

学生フォーミュラの空力開発における 計算資源不足による結果への影響について

サウサンプトン大学 航空宇宙工学科
田崎 雄大

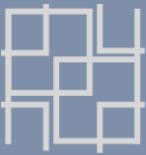
目次

1. 背景

2. 設定の範囲

3. 結果

4. 結論



課題

- CFDを正しく理解して扱うのはハードルが高い：乱流モデルの物理的背景など
- 計算資源がいる：車全体を正しいメッシュ数で回すと1億4千万セルとかになる
- Softwareへのアクセスがない：STARCCM+が使えない、Ansys Fluentのメッシュ数に制限があるなど

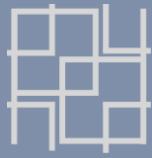
→ それでも空力開発にCFDは必要

CFDを通じて得られる情報

- コンセプトの比較
- SCL, SCD, 空力バランスの絶対量
- エアロマップによる姿勢への空力特性の調査
- パラメータ最適化

→ 少ない計算資源で何が正しく得られる？

空力パッケージの設計サイクル



理想（ざっくりとですが）

① ラップタイムシムなどをもとに目標決定



② モデリング



③ CFDの結果は目標を達成してる？  ②に戻る

Yes



Yes



④ 風洞実験は同じ傾向を示す？  ②に戻る & CFDの設定の評価

No

Yes



⑤ 製作して搭載



⑥ 実車の挙動はよくなった？  ①に戻る & 風洞実験の評価 & CFDの設定の評価

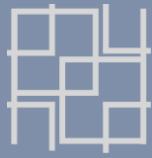
No

Yes

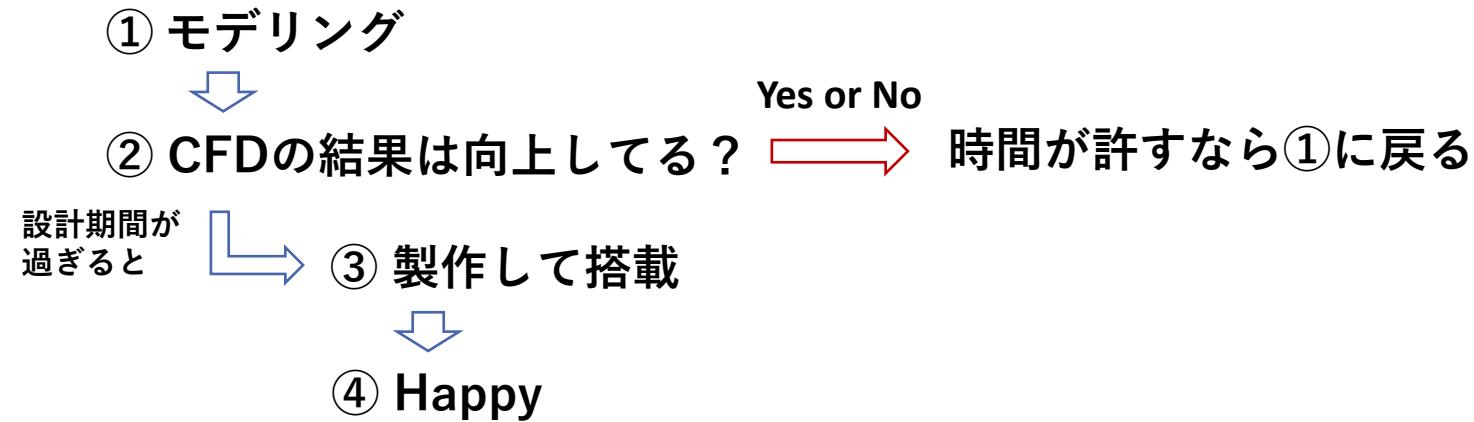


Happy & 次の改善点を検討し同じサイクルを回す

実際はこんなうまくいくわけもなく



現実（※僕の妄想です）



問題点

- 前提であるCFDによる相対評価が正しいという仮定が崩れると成り立たない
- CFDの正当性を確認する時間、および方法がない（CFDが正しい確証があるならこの話は参考にならないかも）
- 空力性能の評価を正しく行えるフェーズが存在しない（風洞の話はあとで触れます）

CFDが間違ってた時の具体的な影響は？



相対比較への影響

- 流れ場が正しく評価できないと、**間違ったコンセプトに誘導されてしまう**
- サイドポッドの流入＆タイヤの後流などはCFDの結果に大きく依存する

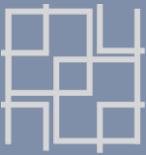
絶対量のミスリードによる影響

- Lap time simなどでエアロの正当性を評価するとき、SCLが真値と違うと**結論が変わる**かも
- 空力バランスのミスリードによる空力的に不安定な設計

パラメータ最適化

- 剥離ギリギリのラインを攻めてるつもりが現実では剥離してる
- ガーニーフラップが機能していない
- ドライバーのパフォーマンスや気候の変動を考慮すると実車試験での細かい設定は現実的でない

重要さ



パフォーマンスチームからの要求

- エアロによる重量増20kgに対してタイムの短縮を正当化するにはSCLが-3必要
- L/Dが-2以上の時はダウンフォースに全振りしてもタイムが少なくなる

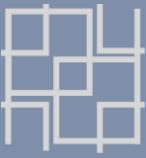
CFDを回すと

- 初期設計がSCL=-2.8, L/D=-2.2だったのでダウンフォースを増してL/D=-2, SCL = -3.2にした
- エアロつけることでタイムは短縮するはず

でもCFDが正しくなくて

- 変更後の真値はSCL = -2.6だった
- エアロつけることでタイムが悪くなる

結論が違ってくるのは困る



問題点

- 前提であるCFDによる相対評価が正しいという仮定が崩れると成り立たない
絶対値は間違ってても相対評価はできるかもしれない
- CFDの正当性を確認する時間、および方法がない（正しい確証があるならこの話は参考にならないかも）
お手本（RANSができる限界）があればいい
- 空力性能の評価を正しく行えるフェーズが存在しない（風洞の話はあとで触れます）

**じゃあお手本作って
ついでに相対評価がどこまで成立するかみてみよう**



目的

- 現実的なセッティングとその制限の提案
- データベースの作成

モチベーション

- 現状役に立つデータベースがない（日産の資料はあるけど、ちょっと情報が少ない）
- 計算資源と時間に余裕がないと、データベース作成ができない

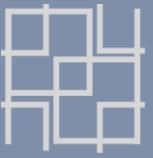
具体的なゴール

- 亂流モデルの比較
- メッシュ数による相対評価への影響の調査
- 使用するうえでの注意事項、および制約のまとめ



前提とする環境

- ソフトウェア : SolidWorks Flow Sim, Ansys Fluent, Openfoam, クラウド系のSimScaleなど
- 人的資源 : CFDのセッティングに最低 1 人割ける
- 計算資源 : Windowsのワークステーション or ノートパソコン = 8-16 CPU core
- 最大計算時間 : 24h / sim
- 知識 : 流体力学と数値計算の基礎、PythonとMatlab (ChatGPTでほとんど事足ります)



ここまでで質問、ご意見あればぜひ！！

(答えられるかわかんないけど)

目次

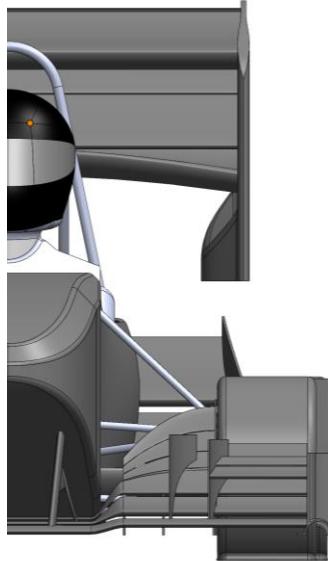
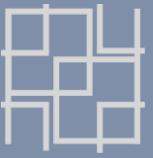
1. 背景

2. 設定の範囲

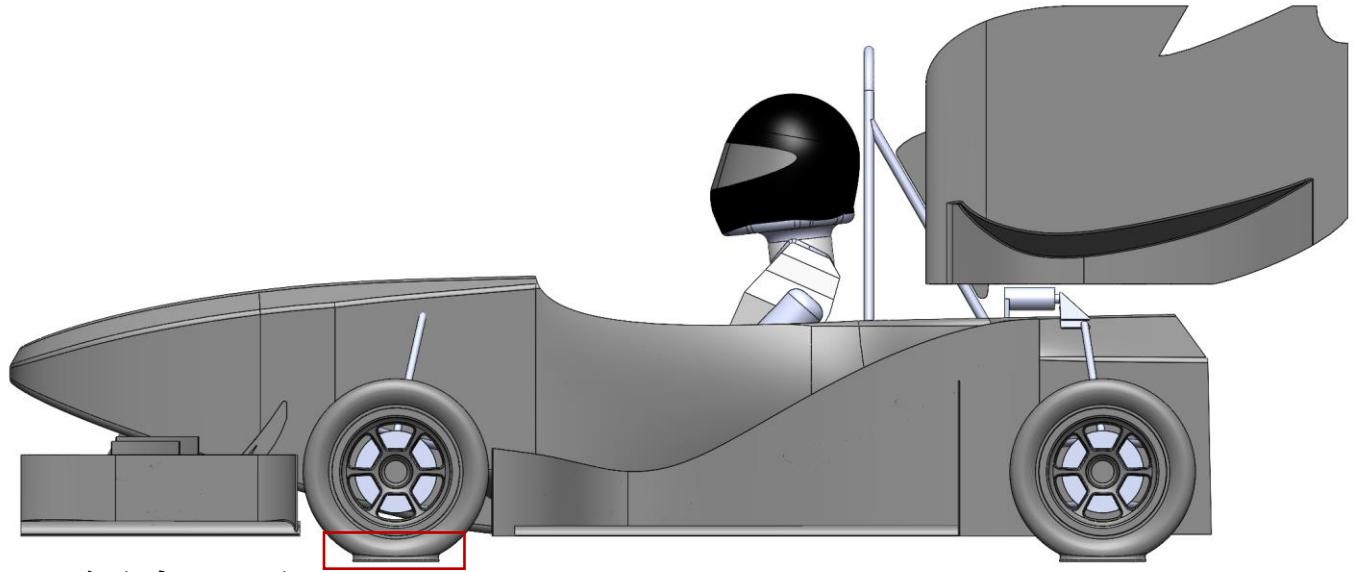
3. 結果

4. 結論

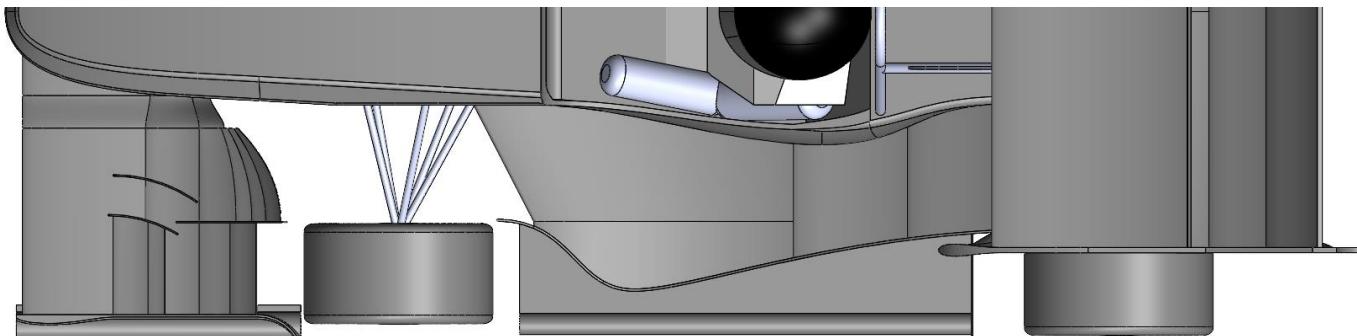
解析対象モデル



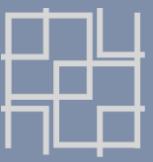
コンタクトパッチ（拡大）



コンタクトパッチ



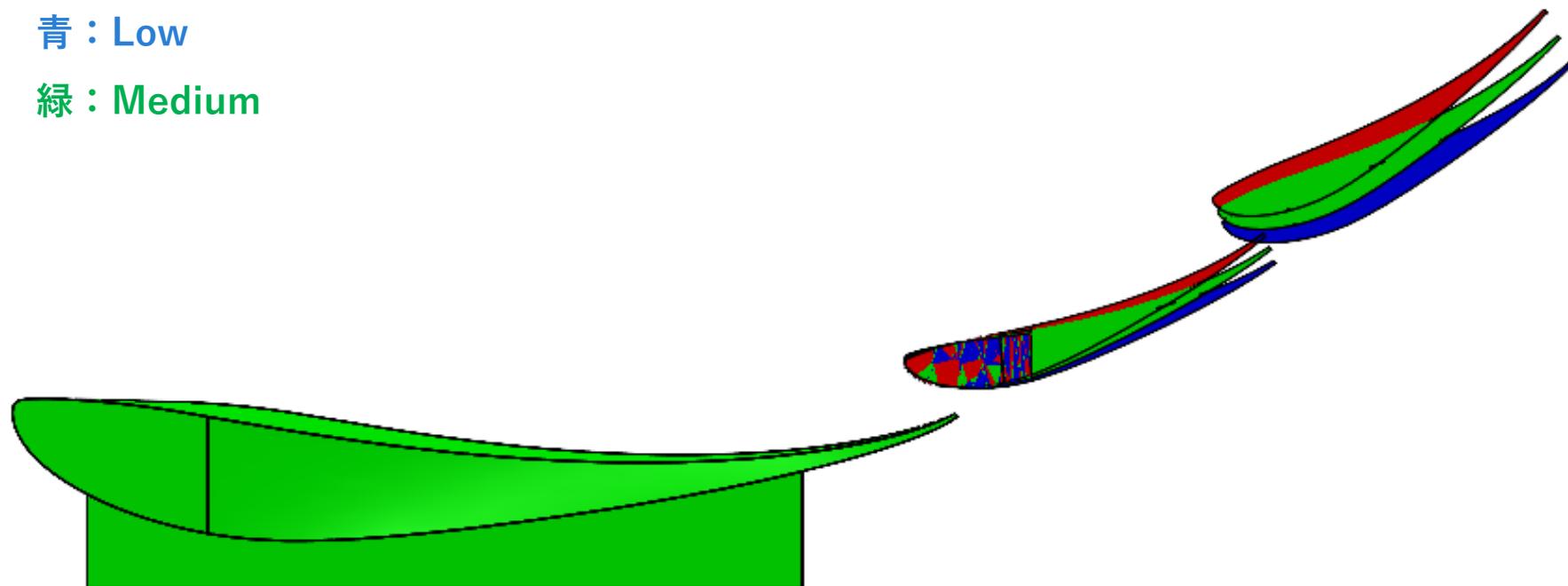
解析対象モデル2



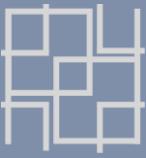
赤 : High

青 : Low

緑 : Medium



SWのAoA3パターン

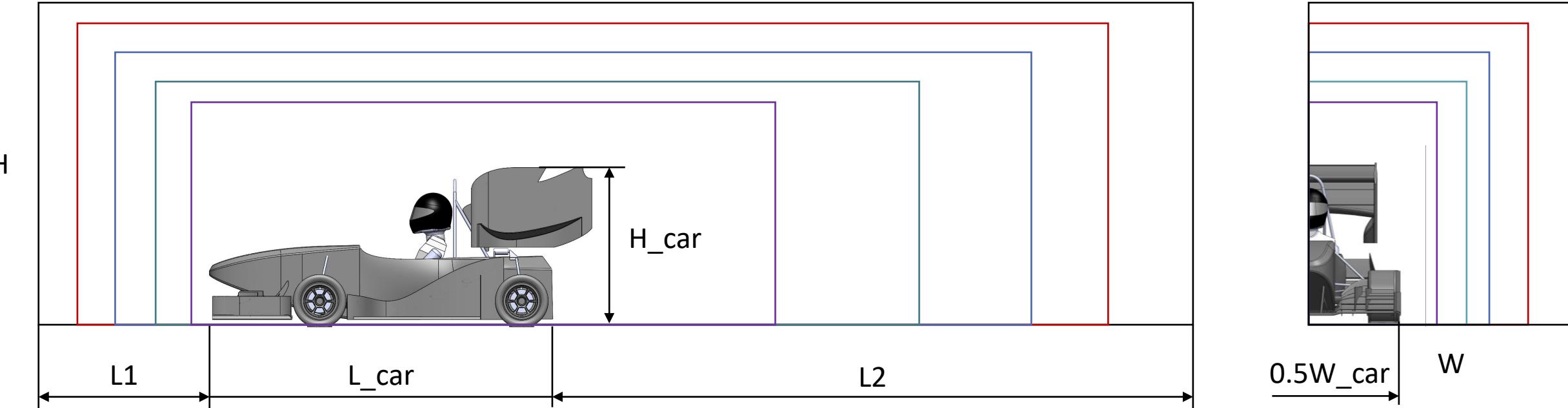


CFD用のCADモデル

- 細かい部分（ドライバーの指やエンジン、ブレーキディスクの穴など）は単純化する
- 小さい隙間はできるだけ減らす（翼と翼の隙間など）
- 翼の後端は鋭利ではなく、カットして平らにしておく
- 45度以下の鋭利な形状にはRをつける（つけないほうがうまくいく形状もある）
- 部品は少しボディが重なるように配置する（ピッタリだとたまにうまくいかない）
- タイヤの高さは0.1-1mm程度地面に食い込んだ設定にしておく（発散を防げる）
- Surface wrapperには極力頼らないようにする

Surface meshを表示してCAD形状の違いを確認しておくといい

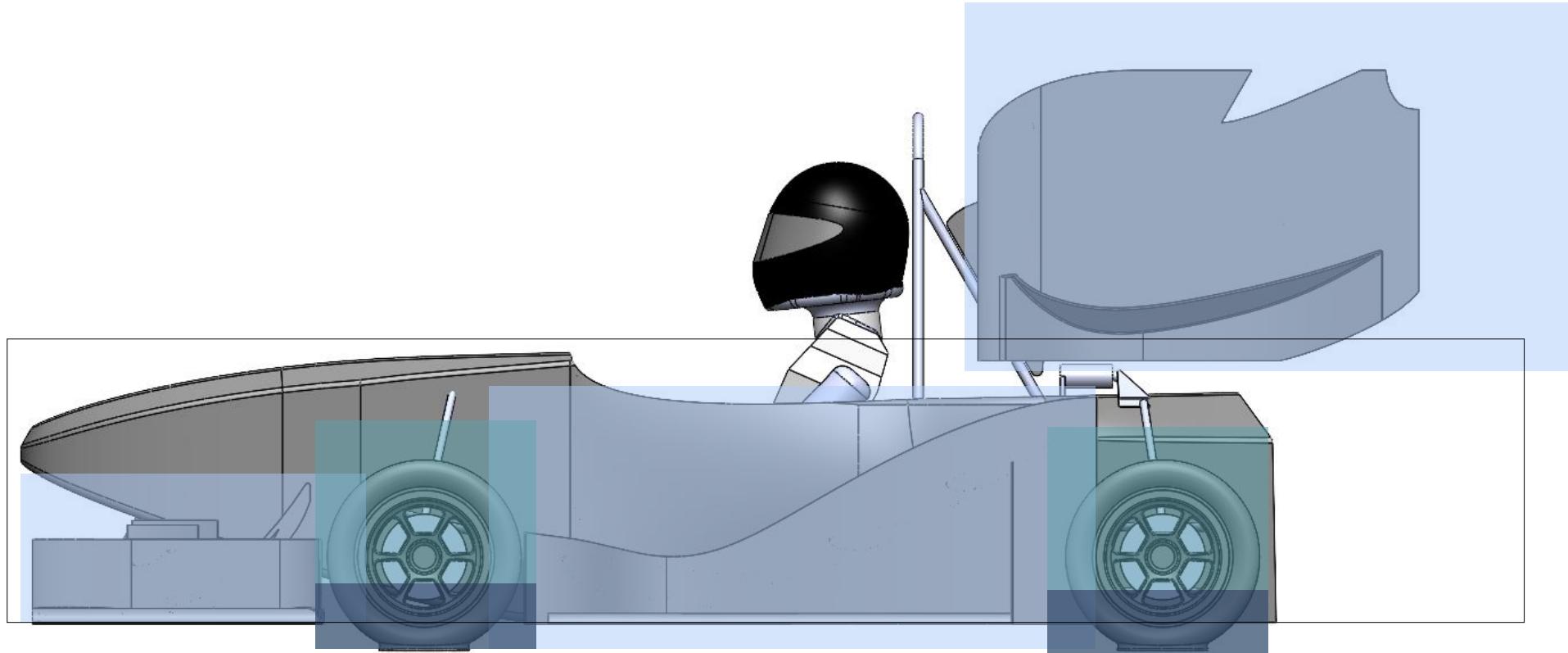
計算領域 1



	L_1	L_2	H	W	Cell number
Large Domain	$5L_{\text{car}}$	$15L_{\text{car}}$	$10H_{\text{car}}$	$10W_{\text{car}}$	40M, 55M, 70M
Medium Domain	$3L_{\text{car}}$	$11L_{\text{car}}$	$7H_{\text{car}}$	$7W_{\text{car}}$	20M, 30M
Small Domain	$1.5L_{\text{car}}$	$7.5L_{\text{car}}$	$4H_{\text{car}}$	$4W_{\text{car}}$	5M, 10M

