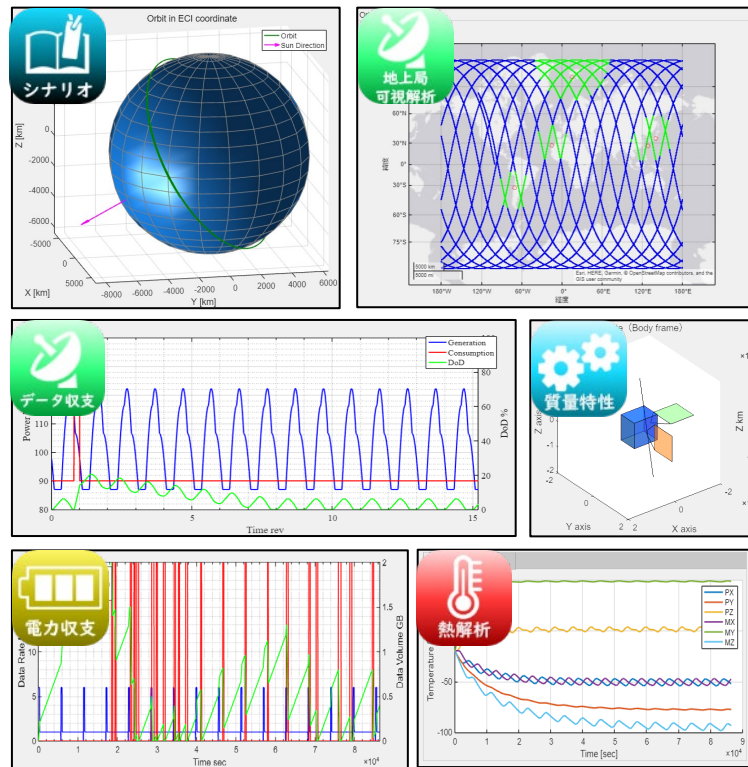


MATLAB Web アプリとコンテナを活用した 衛星概念検討フェーズでの 開発プラットフォームの事例



2022/5/25
宇宙航空研究開発機構
第3研究ユニット
水野 光



第三研究ユニットは、情報技術とシミュレーション技術による宇宙開発への貢献をめざしています。

JEDI's MISSION

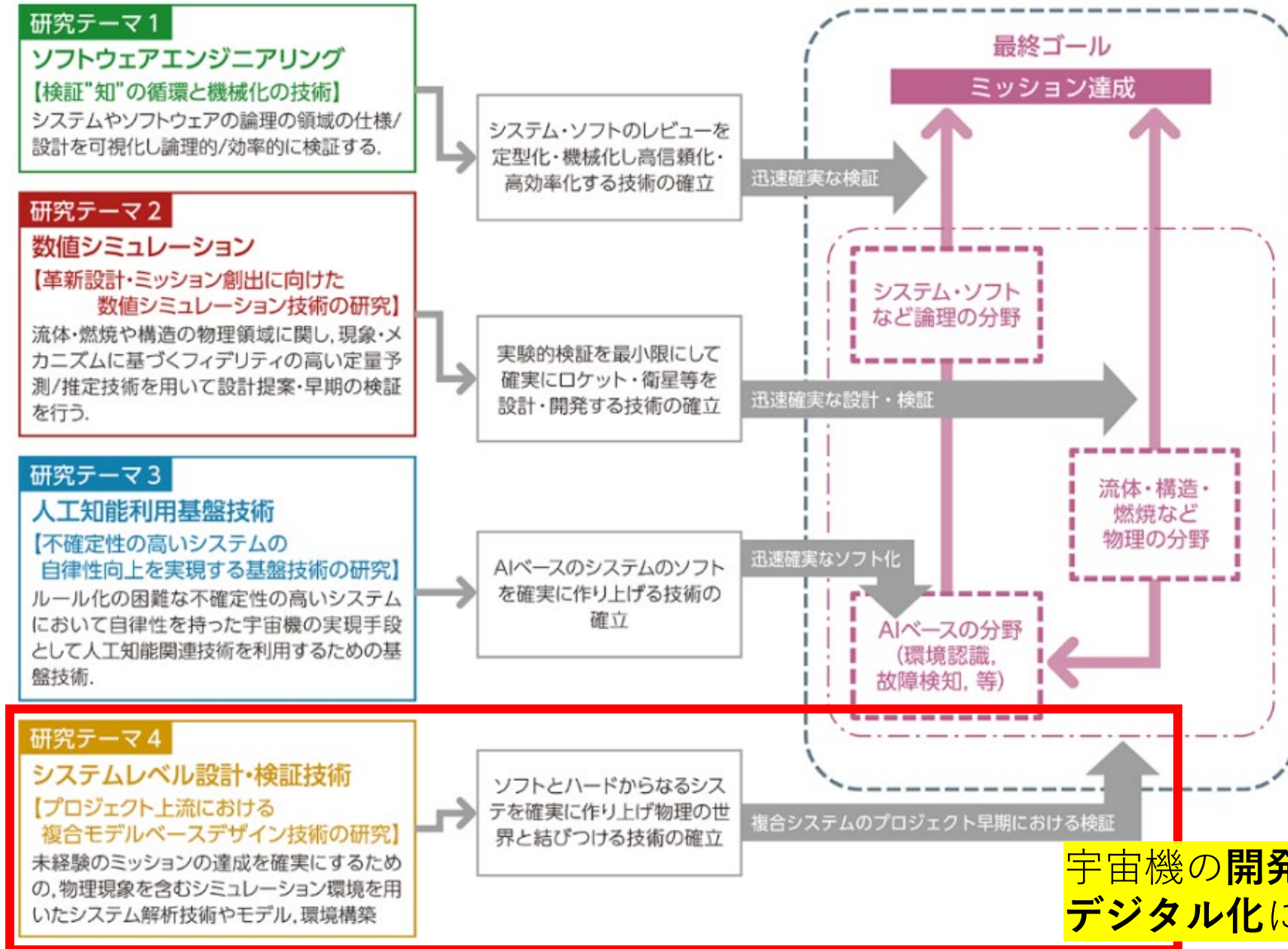
Mission 1. ITを活用した宇宙機・航空機プロジェクトの情報化に関わる研究開発
Mission 2. 数値シミュレーション技術の宇宙機・航空機開発の応用に関わる研究開発

このとおり：
宇宙ステーション補給機(HTV)

• 第三研究ユニット(JEDI)での取り組み

- これまでに培った数値シミュレーション技術やソフトウェア高信頼化手法のさらなる活用
 - より大きなシステムを迅速に評価するシステムレベルシミュレーション分野の強化
 - ソフトウェア IV&V で培った技術のより広い技術検証への取り組み
- AI(人工知能)分野における宇宙開発に重要な分野の見極め
- プロジェクト等の早い段階でのシステム成立性を見極めに役立つ技術開発

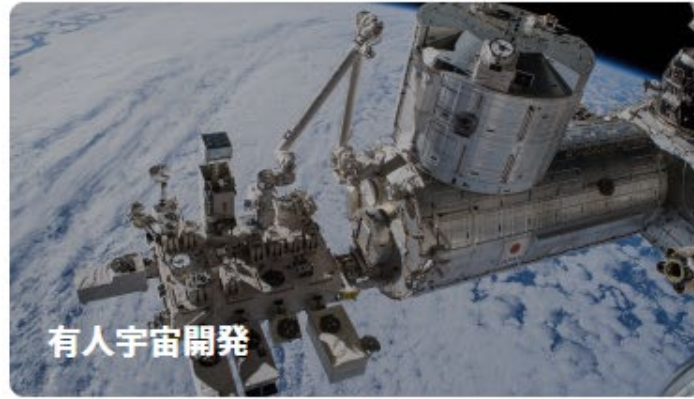
自己紹介：JAXA第三研究ユニットについて



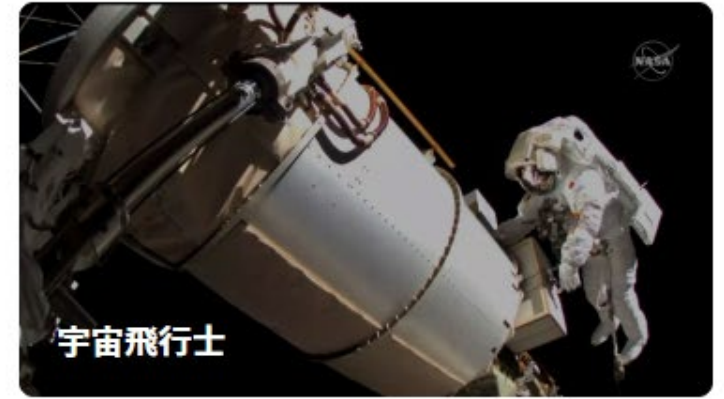
JAXA衛星開発プロセス（概念検討フェーズ）の課題



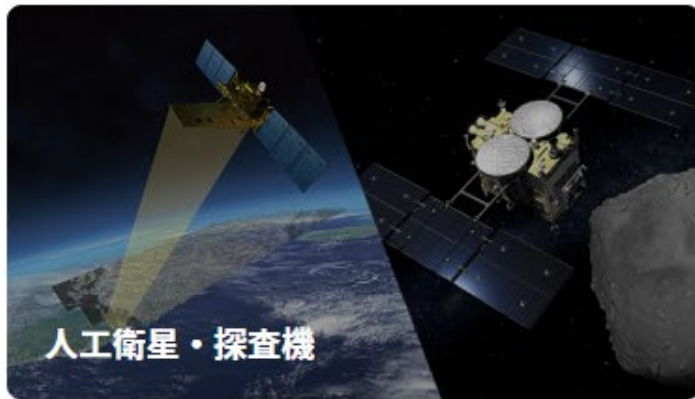
ロケット・輸送システム



有人宇宙開発



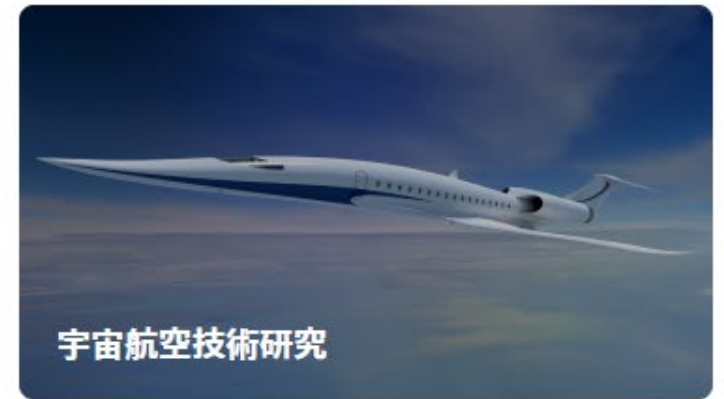
宇宙飛行士



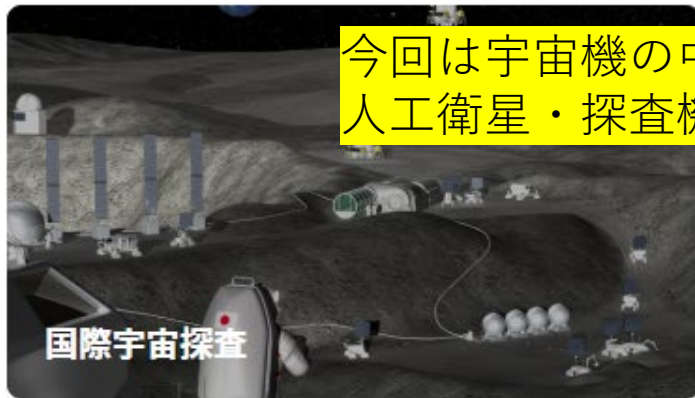
人工衛星・探査機



観測画像



宇宙航空技術研究



国際宇宙探査

今回は宇宙機の中でも
人工衛星・探査機が研究対象



事業所・施設



総合・その他

**JAXAがメインで
検討するフェーズ**

企画・概念検討

- ・メーカーとの前提条件や認識の齟齬を減らしたい
- ・フロントローディングしたい

**衛星メーカー
検討フェーズ**

設計

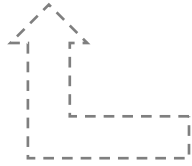
試作

製造

試験

**JAXA
運用フェーズ**

利用/運用



JAXAがメインで
検討するフェーズ

↓今回はここに注力！

企画・概念検討

- ・メーカーとの前提条件や認識の齟齬を減らしたい
- ・フロントローディングしたい

衛星メーカー
検討フェーズ

設計

試作

製造

試験

JAXA
運用フェーズ

利用/運用

デジタル技術(MBD/MBSE)を活用し、

①試作前にコンセプト等を検討/設計する**開発上流フェーズ**で
検討レベルの向上/標準化/フロントローディングをしたい

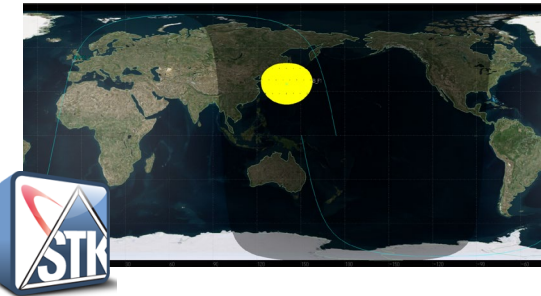
②設計/試作前の**モノが無い段階**で各システム**成立性**を見極め
様々なコンセプトを**クイックに評価/取捨選択**したい

