UNLSI・openVSPワークショップ @東京大学・2024年3月14日

森田直人

本日の内容①



- 事前準備の確認
- オープンソースプロジェクトの扱い方
 - わからないことや改善案があったらissueに書き込む
- openVSPの基本操作
 - 空のプロジェクトから翼を作る・パラメータを調整する・胴体を作る
 - VSPAEROで解析してみよう
 - 作成した翼のCFDメッシュを作成する
 - 作成した翼の構造設計をする・FEAメッシュを作成する

本日の内容②



- UNLSIを用いた解析
 - 空力解析を行う
 - 構造解析を行う
 - 空力弾性解析を行う
- ・ 人力飛行機モデル
 - 各種解析
 - 空力微係数・空力補間曲面を作成する
- フライトシミュレーション
 - FEEのインストール
 - 人力飛行機モデルのシミュレーション

事前準備の確認

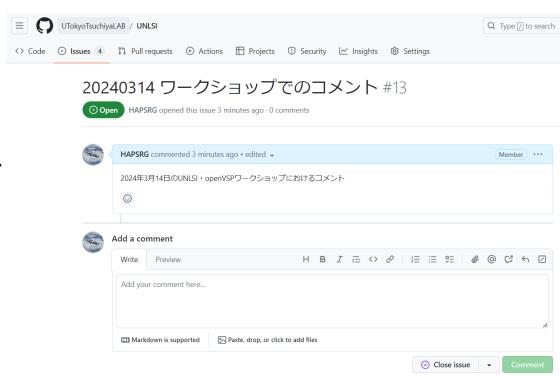


- OpenVSP
 - 3.36(最新版)
- UNLSI
 - V5.03
- MATLAB
 - (できれば)2023a以降。2020以降なら大丈夫だと思う
 - Aerospace ToolboxとDSP toolboxがインストールされている
- FlightGear 2020.3
 - シミュレーションの可視化に使います
 - UNLSImodelの作成と、protocolへのファイル追加
- 1. UNLSIのソースコードをgit cloneもしくはdownload ZIPする
- 2. 適当な場所にexample→UNLSI→wingをコピーして「workshop」にリネームする

オープンソースプロジェクトの扱い方



- オープンソースソフトウェア(OSS)
 - インターネット集合知としてソフトウェアをより良くしていく。
- Issuesを活用する
 - 今回のワークショップでわからないことがあったら、 いつでも右図のIssueに書き込む
 - バグや改善案はどんなものでも良いのでIssueに 追加する。そこから議論が始まっていく
 - Issueが活発なOSS→人が集まる→よりよいソフトウェアへ
- Pull Requestを出す

