ANSA and μETA as a CAE Software Development Platform

Michael Giannakidis, Yianni Kolokythas

BETA CAE Systems SA, Thessaloniki, Greece

Overview

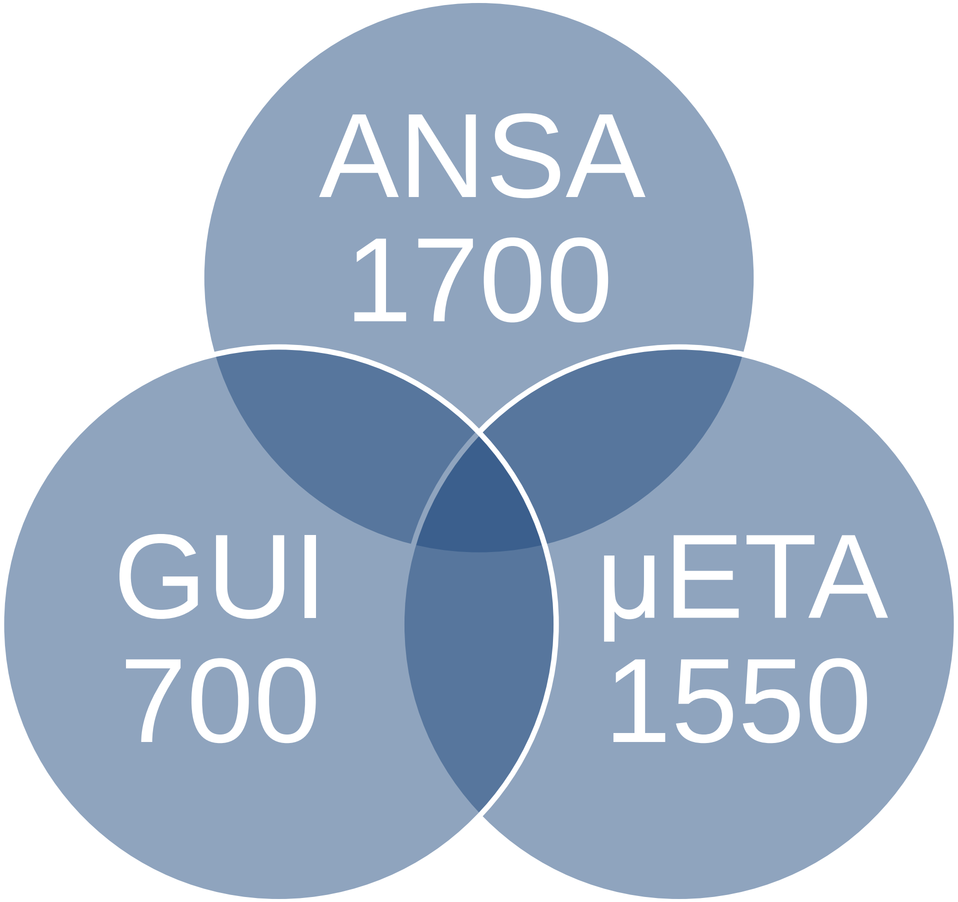
- What have we have done so far
- Current state
- Future direction

Scripting – Application Programming Interface

- A user can programmatically interact with ANSA and μETA.
- Users can create their own custom functionality
- Users can extend existing functionality to fit their simulation processes.

What have we have given you so far

A big library of APIs that give access to the core of the programs



ANSA
1700

GUI
700

μETA
1550

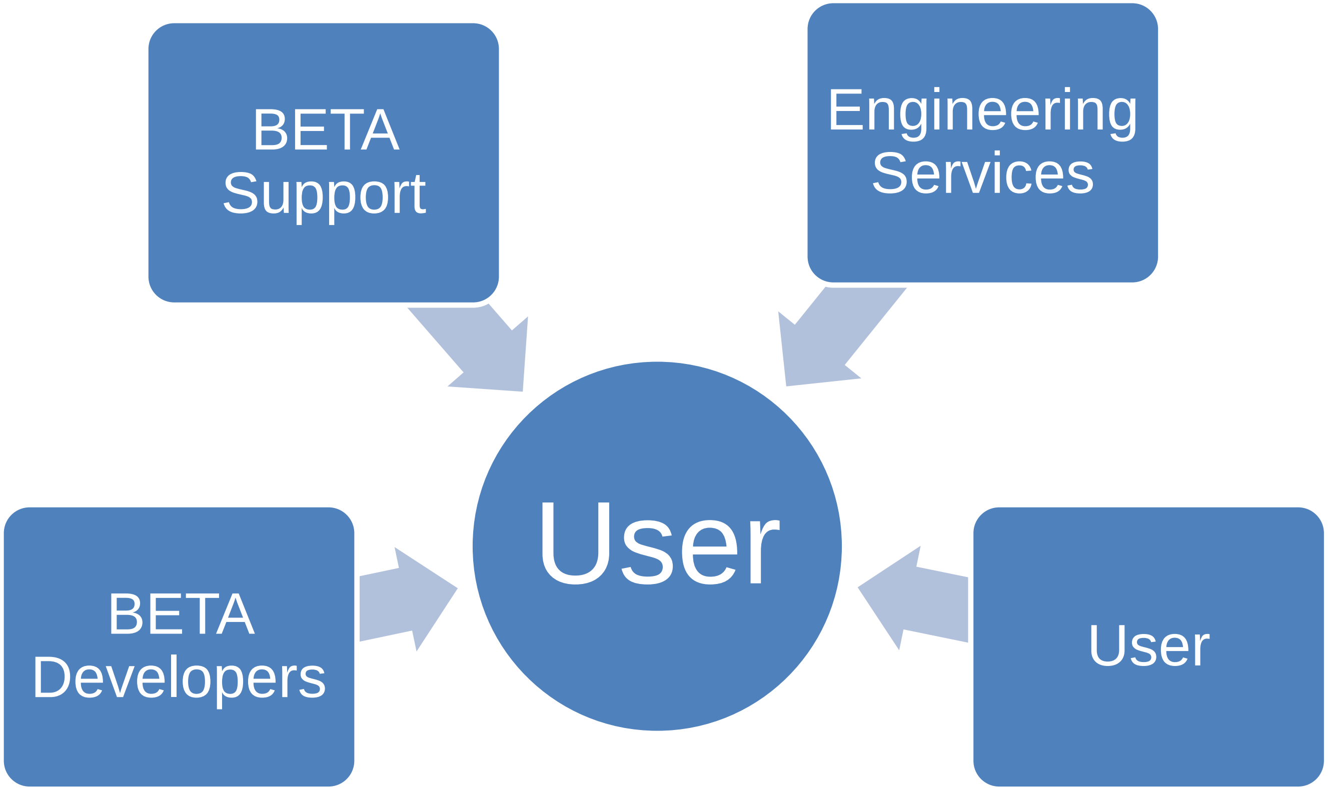
What have we have given you so far

The BETA Script programming language

- A custom made language developed and maintained by BETA
- A combination of many features from languages inspired by C
- It worked well all these years. It helped in the wide adoption of scripting in the ANSA and μETA community

