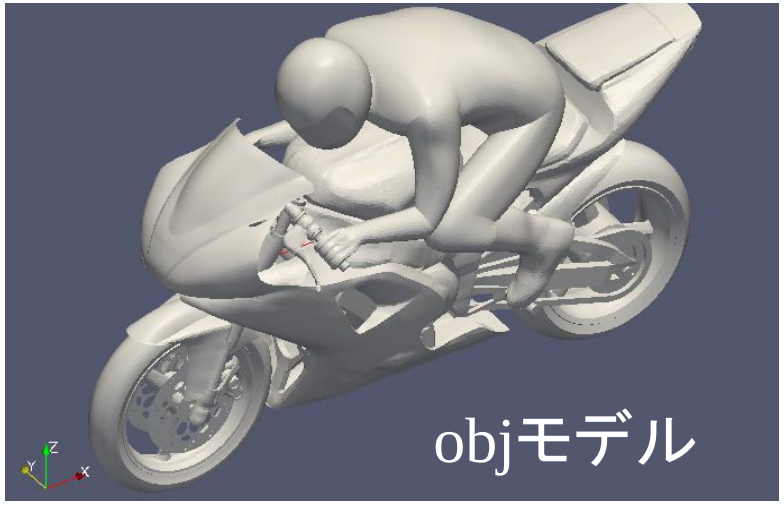
モーターバイクのモデルとは？ ちょっと見てみる。

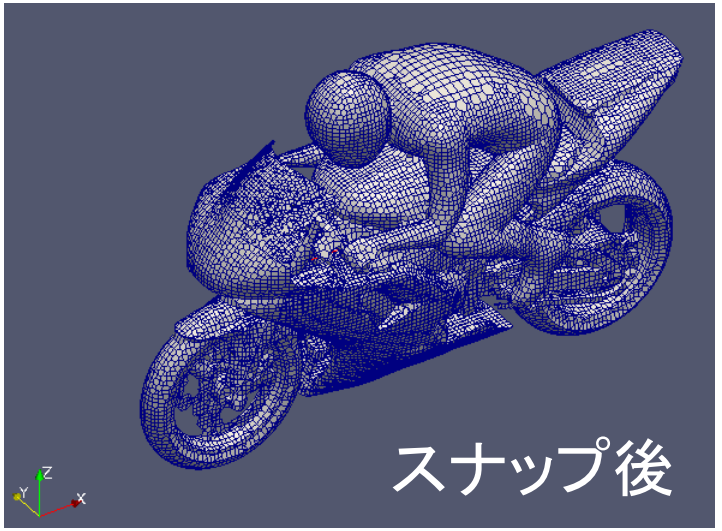


RAS　こっちをチェック



LES





Allrun ×

```
#!/bin/sh
cd ${0%/*} || exit 1    # run from this directory

# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions

# copy motorbike surface from resources directory
cp $FOAM_TUTORIALS/resources/geometry/motorBike.obj.gz constant/triSurface/
runApplication surfaceFeatureExtract

runApplication blockMesh

runApplication decomposePar
runParallel snappyHexMesh 4 -overwrite

#- For non-parallel running
#cp -r 0.org 0 > /dev/null 2>&1

#- For parallel running
ls -d processor* | xargs -I {} rm -rf ./{}/0
ls -d processor* | xargs -I {} cp -r 0.org ./{}/0

runParallel patchSummary 4
runParallel potentialFoam 4
runParallel $(getApplication) 4

runApplication reconstructParMesh -constant
runApplication reconstructPar -latestTime

# ----------------------------------------------------------------- end-of-file
```

チュートリアルは6並列だが自分のノートPCは4コアのためコア数を修正

Allrunでメッシュ，計算までできる

```
sakuramaru@SAKURA-MARU:~/Desktop/ver_2.3.x/motorBike/bik2$ ./Allrun
Running surfaceFeatureExtract on /home/sakuramaru/Desktop/ver_2.3.x/motorBike/bik2
Running blockMesh on /home/sakuramaru/Desktop/ver_2.3.x/motorBike/bik2
Running decomposePar on /home/sakuramaru/Desktop/ver_2.3.x/motorBike/bik2
Running snappyHexMesh in parallel on /home/sakuramaru/Desktop/ver_2.3.x/motorBike/bik2 using 4 processes
Running patchSummary in parallel on /home/sakuramaru/Desktop/ver_2.3.x/motorBike/bik2 using 4 processes
Running potentialFoam in parallel on /home/sakuramaru/Desktop/ver_2.3.x/motorBike/bik2 using 4 processes
Running simpleFoam in parallel on /home/sakuramaru/Desktop/ver_2.3.x/motorBike/bik2 using 4 processes
```

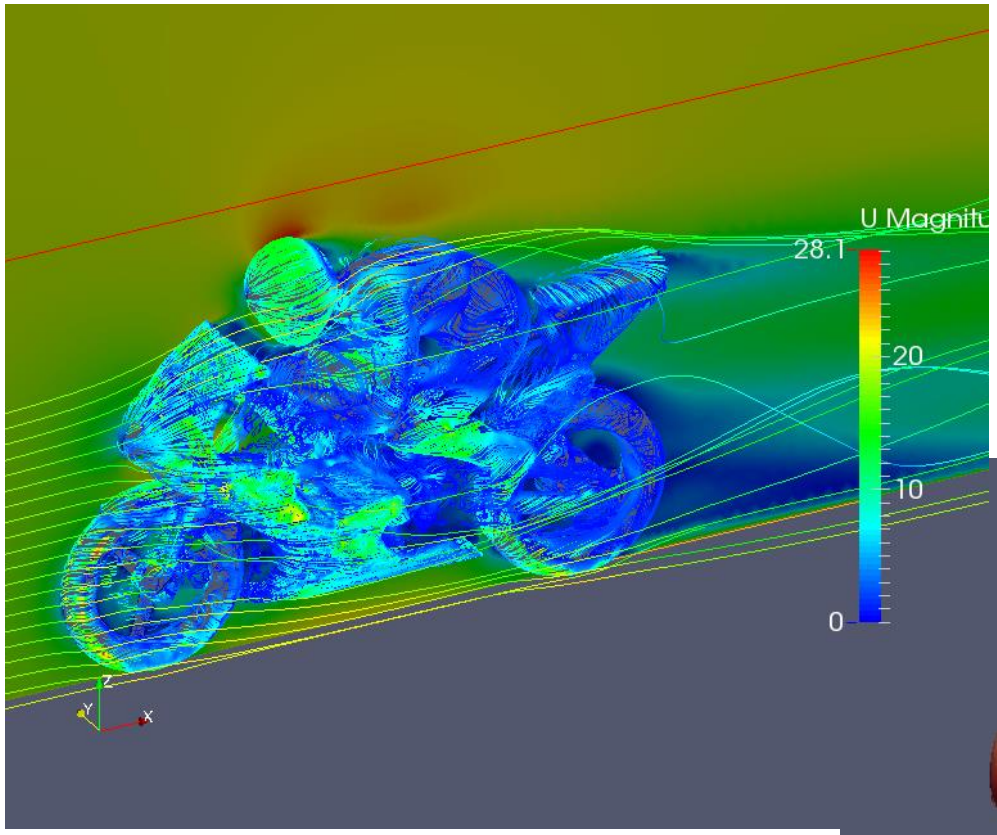
checkMesh –allGeometry -allTopology

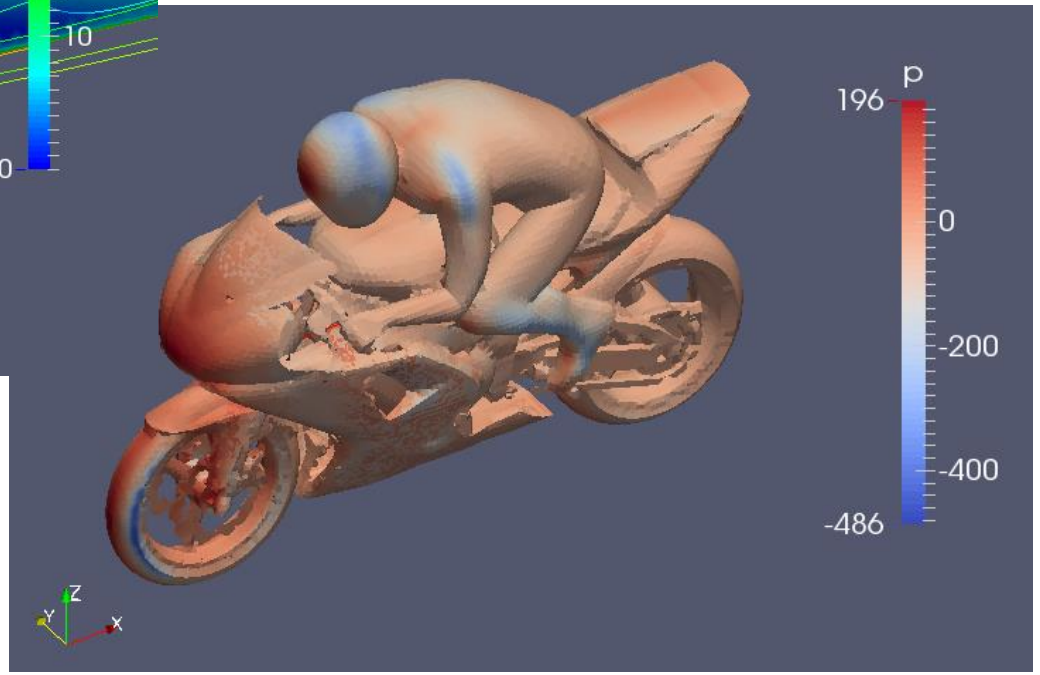
```
Time = 3

Mesh stats
    points:           404840
    faces:            1103978
    internal faces:   1053930
    cells:            352260
    faces per cell:   6.1259
    boundary patches: 72
    point zones:      0
    face zones:       0
    cell zones:       0

Overall number of cells of each type:
    hexahedra:     307191
    prisms:        7225
    wedges:        950
    pyramids:      2
    tet wedges:    1701
    tetrahedra:    59
    polyhedra:     35132
    Breakdown of polyhedra by number o
        faces   number of cells
            4   3936
            5   6569
            6   5457
            7   3001
            8   1288
            9   10599
           10   17
           11   6
           12   2355
           13   2
           14   1
           15   1635
           17   1
           18   263
           21   2

Checking topology...
    Boundary definition OK.
    Cell to face addressing OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Topological cell zip-up check OK.
  <<Number of duplicate (not baffle) faces found: 5. This might indicate a problem.
  <<Number of faces with non-consecutive shared points: 30. This might indicate a problem.
  <<Writing 62 faces with non-standard edge connectivity to set edgeFaces
  <<Writing 12 cells with two non-boundary faces to set twoInternalFacesCells
    Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...
                                  Patch    Faces   Points                  Surface topology Bounding box
                           frontAndBack      360      420  ok (non-closed singly connected) (-5 -4 0) (15
                                  inlet       72       90  ok (non-closed singly connected) (-5 -4 0) (-5
                                 outlet       72       90  ok (non-closed singly connected) (15 -4 0) (15
                              lowerWall     5349     5676  ok (non-closed singly connected) (-5 -4 0) (15
                              upperWall      160      189  ok (non-closed singly connected) (-5 -4 8) (15
                motorBike_frt-fairing:001%1     5320     9897  multiply connected (shared edge) (0.0273
                motorBike_windshield:002%2       51       84  ok (non-closed singly connected) (-0.0268
                motorBike_rr-wh-rim:005%5      123      220  ok (non-closed singly connected) (1.31399
                motorBike_rr-wh-rim:010%10      355      677  ok (non-closed singly connected) (1.20089
                motorBike_fr-wh-rim:011%11      498      825  ok (non-closed singly connected) (-0.2175
                motorBike_fr-wh-brake-disk:012%12       46       87  ok (non-closed singly connected) (
                motorBike_frame:016-shadow%13      104      228  ok (non-closed singly connected) (0.48
                motorBike_rear-susp:014%14      809     1467  ok (non-closed singly connected) (0.81262
                motorBike_rear-susp:014-shadow%15      474     1216  ok (non-closed singly connected) (
                motorBike_frame:016%16       62      157  ok (non-closed singly connected) (0.494347 -0
                motorBike_rr-wh-rim:005-shadow%17       67      174  ok (non-closed singly connected) (
                motorBike_rr-wh-chain-hub:022%22      128      218  ok (non-closed singly connected) (1
                motorBike_rearseat%24      421      513  ok (non-closed singly connected) (1.31991 -0.1
                motorBike_frt-fairing%25      616      851  ok (non-closed singly connected) (-0.081680
                motorBike_windshield%26      377      803  ok (non-closed singly connected) (0.0421841
                motorBike_headlights%27      155      255  ok (non-closed singly connected) (-0.0760509
                motorBike_driversseat%28      357      543  ok (non-closed singly connected) (0.794974
                motorBike_rear-body%29     2039     2517  ok (non-closed singly connected) (0.967132 -0
                motorBike_fuel-tank%30      880     1213  ok (non-closed singly connected) (0.398279 -0
```

メッシュの品質はこんな程度

```
Checking geometry...
    Overall domain bounding box (-5 -4 0) (15 4 8)
    Mesh (non-empty, non-wedge) directions (1 1 1)
    Mesh (non-empty) directions (1 1 1)
    Boundary openness (-2.06714e-17 -6.75991e-17 -1.2197e-15) OK.
    Max cell openness = 8.23487e-16 OK.
    Max aspect ratio = 40.6183 OK.
    Minimum face area = 1.03191e-06. Maximum face area = 1.01336.  Face area magnitudes OK.
    Min volume = 9.7096e-09. Max volume = 1.00461.  Total volume = 1279.67.  Cell volumes OK.
    Mesh non-orthogonality Max: 64.992 average: 9.91287
    Non-orthogonality check OK.
    Face pyramids OK.
***Max skewness = 9.48848, 15 highly skew faces detected which may impair the quality of the results
 <<Writing 15 skew faces to set skewFaces
    Coupled point location match (average 0) OK.
    Face tets OK.
    Min/max edge length = 0.000341636 1.0157 OK.
  *There are 2315 faces with concave angles between consecutive edges. Max concave angle = 79.9697 degrees.
 <<Writing 2315 faces with concave angles to set concaveFaces
    Face flatness (1 = flat, 0 = butterfly) : min = 0.319843  average = 0.998468
  *There are 188 faces with ratio between projected and actual area < 0.8
    Minimum ratio (minimum flatness, maximum warpage) = 0.319843
 <<Writing 188 warped faces to set warpedFaces
    Cell determinant (wellposedness) : minimum: 0 average: 12.4277
***Cells with small determinant (< 0.001) found, number of cells: 69
 <<Writing 69 under-determined cells to set underdeterminedCells
***Concave cells (using face planes) found, number of cells: 15106
 <<Writing 15106 concave cells to set concaveCells
    Face interpolation weight : minimum: 0.0212596 average: 0.465806
***Faces with small interpolation weight (< 0.05) found, number of faces: 1053
 <<Writing 1053 faces with low interpolation weights to set lowWeightFaces
    Face volume ratio : minimum: 0.0100285 average: 0.846319
    Face volume ratio check OK.

Failed 4 mesh checks.

End
```

こんな感じの結果となる





マシン環境

ノートPC　マウスコンピュータ

本体のメモリは32GB，VirtualBox上で動かしています

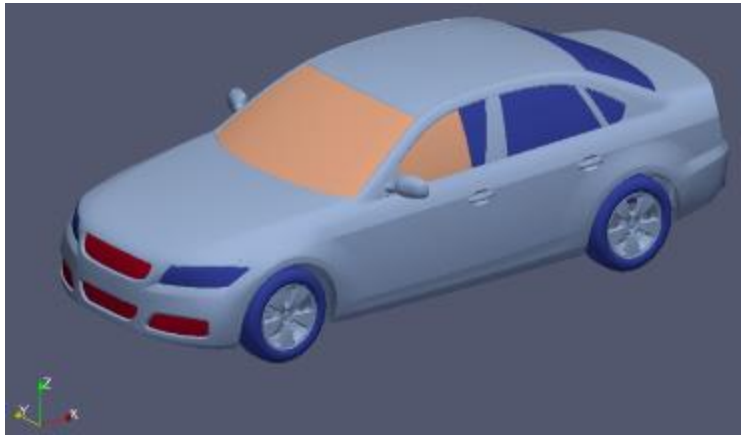
（ノートPCで計算する場合，熱対策が必要な場合があります）

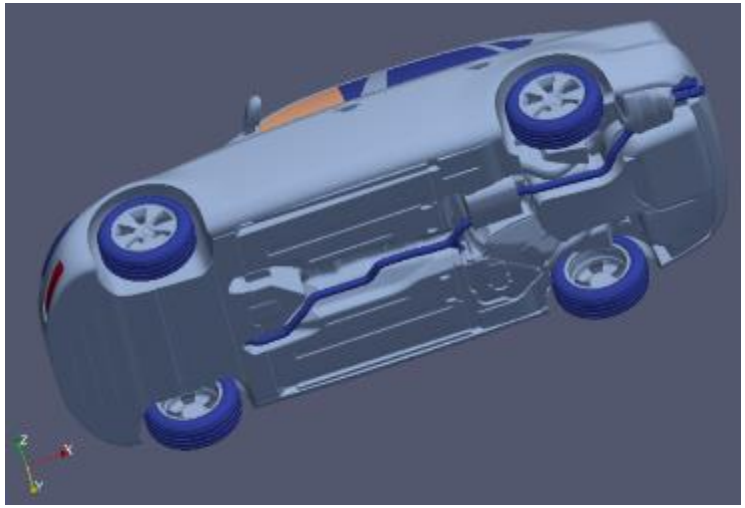
メモリは必要に応じて変えています

2. 公開されているモデル

Automotive Aerodynamicsとして下記のサイトで自動車モデルが公開されています。(ちなみに自分は自動車関連の会社とは全く縁がございません。勉強・趣味レーション？です。)

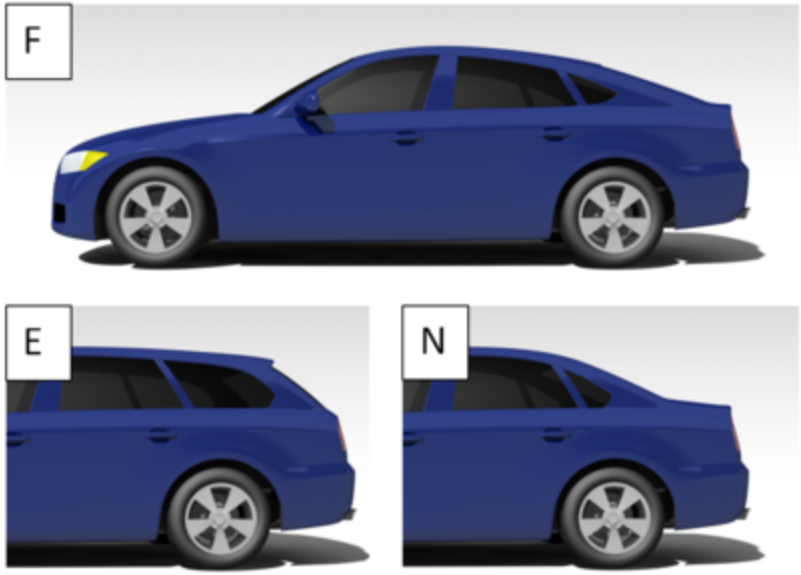
http://www.aer.mw.tum.de/en/research-groups/automotive/





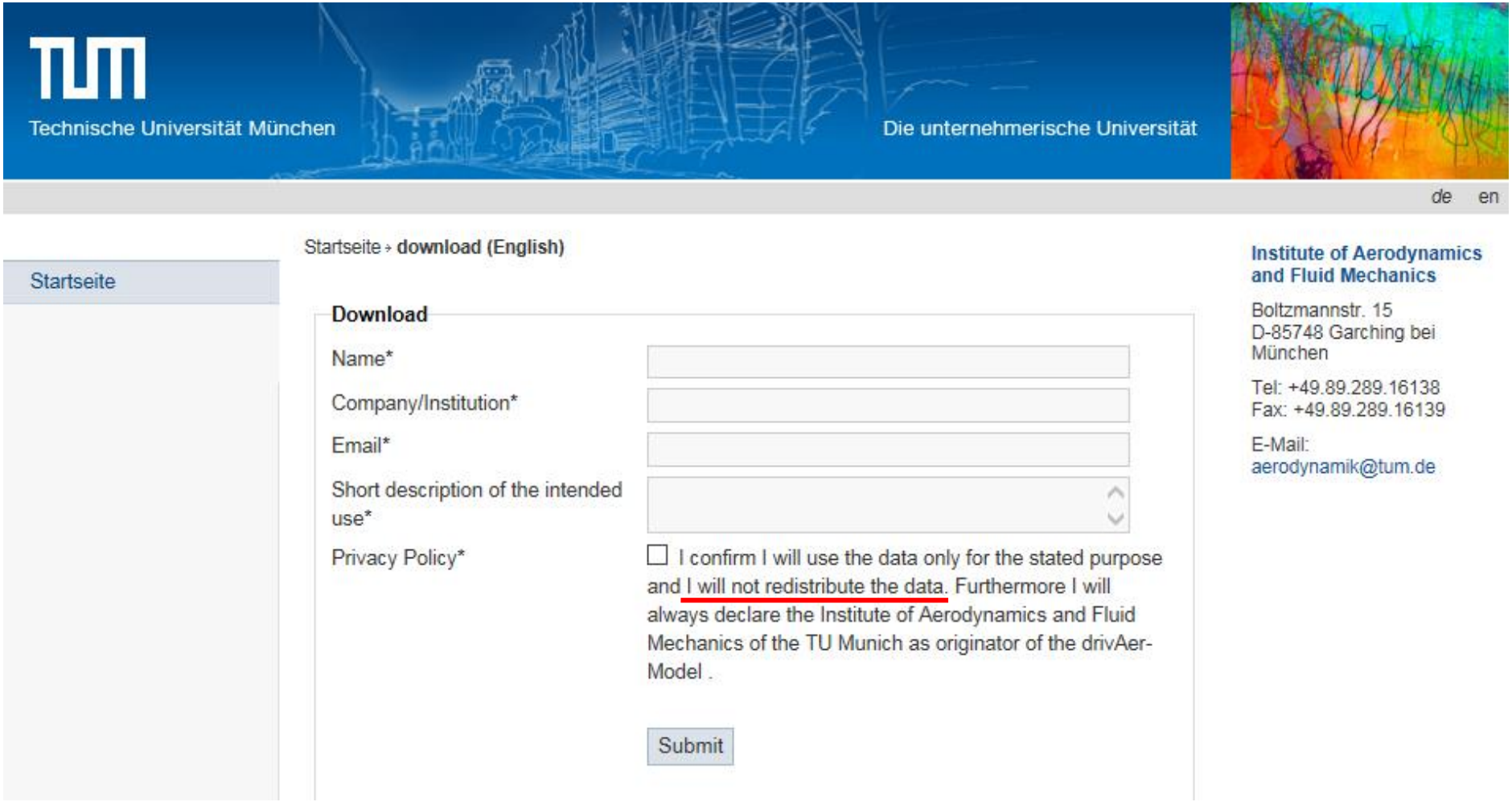
DrivAer Model

A lot of the investigations in automotive aerodynamics are still based on strongly simplified generic bodies such as the Ahmed Body or the SAE body.
A different approach is the direct investigation of production vehicles.
Especially transient investigations often use generic car bodies. This is due to the fact that an unsteady investigation of an actual production car has so far not been feasible without enormous computational and experimental effort.
Additionally simple car models can help to understand the fundamental flow phenomena and to gain basic insights.
However, as their shapes are too different from actual car geometries these results will not be fully transferable to the development of production vehicles. This is especially true where complex body surfaces are involved, such as the A-pillars, the highly curved rear end or the wheelhouse region.



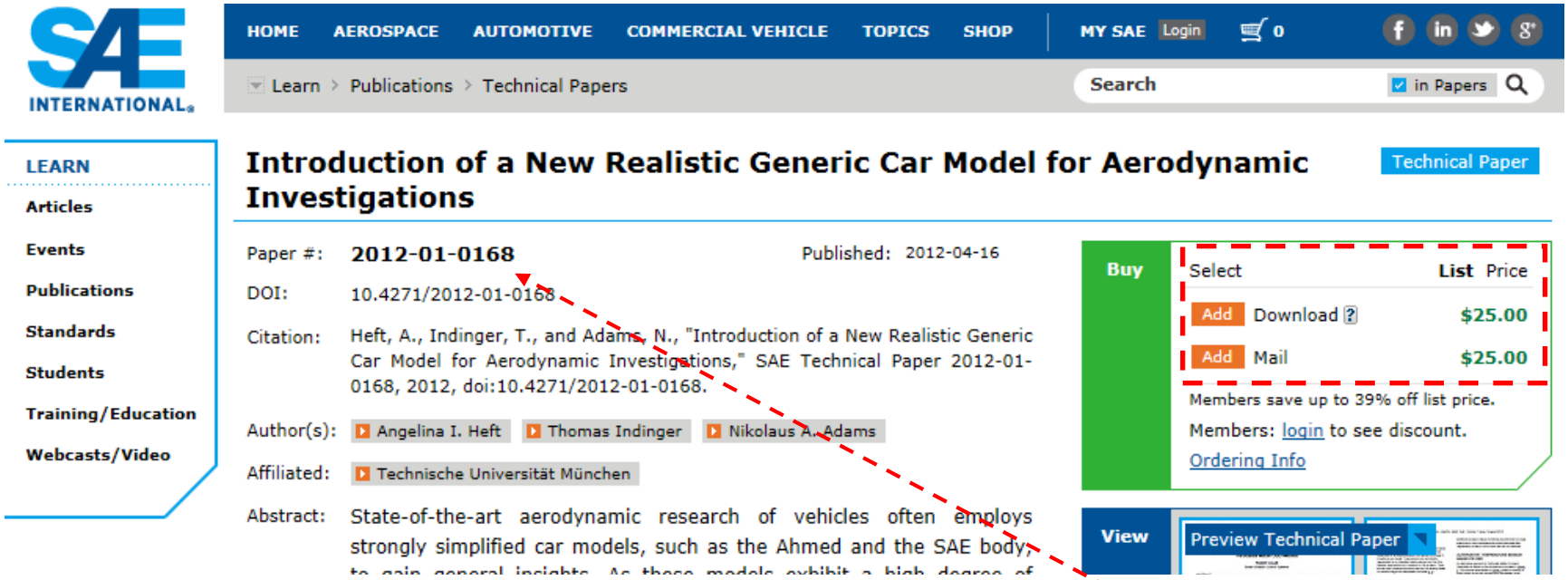
DrivAer body with different tops.

モデルの入手は下記の登録で行います。step，iges，stl形式のデータがダウンロードできます。データの再配布は禁止です。



関係する論文があります。Introduction of a New Realistic Generic Car Model for Aerodynamic Investigations

SAE（米自動車技術会）では有料，delphiでは無料。同じ内容です。



http://www.nav200.delphi.com/pdf/techpapers/2012-01-0168.pdf

この情報は2015.5.5時点です。

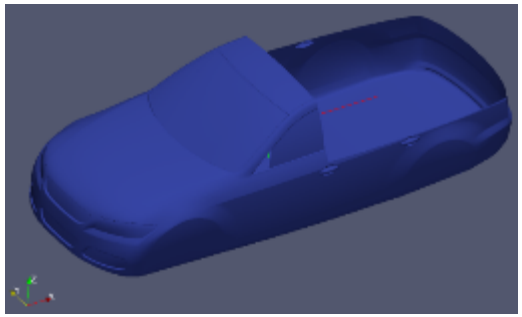
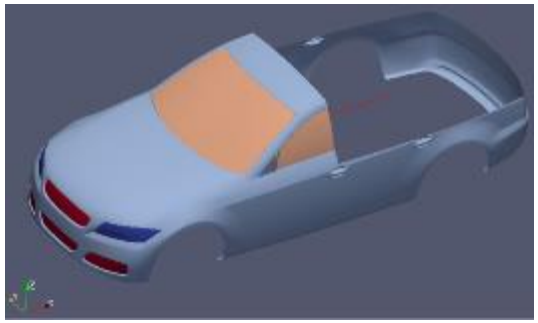
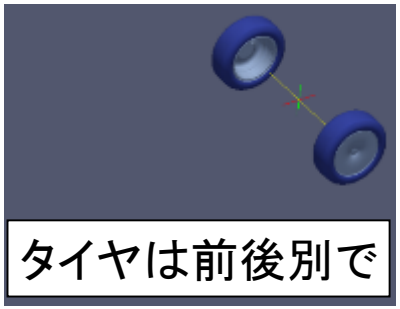
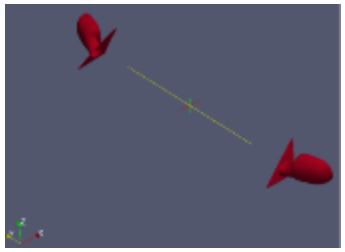
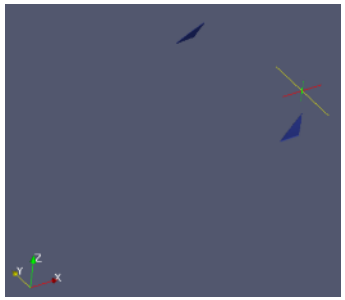
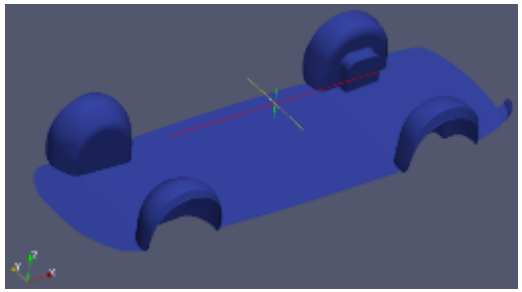
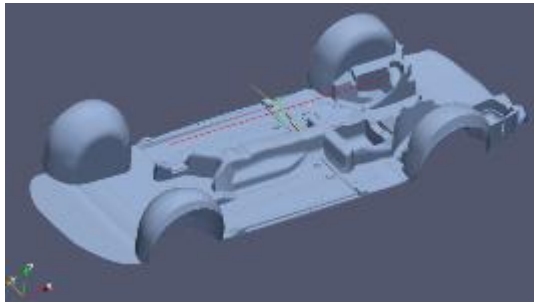
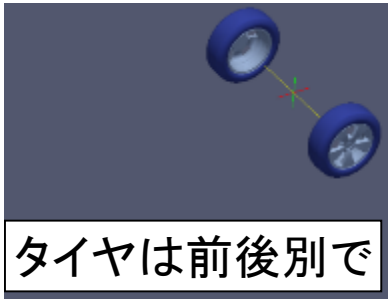
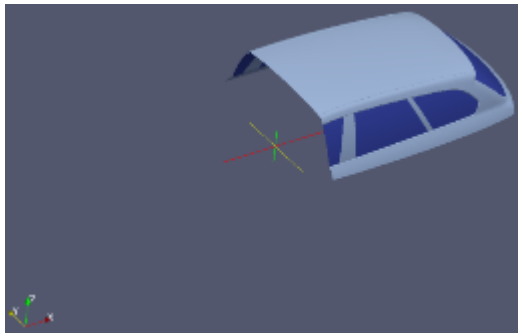
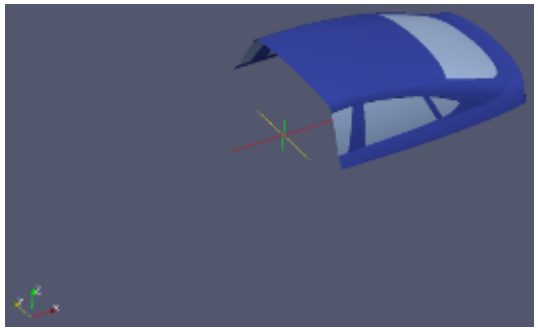
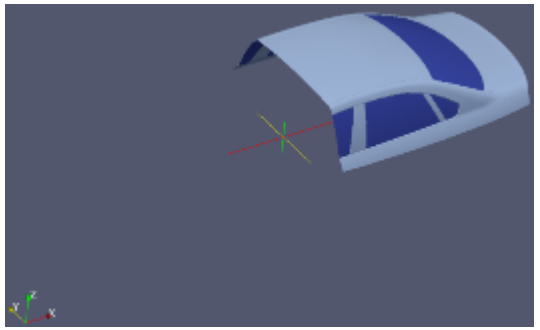
公開されているパーツ類　15モデルとなっている（stepの場合）

01_Body.stp	2012/01/13 14:07	STP ファイル	35,312 KB
01_Body_Closed.stp	2013/05/24 16:29	STP ファイル	32,511 KB
02_Underbody_Detailed.stp	2012/05/14 8:21	STP ファイル	50,371 KB
02_Underbody_Smooth.stp	2012/05/14 8:26	STP ファイル	8,948 KB
03_RearEnd_EstateBack.stp	2012/10/12 12:03	STP ファイル	9,212 KB
03_RearEnd_Fastback.stp	2012/01/13 14:07	STP ファイル	8,290 KB
03_RearEnd_Notchback.stp	2012/01/13 14:07	STP ファイル	10,609 KB
04_ExhaustSystem.stp	2012/01/13 14:07	STP ファイル	5,905 KB
05_Wheels_Front.stp	2012/01/13 14:07	STP ファイル	5,643 KB
05_Wheels_Front_Closed.step	2013/10/14 13:40	STEP ファイル	5,404 KB
05_Wheels_Front_Smooth.step	2013/10/14 14:04	STEP ファイル	8,685 KB
06_Wheels_Rear.stp	2012/01/13 14:07	STP ファイル	5,622 KB
06_Wheels_Rear_Closed.step	2013/10/14 13:38	STEP ファイル	5,411 KB
06_Wheels_Rear_Smooth.step	2013/10/14 14:16	STEP ファイル	8,486 KB
07_Mirrors.stp	2012/01/13 14:07	STP ファイル	6,497 KB

公開されているパーツ類　15モデルとなっている（stlの場合）

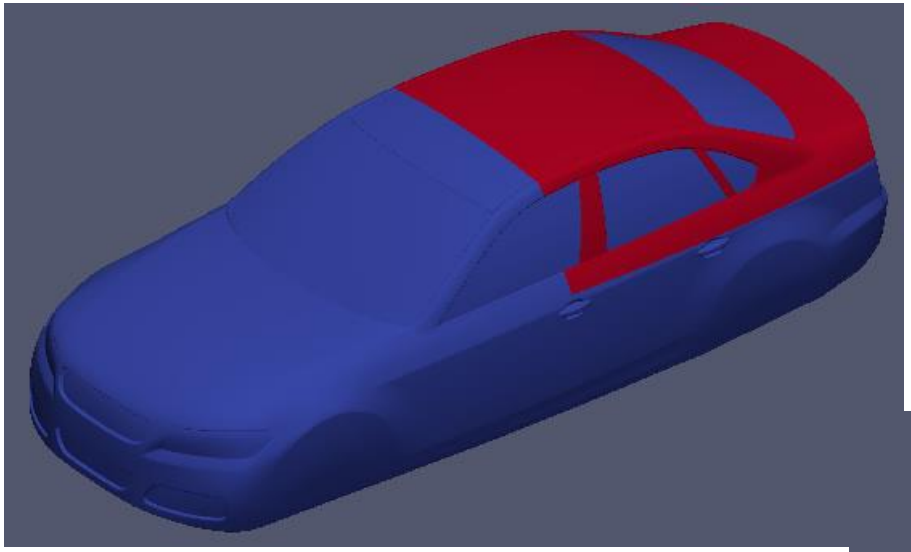
part_01_Body	2013/04/15 11:17	証明書信頼リスト	111,063 KB
part_01_Body_Closed_2	2013/04/30 15:00	証明書信頼リスト	132,893 KB
part_02_UB_Detailed	2013/04/25 14:21	証明書信頼リスト	269,631 KB
part_02_UB_Smooth	2013/04/15 11:18	証明書信頼リスト	52,986 KB
part_03_Estate	2013/04/15 11:18	証明書信頼リスト	62,328 KB
part_03_Fastback	2013/04/15 11:16	証明書信頼リスト	57,469 KB
part_03_Notchback	2013/04/15 11:15	証明書信頼リスト	53,590 KB
part_04_Mirror	2013/04/30 14:59	証明書信頼リスト	8,312 KB
part_04_Mirror_Cover	2013/04/15 11:13	証明書信頼リスト	2,242 KB
part_05_Wheels_Front	2013/10/14 13:32	証明書信頼リスト	42,417 KB
part_05_Wheels_Front_Closed	2013/04/17 9:32	証明書信頼リスト	54,933 KB
part_05_Wheels_Front_Smooth	2013/04/25 16:18	証明書信頼リスト	103,012 KB
part_06_Wheels_Rear	2013/10/14 13:26	証明書信頼リスト	160,172 KB
part_06_Wheels_Rear_Closed	2013/04/17 11:04	証明書信頼リスト	54,427 KB
part_06_Wheels_Rear_Smooth	2013/04/26 9:49	証明書信頼リスト	88,905 KB