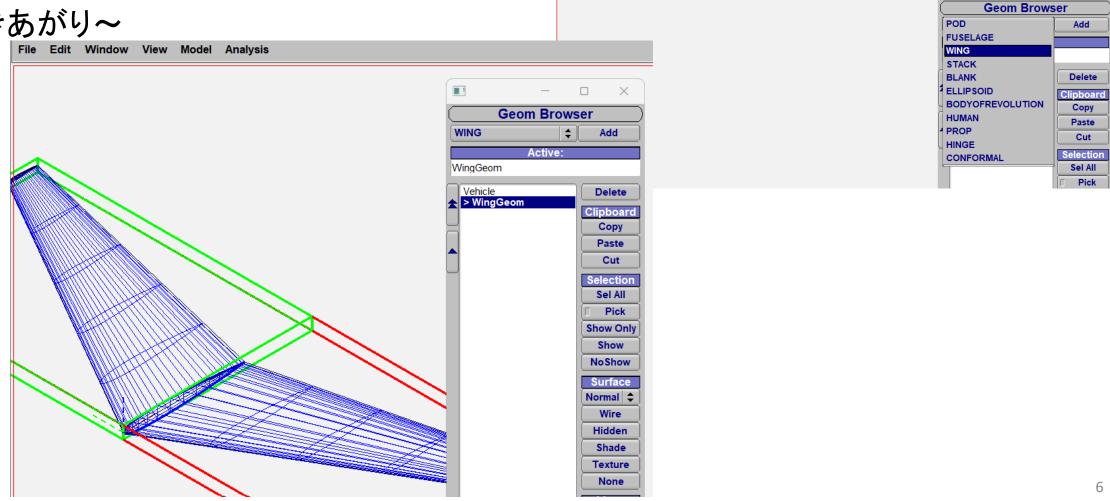


OpenVSPの基本操作・翼を作る



openVSPを起動し、Geom Browserで WINGを追加する

・ できあがり~



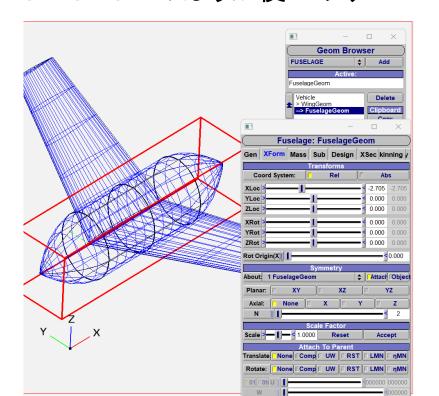
OpenVSP 3.36.0 - Fri 27/10/23

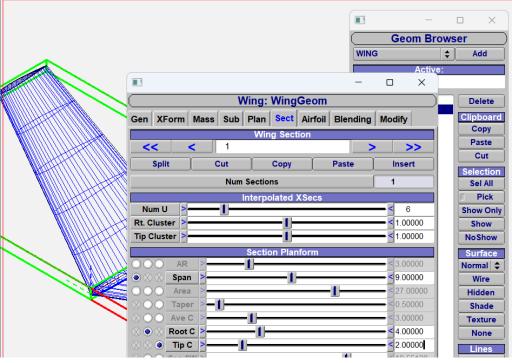
File Edit Window View Model Analysis

OpenVSPの基本操作・パラメータを調整する

HAPS

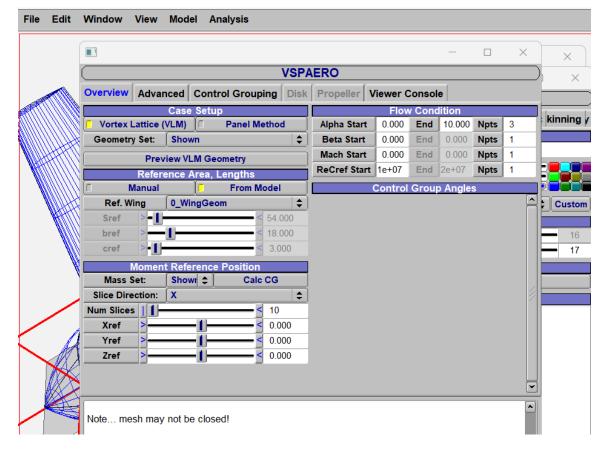
- いろいろパラメータをいじってみよう
- 胴体を追加しよう
 - FUSELAGEを追加
 - 慣れてきたらSTACK GEOMのほうが使いやすい