各Refinementの大きさはメッシュサイズによって調整

計算領域 2

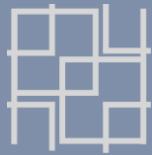


Aero refinement (小)

Tyre refinement (小)

Contact patch refinement (極小)

Chassis refinement (中)



使用するソフトウェア

- Siemens STARCCM+, (SolidWorks Flow Sim)

精度

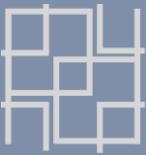
- 2次精度（対流項のみ風上差分）
- 数値は倍精度（Double precision）を使用

境界層の設定

- 厚み：40mm（シャシ）, 20mm（エアロ、タイヤ、ドライバー）, 5mm（サス、ロールフープ）
- 5M, 10M : $y^+ = 30 = 1 \text{ mm}$, それ以外 : $y^+ = 1 = 0.01 \text{ mm}$
- Prism Layerの層数は各メッシュごとに変更：最小が4, 最大が25

収束条件

- すべての残差が $10E-4$ 以下 or 1500 step（本当は2500ぐらい回したいけど、1か月ではちょっと時間が、、、）



メッシュ成長率

- Volume mesh : Very slow
- Surface mesh : Slow

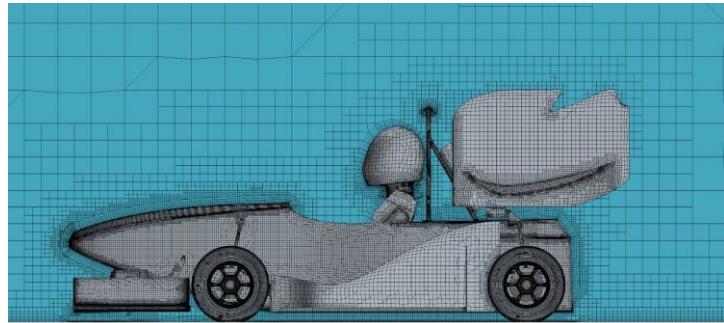
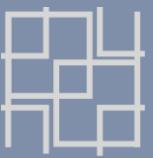
境界条件

- 直線走行を仮定し Symmetry とする
- Inlet : 10, 15 m/s, Turbulent intensity 5%, Viscosity ratio 10
- Outlet: Pressure outlet
- タイヤには接線方向の速度条件、床も 10, 15m/s
- 側面、および上面はすべり条件

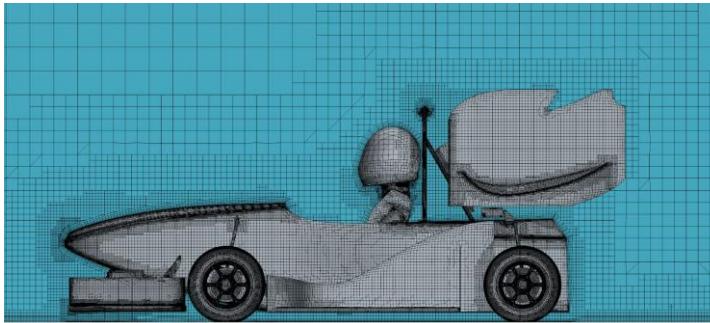
乱流モデル

- Realizable K-Epsilon
- K-Omega SST

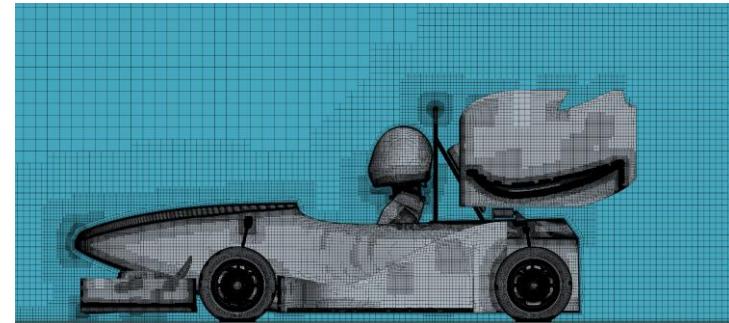
メッシュの密度と外観



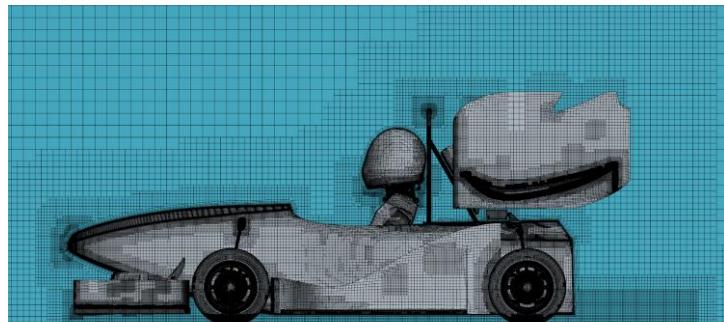
5M



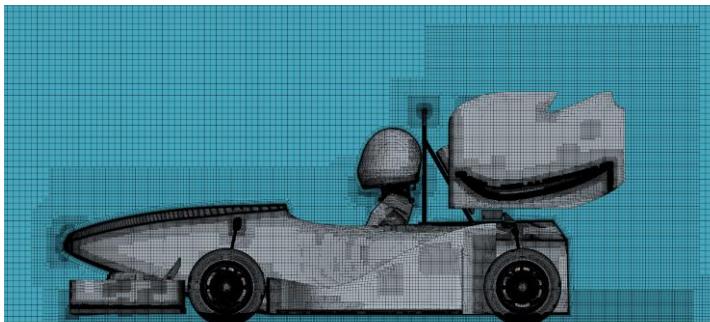
10M



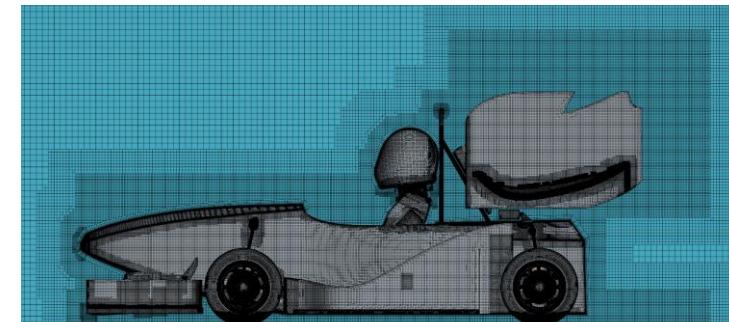
30M



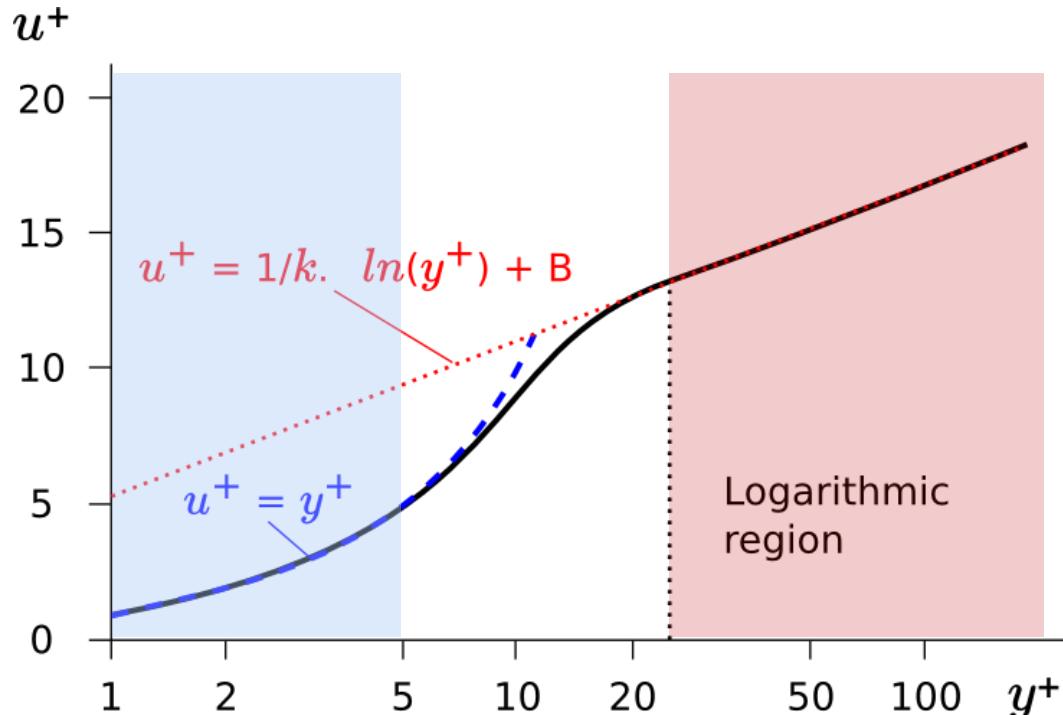
40M



55M



70M



境界層に支配される物理現象

- 壁面せん断応力 \propto 摩擦抵抗
- 流れの剥離 \propto 圧力抵抗, ダウンフォースに直結

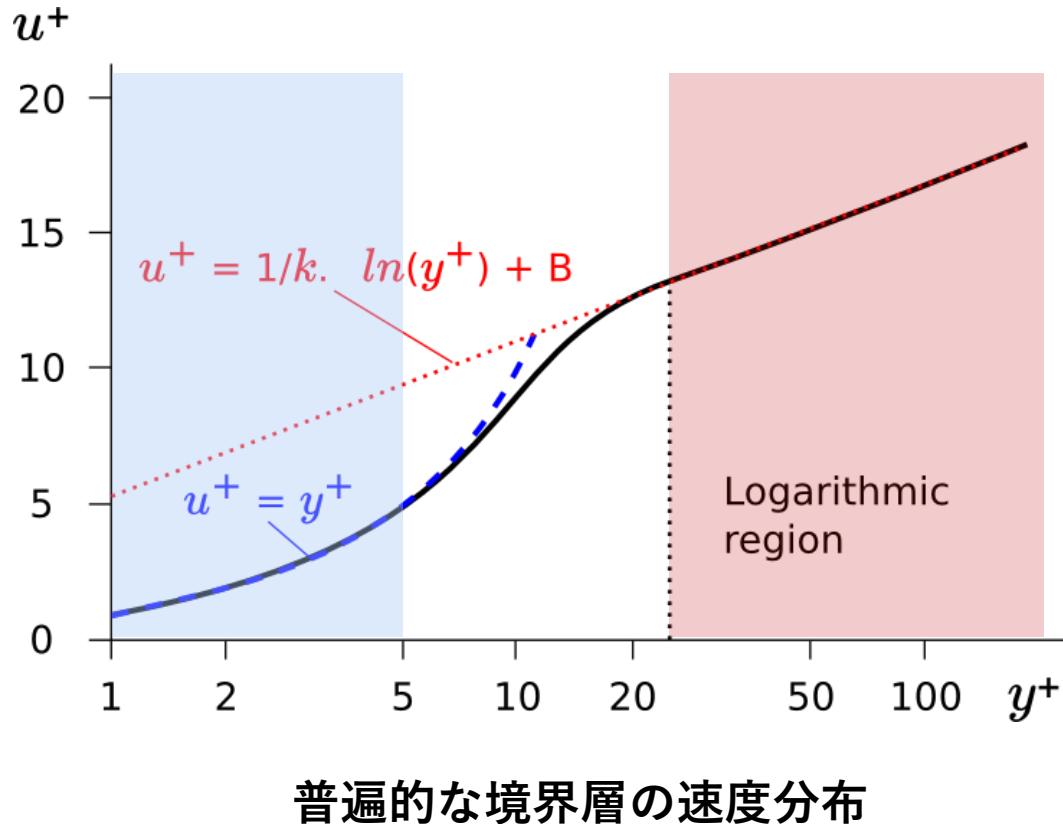
要するにレースカーで見たい現象全部

剥離を捉えるなら

- 初めのメッシュは $y^+ = 1$ 付近にほしい
- 具体的に $10 \text{ m/s}, 3 \text{ m}$ の平板だと 0.01 mm に相当
- y^+ は無次元数で流速に合わせて実スケールは小さくなる
- $y^+ = 5$ までは許容

より現実的には

- どの乱流モデルも DES と比較すると剥離予測ができない



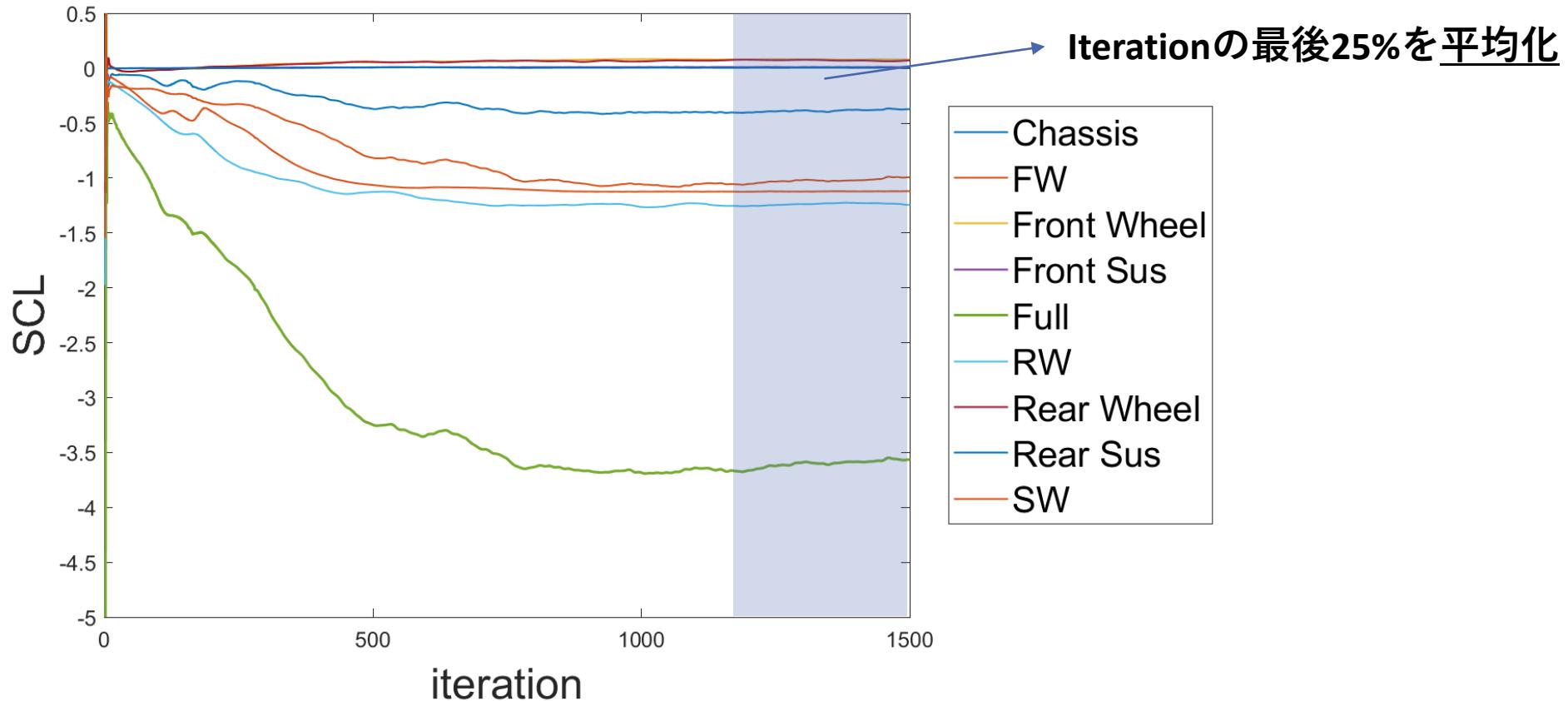
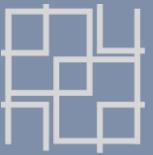
メッシュ的な話

- 第一層の厚みは指定可能
- 境界層厚みも指定する必要がある
- 残りは成長率かPrism layerの層の数
- 成長率は1.2以下が好ましい
- 20層ぐらいあると問題ない

計算資源の制約下における対処

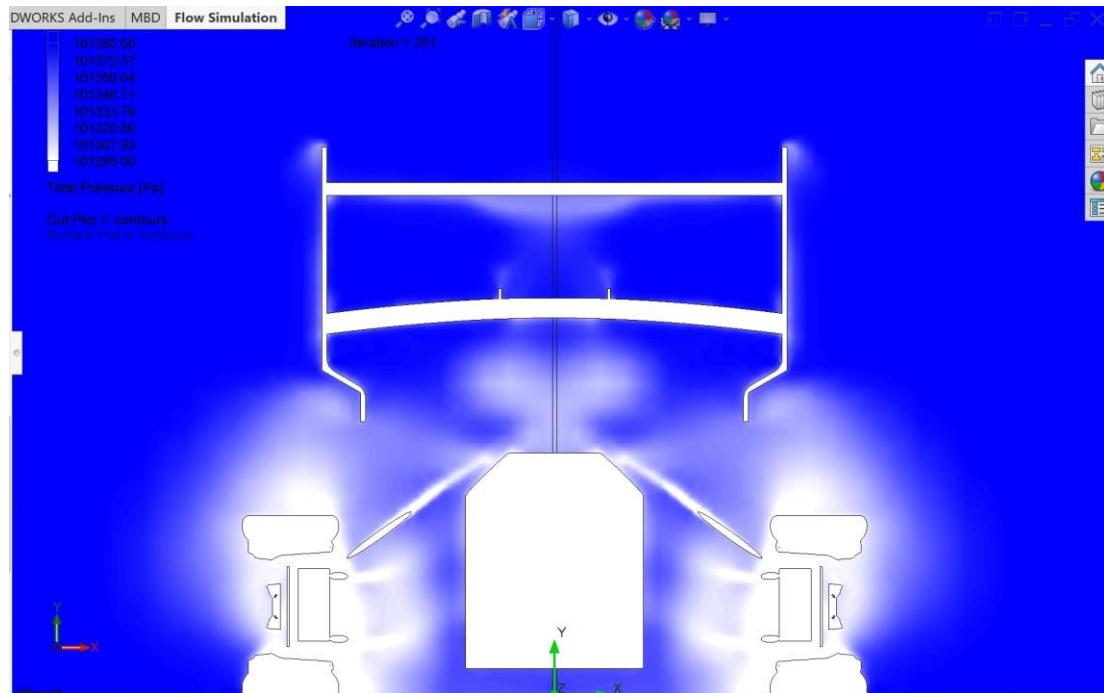
- y^+ を諦める
- 成長率を1.5などにする
- 境界層厚みを薄く設定する