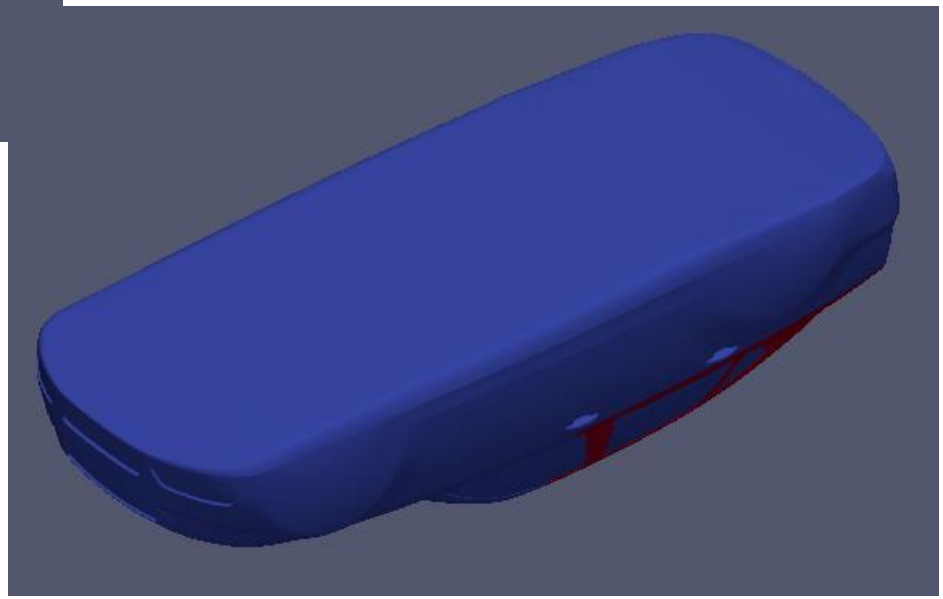
公開されているパーツ類　これを組み合わせてモデルを作ります。




タイヤは前後別で





タイヤは前後別で




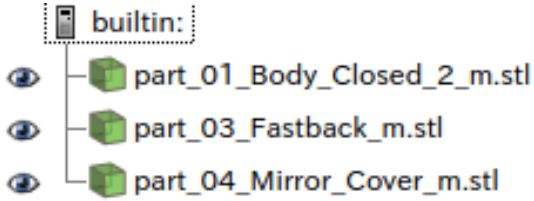
エンジン部分のモデルはありません

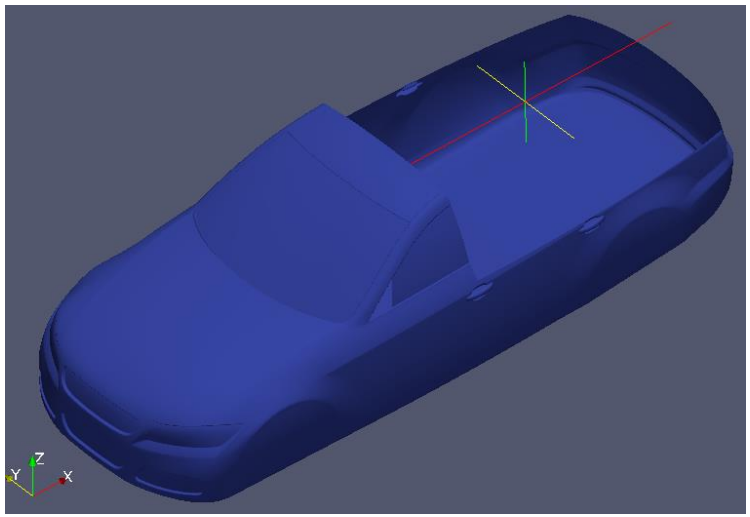
組み合わせ事例1（一番簡単なモデル）



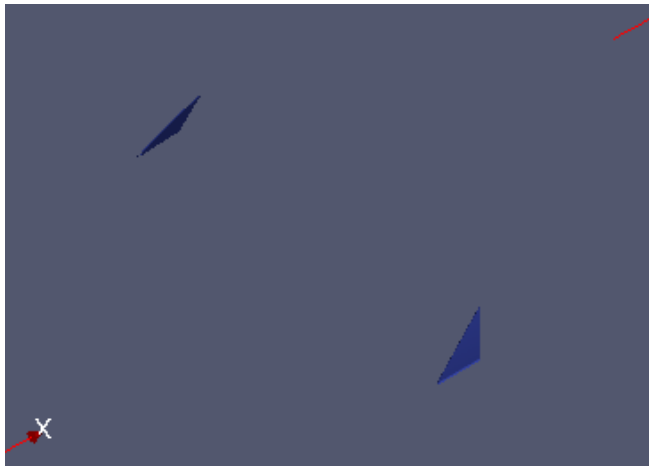


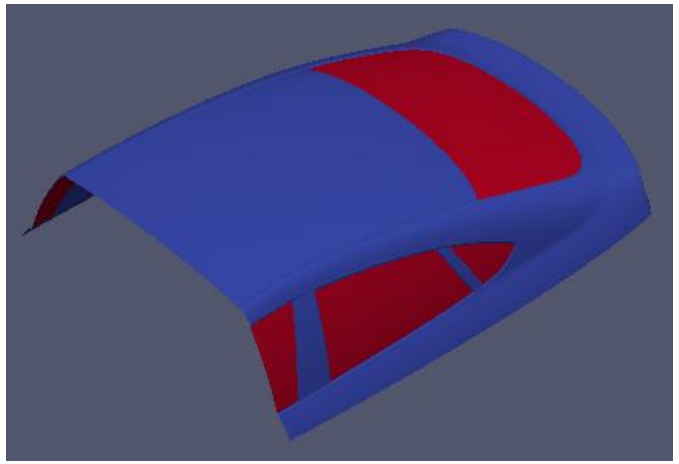
組み合わせ事例1（一番簡単なモデル）



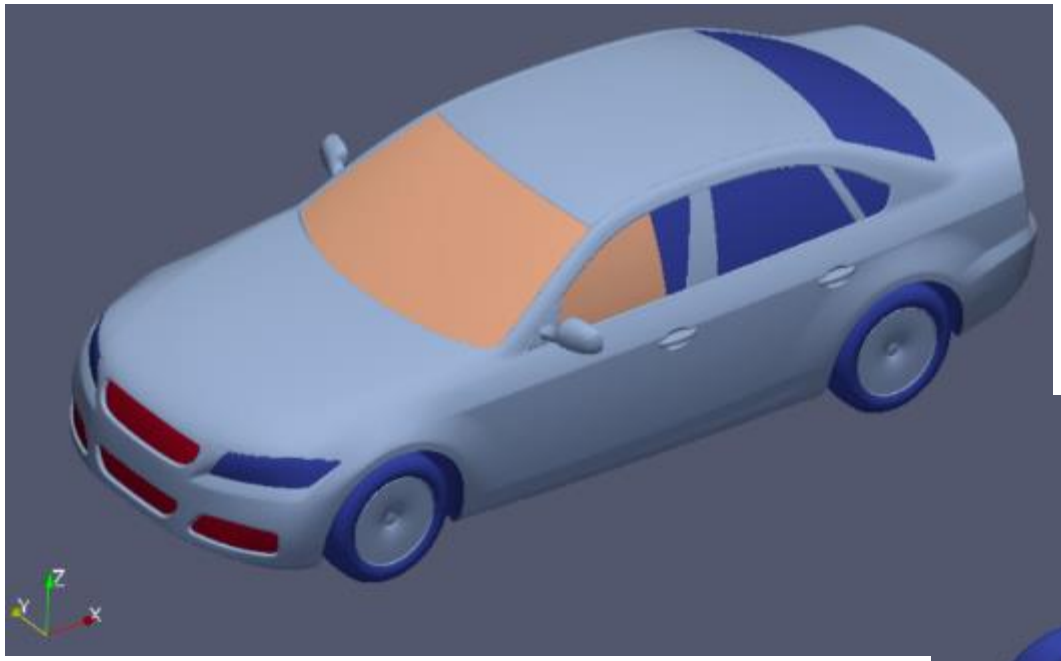


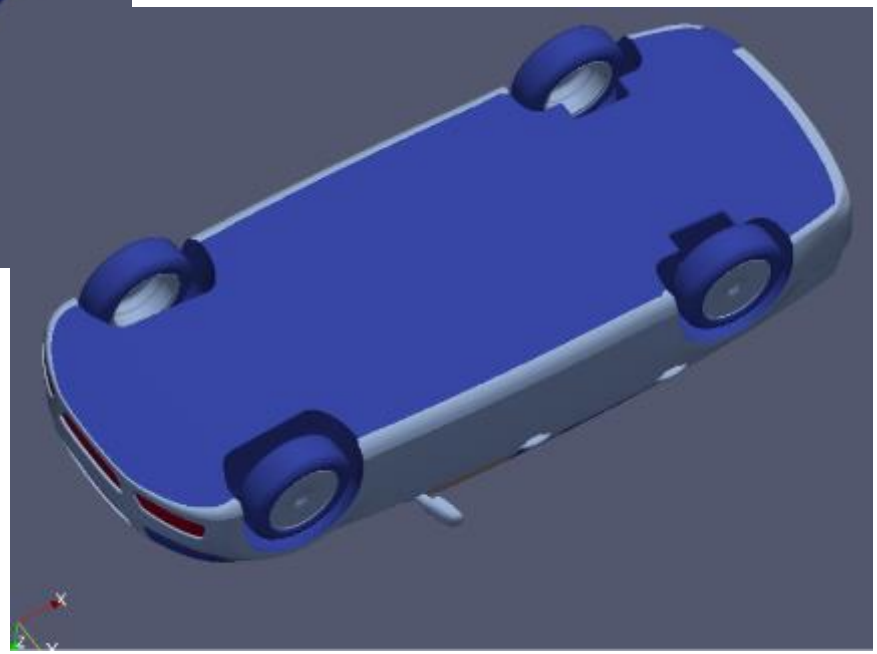
3パーツで構成



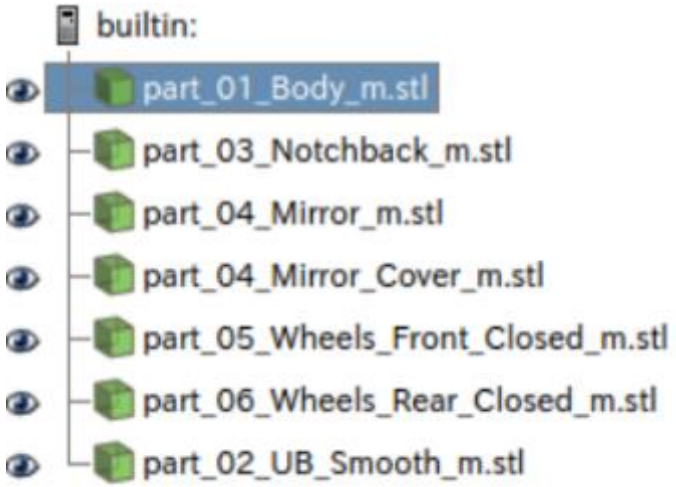


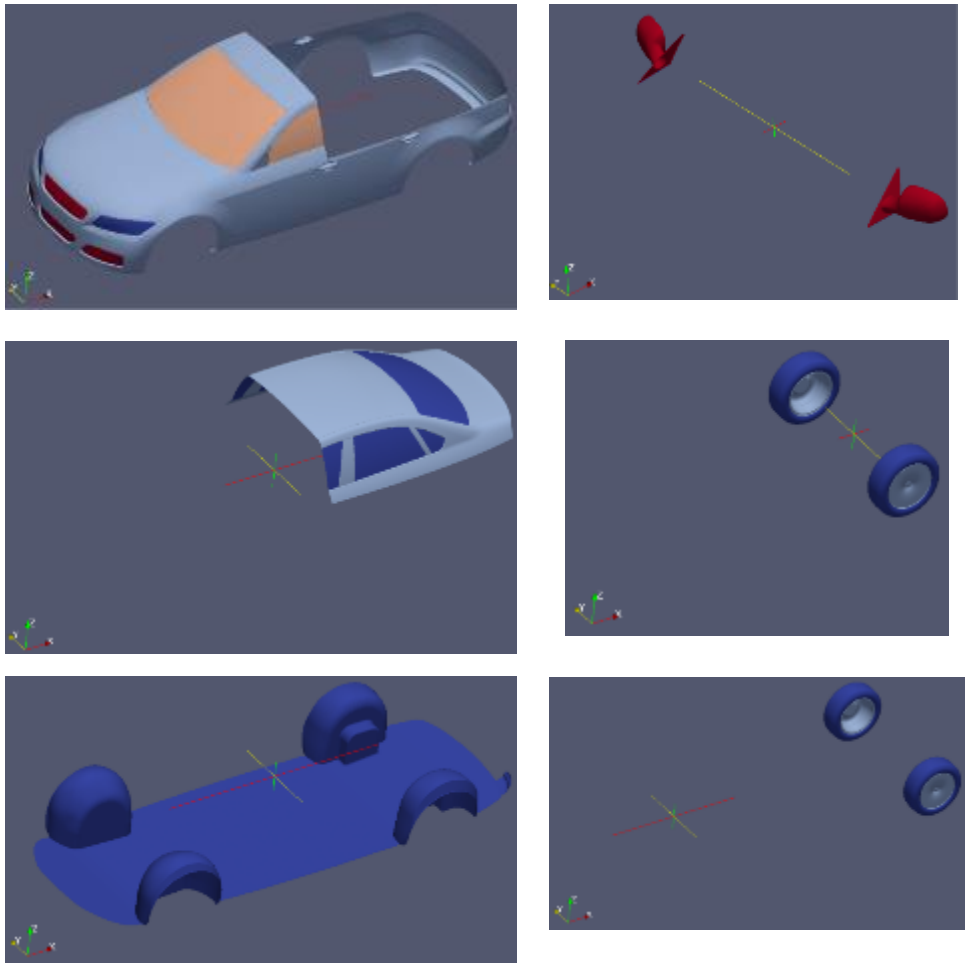
組み合わせ事例2（少し車らしくなってきた？）





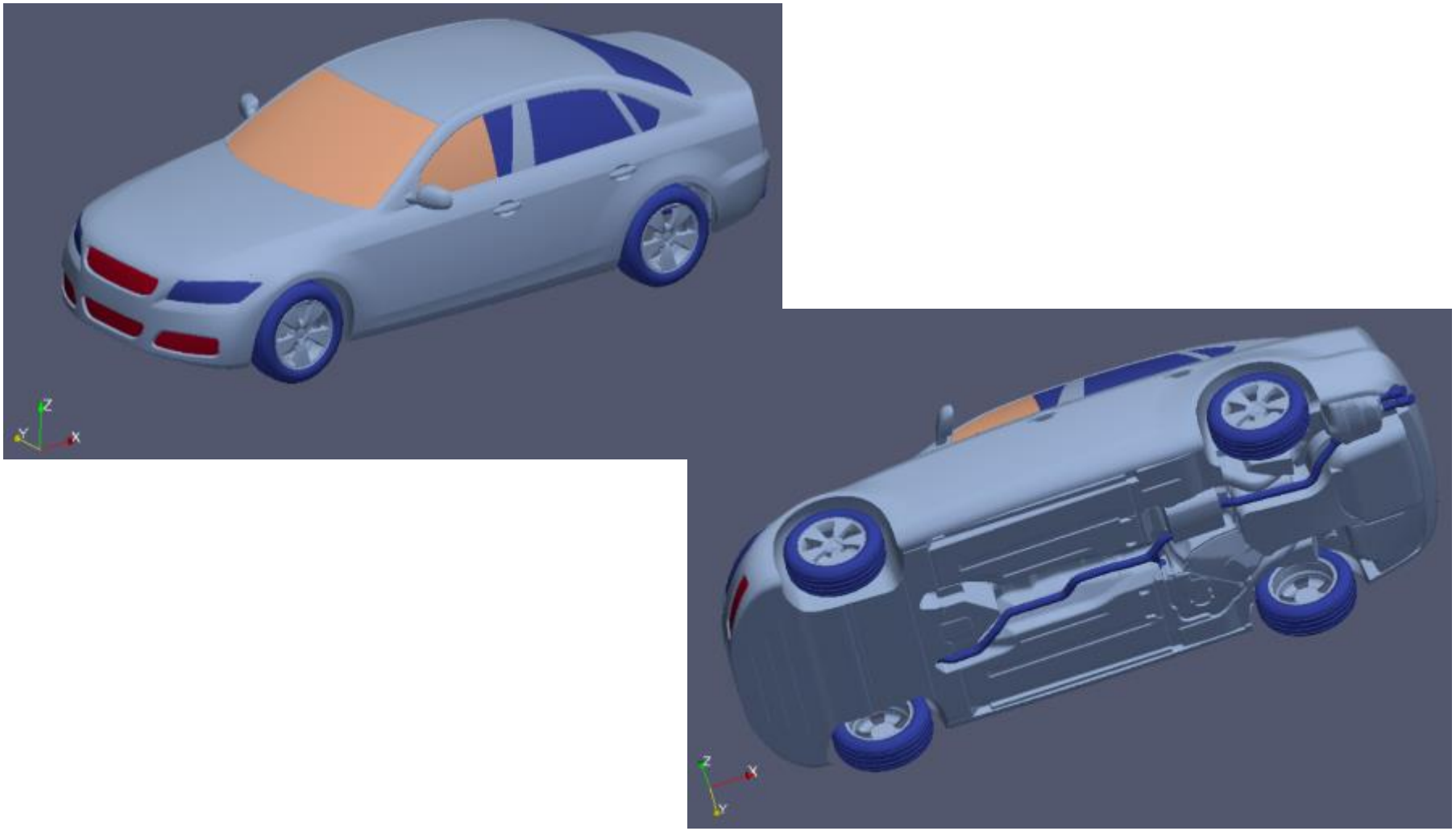
組み合わせ事例2（少し車らしくなってきた？）



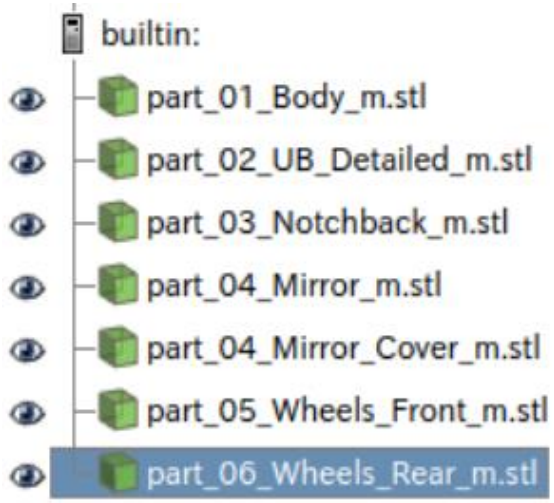


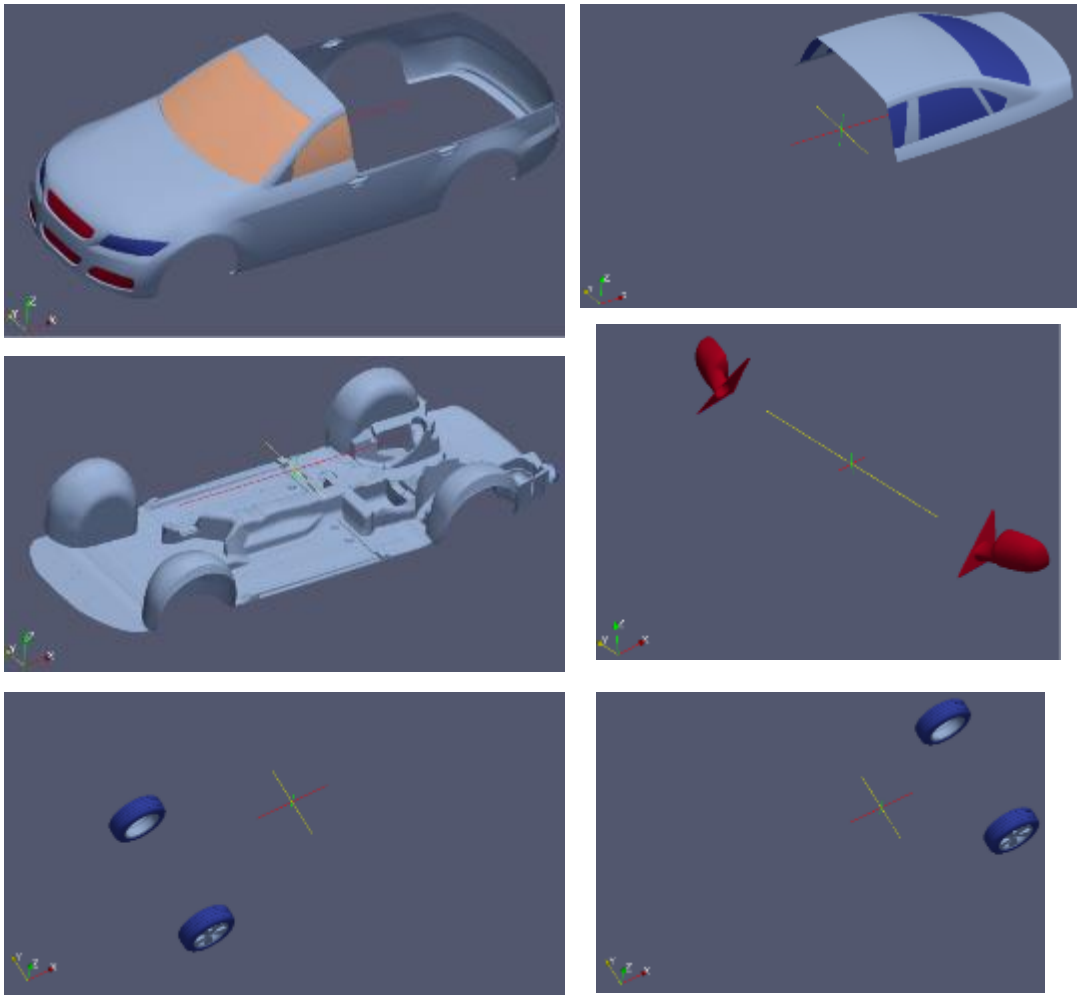
6パーツで構成

組み合わせ事例3（公開されているモデルで一番詳細）



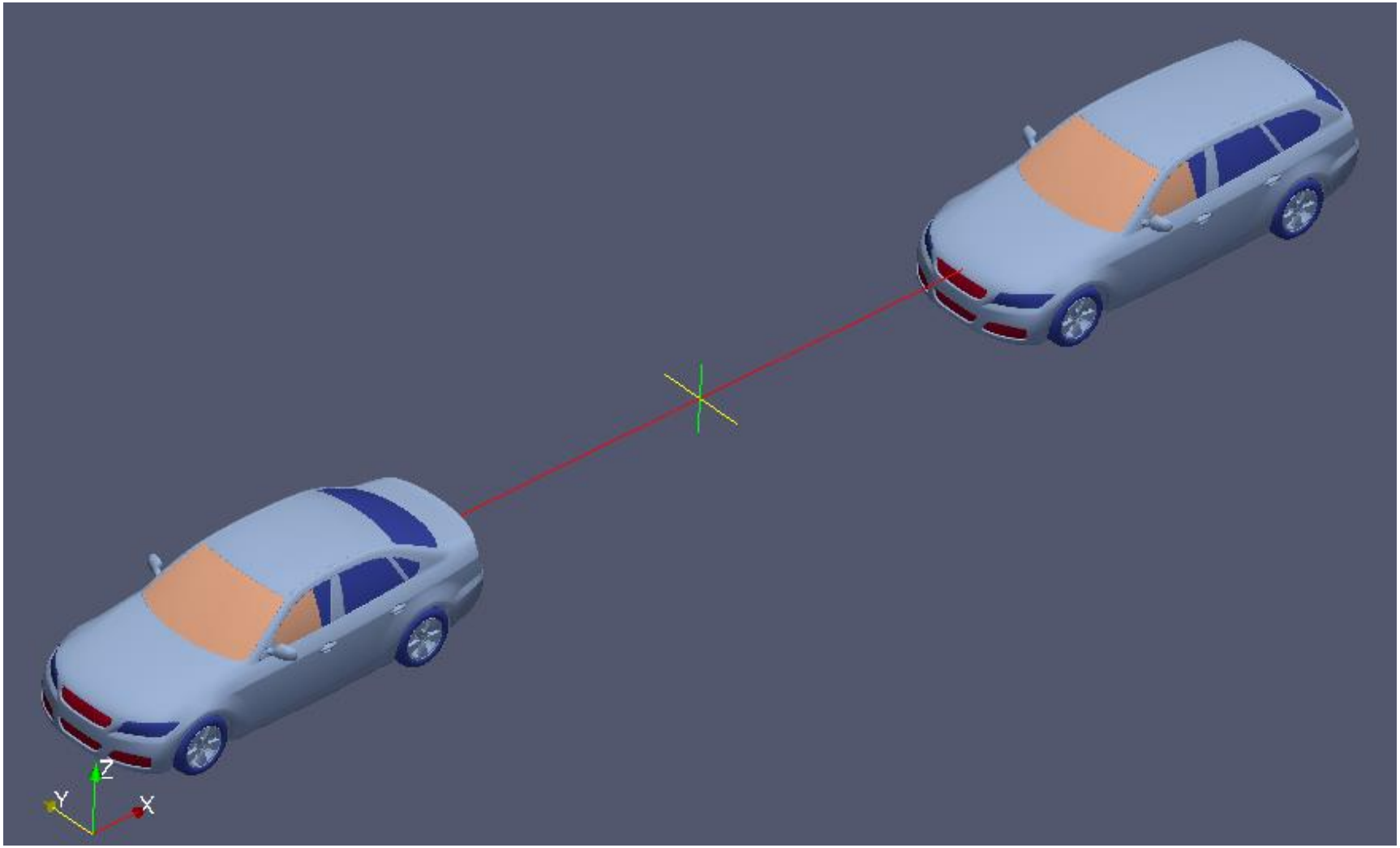
組み合わせ事例3（公開されているモデルで一番詳細）





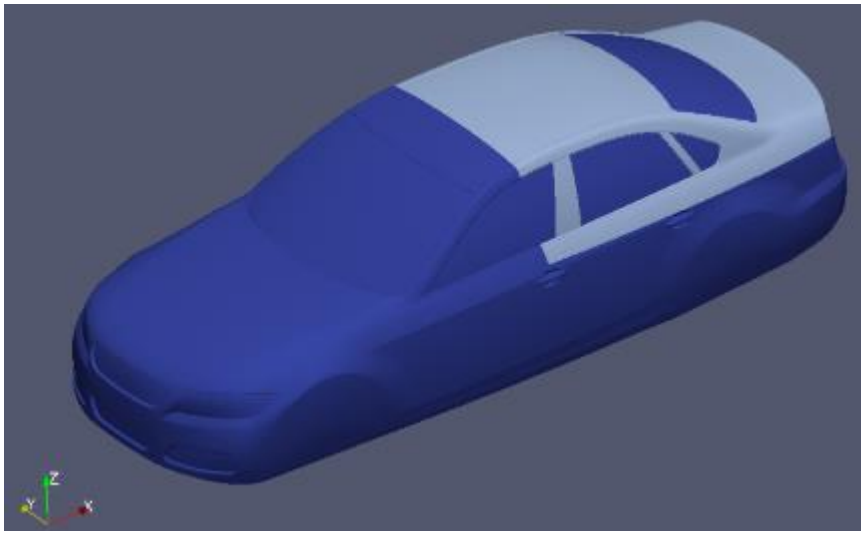
6パーツで構成

組み合わせ事例4　追従走行などのモデルも作成できます。



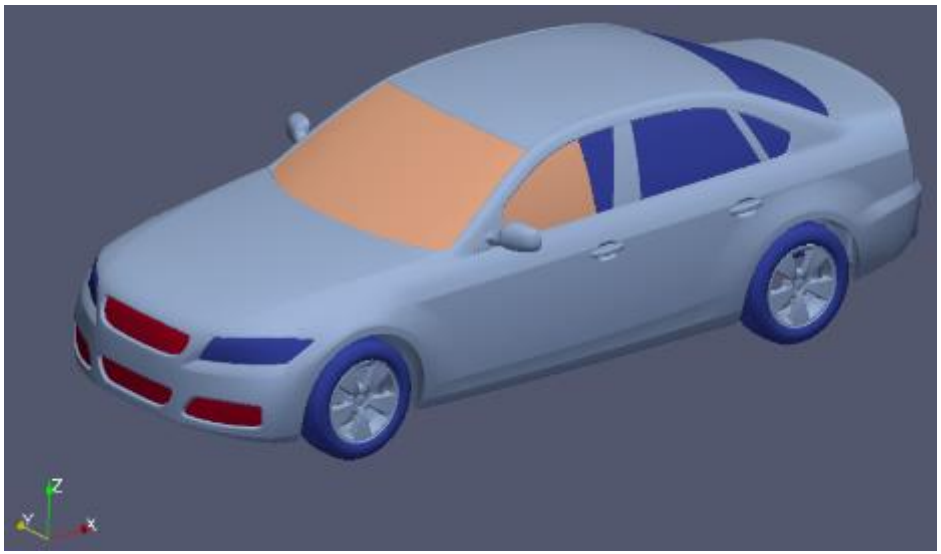
3. 計算モデル作成

ここで紹介する計算モデルは，一番簡略されたモデルと詳細モデルの2タイプです。



簡略モデル

詳細モデル



モデル作成の基本的な流れ

①モデルの縮尺を実車スケールに戻す
②stlモデルをチェック（paraView，エディター）
③blockMesh
④snappyHexMesh
⑤checkMesh　作成したメッシュ品質のチェック
⑥paraFoam　メッシュの目視チェック

流体解析や理論にどんなに詳しくても，モデルを作成し計算しない限り答えは得られません。OpenFOAMにもソフト固有の利用技術が存在しますので，それを早く習得する事がまず第一です（と考えています）。

自分で分からない場合には，下記に質問すると良いでしょう。

https://groups.google.com/forum/#!forum/openfoam

公開されているモデルは，1:2.5と縮尺がかかっているため，ここでは実車スケールに戻します。

surfaceTranfoamPointsコマンドを使い，長さをmmからmに直し，同時にスケールを修正します。

```
surfaceTransformPoints part_01_Body.stl                part_01_Body_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_01_Body_Closed_2.stl       part_01_Body_Closed_2_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_02_UB_Detailed.stl         part_02_UB_Detailed_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_02_UB_Smooth.stl           part_02_UB_Smooth_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_03_Estate.stl              part_03_Estate_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_03_Fastback.stl            part_03_Fastback_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_03_Notchback.stl           part_03_Notchback_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_04_Mirror.stl              part_04_Mirror_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_04_Mirror_Cover.stl        part_04_Mirror_Cover_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_05_Wheels_Front.stl        part_05_Wheels_Front_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_05_Wheels_Front_Closed.stl part_05_Wheels_Front_Closed_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_05_Wheels_Front_Smooth.stl part_05_Wheels_Front_Smooth_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_06_Wheels_Rear.stl         part_06_Wheels_Rear_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_06_Wheels_Rear_Closed.stl  part_06_Wheels_Rear_Closed_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
surfaceTransformPoints part_06_Wheels_Rear_Smooth.stl  part_06_Wheels_Rear_Smooth_m.stl -scale "(0.001357208808 0.001357208808 0.001357208808)"
```

変換前のファイル　　変換後のファイル

X,Y,Zの各方向を伸ばす

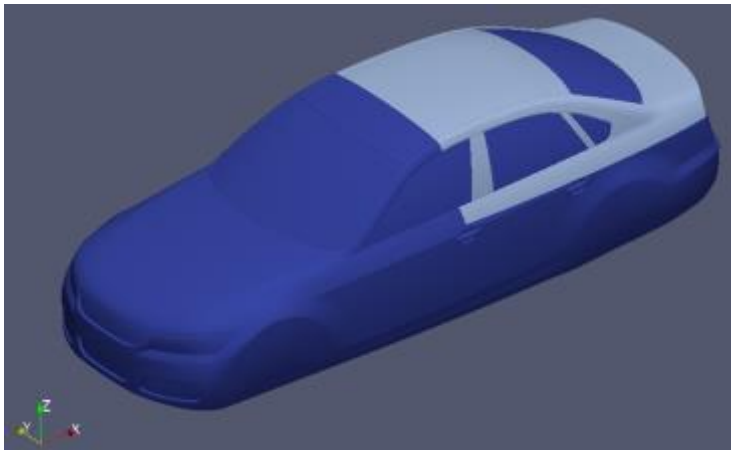
$$\frac{2.5^{\left(\frac{1}{3}\right)}}{1000}$$

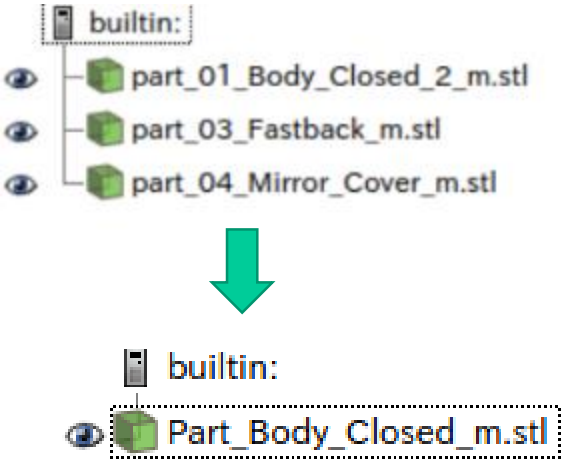
stlファイルには複数のパーツが含まれているものもあるため，OpenFOAMで利用する場合，メッシュの作成方法によっては修正する必要が出てくる場合があります。

part_01_Bpdy_m.stl	lights_front
	Body
	windows
	front_intakes
part_02_UB_Detailed_m.stl	ExhaustSystem
	Underbody_Detailed
part_03_Notchback_m.stl	Body
	windows
part_04_Mirror_m.stl	Body
	Mirrors
part_04_Mirror_Cover_m.stl	そのまま
part_05_Wheels_Front_m.stl	wheels_fr
	tyres_fr
part_06_Wheels_Rear_m.stl	wheels_rear
	tyres_rear

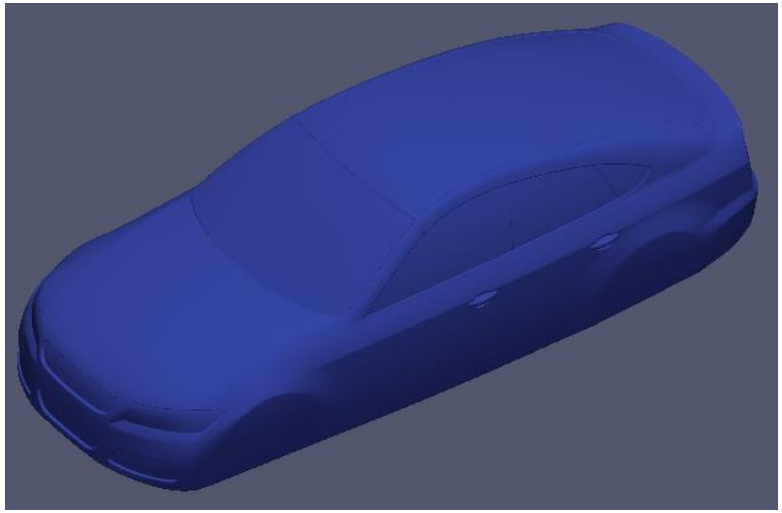
```
      vertex -437.445458 733.799253 400.887435
      vertex -438.324809 732.439486 401.588813
      vertex -436.372045 732.570347 402.724665
    endloop
  endfacet
endsolid lights_front
solid Body
   color       1 1 1
  facet normal 0.0313914399 0.999321839 -0.0192468079
    outer loop
      vertex 2171.33871 881.805606 172.4643
      vertex 2174.9428 881.459457 160.369995
      vertex 2156.40546 882.084806 162.604714
```

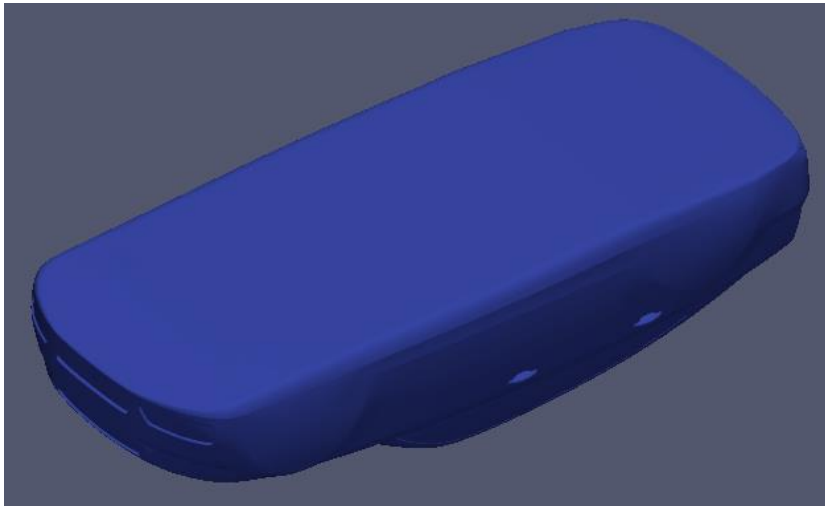
簡略モデルは次のように1つのstlファイルをまとめます。





エディターで1つにまとめる





簡略モデルの寸法確認

Properties

Filename: Part_Body_Closed_m.stl

Path: /Desktop/ver_2.3.x/automotive_aerodynamics/STL/cource

Statistics

Type: Polygonal Mesh

Number of Cells: 854062

Number of Points: 427039

Memory: 35 MB

Data Arrays

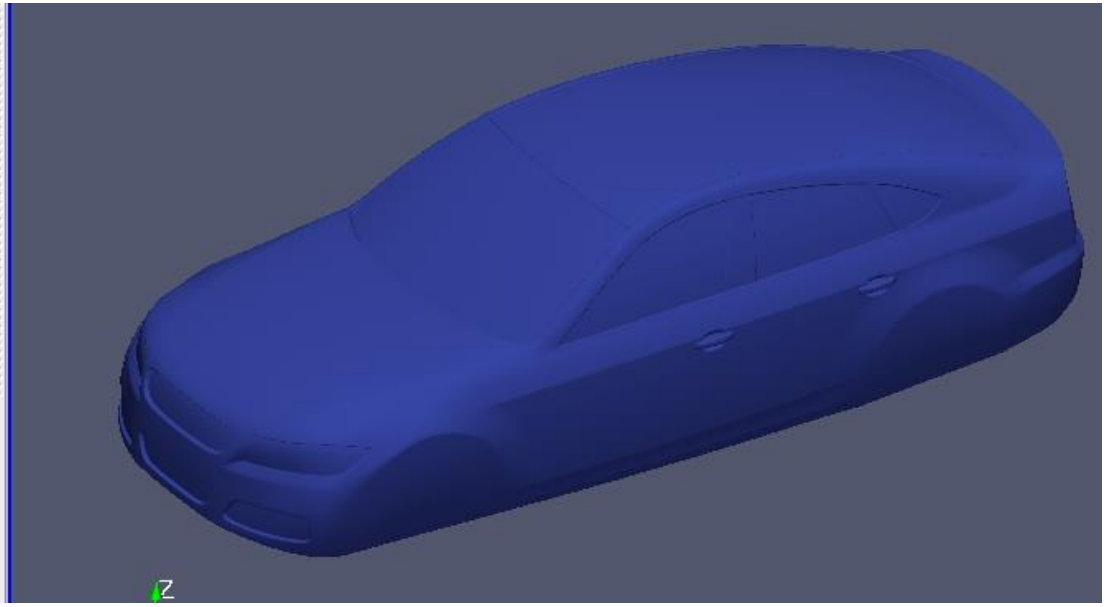
Name	Data Type	Data Ranges
STLSolidLabeling	float	[0, 0]

Bounds

X range: -1.1 to 5.16 (delta: 6.26)

Y range: -1.24 to 1.24 (delta: 2.47)

Z range: -0.227 to 1.49 (delta: 1.72)



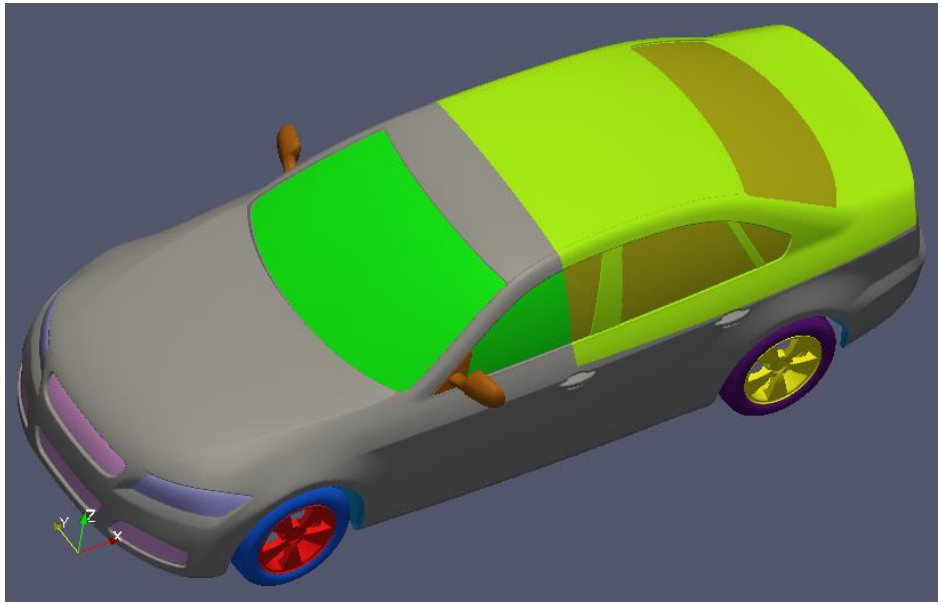
Bounds

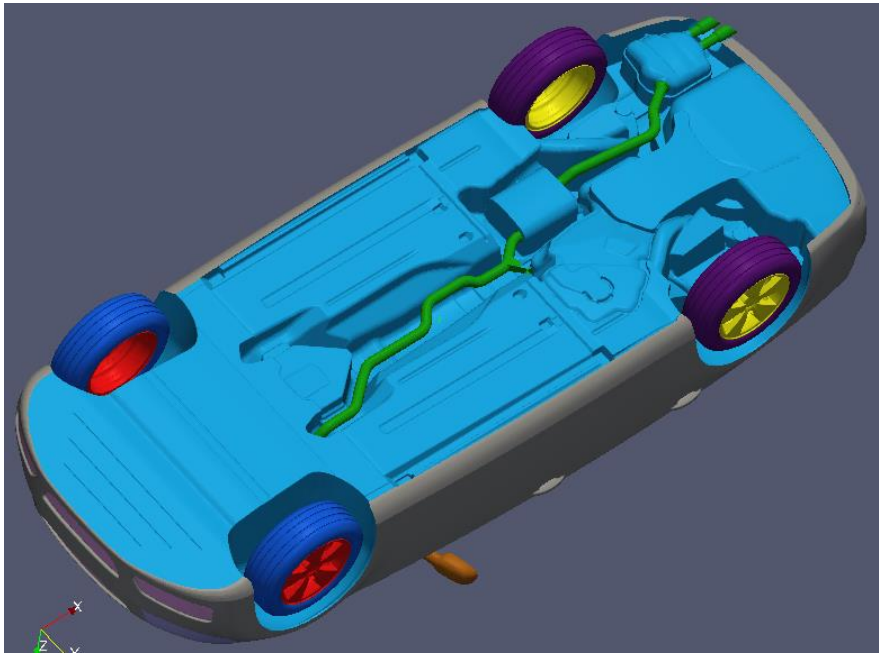
X range: -1.1 to 5.16 (delta: 6.26)

Y range: -1.24 to 1.24 (delta: 2.47)

