

# HyperWorksを用いた データ処理・プロセス自動化

1. Compose製品紹介
2. データ処理事例
3. プロセス自動化事例

アルテアエンジニアリング株式会社  
2018/8/16

# モデルベース開発・システムモデリング

**COMPOSE**<sup>TM</sup>  
solidThinking<sup>®</sup>

統一ビジュアル数式処理環境

- Matlab / Octave 互換
- CAEデータへのダイレクトアクセス
- マルチ言語対応(Python)
- 製品開発環境の提供
- 豊富で高度な数学関数



# Matlab / Octave 互換のスクリプト言語

- Open Matrix Language (OML)を採用し、Matlab / Octaveのファイルが読み込めます。

The screenshot displays the solidThinking Compose 2017.1 IDE interface. The main window is titled "Untitled1.o" and contains a MATLAB/Octave script. The script includes comments in Japanese and code for clearing variables, defining data types, and creating a 2D plot. The plot shows a circular pattern with a color gradient from blue to red. The interface includes a menu bar (File, Edit, View, Tools, Debug), a toolbar with icons for File, Start, Stop, Debug, Over, In, Out, and Until, and a "ツールバー" (Toolbar) button. The Project Browser on the left shows a tree view of the project files. The Session Information window at the bottom left shows a table of variables and their types. The Command Window at the bottom center shows the execution of the script. The File Browser on the right shows a list of files in the project. The Library Browser at the bottom right shows a list of libraries. The interface is labeled with several callouts in Japanese: "プロジェクトブラウザー" (Project Browser), "セッション情報" (Session Information), "変数ブラウザー" (Variable Browser), "コマンド履歴" (Command History), "エディターウィンドウ" (Editor Window), "プロットウィンドウ" (Plot Window), and "ファイルブラウザー" (File Browser).

**プロジェクトブラウザー**

**セッション情報**

**変数ブラウザー**

**コマンド履歴**

**エディターウィンドウ**

```

1 % clear all
2
3
4 % Data Types
5 sca = 1 % is a 1x1
6 vec = [1 3 5] % is a 1x3
7 Mat = [1 2; 3 4] % is a 2x2
8 logical([-1 0 1.1]) % ans = 1 (
9 mn=ones(2,2,2) % 3 dimension
10
11 % Strings
12 % They are handled as arrays of chars
13 string1 = 'Welcome to this webinar'
14 string2 = string1(12:17)
15 string3 = ['ABC'; 'DEF']
16 reverse = 'f':-1:'a'
17
18 % Webinar
19 webinar=[1,2;3,4]; demo= 5+2i; [1
20 webinar(2,1) % will return
21
22 mystruct1 = struct [
23 book: compose
24 year: 2016
25 ]
26 mystruct1 = struct [
27 Struct array of size 1 x 2 with fields:
28 book
29 session

```

**プロットウィンドウ**

**ファイルブラウザー**

**ライブラリブラウザー**

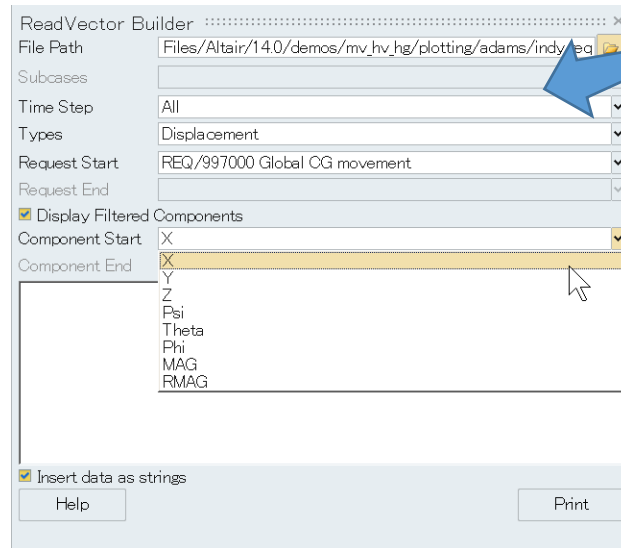
## CAEデータへのダイレクトアクセス

- HyperWorks で開発された CAE データインポート機能を搭載しており、バイナリのCAE結果が直接読み込めます。
- HyperWorksのソルバーのみでなく、LS-Dynaのd3plotやABAQUSのodbなど著名なソルバー結果ファイルに対応しています。
- 複数のデータを一括処理

**readvector / readmultivector** コマンド

- CAEデータ読み込み用GUIを提供

**readvectorbuilder** コマンド



HyperWorks  
h3d、abf  
LS-Dyna  
d3plot  
Abaqus  
odb  
...