User Centered Development

Creative community


BETA Support

Engineering Services

User

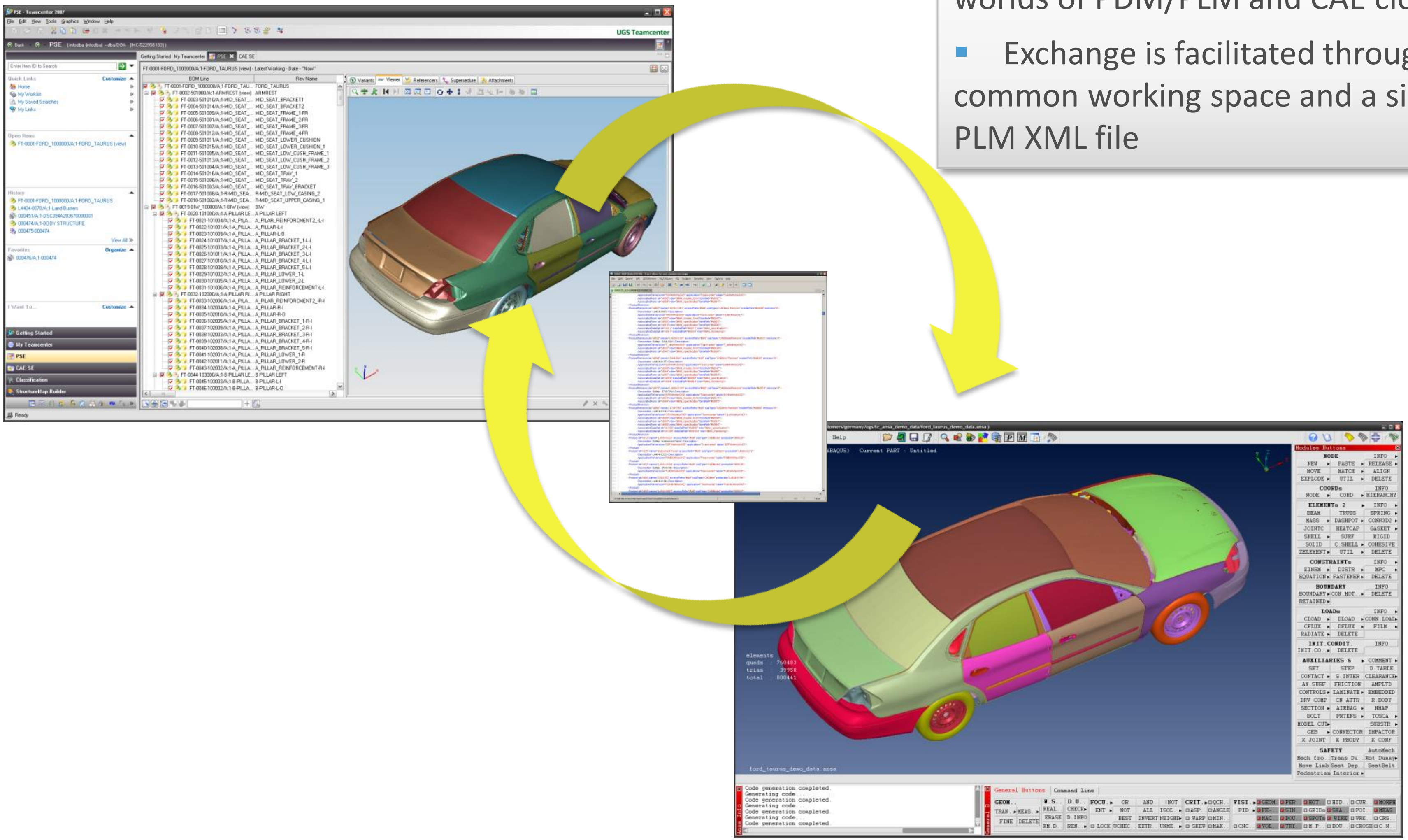
BETA Developers

User

Applications of ANSA Scripting

ANSA Teamcenter Interface Tool

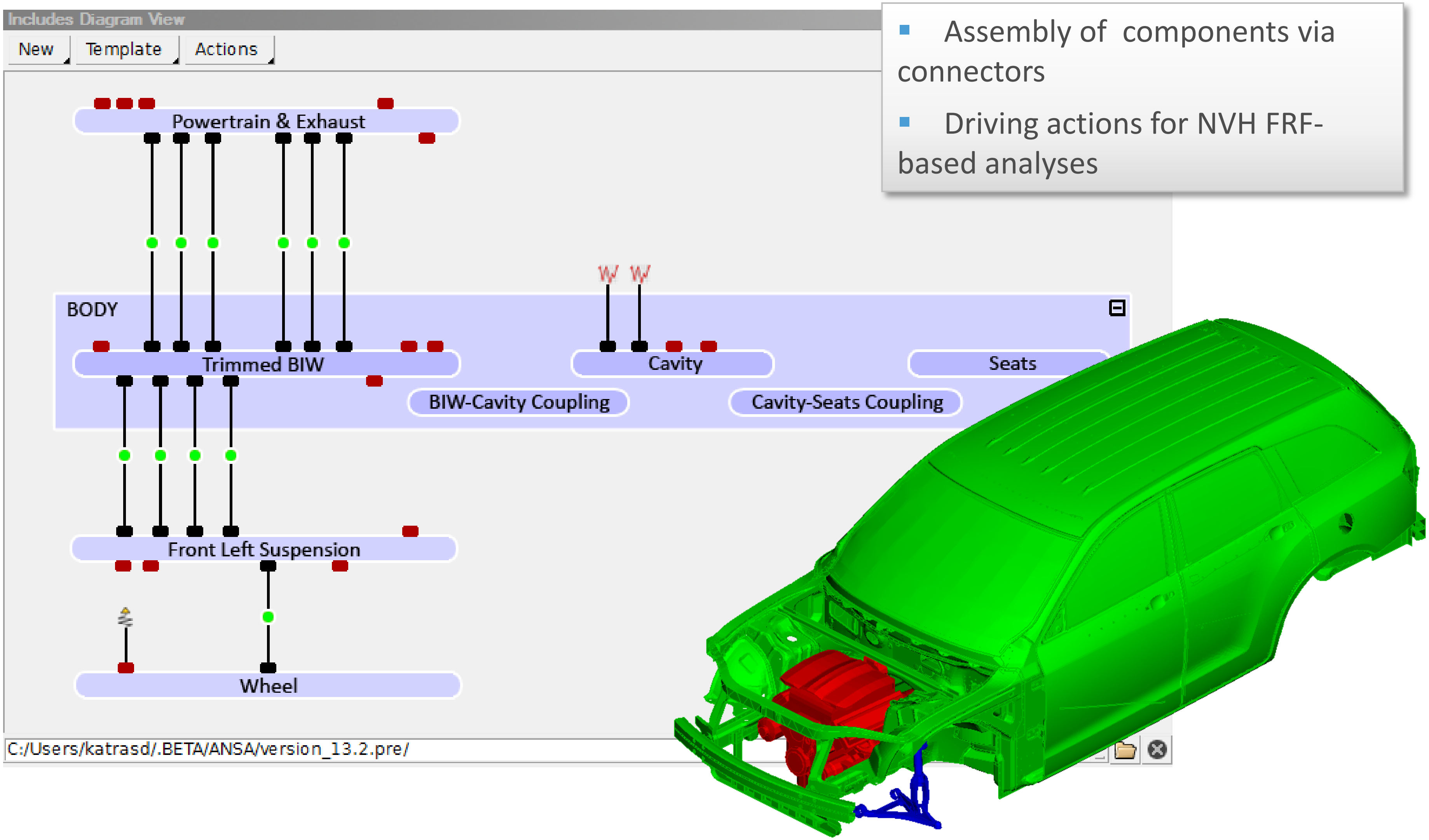
- TC – ANSA interaction brings the worlds of PDM/PLM and CAE closer
- Exchange is facilitated through a common working space and a single PLM XML file



```
Code generation completed.
Generating code ...