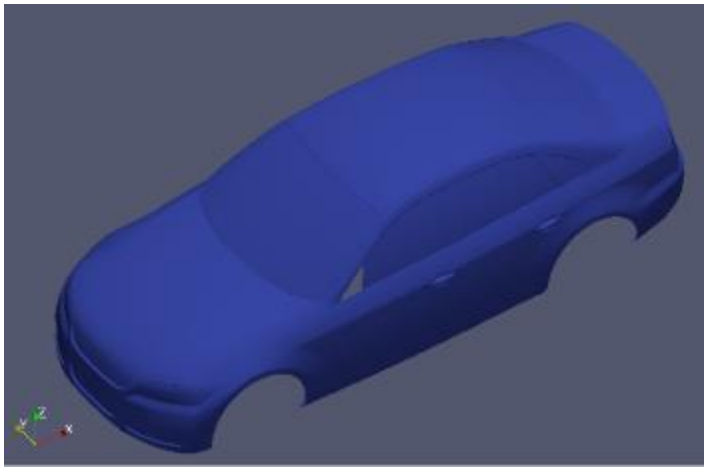
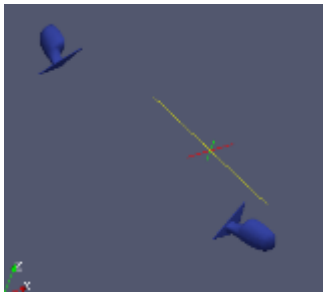
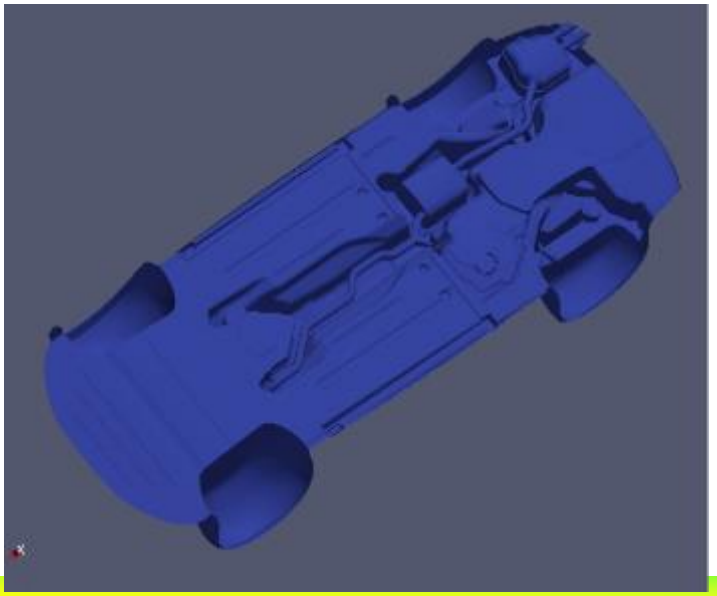
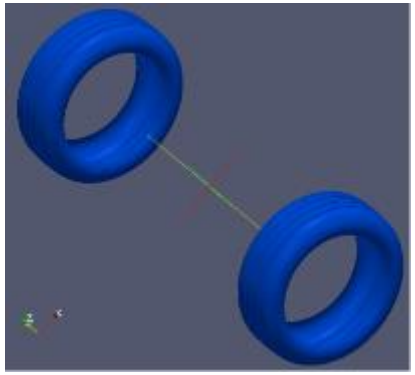
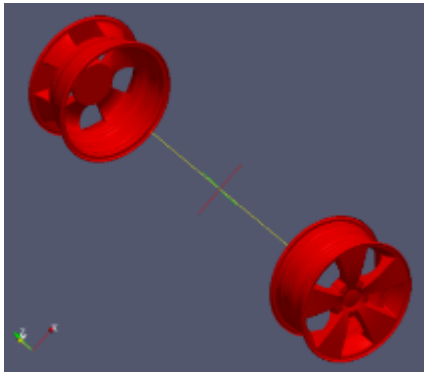
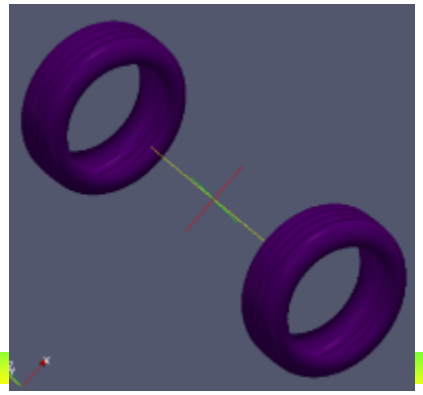
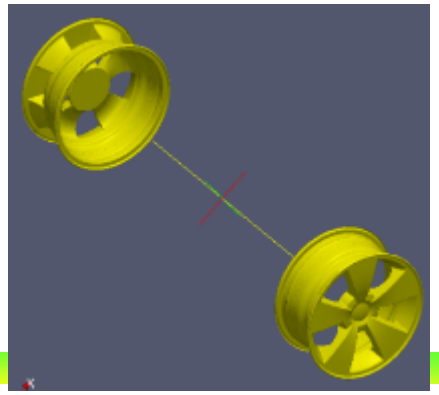
Z range: -0.227 to 1.49 (delta: 1.72)

詳細モデルは次のように多くの領域に分かれています

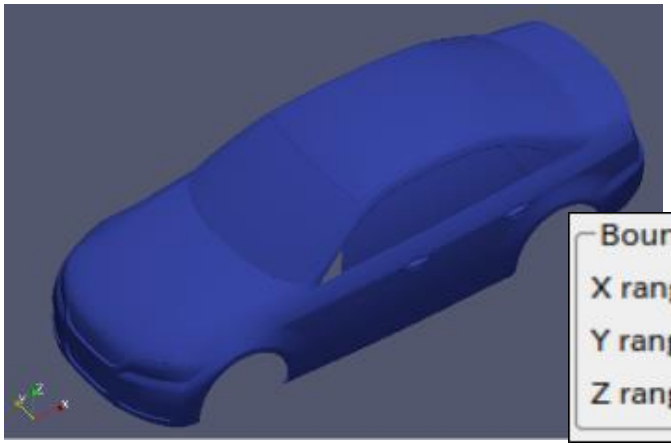




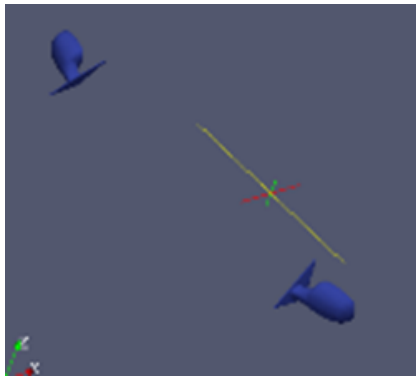
詳細モデルは次のようにstlファイルをまとめます。

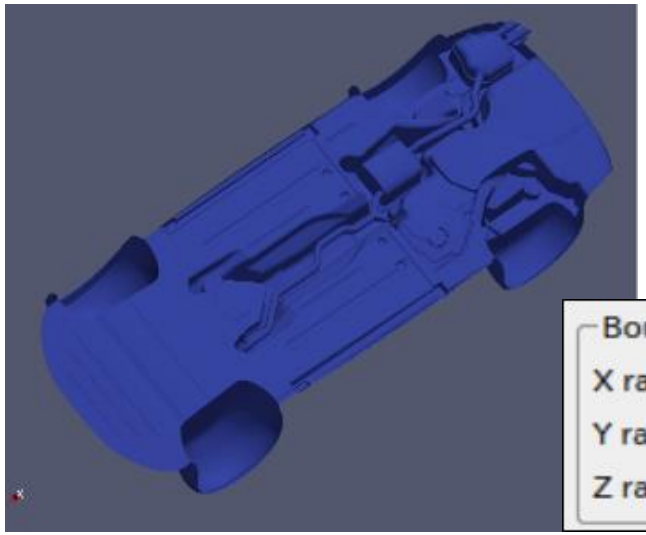
詳細モデルの寸法確認



Bounds
X range: -1.1 to 5.16 (delta: 6.26)
Y range: -1.24 to 1.24 (delta: 2.47)
Z range: -0.219 to 1.5 (delta: 1.72)

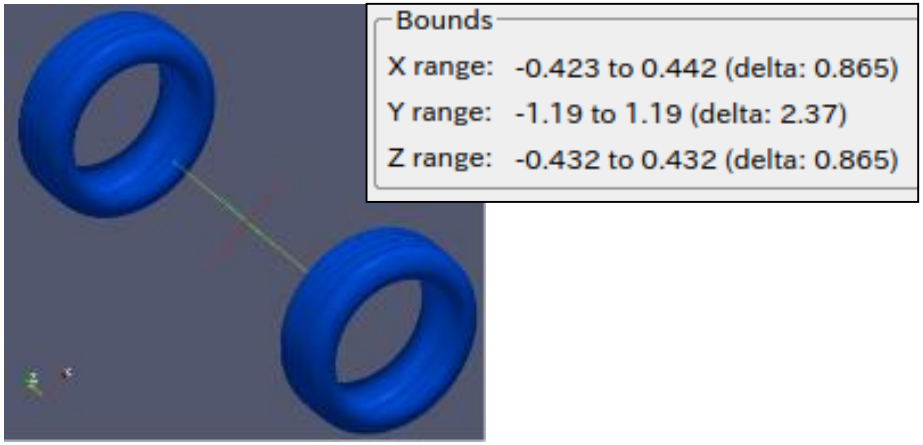


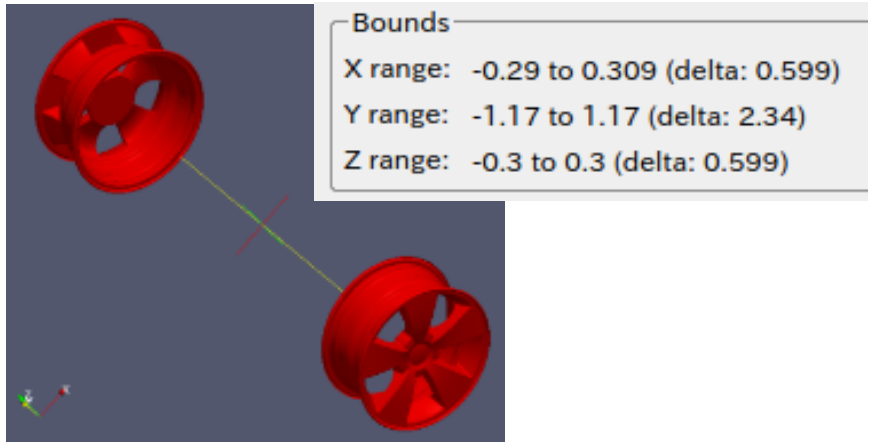
Bounds
X range: 1.01 to 1.36 (delta: 0.349)
Y range: -1.38 to 1.38 (delta: 2.76)
Z range: 0.823 to 1.06 (delta: 0.236)

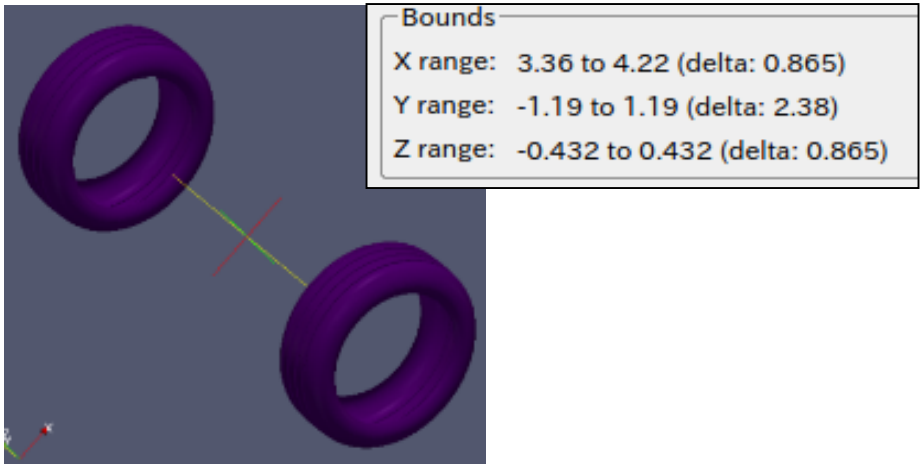


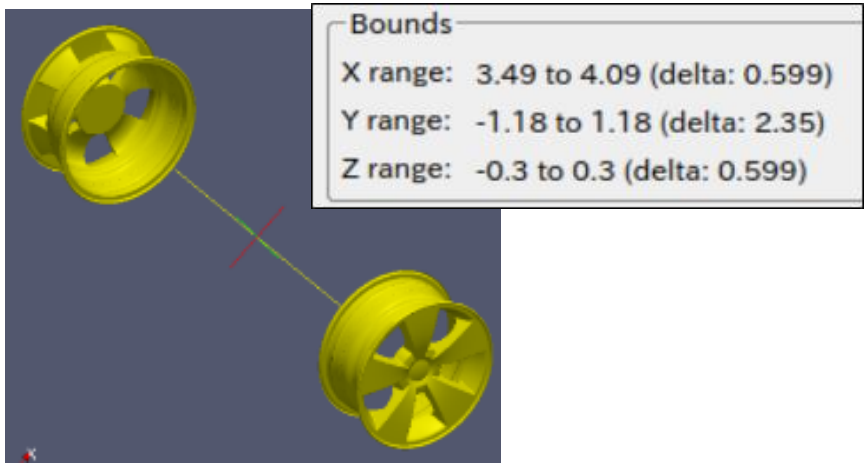
Bounds
X range: -1.1 to 5.16 (delta: 6.26)
Y range: -1.24 to 1.24 (delta: 2.47)
Z range: -0.219 to 1.5 (delta: 1.72)

詳細モデルの寸法確認


Bounds
X range: -0.423 to 0.442 (delta: 0.865)
Y range: -1.19 to 1.19 (delta: 2.37)
Z range: -0.432 to 0.432 (delta: 0.865)


Bounds
X range: -0.29 to 0.309 (delta: 0.599)
Y range: -1.17 to 1.17 (delta: 2.34)
Z range: -0.3 to 0.3 (delta: 0.599)

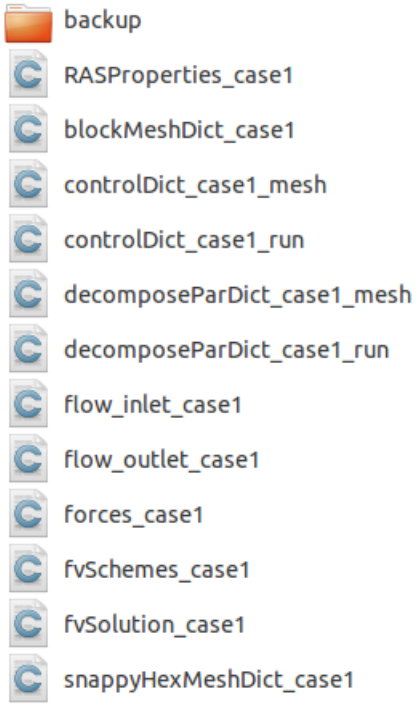

Bounds
X range: 3.36 to 4.22 (delta: 0.865)
Y range: -1.19 to 1.19 (delta: 2.38)
Z range: -0.432 to 0.432 (delta: 0.865)


Bounds
X range: 3.49 to 4.09 (delta: 0.599)
Y range: -1.18 to 1.18 (delta: 2.35)
Z range: -0.3 to 0.3 (delta: 0.599)

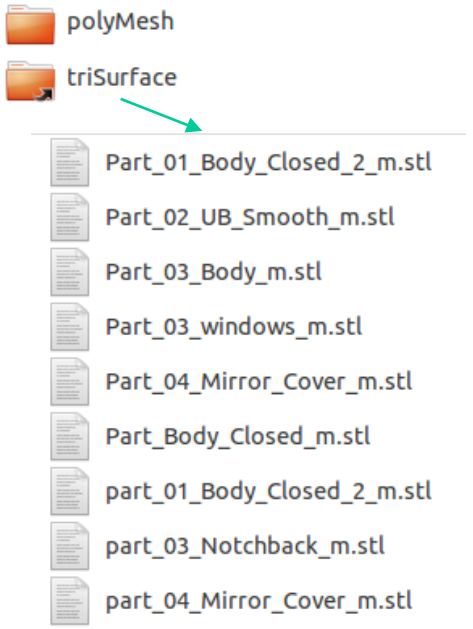
ホルダ構成事例

③　メッシュ作成コマンド
④　計算実施コマンド
⑤　境界作成コマンド

①：OpenFOAMの固有ホルダでない自分の場合は，元ファイルをここに置いてコピーしている

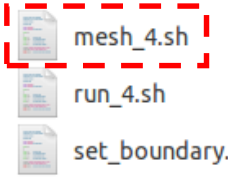


②



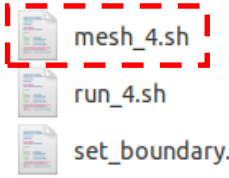
形状の定義ファイル

コマンドによるメッシュ作成事例




```
echo "Start mesh"
CASE_DIR=case_set/case1/set
BLOCK_MESH=blockMeshDict_case1
DECOMPOSE_PAR=decomposeParDict_case1_mesh
SNAPPYHEXMESH_DICT=snappyHexMeshDict_case1
SNAPPYHEXMESH_FEATURE=surfaceFeatureExtractDict_1
SNAPPYHEXMESH_DEFAULT=surfaceFeatureExtractDictDefaults_1
CONTROL_DICT=controlDict_case1_mesh
FV_SCH=fvSchemes_case1
FV_SOL=fvSolution_case1

echo "blockMesh"
cp -r  $CASE_DIR/$CONTROL_DICT system/controlDict
cp -r  $CASE_DIR/$BLOCK_MESH constant/polyMesh/blockMeshDict
$runApplication blockMesh > log.blockMesh

echo "surfaceFeatureExtract"
cp -r  $CASE_DIR/$SNAPPYHEXMESH_FEATURE system/surfaceFeatureExtractDict
cp -r  $CASE_DIR/$SNAPPYHEXMESH_DEFAULT system/surfaceFeatureExtractDictDefaults
$runApplication surfaceFeatureExtract

echo "decomposePar 4 blocks"
cp -r  $CASE_DIR/$DECOMPOSE_PAR system/decomposeParDict
$runApplication decomposePar > log.decomposePar

echo "snappyHexMesh 4CPU"
cp -r $CASE_DIR/$SNAPPYHEXMESH_DICT system/snappyHexMeshDict
cp -r $CASE_DIR/$FV_SCH  system/fvSchemes
cp -r $CASE_DIR/$FV_SOL  system/fvSolution
$runApplication mpirun -np 4 snappyHexMesh -parallel > log.snappyHexMesh

echo "reconstructParMesh"
$runApplication reconstructParMesh -time 3 -mergeTol 1e-6 > log.reconstructParMesh
```

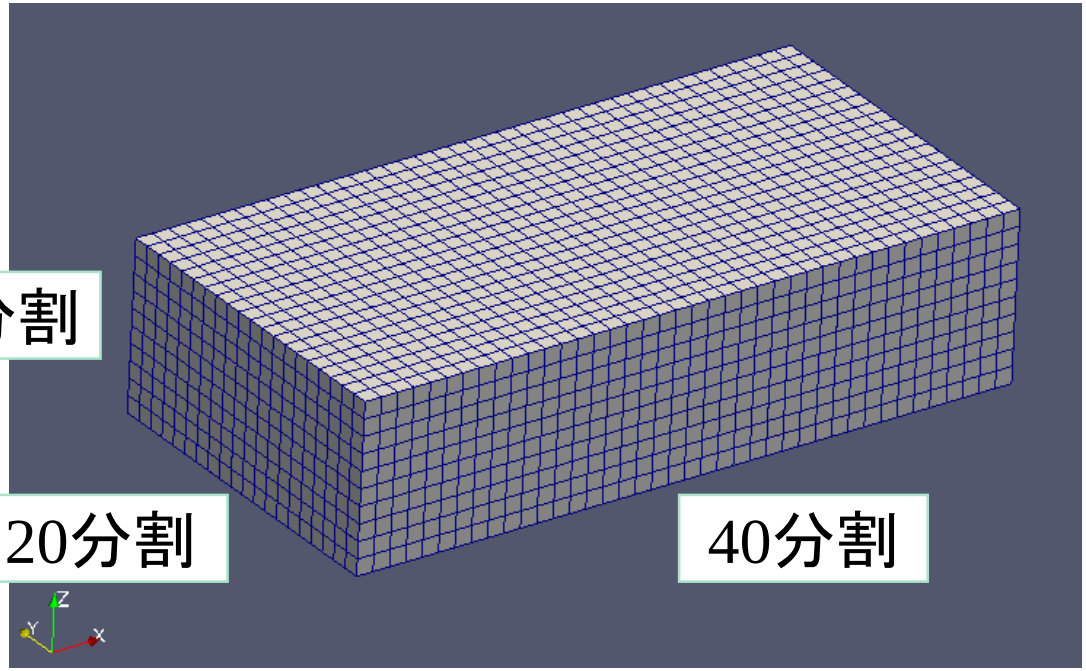
コマンドによるメッシュ作成事例




```
rm -r processor0
rm -r processor1
rm -r processor2
rm -r processor3

echo "checkMesh"
$runApplication checkMesh -allGeometry -allTopology> log.checkMesh

echo "renumberMesh"
$runApplication renumberMesh -time 3 > log.renumberMesh

echo "rename holder name"
mv constant/polyMesh constant/polyMesh_blockMesh

echo "set data"
cp -r 4/polyMesh constant
rm -r 3
rm -r 4

echo "End of mesh"
```

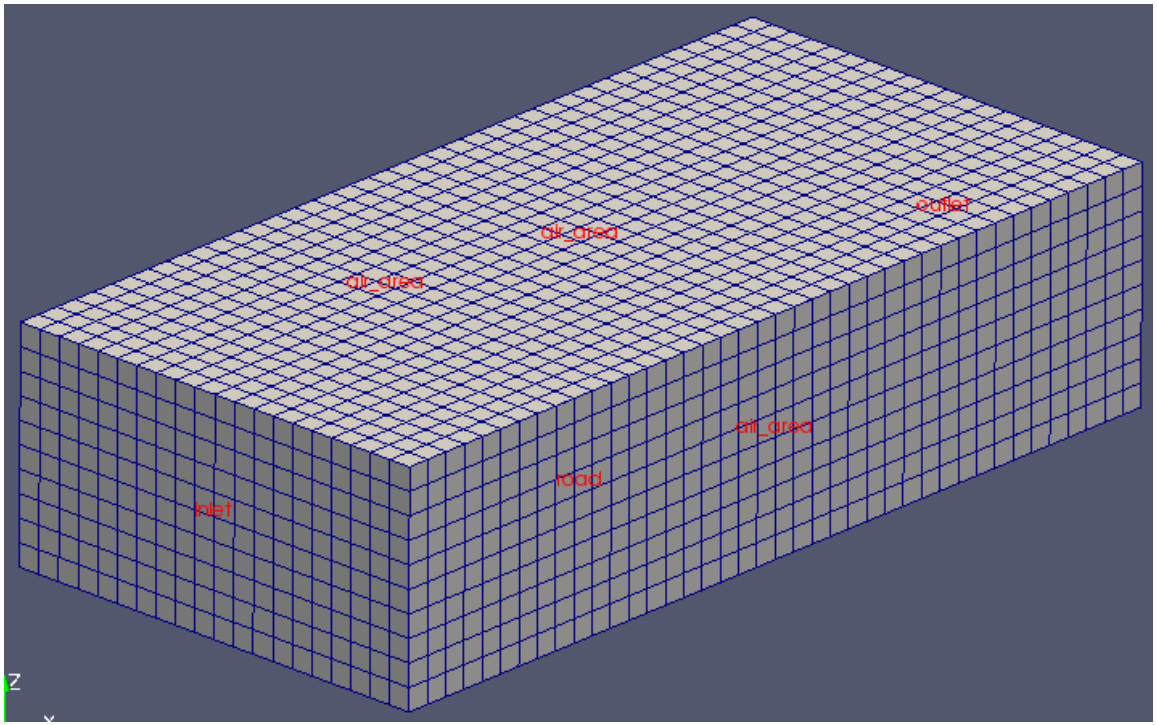
blcokMesh（簡易モデル，詳細モデル共通）

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

convertToMeters 1;

vertices
(
    (-10.96  -10.48  -0.432)
    ( 30.0   -10.48  -0.432)
    ( 30.0    10.0   -0.432)
    (-10.96   10.0   -0.432)
    (-10.96  -10.48   9.808)
    ( 30.0   -10.48   9.808)
    ( 30.0    10.0    9.808)
    (-10.96   10.0    9.808)
);

blocks
(
  hex (0 1 2 3 4 5 6 7) (40 20 10) simpleGrading (1 1 1)
);
```



出来るだけサイコロ形状に

blcokMesh

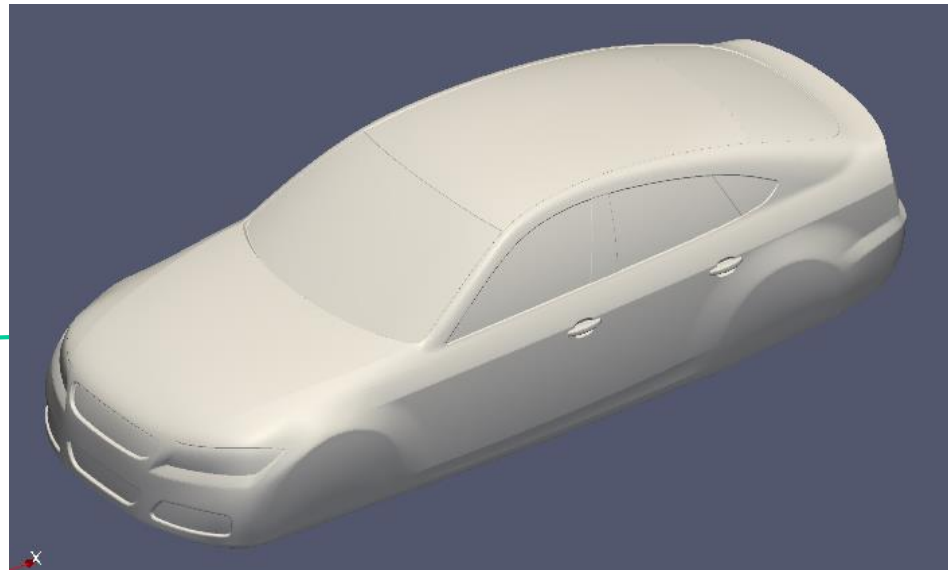



```
boundary
(
    inlet
    {
        type patch;
        faces
        (
            (0 4 7 3)
        );
    }

    outlet
    {
        type patch;
        faces
        (
            (1 2 6 5)
        );
    }

    road
    {
        type wall;
        faces
        (
            (1 0 3 2)
        );
    }

    air_area
    {
        type wall;
        faces
        (
            (2 3 7 6)
            (6 7 4 5)
            (1 5 4 0)
        );
    }
);
```

簡略モデルのメッシュ作成

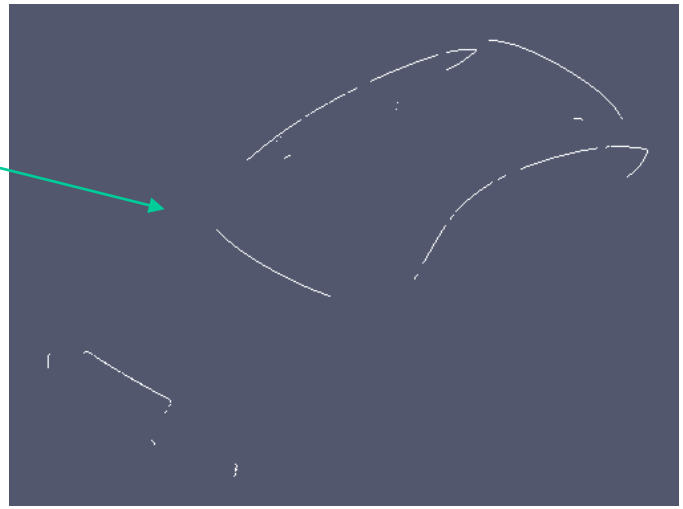
簡略モデルのメッシュ作成（sHM）

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      snappyHexMeshDict;
}
castellatedMesh true;
snap            true;
addLayers       true;
geometry
{

    Part_Body_Closed_m.stl
    {
        type        triSurfaceMesh;
        name        Body_Closed;
    }
};
castellatedMeshControls
{
    maxLocalCells 2000000;
    maxGlobalCells 8000000;
    minRefinementCells 10;
    maxLoadUnbalance 0.10;
    nCellsBetweenLevels 3;
```



簡略モデルのメッシュ作成（sHM）

```
    features
    (
        {
            file        "Part_Body_Closed_m.eMesh";
            level       5;
        }
    );
    refinementSurfaces
    {
       Body_Closed
       {
            level       (5 7);
       }
    }
    resolveFeatureAngle 30;
    refinementRegions
    {
    }
    locationInMesh (-1.5013 0.0013 0.0013);
    allowFreeStandingZoneFaces true;
}
snapControls
{
    nSmoothPatch 3;
    tolerance 2.0;
    nSolveIter 300;
    nRelaxIter 5;
    nFeatureSnapIter 10;
    implicitFeatureSnap false;
    explicitFeatureSnap true;
    multiRegionFeatureSnap false;
}
```



メッシュを切り始める初期位置

簡略モデルのメッシュ作成（sHM）

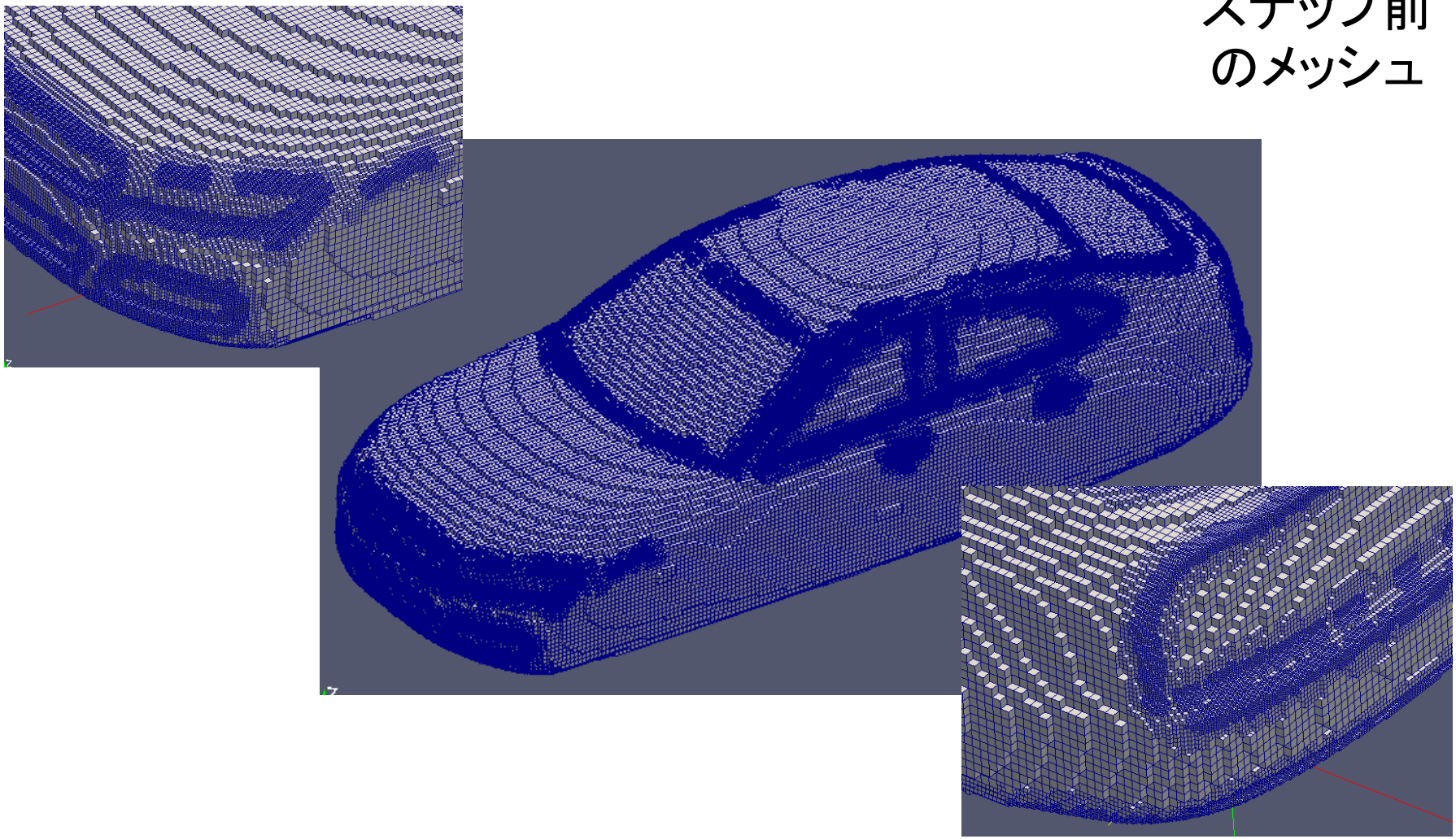
境界層の数を指定

```
addLayersControls
{
    relativeSizes true;
    expansionRatio 1.0;
    finalLayerThickness 0.3;
    minThickness 0.1;
    layers
    {
        Body_Closed
        {
            nSurfaceLayers 2;
        }
        road
        {
            nSurfaceLayers 3;
        }
    }
    nGrow 0;
    featureAngle 60;
    maxFaceThicknessRatio 0.5;
    nSmoothSurfaceNormals 1;
    nSmoothThickness 10;
    minMedialAxisAngle 90;
    maxThicknessToMedialRatio 0.3;
    nSmoothNormals 3;
    slipFeatureAngle 30;
    nRelaxIter 3;
    nBufferCellsNoExtrude 0;
    nLayerIter 50;
}
```

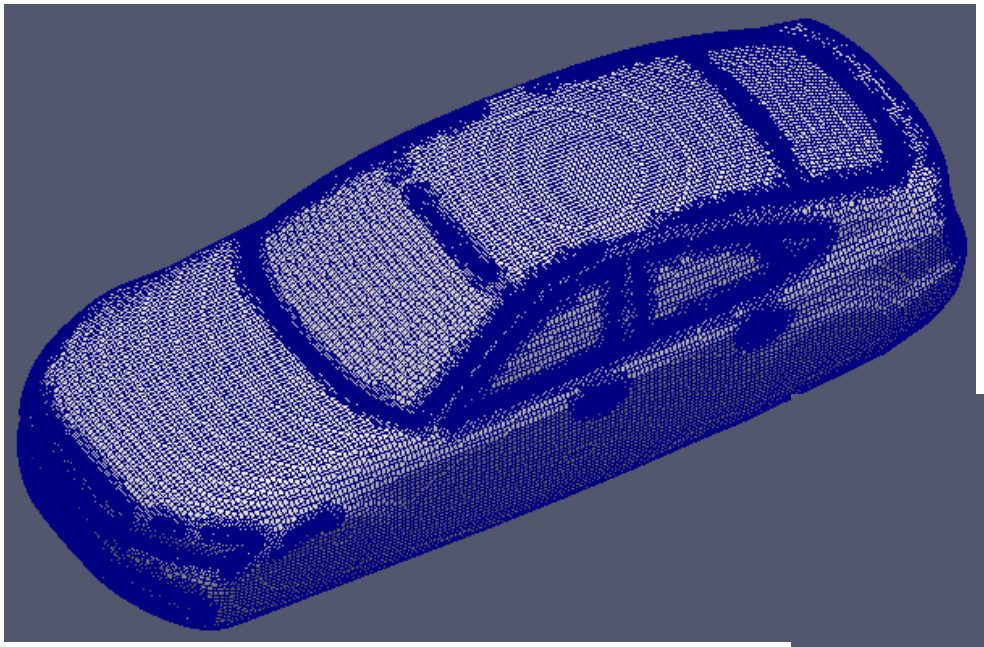
```
meshQualityControls
{
    maxNonOrtho 65;
    maxBoundarySkewness 20;
    maxInternalSkewness 4;
    maxConcave 80;
    minVol 1e-13;
    minTetQuality 1e-10;
    minArea -1;
    minTwist 0.02;
    minDeterminant 0.001;
    minFaceWeight 0.02;
    minVolRatio 0.01;
    minTriangleTwist -1;
    nSmoothScale 4;
    errorReduction 0.75;
}
debug 0;
mergeTolerance 1e-6;
```