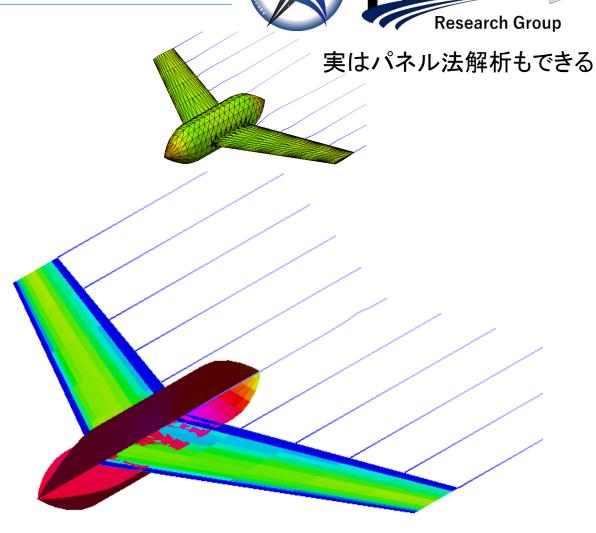




OpenVSPの基本操作・VSPAEROで解析をする

• VSPAERO-VLMでの解析

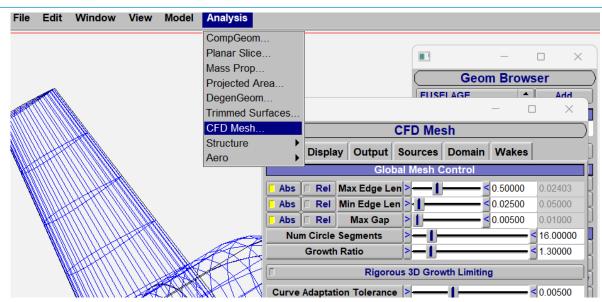




なにかおかしい・・・

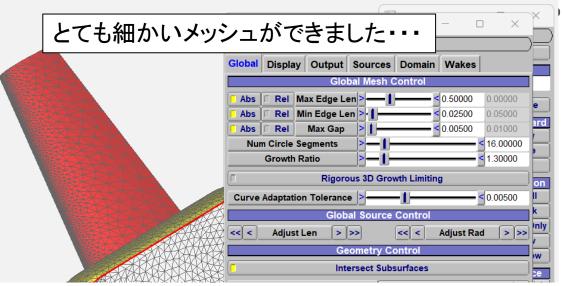
→VSPAEROのVLMはキャンバー面での計算のため、近似形状になる 翼の効果が支配的な機体の揚力の計算はとても得意

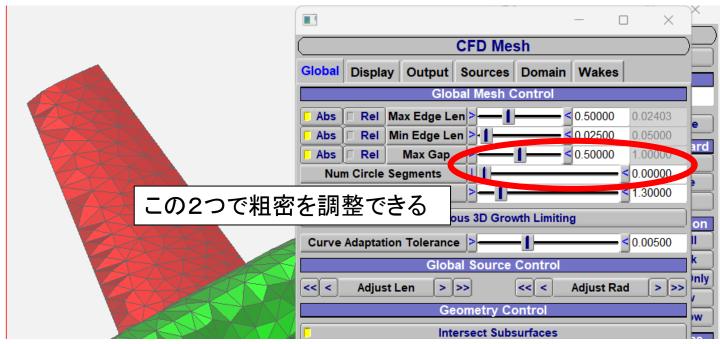
CFDメッシュ(空力メッシュ)の作成



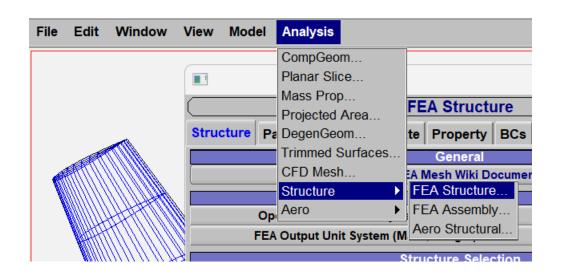
- DomainのGenerate Half Meshを チェックして作成
 - Total Num Tris が2000ぐらいだとそこ そこ早く解析が終わる
- UNLSIで使用するのはvspgeomなので、Outputでこれだけにしてもよい



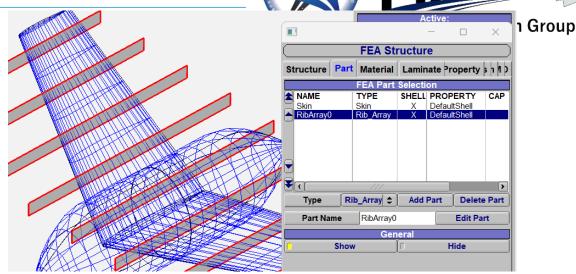


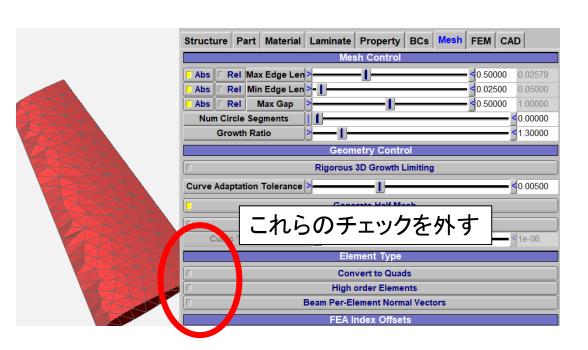


構造設計・FEAメッシュ(構造メッシュ)の作成



- Structure→Add Strucure
- Part → Add Rib_Array
- あとは空力メッシュと同様にメッシュを作成
- FEAS tructureではExport FEMを押さないとファイルが書き出されないので注意





UNLSIによる空力解析

HAPS

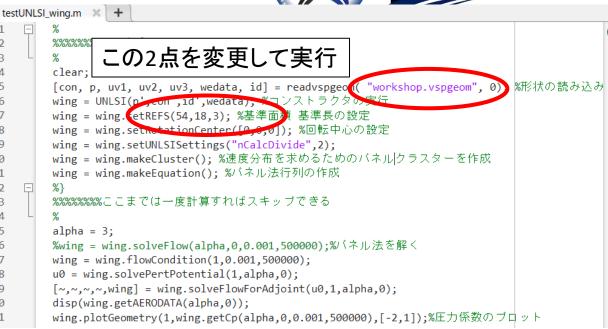
- ・ 最初に作成したworkshopフォルダにopenVSP で作成した「workshop.vspgeom」と 「workshop_wingGeom_Struct0.msh」をコピー
- Readvspgeomの引数と基準値を変更して解析 を実行