「最上流で何をどこまで考
えた？」を明確にしたい

JAXA衛星開発プロセス（企画・概念検討フェーズ）の課題




軌道解析
（β角・地上局可視解析）



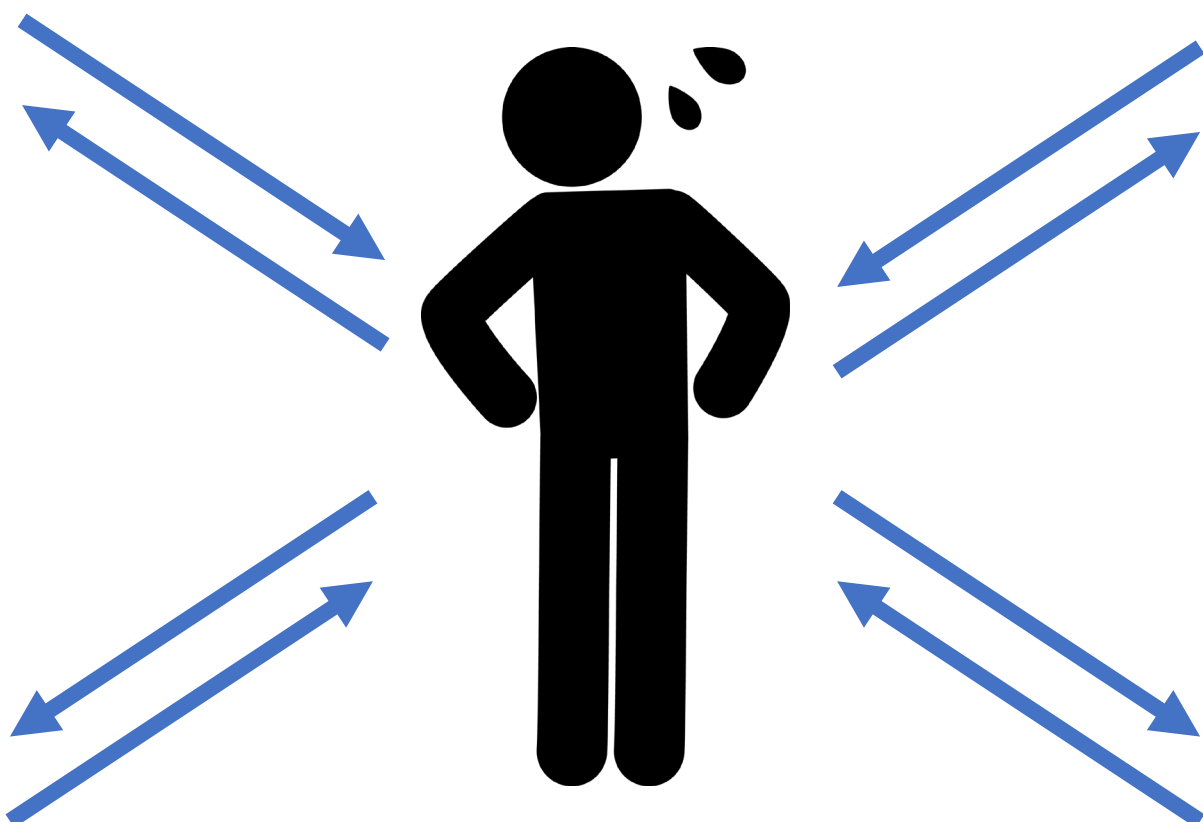
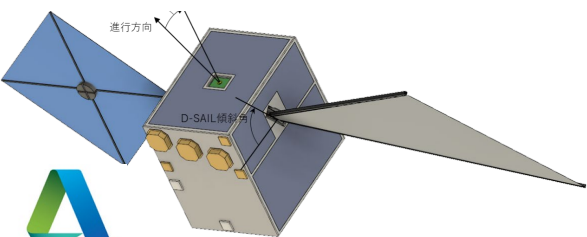
従来は担当が
専用ツール/ソフトで個別に計算・解析

各種バジェット解析
（質量・電力・通信・姿勢）

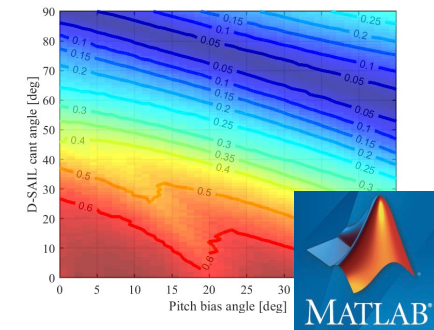


項目	単位	値	許容値	余裕率	備考
中心周波数	MHz	2110.0	2290.0	8400.0	2025-2110, 2200-2290, 8025-8400.0 余裕
送信機出力	W	3.6	3.6	1.0	
伝送損失	dB	-15.5	0.0		
伝送電力	dB	4.5	1.0		
伝送電力損失	dB	-3.0	13.5		伝送機は変動不定時を考慮した値
伝送電力損失	dB	-23.0	12.5		0.0 欠陥はIsosurfアンテナのためゼロとする
伝送電力損失	dB	0.0	0.0		
伝送電力損失	dB	570.0	570.0		
伝送電力損失	dB	5.00	5.00		0.04e(1) 1dB
伝送電力損失	km	2258.6	2258.6	2258.6	
伝送電力損失	dB	166.0	166.7	176.0	20log(10(4πR ²))
伝送電力損失	dB	0.5	0.5		
伝送電力損失	dB	0.3	0.3	0.5	
伝送電力損失	dB	0.0	0.0	0.0	
伝送電力損失	dB	0.0	0.0	0.0	
伝送電力損失	dB	-113.8	-105.5	-168.4	10log(10(4πR ²)) + 22.8 6 天窓損失
伝送電力損失	dB	-6.0			
伝送電力損失	dB	4.5			
伝送電力損失	K				
伝送電力損失	dBK	39.1			
伝送電力損失	dBK	-48.8	17.0	38.3	10log(10(4πR ²)) + 22.8 6 天窓損失
伝送電力損失	dB	0.0	0.0	1.0	
伝送電力損失	dB	0.0	0.2	0.2	伝送電力に含む
伝送電力損失	dB + Hz	63.2	55.1	66.0	再帰損失電力 + 0.7 + 22.8 6 天窓損失
伝送電力損失	dB				
伝送電力損失	dB	10.5	8.6	7.8	10log(10(4πR ²)) + 22.8 6 天窓損失
伝送電力損失	dB	3.5	2.5	1.0	10log(10(4πR ²)) + 22.8 6 天窓損失
伝送電力損失	dB	0.0	5.2	0.0	10log(10(4πR ²)) + 22.8 6 天窓損失
伝送電力損失	dB	4.0	64.0	33000.0	
伝送電力損失	dB	36.0	48.1	75.1	
伝送電力損失	dB	12.0	8.0	0.0	
伝送電力損失	dB	62.0	55.0	83.9	
伝送電力損失	dB	3.2	0.1	5.6	10log(10(4πR ²)) + 22.8 6 天窓損失

機器配置・質量特性



姿勢解析



【MATLAB】

- 数値計算～多様なプロットによる可視化～GUIアプリ化～Webアプリ化まで**一貫して実現可能**
- 他エンジニアリングツール（STK/AMESIM/SimulationX/ADAMSなど）との**豊富な接続性**
- Simulink®/StateFlow®を含む**複雑なモデルもWebアプリ化可能**
- SimulinkやオートコードによるSILS/HILSなどの**後工程との親和性/拡張性**

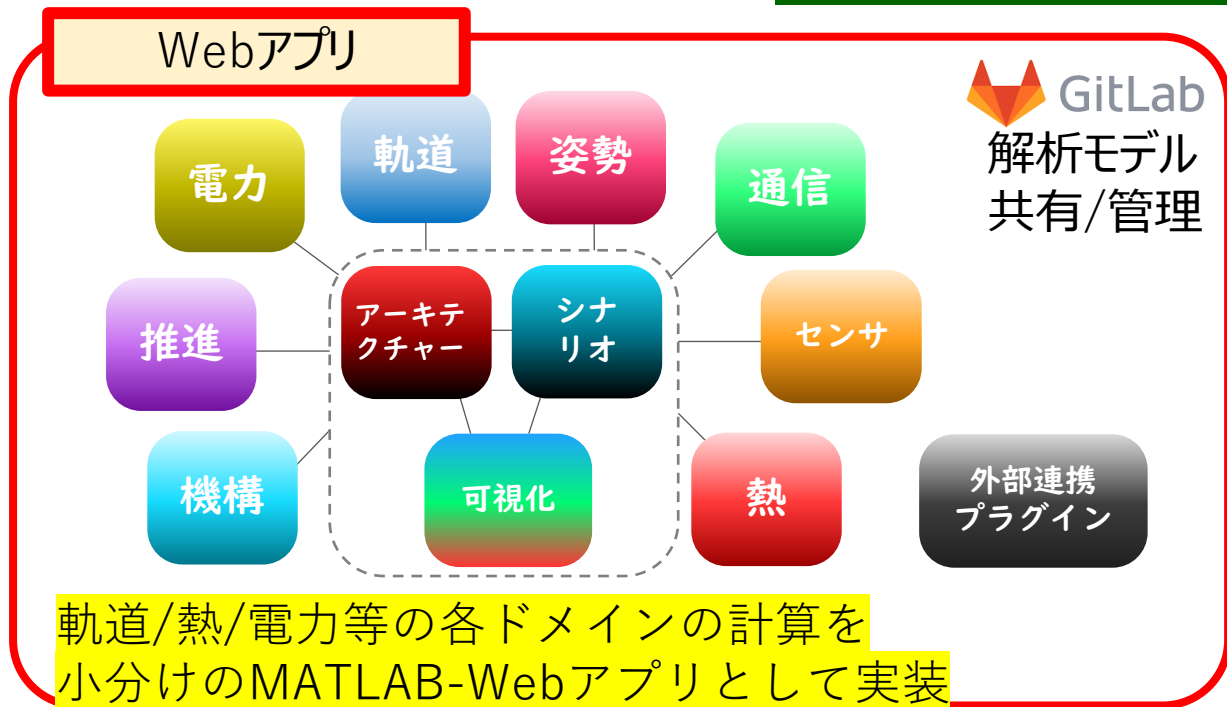
【Webアプリ】

- Webブラウザのみで**インストール不要、MATLABライセンス不要**ですぐ社内で利用可能
- 従来より**多くの人**がシミュレーションを実行可能
- GUIアプリにより、計算のやり方/手順/計算ロジックを**ある程度標準化が可能**
- みんなが**同一のモデル/方法/前提条件**の元で議論できるようになる

【コンテナ】

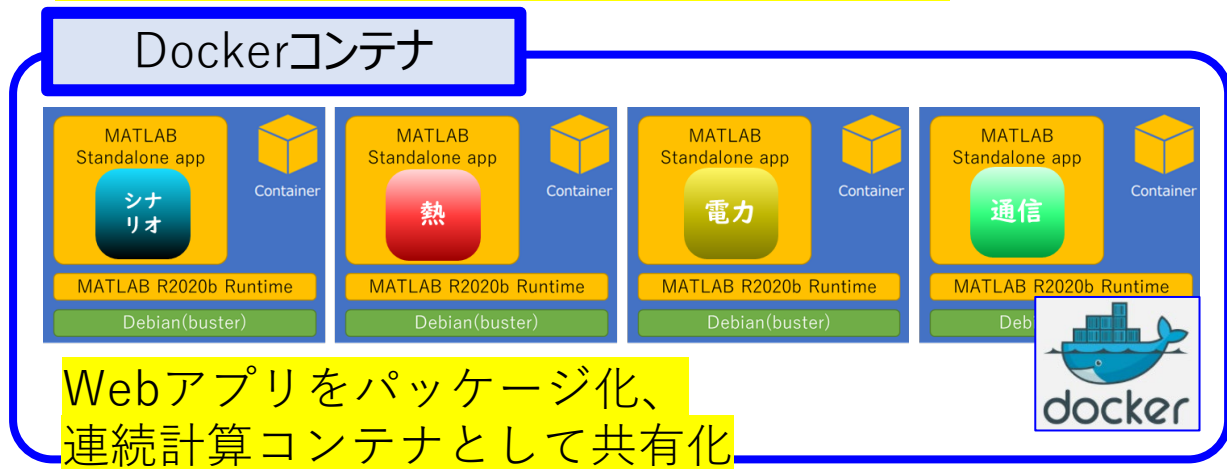
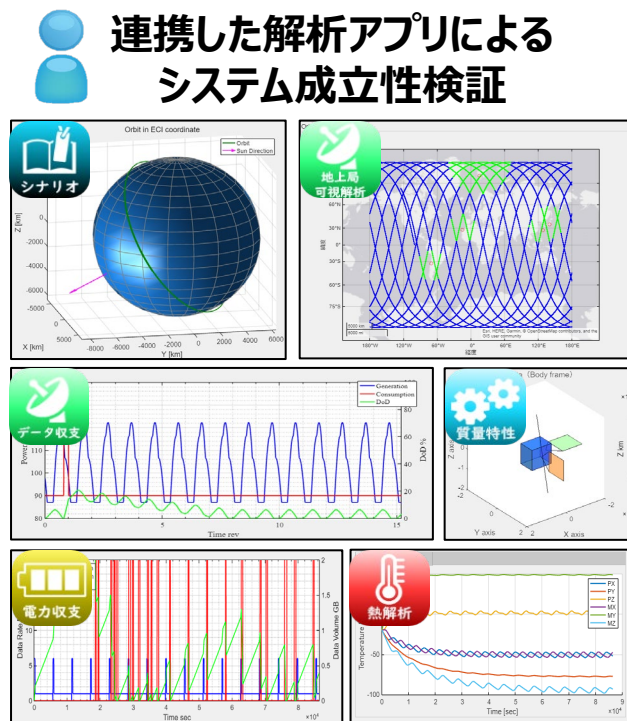
- Webアプリが苦手とするパラスタやモンテカルロSim等の**計算の連続実行が可能**
- **コンテナは実行環境に依存しない**。Windows/Mac/Linuxのどの環境でも動作可能
- スーパーコンピュータやパブリッククラウドでも実行可能。計算機能力の拡張性。
- 実行時は**MATLABライセンス不要**