Code generation completed.
Generating code ...
Code generation completed.
Generating code ...
Code generation completed.
```

NVH Console

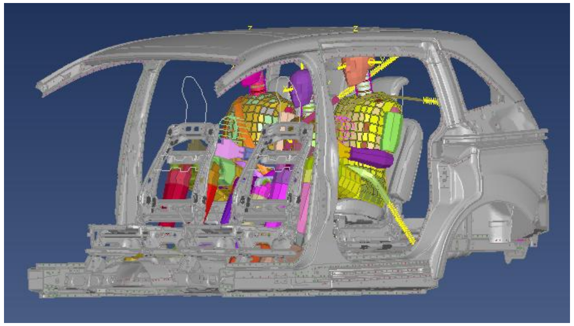
- Assembly of components via connectors
- Driving actions for NVH FRF-based analyses



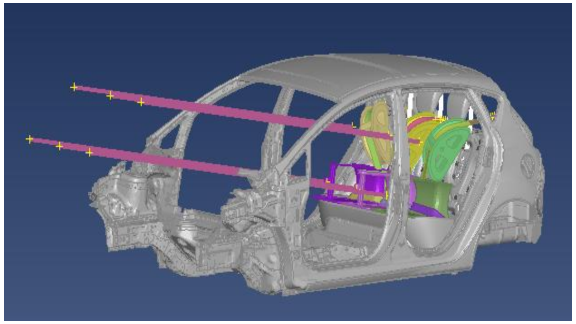
C:/Users/katrasd/.BETA/ANSA/version_13.2.pre/

Process Automation – Task Manager

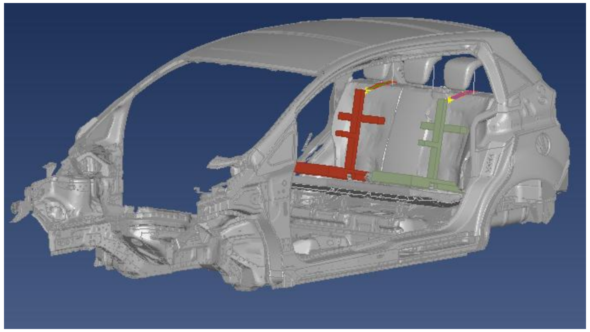
Restrain Integrity (Sled Test)