数値の傾向と後処理

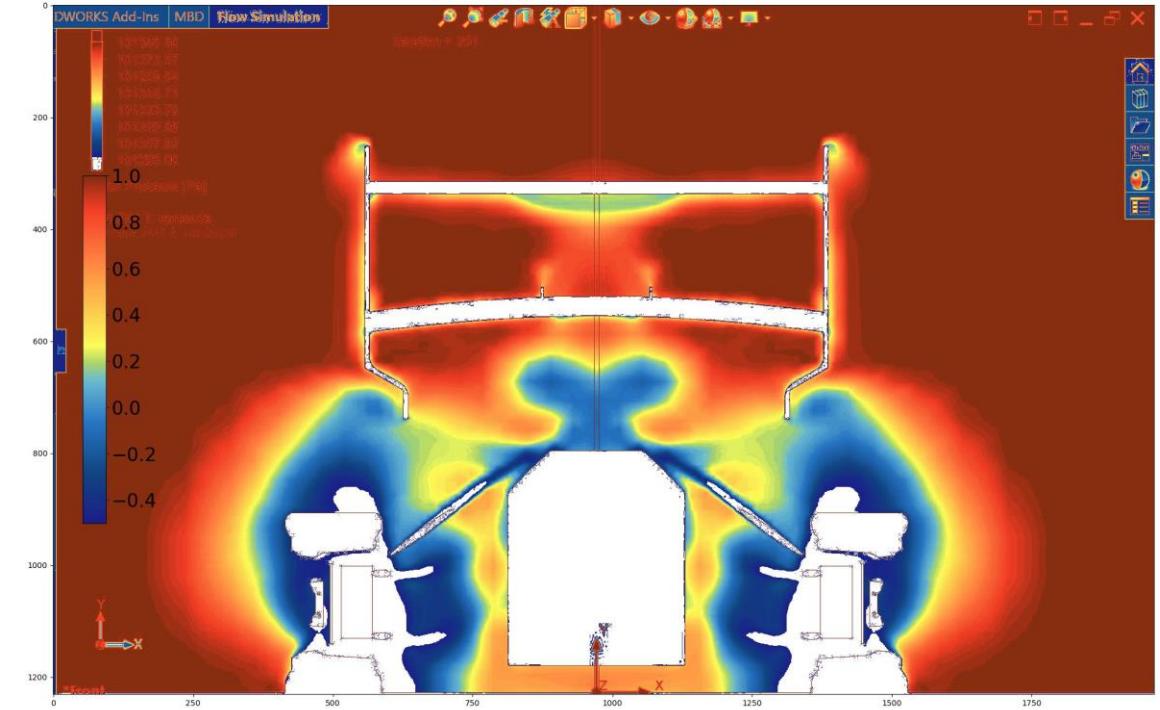


- 1000stepほどで収束状態に達する
- K-Omega SSTにはバラツキがあるがK-Epsilonはより滑らか
- 平均化処理は流れ場にも課されている (CpTやCpはiteration間の平均を出力)

カラースケールの後処理



CpTプロット変換前



CpTプロット変換後
タイヤの外の白は-0.5以下の領域

カラースケールの後処理

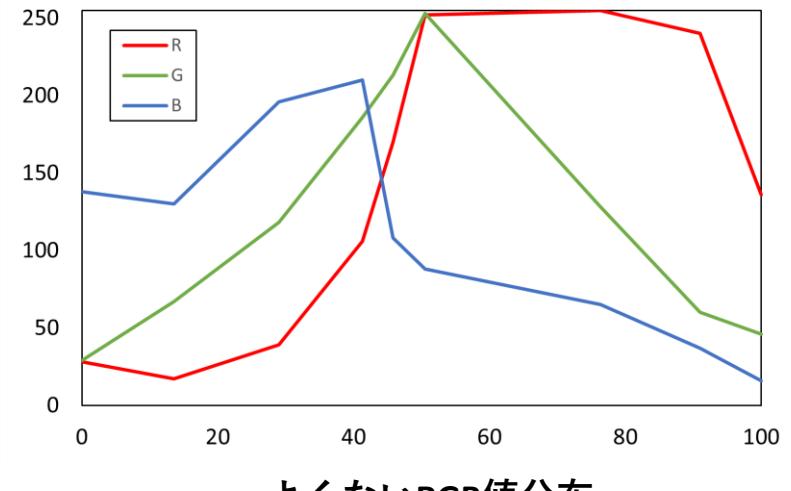


手順

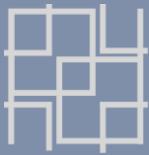
- 各ピクセルのRGB値を取得
- それぞれのRGB値からピクセルごとのスカラー値を推定
- 推定したスカラー値をもとにRGB値を変換

ポイント

- スカラー値を取得する誤差を減らしたい
- カラーマップのRGB値分布が単純なものを選ぶのがコツ（単調増加か単調減少が望ましい）
- だから青から白にRG分布が線形変化するカラーマップをSolidWorks側で選択した

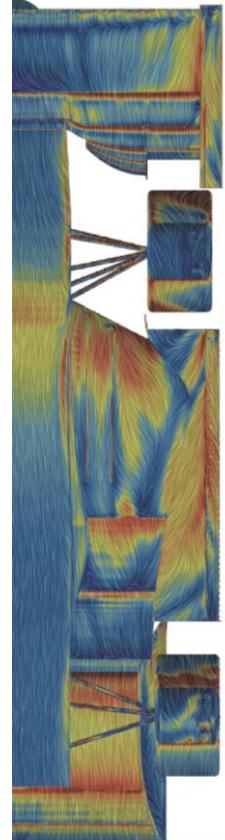


よくないRGB値分布

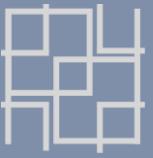


ソフトウェア的な制約

- 乱流モデルがK-Epsilonしかない：一般にK-Epsilonモデルは剥離流れの予測には不向き
- 境界層メッシュ（Prism layer）が存在せず、境界層はlog法則の壁関数でしか扱えない
- Windowsでしか回らない（HPCに突っ込めない）
- デフォルトの関数が少ない（抗力係数や揚力係数は自分で定義しないといけない）
- 用意されていない場の関数は定義できない（ $y+$ とかHelicityとか）
- 各スキームの詳細が不明（一次精度なのか二次精度なのか、緩和係数の設定など）
- Iterationに対する連続の式の残差が出てこない？（travelは対流時間）
- Adaptive mesh機能がない
- LICプロットはGPUないと出力してくれない
- 遅い（あと大規模計算のスケーラビリティが不明）
- デフォルトの指向性ライトの設定のままだとsurface plotに色の反射が反映される

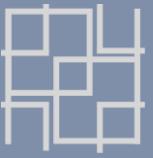


LICプロット



使用する利点

- 少ないメッシュでの計算が安定している
- 収束が早い
- 比較的操作が簡単



ここまでで質問、ご意見あればぜひ！！

(答えられるかわかんないけど)

目次

1. 背景

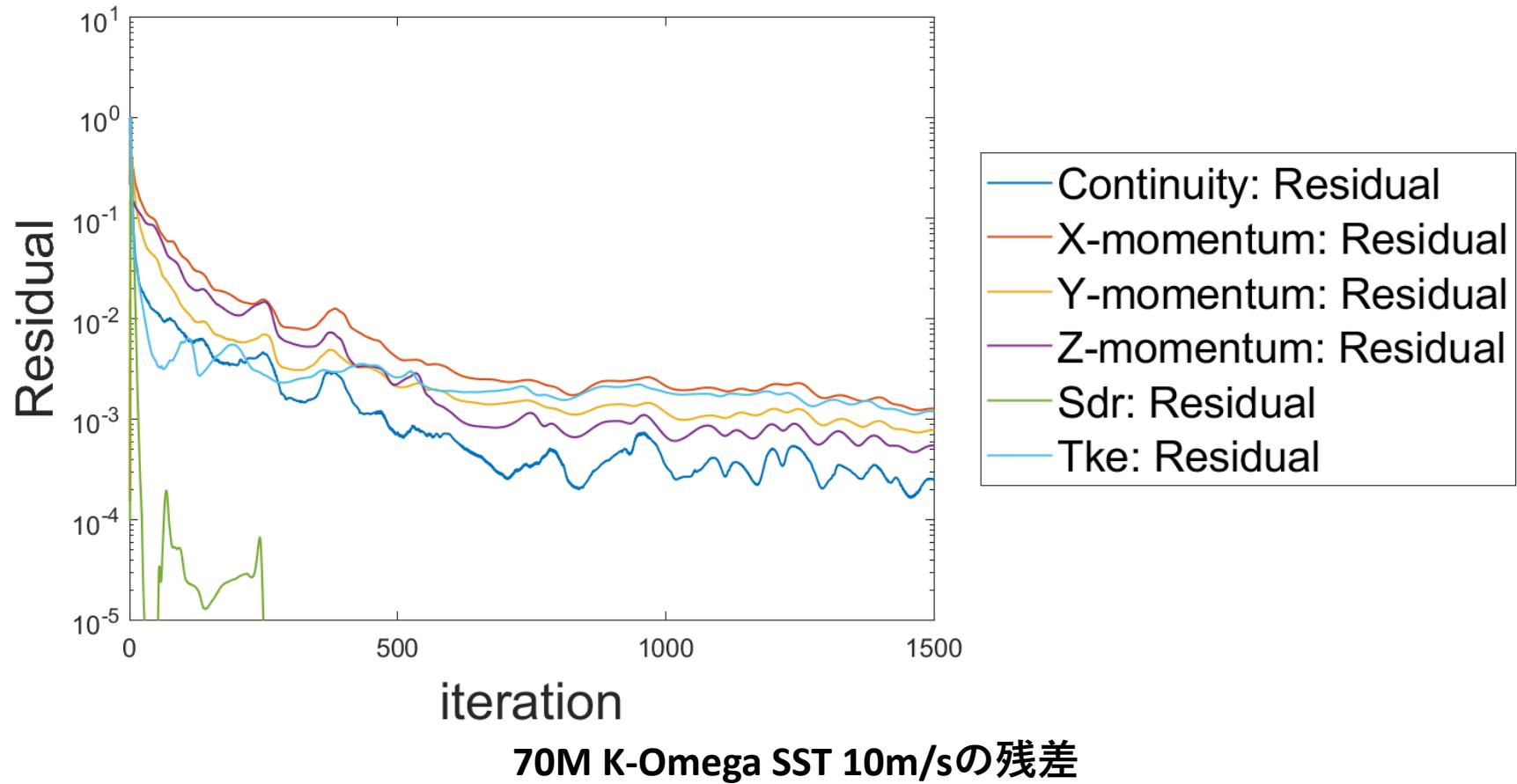
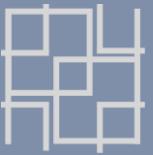
2. 設定の範囲

3. 結果

4. あの風洞について

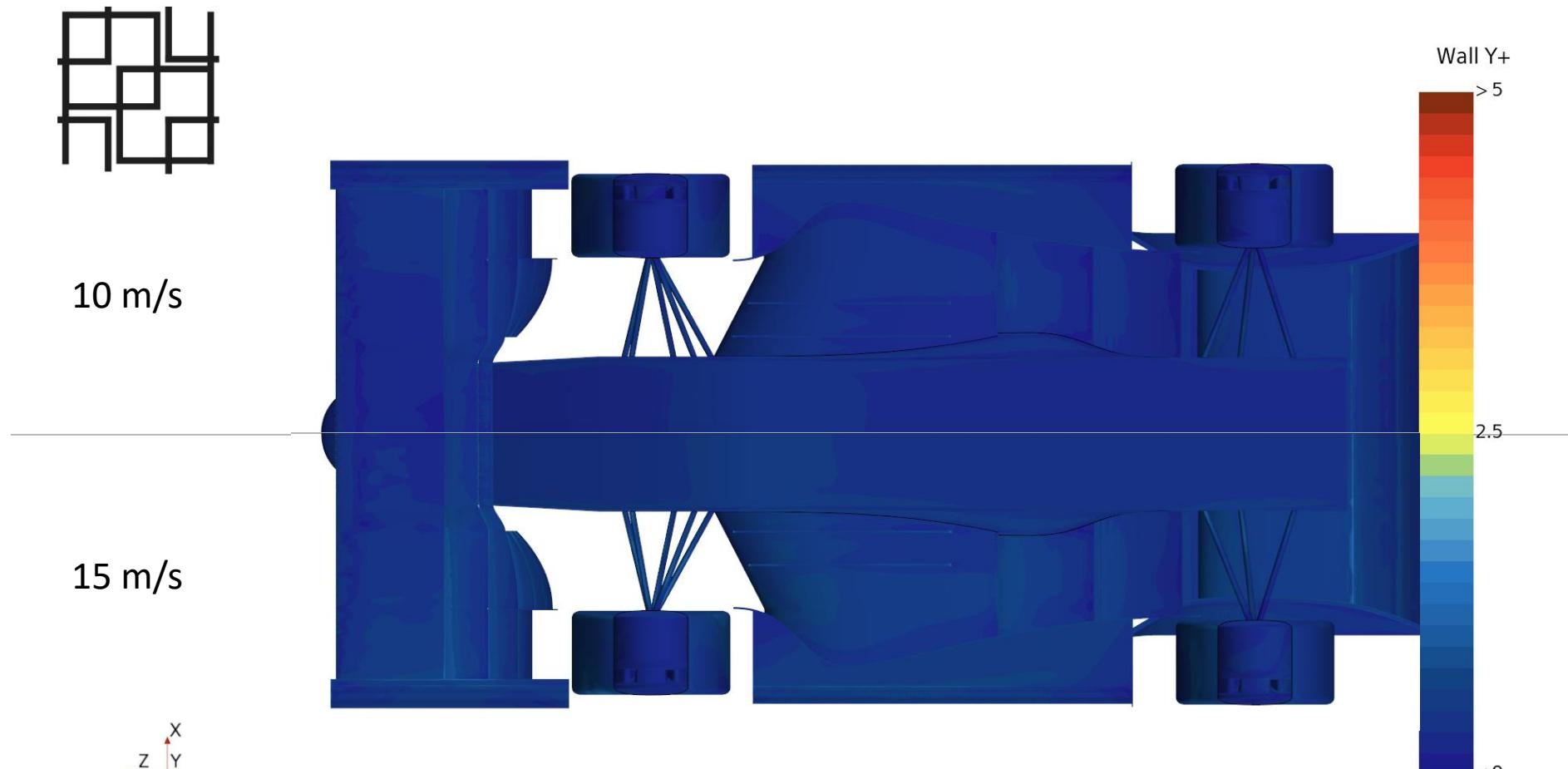
5. 結論

残差の傾向



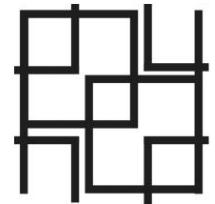
- 3Dで 10^{-3} のスケールまで落ちているので収束性は良好

Y+の確認 1



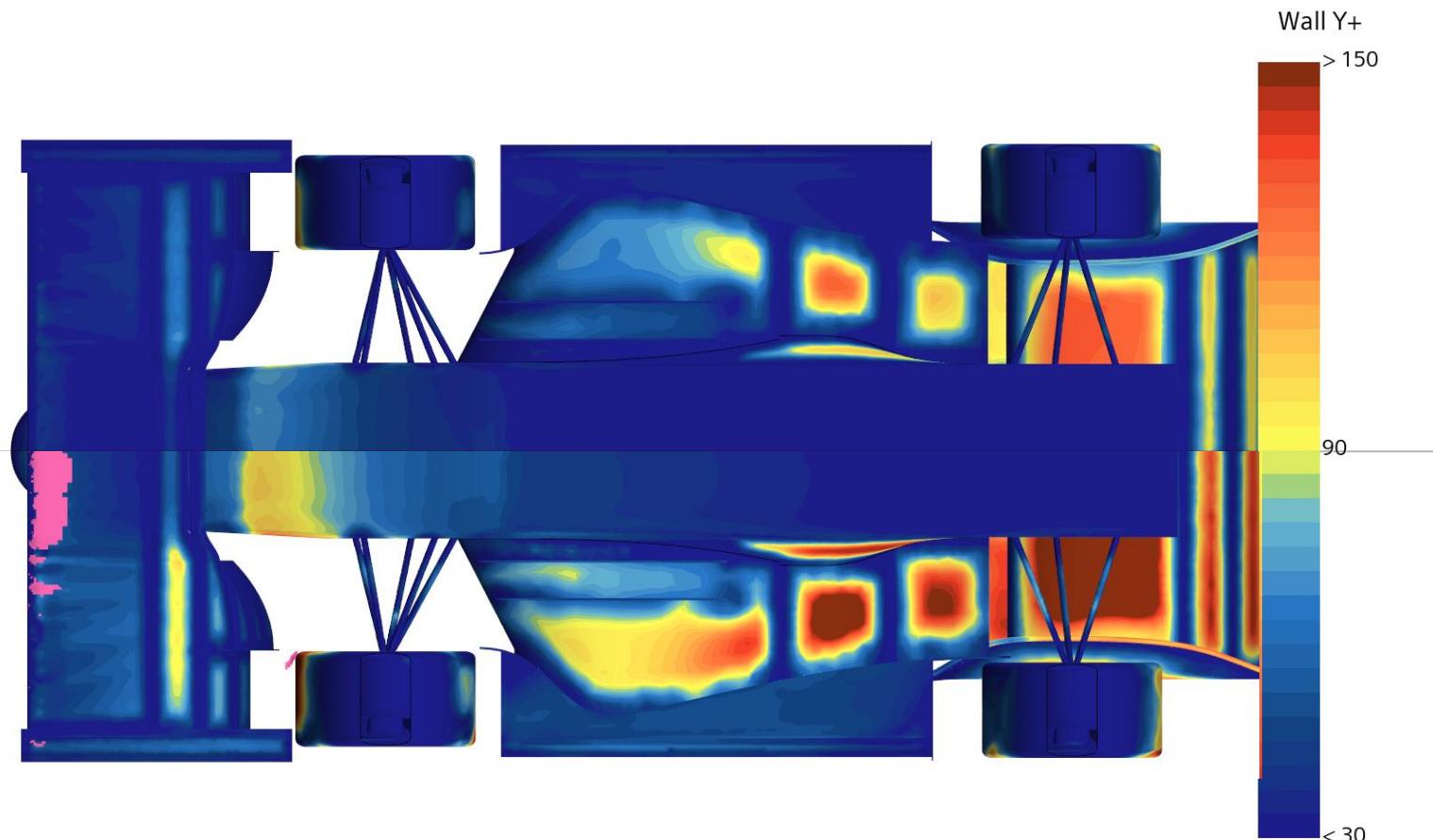
$y^+=1$ をターゲットにしている

Y+の確認 2

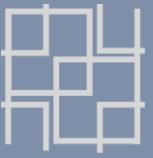


10 m/s

15 m/s

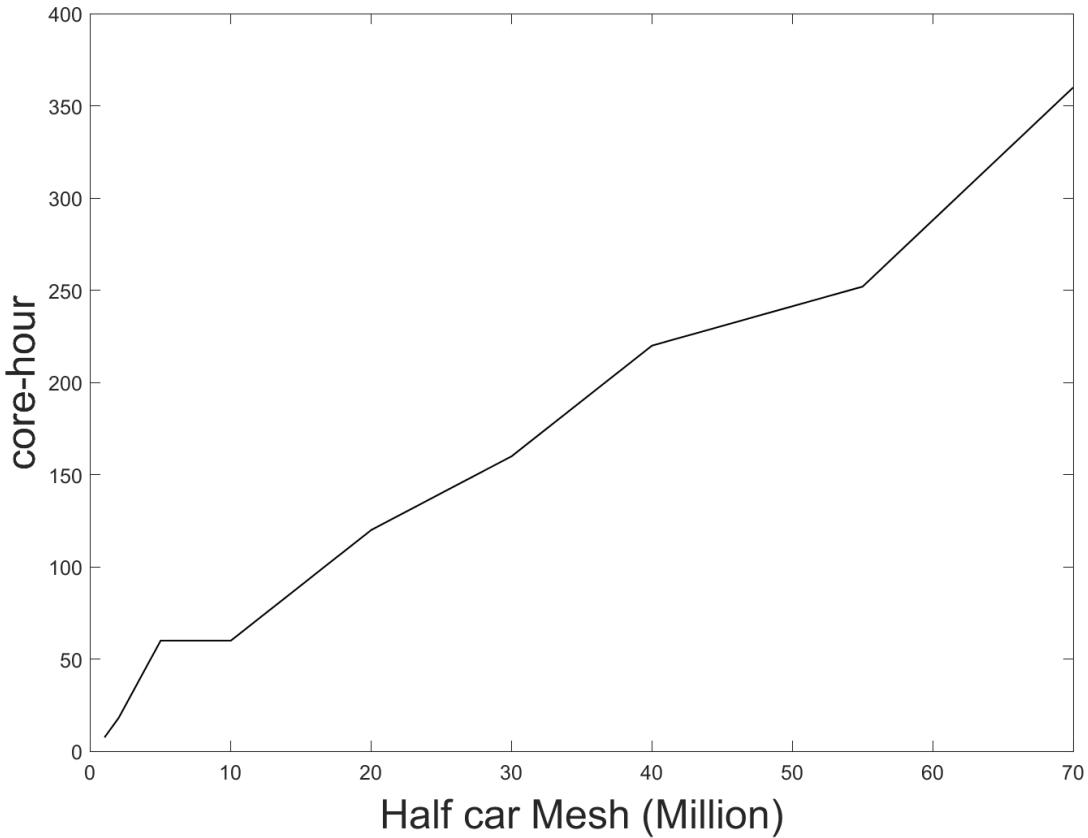
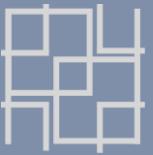