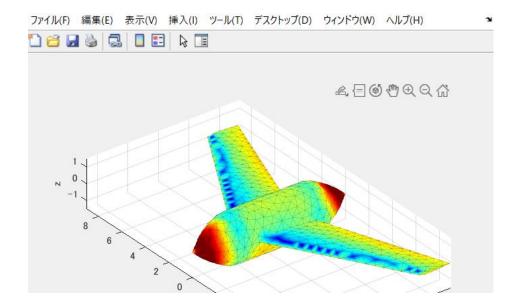
12

13

14

- 解析例ではadjoint法っぽく解析を実行している 15 が、コメントアウトされたsolveFlow関数の実行で 18 もOK
- 終わったら構造解析に向けて 「testUNLSI_wing」をリネームしてください



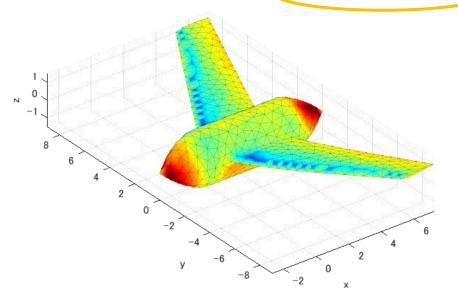


UNLSIによる構造解析

HAPS
Research Group

- ・ example→unlsi→wingStructureからtestUNLSI_wingをコピー
- 必要な箇所を変更
 - 読み込みファイルの名前
 - setFemMaterials([1,2,3],~~~)をsetFemMaterials([1,2],~~~)に変更
- 解析実行!
- うまくいくと右図のように変位した図が出てくる
- 動圧を変えたりしてどのように変化するか試してみよう
- 終わったら空力弾性解析に向けて「testUNLSI_wing」 をリネームしてください

```
[con, p, uv1, uv2, uv3, wedata, id] = readvspgedm( "workshop.vspgeom
wing = UNLSI(p',con',id',wedata,1); %コンストラクタの
wing = wing.setREFS(80,20,4); %基準面積 基準長の設定
    = wing.setRotationCenter([0,0,0]); %回転中心の設定
    = wing.setUNLSISettings("nCalcDivide",2);%パネル法行列の作成における分割性
    = wing.makeCluster(); %速度分布を求めるためのパネルクラスターを作成
wing = wing.makeEquation(); %パネル法行列の作成
%%%%%%%ここまでは一度計算すればスキップできる
alpha = 10;
Re = Vinf * 4 / 1.512 * 1e5;
wing = wing.solveFlow(alpha,0,0.001,Re);%パネル法を解く
wing.plotGeometry(1,wing.getCp(alpha,0,0.001,Re),[-3,1.5]);%圧力係数のブロッ
disp(wing.getAERODATA(alpha,0));
[con,verts,femID] = readFemMesh('workshop_WingGeom_Struct0.msh
wing = wing.setFemMesh(verts,con,femil 1); ** + ** T T T
[wing,weight] = wing.setFemMateria(s([1,2],[0.002,0.003],[73500000000],73500
```

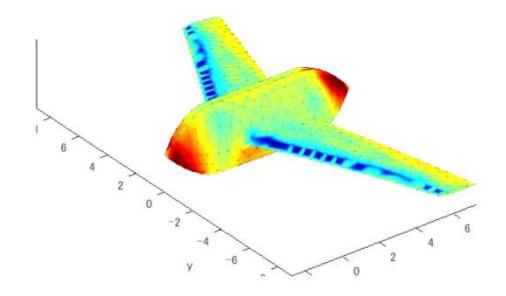


UNLSIによる空力弾性解析



- ・ example→unlsi→wingAeroelasticから testUNLSI_wingをコピー
- 必要な箇所を変更
 - 読み込みファイルの名前
 - setFemMaterials([1,2,3],~~~)をsetFemMaterials([1,2],~~~)に変更
- 解析実行!
- 線形シェルFEMの剛性行列は「元の形状」に対して作成されているので、wingとwing2のように「変形前」と「変形後」を考慮する必要がある。