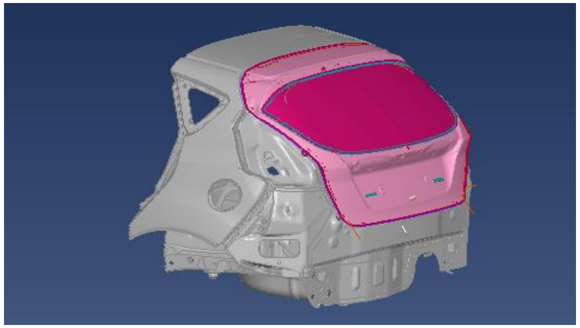
Seatbelt Static Strength

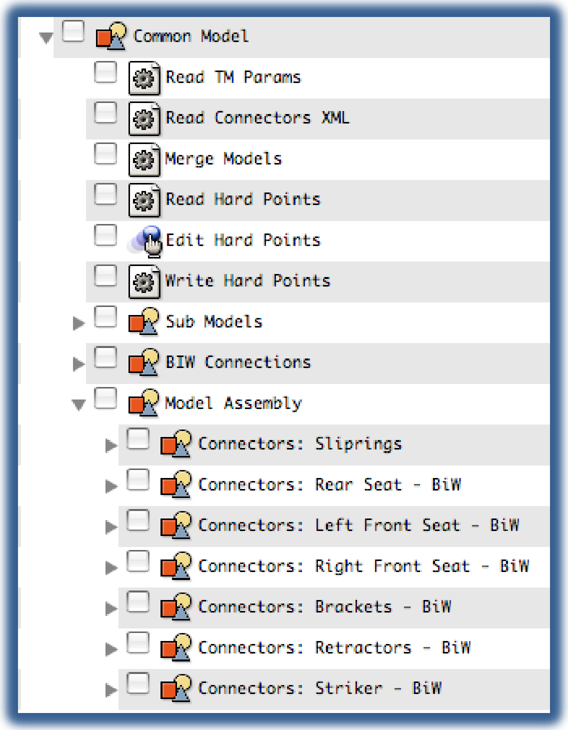


ISOFIX



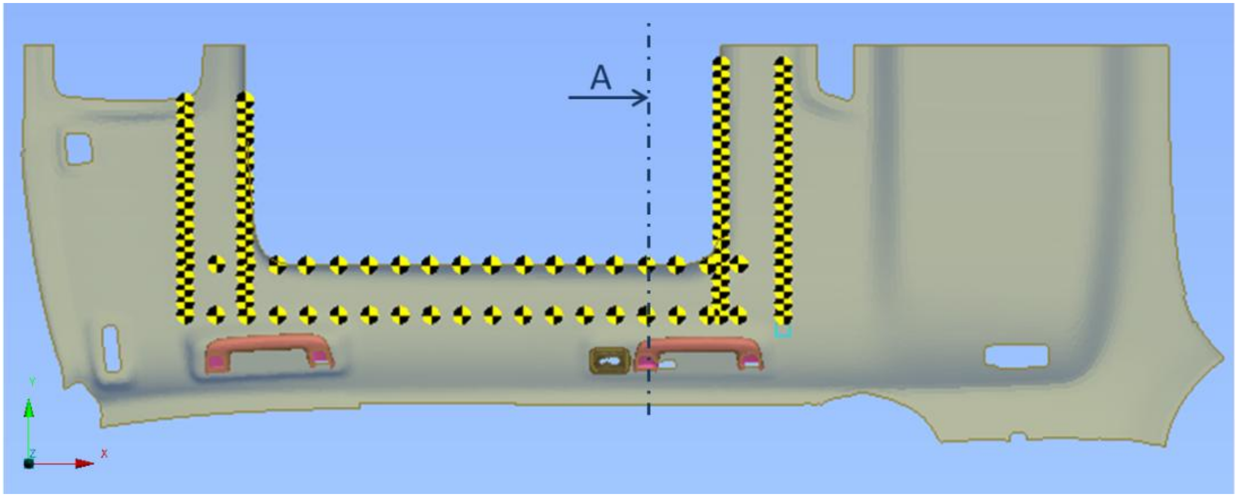
Liftgate Slam Analysis

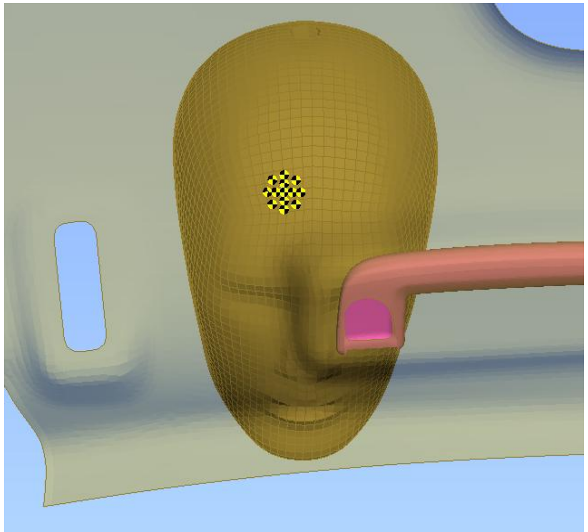




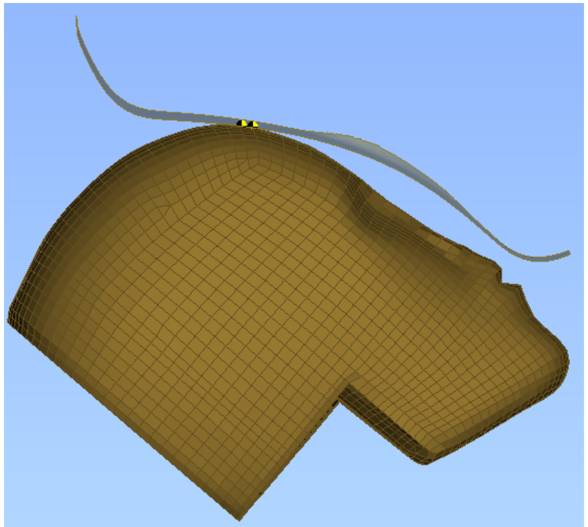
- ▼ □ Common Model
  - □ Read TM Params
  - □ Read Connectors XML
  - □ Merge Models
  - □ Read Hard Points
  - □ Edit Hard Points
  - □ Write Hard Points
  - ▶ □ Sub Models
  - ▶ □ BIW Connections
  - ▼ □ Model Assembly
    - ▶ □ Connectors: Sliprings
    - ▶ □ Connectors: Rear Seat - BiW
    - ▶ □ Connectors: Left Front Seat - BiW
    - ▶ □ Connectors: Right Front Seat - BiW
    - ▶ □ Connectors: Brackets - BiW
    - ▶ □ Connectors: Retractors - BiW
    - ▶ □ Connectors: Striker - BiW