$y^+=30-50$ をターゲットにしている



領域サイズ

- 今回の計算で使用する値が最低限
- 小さくすると, SCLを大きく見積もる（最大で1程度変動する）
- 最近の論文ではメッシュ数は80M程度だが各辺10倍以上のドメインサイズを使用するものもある
- 211L(前:後=50:160) 200W 200Hとか



メッシュ数と計算コスト

わかること

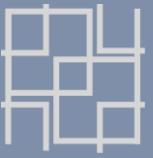
- 意外と線形
- SolidWorksのほうが相対的計算コストが高い

わからないこと

- 各ソフトウェアの大規模並列化における効率

気を付けること

- 並列化するにはそれなりのメモリが必要
- 12並列を超えるとメモリ32GBはほしいかも



CAD

- 形状をメッシュが切りやすいように調節する (6h)

メッシュ

- 設定して切ってみる (6h)
- 解が発散するので、発散するところをデバッグする (12-60h)

ソルバーと境界条件

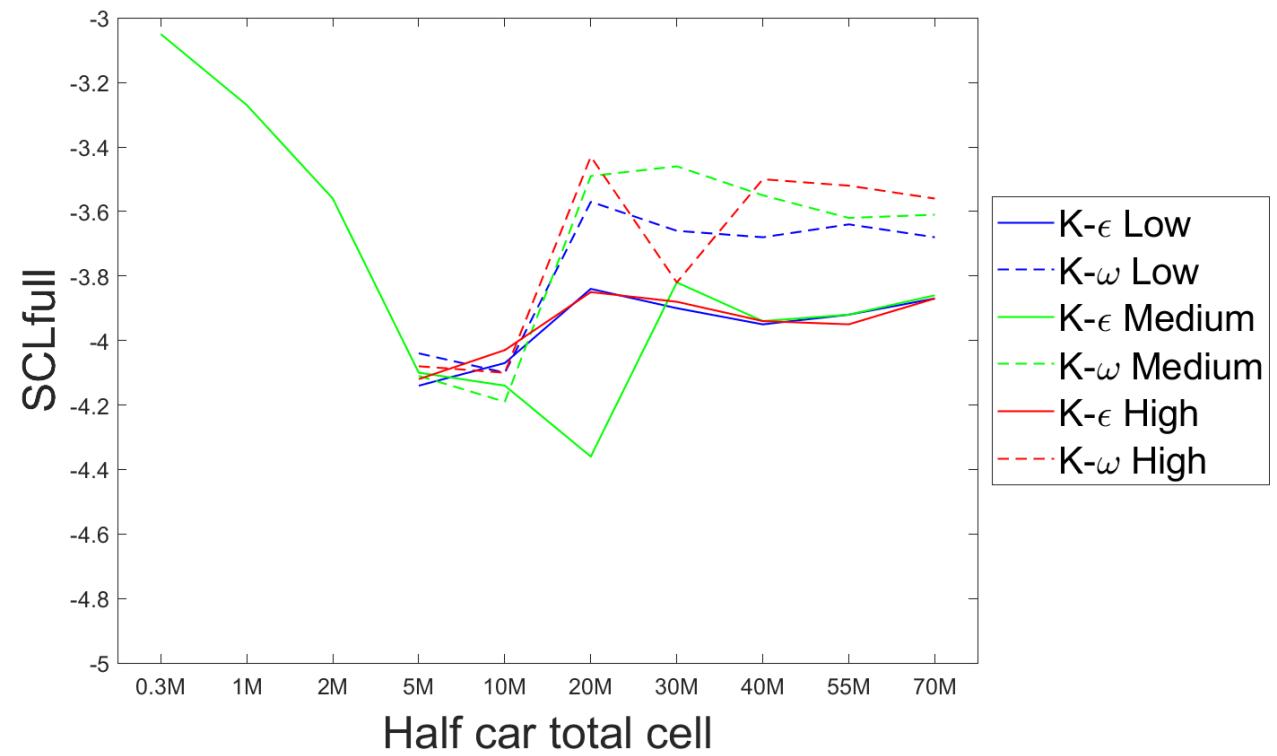
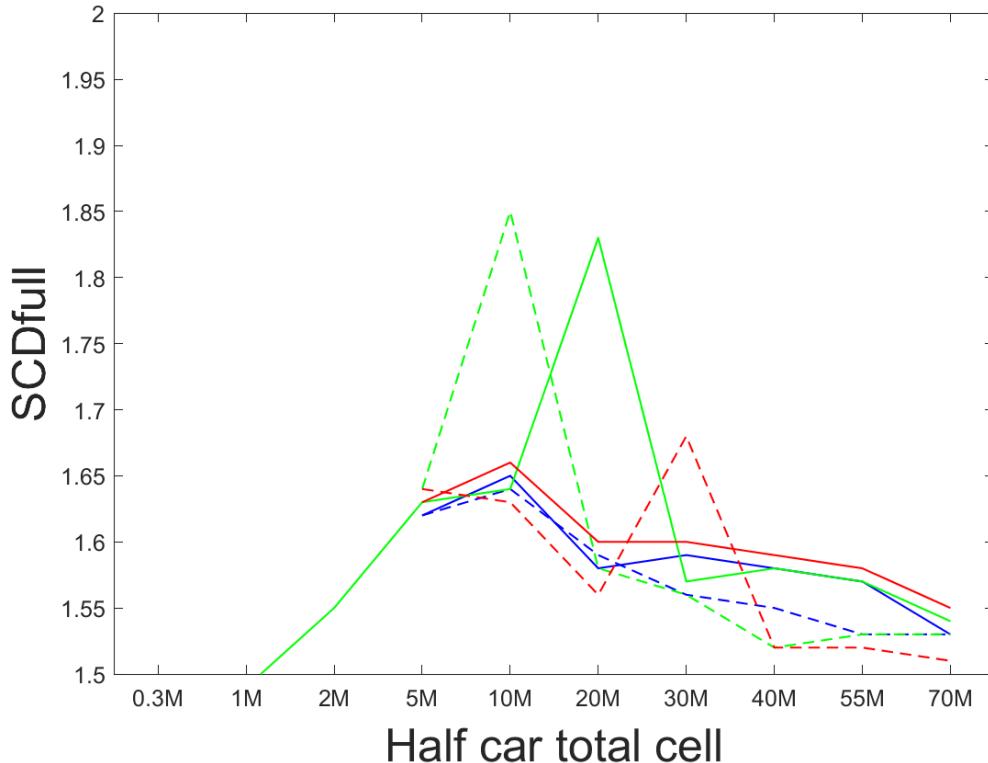
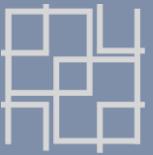
- 設定 (3h)
- 境界条件が意図したとおりに動いているか確認 (3h)

事後処理の自動化

- 設定 (3h)
- 意図したとおりに動いているか確認 (3h)

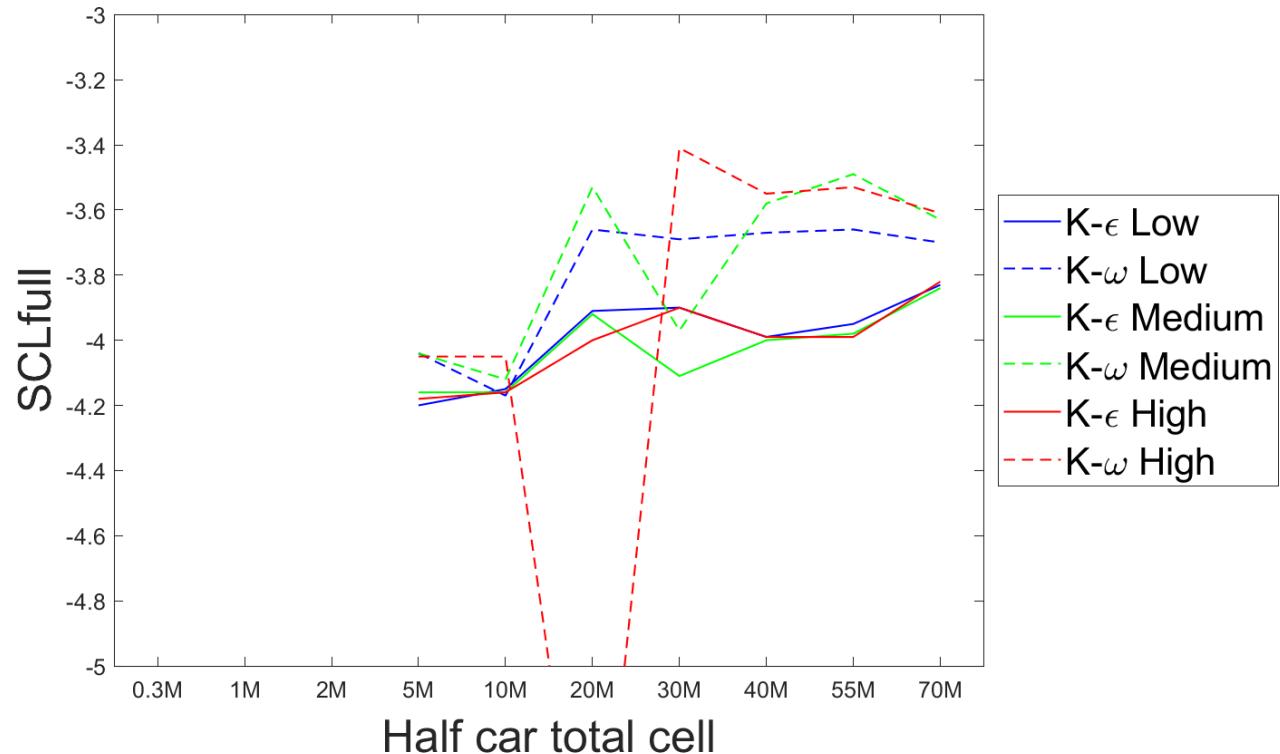
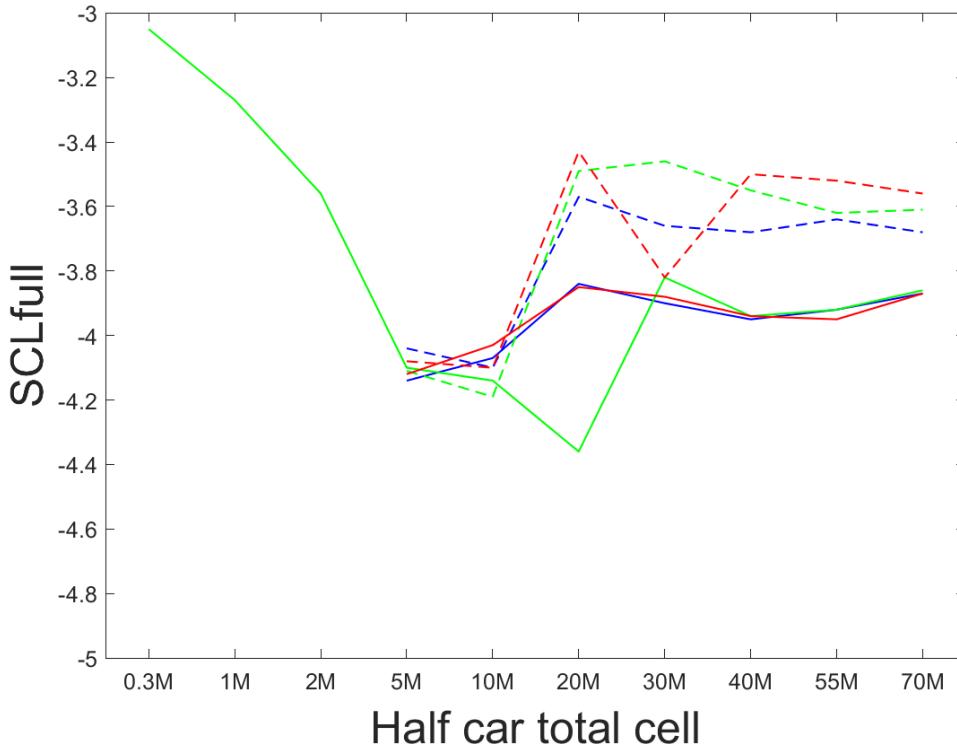
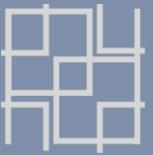
36h-100h程度はシミュレーションの時間外に必要（コストすご）

数値的結果の傾向：SCLとSCD



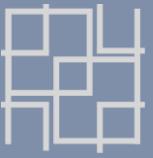
- 40-70Mで収束
- サイドウィングの迎え角による相関も40Mも境目に変化している
- 乱流モデルによる差は特にSCLで顕著 (K-Epsilonが過剰予測)

数値的結果の傾向：SCLと速度 (Re数)



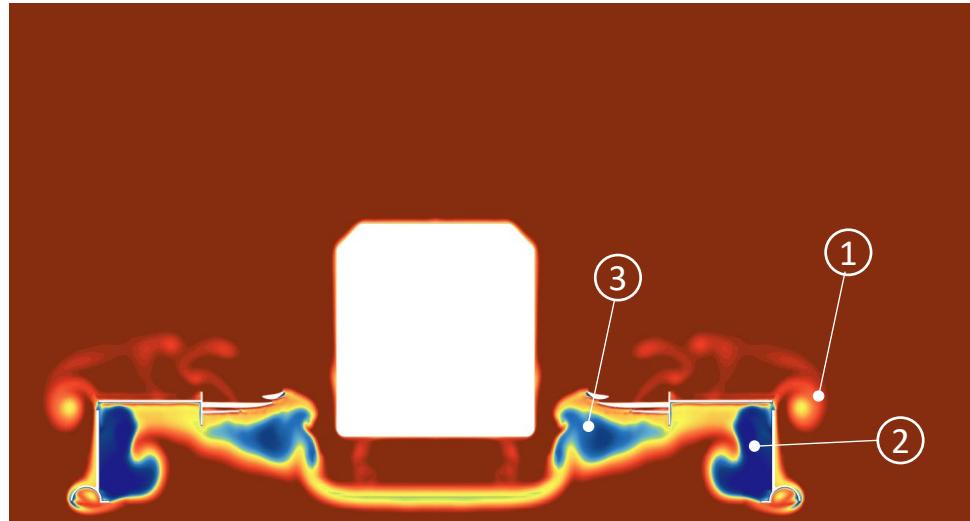
- SCL値が5Mから70Mで0.8も変動している
- サイドウィングの迎え角による相関も30Mも境目に変化している
- 乱流モデルによる差はメッシュ数の増加に伴い小さくなる