Concurrent Design Platform



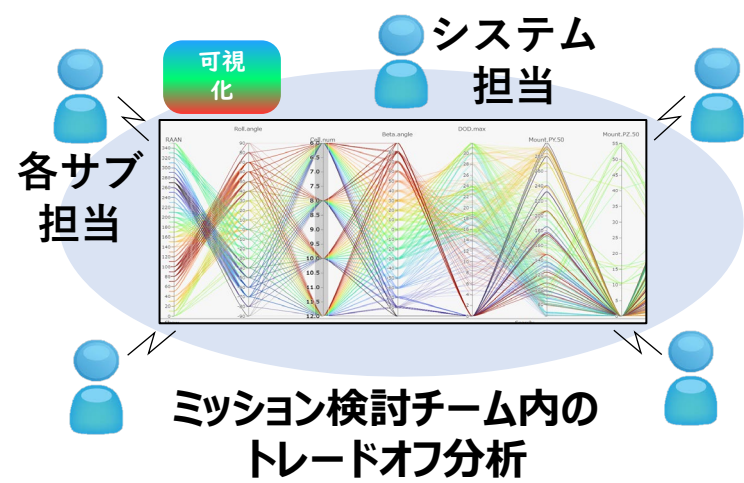
Web上で条件入力

解析結果DL (mat/csv)



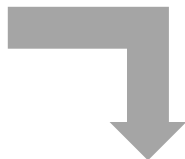
可視化アプリに入力

コンテナのDL連続実行



データ収支解析ワークフロー

データ収支解析ワークフロー



Webアプリ③“データ収支”

- ・衛星の通信関係パラメータ
- ・“シナリオ”の計算結果
- ・“地上局解析”の計算結果

↓
データ収支結果を
MATファイルで出力

Webアプリ①“シナリオ”

- ・衛星の軌道パラメータ
- ・シミュレーション時間

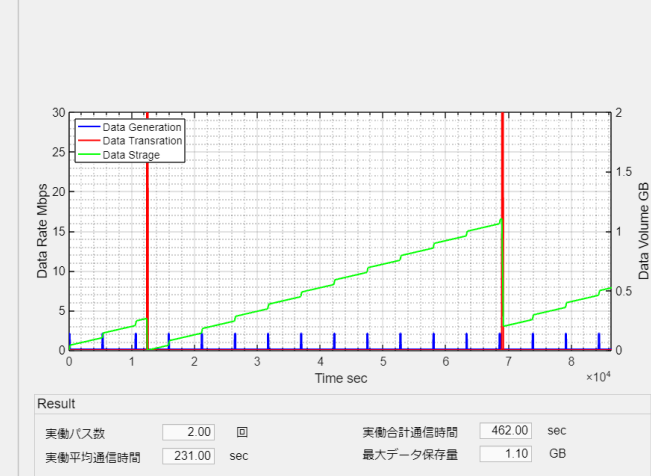
↓
軌道計算結果を
MATファイルで出力

Webアプリ②“地上局可視解析”

- ・地上局の緯度経度
- ・“シナリオ”の計算結果

↓
地上局可視情報を
MATファイルで出力

The screenshot displays the software interface for satellite simulation and data budget analysis. It includes sections for 'Initial condition' (Sun-sync Sub-recurrent), 'Input' (File Input), 'Output Table' (listing Epoch, Time, X_eci, Y_eci), 'Orbit in ECI coordinate' (3D plot), 'GroundStationPosition' (Tsukuba, Japan), 'Orbit Path' (2D map), and 'OrbitalParamPanel' (Semi-Major axis, Eccentricity, etc.).



JAXA Concurrent Design Platform操作デモ (Webアプリ)



データ
収支解析
ワーク
フロー



Webアプリで順次シミュレーションを実行することにより、様々なワークフローの構築/検討が実施可能

JAXA Concurrent Design Platform (Dockerコンテナ)



パラメータスタディのような入力パラメータを
少しずつ変更した計算を連続実行したいニーズに対応：

計算条件（入力パラメータの値）をExcelシートに記述

保存ファイル名

入力パラメータ

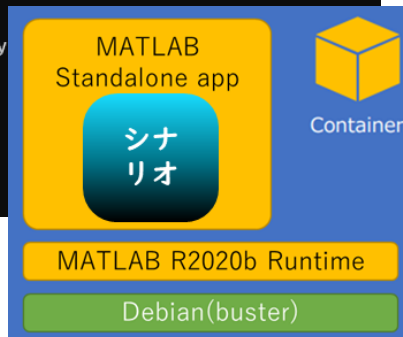
ResultFileName	StartTime	StopTime	OutputRate	UseAirDrug	AreaMassRatio	Cd	UseSRP	AreaMassRatio_SRP	Cr	AttitudeControlMode
TestCalc_Keplerian_LVLH	20 Mar 2022 12:00:00	20 Mar 2022 15:00:00	1	0	0	0	0	0	0	LVLH
TestCalc_Keplerian_LVLH_bias	20 Mar 2022 12:00:00	20 Mar 2022 15:00:00	1	0	0	0	0	0	0	LVLH+bias
TestCalc_Keplerian_SunPointing	20 Mar 2022 12:00:00	20 Mar 2022 15:00:00	1	0	0	0	0	0	0	Sun pointing
TestCalc_Circular	20 Mar 2022 12:00:00	20 Mar 2022 15:00:00	1	0	0	0	0	0	0	LVLH
TestCalc_Geo-synchronous	20 Mar 2022 12:00:00	20 Mar 2022 15:00:00	1	0	0	0	0	0	0	LVLH
TestCalc_Sun-synchronous	20 Mar 2022 12:00:00	20 Mar 2022 15:00:00	1	0	0	0	0	0	0	LVLH
TestCalc_Rec-current	20 Mar 2022 12:00:00	20 Mar 2022 15:00:00	1	0	0	0	0	0	0	LVLH
TestCalc_Rec-Sub-recurrent	20 Mar 2022 12:00:00	20 Mar 2022 15:00:00	1	0	0	0	0	0	0	LVLH

例えば↑の回帰日数を
33,34,35,36,・・・のように
したい場合などを想定

The screenshot shows the software interface with various input fields and a 3D plot of an orbit in ECI coordinates. The plot shows a blue sphere representing Earth with a green orbit path and a pink arrow indicating the Sun's direction.

Dockerコンテナ内のCLIから計算条件Excelを指定し
シェルスクリプトにて計算実行

```
mizunoh@JEDIMBL2111: ~/doc
root@24845bd0e1ee: /home# /opt/mcr/application/run_ScenarioCalc.sh /opt/mcr/v99 InputParamsOfScenarioApp_testInput.xlsx
Setting up environment variables
---
LD_LIBRARY_PATH is ./opt/mcr/v99/runtime/glnxa64:/opt/mcr/v99/bin/glnxa64:/opt/mcr/v99/sy
inputTable =
```



- 計算結果MATファイル
- Scenario_1.mat
 - Scenario_2.mat
 - Scenario_3.mat
 - Scenario_4.mat
 - Scenario_5.mat
 - Scenario_6.mat
 - Scenario_7.mat
 - Scenario_8.mat
 - Scenario_9.mat
 - Scenario_10.mat
 - Scenario_11.mat
 - Scenario_12.mat
 - Scenario_13.mat
 - Scenario_14.mat

得られたMATファイルを次のWebアプリまたはコンテナで読み込み
計算を続ける

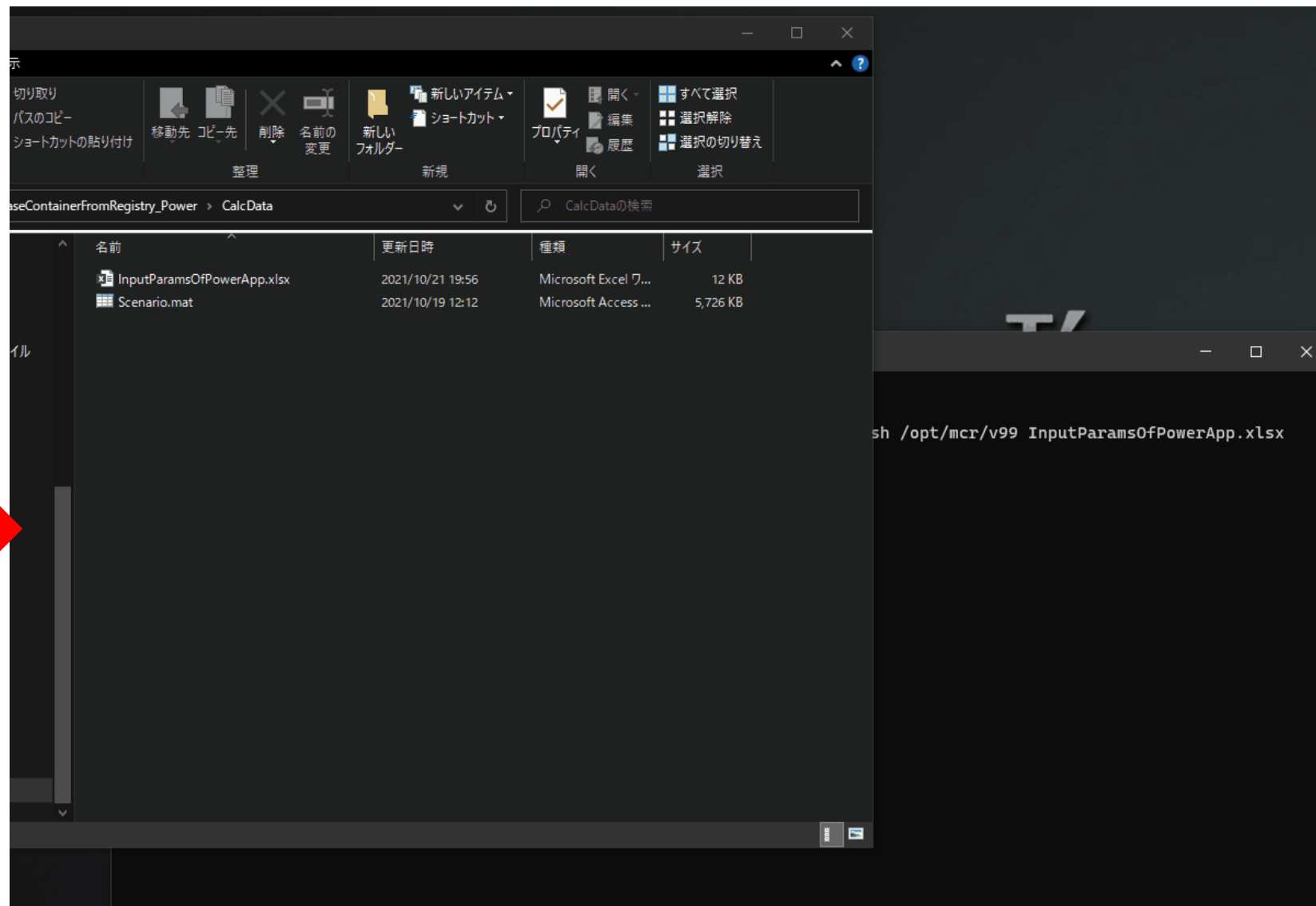
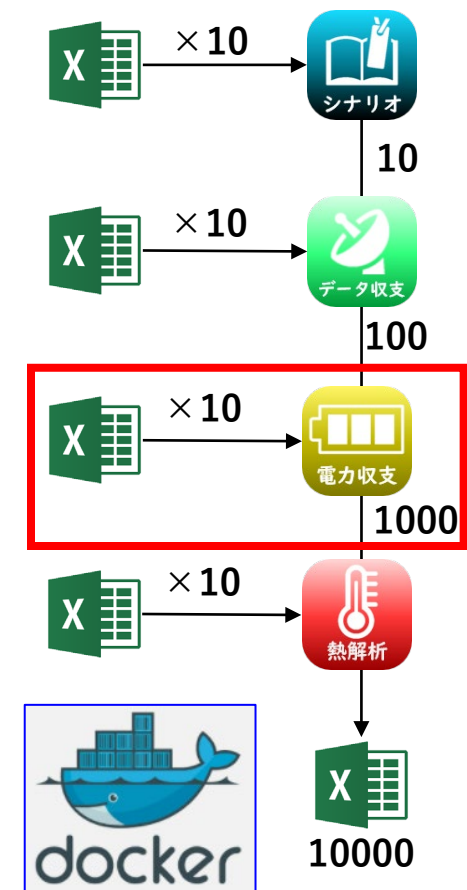