FMVSS201U Automated Process for Volvo Cars

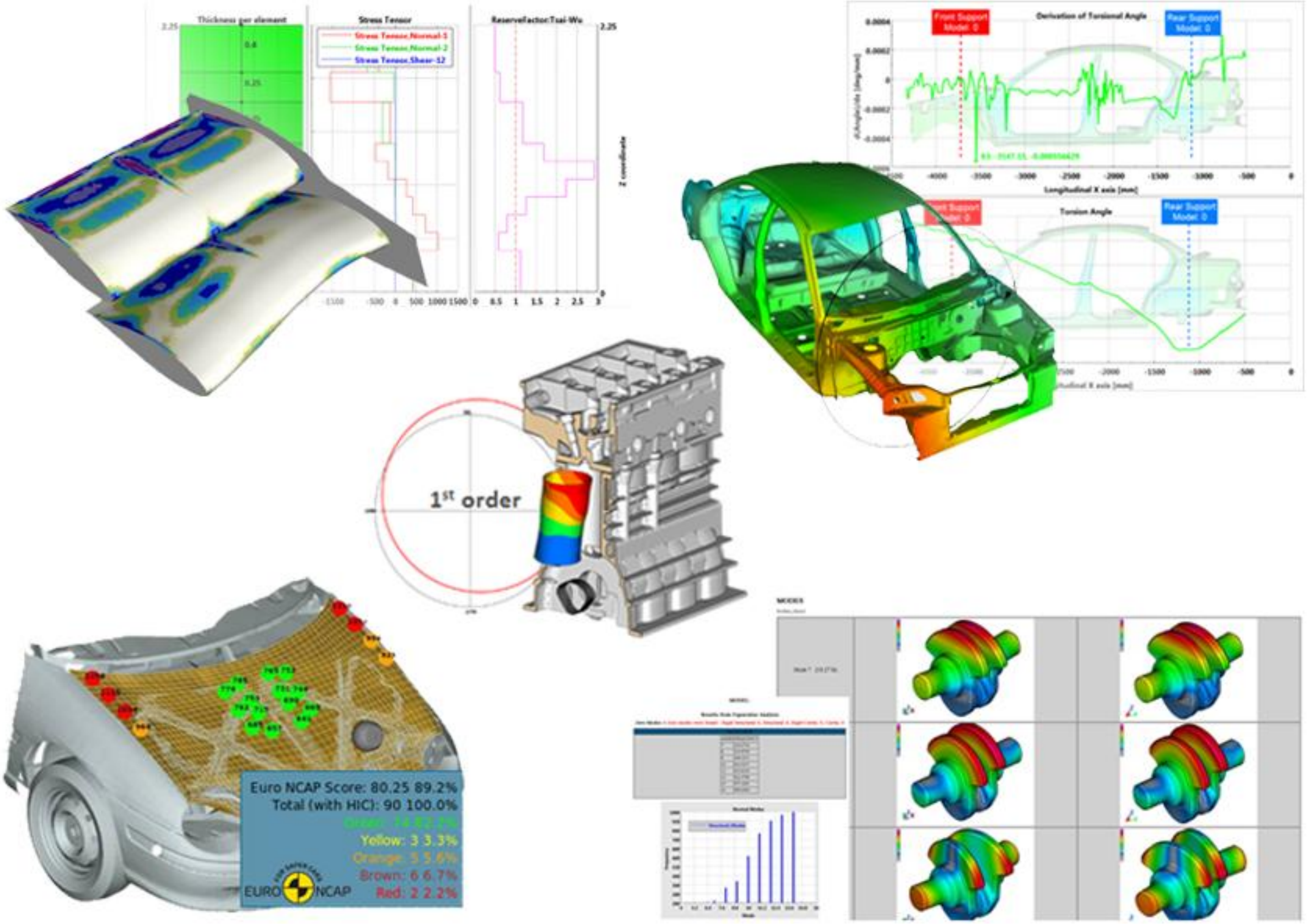




- Automatic Targeting
- Positioning achieving maximum vertical angle and Conservative positioning
- Massive Positioning and Output keyword files
- Robustness studies for the worst case scenario



μETA Toolbars



Beyond BETA Script

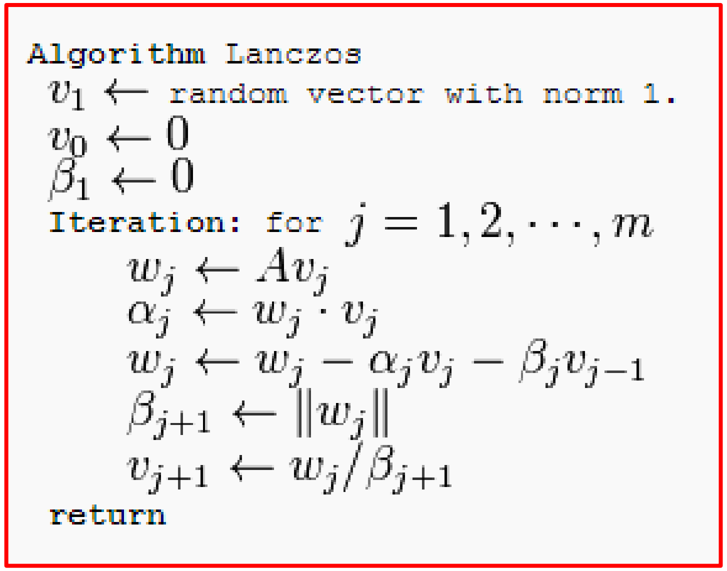
Why change?

- BETA Script has reached its limitations
- A modern language was needed
- More functionality. 4000 functions are not enough.
- Access more technologies
- A more popular language worth investing in for the users.

Why Change?

Compute the eigenvalues and right eigenvectors of a square array.

$$\mathbf{A} = \begin{bmatrix} 9 & 13 & 5 & 2 \\ 1 & 11 & 7 & 6 \\ 3 & 7 & 4 & 1 \\ 6 & 0 & 7 & 10 \end{bmatrix}.$$



```
Algorithm Lanczos
 $v_1 \leftarrow$ random vector with norm 1.
 $v_0 \leftarrow 0$
 $\beta_1 \leftarrow 0$
 Iteration: for $j = 1, 2, \cdots, m$
     $w_j \leftarrow Av_j$
     $\alpha_j \leftarrow w_j \cdot v_j$
     $w_j \leftarrow w_j - \alpha_j v_j - \beta_j v_{j-1}$
     $\beta_{j+1} \leftarrow \|w_j\|$
     $v_{j+1} \leftarrow w_j / \beta_{j+1}$
 return
```

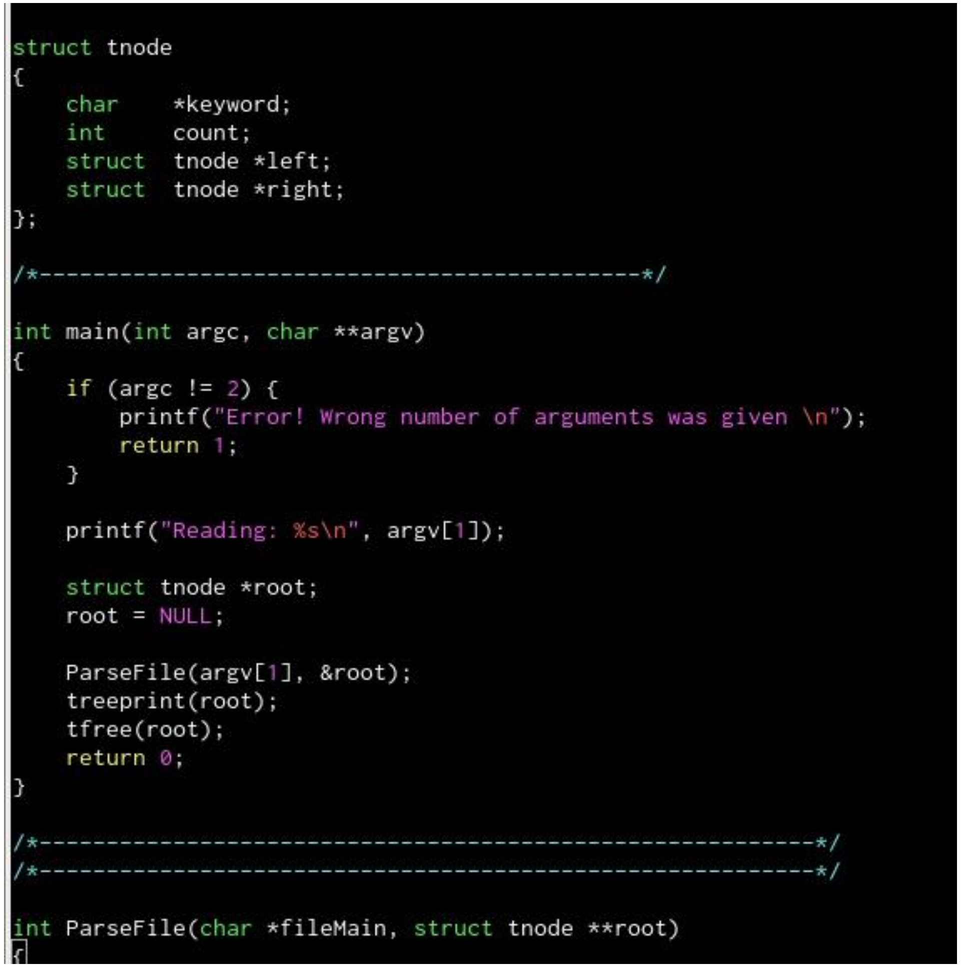
Why Change?