比較する上で特徴的な流れ場：前方



FWの流れ場

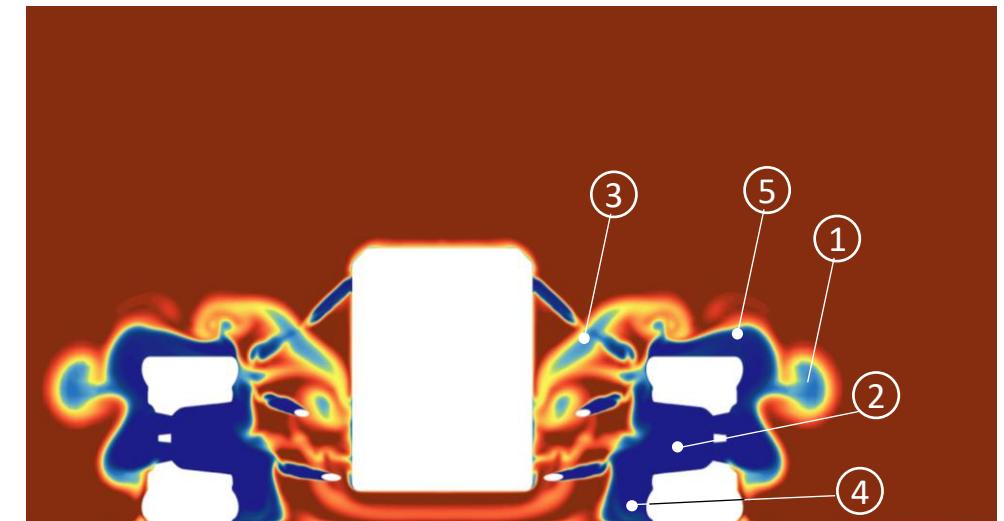
- ① Outwashによる渦
- ② 渦のバースト
- ③ 流れの剥離



CpT plot Z

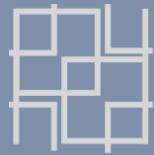
フロントタイヤの流れ場

- ①② 渦の合体
- ③ 流れの剥離
- ④ 流れの押し潰し
- ⑤ タイヤからの剥離



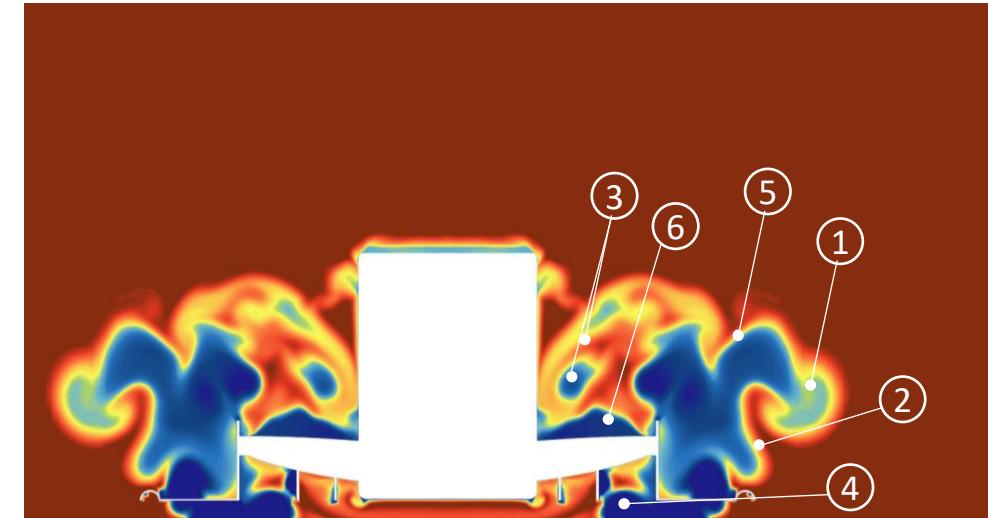
CpT plot Z

比較する上で特徴的な流れ場：SW前方



SW前方の流れ場

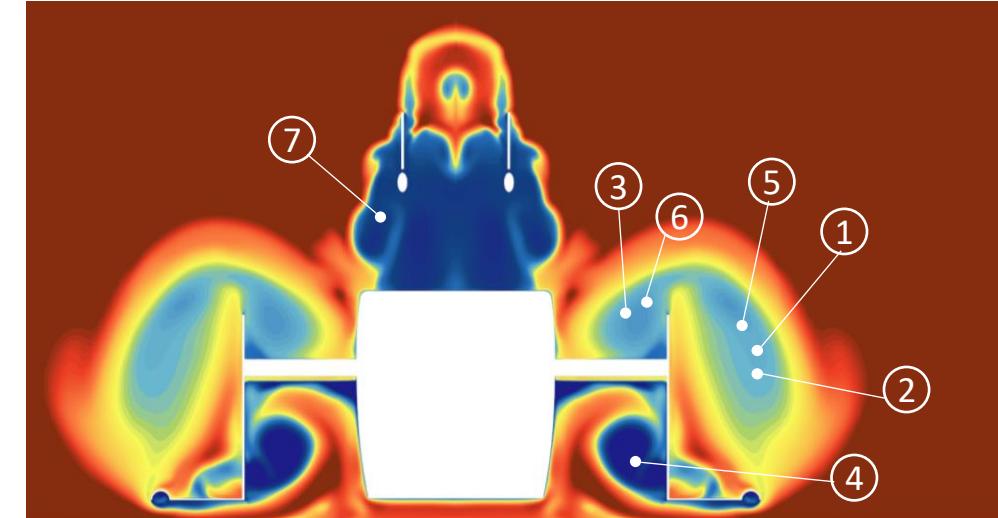
- ①②⑤ outwashの渦などが合体
- ③ 流れの剥離（分裂した）
- ④ 流れの押し潰し
- ⑥ SW上面の剥離



CpT plot Z

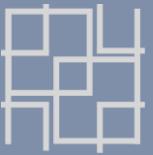
SW後方の流れ場

- ①②③⑤⑥ outwashの渦などが合体
- ④ 流れの押し潰し
- ⑦ ドライバーの剥離



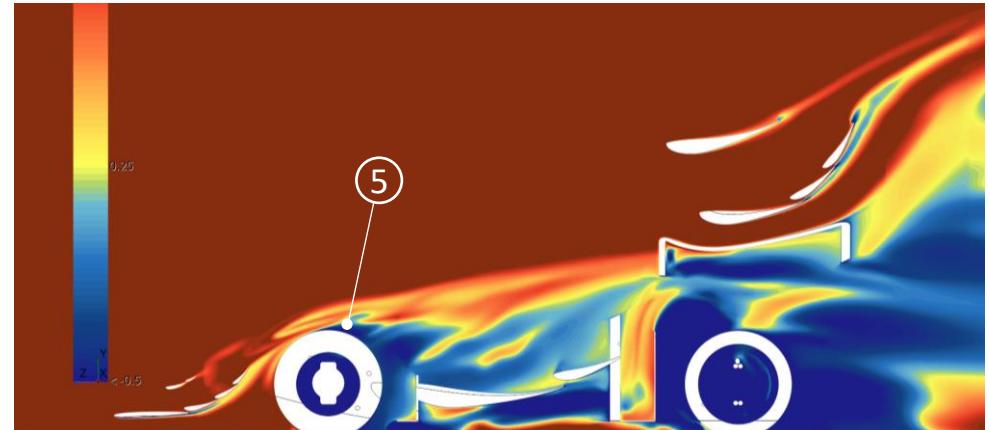
CpT plot Z

比較する上で特徴的な流れ場：RW

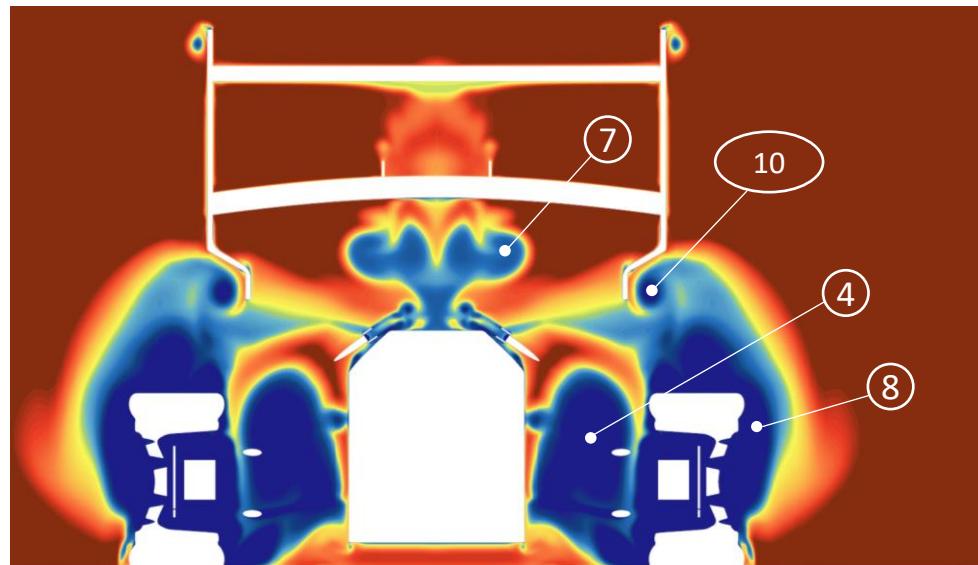


RWの流れ場

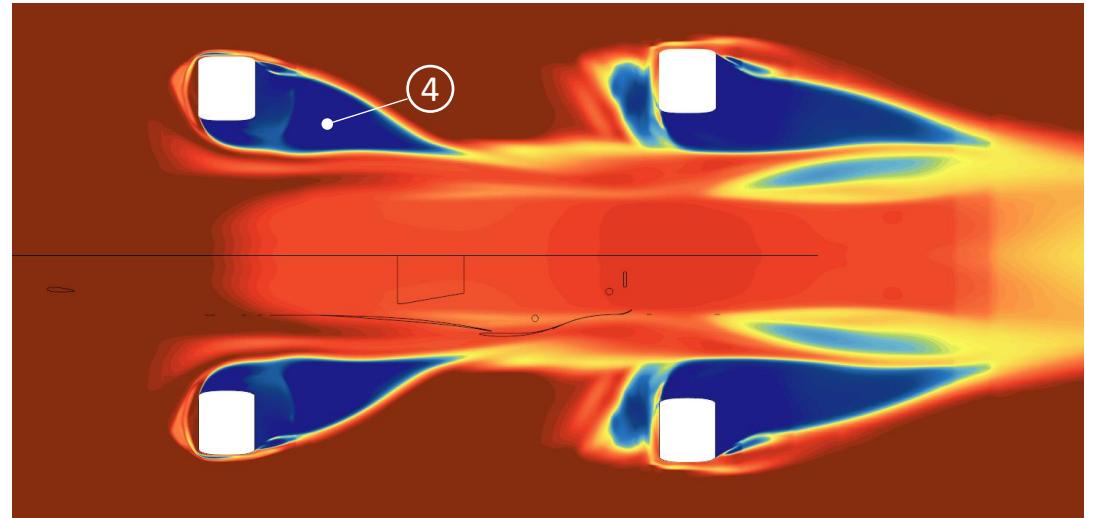
- ⑧ outwashの渦とタイヤの後流が合体
- ④ 流れの押し潰し
- ⑦ ドライバーの剥離
- ⑩ 翼端板による渦（これが賢い）



CpT plot X

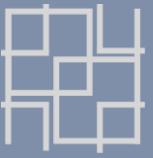


CpT plot Z

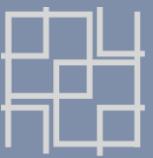


CpT plot Y

比較する上で特徴的な流れ場：全体を通して



メッシュの変化による流れ場の比較



赤：比較対象のエネルギーが大きい

青：比較対象のエネルギーが小さい（剥離の予測ミス）

緑：プログラムがスカラー値を検出できなかった or Colour map外

5M vs 70M

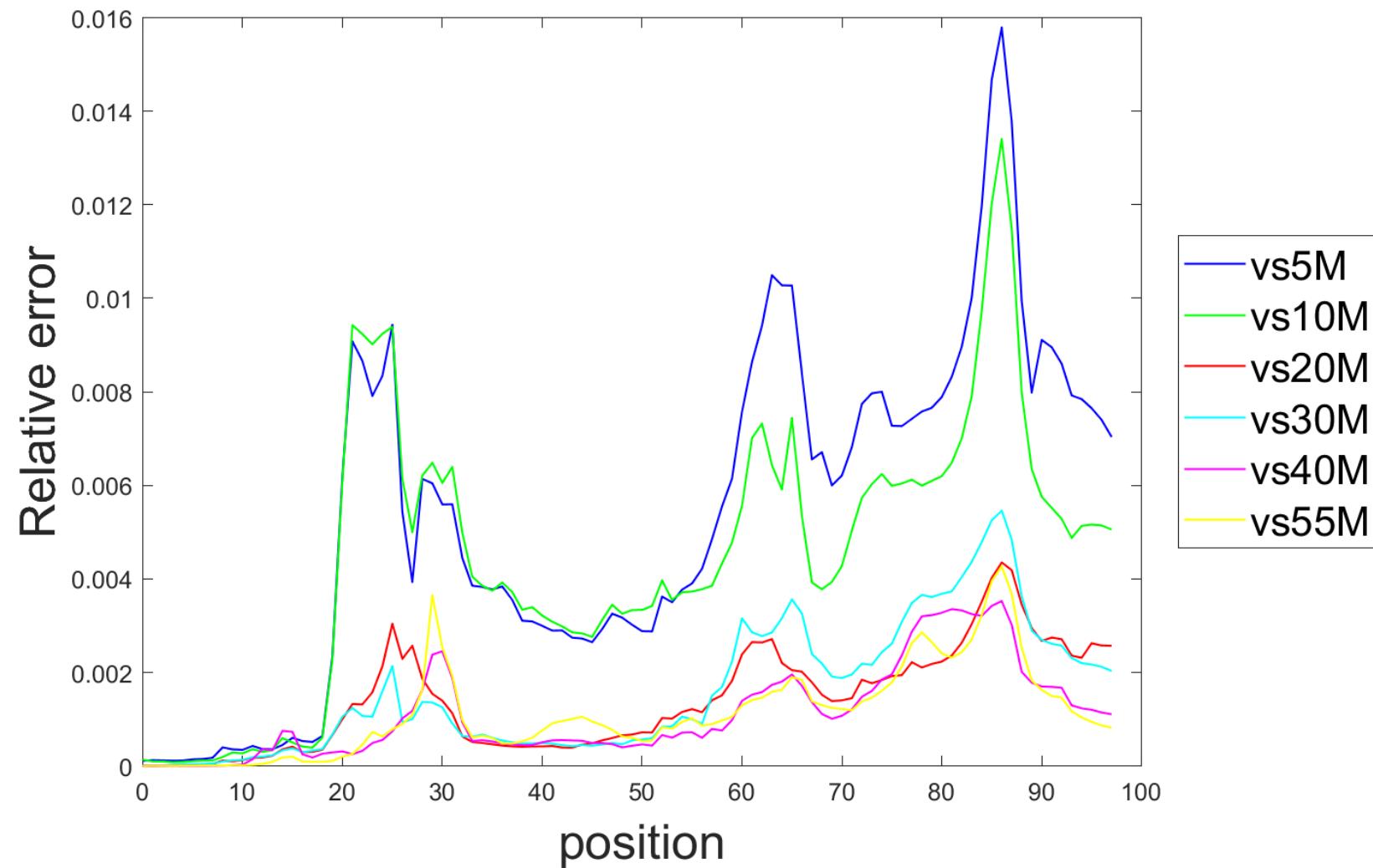
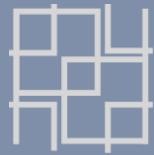
10M vs 70M

20M vs 70M

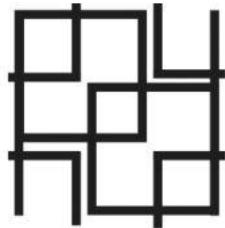
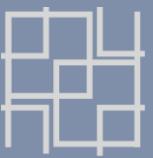
30M vs 70M

CpT Delta Plot

メッシュの変化による流れ場の定量的比較



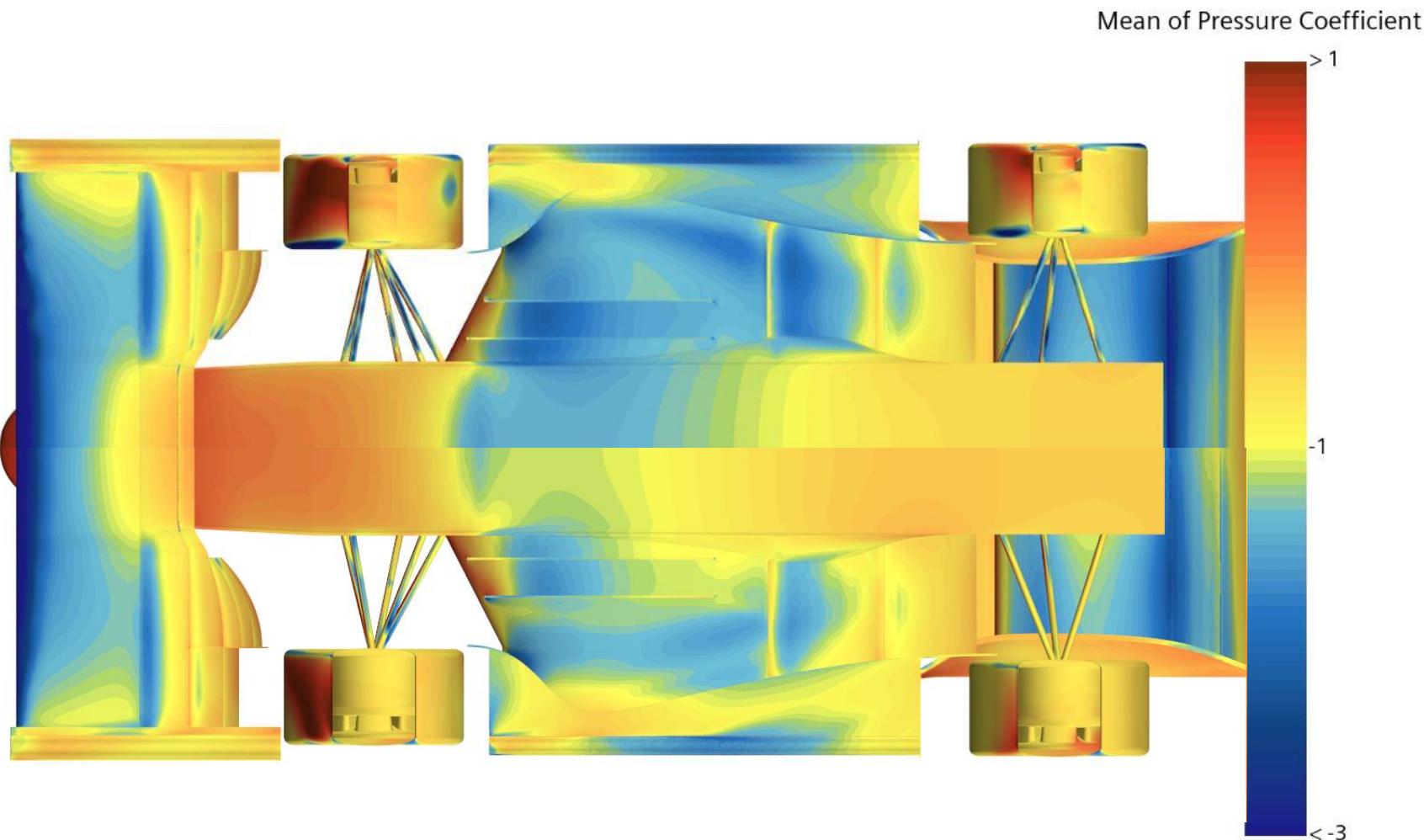
メッシュ数による圧力分布の違い



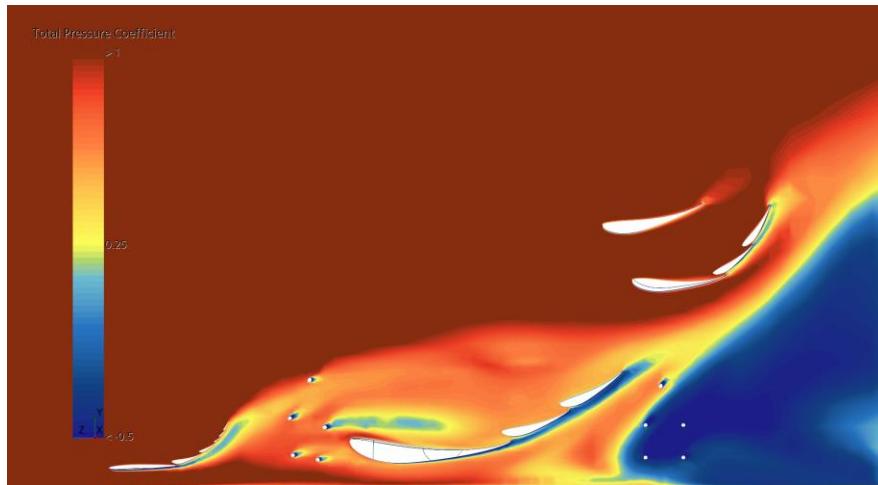
5M K-Omega



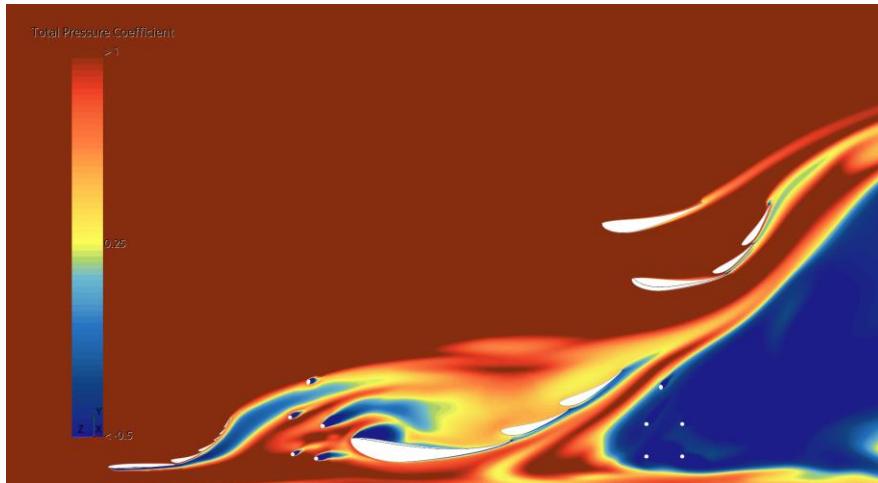
70M K-Omega



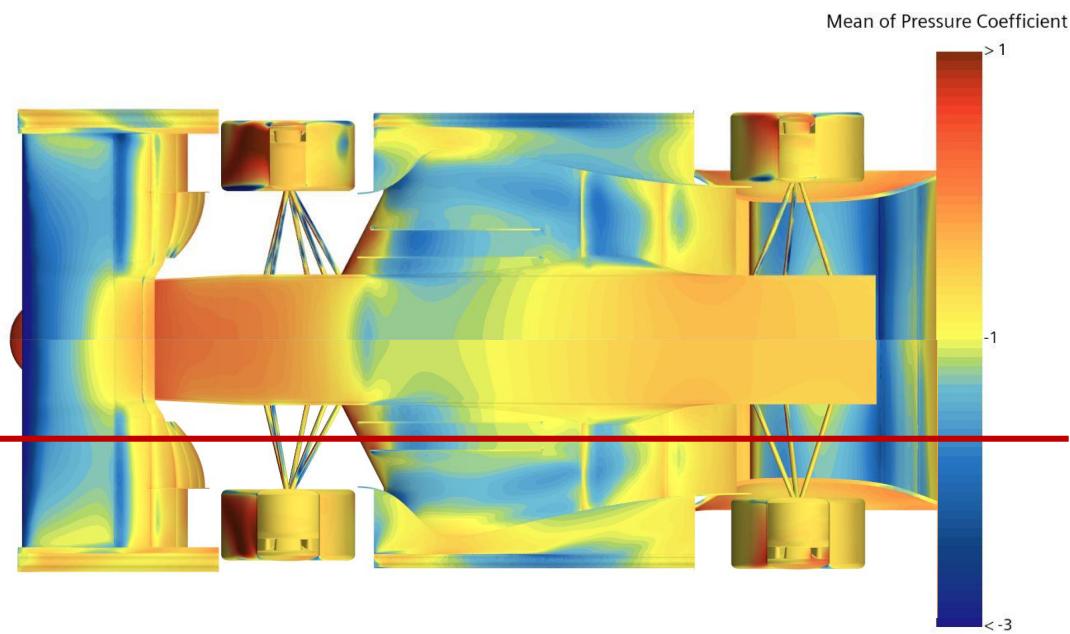
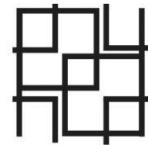
メッシュ数による結果の違い