JAXA Concurrent Design Platform操作デモ（コンテナ）



コンテナ化技術により連続計算が可能となり、開発早期のシステム検討の高度化が可能となった
→ JAXAスーパーコンピュータ（JSS3）を使った大規模並列計算も可能（後述）

パラスタ例：
軌道/熱/電力/データ収支の
成立性検討



【背景】

- 宇宙機開発のモデルベース化/デジタル化
- 開発早期の検討レベル高度化/フロントローディング

【課題】

- 従来多種多様なツールを担当が独自に組合わせて検討→知見の人依存
- 社内でシミュレーションをやれる人材不足

【解決策】

- MATLAB
- Webアプリ
- コンテナ

3つの技術を組み合わせた宇宙機概念検討プラットフォーム
“Concurrent Design Platform”を構築

後半では解決策のMATLAB関連の技術的なトピックを紹介

【MATLAB絡みの技術トピック】

- MATLAB Web Appによるシミュレーション環境の共有
 - MATLAB CompilerによるApp Designer GUIアプリのWebアプリ化と共有
- Dockerコンテナによるシミュレーション連続実行
 - MATLAB CompilerによるLinux環境でのStandalone Application化とコンテナ化
 - JAXAスパコン上での実行とコンテナ共有プラットフォーム

【まとめと今後の方向性】

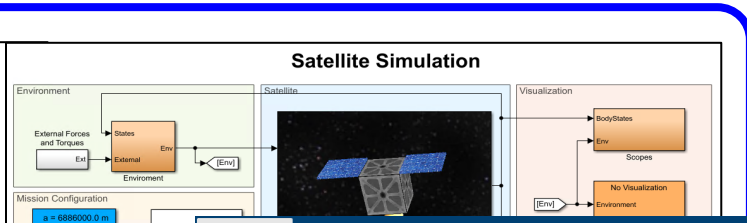
- 更に開発上流の検討PFに向けて：衛星コンステレーション/衛星サービス検討
- ユースケースを増やし、広げていく：衛星メーカー/大学共有、などなど

MATLAB Web Appによるシミュレーション環境の共有



スクリプトMファイル(+ Simulinkモデル)

```
clear all; clc;  
.  
.  
.  
out = sim(in);  
.  
.  
.  
plot(time, data);
```



App Designerによる
GUIアプリ化

MATLAB® Compiler™
Simulink® Compiler™
によるWebアプリ化

Web アプリ
MATLAB Compiler を使用して配布用 Web アプリを作成します

```
classdef ScenarioApp < matlab.apps.AppBase  
    % Properties that correspond to app components  
    properties (Access = public) ...  
  
    % Properties that correspond to apps with auto-reflow  
    properties (Access = private)  
        onePanelWidth = 576;  
    end  
  
    properties (Access = public) ...  
  
    properties (Access = private) ...  
  
    methods (Access = private) ...  
  
    methods (Access = public) ...  
  
    % Callbacks that handle component events  
    methods (Access = private)  
  
    % Code that executes after component creation  
    function startupFcn(app) ...  
  
    % Button pushed function: RunButton  
    function RunButtonPushed(app, event) ...  
  
    % Value changed function: ControlModeDropDown  
    function ControlModeDropDownValueChanged(app, event) ...  
  
    % Button pushed function: SaveButton  
    function SaveButtonPushed(app, event) ...
```

計算と Simulink の実行は
コールバック内に実装



WebApp
Serverで
ホスト

Webアプリ実行は
MATLABライセンス不要

Epoch	Time [sec]	X_eci [m]	Y_eci [m]	Z_eci [m]
2022 12:00:00.000	0	6.8859e+06	-3.4209e+04	-1.4804e+01
2022 12:00:01.000	1	6.8859e+06	-3.5216e+04	-7.3224e+00
20 Mar 2022 12:00:02.000	2	6.8859e+06	-3.6222e+04	219.0564
20 Mar 2022 12:00:03.000	3	6.8859e+06	-3.7228e+04	7.7605e+00
20 Mar 2022 12:00:04.000	4	6.8859e+06	-3.8234e+04	1.5302e+01
20 Mar 2022 12:00:05.000	5	6.8859e+06	-3.9240e+04	2.3000e+01