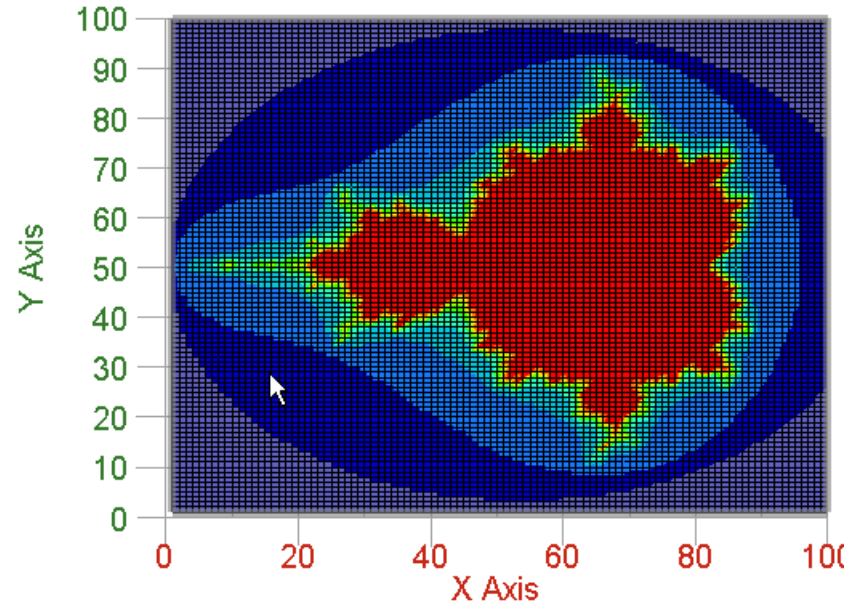
# Python サポート

- Python をサポート。OMLとPython間で関数の相互利用ができます。

```
function plotmandelbrot()  
    pyfile = [currentfile,'¥','Mandelbrot_set_demo','.py'];  
    fileflag = evalpythonfile(pyfile);  
    array = getpythonvar('array');  
    fid = contour(array);  
end
```

**'Mandelbrot\_set\_demo.py'**

```
def mandelbrot( h,w, maxit=20 ):  
    y,x = np.ogrid[ -1.4:1.4:h*1j, -2:0.8:w*1j ]  
    c = x+y*1j  
    z = c  
    divtime = maxit + np.zeros(z.shape, dtype=int)  
    for i in range(maxit):  
        z = z**2 + c  
        diverge = z*np.conj(z) > 2**2  
        div_now = diverge & (divtime==maxit)  
        divtime[div_now] = i  
        z[diverge] = 2  
    return divtime  
array = mandelbrot(100,100)
```



## ツールボックスを標準で提供

- SignalProcessing, StatisticalAnalysis, Optimizationなどのツールボックスを標準でご提供しています。
- 今後も継続的に開発、追加していきます。

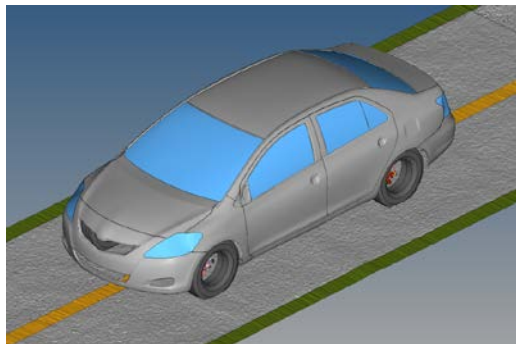
- CAE
- CAEReaders
- Calculus
- CoreMinimalInterpreter
- DataStructures
- DifferentialEquations
- ElementaryMath
- FileIO
- Gui
- LinearAlgebra
- Logical
- Optimization
- Plotting
- PolynomialMath
- PythonBridgeOmlCommands
- SignalProcessing
- StatisticalAnalysis
- StringOperations
- System
- Time
- Trigonometry