簡略モデルのメッシュ作成（事例１）

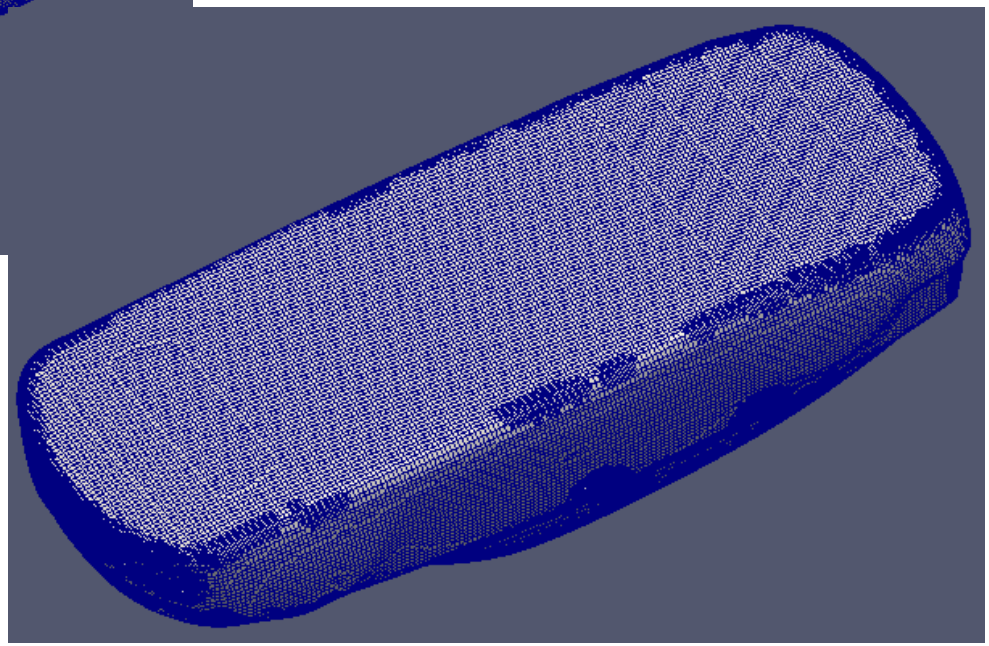
スナップ前
のメッシュ



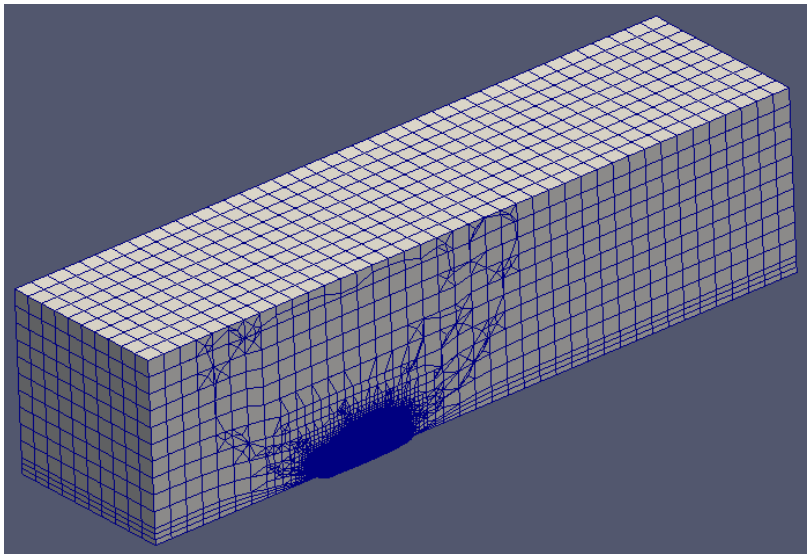
簡略モデルのメッシュ作成（事例１）

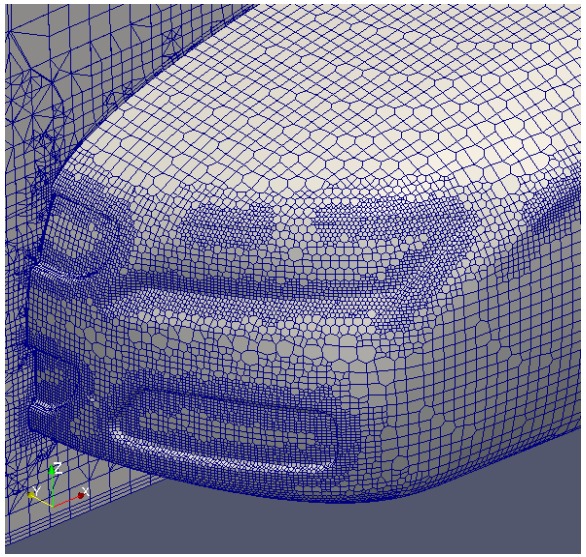


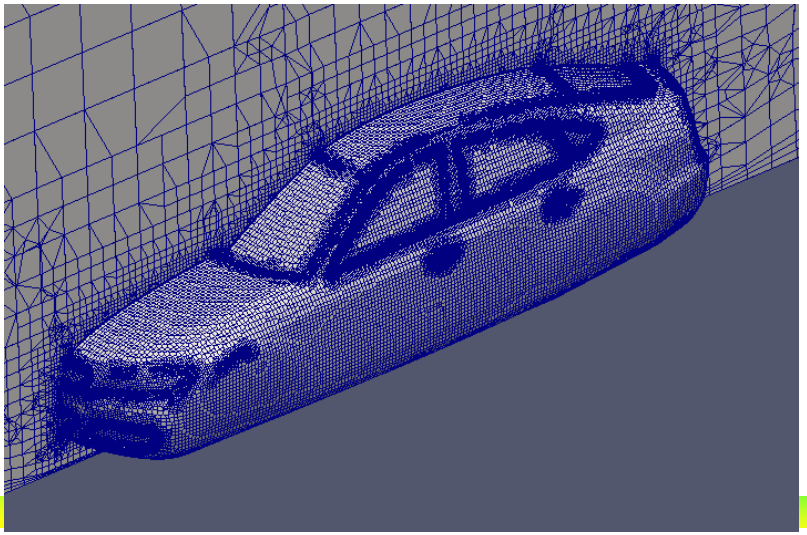
スナップ後
のメッシュ

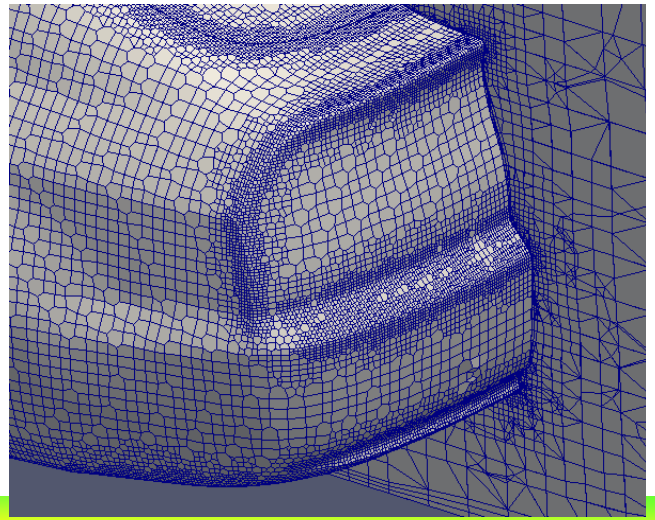


簡略モデルのメッシュ作成（事例１）

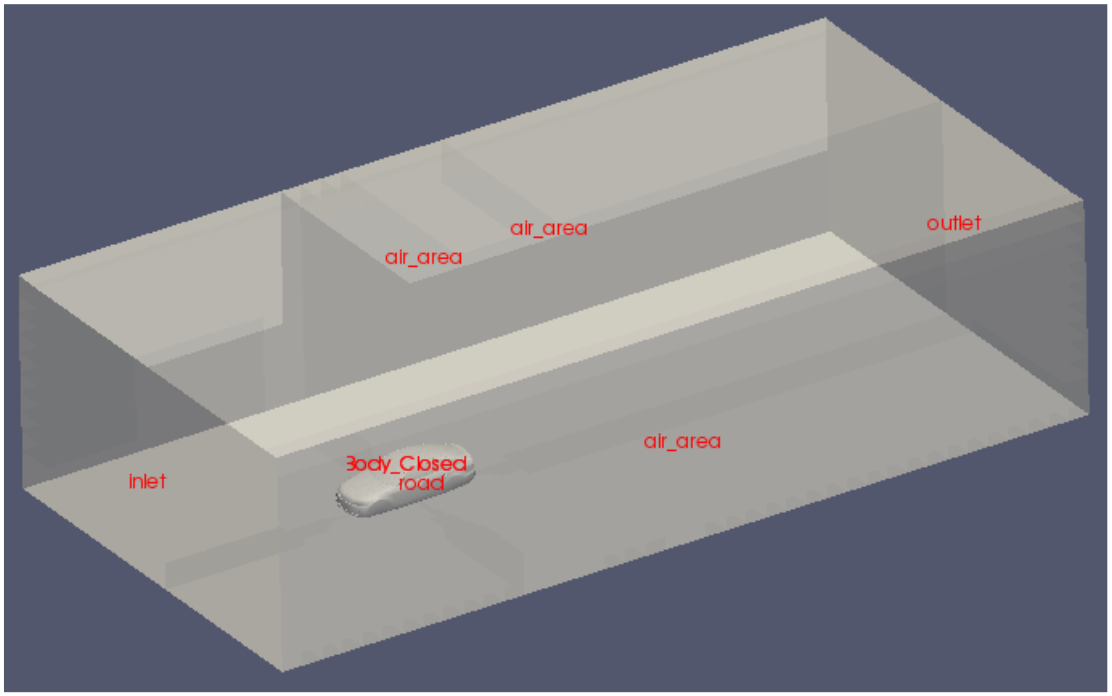


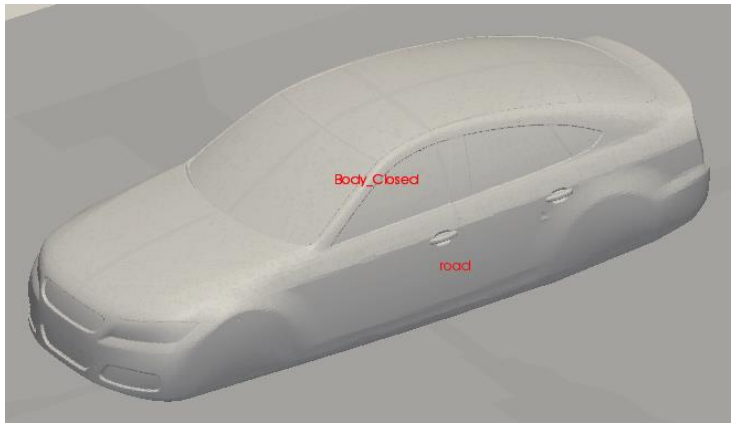


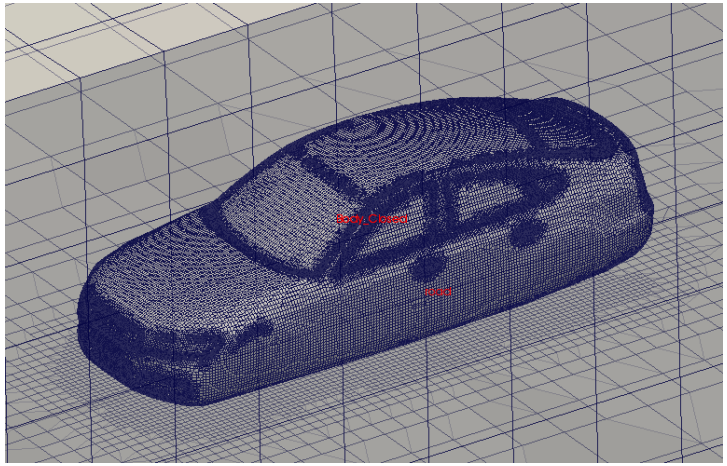




簡略モデルのメッシュ作成（事例１）








```
Time = 3

Mesh stats
    points:           1122086
    faces:            2902231
    internal faces:   2762887
    cells:            899741
    faces per cell:   6.29639
    boundary patches: 5
    point zones:      0
    face zones:       0
    cell zones:       0
```

簡略モデルのcheckMesh（事例１）

```
Time = 3

Mesh stats
    points:           1122086
    faces:            2902231
    internal faces:   2762887
    cells:            899741
    faces per cell:   6.29639
    boundary patches: 5
    point zones:      0
    face zones:       0
    cell zones:       0

Overall number of cells of each type:
    hexahedra:     708729
    prisms:        23921
    wedges:        1
    pyramids:      0
    tet wedges:    1121
    tetrahedra:    13
    polyhedra:     165956
    Breakdown of polyhedra by number of faces:
        faces   number of cells
            4   12717
            5   10524
            6   30145
            7   37736
            8   14166
            9   40064
           10   1232
           11   311
           12   12228
           13   83
           14   137
           15   6099
           16   8
           17   12
           18   494

Checking topology...
    Boundary definition OK.
    Cell to face addressing OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Topological cell zip-up check OK.
  <<Number of duplicate (not baffle) faces found: 2. This might indicate a problem.
  <<Number of faces with non-consecutive shared points: 3. This might indicate a problem.
  <<Writing 6 faces with non-standard edge connectivity to set edgeFaces
    Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...
                Patch     Faces    Points  Surface topology Bounding box
                inlet       260       294  ok (non-closed singly connected) (-10.96 -10.4
               outlet       260       294  ok (non-closed singly connected) (30 -10.48 -0
                 road      5573      5801  ok (non-closed singly connected) (-10.96 -10.4
             air_area      1840      1927  ok (non-closed singly connected) (-10.96 -10.4
          Body_Closed    131411    150729       ok (closed singly connected) (-1.09641 -1.
```

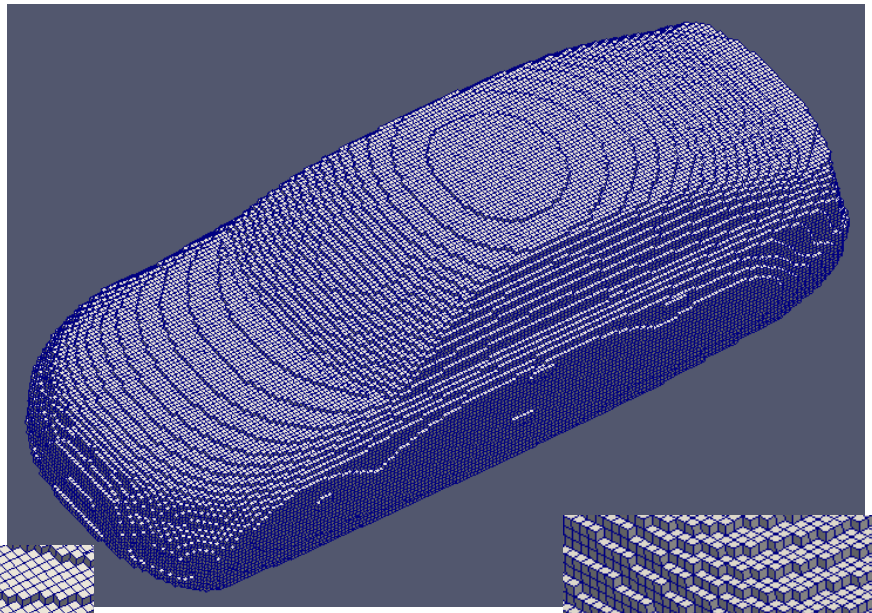
簡略モデルのcheckMesh（事例１）

```
Checking geometry...
    Overall domain bounding box (-10.96 -10.48 -0.432) (30 10 9.84806)
    Mesh (non-empty, non-wedge) directions (1 1 1)
    Mesh (non-empty) directions (1 1 1)
    Boundary openness (-2.57807e-16 -3.10263e-16 1.31201e-15) OK.
    Max cell openness = 6.89112e-16 OK.
    Max aspect ratio = 21.6582 OK.
    Minimum face area = 3.53392e-07. Maximum face area = 1.17123.  Face area magnitudes OK.
    Min volume = 4.50903e-09. Max volume = 1.12567.  Total volume = 8572.38.  Cell volumes OK.
    Mesh non-orthogonality Max: 64.9803 average: 11.1545
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 3.56637 OK.
    Coupled point location match (average 0) OK.
 ***Error in face tets: 2 faces with low quality or negative volume decomposition tets.
  <<Writing 2 faces with low quality or negative volume decomposition tets to set lowQualityTetFaces
    Min/max edge length = 0.00013855 1.15131 OK.
   *There are 8768 faces with concave angles between consecutive edges. Max concave angle = 79.5161 degrees.
  <<Writing 8768 faces with concave angles to set concaveFaces
    Face flatness (1 = flat, 0 = butterfly) : min = 0.514259  average = 0.999008
   *There are 41 faces with ratio between projected and actual area < 0.8
    Minimum ratio (minimum flatness, maximum warpage) = 0.514259
  <<Writing 41 warped faces to set warpedFaces
    Cell determinant (wellposedness) : minimum: 0.000291952 average: 14.4636
 ***Cells with small determinant (< 0.001) found, number of cells: 5
  <<Writing 5 under-determined cells to set underdeterminedCells
 ***Concave cells (using face planes) found, number of cells: 60666
  <<Writing 60666 concave cells to set concaveCells
    Face interpolation weight : minimum: 0.023859 average: 0.448975
 ***Faces with small interpolation weight (< 0.05) found, number of faces: 207
  <<Writing 207 faces with low interpolation weights to set lowWeightFaces
    Face volume ratio : minimum: 0.0107778 average: 0.772307
    Face volume ratio check OK.

Failed 4 mesh checks.

End
```

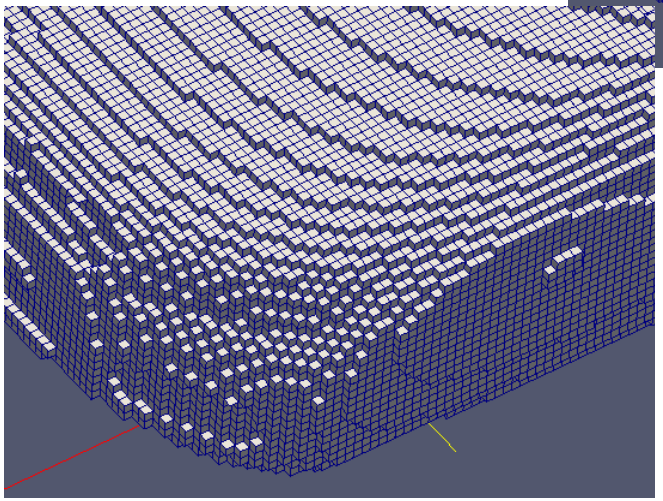
簡略モデルのメッシュ作成（事例2）

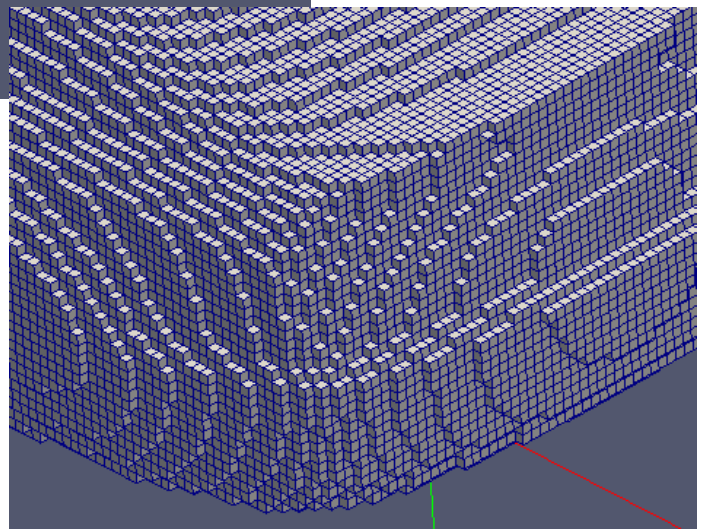
```
refinementSurfaces
{
    Body_Closed
    {
        level       (5 5);
    }
}
```

この値を同じにした場合

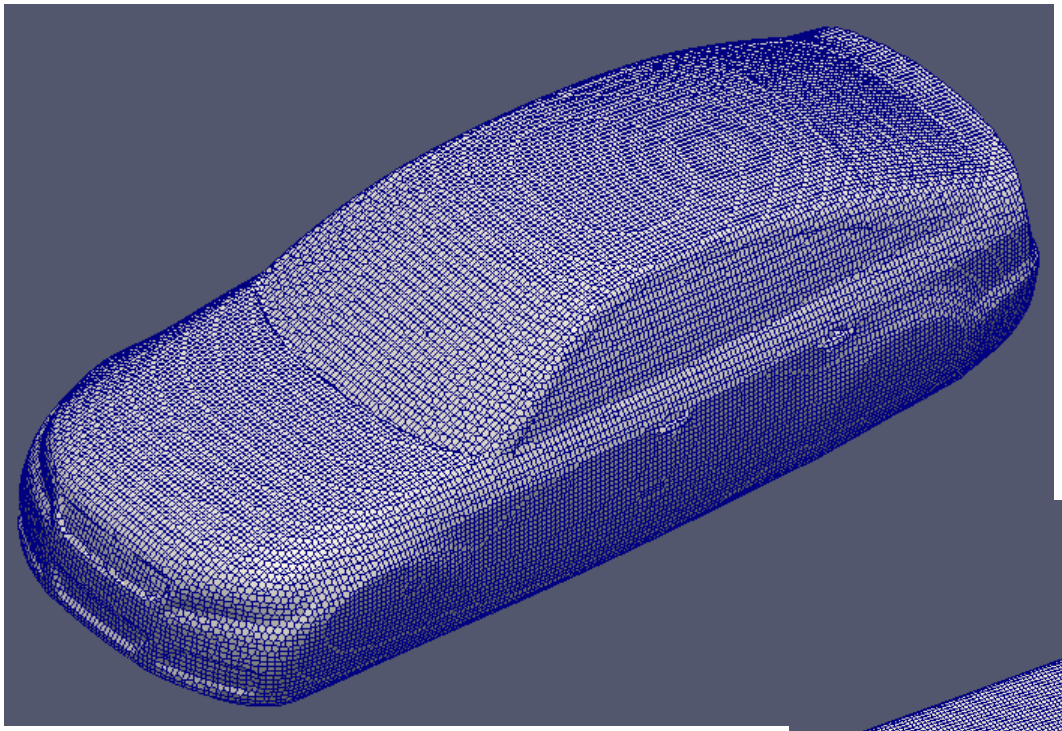


スナップ前のメッシュ

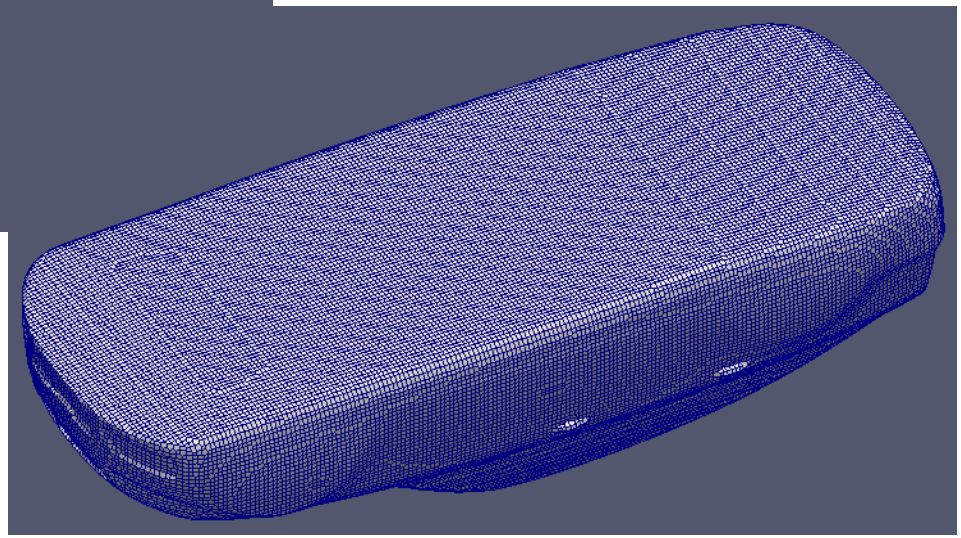




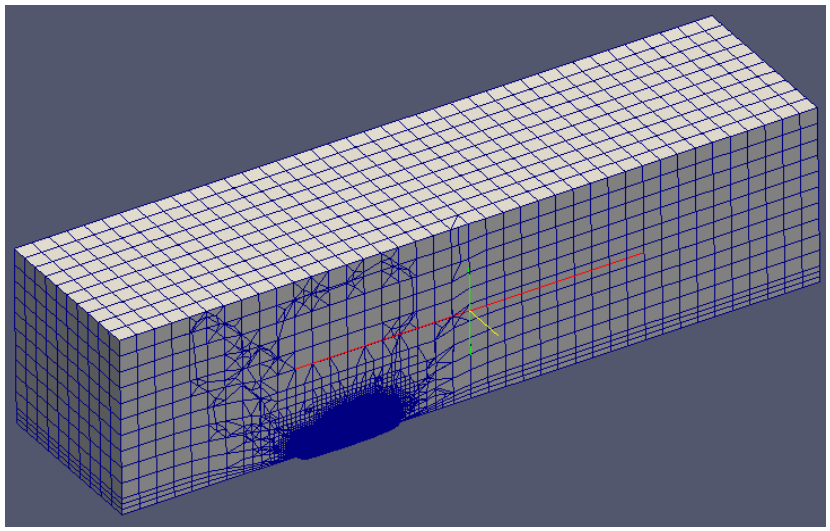
簡略モデルのメッシュ作成（事例2）

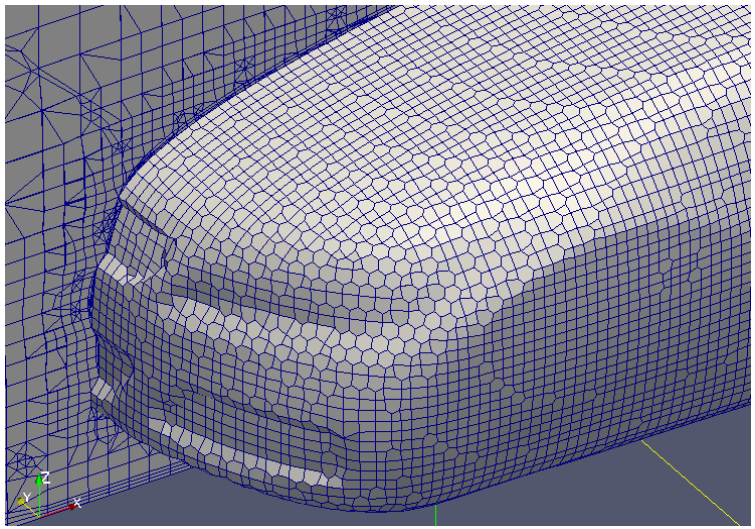


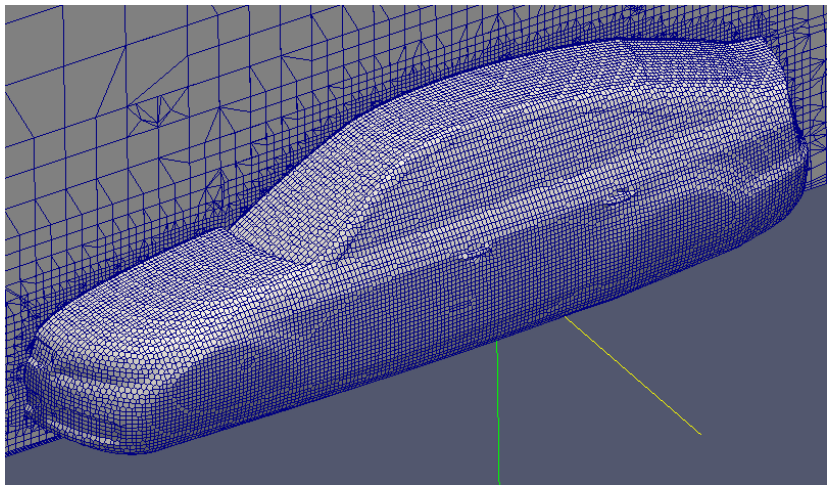
スナップ後
のメッシュ

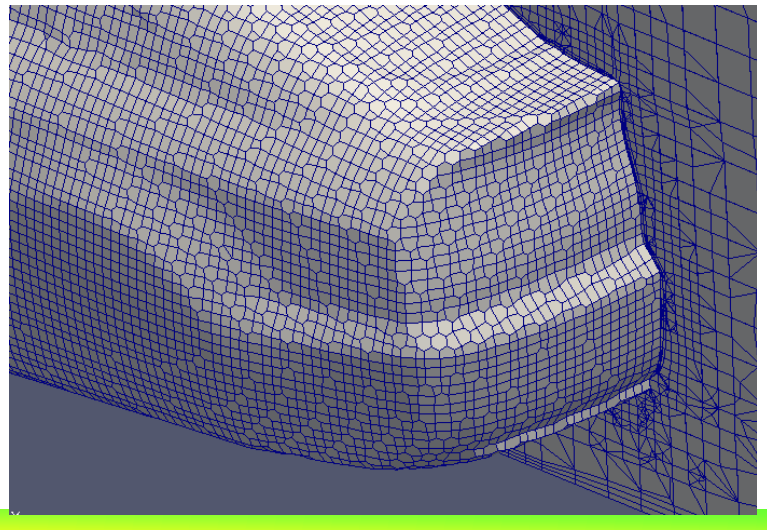


簡略モデルのメッシュ作成（事例2）

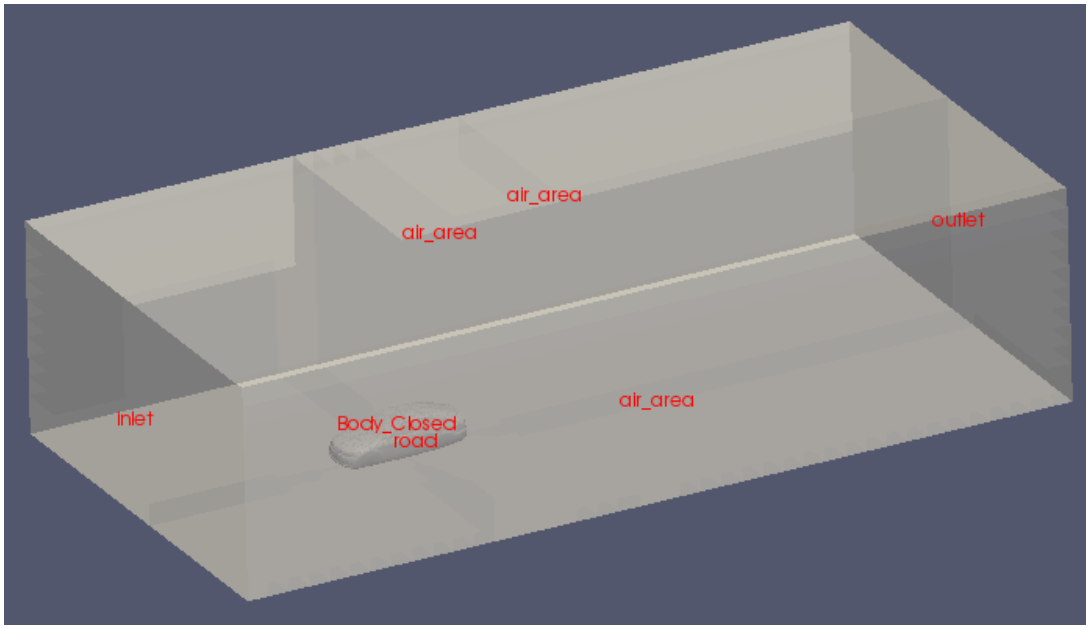


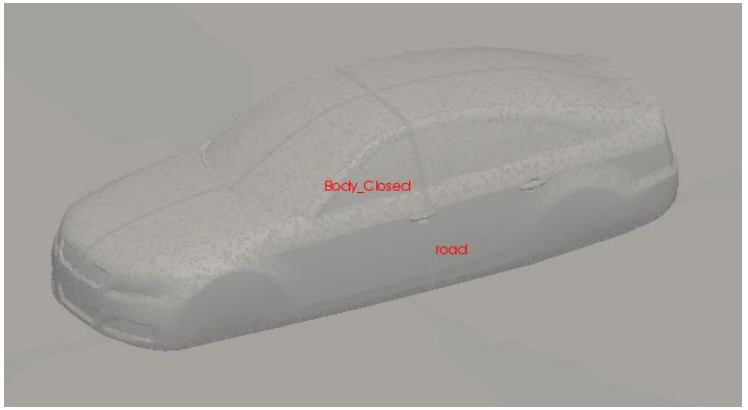


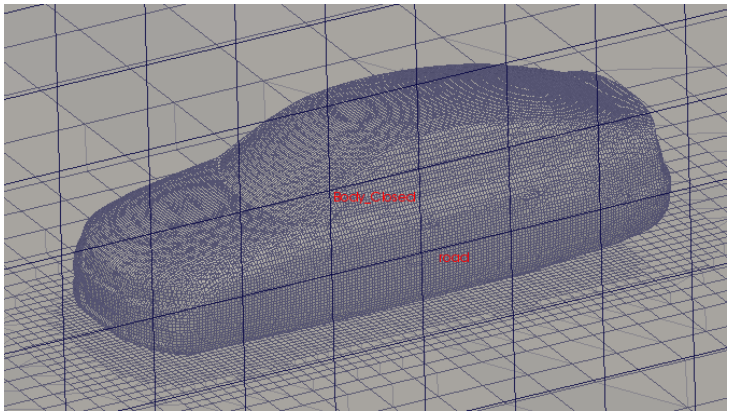




簡略モデルのメッシュ作成（事例2）








```
Time = 3

Mesh stats
    points:           362089
    faces:            969446
    internal faces:   921125
    cells:            304842
    faces per cell:   6.20181
    boundary patches: 5
    point zones:      0
    face zones:       0
    cell zones:       0
```

簡略モデルのcheckMesh（事例2）

```
Time = 3

Mesh stats
    points:           362089
    faces:            969446
    internal faces:   921125
    cells:            304842
    faces per cell:   6.20181
    boundary patches: 5
    point zones:      0
    face zones:       0
    cell zones:       0

Overall number of cells of each type:
    hexahedra:     267066
    prisms:        5118
    wedges:        0
    pyramids:      0
    tet wedges:    40
    tetrahedra:    0
    polyhedra:     32618
    Breakdown of polyhedra by number of faces:
        faces   number of cells
            4   2206
            5   1825
            6   5178
            7   6217
            8   1567
            9   11390
           10   48
           12   2871
           15   1259
           18   57

Checking topology...
    Boundary definition OK.
    Cell to face addressing OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Topological cell zip-up check OK.
  <<Number of faces with non-consecutive shared points: 1. This might indicate a problem.
  <<Writing 2 faces with non-standard edge connectivity to set edgeFaces
    Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...
                          Patch     Faces   Points                  Surface topology Bounding box
                          inlet       260      294  ok (non-closed singly connected) (-10.96 -10.
                         outlet       260      294  ok (non-closed singly connected) (30 -10.48 -
                           road      5324     5541  ok (non-closed singly connected) (-10.96 -10.
                       air_area      1840     1927  ok (non-closed singly connected) (-10.96 -10.
                    Body_Closed     40637    42928       ok (closed singly connected) (-1.09636 -1
1.49275)
```

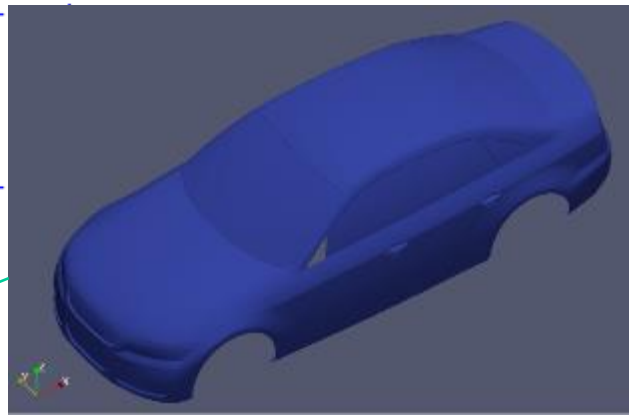
簡略モデルのcheckMesh（事例2）

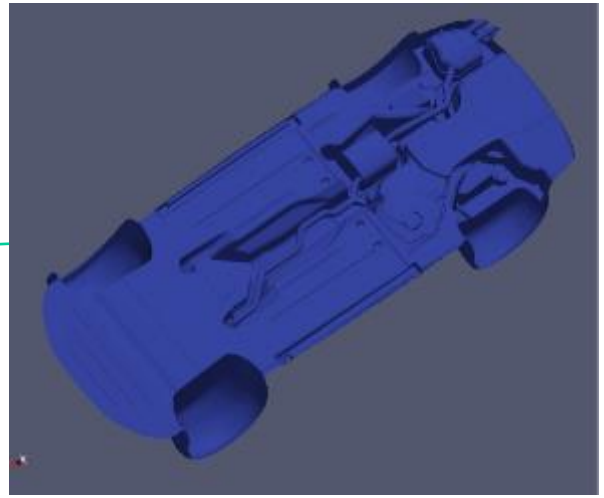
```
Checking geometry...
    Overall domain bounding box (-10.96 -10.48 -0.432) (30 10 9.84806)
    Mesh (non-empty, non-wedge) directions (1 1 1)
    Mesh (non-empty) directions (1 1 1)
    Boundary openness (-2.59764e-16 -1.00296e-16 1.7122e-15) OK.
    Max cell openness = 5.34706e-16 OK.
    Max aspect ratio = 17.8252 OK.
    Minimum face area = 3.31135e-05. Maximum face area = 1.19854.  Face area magnitudes OK.
    Min volume = 3.06288e-07. Max volume = 1.13568.  Total volume = 8572.38.  Cell volumes OK.
    Mesh non-orthogonality Max: 64.607 average: 9.52466
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 2.86913 OK.
    Coupled point location match (average 0) OK.
    Face tets OK.
    Min/max edge length = 0.00160047 1.17301 OK.
  *There are 1552 faces with concave angles between consecutive edges. Max concave angle = 78.9284 degrees.
 <<Writing 1552 faces with concave angles to set concaveFaces
    Face flatness (1 = flat, 0 = butterfly) : min = 0.811913  average = 0.999519
    All face flatness OK.
    Cell determinant (wellposedness) : minimum: 0.000220362 average: 11.8772
***Cells with small determinant (< 0.001) found, number of cells: 1
 <<Writing 1 under-determined cells to set underdeterminedCells
***Concave cells (using face planes) found, number of cells: 14994
 <<Writing 14994 concave cells to set concaveCells
    Face interpolation weight : minimum: 0.0273295 average: 0.462649
***Faces with small interpolation weight (< 0.05) found, number of faces: 32
 <<Writing 32 faces with low interpolation weights to set lowWeightFaces
    Face volume ratio : minimum: 0.0164779 average: 0.835756
    Face volume ratio check OK.

Failed 3 mesh checks.

End
```

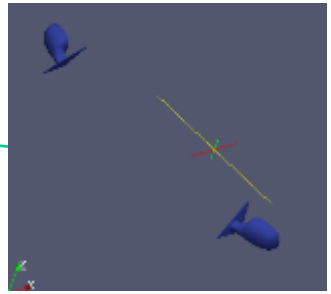
詳細モデルのメッシュ作成

詳細モデルのメッシュ作成（sHM）

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      snappyHexMeshDict;
}
castellatedMesh true;
snap            true;
addLayers       true;
geometry
{
    Part_01_Body_Notchback_m.stl
    {
        type        triSurfaceMesh;
        name        Body_01;
    }
    Part_02_UB_Detailed_m.stl
    {
        type        triSurfaceMesh;
        name        UB_02;
    }
    Part_04_Mirror_m.stl
    {
        type        triSurfaceMesh;
        name        Mirror_04;
    }
```







詳細モデルのメッシュ作成（sHM）

①

```
Part_05_tyres_fr_m.stl
{
    type        triSurfaceMesh;
    name        tyres_fr;
}
```

②

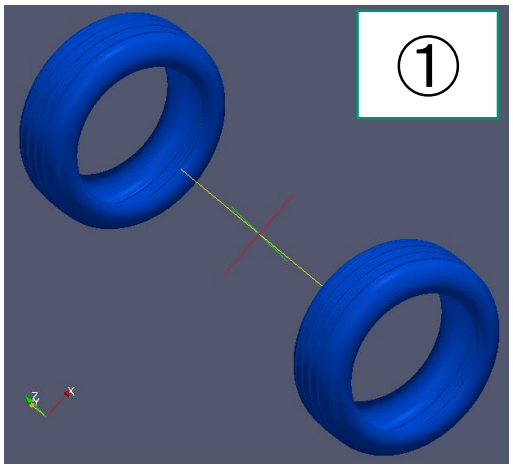
```
Part_05_wheels_fr_m.stl
{
    type        triSurfaceMesh;
    name        wheels_fr;
}
```

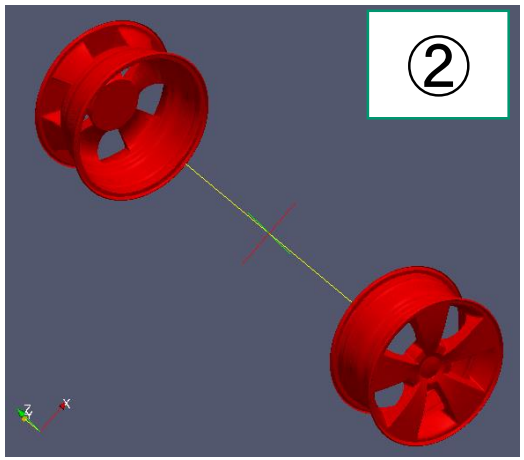
③

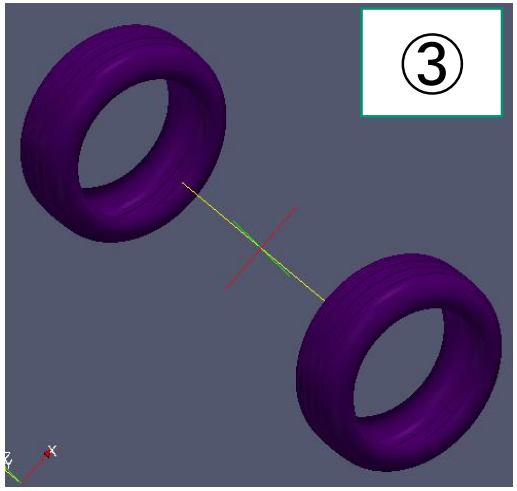
```
Part_06_tyres_rear_m.stl
{
    type        triSurfaceMesh;
    name        tyres_rear;
}
```

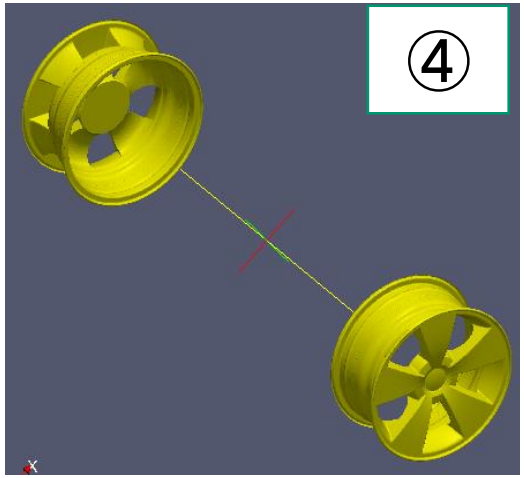
④

```
Part_06_wheels_rear_m.stl
{
    type        triSurfaceMesh;
    name        wheels_rear;
}
```









詳細モデルのメッシュ作成（sHM）

```
refine_area
{
        type searchableBox;
        min (-2.1  -2.24 -0.432);
        max ( 7.16  2.24  2.49);
}
refine_area_2
{
        type searchableBox;
        min (-3.1  -3.24 -0.432);
        max ( 8.16  3.24  3.49);
}
refine_Wheels_Front_01
{
        type searchableCylinder;
        point1 (0.0095 0.859 0.0);
        point2 (0.0095 1.390 0.0);
        radius 0.46;
}
refine_Wheels_Front_02
{
        type searchableCylinder;
        point1 (0.0095 -1.390 0.0);
        point2 (0.0095 -0.859 0.0);
        radius 0.46;
}
```

自動車を囲む領域のメッシュ密度を設定するために設定

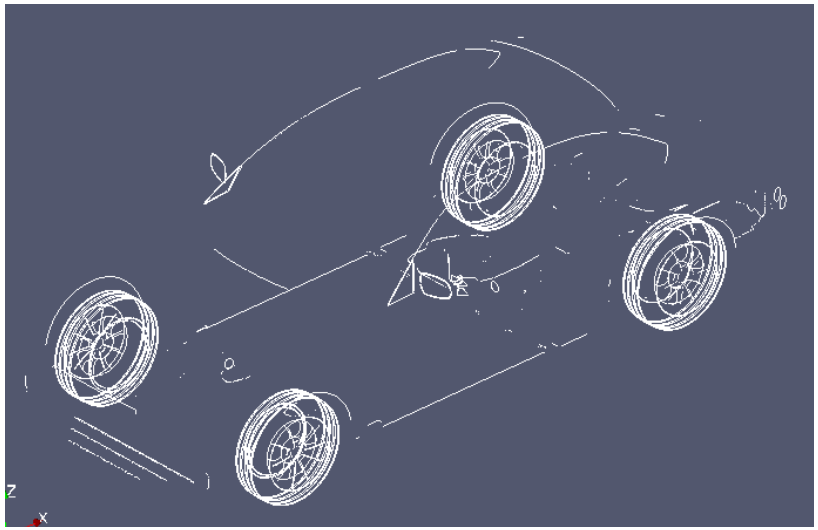
```
    refine_Wheels_Rear_01
    {
        type searchableCylinder;
        point1 (3.79 0.863 0.0);
        point2 (3.79 1.390 0.0);
        radius 0.46;
    }
    refine_Wheels_Rear_02
    {
        type searchableCylinder;
        point1 (3.79 -1.390 0.0);
        point2 (3.79 -0.863 0.0);
        radius 0.46;
    }
};
```

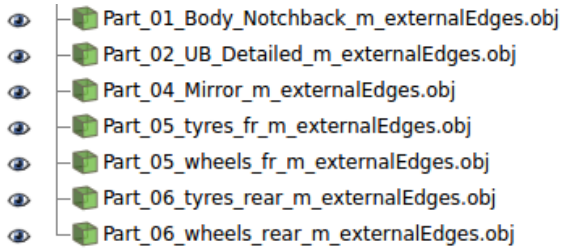
この設定は必修でない

タイヤ，ホィールを囲む領域のメッシュ密度を設定するために設定

詳細モデルのメッシュ作成（sHM）