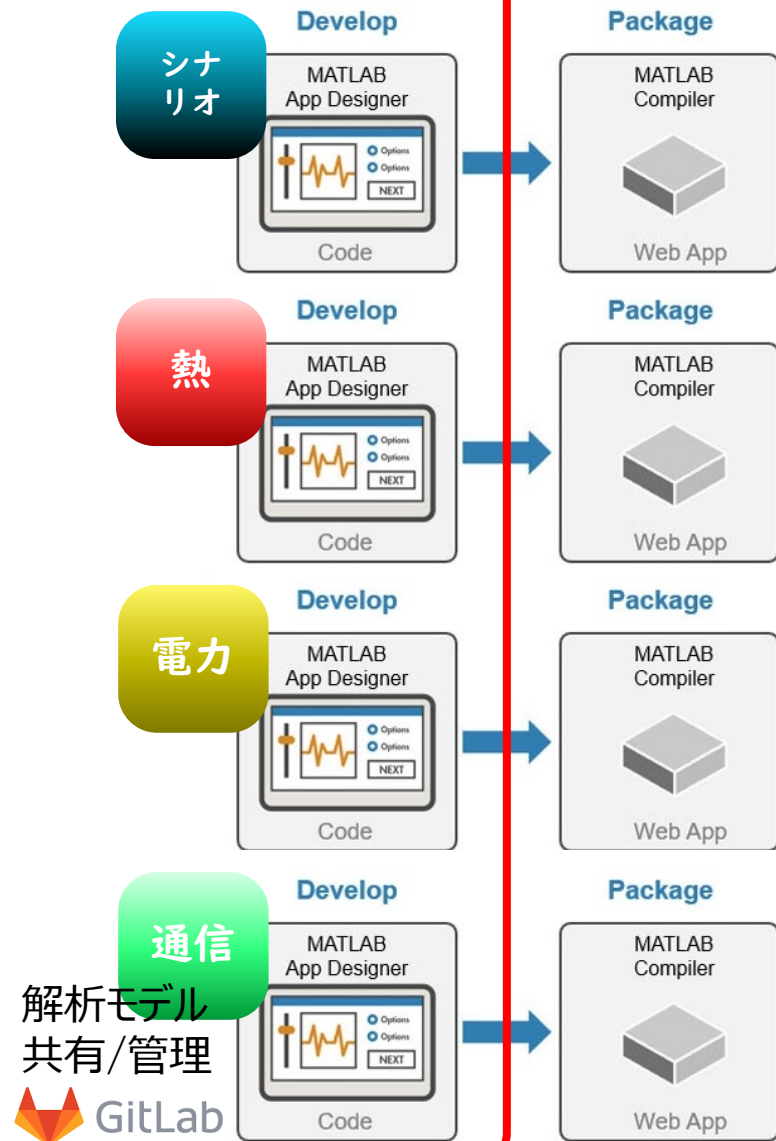
Orbit in ECI coordinate

SimulinkやStateFlowを含むモデルもWebアプリ化可能！
→他のプログラミング言語には無いMATLABならではのメリット！

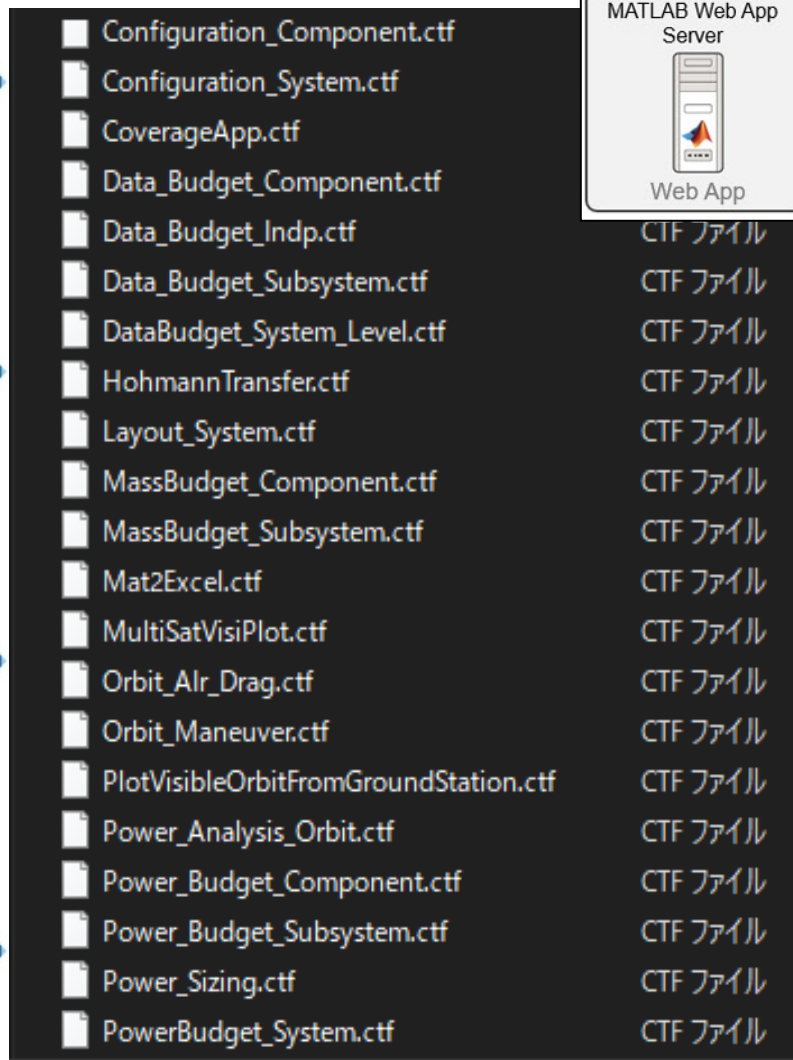
MATLAB Web Appによるシミュレーション環境の共有



開発者



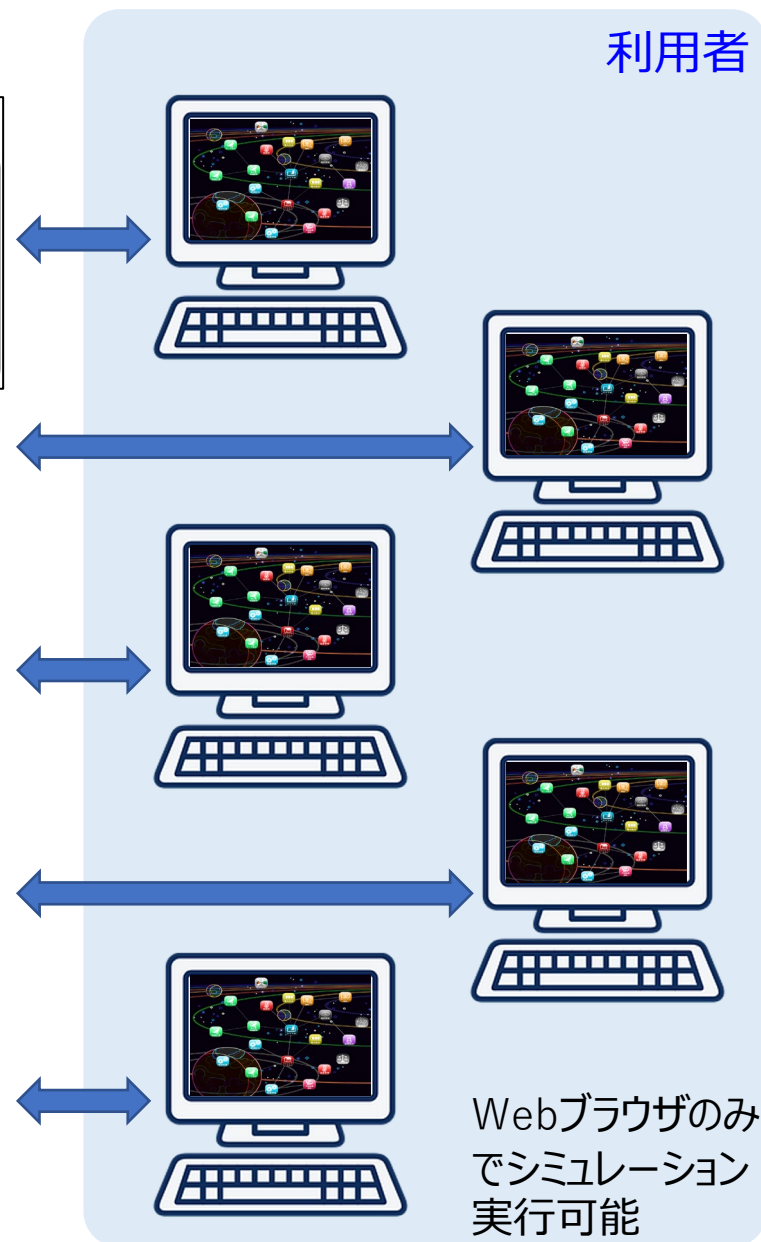
解析モデル
共有/管理
GitLab



Share



利用者



開発者

利用者



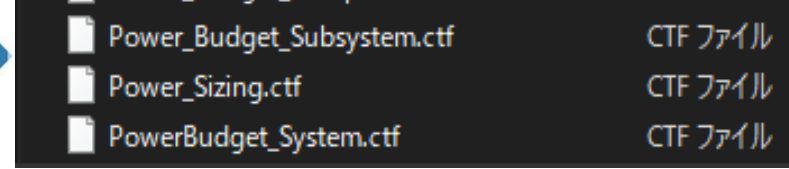
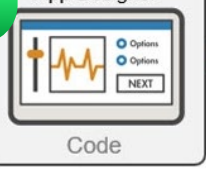
【利用者としてのWebアプリのメリット】

- Webブラウザのみでいつでもどこでも誰でも利用可能
- 計算結果だけでなく**計算ロジックも共有**できる
- Webアプリ実行は**MATLABライセンス不要**
- MATLAB無しでMATLABの高度な計算が実行できる

【開発環境としてのMATLABのメリット】

- Mファイル→GUI App→Web Appまで**ワンストップ**
- Web開発の**HTML/CSS/JavaScriptの知識不要**
- MATLAB/Simulinkの計算/モデル/コマンドが利用可能
- MATLAB ProjectとGitの組合せにより開発環境の共有が可能

解析モデル
共有/管理



Webブラウザのみ
でシミュレーション
実行可能

【MATLAB絡みの技術トピック】

- MATLAB Web Appによるシミュレーション環境の共有
 - MATLAB CompilerによるApp Designer GUIアプリのWebアプリ化と共有
- Dockerコンテナによるシミュレーション連続実行
 - MATLAB CompilerによるLinux環境でのStandalone Application化とコンテナ化
 - JAXAスパコン上での実行とコンテナ共有プラットフォーム

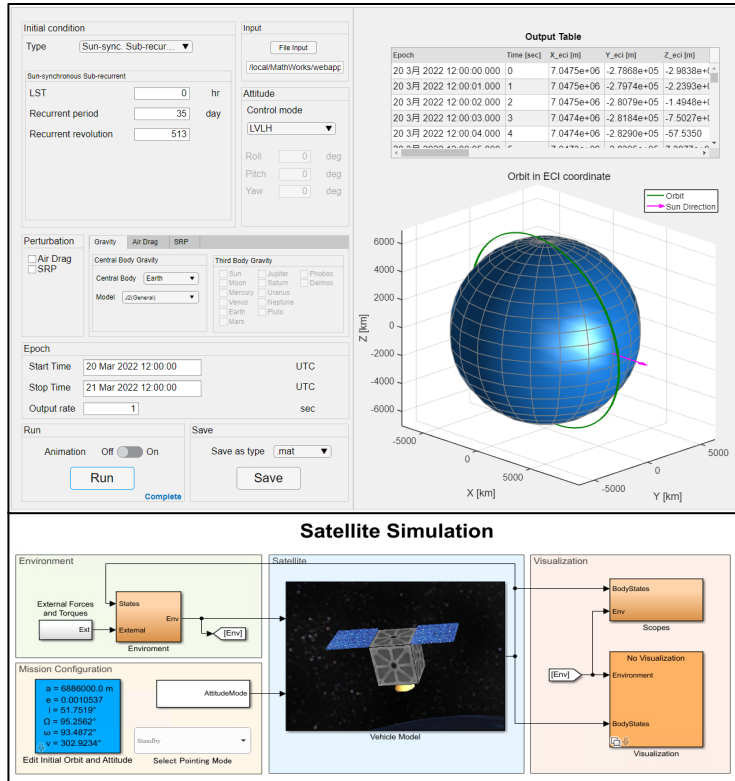
【まとめと今後の方向性】

- 更に開発上流の検討PFに向けて：衛星コンステレーション/衛星サービス検討
- ユースケースを増やし、広げていく：衛星メーカー/大学共有、などなど

Dockerコンテナによるシミュレーション連続実行

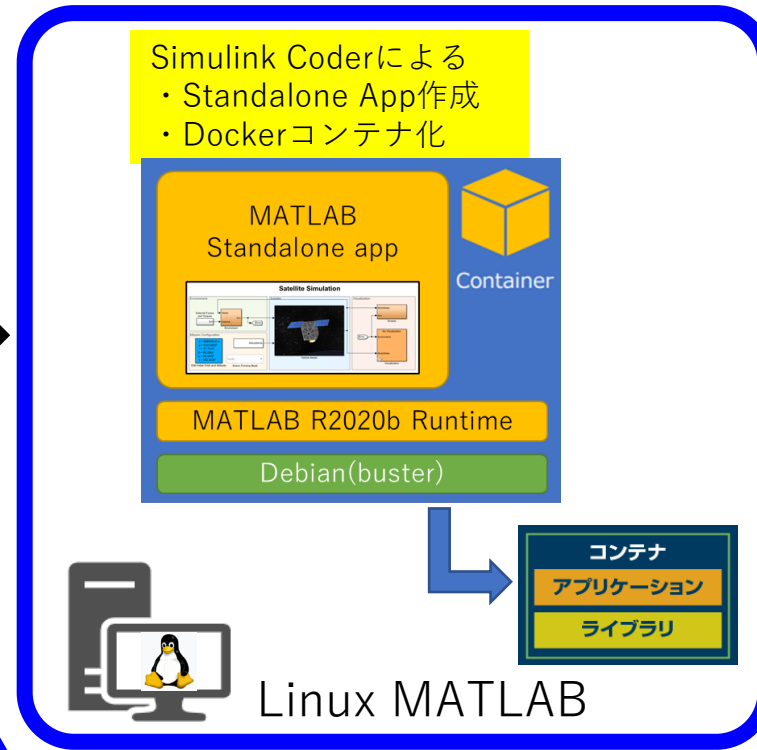


MATLAB/Simulinkモデル



モデル作成者

Windows MATLAB + WSL2



コンテナ取得

コンテナ取得

コンテナ取得

コンテナ取得



MATLAB AppDesigner GUIアプリ → Dockerコンテナ構築方法



最小限の手間でAppDesignerアプリを自動化する方法：クラス継承と関数のオーバーライドの活用

熱解析アプリ(ThermalAnalysis_System.mlapp)の例：

【アイデア】

AppDesignerのアプリは巨大なクラスなので、クラスの作法を使えば、基本的にGUIアプリの操作（ボタンを押す、数値を入力するなど）は全てMファイルのスクリプトとして記述できる



The screenshot shows the MATLAB App Designer interface. The code editor displays the class definition for ThermalAnalysis_System, which inherits from matlab.apps.AppBase. The layout designer shows a GUI with various controls and a plot. A red box highlights the class definition line: `classdef ThermalAnalysis_System < matlab.apps.AppBase`. A blue box highlights a callback function: `function ScenarioFileOpenButtonPushed(app, event) ...`. Another blue box highlights another callback function: `function SaveButtonPushed(app, event) ...`. A yellow box highlights the text: "ファイル読み込みと書き出しを実行するコールバックはWebアプリでは手動で(マウスで)操作するため自動化(Excel連携)できない".

AppDesignerアプリのクラスを継承したコンテナ実行用のクラスMファイルを作成 (=アプリの大部分のコードは再利用)

The screenshot shows the code editor for ThermalAnalysis_Container.m. A red box highlights the class definition line: `classdef ThermalAnalysis_Container < ThermalAnalysis_System`. A blue box highlights the methods section: `function ScenarioFileOpenButtonPushed(app, event) ...` and `function SaveButtonPushed(app, event) ...`. A yellow box highlights the text: "↑ 手動操作部のファイル入出力コールバックをExcelと連携する関数に書き換え(関数オーバーライド)".

↑ 手動操作部のファイル入出力コールバックをExcelと連携する関数に書き換え(関数オーバーライド)

MATLAB AppDesigner GUIアプリ → Dockerコンテナ構築方法



関数Mファイル

入力/出力MATファイル名

パラメータ

```

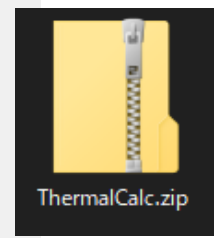
1 function ThermalCalc(fileName)
2
3 %% cleanup workspace
4
5 Excelに記載した計算条件を
6 Table形式で読み込み↓
7
8
9
10 % read excel table
11 inputTable = readtable(excelFileName, "UseExcel", false);
12 [numSim] = size(inputTable);
13
14 for idx=1:numSim
15
16 % create app instance
17 app = ThermalAnalysis_Container; ←クラス実体化
18
19
20 % define file names
21 app.fileName = inputTable.InputFileName{idx};
22 % app.pathName = excelFilePath;
23 app.saveFileName = inputTable.ResultFileName{idx};
24
25 % file open
26 app.ScenarioFileOpenButtonPushed; ←Excel連携の関数を実行
27
28 % update parameters
29 app.EditField_Es.Value = inputTable.Es{idx};
30 app.EditField_Ca.Value = inputTable.Ca{idx};
31 app.EditField_Ei.Value = inputTable.Ei{idx};
32 app.EditField_stph.Value = inputTable.stph{idx};
33 app.LxEditField.Value = inputTable.Lx{idx};
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58 % run simulation
59 tic
60 app.RunButtonPushed_pub;
61
62 % save mat file
63 app.SaveButtonPushed; ←Excel連携の関数を実行
64 toc
65
66
67
68
69
70
71
72
73
74 % show simulation status
75 fprintf('%s is saved.\n\n', app.saveFileName);
76
77 % delete app
78 delete(app);
79
80 end
    
```

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	InputFileName	ResultFileName	IniTemp	Es	Ca	Ei	stph	Lx	Ly	Lz	C_alm	C_MLI	MLI_px
2	Scenario.mat	Thermal1.mat	-20	1420	0.5	180	5.67E-08	1	1	1	880	1000	0
3	Scenario.mat	Thermal2.mat	-10	1420	0.5	180	5.67E-08	1	1	1	880	1000	
4	Scenario.mat	Thermal3.mat	0	1420	0.5	180	5.67E-08	1	1	1	880	1000	
5													



Linux環境でMATLAB Compilerにより Standalone Applicationとしてビルド

Zip圧縮されたStandalone Applicationが生成される



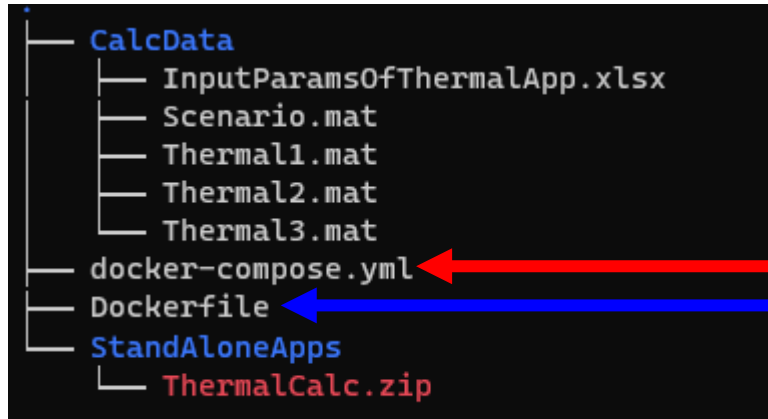
途中省略

←Excel連携の関数を実行 (MATファイル出力)

MATLAB AppDesigner GUIアプリ → Dockerコンテナ構築方法



下記のようなフォルダ構成とする



```
1 version: "3"
2 services:
3   thermalappcontainer:
4     build: .
5     container_name: thermalappcontainer
6     tty: true
7     volumes:
8     - ./CalcData:/home
```