CpT Plot: 5M K-Omega SST

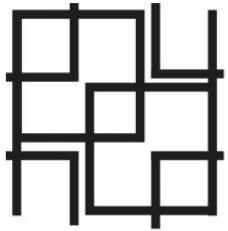


CpT Plot: 70M K-Omega SST



Cp Plot
上：5M K-Epsilon
下：70M K-Omega

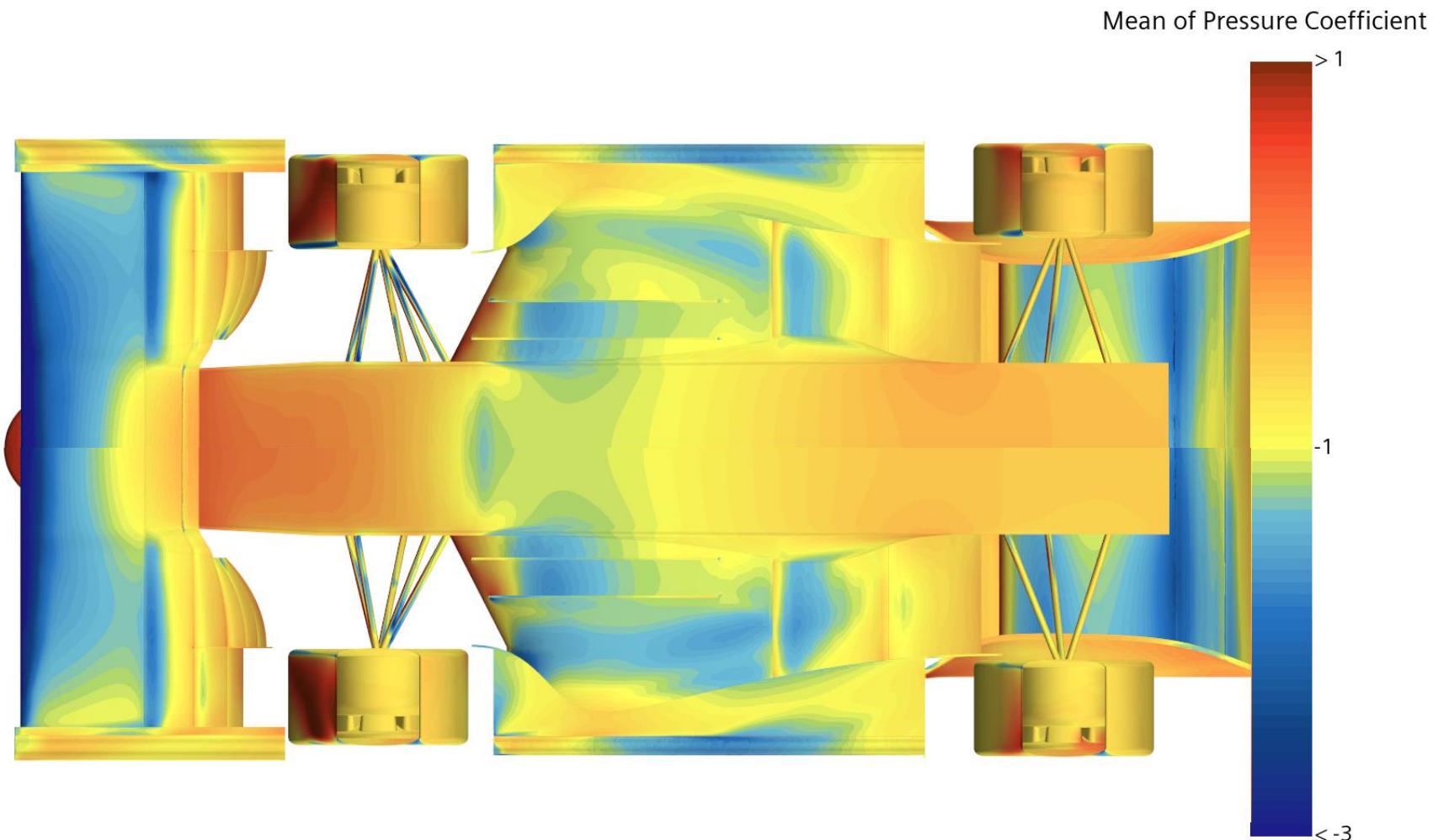
乱流モデルによる結果の違い



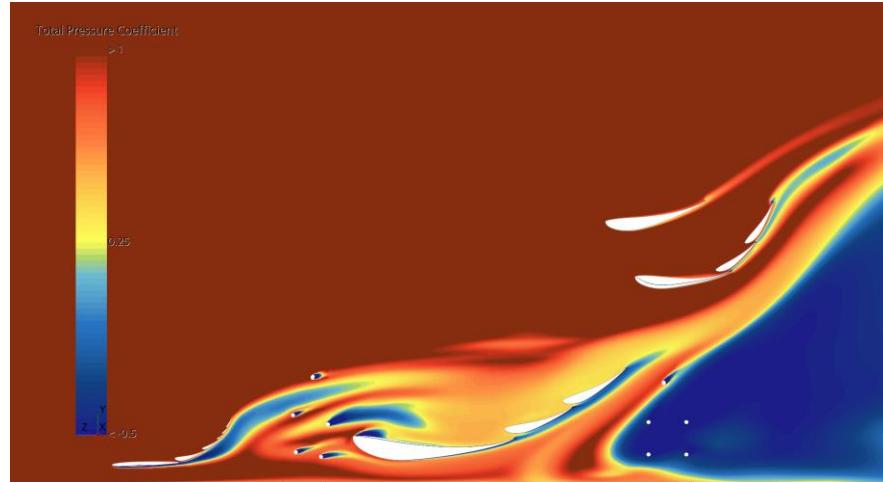
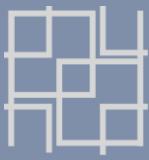
70M K-Epsilon



70M K-Omega



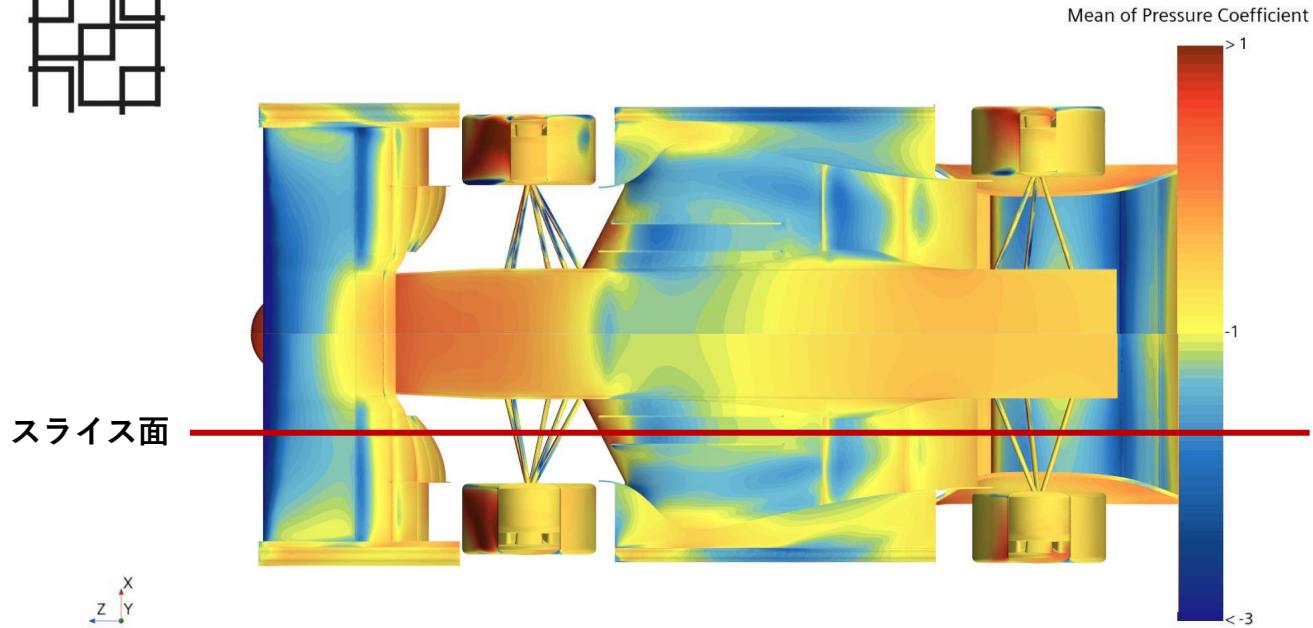
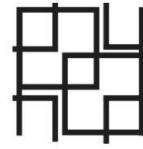
乱流モデルによる結果の違い



CpT Plot: 70M K-Epsilon

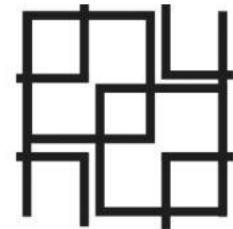
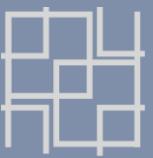


CpT Plot: 70M K-Omega SST



Cp Plot
上：5M K-Epsilon
下：70M K-Omega

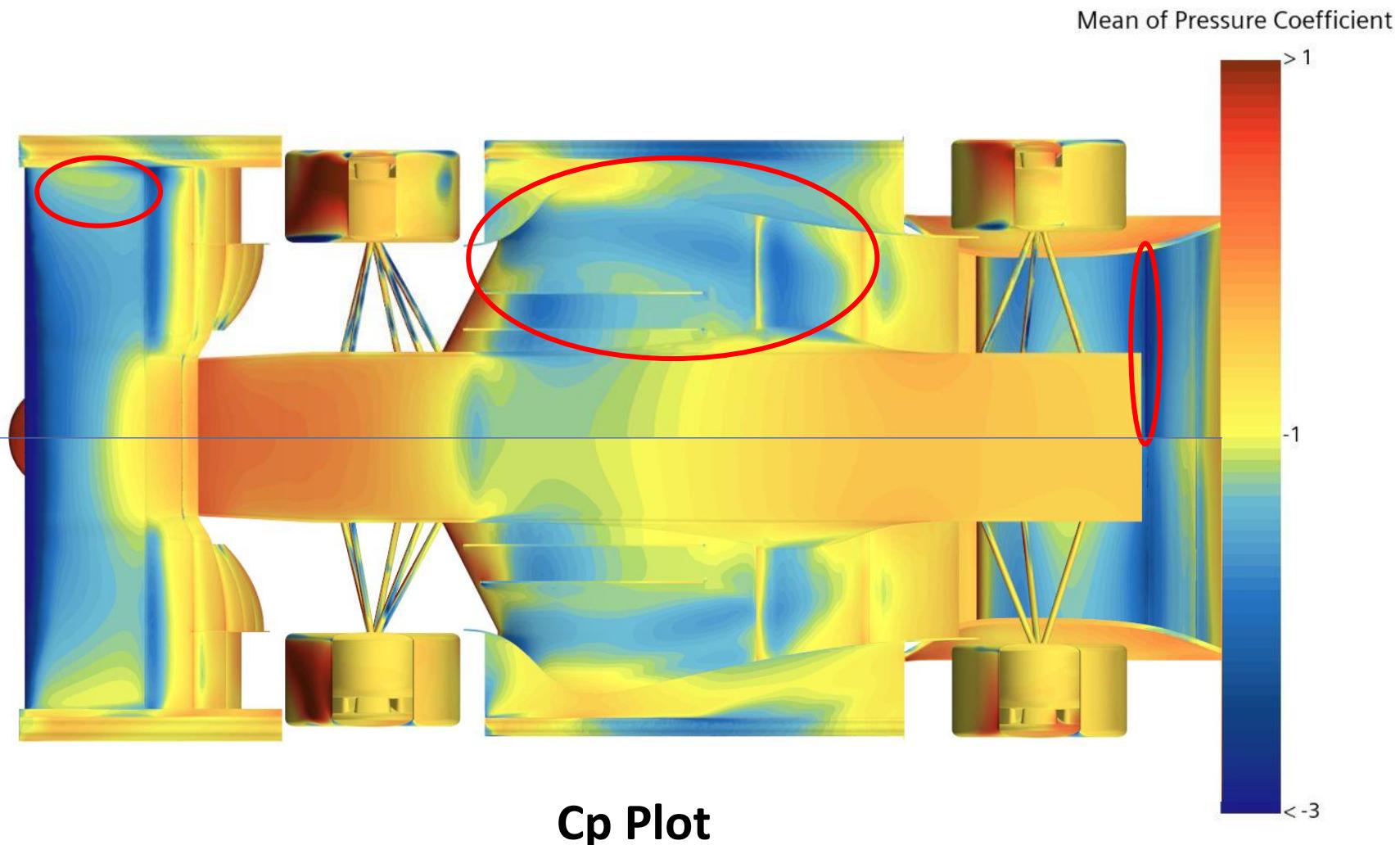
メッシュ数と乱流モデルによる結果の違い



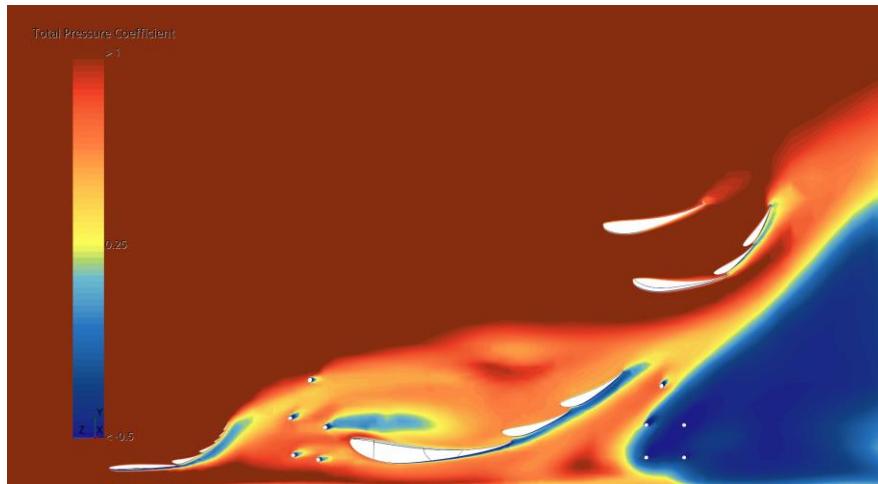
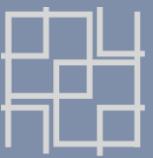
5M K-Epsilon