```
castellatedMeshControls
{
    maxLocalCells 2000000;
    maxGlobalCells 8000000;
    minRefinementCells 10;
    maxLoadUnbalance 0.10;
    nCellsBetweenLevels 3;
    features
    (
        {
            file        "Part_01_Body_Notchback_m.eMesh";
            level       5;
        }
        {
            file        "Part_02_UB_Detailed_m.eMesh";
            level       5;
        }
        {
            file        "Part_04_Mirror_m.eMesh";
            level       6;
        }
        {
            file        "Part_05_tyres_fr_m.eMesh";
            level       7;
        }
        {
            file        "Part_05_wheels_fr_m.eMesh";
            level       7;
        }
        {
            file        "Part_06_tyres_rear_m.eMesh";
            level       7;
        }
        {
            file        "Part_06_wheels_rear_m.eMesh";
            level       7;
        }
    );
```





詳細モデルのメッシュ作成（sHM）

```
refinementSurfaces
{
    Body_01
    {
        level       (5 7);
        patchInfo
        {
            type wall;
            inGroups (carGroup);
        }
    }
    UB_02
    {
        level       (5 7);
        patchInfo
        {
            type wall;
            inGroups (carGroup);
        }
    }
    Mirror_04
    {
        level       (5 7);
        patchInfo
        {
            type wall;
            inGroups (carGroup);
        }
    }
    tyres_fr
    {
        level       (5 7);
        patchInfo
        {
            type wall;
            inGroups (carGroup);
        }
    }
```

車のモデル部分はcarGroupとしてまとめておく。
これは無くてもよいが，表面近傍の流線を書き出す場合に利用している。

詳細モデルのメッシュ作成（sHM）

```
    wheels_fr
    {
        level       (5 7);
        patchInfo
        {
            type wall;
            inGroups (carGroup);
        }
    }
    tyres_rear
    {
        level       (5 7);
        patchInfo
        {
            type wall;
            inGroups (carGroup);
        }
    }
    wheels_rear
    {
        level       (5 7);
        patchInfo
        {
            type wall;
            inGroups (carGroup);
        }
    }
}
resolveFeatureAngle 30;
```

車のモデル部分はcarGroupとしてまとめておく。
これは無くてもよいが，表面近傍の流線を書き出す場合に利用している。

詳細モデルのメッシュ作成（sHM）

```
    refinementRegions
    {
        refine_area
        {
           mode        inside;
           levels      ((1E15 4));
        }
        refine_area_2
        {
           mode        inside;
           levels      ((1E15 2));
        }
        refine_Wheels_Front_01
        {
           mode        inside;
           levels      ((1E15 5));
        }
        refine_Wheels_Front_02
        {
           mode        inside;
           levels      ((1E15 5));
        }
        refine_Wheels_Rear_01
        {
           mode        inside;
           levels      ((1E15 5));
        }
        refine_Wheels_Rear_02
        {
           mode        inside;
           levels      ((1E15 5));
        }
    }
    locationInMesh (-1.5013 0.0013 0.0013);
    allowFreeStandingZoneFaces true;
}
```

```
snapControls
{
    nSmoothPatch 3;
    tolerance 2.0;
    nSolveIter 300;
    nRelaxIter 5;
    nFeatureSnapIter 10;
    implicitFeatureSnap false;
    explicitFeatureSnap true;
    multiRegionFeatureSnap false;
}
```

メッシュを切り始める初期位置

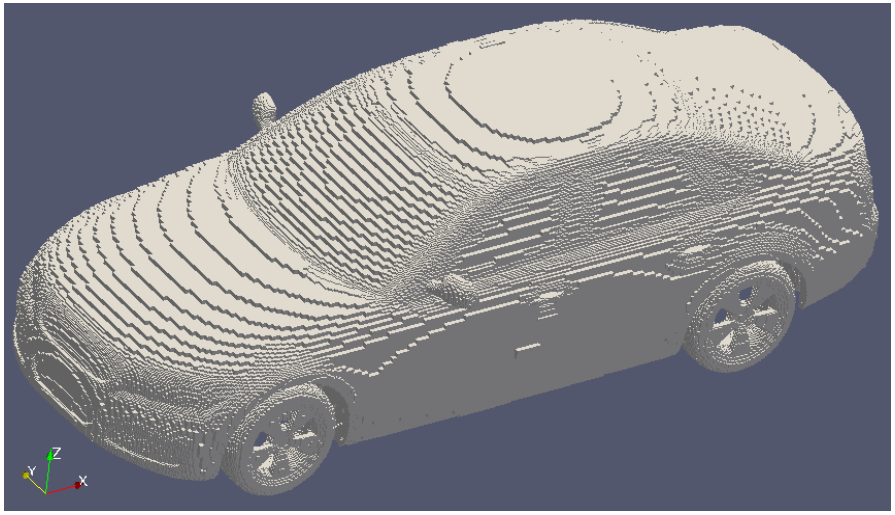
詳細モデルのメッシュ作成（sHM）

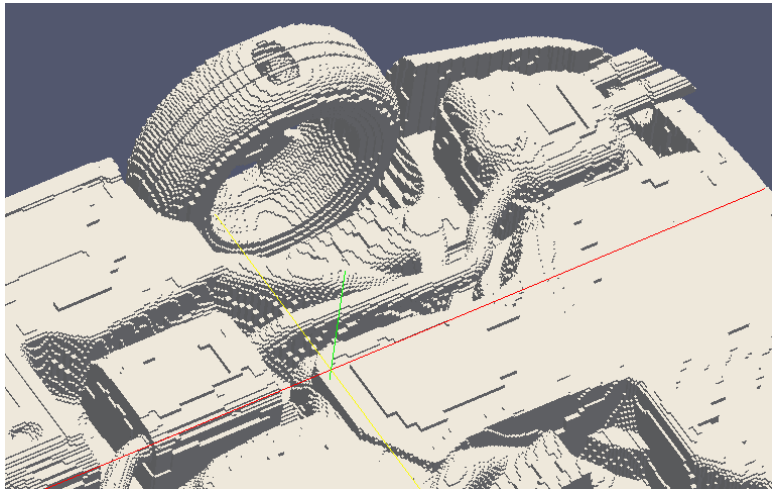
```
addLayersControls
{
    relativeSizes true;
    expansionRatio 1.0;
    finalLayerThickness 0.3;
    minThickness 0.1;
    layers
    {
        Body_01
        {
            nSurfaceLayers 1;
        }
        UB_02
        {
            nSurfaceLayers 1;
        }
        Mirror_04
        {
            nSurfaceLayers 1;
        }
        tyres_fr
        {
            nSurfaceLayers 1;
        }
        wheels_fr
        {
            nSurfaceLayers 1;
        }
        tyres_rear
        {
            nSurfaceLayers 1;
        }
        wheels_rear
        {
            nSurfaceLayers 1;
        }
        road
        {
            nSurfaceLayers 1;
        }
    }
    nGrow 0;
    featureAngle 60;
    maxFaceThicknessRatio 0.5;
    nSmoothSurfaceNormals 1;
    nSmoothThickness 10;
    minMedialAxisAngle 90;
    maxThicknessToMedialRatio 0.3;
    nSmoothNormals 3;
    slipFeatureAngle 30;
    nRelaxIter 3;
    nBufferCellsNoExtrude 0;
    nLayerIter 50;
}
```

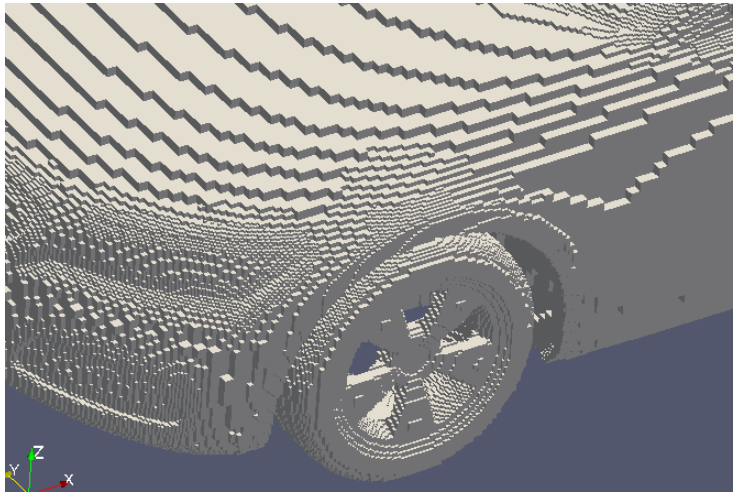
境界層を1層入れる

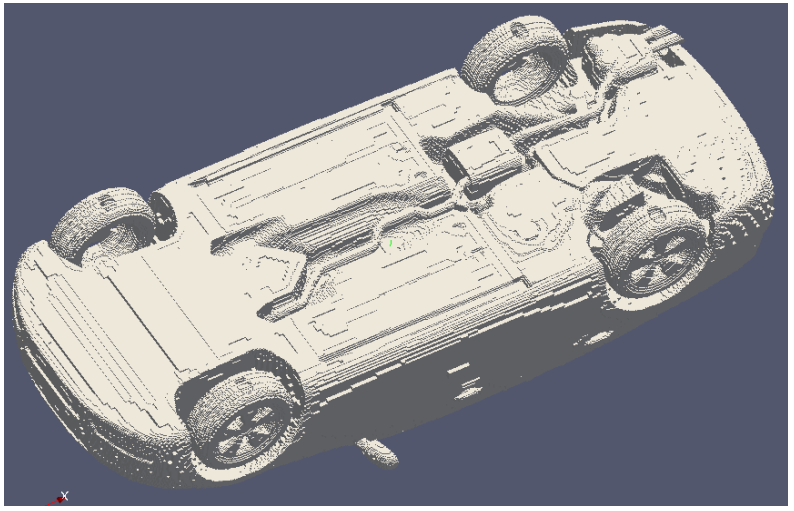
```
meshQualityControls
{
    maxNonOrtho 65;
    maxBoundarySkewness 20;
    maxInternalSkewness 4;
    maxConcave 80;
    minVol 1e-13;
    minTetQuality 1e-8;
    minArea -1;
    minTwist 0.02;
    minDeterminant 0.001;
    minFaceWeight 0.02;
    minVolRatio 0.01;
    minTriangleTwist -1;
    nSmoothScale 4;
    errorReduction 0.75;
}
debug 0;
mergeTolerance 1e-6;
```

詳細モデルのメッシュ作成（事例１）　スナップ前のメッシュ

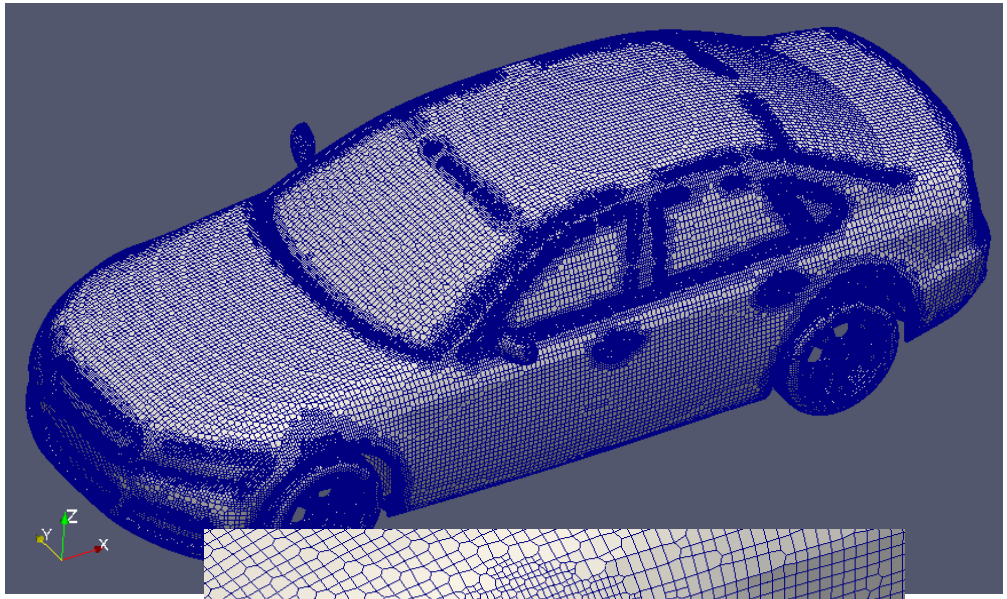


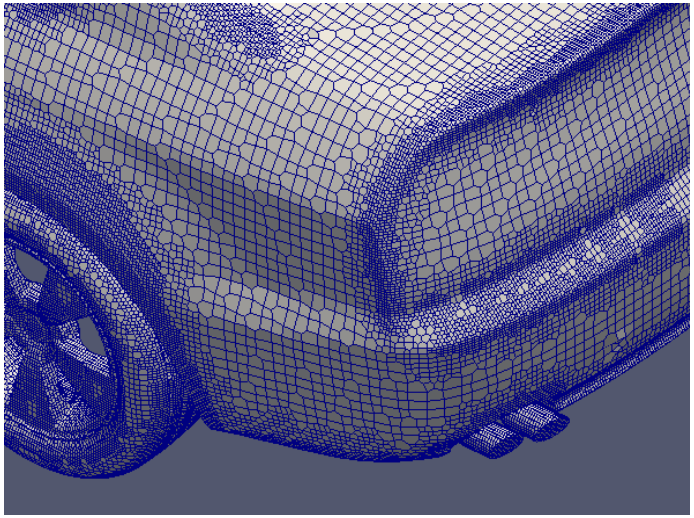


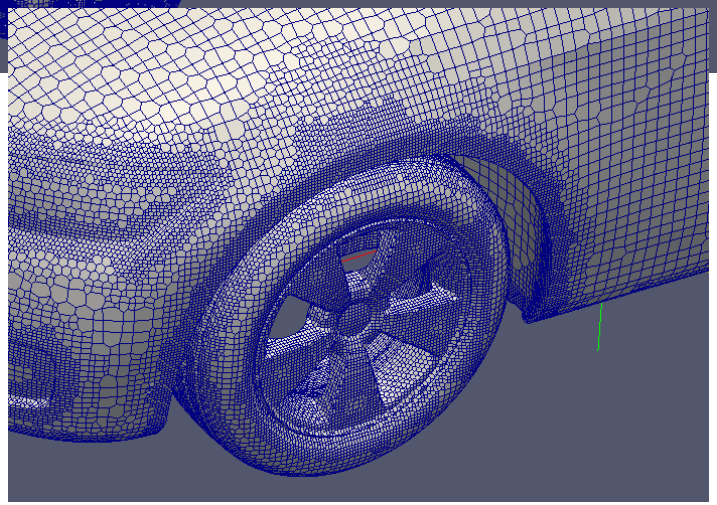


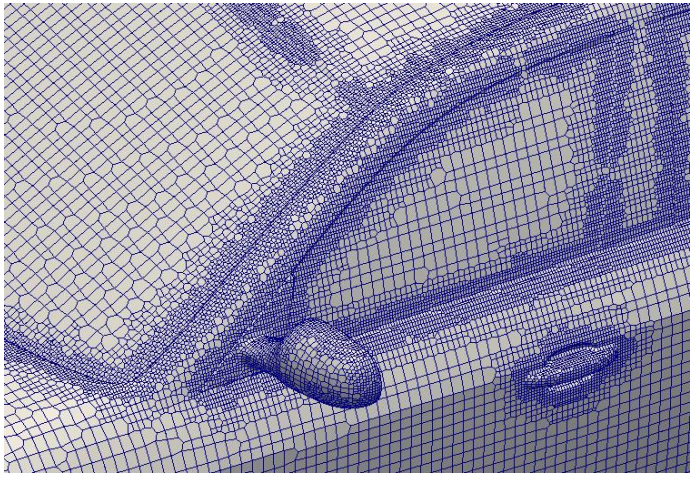


詳細モデルのメッシュ作成（事例１）

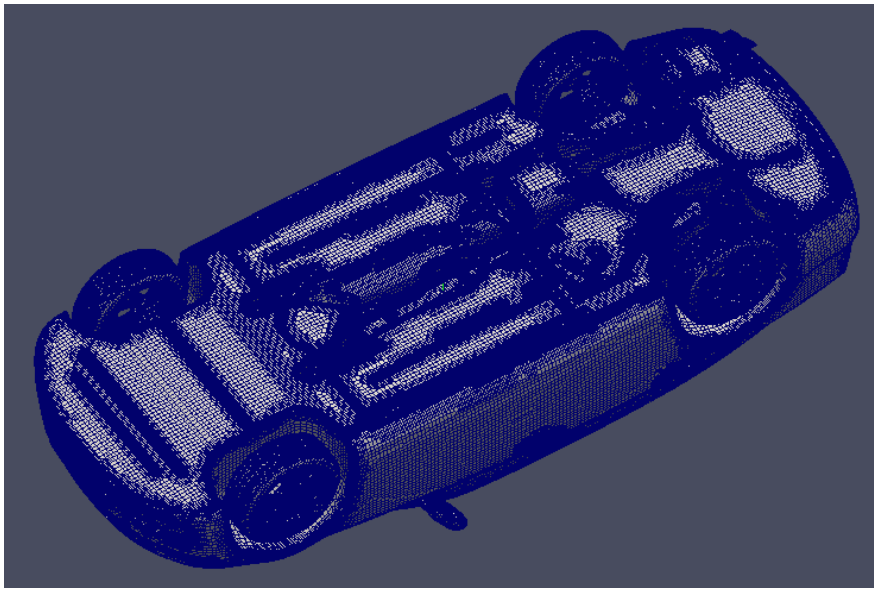


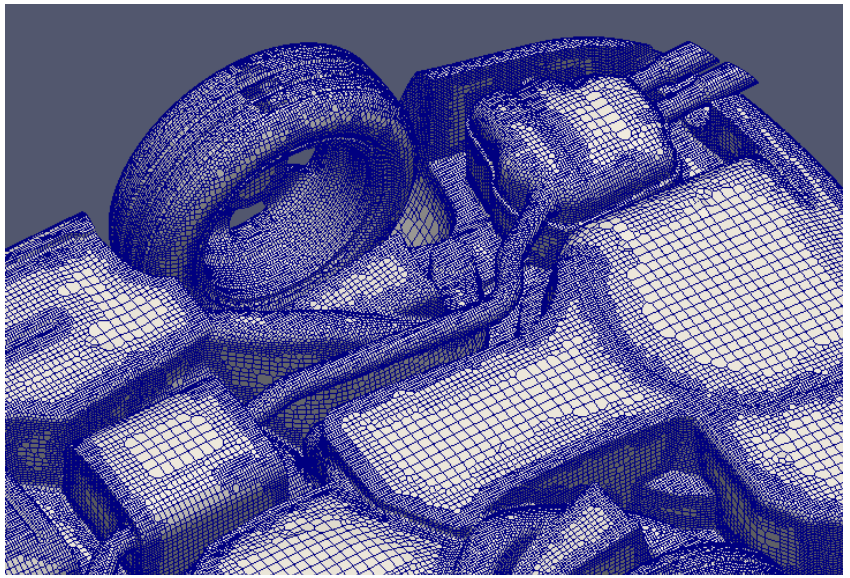


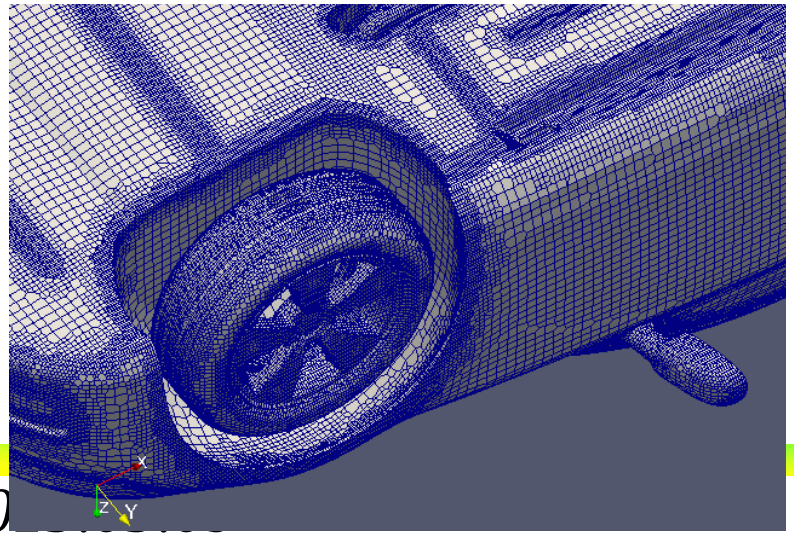


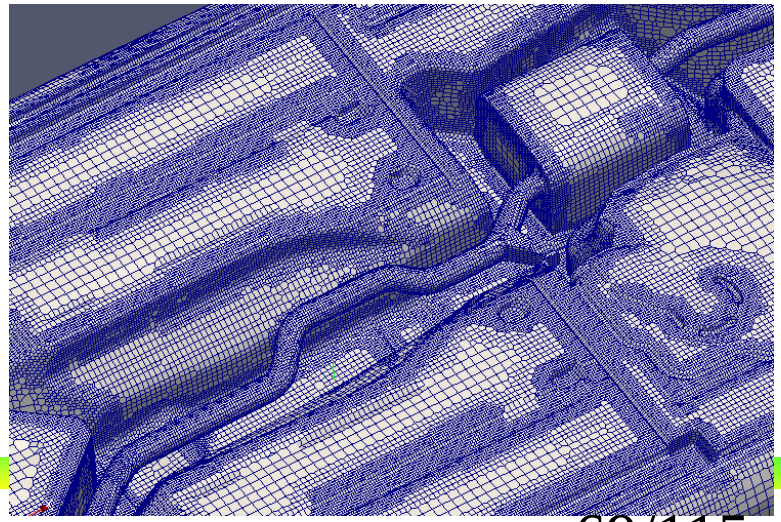


詳細モデルのメッシュ作成（事例１）

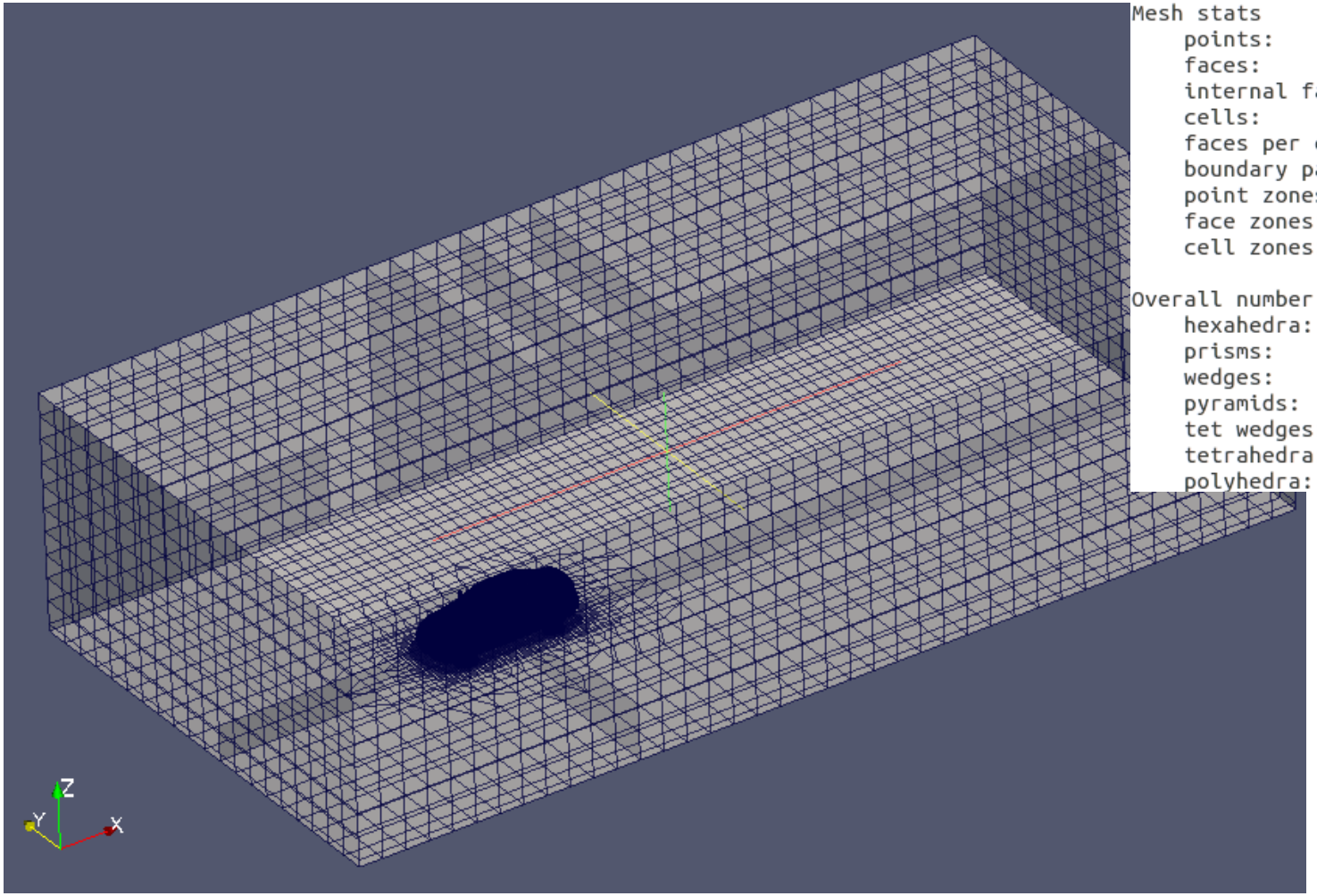






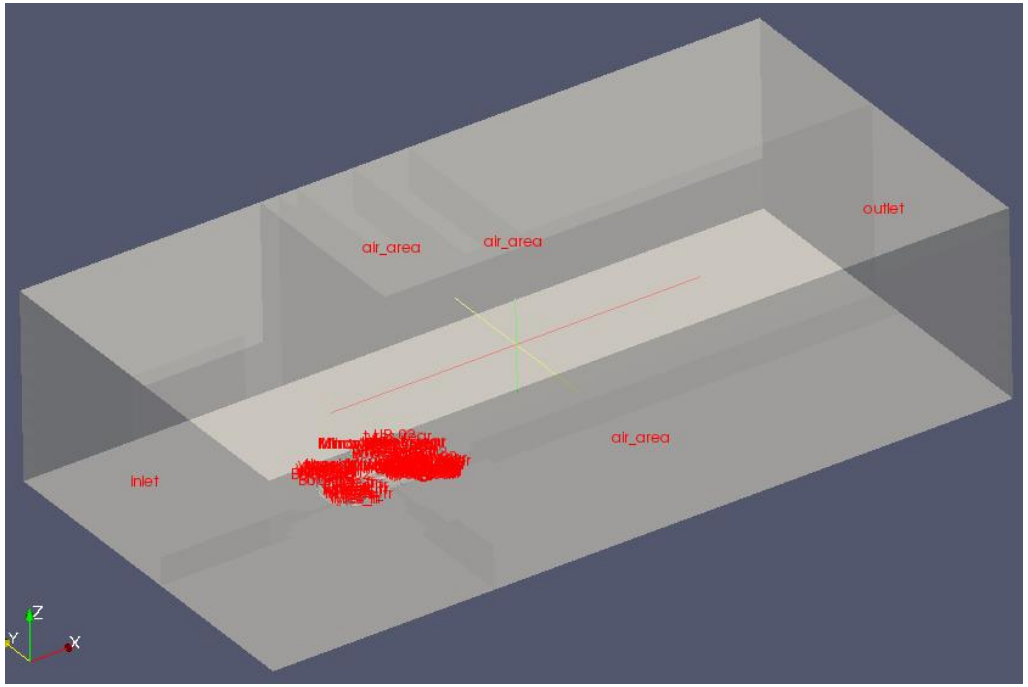


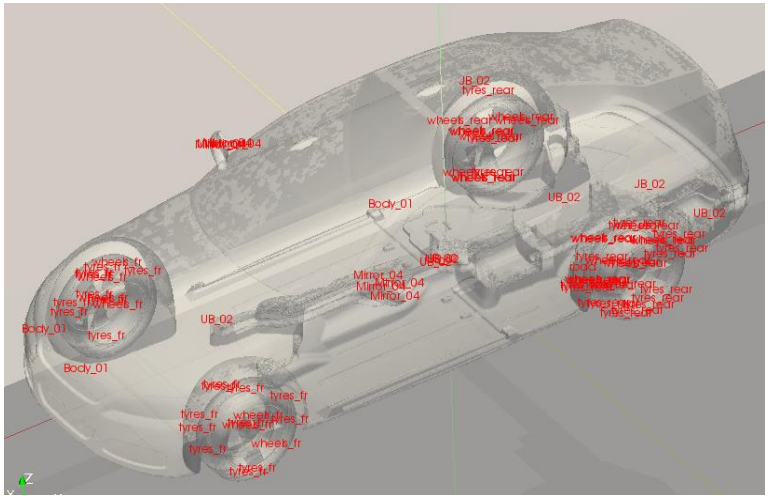
詳細モデルのメッシュ作成（事例１）

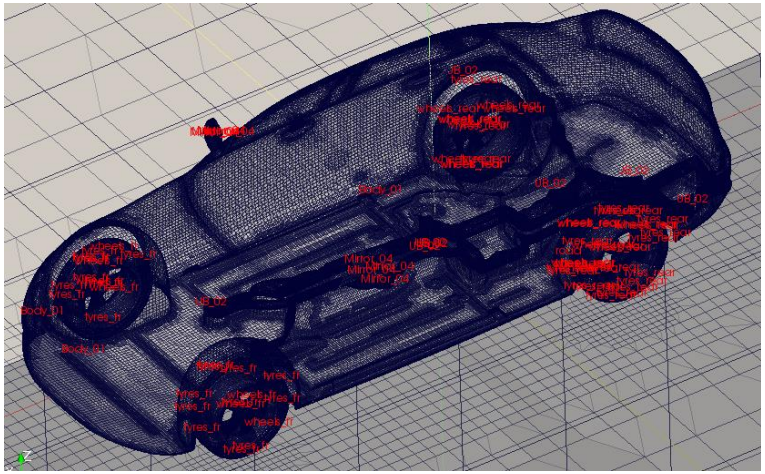



```
Time = 3

Mesh stats
    points:            2964520
    faces:             7695740
    internal faces:    7253808
    cells:             2395130
    faces per cell:    6.24164
    boundary patches:  11
    point zones:       0
    face zones:        0
    cell zones:        0

Overall number of cells of each type:
    hexahedra:      1885782
    prisms:         97653
    wedges:         6342
    pyramids:       1
    tet wedges:     10912
    tetrahedra:     143
    polyhedra:      394297
```

詳細モデルのメッシュ作成（事例１）







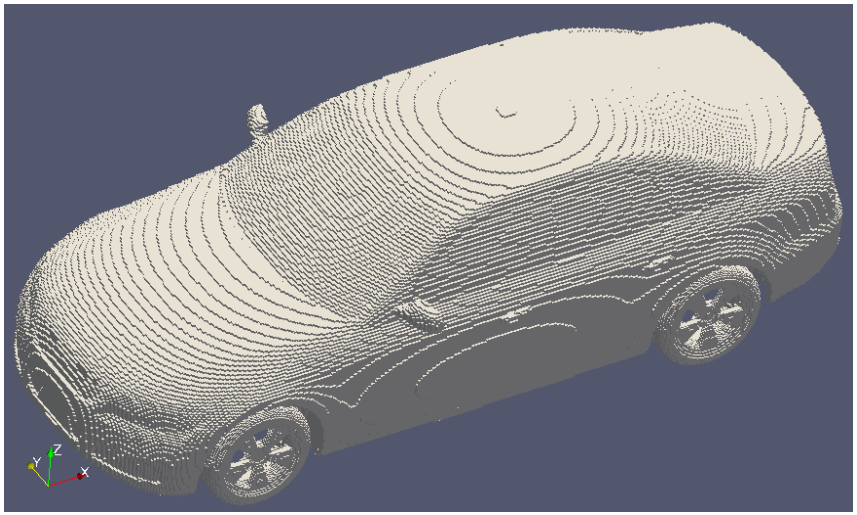
詳細モデルのcheckMesh（事例１）

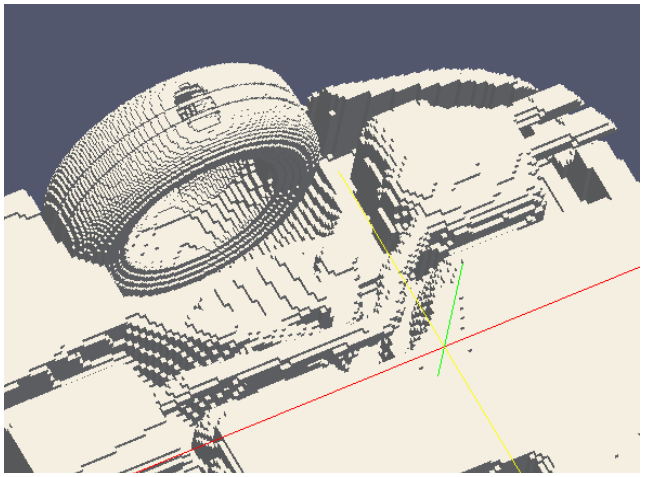
```
Time = 3

Mesh stats
    points:           2964520
    faces:            7695740
    internal faces:   7253808
    cells:            2395130
    faces per cell:   6.24164
    boundary patches: 11
    point zones:      0
    face zones:       0
    cell zones:       0

Overall number of cells of each type:
    hexahedra:     1885782
    prisms:        97653
    wedges:        6342
    pyramids:      1
    tet wedges:    10912
    tetrahedra:    143
    polyhedra:     394297
    Breakdown of polyhedra by number
        faces   number of cells
            4   40198
            5   40462
            6   80759
            7   52670
            8   18284
            9   103749
           10   1354
           11   493
           12   34380
           13   178
           14   231
           15   18911
           16   23
           17   54
           18   2546
           21   5

Checking topology...
    Boundary definition OK.
    Cell to face addressing OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Topological cell zip-up check OK.
  <<Number of duplicate (not baffle) faces found: 2. This might indicate a problem.
  <<Number of faces with non-consecutive shared points: 7. This might indicate a problem.
  <<Writing 16 faces with non-standard edge connectivity to set edgeFaces
  <<Writing 18 cells with two non-boundary faces to set twoInternalFacesCells
    Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...
        Patch    Faces   Points                  Surface topology Bounding box
        inlet      220      252  ok (non-closed singly connected) (-10.96 -10.4
       outlet      220      252  ok (non-closed singly connected) (30 -10.48 -0
         road    12903    13739  ok (non-closed singly connected) (-10.96 -10.4
     air_area     1680     1763  ok (non-closed singly connected) (-10.96 -10.4
      Body_01   123849   145272  ok (non-closed singly connected) (-1.09641 -1.
        UB_02   188859   210650  ok (non-closed singly connected) (-1.08001 -1.
    Mirror_04     5316     6361  ok (non-closed singly connected) (1.01178 -1.3
     tyres_fr    30228    37755  ok (non-closed singly connected) (-0.422741 -1
    wheels_fr    23201    27245  ok (non-closed singly connected) (-0.291982 -1
   tyres_rear    32376    39228  ok (non-closed singly connected) (3.35867 -1.1
  wheels_rear    23080    27261  ok (non-closed singly connected) (3.49076 -1.1
```

詳細モデルのcheckMesh（事例１）

```
Checking geometry...
    Overall domain bounding box (-10.96 -10.48 -0.432) (30 10 9.82177)
    Mesh (non-empty, non-wedge) directions (1 1 1)
    Mesh (non-empty) directions (1 1 1)
    Boundary openness (1.18253e-16 4.5634e-17 -3.32596e-16) OK.
    Max cell openness = 8.7849e-16 OK.
    Max aspect ratio = 35.5591 OK.
    Minimum face area = 3.04374e-07. Maximum face area = 1.07271.  Face area magnitudes OK.
    Min volume = 2.16184e-09. Max volume = 1.08783.  Total volume = 8573.89.  Cell volumes OK.
    Mesh non-orthogonality Max: 64.9988 average: 11.4029
    Non-orthogonality check OK.
    Face pyramids OK.
 ***Max skewness = 4.94077, 3 highly skew faces detected which may impair the quality of the results
  <<Writing 3 skew faces to set skewFaces
    Coupled point location match (average 0) OK.
 ***Error in face tets: 4 faces with low quality or negative volume decomposition tets.
  <<Writing 2 faces with low quality or negative volume decomposition tets to set lowQualityTetFaces
    Min/max edge length = 9.50673e-05 1.04682 OK.
   *There are 12042 faces with concave angles between consecutive edges. Max concave angle = 86.6661 degrees.
  <<Writing 12042 faces with concave angles to set concaveFaces
    Face flatness (1 = flat, 0 = butterfly) : min = 0.105317  average = 0.998682
   *There are 441 faces with ratio between projected and actual area < 0.8
    Minimum ratio (minimum flatness, maximum warpage) = 0.105317
  <<Writing 441 warped faces to set warpedFaces
    Cell determinant (wellposedness) : minimum: 0 average: 15.9885
 ***Cells with small determinant (< 0.001) found, number of cells: 244
  <<Writing 244 under-determined cells to set underdeterminedCells
 ***Concave cells (using face planes) found, number of cells: 138726
  <<Writing 138726 concave cells to set concaveCells
    Face interpolation weight : minimum: 0.0233673 average: 0.447078
 ***Faces with small interpolation weight (< 0.05) found, number of faces: 5585
  <<Writing 5585 faces with low interpolation weights to set lowWeightFaces
    Face volume ratio : minimum: 0.0100356 average: 0.765779
    Face volume ratio check OK.

Failed 5 mesh checks.