Find the points were two given functions intersect

$$y_1 = x_1^2$$
$$y_2 = \ x_2 + 1$$

Why Change?

Run External C/C++ code



```
struct tnode
{
    char    *keyword;
    int     count;
    struct  tnode *left;
    struct  tnode *right;
};

/*---------------------------------------------*/

int main(int argc, char **argv)
{
    if (argc != 2) {
        printf("Error! Wrong number of arguments was given \n");
        return 1;
    }

    printf("Reading: %s\n", argv[1]);

    struct tnode *root;
    root = NULL;

    ParseFile(argv[1], &root);
    treeprint(root);
    tfree(root);
    return 0;
}

/*----------------------------------------------------------*/
/*----------------------------------------------------------*/

int ParseFile(char *fileMain, struct tnode **root)
{
```

The Answer is

python

➕

Libraries

Why Python?

- High level object oriented programming language
- Clear and expressive syntax
- Large Standard Library
- Large Selection of third party scientific and mathematical libraries
- Is emerging as the language of choice for computational sciences and engineering

Immediate benefits

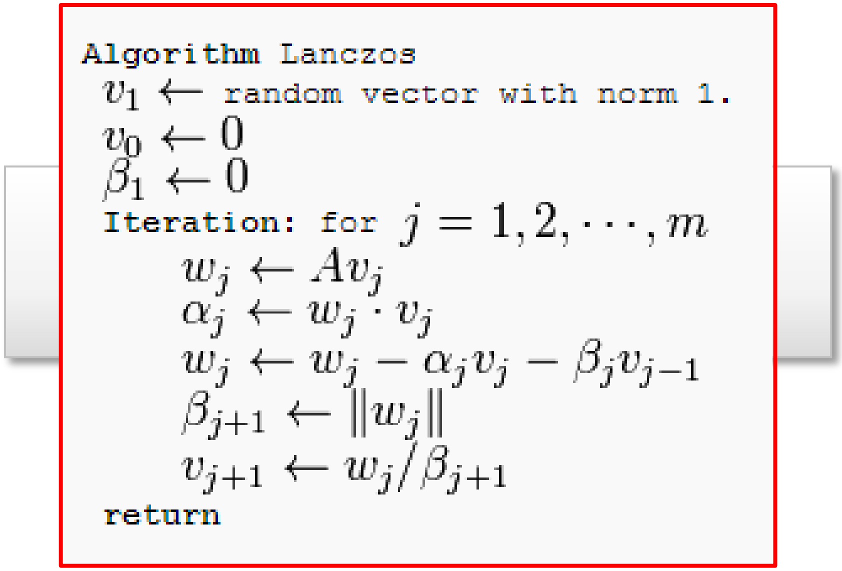
- Access to a huge number of standard python modules
- Access to 3rd party open source libraries (NumPy, SciPy, MatPlot lib etc)
- Ability to run external code written in C / C++
- Abundance of helpful resources on the internet.
- New engineering graduates may already know Python

Libraries

Libraries – NumPy

Compute the eigenvalues and right eigenvectors of a square array.

$$\mathbf{A} = \begin{bmatrix} 9 & 13 & 5 & 2 \\ 1 & 11 & 7 & 6 \\ 3 & 7 & 4 & 1 \\ 6 & 0 & 7 & 10 \end{bmatrix}.$$



```
Algorithm Lanczos
 $v_1 \leftarrow$ random vector with norm 1.
 $v_0 \leftarrow 0$
 $\beta_1 \leftarrow 0$
 Iteration: for $j = 1, 2, \cdots, m$
     $w_j \leftarrow Av_j$
     $\alpha_j \leftarrow w_j \cdot v_j$
     $w_j \leftarrow w_j - \alpha_j v_j - \beta_j v_{j-1}$
     $\beta_{j+1} \leftarrow \|w_j\|$
     $v_{j+1} \leftarrow w_j / \beta_{j+1}$
 return
```

Libraries – SciPy

Find the points were two given functions intersect

$$y_1 = x^2$$

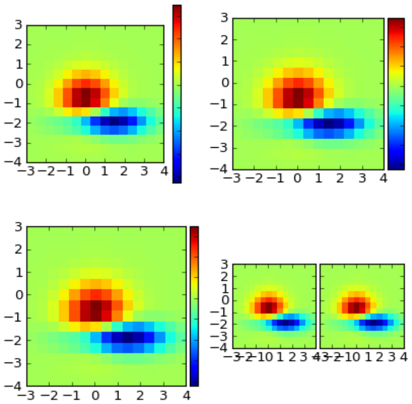
$$y_2 = x_2 + 1$$

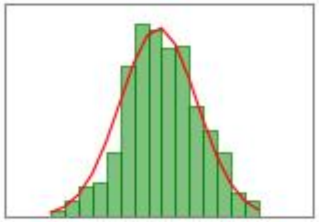
```
from scipy.optimize import fsolve
def f(xy):
     x, y = xy
     z = np.array([y-x**2, y-x-1.0])
     return z
fsolve(f, [1.0, 2.0])
```

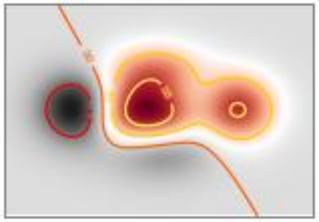
Libraries – NumPy + SciPy

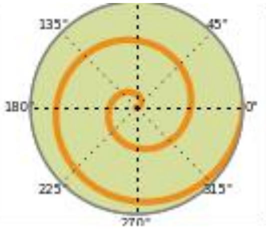
- Array Handling
- Linear Algebra
- Interpolation
- Differential Equations
- Optimization
- Signal Processing
- Statistics
- Fitting Techniques
- and many more…