70M K-Omega



メッシュ数と乱流モデルによる結果の違い



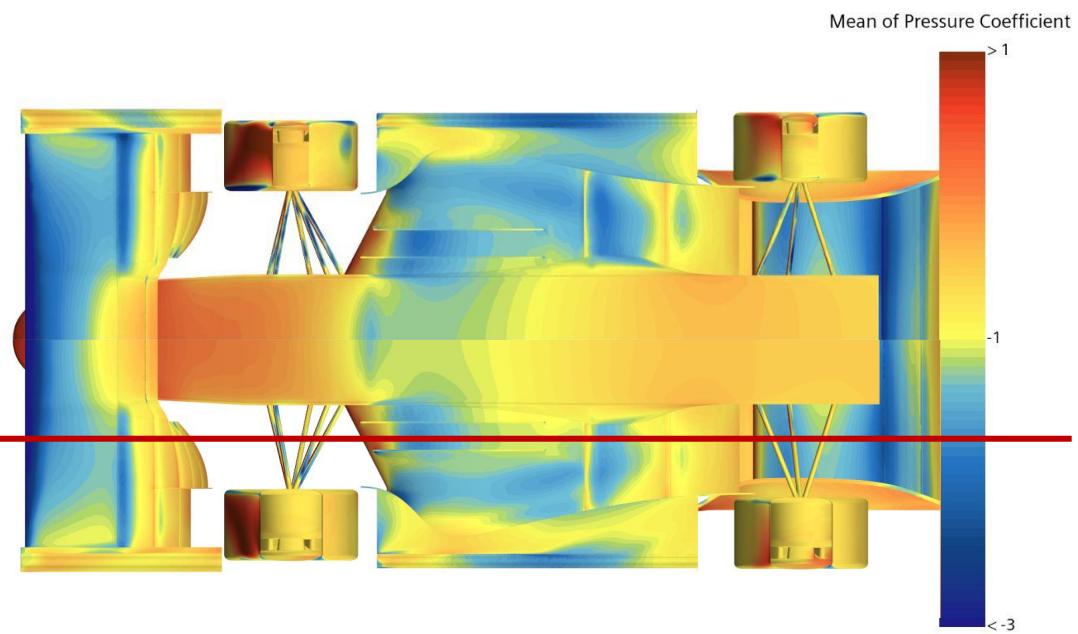
CpT Plot: 5M K-Epsilon



CpT Plot: 70M K-Omega SST

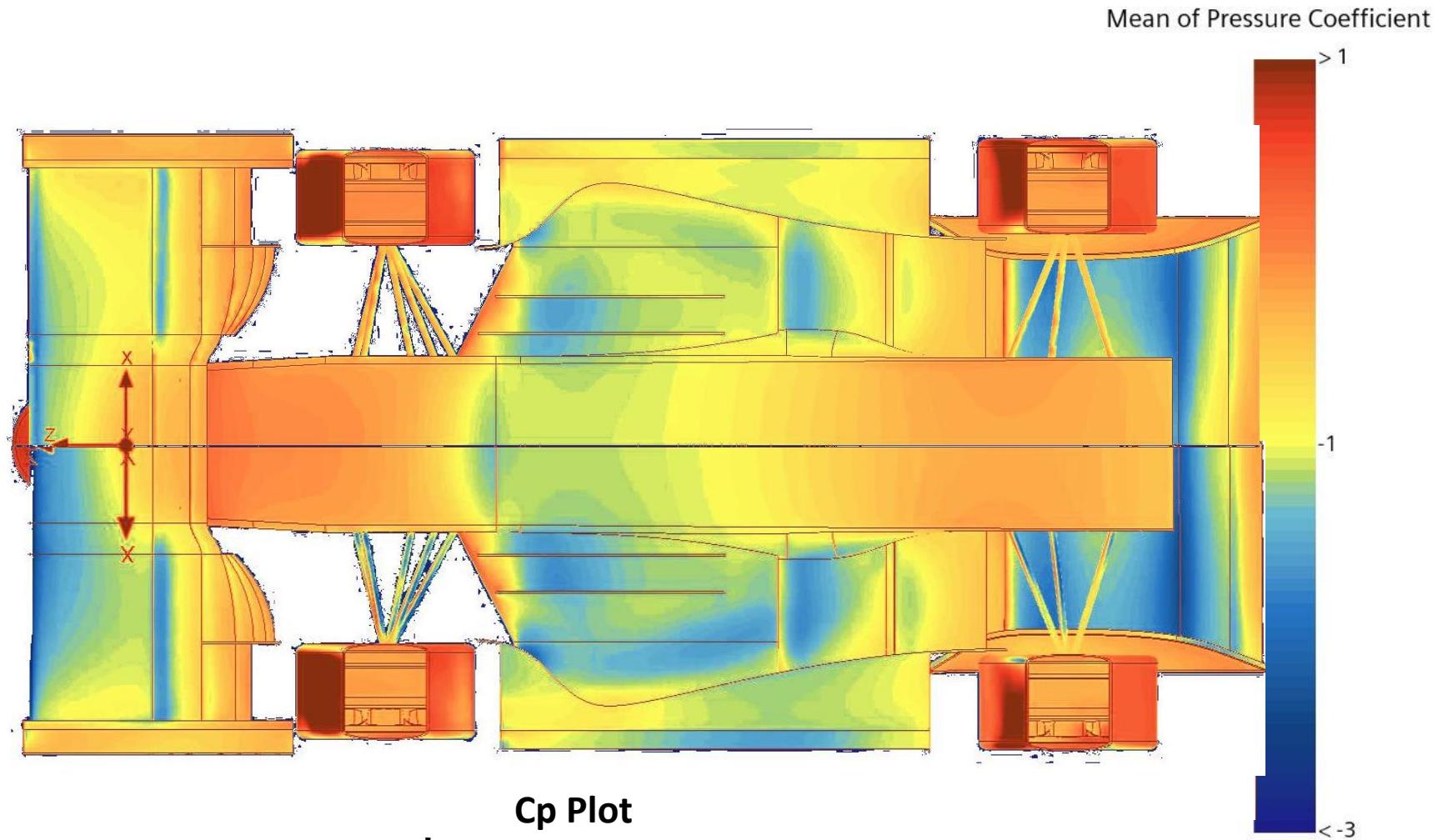
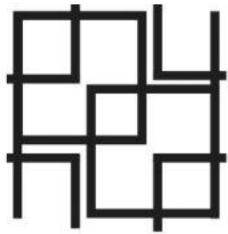


スライス面



Cp Plot
上：5M K-Epsilon
下：70M K-Omega

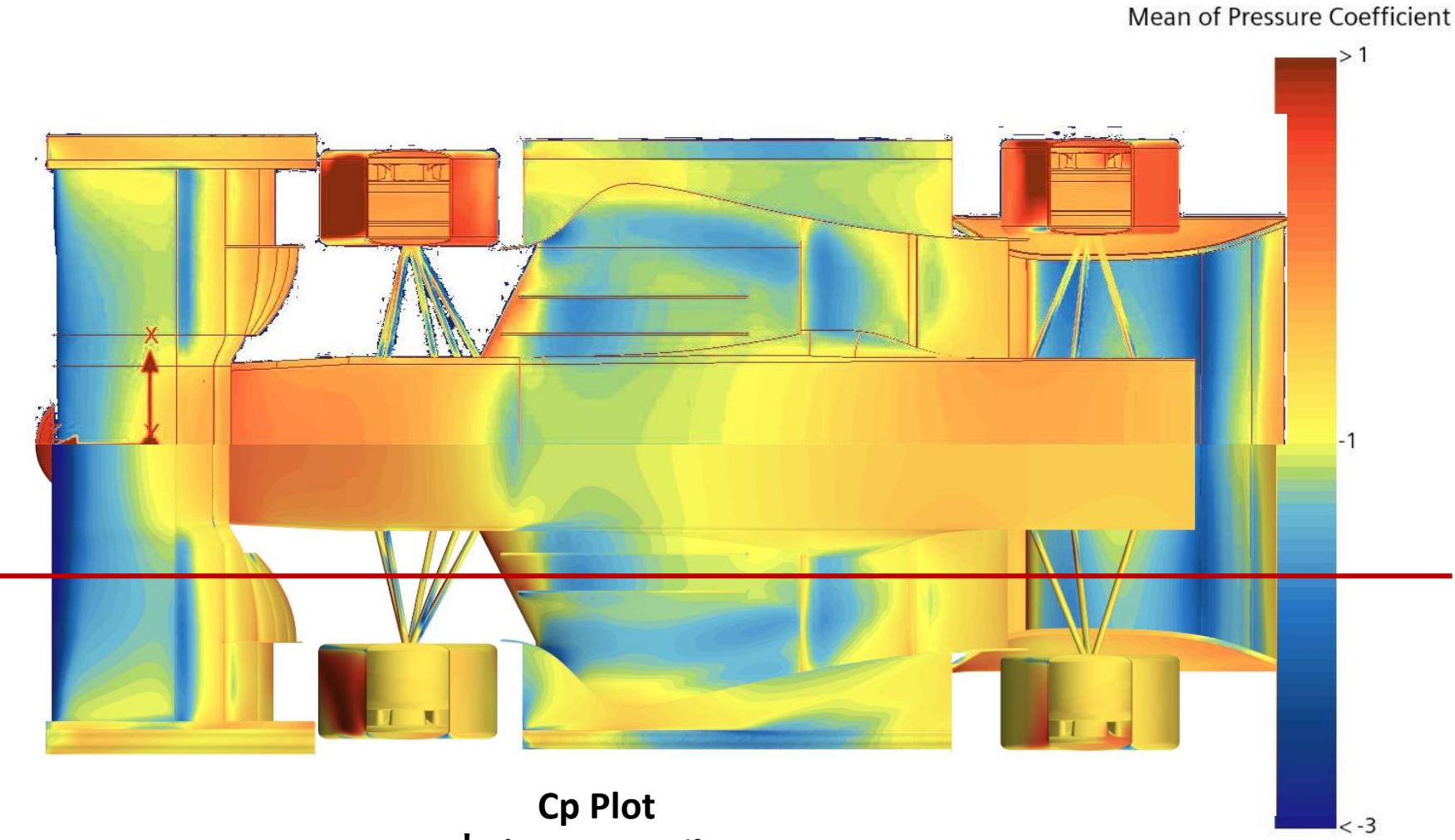
SolidWorksの比較



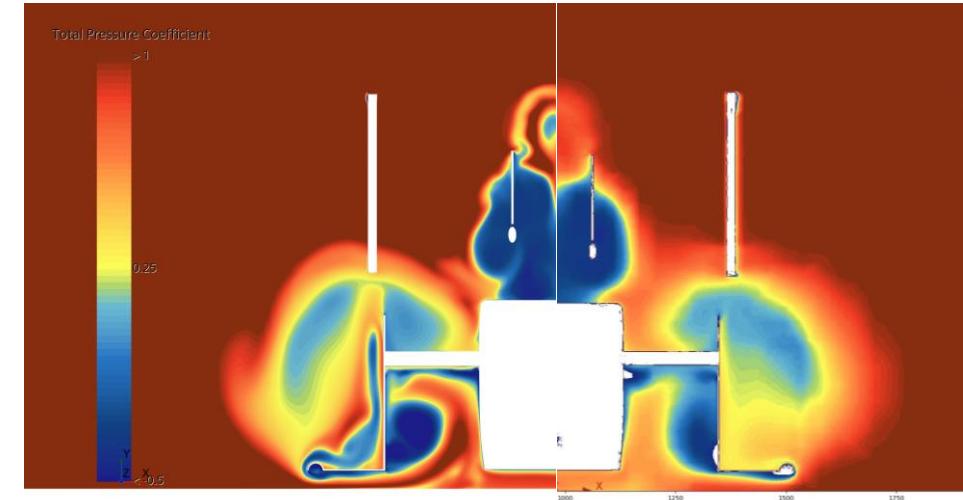
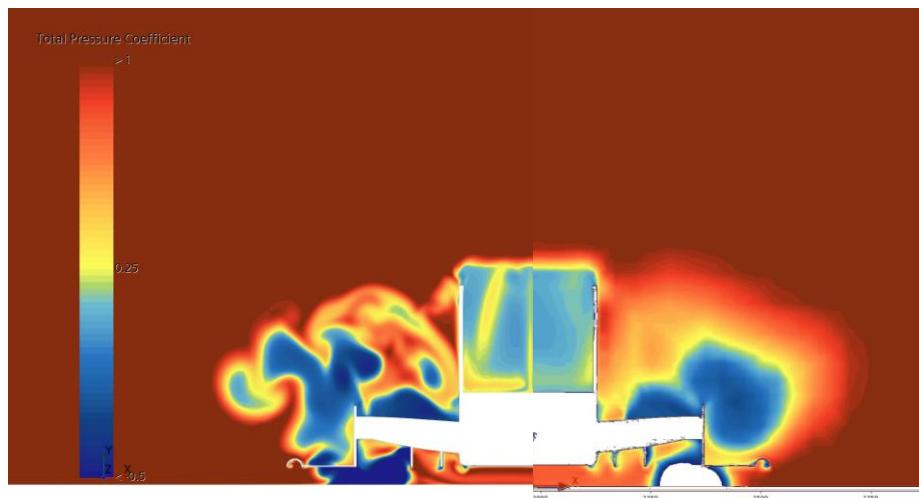
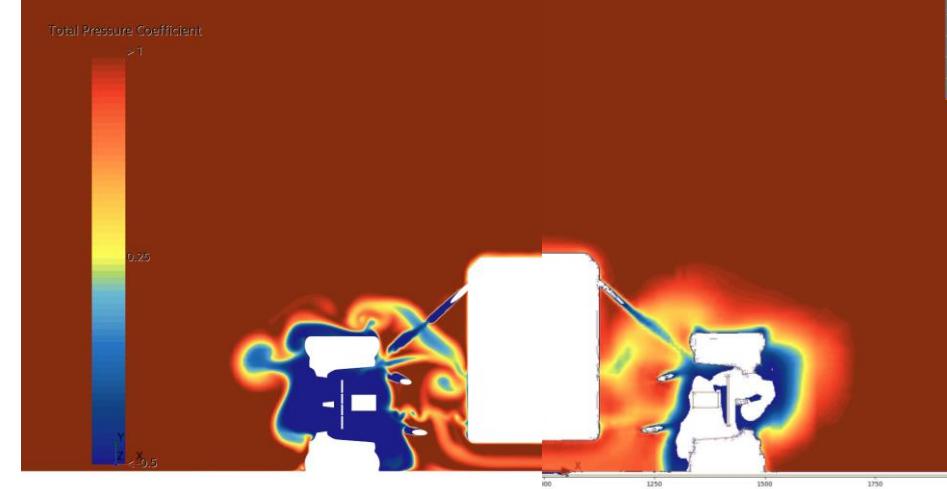
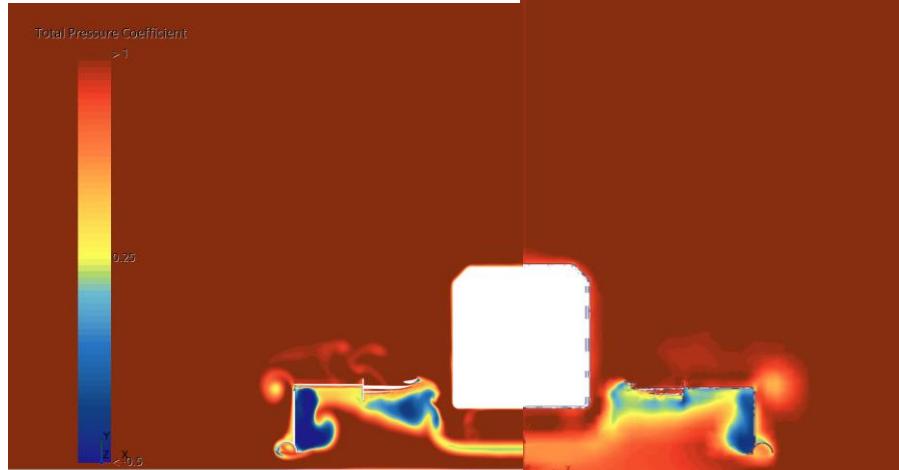
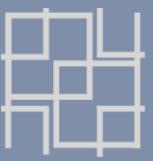
SolidWorks vs STARCCM+



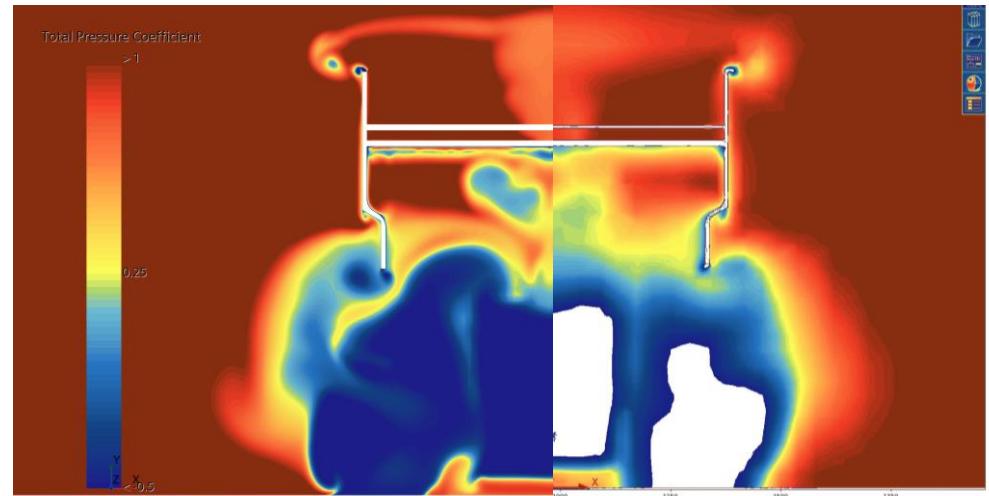
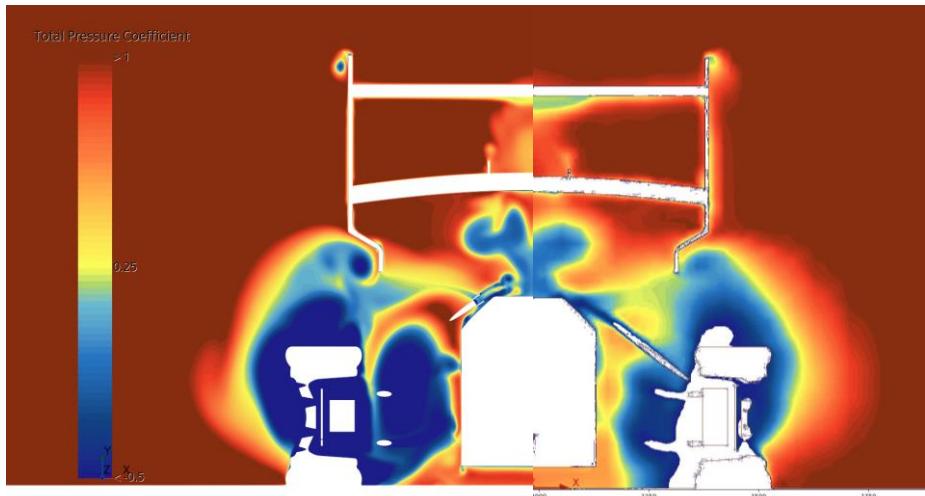
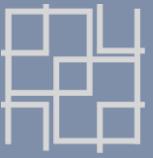
スライス面

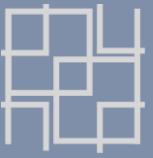


SolidWorks vs STARCCM+



SolidWorks vs STARCCM+



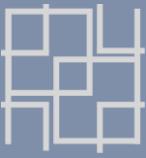


ここまでで質問あればぜひ！！

(答えられるかわかんないけど)

目次

1. 背景
2. 設定の範囲
3. 結果
4. あの風洞について
5. 結論



前提

- 実験的アプローチが気軽にできるようになるのは素晴らしい
- 風洞の民主化という彼らの思想にも賛成
- 従来の風洞はかなりコストがかかるので、低コスト化できる新技術開発は大事

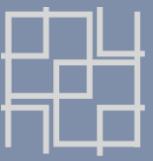
レースカーエアロという観点

- Inletが小さいし、ファンをつなぎ合わせてるという観点からせん断乱流の影響が気になる
- タイヤが回ってない
- 地面も動かない

なぜ風洞実験をするか

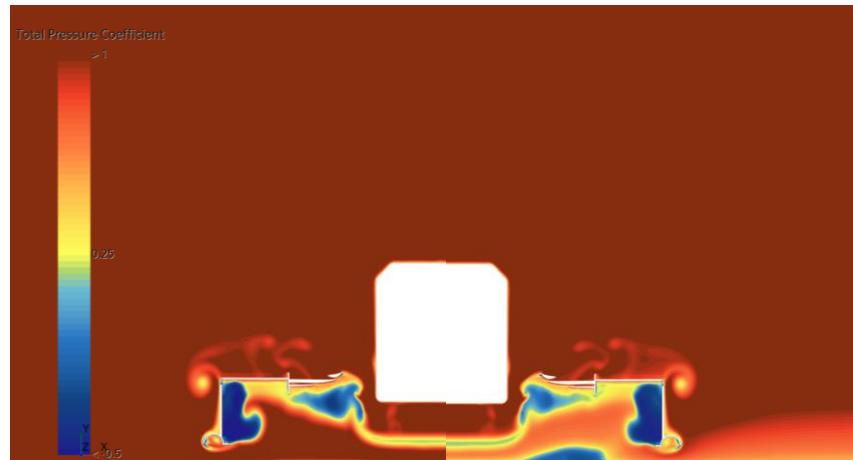
- CFDでわからない部分の詳細を確認したいから
- 翼端渦見て満足するわけにはいかない
- ではわからないのは主にどこ？