End
```

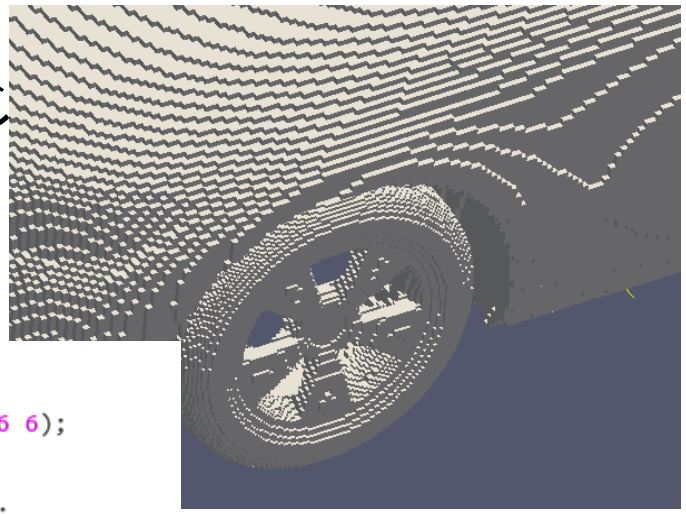
詳細モデルのメッシュ作成（事例2）

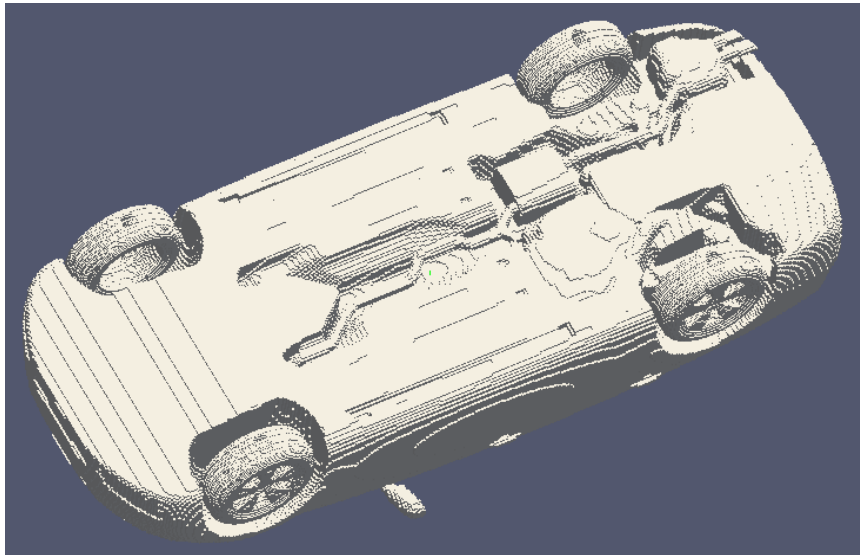




スナップ前のメッシュ

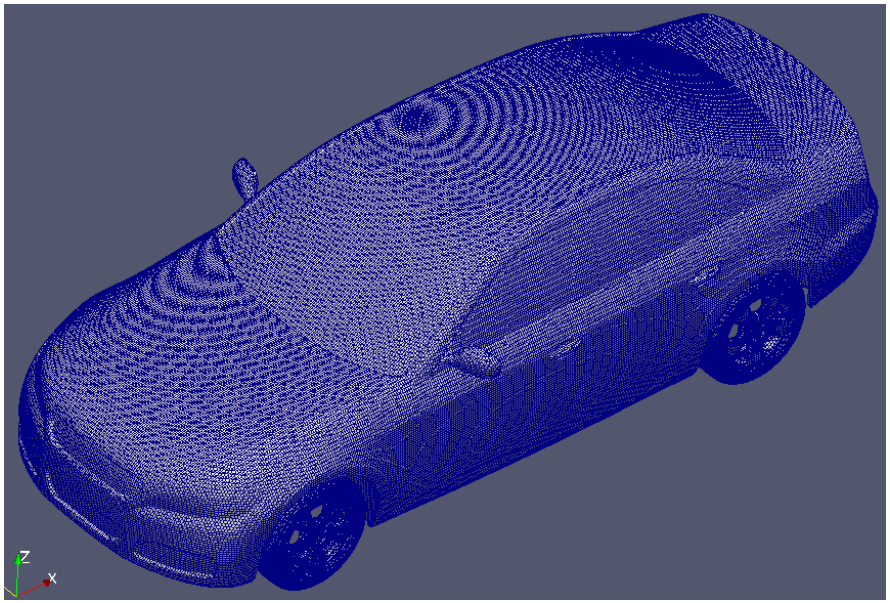
この値を同じにした場合

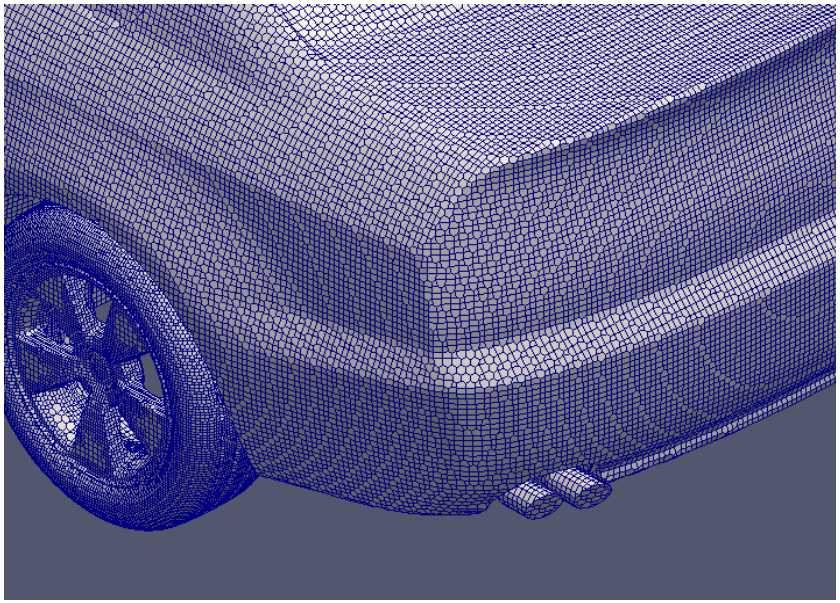


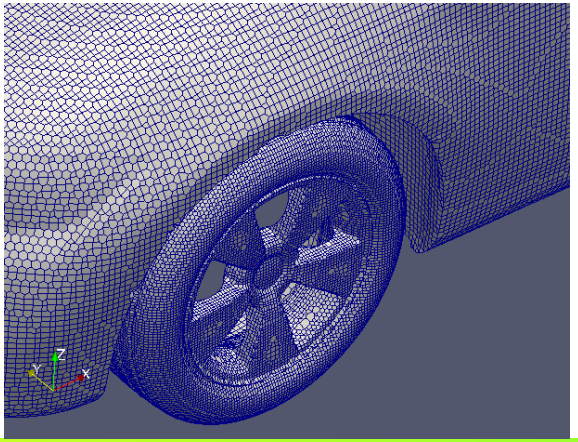


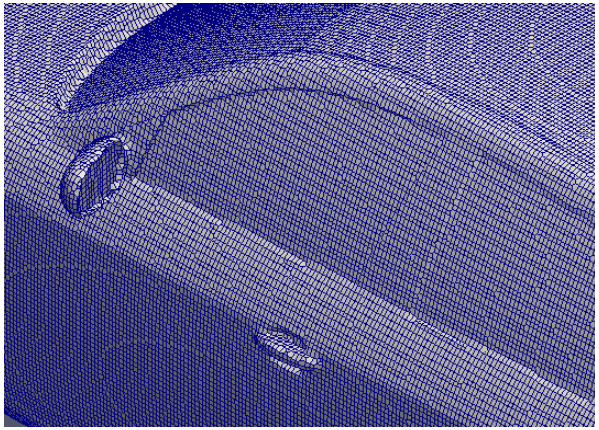
```
Body_01
{
    level       (6 6);
    patchInfo
    {
        type wall;
        inGroups (carGroup);
    }
}
```

詳細モデルのメッシュ作成（事例2）

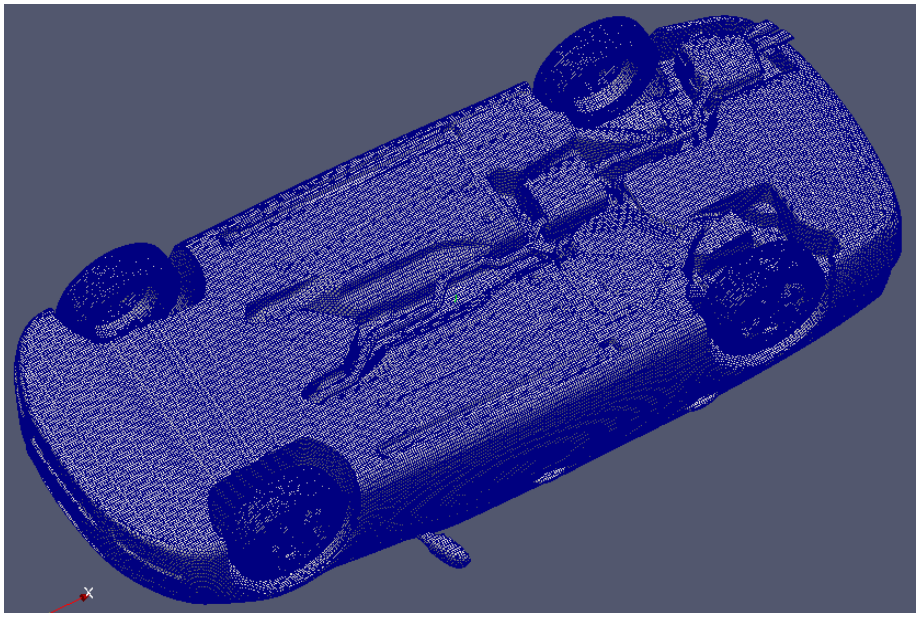


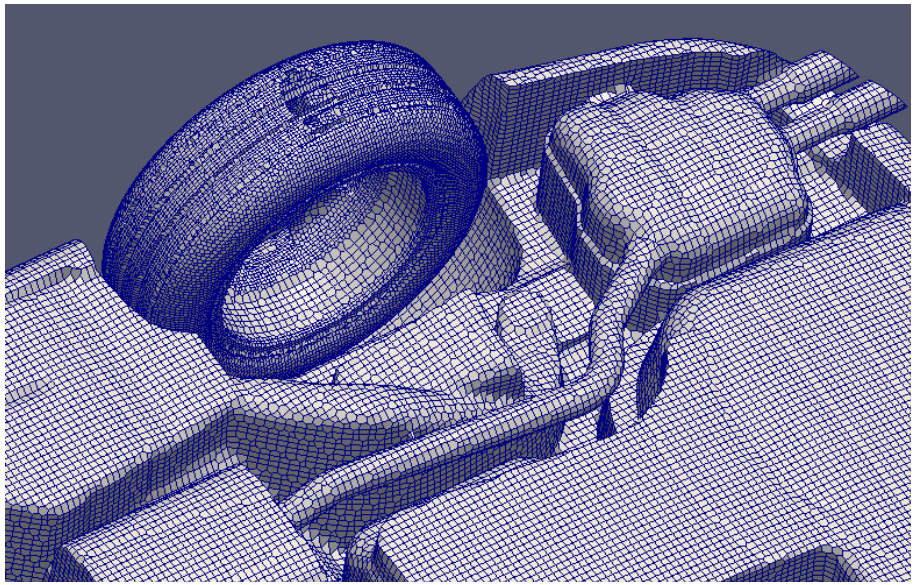


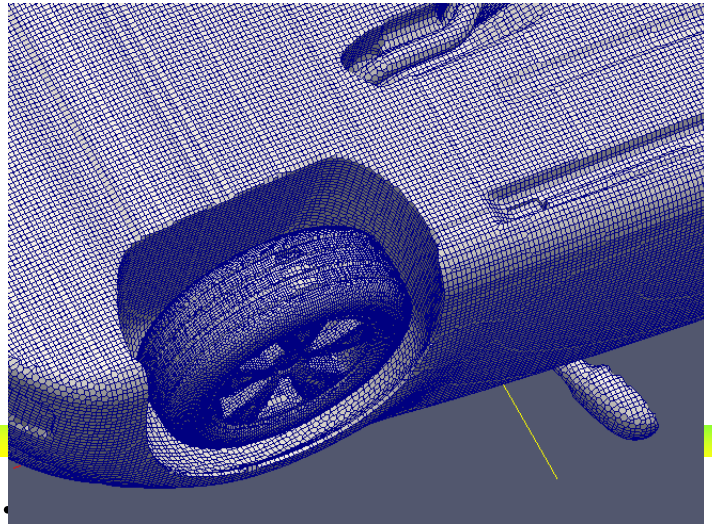


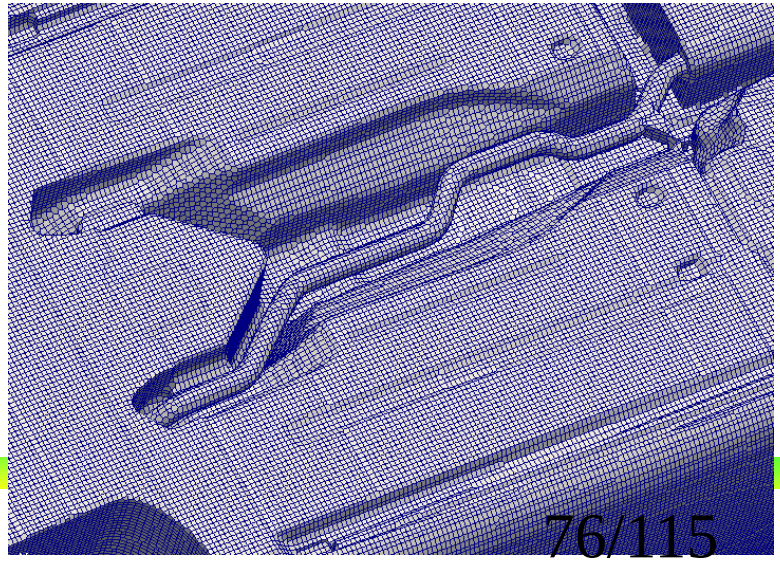


詳細モデルのメッシュ作成（事例2）

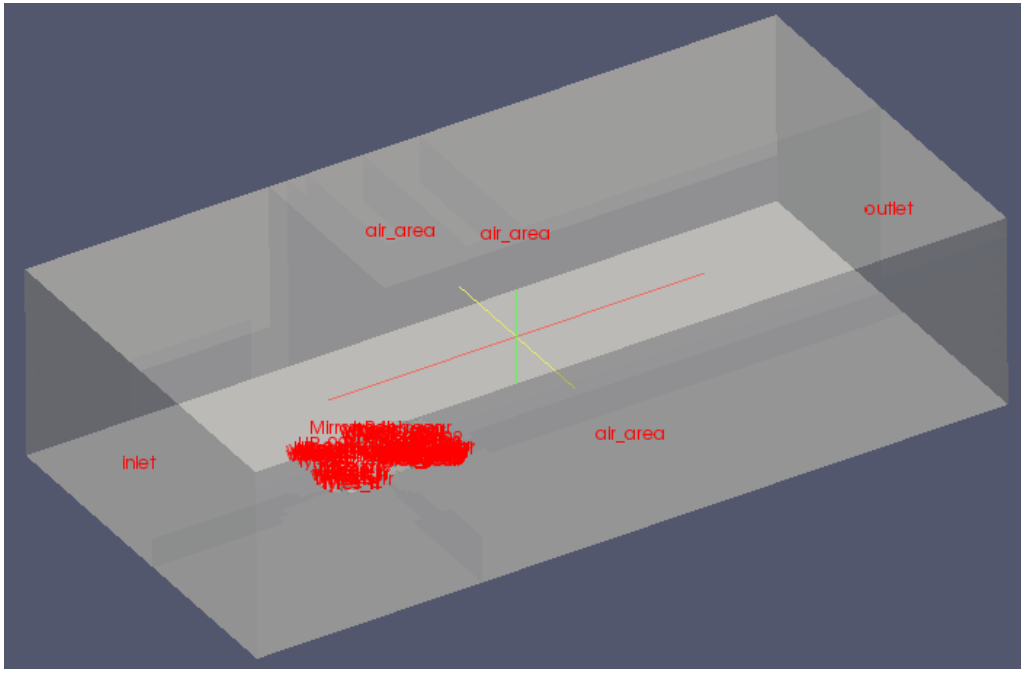


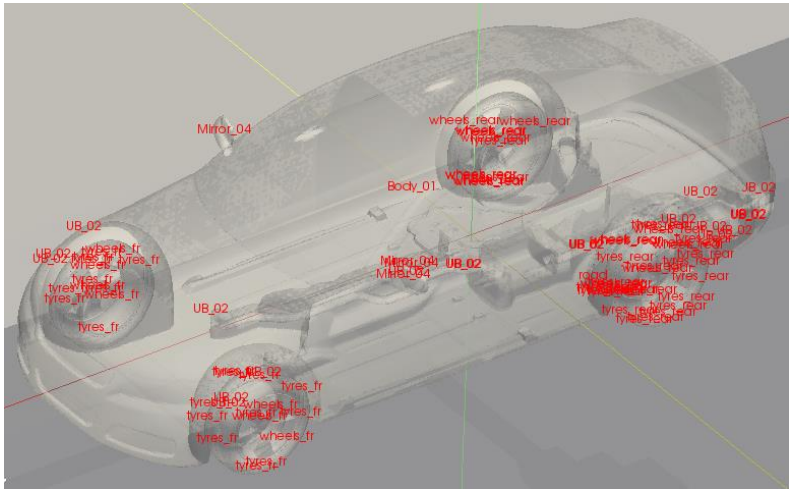


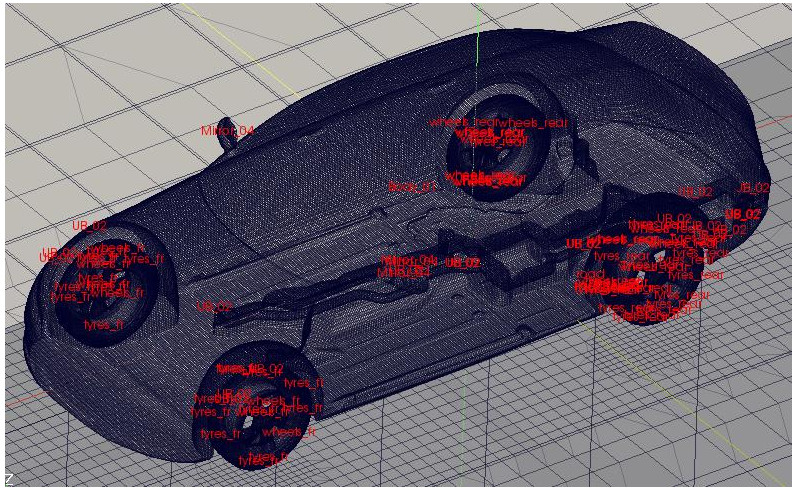




詳細モデルのメッシュ作成（事例2）







詳細モデルのメッシュ作成（事例2）

```
Time = 3

Mesh stats
    points:           2073498
    faces:            5484166
    internal faces:   5170244
    cells:            1721472
    faces per cell:   6.18913
    boundary patches: 11
    point zones:      0
    face zones:       0
    cell zones:       0

Overall number of cells of each type:
    hexahedra:     1419057
    prisms:        59704
    wedges:        4325
    pyramids:      0
    tet wedges:    6856
    tetrahedra:    115
    polyhedra:     231415
    Breakdown of polyhedra by number o
        faces   number of cells
            4   28359
            5   28267
            6   39906
            7   28517
            8   8334
            9   63659
           10   117
           11   37
           12   21487
           13   9
           14   17
           15   11404
           16   1
           17   3
           18   1292
           21   6

Checking topology...
    Boundary definition OK.
    Cell to face addressing OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Topological cell zip-up check OK.
  <<Number of duplicate (not baffle) faces found: 2. This might indicate a problem.
  <<Number of faces with non-consecutive shared points: 15. This might indicate a problem.
  <<Writing 30 faces with non-standard edge connectivity to set edgeFaces
  <<Writing 7 cells with two non-boundary faces to set twoInternalFacesCells
    Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...
                Patch    Faces   Points                  Surface topology Bounding box
                inlet      220      252  ok (non-closed singly connected) (-10.96 -10.4
               outlet      220      252  ok (non-closed singly connected) (30 -10.48 -0
                 road    13353    14176  ok (non-closed singly connected) (-10.96 -10.4
             air_area     1680     1763  ok (non-closed singly connected) (-10.96 -10.4
              Body_01   103786   113379  ok (non-closed singly connected) (-1.09642 -1.
                UB_02    81744    88448  multiply connected (shared edge) (-1.07551 -1.
            Mirror_04     1568     1889  ok (non-closed singly connected) (1.00835 -1.3
             tyres_fr    35534    42920  ok (non-closed singly connected) (-0.422953 -1
            wheels_fr    19641    23386  ok (non-closed singly connected) (-0.292191 -1
           tyres_rear    37974    44108  ok (non-closed singly connected) (3.35867 -1.1
          wheels_rear    18202    22074  ok (non-closed singly connected) (3.48848 -1.1
  <<Writing 4 conflicting points to set nonManifoldPoints
```

詳細モデルのcheckMesh（事例2）

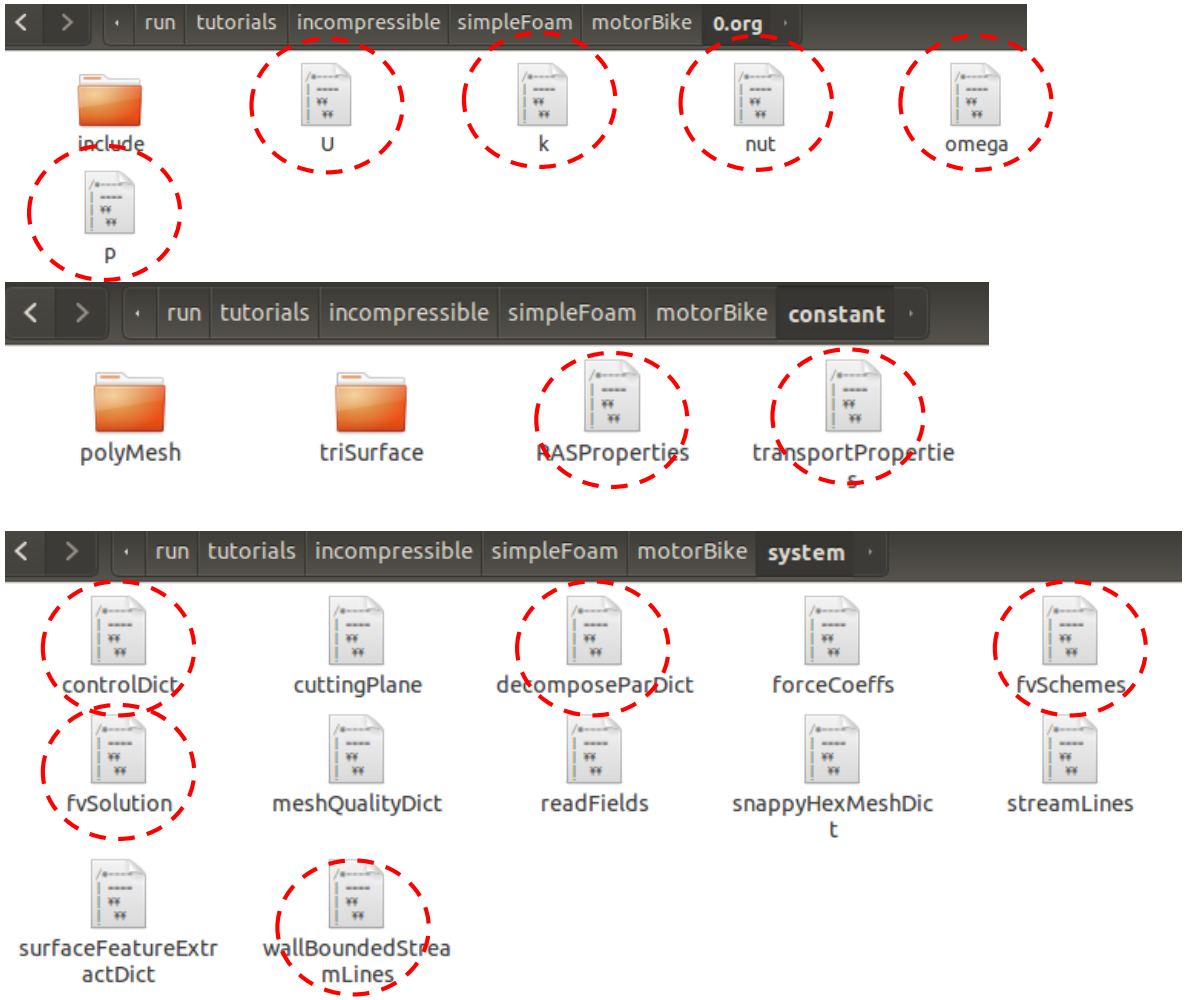
```
Checking geometry...
    Overall domain bounding box (-10.96 -10.48 -0.432) (30 10 9.82177)
    Mesh (non-empty, non-wedge) directions (1 1 1)
    Mesh (non-empty) directions (1 1 1)
    Boundary openness (-5.09467e-17 -1.51412e-16 2.1587e-15) OK.
    Max cell openness = 7.72204e-16 OK.
    Max aspect ratio = 36.0474 OK.
    Minimum face area = 1.88834e-07. Maximum face area = 1.07985.  Face area magnitudes OK.
    Min volume = 2.92525e-09. Max volume = 1.09085.  Total volume = 8573.89.  Cell volumes OK.
    Mesh non-orthogonality Max: 64.9562 average: 11.011
    Non-orthogonality check OK.
    Face pyramids OK.
***Max skewness = 4.24644, 2 highly skew faces detected which may impair the quality of the results
 <<Writing 2 skew faces to set skewFaces
   Coupled point location match (average 0) OK.
***Error in face tets: 6 faces with low quality or negative volume decomposition tets.
 <<Writing 6 faces with low quality or negative volume decomposition tets to set lowQualityTetFaces
  *Edges too small, min/max edge length = 2.5e-05 1.06241, number too small: 4
 <<Writing 4 points on short edges to set shortEdges
  *There are 7644 faces with concave angles between consecutive edges. Max concave angle = 80.0589 degrees.
 <<Writing 7644 faces with concave angles to set concaveFaces
   Face flatness (1 = flat, 0 = butterfly) : min = 0.183113  average = 0.998631
  *There are 359 faces with ratio between projected and actual area < 0.8
   Minimum ratio (minimum flatness, maximum warpage) = 0.183113
 <<Writing 359 warped faces to set warpedFaces
   Cell determinant (wellposedness) : minimum: 0 average: 14.5744
***Cells with small determinant (< 0.001) found, number of cells: 310
 <<Writing 310 under-determined cells to set underdeterminedCells
***Concave cells (using face planes) found, number of cells: 76610
 <<Writing 76610 concave cells to set concaveCells
   Face interpolation weight : minimum: 0.0206328 average: 0.45208
***Faces with small interpolation weight (< 0.05) found, number of faces: 4029
 <<Writing 4029 faces with low interpolation weights to set lowWeightFaces
   Face volume ratio : minimum: 0.0100268 average: 0.789277
   Face volume ratio check OK.

Failed 5 mesh checks.

End
```

メッシュの構築がよくないと，表面近傍の流れを作成する場合にエラーが発生する事がある。こうなった場合には，メッシュを作り直すしか方法がない（と考える）。

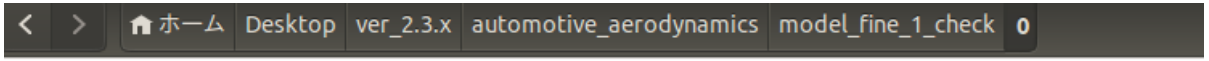
```
nearWallFields near1: Sampling 1 fields
[3]
[3]
[3] --> FOAM FATAL ERROR:
[3] No base point for face 1481795, 5(122318 109481 122319 95252 95250), produces a decomposition that has a minimum volume
greater than tolerance.
[3]
[3]     From function inline void Foam::particle::crossEdgeConnectedFace(const label& cellI,label& tetFaceI,label& tetPtI,const
edge& e)
[3]     in file /home/sakuramaru/OpenFOAM/OpenFOAM-2.3.x/src/lagrangian/basic/lnInclude/particleI.H at line 564.
[3]
FOAM parallel run aborting
[3]
[3] #0  Foam::error::printStack(Foam::Ostream&) at ??:?
[3] #1  Foam::error::abort() at ??:?
[3] #2  double Foam::particle::trackToFace<Foam::findCellParticle::trackingData>(Foam::Vector<double> const&,
Foam::findCellParticle::trackingData&) at ??:?
[3] #3  Foam::findCellParticle::move(Foam::findCellParticle::trackingData&, double) at ??:?
[3] #4  void Foam::Cloud<Foam::findCellParticle>::move<Foam::findCellParticle::trackingData>
(Foam::findCellParticle::trackingData&, double) at ??:?
[3] #5  Foam::nearWallFields::calcAddressing() at ??:?
[3] #6  Foam::nearWallFields::read(Foam::dictionary const&) at ??:?
[3] #7  Foam::nearWallFields::nearWallFields(Foam::word const&, Foam::objectRegistry const&, Foam::dictionary const&, bool)
at ??:?
[3] #8  Foam::OutputFilterFunctionObject<Foam::nearWallFields>::allocateFilter() at ??:?
[3] #9  Foam::OutputFilterFunctionObject<Foam::nearWallFields>::start()[1]
```

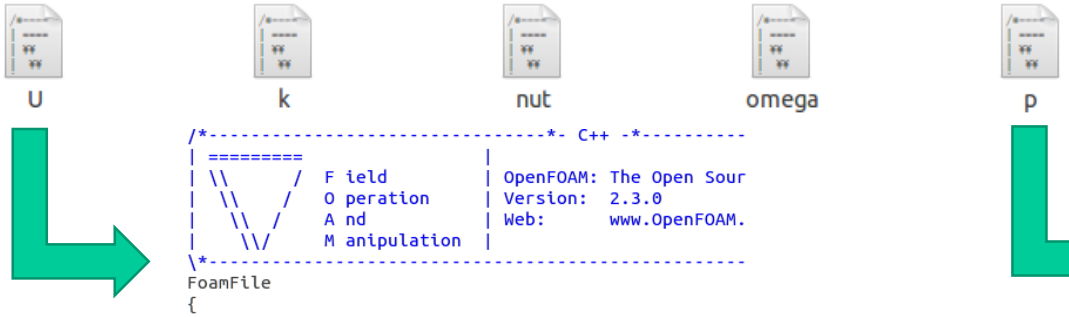
4. 計算設定

今回は，モータバイクの事例を参考に設定しています



計算設定 U,p,k,などはpyFoamを利用して設定
モータバイクのチュートリアルから0ホルダをコピー






```
/*--------------------------------*- C++ -*----------
| =========                 |
| \\      /  F ield         | OpenFOAM: The Open Sour
|  \\    /   O peration     | Version:  2.3.0
|   \\  /    A nd           | Web:      www.OpenFOAM.
|    \\/     M anipulation  |
\*---------------------------------------------------
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    location    "0";
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * *

#include        "include/initialConditions"

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform $flowVelocity;

boundaryField
{
    //- Set patchGroups for constraint patches
    #includeEtc "caseDicts/setConstraintTypes"

    #include "include/fixedInlet"

    outlet
    {
        type            inletOutlet;
        inletValue      uniform (0 0 0);
        value           $internalField;
    }

    lowerWall
    {
        type            fixedValue;
        value           $internalField;
```

```
/*--------------------------------*- C++ -*----------
| =========                 |
| \\      /  F ield         | OpenFOAM: The Open Sour
|  \\    /   O peration     | Version:  2.3.0
|   \\  /    A nd           | Web:      www.OpenFOAM.
|    \\/     M anipulation  |
\*---------------------------------------------------
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p;
}
// * * * * * * * * * * * * * * * * * * * * * * * * *

#include        "include/initialConditions"

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform $pressure;

boundaryField
{
    //- Set patchGroups for constraint patches
    #includeEtc "caseDicts/setConstraintTypes"

    inlet
    {
        type            zeroGradient;
    }

    outlet
    {
        type            fixedValue;
        value           $internalField;
    }

    lowerWall
    {
        type            zeroGradient;
    }
```

Includeなどの使わない部分を削除

```
/*--------------------------------*- C++ -*----------
| =========                 |
| \\      /  F ield         | OpenFOAM: The Open Sour
|  \\    /   O peration     | Version:  2.3.0
|   \\  /    A nd           | Web:      www.OpenFOAM.
|    \\/     M anipulation  |
\*---------------------------------------------------
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    location    "0";
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * *

#include        "include/initialConditions"

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform $flowVelocity;

boundaryField
{
    //- Set patchGroups for constraint patches
    #includeEtc "caseDicts/setConstraintTypes"

    #include "include/fixedInlet"

    outlet
    {
        type            inletOutlet;
        inletValue      uniform (0 0 0);
        value           $internalField;
    }

    lowerWall
    {
        type            fixedValue;
        value           $internalField;
```

```
|  \\    /   O peration     | Version:  2.3.0
|   \\  /    A nd           | Web:      www.OpenF
|    \\/     M anipulation  |
\*---------------------------------------------------
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    location    "0";
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * *

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform $flowVelocity;

boundaryField
{

    outlet
    {
        type            inletOutlet;
        inletValue      uniform (0 0 0);
        value           $internalField;
    }

    lowerWall
    {
        type            fixedValue;
        value           $internalField;
    }

    motorBikeGroup
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

}
```

*set_boundary.sh ×

```
echo "start of set boundary"

echo "set of U"
$runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/U

echo "set of p"
$runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/p

echo "set of k"
$runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/k

echo "set of omega"
$runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/omega

echo "set of nut"
$runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/nut

echo "end of set boundary"
```

こんな感じで境界面が設定されるため，初期条件に合うようにエディターで修正する

```
// -*- C++ -*-
// File generated by PyFoam - sorry for the ugliness

FoamFile
{
 version 2.0;
 format ascii;
 class volVectorField;
 location "0";
 object U;
}

dimensions [ 0 1 -1 0 0 0 0 ];

internalField uniform $flowVelocity;

boundaryField
{
  outlet
  {
    type inletOutlet;
    inletValue uniform (0 0 0);
    value $internalField;
  }
  inlet
  {
    type zeroGradient;
  }
  road
  {
    type zeroGradient;
  }
  air_area
  {
    type zeroGradient;
  }
  Body_01
  {
    type zeroGradient;
```

Include等の部分を削除しないと上手くいかない

```
sakuramaru@SAKURA-MARU:~/Desktop/ver_2.3.x/automotive_aerodynamics/model_fine_1_check$ sh set_boundary.sh
start of set boundary
set of U
 Error in /usr/local/bin/pyFoamCreateBoundaryPatches.py : Error in PyFoamParser: 'Syntax error in file /home/saku
maru/Desktop/ver_2.3.x/automotive_aerodynamics/model_fine_1_check/0/U at token' @ 'includeEtc' (Type: NAME ) in li
e 27 at position 958
set of p
 Error in /usr/local/bin/pyFoamCreateBoundaryPatches.py : Error in PyFoamParser: 'Syntax error in file /home/saku
maru/Desktop/ver_2.3.x/automotive_aerodynamics/model_fine_1_check/0/p at token' @ 'includeEtc' (Type: NAME ) in li
e 26 at position 933
set of k
 Error in /usr/local/bin/pyFoamCreateBoundaryPatches.py : Error in PyFoamParser: 'Syntax error in file /home/saku
maru/Desktop/ver_2.3.x/automotive_aerodynamics/model_fine_1_check/0/k at token' @ 'includeEtc' (Type: NAME ) in li
e 26 at position 936
set of omega
 Error in /usr/local/bin/pyFoamCreateBoundaryPatches.py : Error in PyFoamParser: 'Syntax error in file /home/saku
maru/Desktop/ver_2.3.x/automotive_aerodynamics/model_fine_1_check/0/omega at token' @ 'includeEtc' (Type: NAME ) i
 line 26 at position 943
set of nut
 Error in /usr/local/bin/pyFoamCreateBoundaryPatches.py : Error in PyFoamParser: 'Syntax error in file /home/saku
maru/Desktop/ver_2.3.x/automotive_aerodynamics/model_fine_1_check/0/nut at token' @ 'includeEtc' (Type: NAME ) in
ine 25 at position 903
end of set boundary
```

計算設定 RASProperties

乱流モデルは色々ありますが，今回はkOmegaSST

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      RASProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

RASModel          kOmegaSST;

turbulence        on;

printCoeffs       on;

laminarCoeffs
{
}

kEpsilonCoeffs
{
    Cmu             0.09;
    C1              1.44;
    C2              1.92;
    alphaEps        0.76923;
}
```

計算設定 fvSchemes

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * *

ddtSchemes
{
    default steadyState;
}

gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
    grad(U)         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)       bounded Gauss upwind;
    div(phi,k)       bounded Gauss upwind;
    div(phi,epsilon) bounded Gauss upwind;
    div(phi,omega)   bounded Gauss upwind;
    div(div(phi,U)) Gauss linear;
    div((nuEff*dev(T(grad(U))))) Gauss linear;
}
```

```
laplacianSchemes
{
    default         Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
    interpolate(U)  linear;
}

snGradSchemes
{
    default         corrected;
}

fluxRequired
{
    default         no;
    pcorr           ;
    p               ;
    Phi             ;
}
```

計算設定 fvSolution

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * *

solvers
{
    p
    {
        solver          GAMG;
        tolerance       1e-08;
        relTol          0.05;
        smoother        GaussSeidel;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 20;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }

    U
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        nSweeps         2;
        tolerance       1e-07;
        relTol          0.1;
    }
```

```
    k
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        nSweeps         2;
        tolerance       1e-07;
        relTol          0.1;
    }

    omega
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        nSweeps         2;
        tolerance       1e-07;
        relTol          0.1;
    }

    Phi
    {
        solver          GAMG;
        smoother        DIC;
        cacheAgglomeration on;
        agglomerator    faceAreaPair;
        nCellsInCoarsestLevel 10;
        mergeLevels     1;

        tolerance       1e-06;
        relTol          0.01;
    }
}
```

```
potentialFlow
{
    nNonOrthogonalCorrectors 20;
}

SIMPLE
{
    nNonOrthogonalCorrectors 1;
    convergence 1e-3;
    pRefCell        0;
    pRefValue       0;
}

PISO
{
    nCorrectors     2;
    nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
    fields
    {
        p               0.3;
    }
    equations
    {
        U               0.7;
        k               0.7;
        omega           0.7;
    }
}

cache
{
    grad(U);
}
```

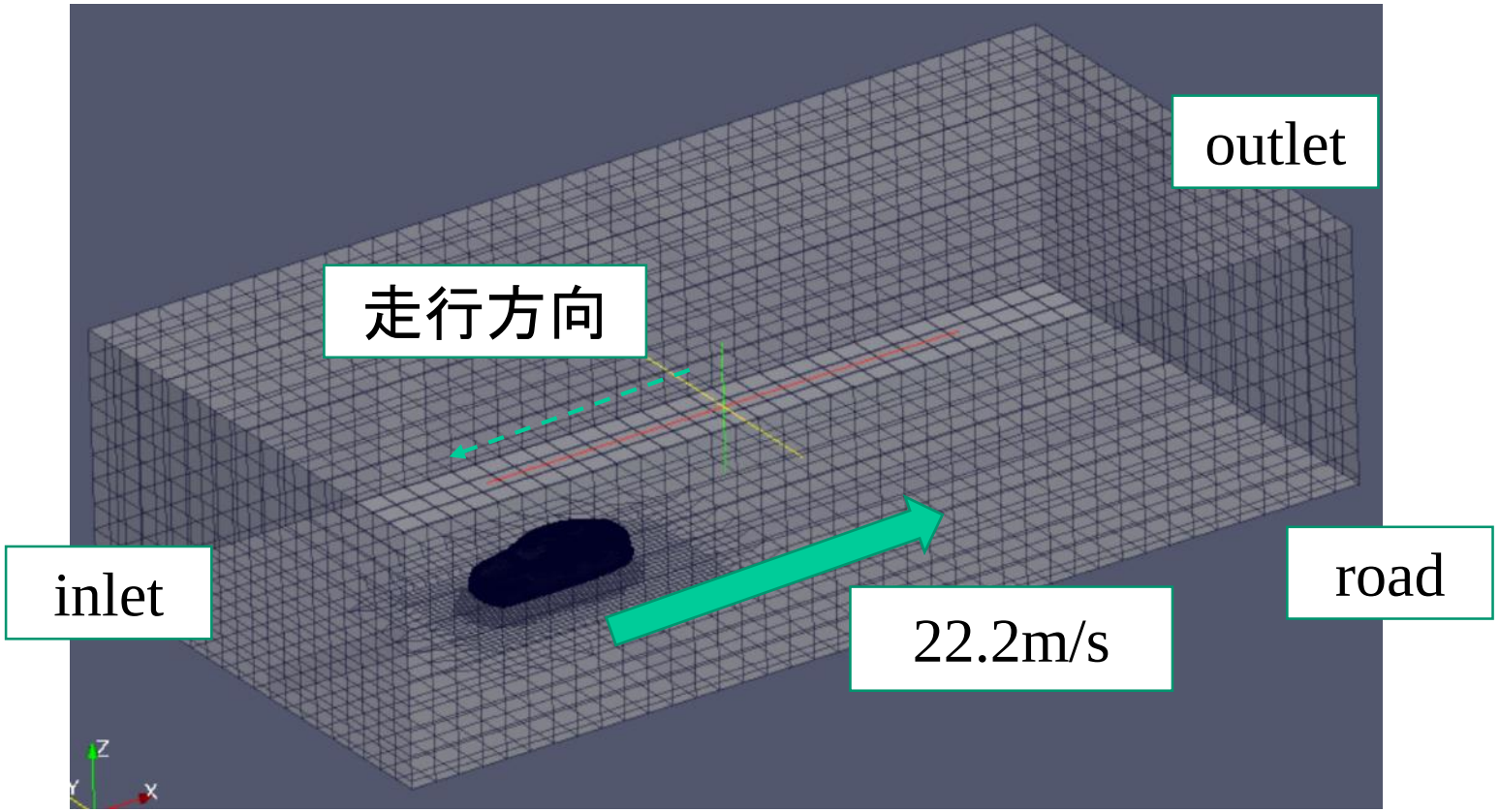
計算設定 controlDict

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * *
application     simpleFoam;
startFrom       latestTime;
startTime       0;
stopAt          endTime;
endTime         1500;
deltaT          1;
writeControl    timeStep;
writeInterval   1500;
purgeWrite      0;
writeFormat     ascii;
writePrecision  6;
writeCompression compressed;
timeFormat      general;
timePrecision   6;
runTimeModifiable yes;
// ****************************
```

```
functions
{
      #include "wallBoundedStreamLines1"
      #include "wallBoundedStreamLines2"
}
libs (
    "libOpenFOAM.so"
    "libincompressibleTurbulenceModel.so"
    "libincompressibleRASModels.so"
);
```

計算条件（簡略モデル）

　車が80km/h（22.2m/s)で走行すると仮定して，走行方向と逆に風速を与える。



計算設定(簡略モデル)

U

```
FoamFile
{
 version 2.0;
 format ascii;
 class volVectorField;
 location "0";
 object U;
}
dimensions [ 0 1 -1 0 0 0 0 ];
internalField uniform (0 0 0);
boundaryField
{
  inlet
  {
        type            fixedValue;
        value           uniform (22.2 0 0);
  }
  outlet
  {
    type zeroGradient;
  }
  road
  {
        type            fixedValue;
        value           uniform (0 0 0);
  }
  air_area
  {
        type            slip;
  }
  Body_Closed
  {
        type            fixedValue;
        value           uniform (0 0 0);
  }
}
```

p

```
FoamFile
{
 version 2.0;
 format ascii;
 class volScalarField;
 location "0";
 object p;
}
dimensions [ 0 2 -2 0 0 0 0 ];
internalField uniform 0;
boundaryField
{
  inlet
  {
    type zeroGradient;
  }
  outlet
  {
        type            fixedValue;
        value           uniform 0;
  }
  road
  {
    type zeroGradient;
  }
  air_area
  {
        type            slip;
  }
  Body_Closed
  {
    type zeroGradient;
  }
}
```

k