Libraries – Matplotlib



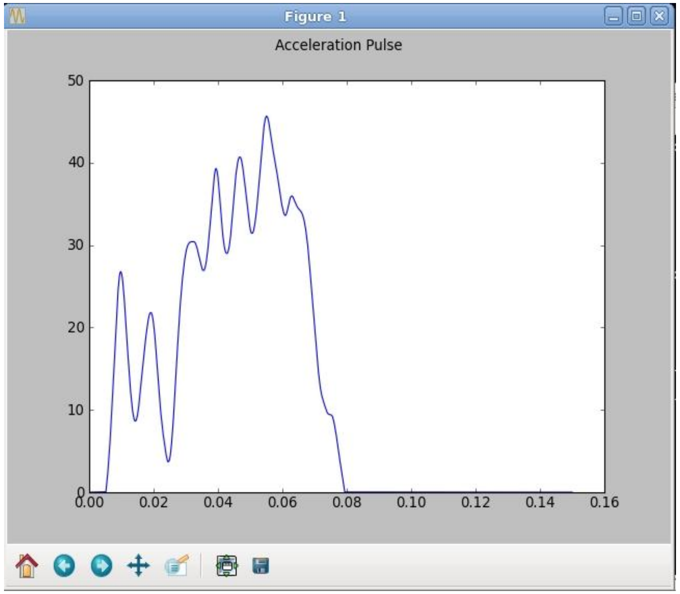




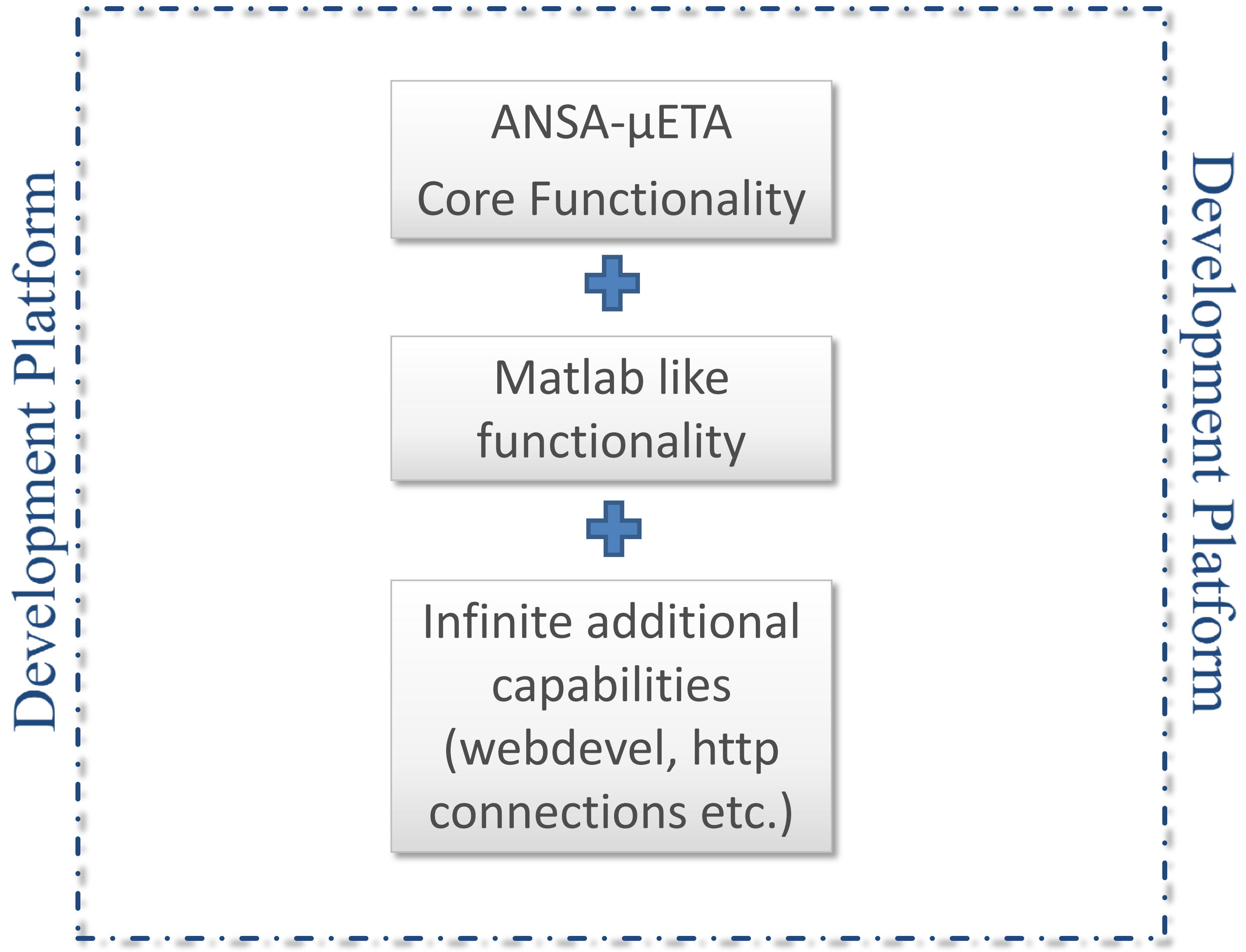


```
import matplotlib.pyplot as plt

def my_plot():
        file=open('/home/yianni/sled_pulse.txt')
        x_vals = list()
        y_vals = list()
        for line in file:
                ret = line.split(',')
                x_vals.append(float(ret[0]))
        y_vals.append(float(ret[1]))
        plt.suptitle('Acceleration Pulse')
        ax = plt.subplot('111')
        ax.plot(x_vals, y_vals)
        plt.show()
```



Platform Overview



Development Platform

ANSA-μETA

Core Functionality

Matlab like functionality

Infinite additional capabilities (webdevel, http connections etc.)