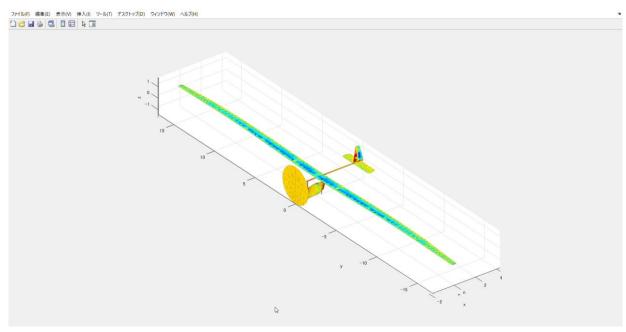




人力飛行機モデル



- ・ 「空力解析」「構造解析」「空力弾性解析」までやってみよう。
 - Example→HPAにてopenVSPファイルを開き、 ちょっとパラメータをいじってみる
 - 2. testUNLSI_HPAを実行し空力解析をしてみる
 - Example→HPAstructureからコードをコピーし、 構造解析を実行してみる
 - 4. Example→HPAaeroelasticからコードをコピーし、 空力弾性解析を実行してみる。
- ここまでできたら、空力微係数・補間曲面作成をやってみよう
 - aeroRBF.matが生成される(計算コスト大)



```
wing = wing.solveFlow(alpha,beta,0.001,300000);
wing.plotGeometry(1,wing.getCp(0,0,0.001,300000),[-2,1]);%圧力係数のプロット
disp(wing.getAERODATA(aloba,bil,1.001,0000),[-2,1]);%圧力係数の補間曲面の作成
wing = wing.makeSurrogateModel([-10,-5,0,5,10],[-10,-5,0,5,10],0.0001,300000);
[ppCoef,ppDyn] = wing.getSurrogateModel();
wing.plotSurrogateModel(ppCoef,ppDyn,linspace(-10,10,100),linspace(-10,10,100),0.001,1);
UREF = 7.2;
Mach = 0.001;
REFS = [wing.SREF,wing.BREF,wing.CREF];
save aeroRBF.mat ppCoef ppDyn REFS mass Inatia UREF Mach -mat;
%}

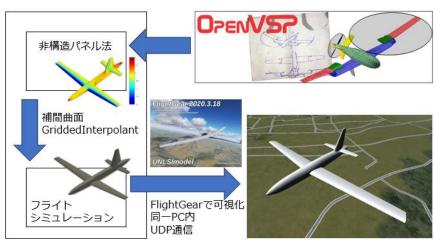
example → HPA → testUNLSI HPA
```

FEEのインストールと動作確認



FEEのインストール

- 1. 以下のgithubからソースコードを入手(要招待)
 - https://github.com/NaotoMoritaUT/FEE
- 2. FEE/libにMATLABのパスを通す。
- 3. simpleSateliteにディレクトリを移動して、「FEEmain();」とコマンドウィンドウに打ち込む
- 4. エラーが出なければ成功
- 5. 以下のgithubから人力飛行機モデルを入手
- 6. aeroRBF.matが存在することを確認



FlightGearのセットアップ

- 1. 「UNLSImodel」をC:¥Program Files¥FlightGear 2020.3¥data¥Aircraft にコピー
- 2. 「getGroundLevel.xml」をC:¥Program Files¥FlightGear 2020.3¥data¥Protocolにコピー
- 3. 人力飛行機モデルのディレクトリでrunFG.datを右クリック→MATLABの外部で実行
- 4. FlightGearの起動を確認
- 5. もう一度simpleSateliteを実行し、可視化がうまく行っ ているかを確認

FEE構成ファイルの解説



- necessarySetting.m
 - シミュレーション実行に絶対必要な変数を記述
- vehcleSetting.m
 - ユーザー設定をveh構造体に格納する。vehicleSetting.m→necessarySetting.mの順で実行される
- SimulationForces.m
 - ダイナミクスを記述する。それぞれの軸での力とモーメントを記述。ここでのveh構造体への変更は保存されない
- SimulationControl.m
 - veh.dt秒ごとに呼ばれる。veh構造体への変更を記述。積分の切れ目なので"何をしてもよい"
- SimulationWind.m.
 - 風を記述する
- SimulationTerminal.m.
 - フラグを立てるとシミュレーションを終了する。結果の可視化などを記述する。

HPAシミュレーションの実行



- 当該フォルダへ移動し、runFG.batを右クリック→MATLAB外部で起動
- 起動したらコマンドウィンドウに「FEEmain();」を入力。
- シミュレーション開始!(停止はctrl+c)