```
FoamFile
{
 version 2.0;
 format ascii;
 class volScalarField;
 location "0";
 object k;
}
dimensions [ 0 2 -2 0 0 0 0 ];
internalField uniform 0.06;
boundaryField
{
  inlet
  {
      type       inletOutlet;
      inletValue $internalField;
      value      $internalField;
  }
  outlet
  {
      type       inletOutlet;
      inletValue $internalField;
      value      $internalField;
  }
  road
  {
    type kqRWallFunction;
    value $internalField;
  }
  air_area
  {
        type            slip;
  }
  Body_Closed
  {
    type kqRWallFunction;
    value $internalField;
  }
}
```

計算設定（簡略モデル）

omega

```
FoamFile
{
 version 2.0;
 format ascii;
 class volScalarField;
 location "0";
 object omega;
}
dimensions [ 0 0 -1 0 0 0 0 ];
internalField uniform 20;
boundaryField
{
  inlet
  {
    type       inletOutlet;
    inletValue $internalField;
    value      $internalField;
  }
  outlet
  {
    type       inletOutlet;
    inletValue $internalField;
    value      $internalField;
  }
  road
  {
    type omegaWallFunction;
    value $internalField;
  }
  air_area
  {
        type            slip;
  }
  Body_Closed
  {
    type omegaWallFunction;
    value $internalField;
  }
}
```

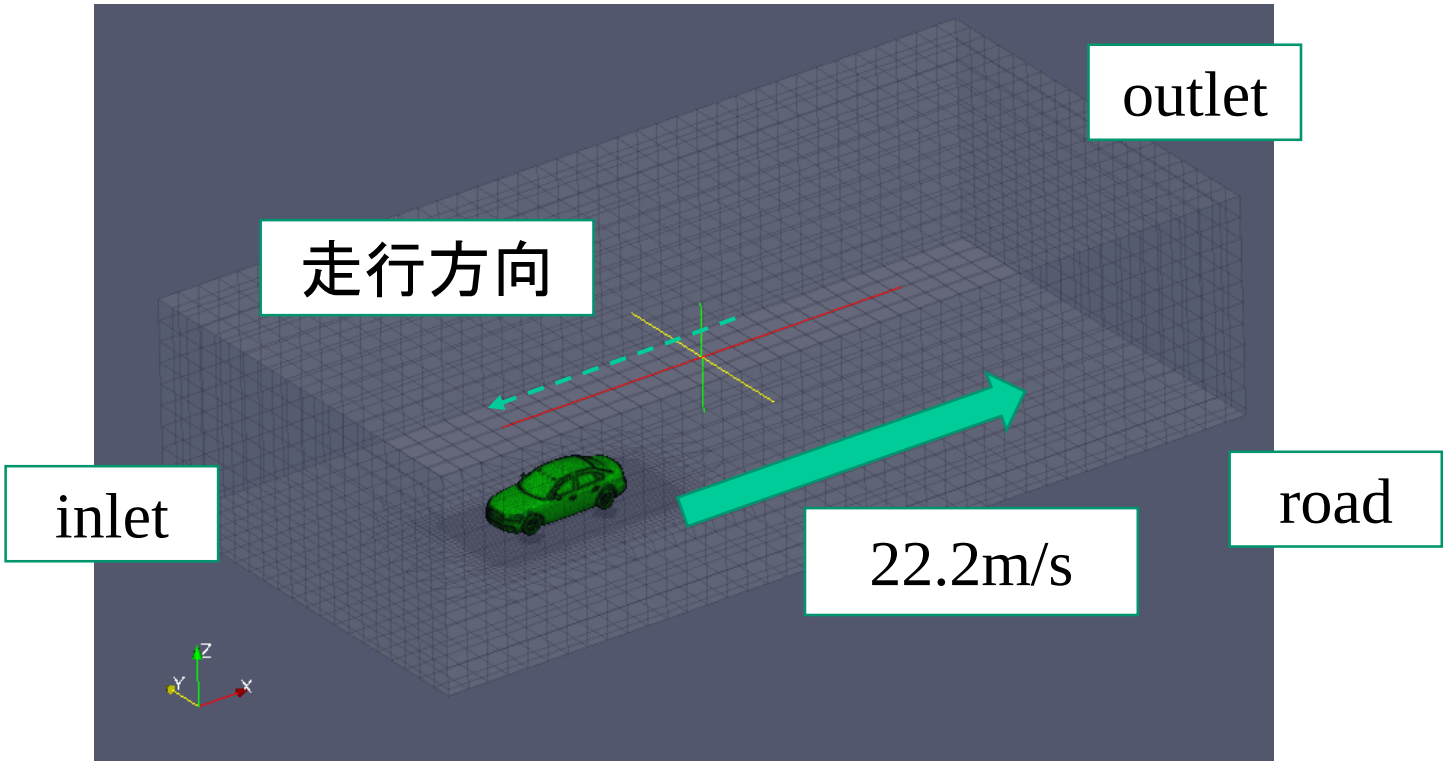
nut

```
FoamFile
{
 version 2.0;
 format ascii;
 class volScalarField;
 location "0";
 object nut;
}
dimensions [ 0 2 -1 0 0 0 0 ];
internalField uniform 0;
boundaryField
{
  inlet
  {
    type calculated;
    value uniform 0;
  }
  outlet
  {
    type calculated;
    value uniform 0;
  }
  road
  {
    type nutkWallFunction;
    value uniform 0;
  }
  air_area
  {
        type            slip;
  }
  Body_Closed
  {
    type nutkWallFunction;
    value uniform 0;
  }
}
```

計算条件（詳細モデル）

①：車が80km/h（22.2m/s)で走行すると仮定して，走行方向と逆に風速を与える

②：①に加えてタイヤの回転を考慮する



計算条件（詳細モデル）
①：簡略モデルとの差は車部分の設定

U

```
Body_01
{
      type            fixedValue;
      value           uniform (0 0 0);
}
UB_02
{
      type            fixedValue;
      value           uniform (0 0 0);
}
Mirror_04
{
      type            fixedValue;
      value           uniform (0 0 0);
}
tyres_fr
{
      type            fixedValue;
      value           uniform (0 0 0);
}
wheels_fr
{
      type            fixedValue;
      value           uniform (0 0 0);
}
tyres_rear
{
      type            fixedValue;
      value           uniform (0 0 0);
}
wheels_rear
{
      type            fixedValue;
      value           uniform (0 0 0);
}
```

p

```
Body_01
{
  type zeroGradient;
}
UB_02
{
  type zeroGradient;
}
Mirror_04
{
  type zeroGradient;
}
tyres_fr
{
  type zeroGradient;
}
wheels_fr
{
  type zeroGradient;
}
tyres_rear
{
  type zeroGradient;
}
wheels_rear
{
  type zeroGradient;
}
```

k

```
Body_01
{
  type kqRWallFunction;
  value $internalField;
}
UB_02
{
  type kqRWallFunction;
  value $internalField;
}
Mirror_04
{
  type kqRWallFunction;
  value $internalField;
}
tyres_fr
{
  type kqRWallFunction;
  value $internalField;
}
wheels_fr
{
  type kqRWallFunction;
  value $internalField;
}
tyres_rear
{
  type kqRWallFunction;
  value $internalField;
}
wheels_rear
{
  type kqRWallFunction;
  value $internalField;
}
```

計算条件（詳細モデル）
①：簡略モデルとの差は車部分の設定

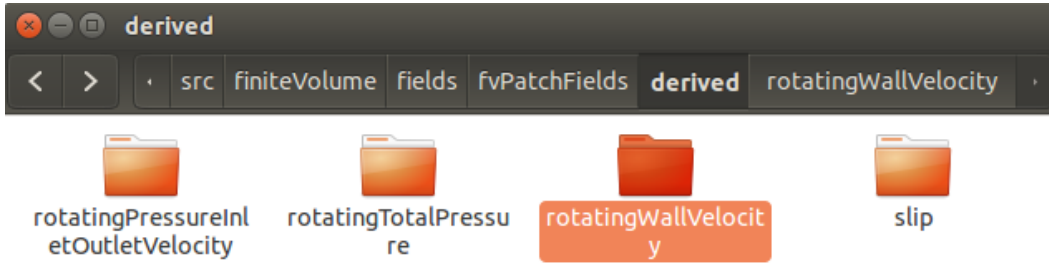
omega

```
Body_01
{
  type omegaWallFunction;
  value $internalField;
}
UB_02
{
  type omegaWallFunction;
  value $internalField;
}
Mirror_04
{
  type omegaWallFunction;
  value $internalField;
}
tyres_fr
{
  type omegaWallFunction;
  value $internalField;
}
wheels_fr
{
  type omegaWallFunction;
  value $internalField;
}
tyres_rear
{
  type omegaWallFunction;
  value $internalField;
}
wheels_rear
{
  type omegaWallFunction;
  value $internalField;
}
```

nut

```
Body_01
{
  type nutkWallFunction;
  value uniform 0;
}
UB_02
{
  type nutkWallFunction;
  value uniform 0;
}
Mirror_04
{
  type nutkWallFunction;
  value uniform 0;
}
tyres_fr
{
  type nutkWallFunction;
  value uniform 0;
}
wheels_fr
{
  type nutkWallFunction;
  value uniform 0;
}
tyres_rear
{
  type nutkWallFunction;
  value uniform 0;
}
wheels_rear
{
  type nutkWallFunction;
  value uniform 0;
}
```

計算条件（詳細モデル）
②：タイヤの回転を考慮する

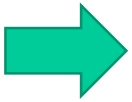



```
Description
    This boundary condition provides a rotational velocity condition.

    \heading Patch usage

    \table
        Property     | Description             | Required    | Default value
        origin       | origin of rotation in Cartesian co-ordinates | yes|
        axis         | axis of rotation        | yes         |
        omega        | angular velocty of the frame [rad/s] | yes    |
    \endtable

    Example of the boundary condition specification:
    \verbatim
    myPatch
    {
        type            rotatingWallVelocity;
        origin          (0 0 0);
        axis            (0 0 1);
        omega           100;
    }
    \endverbatim

    The \c omega entry is a DataEntry type, able to describe time varying
    functions.
```

計算条件(詳細モデル)

U:タイヤの回転部分のみ変更

```
tyres_fr
{
      type            fixedValue;
      value           uniform (0 0 0);
}
wheels_fr
{
      type            fixedValue;
      value           uniform (0 0 0);
}
tyres_rear
{
      type            fixedValue;
      value           uniform (0 0 0);
}
wheels_rear
{
      type            fixedValue;
      value           uniform (0 0 0);
}
```



```
tyres_fr
{
      type            rotatingWallVelocity;
      origin          (0.0095 0 0);
      axis            (0 -1 0);
      omega           51.381;
}
wheels_fr
{
      type            rotatingWallVelocity;
      origin          (0.0095 0 0);
      axis            (0 -1 0);
      omega           51.381;
}
tyres_rear
{
      type            rotatingWallVelocity;
      origin          (3.79 0 0);
      axis            (0 -1 0);
      omega           51.381;
}
wheels_rear
{
      type            rotatingWallVelocity;
      origin          (3.79 0 0);
      axis            (0 -1 0);
      omega           51.381;
}
```

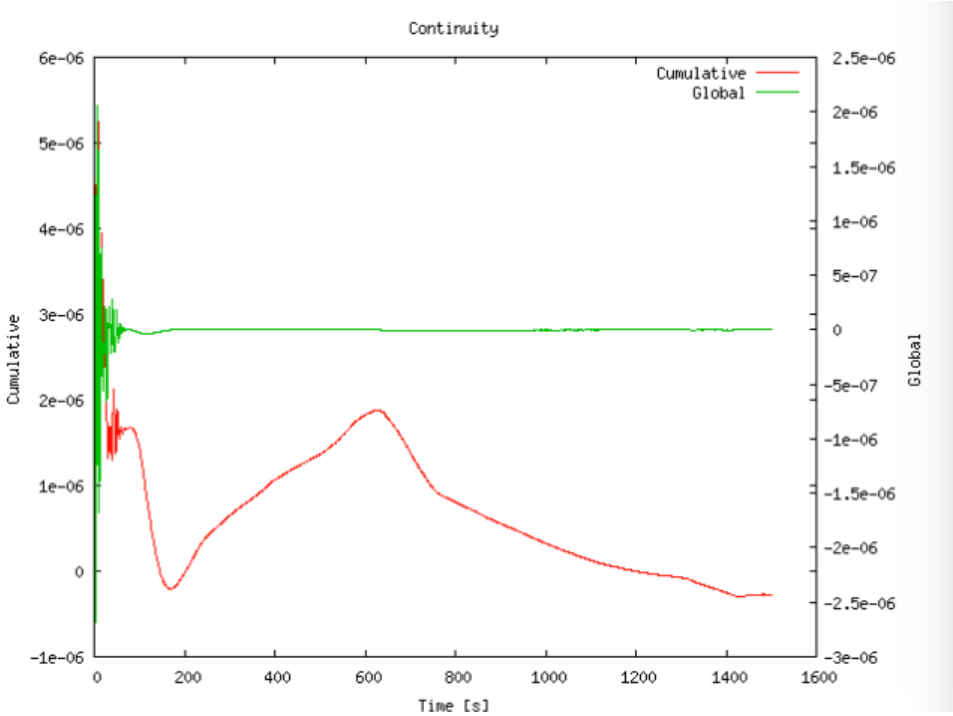
5. 計算実施

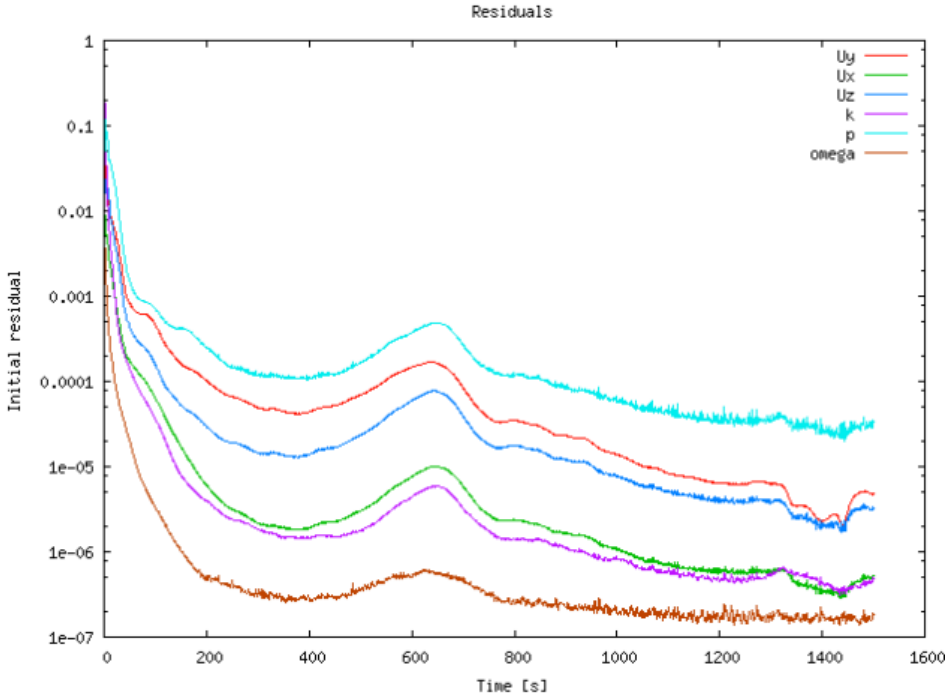
コマンドによる計算実行事例

run_4.sh

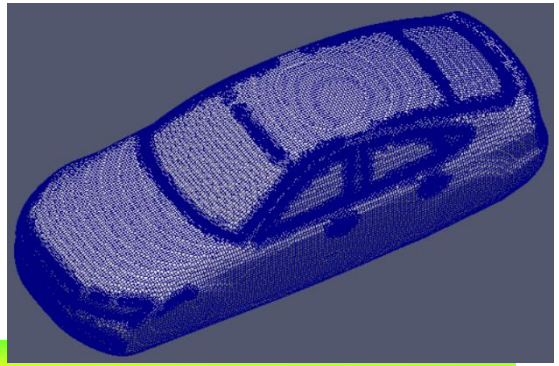
```
CASE_DIR=case_set/case1/set
UP_DIR=case_set/case1/set/backup
DECOMPOSE_PAR=decomposeParDict_case1_run
CONTROL_DICT=controlDict_case1_run
RAS_P=RASProperties_case1
TRAS_P=transportProperties_case1
TURB_P=turbulenceProperties_case1
FV_SCH=fvSchemes_case1
FV_SOL=fvSolution_case1
WALL_LINE1=wallBoundedStreamLines_case1
WALL_LINE2=wallBoundedStreamLines_case2

echo "Start of calculation"
cp -r $UP_DIR/normal1_type/U   0/
cp -r $UP_DIR/normal1_type/p   0/
cp -r $UP_DIR/normal1_type/k   0/
cp -r $UP_DIR/normal1_type/nut 0/
cp -r $UP_DIR/normal1_type/omega 0/
cp -r $CASE_DIR/$RAS_P         constant/RASProperties
cp -r $CASE_DIR/$TRAS_P        constant/transportProperties
cp -r $CASE_DIR/$TURB_P        constant/turbulenceProperties
cp -r $CASE_DIR/$FV_SCH        system/fvSchemes
cp -r $CASE_DIR/$FV_SOL        system/fvSolution
cp -r $CASE_DIR/$CONTROL_DICT  system/controlDict
cp -r $CASE_DIR/$DECOMPOSE_PAR system/decomposeParDict
cp -r $CASE_DIR/$WALL_LINE1    system/wallBoundedStreamLines1
cp -r $CASE_DIR/$WALL_LINE2    system/wallBoundedStreamLines2

echo "potentialFoam"
$runApplication potentialFoam -writep > log.potentialFoam

echo "decomposePar 4blocks"
$runApplication decomposePar > log.decomposePar

renumberMesh -case processor0 -overwrite
renumberMesh -case processor1 -overwrite
renumberMesh -case processor2 -overwrite
renumberMesh -case processor3 -overwrite

echo "simpleFoam 4CPU"
pyFoamPlotRunner.py mpirun -np 4 simpleFoam -parallel > log.si

echo "reconstructPar"
$runApplication reconstructPar > log.reconstructPar

echo "set data"
rm -r processor0
rm -r processor1
rm -r processor2
rm -r processor3

yPlusRAS -latestTime > log.yPlusRAS_latestTime_1
pyFoamPlotWatcher.py --solver-not-running-anymore --hardcopy l

echo "End of calculation"
```

簡略モデル:収束状態(事例1)

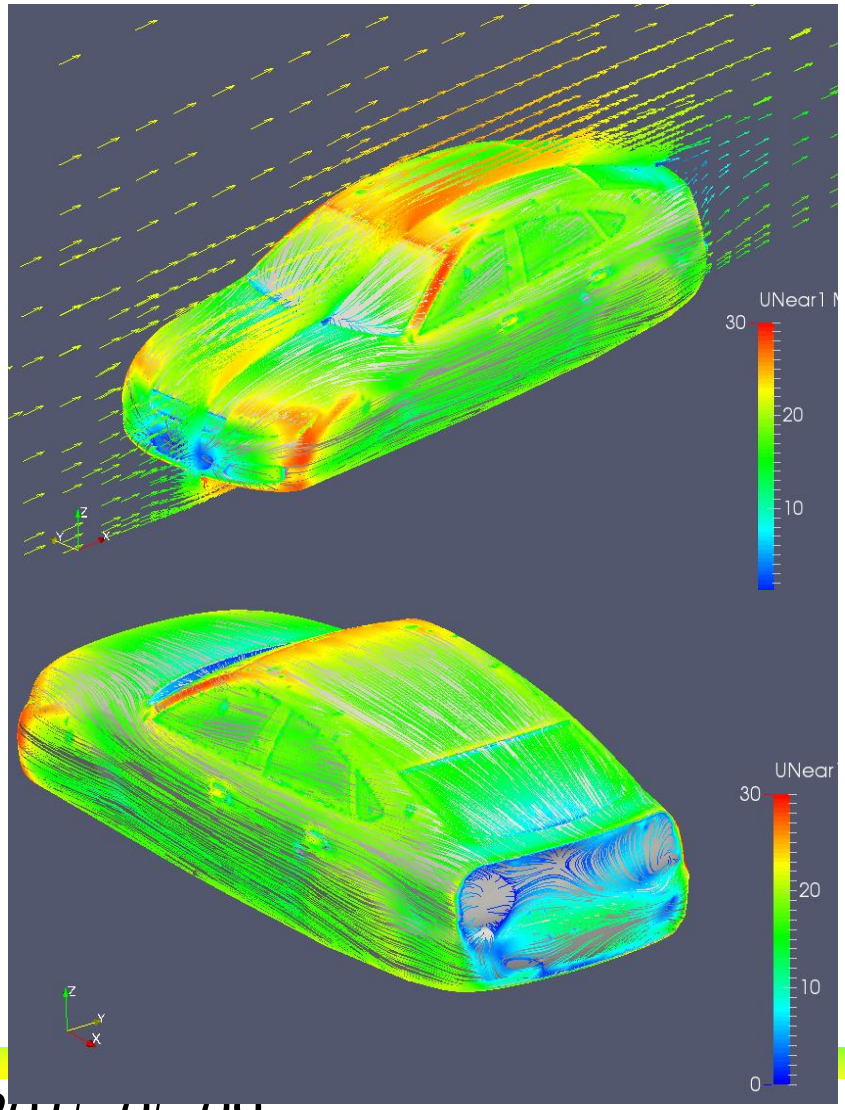


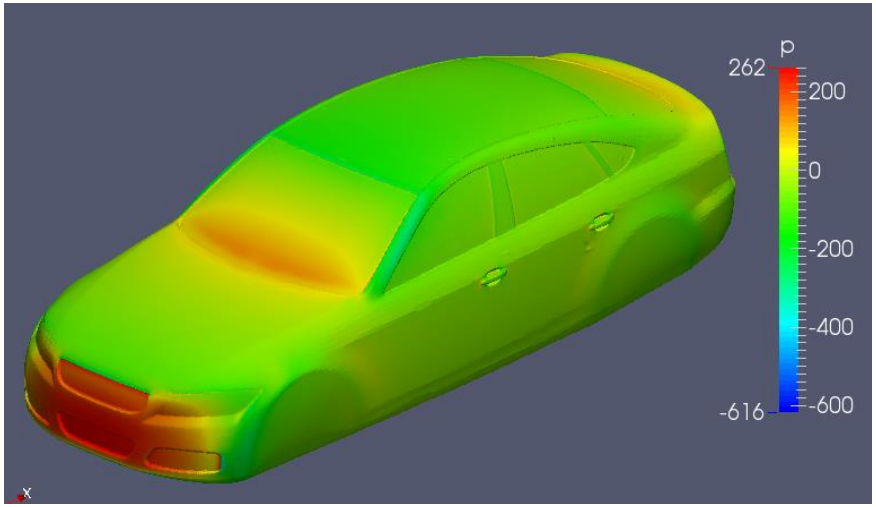


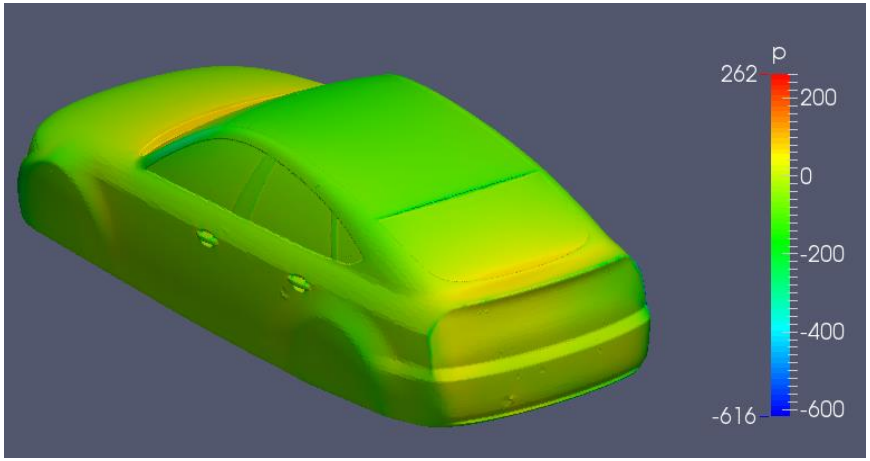
計算時間:4533秒



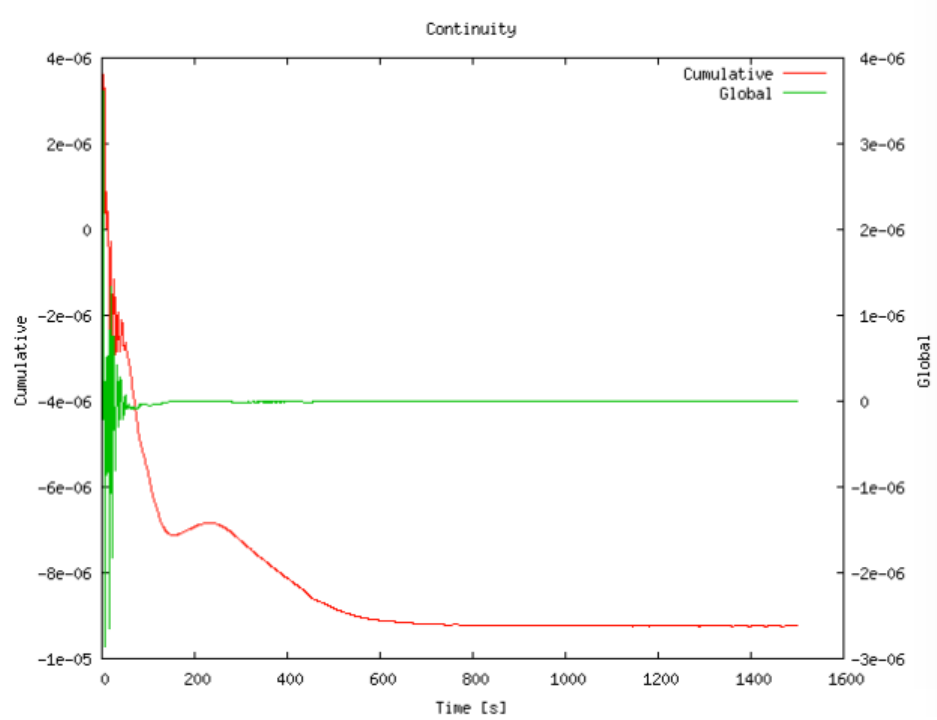
簡略モデル:結果(事例1)

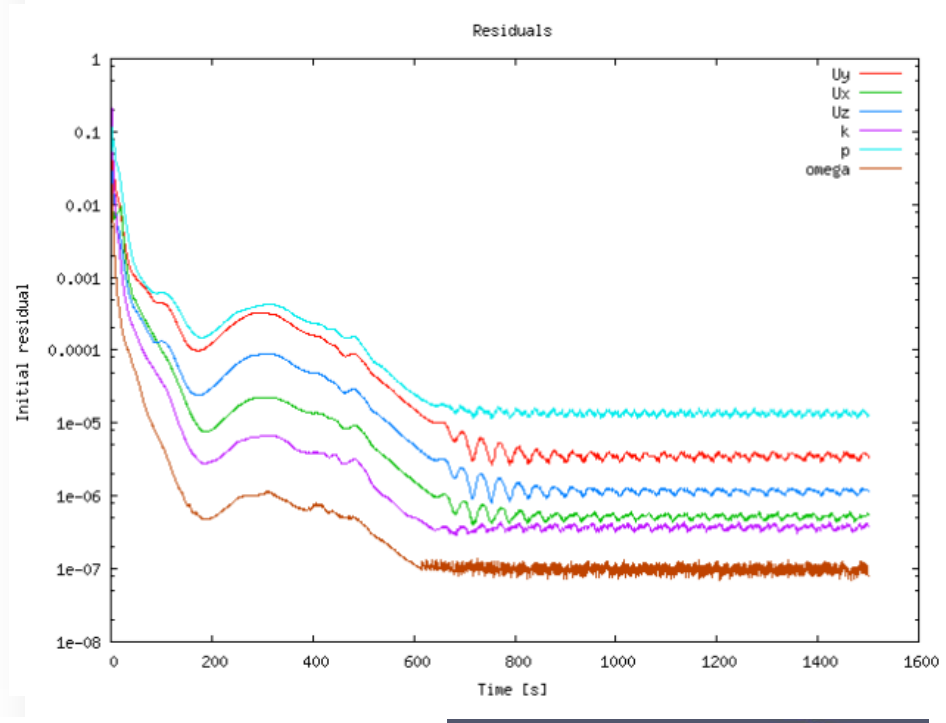

UNear1 M
30
20
10
UNear
30
20
10
0


p
262
200
0
-200
-400
-600
-616

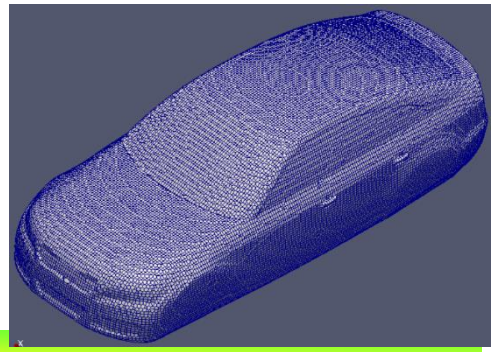

p
262
200
0
-200
-400
-600
-616

簡略モデル：収束状態（事例2）

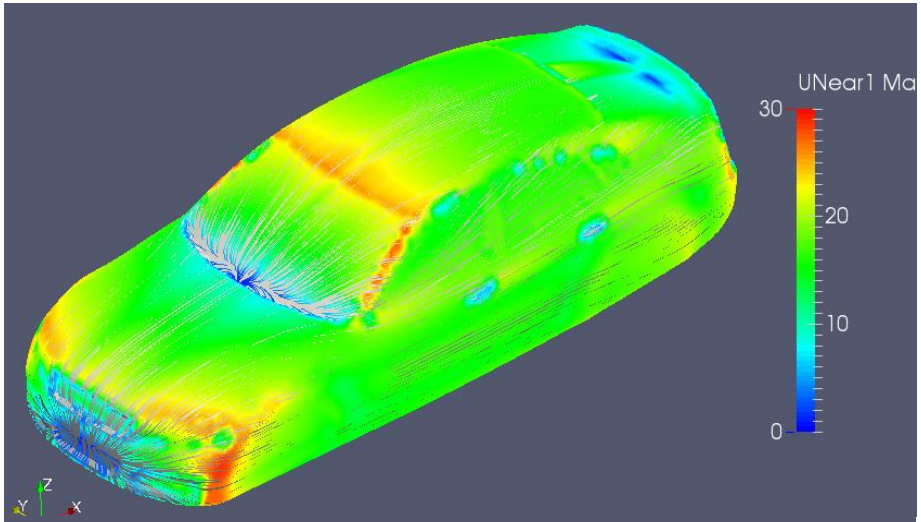


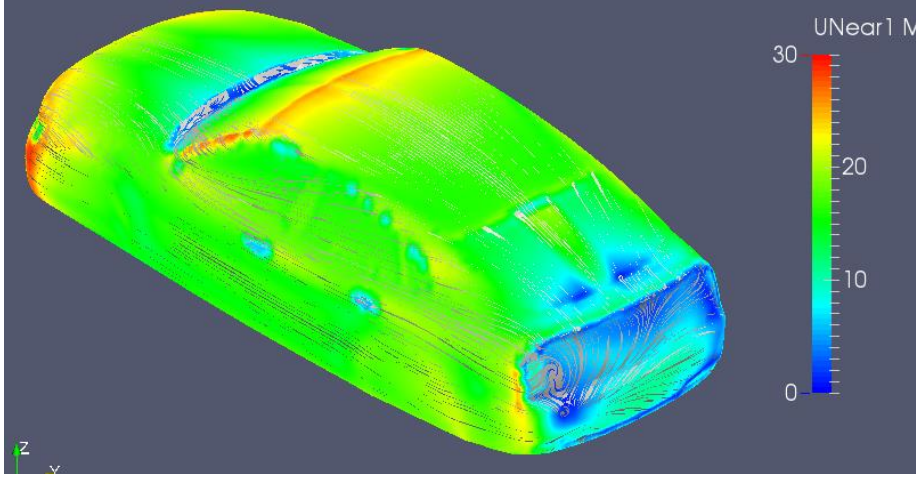


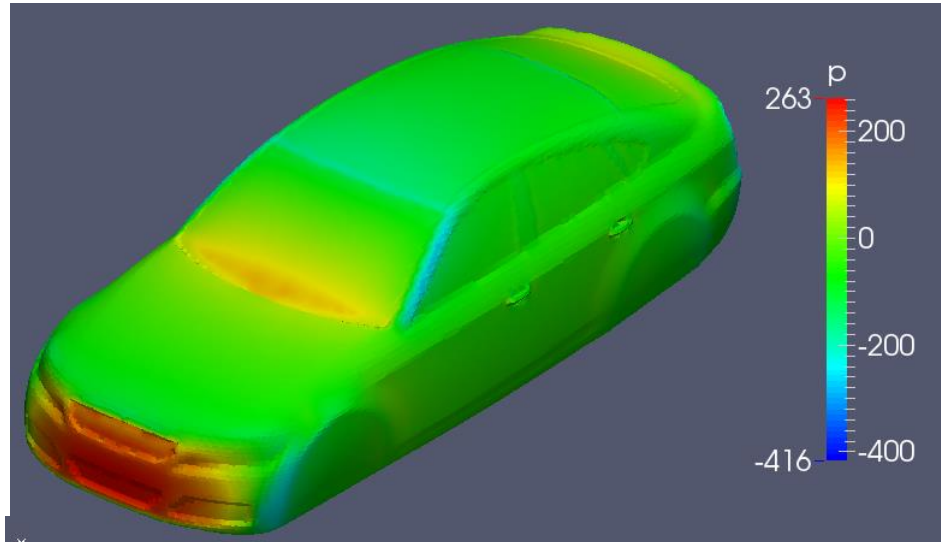
計算時間：1960秒

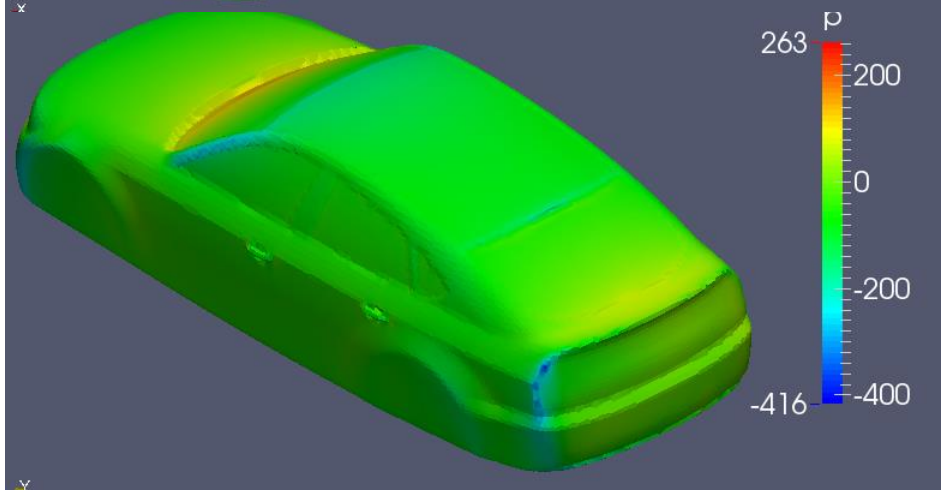


簡略モデル：結果（事例２）

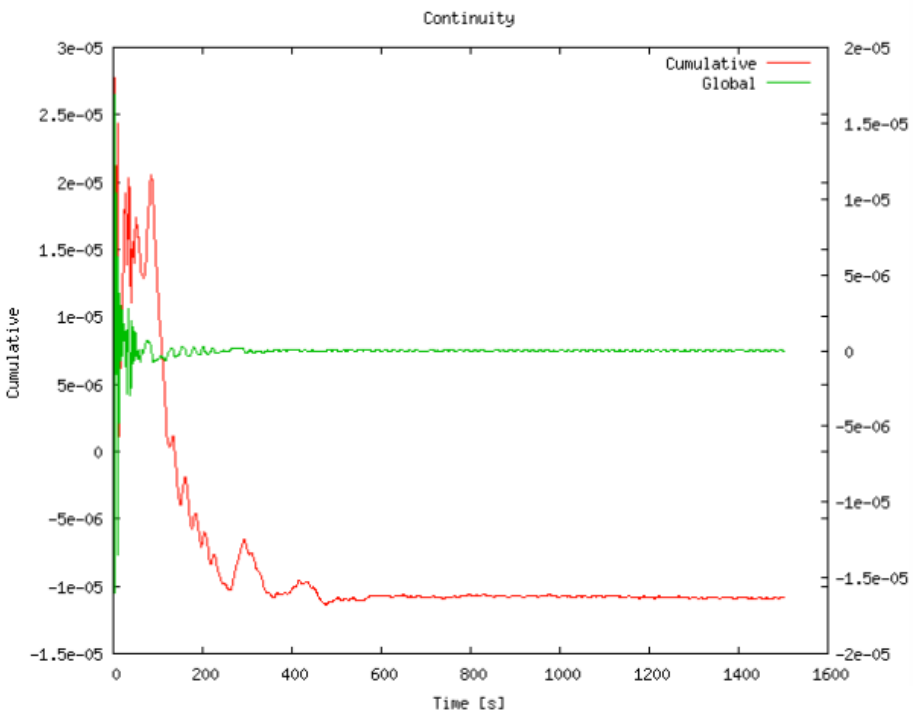

UNear1 Ma
30
20
10
0

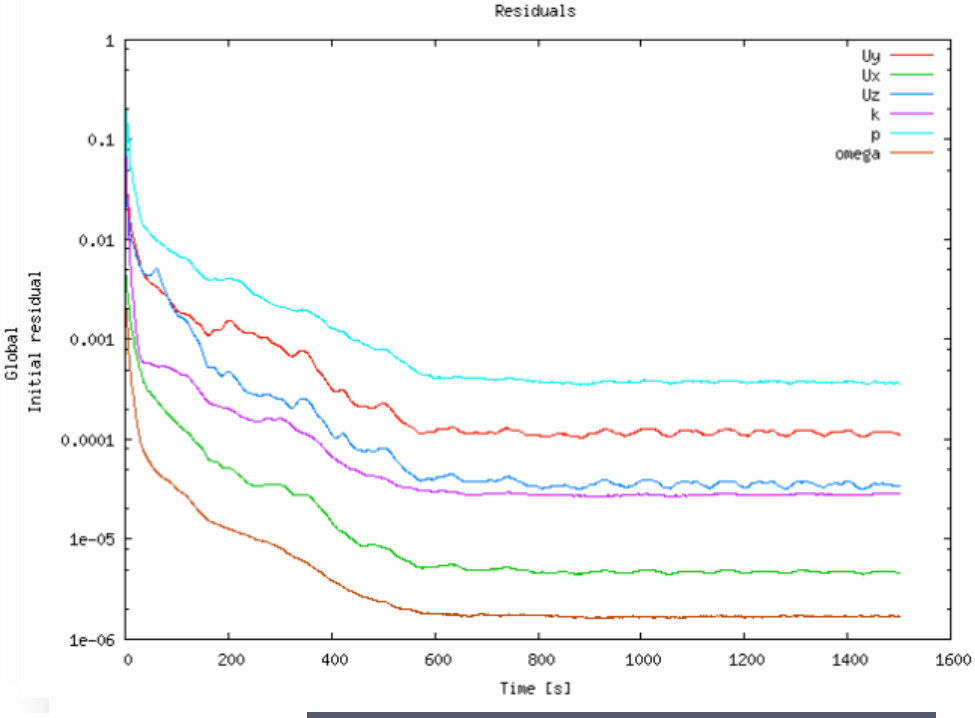

UNear1 M
30
20
10
0


p
263
200
0
-200
-400
-416

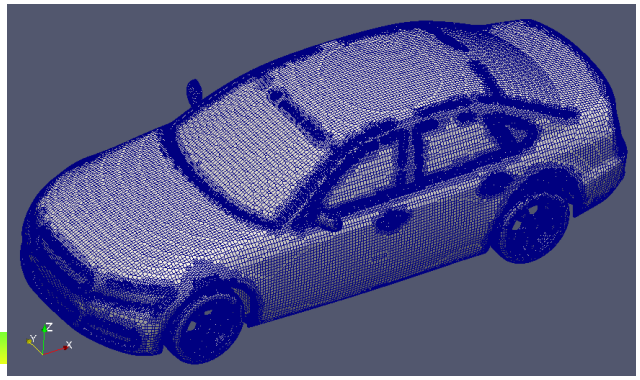

p
263
200
0
-200
-400
-416

詳細モデル:収束状態(事例1)



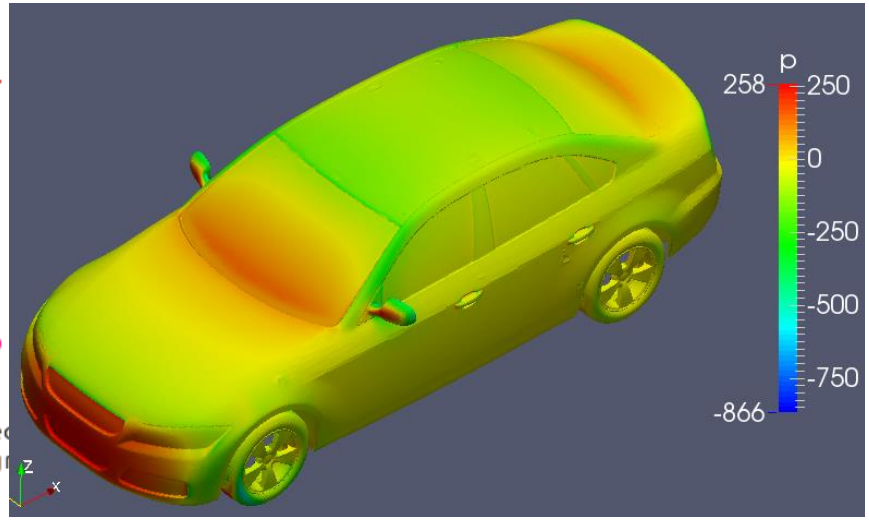


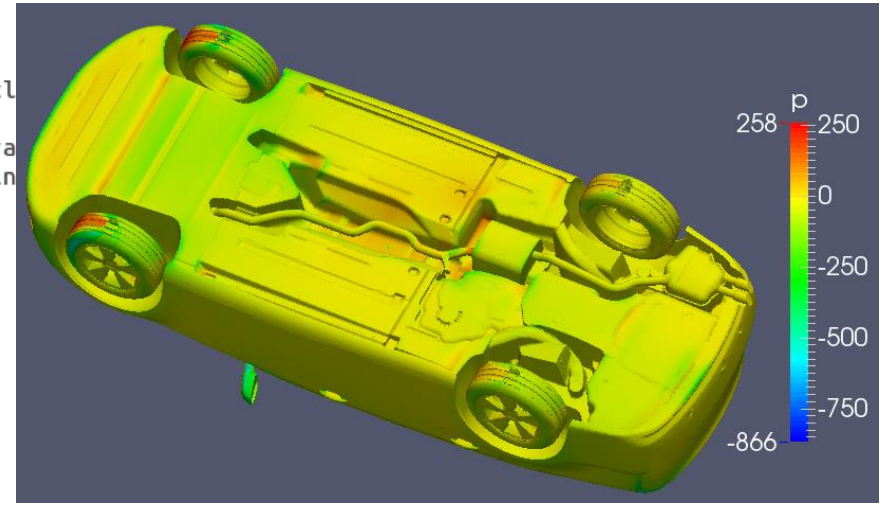
計算時間:9467秒



詳細モデル:結果(事例1)

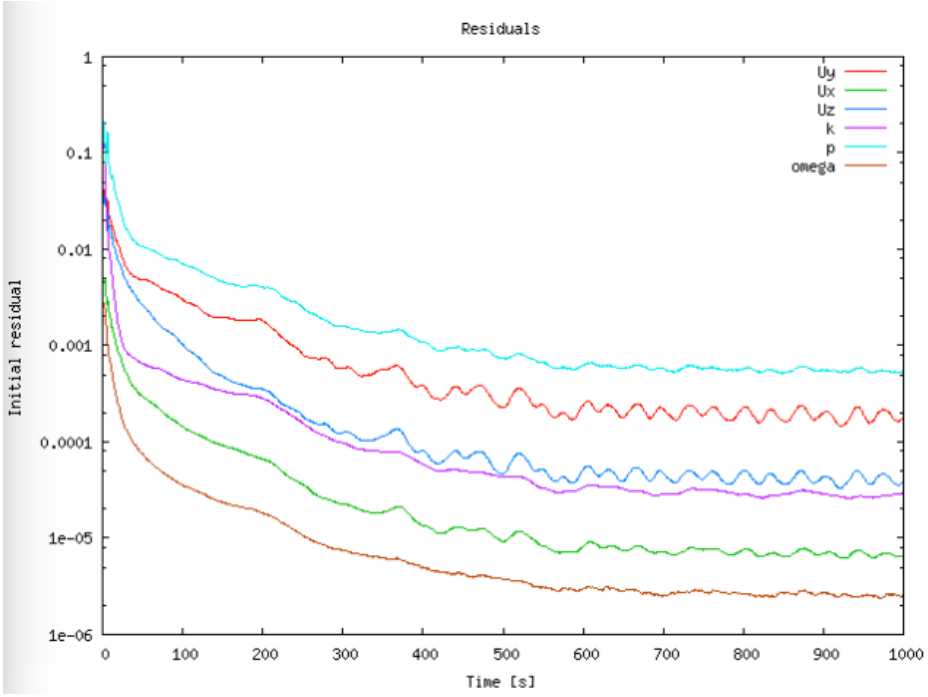
```
SIMPLE: no convergence criteria found. Calculations will run for