Dockerに関しては Dockerfile/docker-composeの作法を素直に利用

←フォルダ永続化設定

```
1 # MATLAB Standalone Application Container (2020b)
2 #
3 # This docker file will configure an environment into which the Matlab compiler
4 # runtime will be installed and in which stand-alone matlab routines (such as
5 # those created with Matlab's deploytool) can be executed.
6 #
7 FROM debian:buster
8
9 ENV DEBIAN_FRONTEND noninteractive
10
11 RUN apt-get update && apt-get install -y xorg unzip x11-apps wget && \
12     apt-get clean
```

途中省略

```
24 # from Mathworks -silently install it
25 RUN mkdir /mcr-install && \
26     mkdir /opt/mcr
27
28 # copy matlab runtime zip file
29 COPY /StandAloneApps/ThermalCalc.zip /mcr-install
30
31 # silent install Standalone app with MCR
32 RUN cd /mcr-install && \
33     unzip -q ThermalCalc.zip && \
34     rm -f ThermalCalc.zip && \
35     cd /mcr-install/ThermalCalc/for_redistribution && \
36     ./MyAppInstaller_mcr.install -destinationFolder /opt/mcr -agreeToLicense yes -mode silent && \
37     cd / && \
38     rm -rf mcr-install
```

MATLABランタイムを含む Standalone Applicationのインストール

CUIからコンテナ内の計算条件Excelを指定して実行

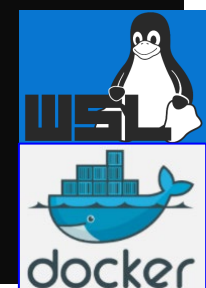
```
mizunoh@JEDIMBL2111: ~/dot
root@6a0408502954:/home# /opt/mcr/application/run_ThermalCalc.sh /opt/mcr/v99 InputParamsOfThermalApp.xlsx
Setting up environment variables
LD_LIBRARY_PATH is ./opt/mcr/v99/runtime/glnxa64:/opt/mcr/v99/bin/glnxa64:/opt/mcr/v99/sys/os/glnxa64:/opt/
inputTable =
3x45 table
  InputFileName      ResultFileName      IniTemp      Es      Ca      Ei      stph      Lx      Ly      Lz
  RAD_eff      Conductance      Qin_px      Qin_py      Qin_pz      Qin_mx      Qin_my      Qin_mz
  -----
  {'Scenario.mat'}  {'Thermal1.mat'}    -20          1420     0.5      180      5.67e-08   1      1      1
  0.8              0                  0              0              0              0              0
  {'Scenario.mat'}  {'Thermal2.mat'}    -10          1420     0.5      180      5.67e-08   1      1      1
  0.8              0                  0              0              0              0              0
  {'Scenario.mat'}  {'Thermal3.mat'}    0            1420     0.5      180      5.67e-08   1
  0.25            0.02              0.04          0.8              0              0              0

Elapsed time is 14.188707 seconds.
Thermal1.mat is saved.

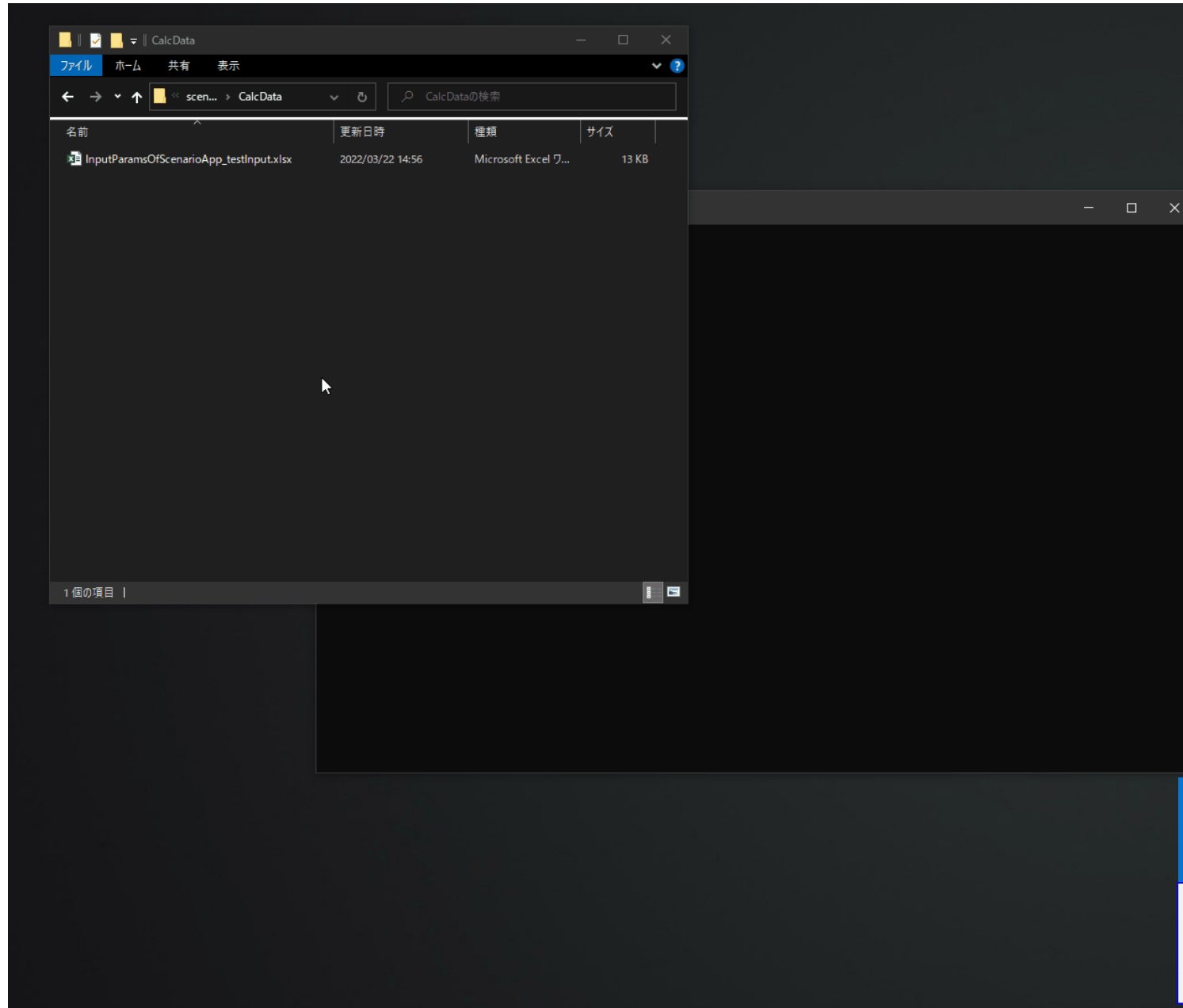
Elapsed time is 11.208282 seconds.
Thermal2.mat is saved.

Elapsed time is 10.763208 seconds.
Thermal3.mat is saved.
```

←Excelに記載の計算が連続実行できた



Dockerコンテナ実行例



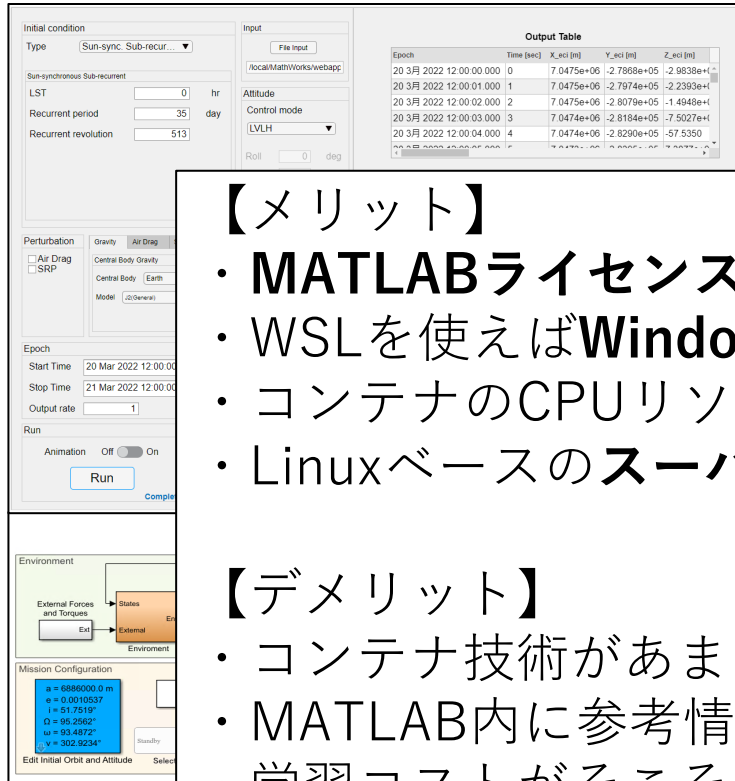
Excelに記載した計算が
連続実行できた



Dockerコンテナによるシミュレーション連続実行



MATLAB/Simulinkモデル



Simulink Coderによる
・ Standalone App作成
・ Dockerコンテナ化

MATLAB
Standalone app



【メリット】

- ・ **MATLABライセンス無し**で実行可能
- ・ WSLを使えば**Windows環境でも実行可能**（もちろんMac環境も）⇒より汎用的
- ・ コンテナのCPUリソース割当てを工夫すれば多数のコンテナ並列計算も設定可能
- ・ Linuxベースの**スーパーコンピュータ上でも実行可能**

【デメリット】

- ・ コンテナ技術があまりメジャーではない
- ・ MATLAB内に参考情報が少ない
- ・ 学習コストがそこそこ高い（Docker/Docker-compose/WSLなど）



モデル作成者



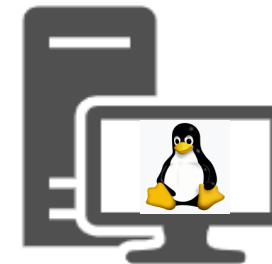
Windows MATLAB環境 + WSL2



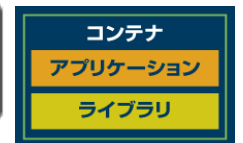
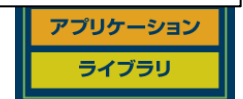
Windows + WSL2