タイヤと地面が止まるということ



FW周り

- 境界層厚みによる全圧の低下
- Upwashによる地面の境界層剥離
- 翼端の渦は特に影響なし



回転あり vs 回転なし

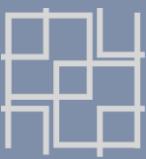
フロントタイヤ周り

- 境界層剥離による新たな渦の発生
- 車体下部の全圧低下
- そのほかは大きな影響なし



回転あり vs 回転なし

タイヤと地面が止まるということ

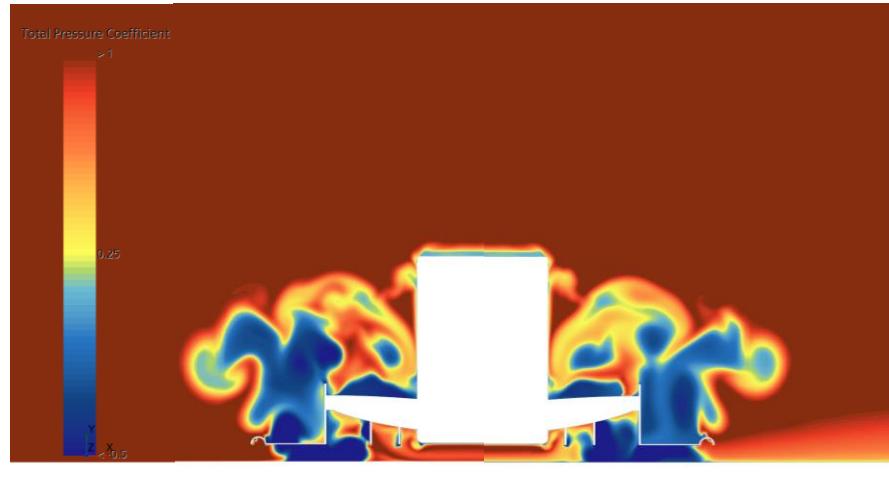


SW前方

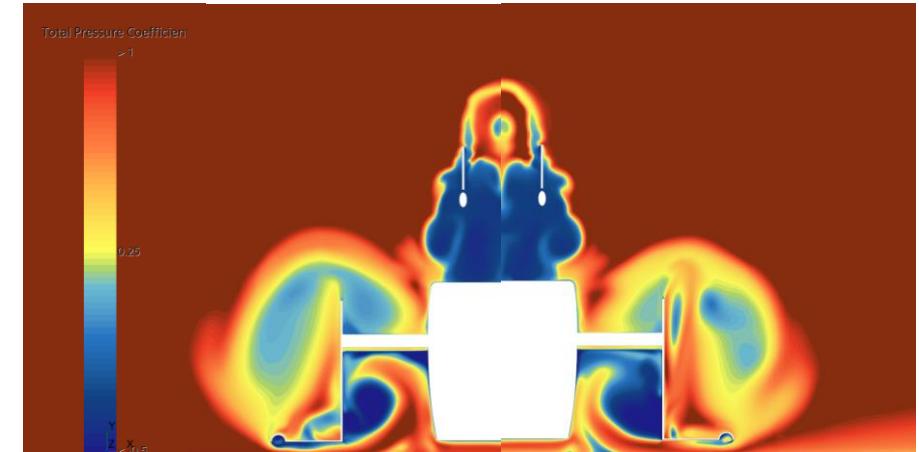
- 外側の領域の全圧の区域が異なる
- フロントタイヤの後流の様子が異なる
- 上面の剥離はともに捉えられている

SW後方

- 翼下部の流れ場が大きく異なる
- SW上面、外側の流れにも影響あり
- ドライバーの後流は正しく予測できてる



回転あり vs 回転なし



回転あり vs 回転なし

タイヤと地面が止まるということ

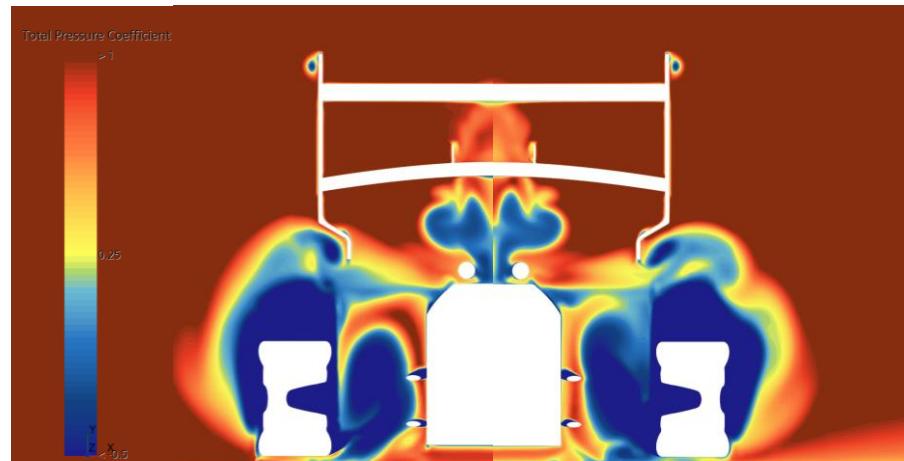


RW前方

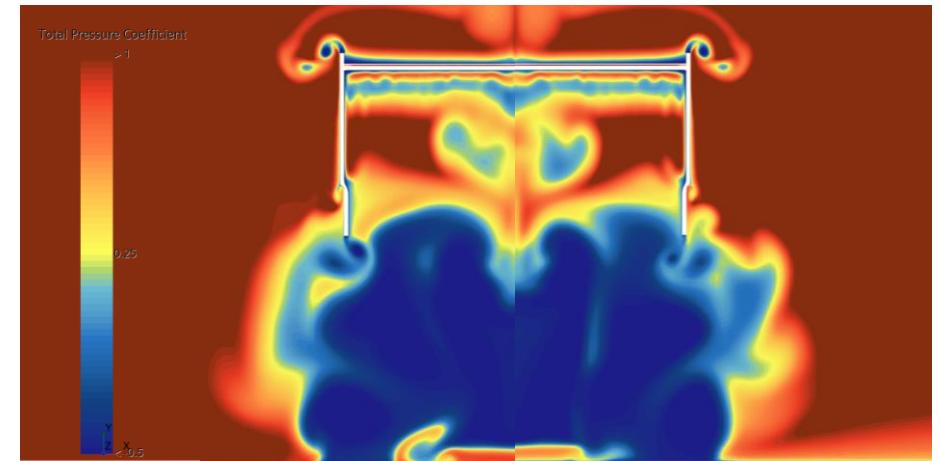
- 翼端板の動きは正しく反映されている
- リアタイヤは前方に板があるので変化なし
- SWからの流れが違う

RW後方

- RW単体の流れ場は大きく変化なし
- SW由来の流れの傾向は最後まで異なる

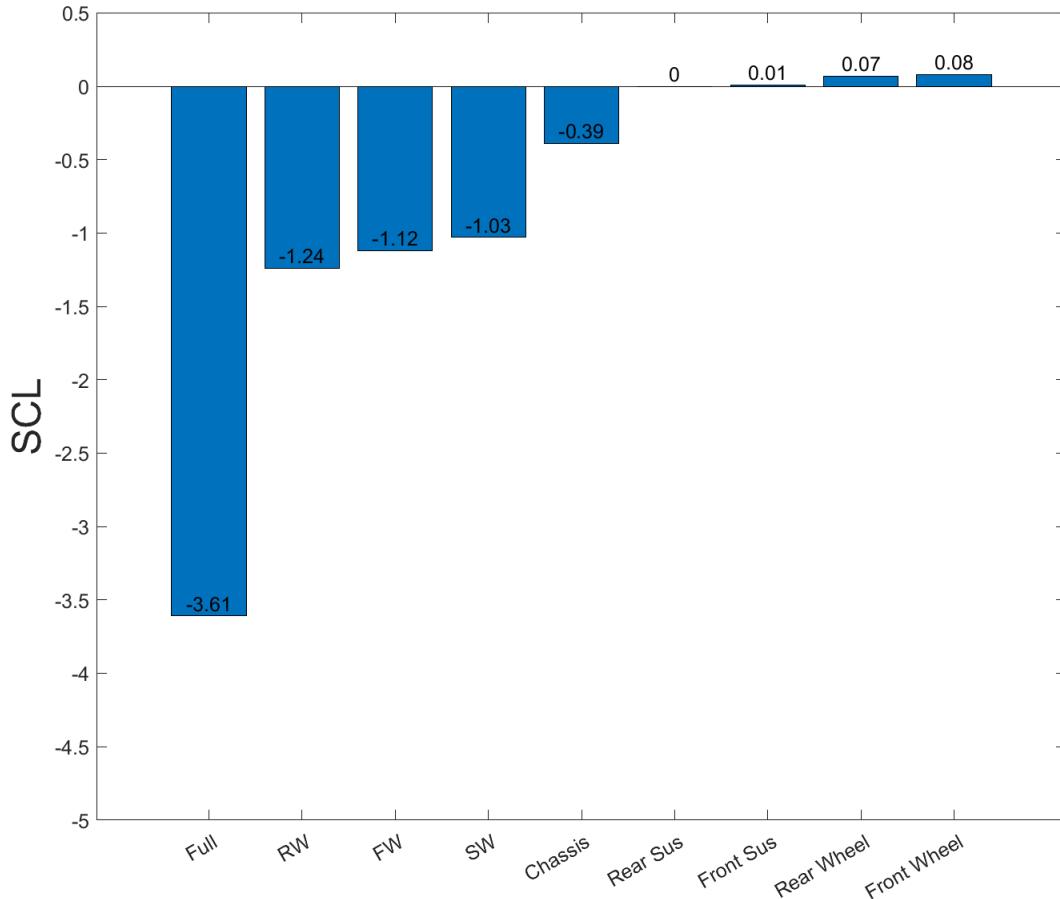


回転あり vs 回転なし

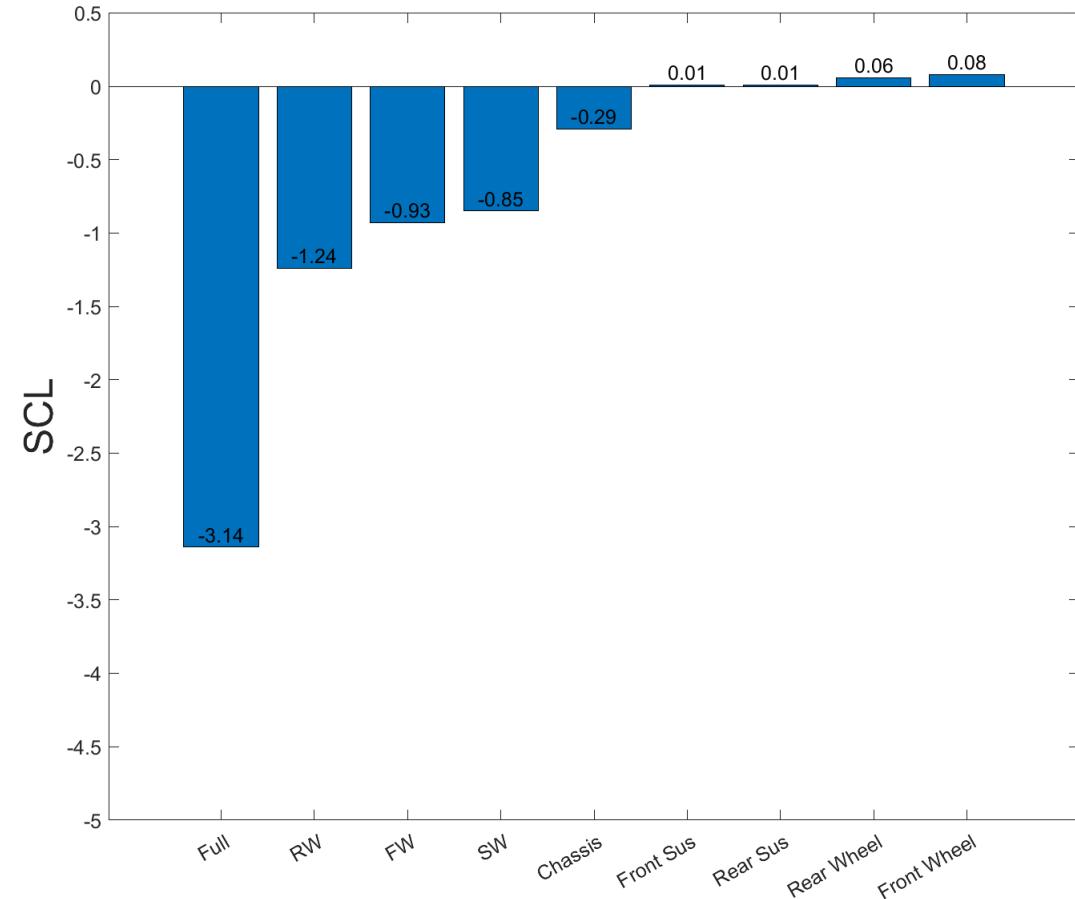


回転あり vs 回転なし

数値的な違い 1

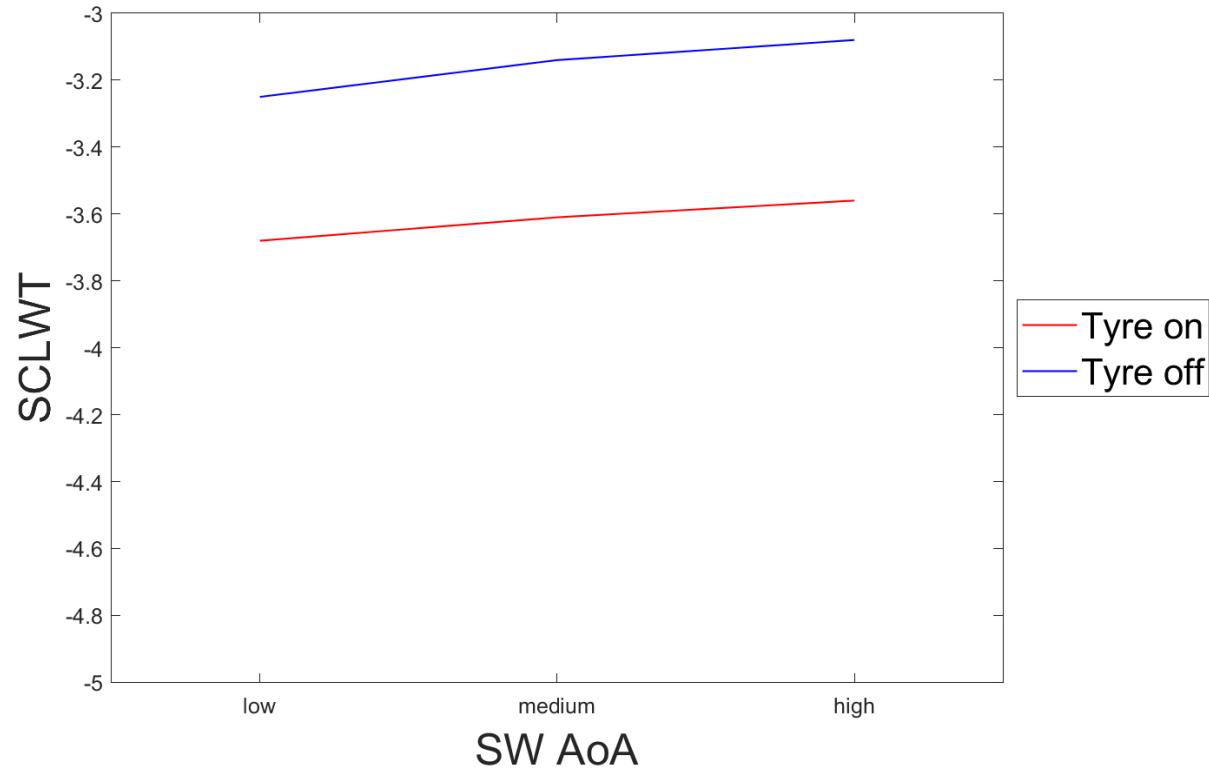
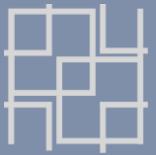


回転あり



回転なし

各部位のダウンフォース量の比較



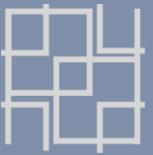
形状の違いによる相関への影響

各部位の相違点

- FWとRWの予想が大きく異なる
- RWの性能は正しく予想できる
- シャシの予想にも影響あり

相関からわかる相違点

- タイヤが回ってるほうが上がり幅が小さい
- 傾向そのものは一致
- 絶対値は15%ほど異なる



タイヤと地面が止まると

- 地面の境界層がUpwashで引っ張られる
- タイヤの後流の挙動が大きく異なる
- タイヤの剥離点が変わる



個人的には、かなり渋い

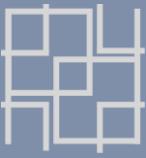
一般には

Low quality data is worse than nothing

ただ、駄々をこねるはエンジニアの仕事内容ではない

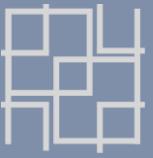
この場合

- 使用できる施設は既知
- 施設の特性は適正使用範囲を評価、理解し実験計画を立てることでデータの質は大きく変わる
- さて、自分ならどうしようかな



ではどのように機会を活用するか（自分なら）

- まずは全く同じ条件でのシミュレーションを回しておく（乱流強度を事前に知る必要がある）
- CpTプロットと壁面せん断応力のLICプロットをA3ぐらいで50枚ぐらい印刷して持参する
- 安全面の観点からドライバーは乗れるのか？乗れないならマネキンみたいなの作っておくべき
- 車体にタフトを貼っていき、LICプロットと比較する（正面から写真とって画像処理して重ねるかな）
- あえてRWが剥離するセットアップをシミュレーションで用意して、タフトを使って剥離点の比較をする
- 気になる渦の位置があって、それが速度プローブなどで確認できるなら素晴らしい
- 圧力センサにアクセスできるチームは翼下面の圧力分布が測定できると嬉しい（CMWSさんとか）
- サスペンションのストロークセンサなどから各部位のDW推定



ここまでで質問、ご意見あればぜひ！！

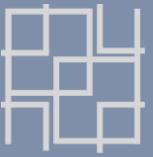
(答えられるかわかんないけど)

目次

1. 背景
2. 設定の範囲
3. 結果
4. あの風洞について
5. 結論



- 異なるメッシュ数における流れ場予測の評価ができるtest caseができた
- SolidWorks Flow simでも最低限の性能予測は可能だが剥離を伴う流れ場の評価には注意が必要
- 少ないメッシュ数では全圧を大きく見積もる傾向にある
- 10m/sより15m/sのほうが相間に安定性が見られたのでこの速度域でもRe数による影響がある
- 同じメッシュ数でも、ドメインサイズは大きくするほうが真値に近い値が予想できる
- 乱流モデルを変更することで相関関係も変わってくる
- 車半分のモデルで70M程度のメッシュを切っても結果は安定しない
- タイヤと地面の条件が違うとFWとSW（サイドポッド）に大きな差異が生じる

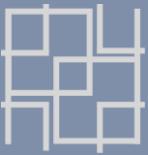


データの使い方

- 僕のwebsiteからダウンロードできる
- 70M、mediumのCADデータ、およびCFDのCpTプロット、Cpプロット、数値データを公開
- CADデータを使いCFDを自身の環境で回し、結果を比較すれば今のCFDの精度がわかるはず
- シミュレーションの制約などについて考える機会となれば幸いです

CFD関係のわからないことについて

- SolidWorks Flow Simに関しては今回で自分はある程度できることが理解できた
- もし問題があったら、ある程度は役に立つことができるかも
- とりあえず、公式ドキュメントとか公開されているWhite paperなどには一度目を通しておく

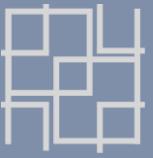


Contribution

- STARCCM+ (80%)、SolidWorks Flow Sim(25%)、データ処理 (100%)、考察(60%)：僕
- STARCCM+ (20%)、SolidWorks Flow Sim(75%)、考察 (40%)：さとう君

Thanks to

- 睡眠時間返上で頑張ってくれたさとう君
- Laptopを貸してくれてる+ GPUの試用に協力してくれた板橋君
- DMでCFDの設定を教えていただいた方々
- 普段からHPCをメンテナンスしてくれているIridis5のスタッフたち



長々とありがとうございました