Starting time loop

nearWallFields near1: Sampling 1 fields
[3]
[3]
[3] --> FOAM FATAL ERROR:
[3] No base point for face 1402167, 5(123383 109248 123384 93039
tolerance.
[3]
[3]     From function inline void Foam::particle::crossEdgeConne
[3]     in file /home/sakuramaru/OpenFOAM/OpenFOAM-2.3.x/src/lag
[3]
FOAM parallel run aborting
[3]
[3] #0  Foam::error::printStack(Foam::Ostream&) at ??:?
[3] #1  Foam::error::abort() at ??:?
[3] #2  double Foam::particle::trackToFace<Foam::findCellParticl
Foam::findCellParticle::trackingData&) at ??:?
[3] #3  Foam::findCellParticle::move(Foam::findCellParticle::tra
[3] #4  void Foam::Cloud<Foam::findCellParticle>::move<Foam::fin
```



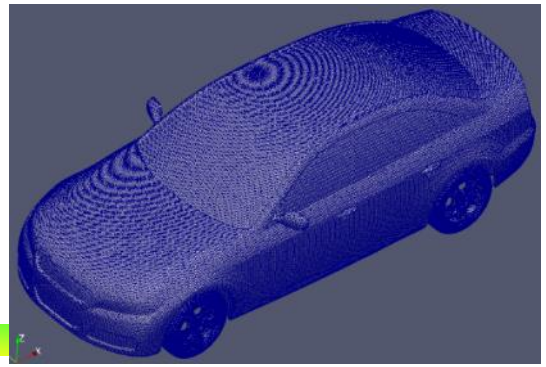


メッシュエラーのため近傍場の流れ計算エラー

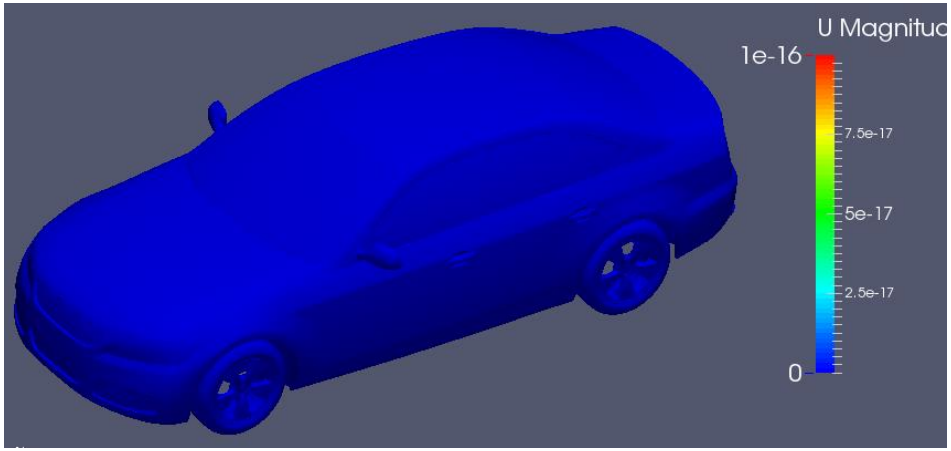
詳細モデル：収束状態（事例2 タイヤ回転なし）

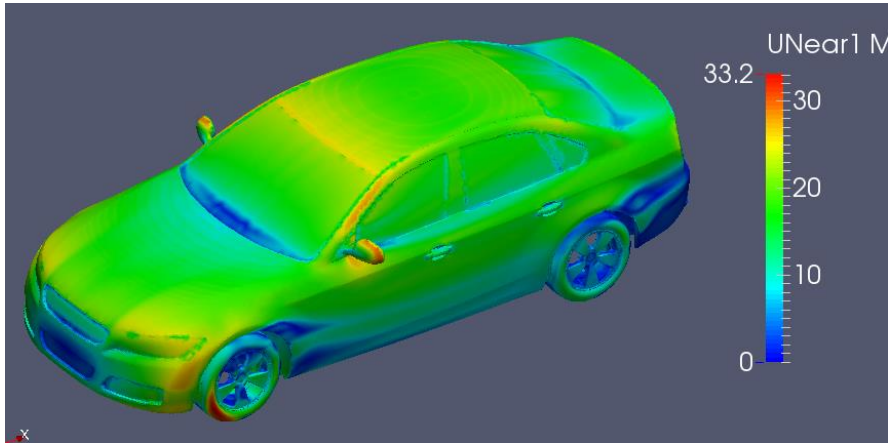


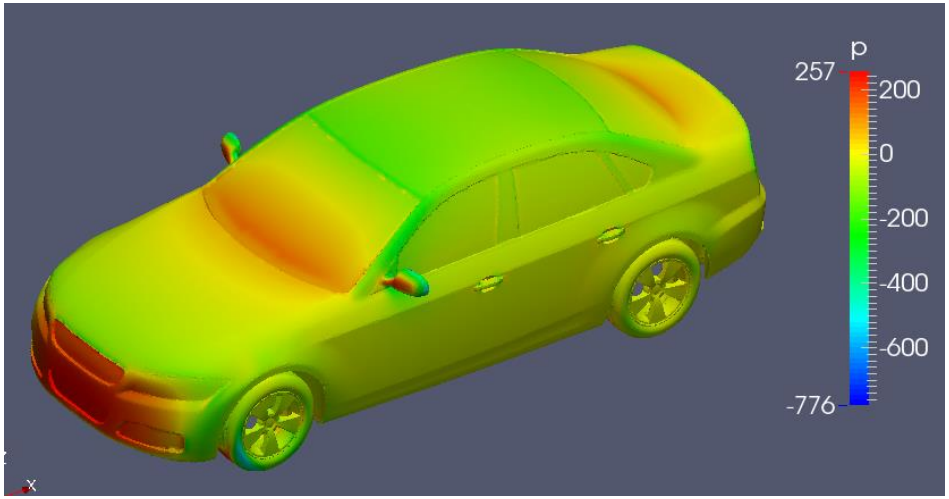
計算時間：4937秒（1000ステップ）

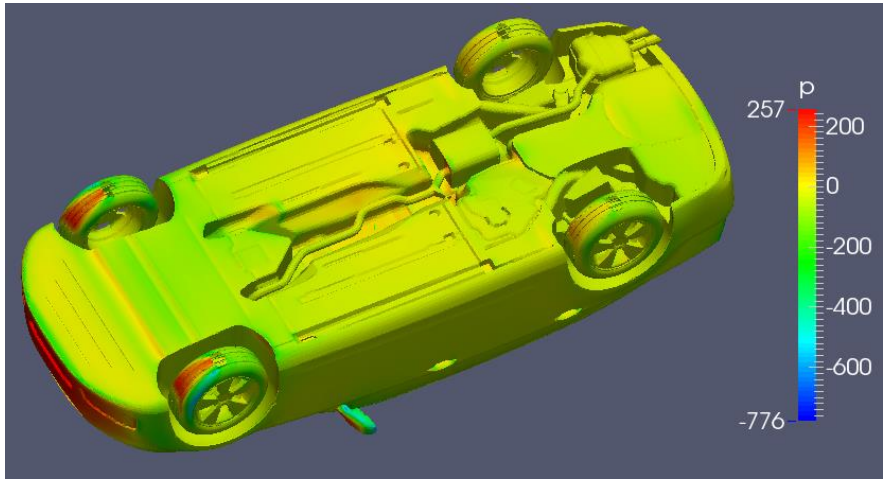


詳細モデル：事例2 タイヤ回転なし

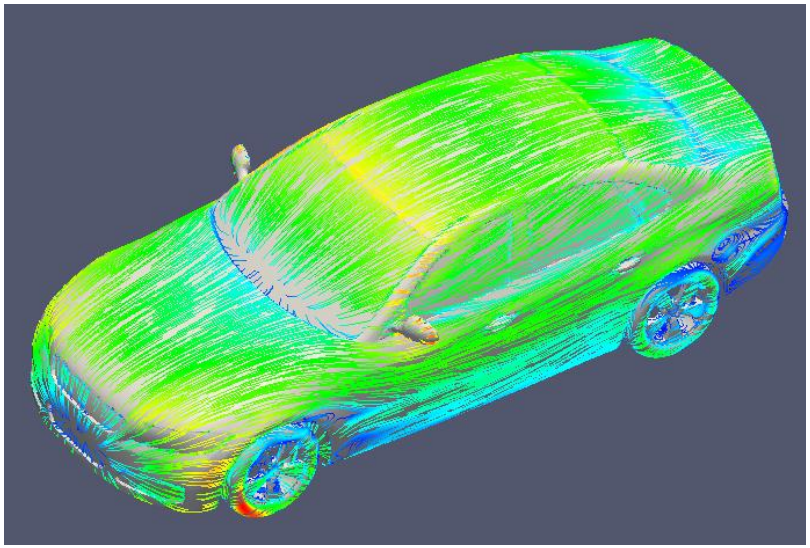


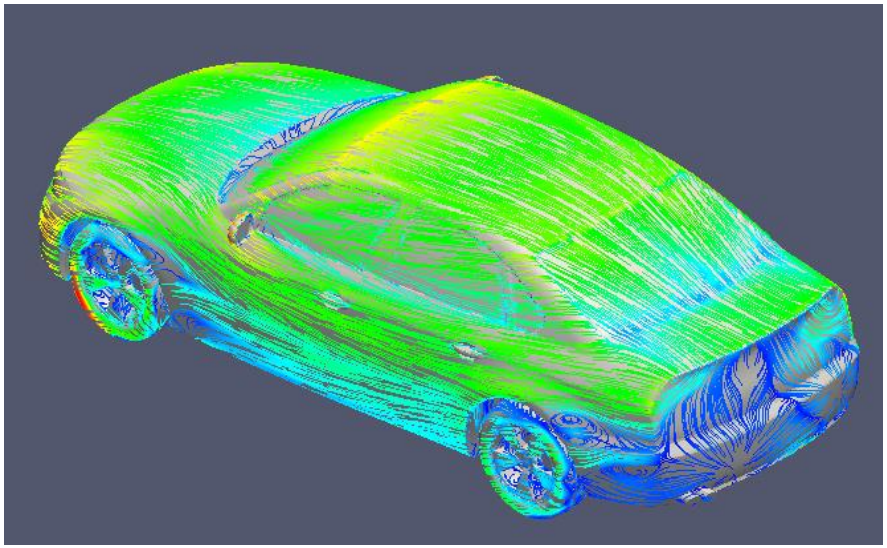


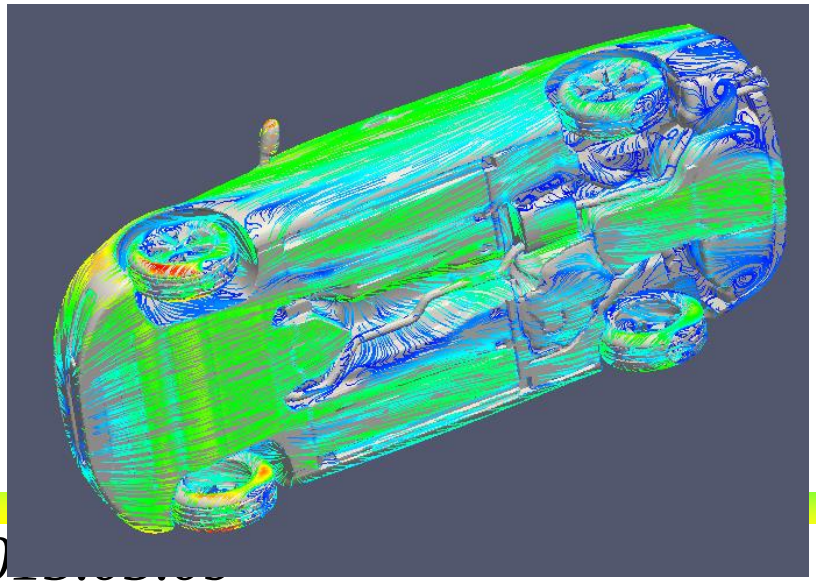


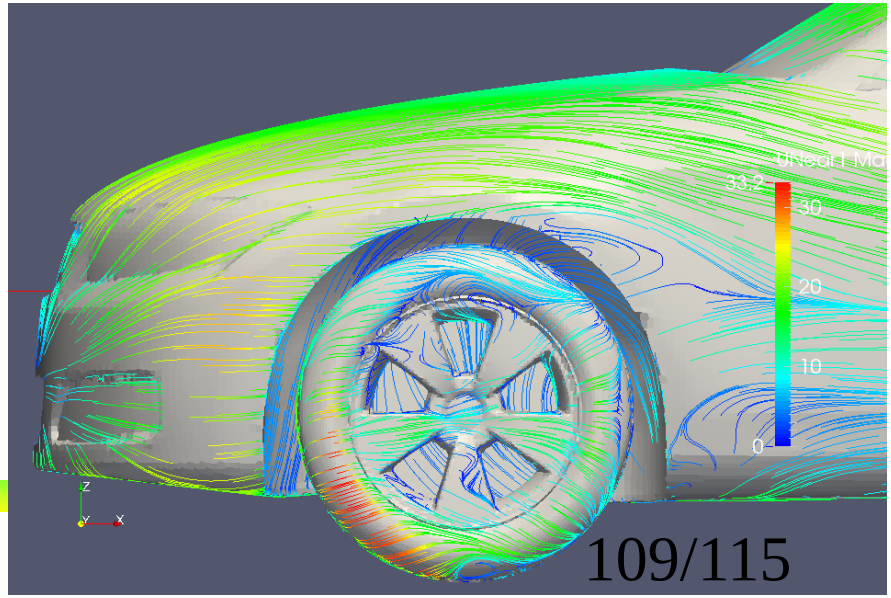


詳細モデル：事例2 タイヤ回転なし

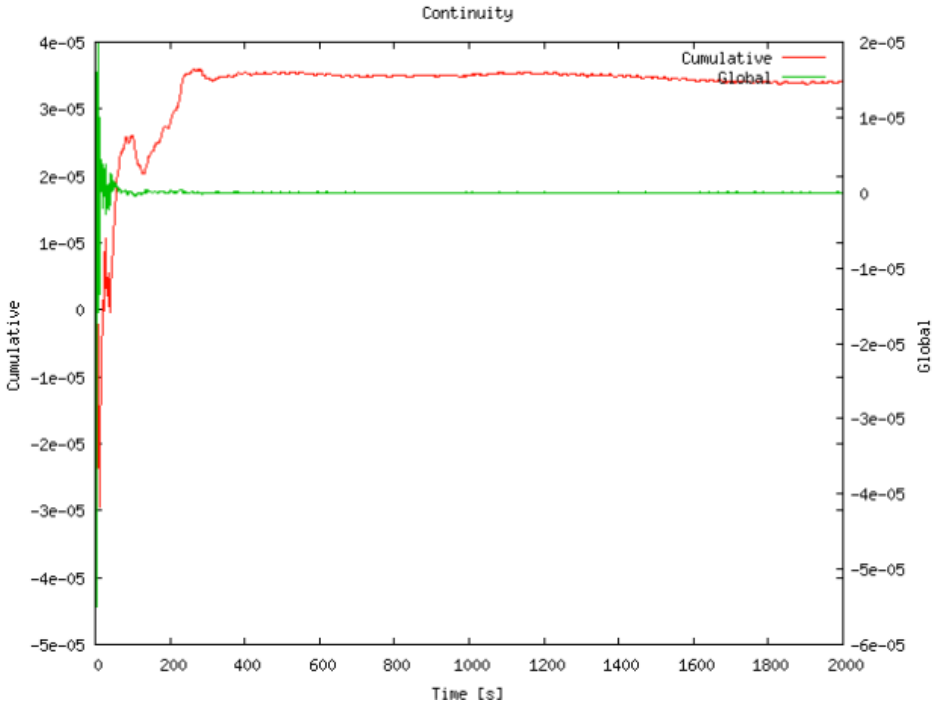


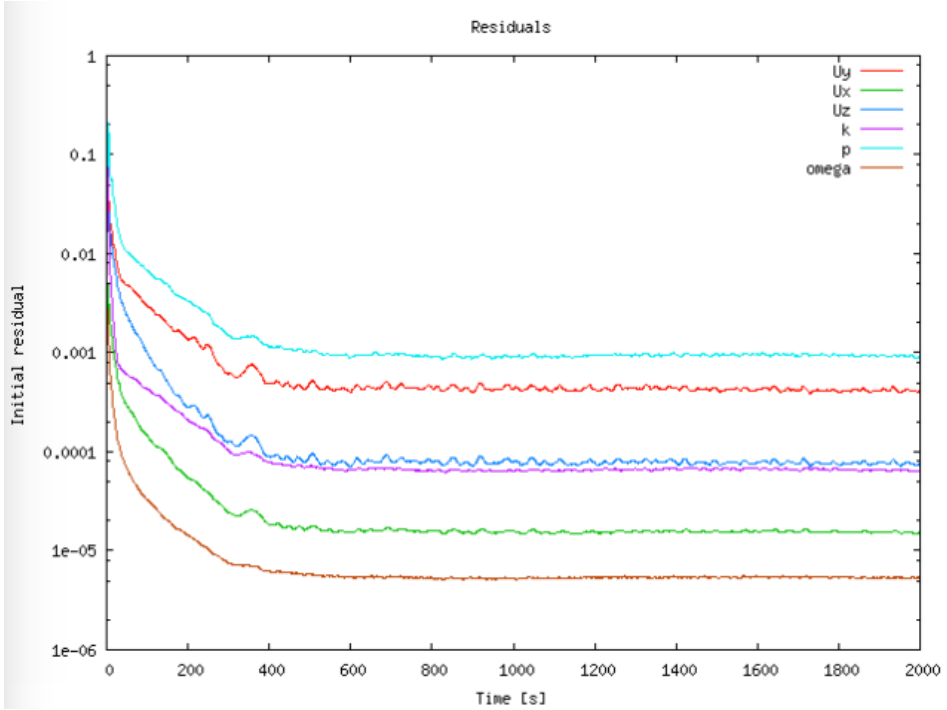




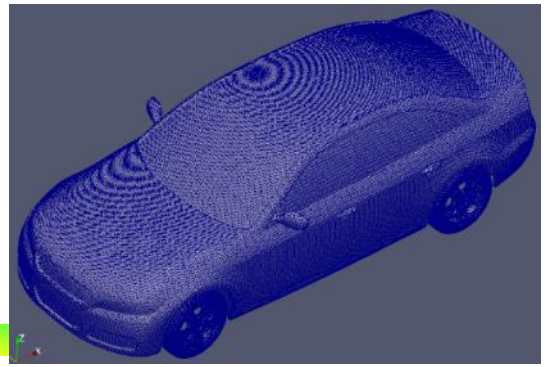


詳細モデル：収束状態（事例2 タイヤ回転あり）

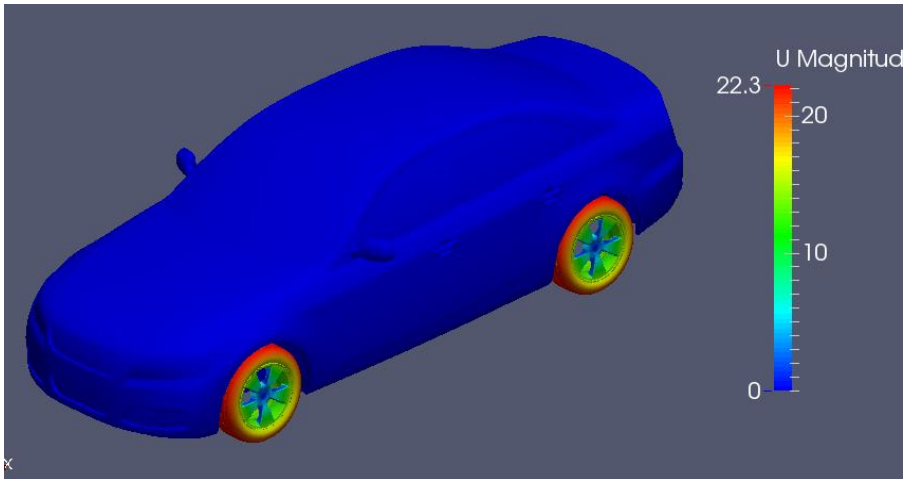


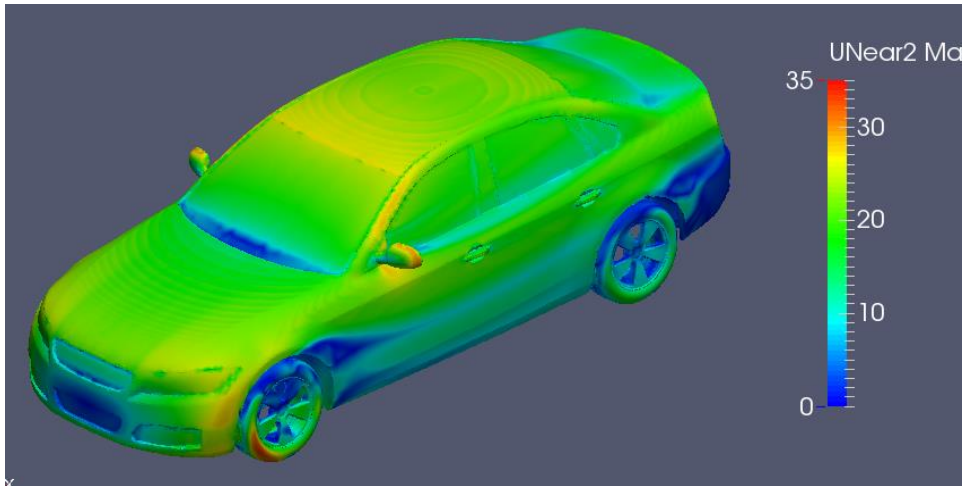


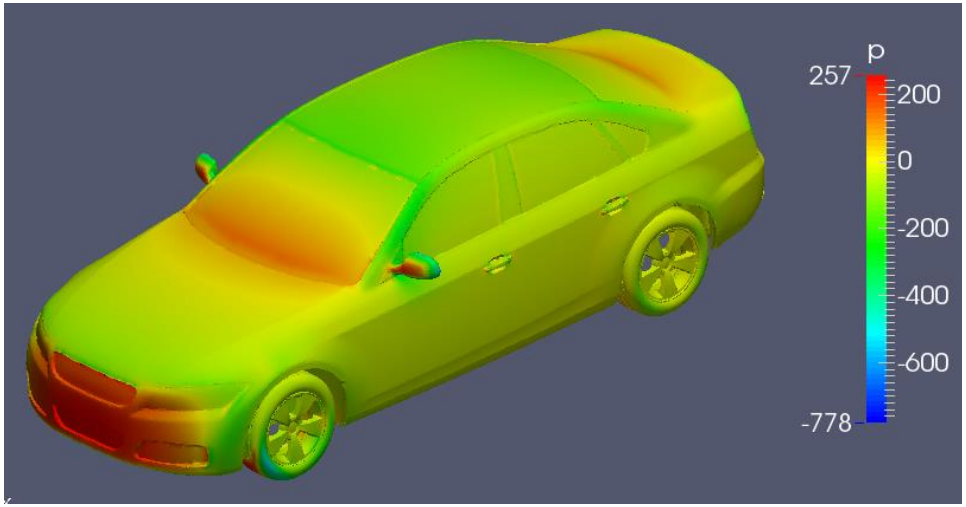
計算時間：9401秒（2000ステップ

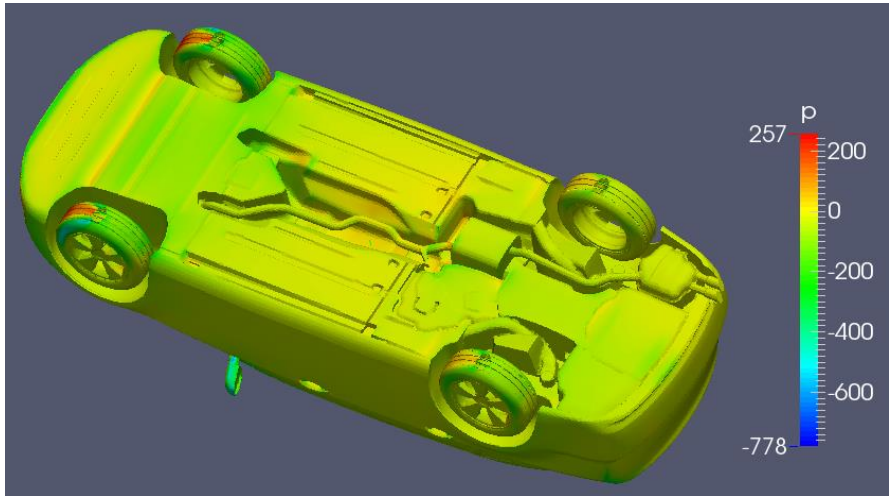


詳細モデル：事例2 タイヤ回転あり

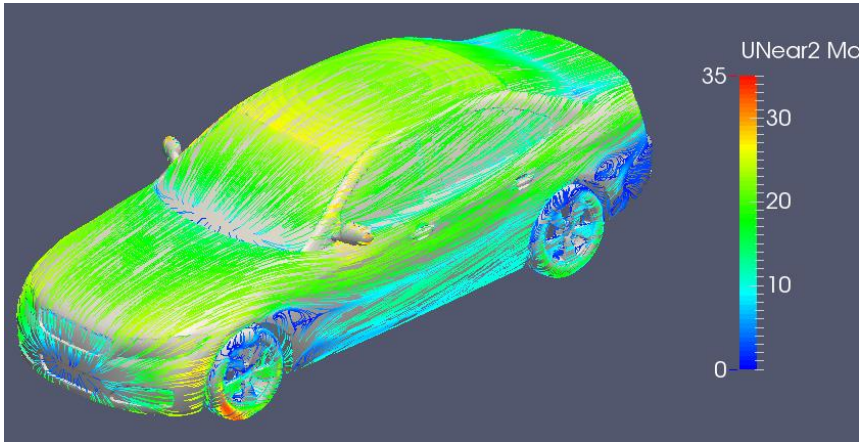


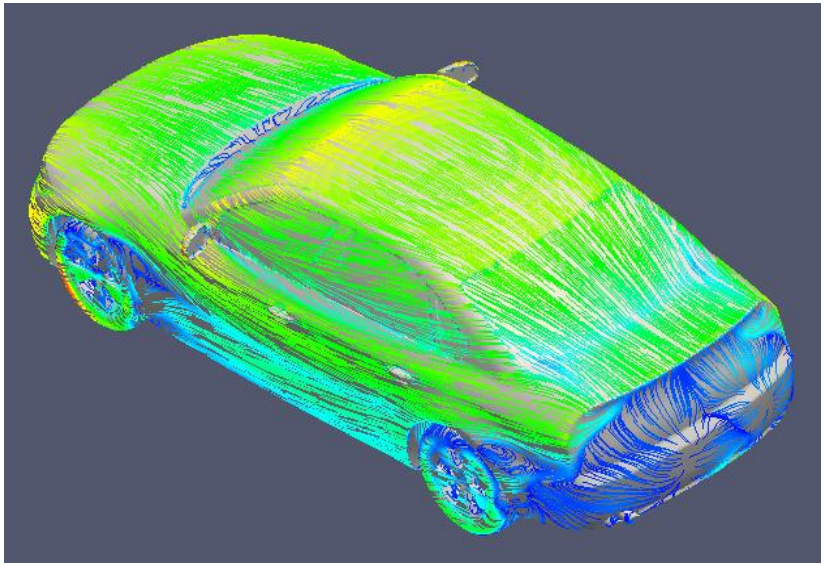


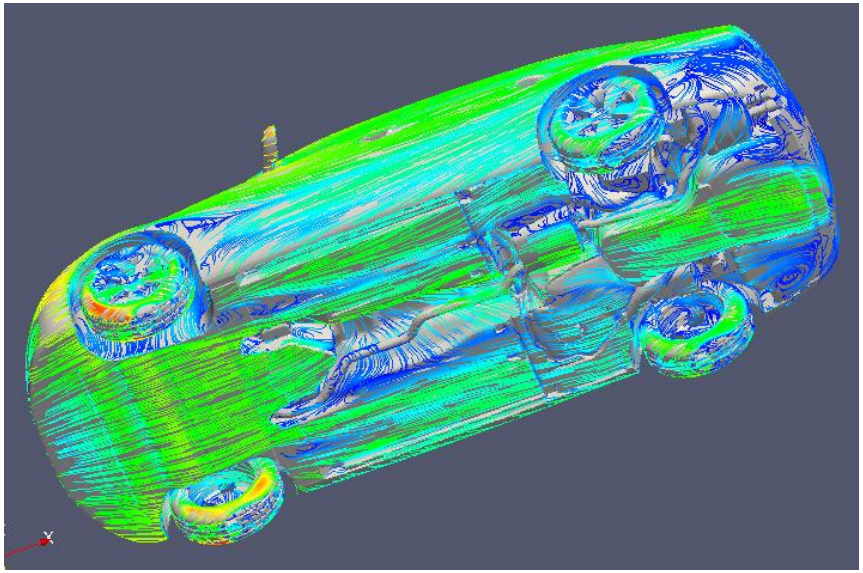


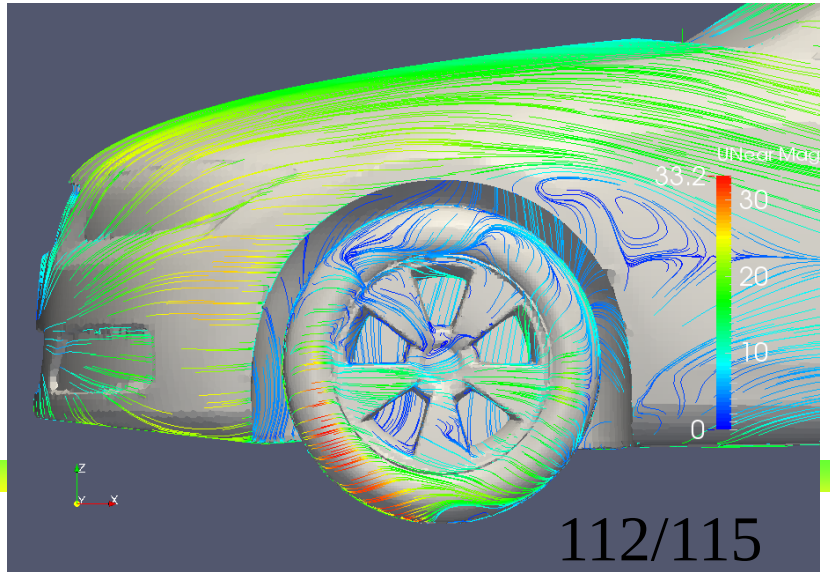


詳細モデル：事例2 タイヤ回転あり



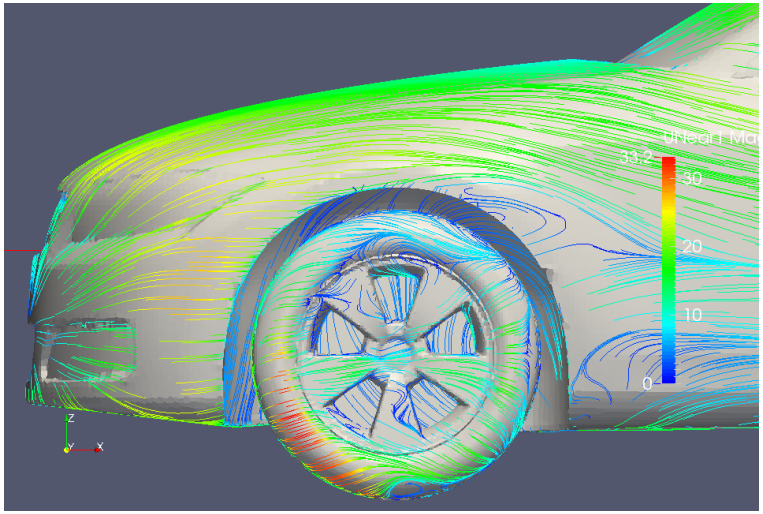


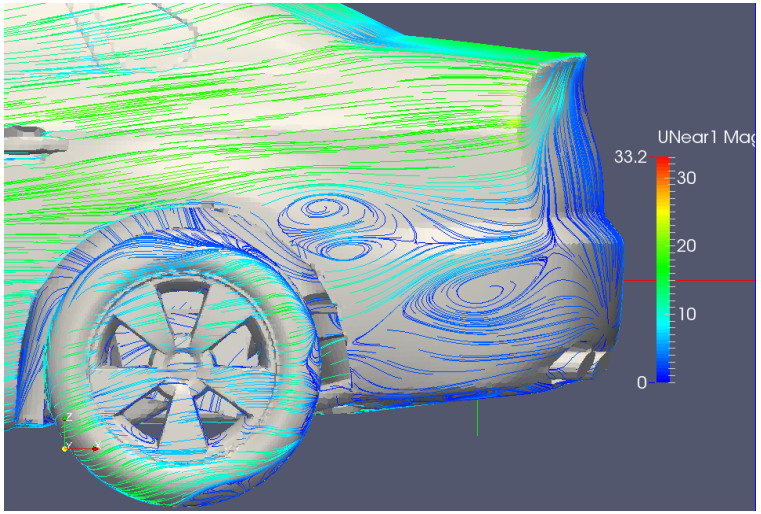




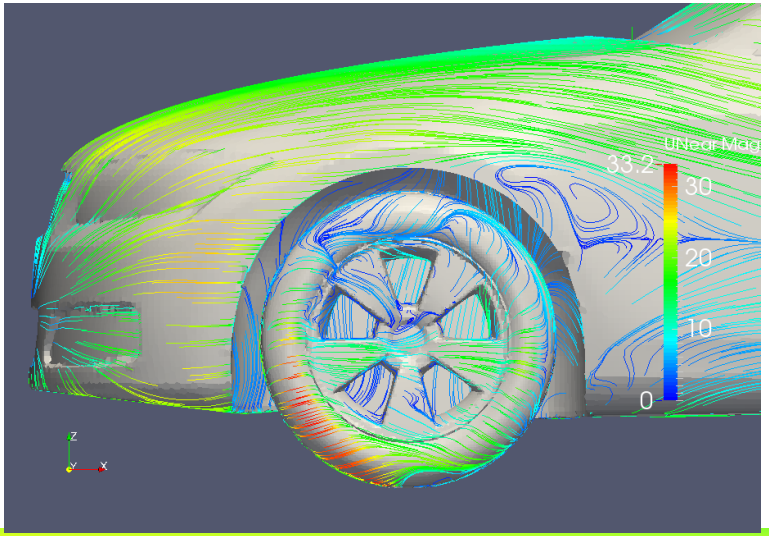
詳細モデル：事例2 タイヤ回転なし，ありの差

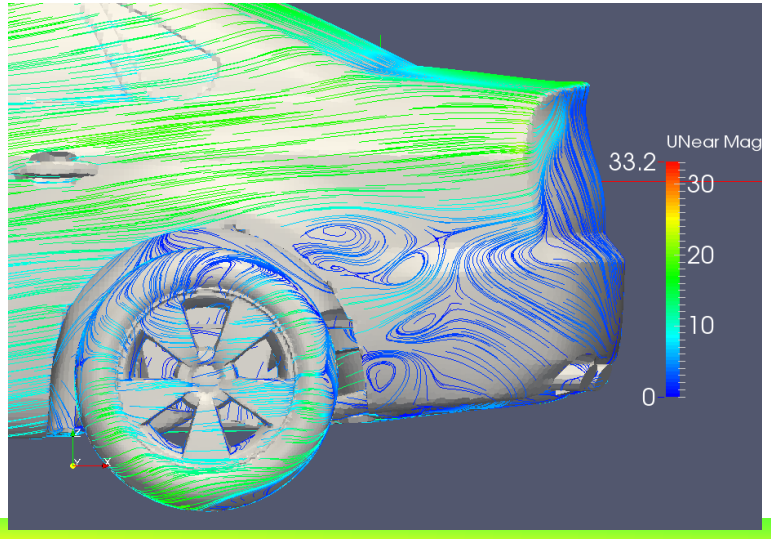
回転なし





回転あり





６．まとめ

６．まとめ

・公開されている自動車モデルを利用して，OpenFOAMにおけるモデリング，計算設定，計算結果の事例を示した。

・snappyHexMeshでメッシュを作成する場合，設定パラメータによってはメッシュ品質が問題になる場合がある。

・メッシュ状態によっては，表面近傍の流れ速度を作る機能が上手く働かない場合がある。