共有WS/サーバー



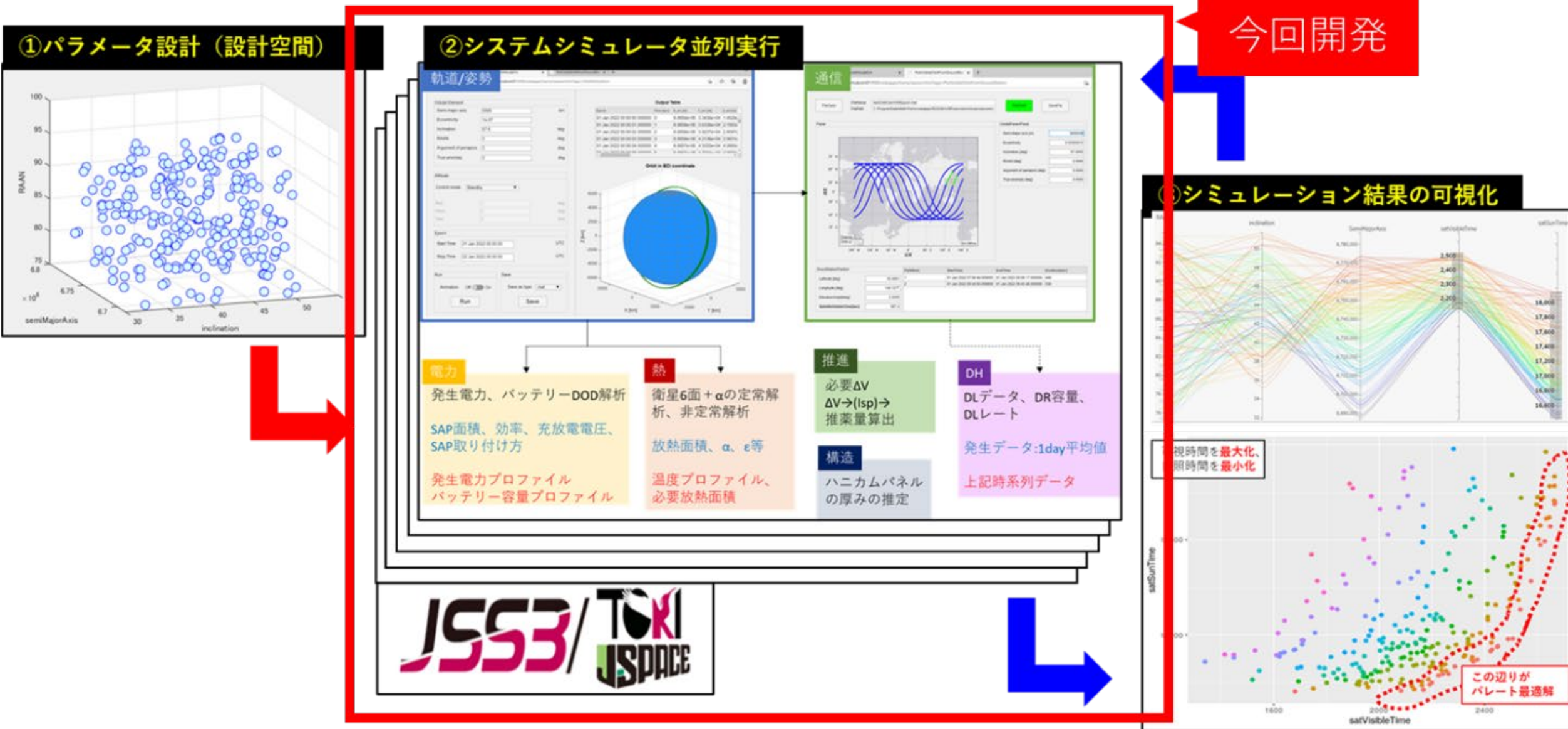
Linux



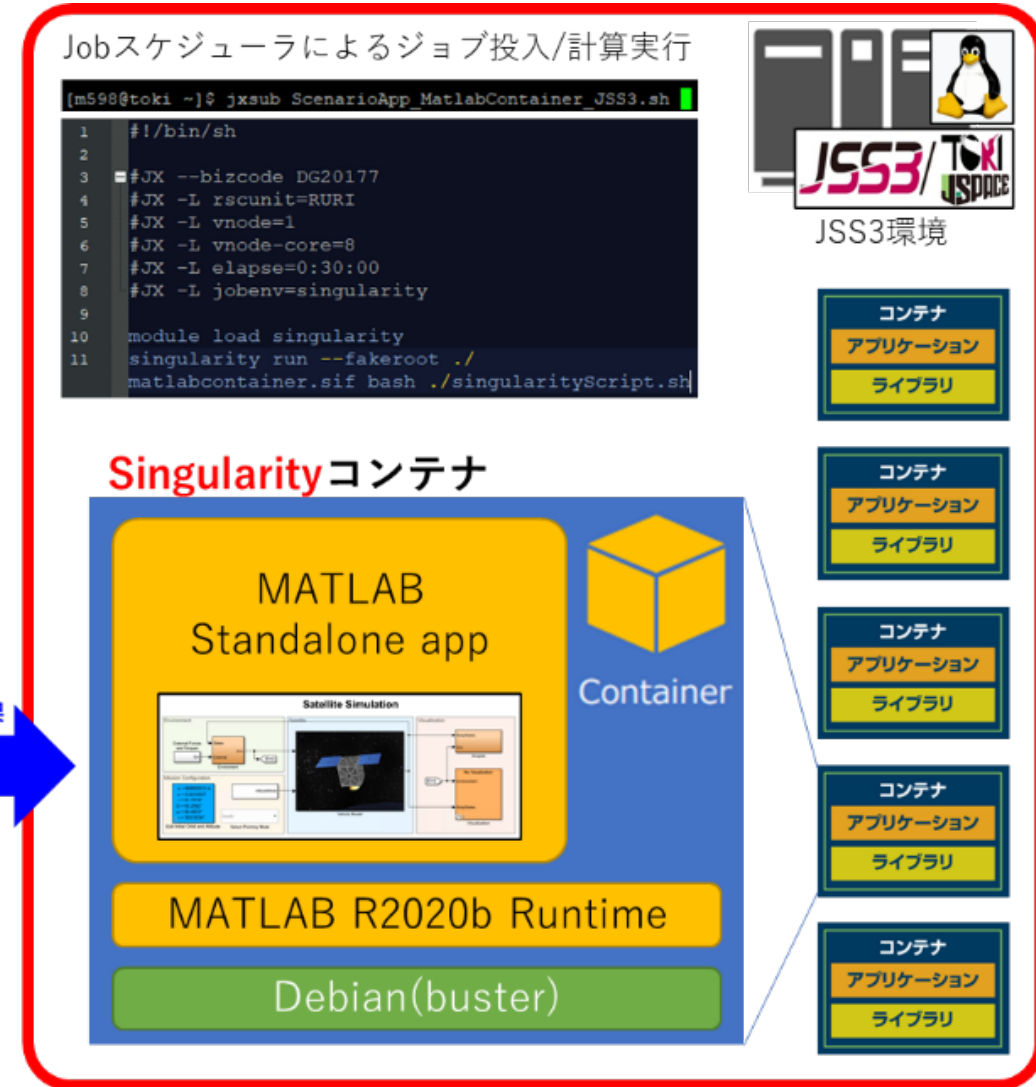
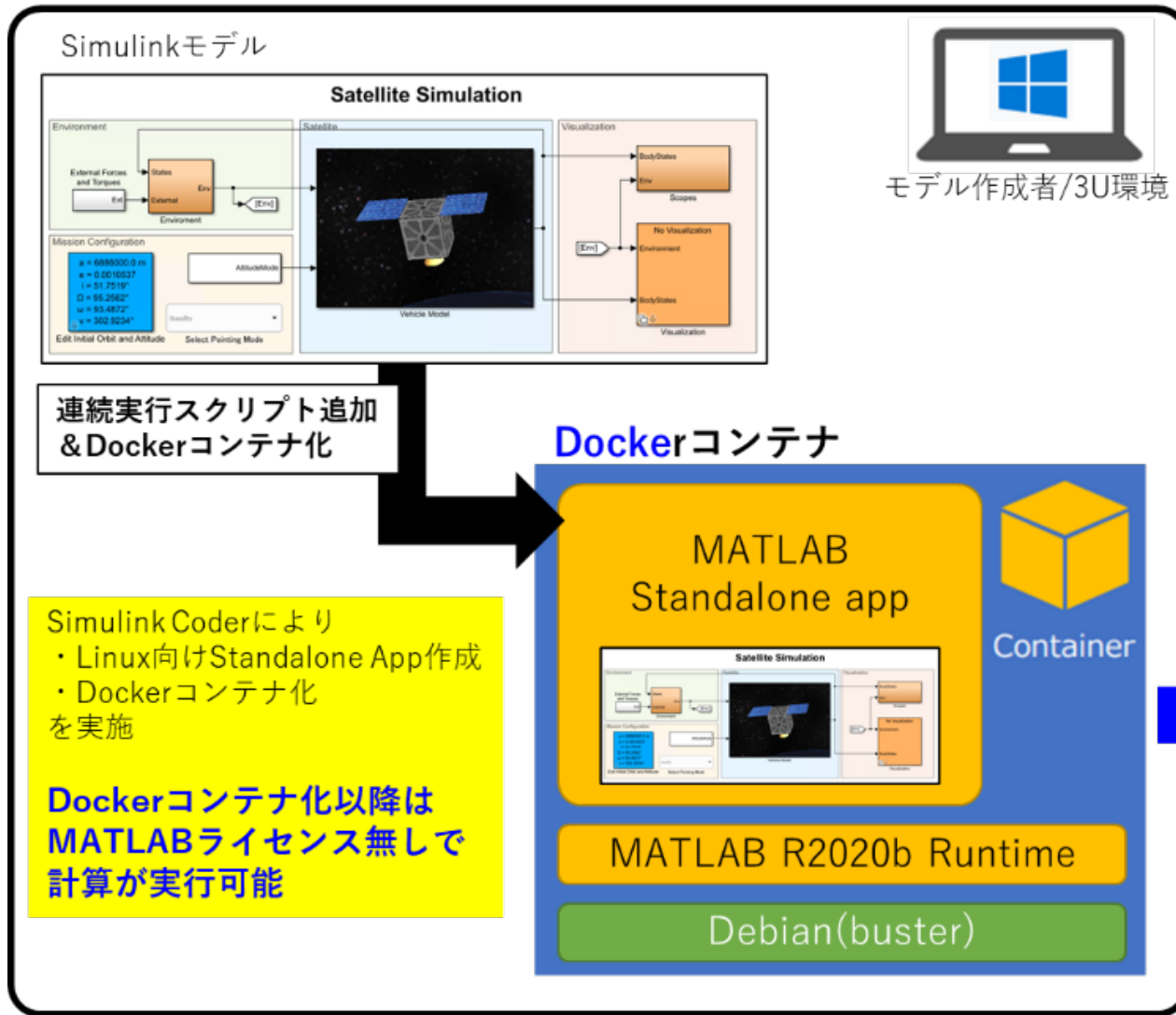
(参考)JAXAスーパーコンピュータ上でのコンテナの実行



解空間の探索や最適化を実施するためには更なる大規模実行環境が必要→JAXAスパコンの利用



(参考) JAXAスーパーコンピュータ上でのコンテナの実行



各担当のPCでの連続実行からスパコンの世界への橋渡し
スパコン向けに実施する必要がある手順はSingularityコンテナへの変換だけ

(参考) JAXAスーパーコンピュータ上でのコンテナの実行



RAAN (36水準) × Roll (12水準) × Pitch (12水準) × Yaw (12水準) の
トータル2万を超える組合せの軌道計算を実施

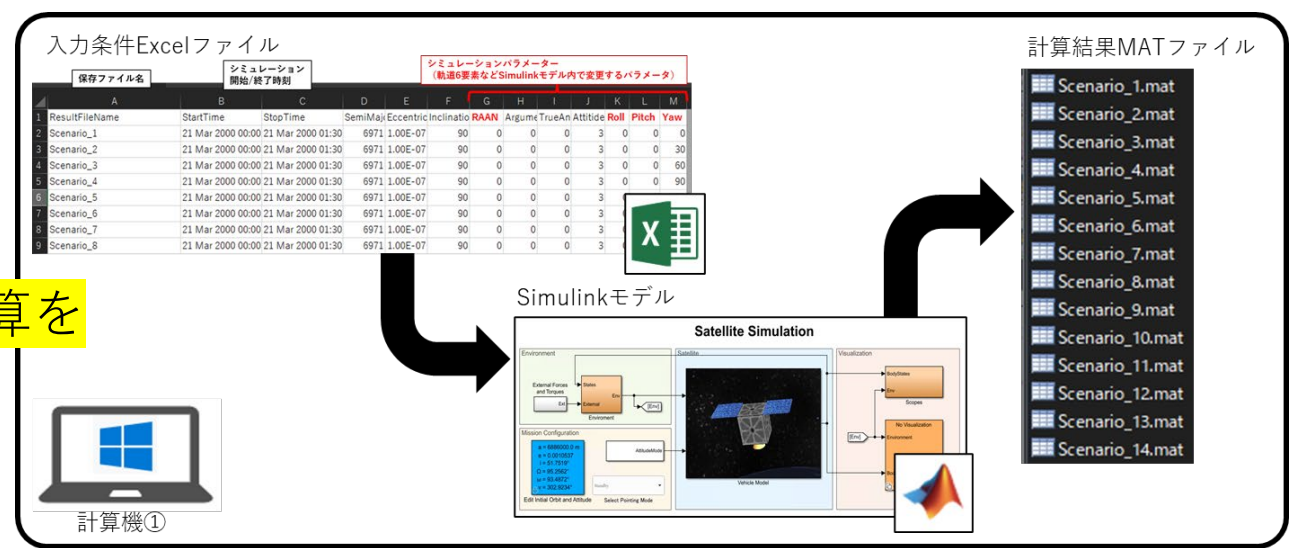
ResultFileName	StartTime	StopTime	SemiMaj	Eccentric	Inclinatio	RAAN	Argume	TrueAn	Attitude	Roll	Pitch	Yaw
Scenario_1	21 Mar 2000 00:00	21 Mar 2000 01:30	6971	1.00E-07	90	0	0	0	3	0	0	0
Scenario_2	21 Mar 2000 00:00	21 Mar 2000 01:30	6971	1.00E-07	90	0	0	0	3	0	0	30
Scenario_3	21 Mar 2000 00:00	21 Mar 2000 01:30	6971	1.00E-07	90	0	0	0	3	0	0	60
Scenario_4	21 Mar 2000 00:00	21 Mar 2000 01:30	6971	1.00E-07	90	0	0	0	3	0	0	90
Scenario_5	21 Mar 2000 00:00	21 Mar 2000 01:30	6971	1.00E-07	90	0	0	0	3	0	0	120
Scenario_6	21 Mar 2000 00:00	21 Mar 2000 01:30	6971	1.00E-07	90	0	0	0	3	0	0	150
...												
Scenario_20732	21 Mar 2000 00:00	21 Mar 2000 01:30	6971	1.00E-07	90	330	0	0	3	330	330	210
Scenario_20733	21 Mar 2000 00:00	21 Mar 2000 01:30	6971	1.00E-07	90	330	0	0	3	330	330	240
Scenario_20734	21 Mar 2000 00:00	21 Mar 2000 01:30	6971	1.00E-07	90	330	0	0	3	330	330	270
Scenario_20735	21 Mar 2000 00:00	21 Mar 2000 01:30	6971	1.00E-07	90	330	0	0	3	330	330	300
Scenario_20736	21 Mar 2000 00:00	21 Mar 2000 01:30	6971	1.00E-07	90	330	0	0	3	330	330	330