# データ処理事例

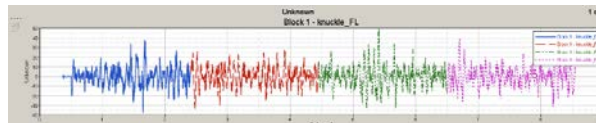
# 時刻歴波形の周波数分析 FFT

- 時刻歴の試験/CAE結果をFFTで周波数分析できます。
- CAE結果の場合は、バイナリを直接読み込んで、処理できます(CAEリーダー搭載)。

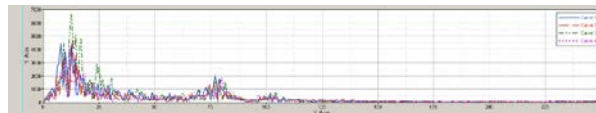
悪路走行時の各点の  
加速度を計測/シミュレーション



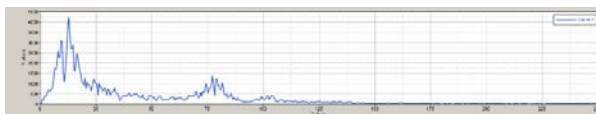
時刻歴波形を分割



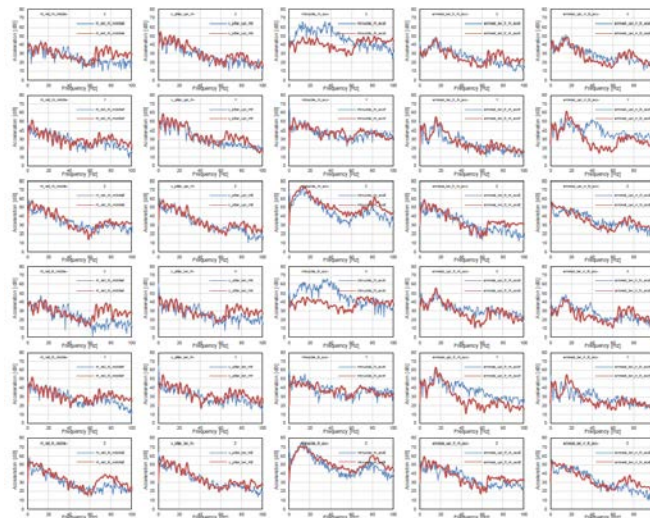
各領域にウィンドウ処理、FFT



各FFTを平均化



全計測点に対して一括処理、試験とCAEを比較



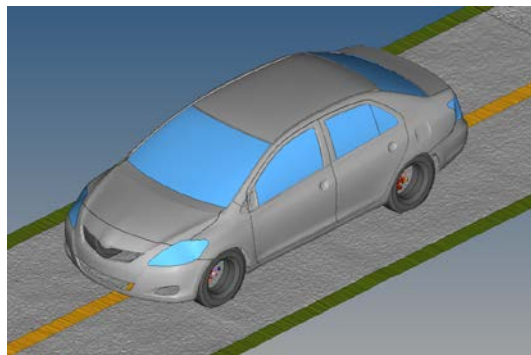
悪路走行時の加速度評価



# ソルバー間のデータの受け渡し（結果ファイルからの入力データ作成）

- 機構解析の時刻歴結果を周波数分析し、Bulk Formatで書き出すことで、振動解析の入力ファイルとして利用できます。複数の解析をシームレスに繋がります。

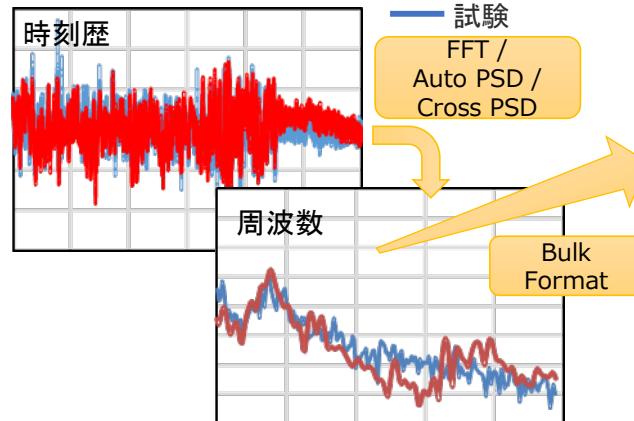
## MotionSolve : 機構解析



タイヤの回転、サスの大ストローク等非線形性を含む現象を機構解析を用いて、過渡で解析します。

## 悪路走行シミュレーション

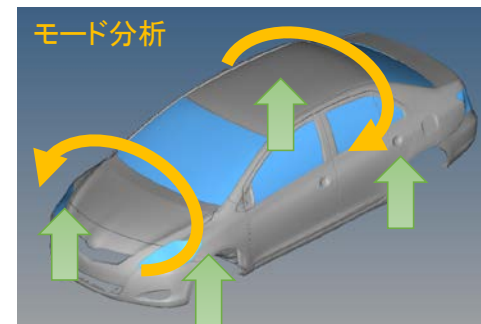
## Compose



解析した時刻歴の荷重履歴を周波数分析し、Bulk Formatで書き出します。

## 周波数分析

## Optistruct : 振動解析



周波数分析した荷重を入力とし、周波数応答解析により、振動モードを確認します。

## 振動モード分析

# プロセス自動事例

# モデル修正、ソルバー実行、ポスト処理の自動化

- Composeから各種プリ、ソルバーが実行可能です。
- 実稼動解析と構造最適化など異なるソルバーを組み合わせた新しいソリューションが実現可能です。