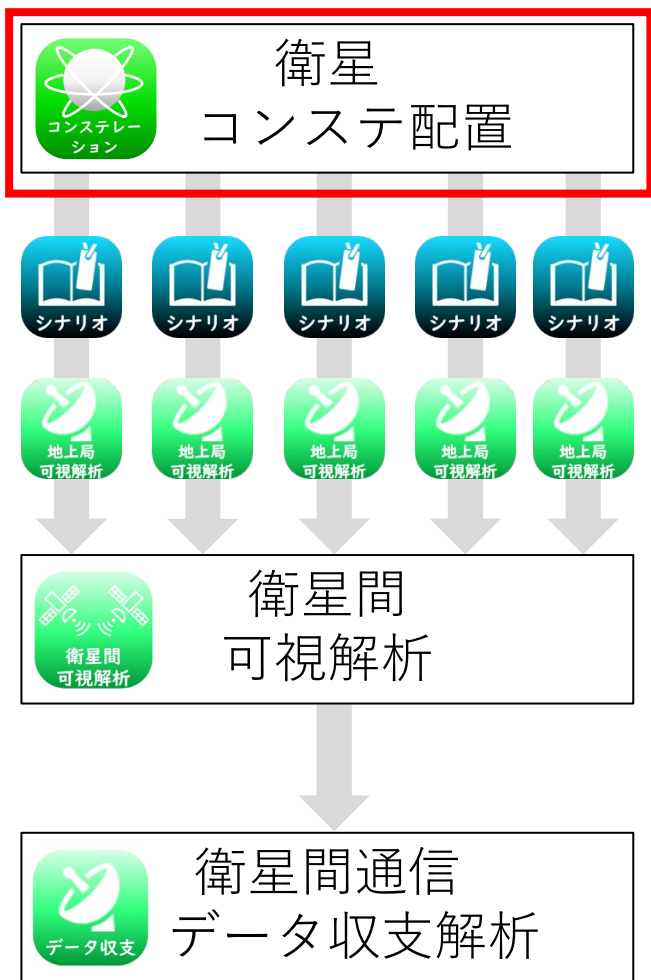
名前	サイズ	更新日時
Scenario_1.mat	586 KB	2021/08/16 12:21:28
Scenario_2.mat	584 KB	2021/08/16 12:21:28
Scenario_3.mat	585 KB	2021/08/16 12:21:28
Scenario_4.mat	582 KB	2021/08/16 12:21:28
Scenario_5.mat	583 KB	2021/08/16 12:21:28
Scenario_6.mat	586 KB	2021/08/16 12:21:28
...		
Scenario_20732.mat	592 KB	2021/08/16 13:38:15
Scenario_20733.mat	592 KB	2021/08/16 13:38:15
Scenario_20734.mat	592 KB	2021/08/16 13:38:15
Scenario_20735.mat	592 KB	2021/08/16 13:38:15
Scenario_20736.mat	592 KB	2021/08/16 13:38:15

約20分で
計算完了



コンテナ技術をうまく活用することにより
PC上での計算(シミュレーション)と全く同じ計算を
大規模にスパコン上で実行可能
→様々なスケールでMATLAB/Simulink計算が
実行できる環境になった





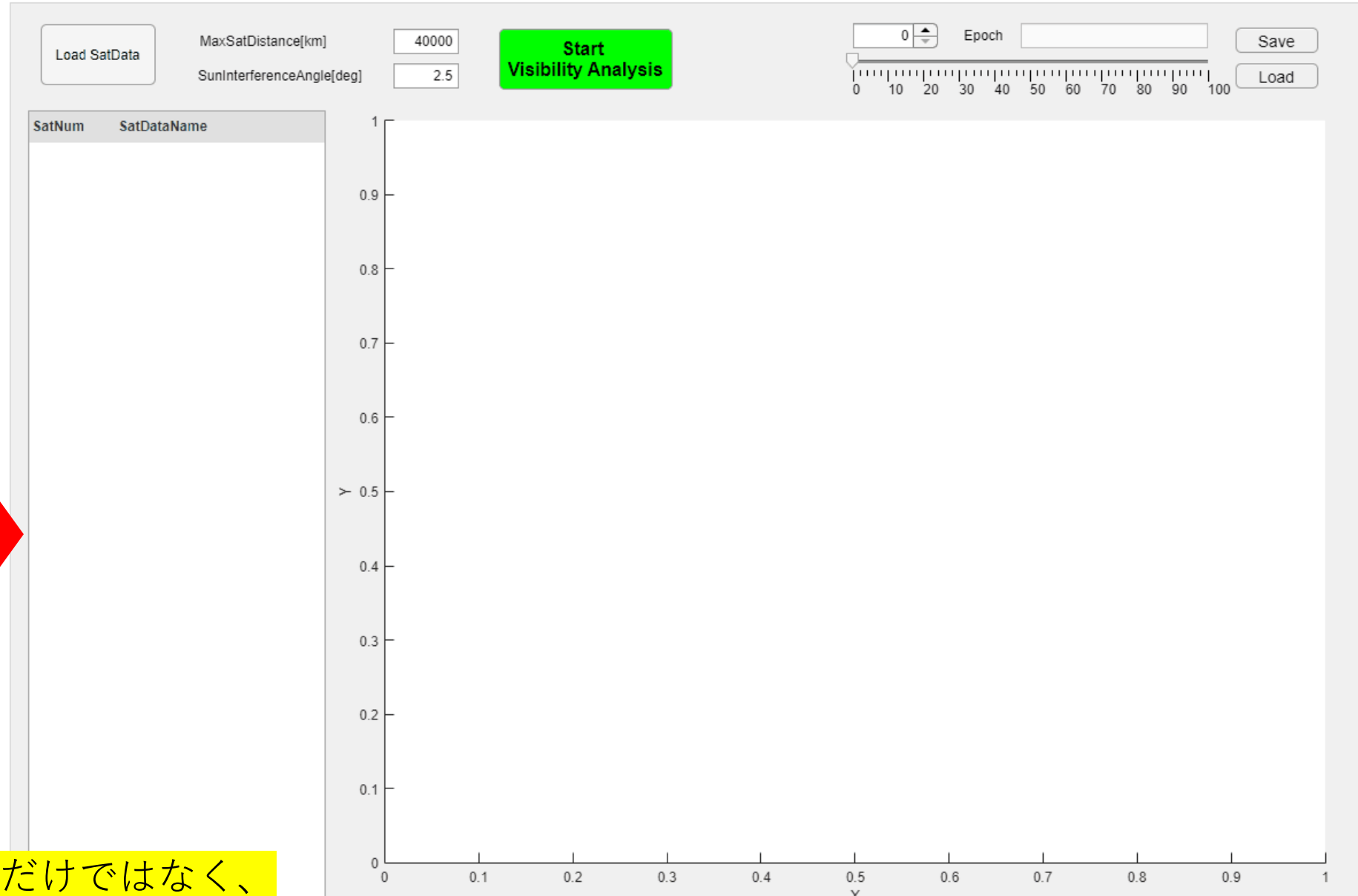
The screenshot shows the "Constellation Setting" interface. The settings are as follows:

- Type: Walker
- a: semi-major axis: 7800 km
- i: Inclination: 50 deg
- Number of Sat: 4 / plane
- p: Number of Plane: 3
- f: Phase difference: 0 / PU (with constraint $0 \leq f \leq p-1$)
- t: Number of Total Sat: 12

Buttons include "Plot Orbits" (green), "Save", and "Load". A slider is set to 0. Below the settings is a table with columns: No., a [km], e, i [deg], w [deg], RAAN [deg], nu [deg]. The table is currently empty. To the right is a plot area with X and Y axes ranging from 0 to 1.

単体の衛星設計検討ツールだけではなく、複数衛星を組み合わせた衛星コンステレーション検討ツールへ進化させていく

今後の方向性：衛星コンステレーション検討プラットフォーム

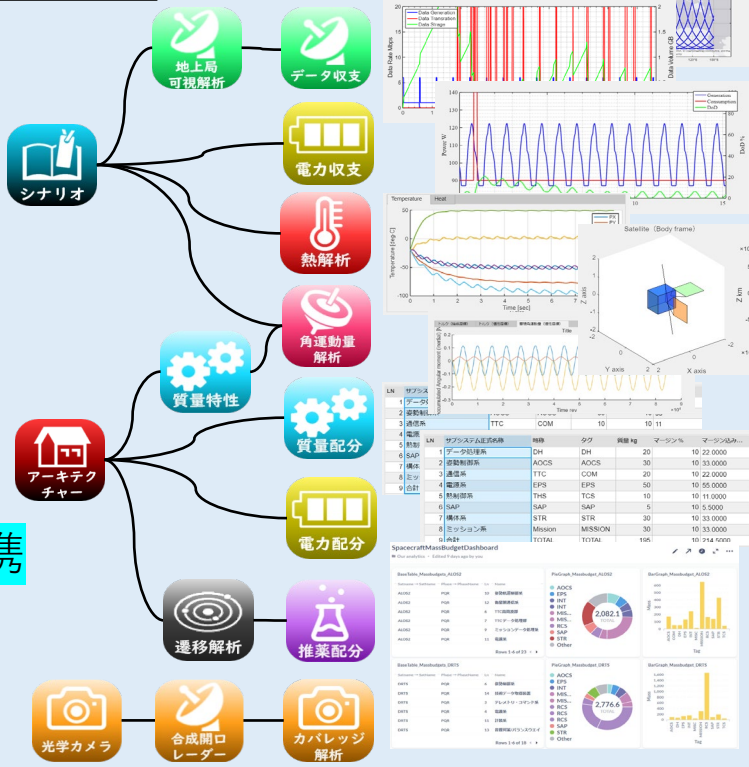


単体の衛星設計検討ツールだけではなく、複数衛星を組み合わせた衛星コンステレーション検討ツールへ進化させていく

今後の方向性：プラットフォームとして広げていく



既存Webアプリ改良



- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ

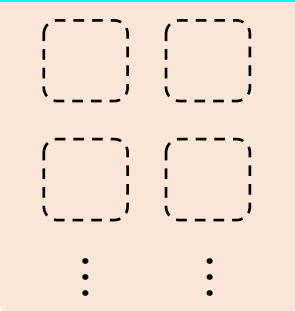


クラウド上への実装

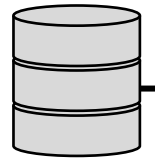


- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ
- コンテナ アプリケーション ライブラリ

新たなWebアプリ開発



データベース連携

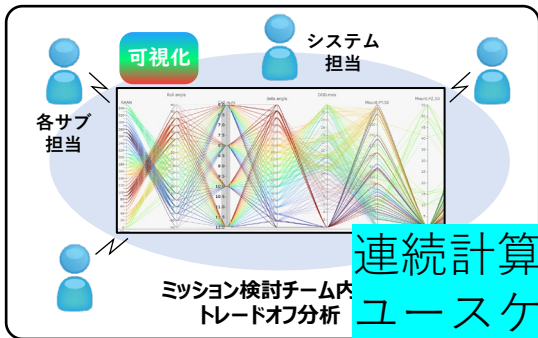
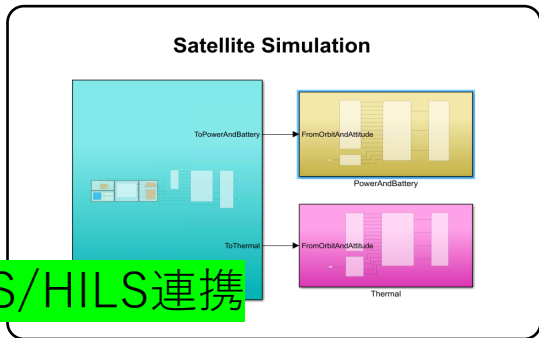


課題を
フィードバック

社外との連携を
やりやすくする

- 衛星事業者
- 大学/研究機関
- サービス事業者

SILS/HILS連携



連続計算～可視化の
ユースケース増やす

⋮

【まとめ】

- 背景（MBD/フロントローディングニーズ）とJAXA衛星開発の課題（知見の人依存）の解決を目指し開発をスタート
- **MATLAB/Webアプリ/コンテナ技術**を組み合わせ 宇宙機システム解析プラットフォーム
“Concurrent Design Platform”を構築
- Webアプリとコンテナ技術により**多人数での同時並行のアクセスと
パラスタ/最適化向けの大量計算ニーズを両立**

【やってみてわかったこと】

- 小規模で簡単なモデルやスクリプトでも**標準化して関係者間で共有するメリットは確実にある**
 - **自分たちで手を動かし内製でシステム構築**しないとクイック/アジャイルに開発やリリースができない
 - **汎用性⇔専用性のジレンマ**、Webアプリの粒度のトレードオフ
- ニーズをどこまで汎用化して取り込むか、専用ツールとの棲み分けのボーダーライン

【今後取り込むこと】

- 1つの宇宙機概念検討ツールから**衛星コンステ/サービス検討ツール**へ進化させていく
- **開発中期～後期への適用、つながり**を増やしていく
- Webアプリ/コンテナ→**マイクロサービス化**による更なるWeb化/汎用化の実現