Development Platform

“Python is a super glue for modern scientific workflow”

by Joshua Bloom, PhD

University of California, Berkeley

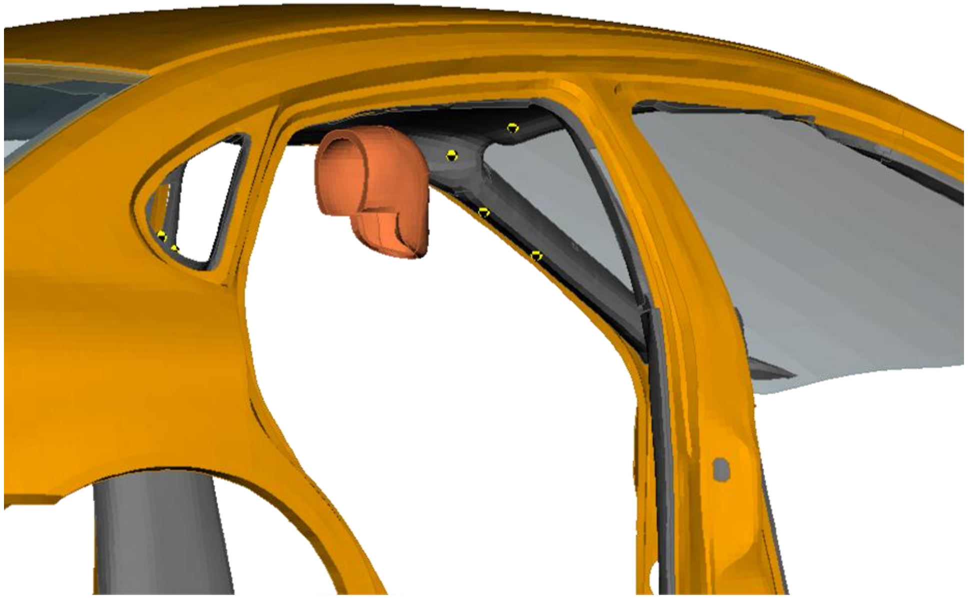
Astronomy Department

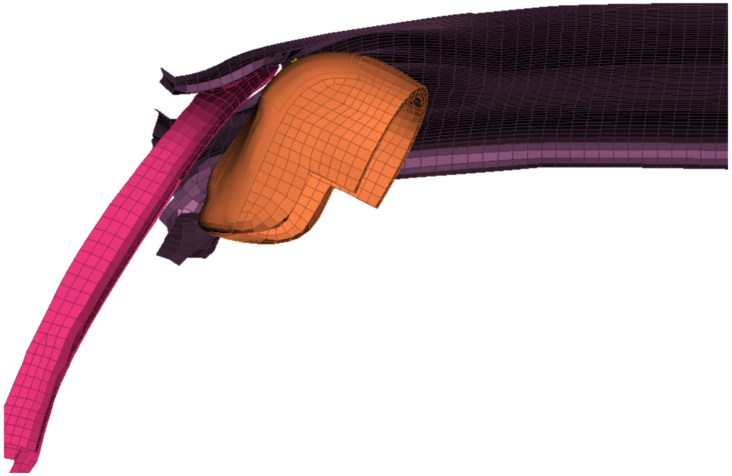
Google
facebook
CERN
YouTube
Dropbox
3DS SIMULIA
bitly
INDUSTRIAL LIGHT & MAGIC
A LUCASFILM COMPANY
NASA
Autodesk
PIXAR
ANIMATION STUDIOS
SPACEX
Space Exploration Technologies
BETA CAE Systems SA
SPACE TELESCOPE SCIENCE INSTITUTE

Examples of the New Capabilities

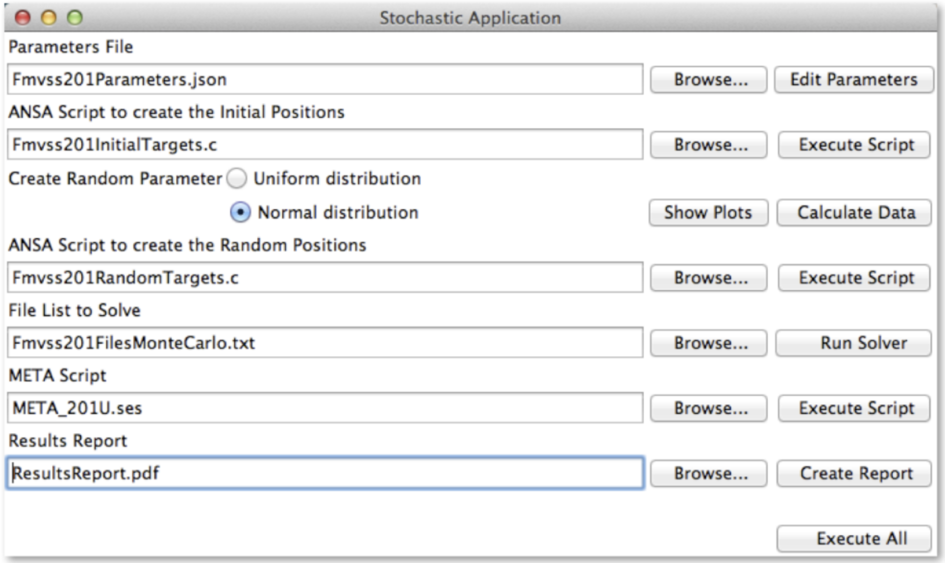
Robustness Analysis Tool

FMVSS201U Analysis is highly dependent on positioning parameters





Robustness Analysis Tool



Parameters File: Fmvss201Parameters.json

ANSA Script to create the Initial Positions: Fmvss201InitialTargets.c

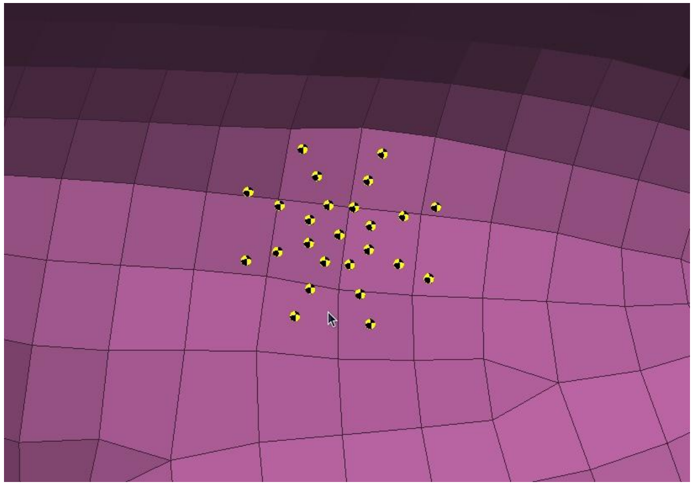
Create Random Parameter ○ Uniform distribution ● Normal distribution

ANSA Script to create the Random Positions: Fmvss201RandomTargets.c

File List to Solve: Fmvss201FilesMonteCarlo.txt

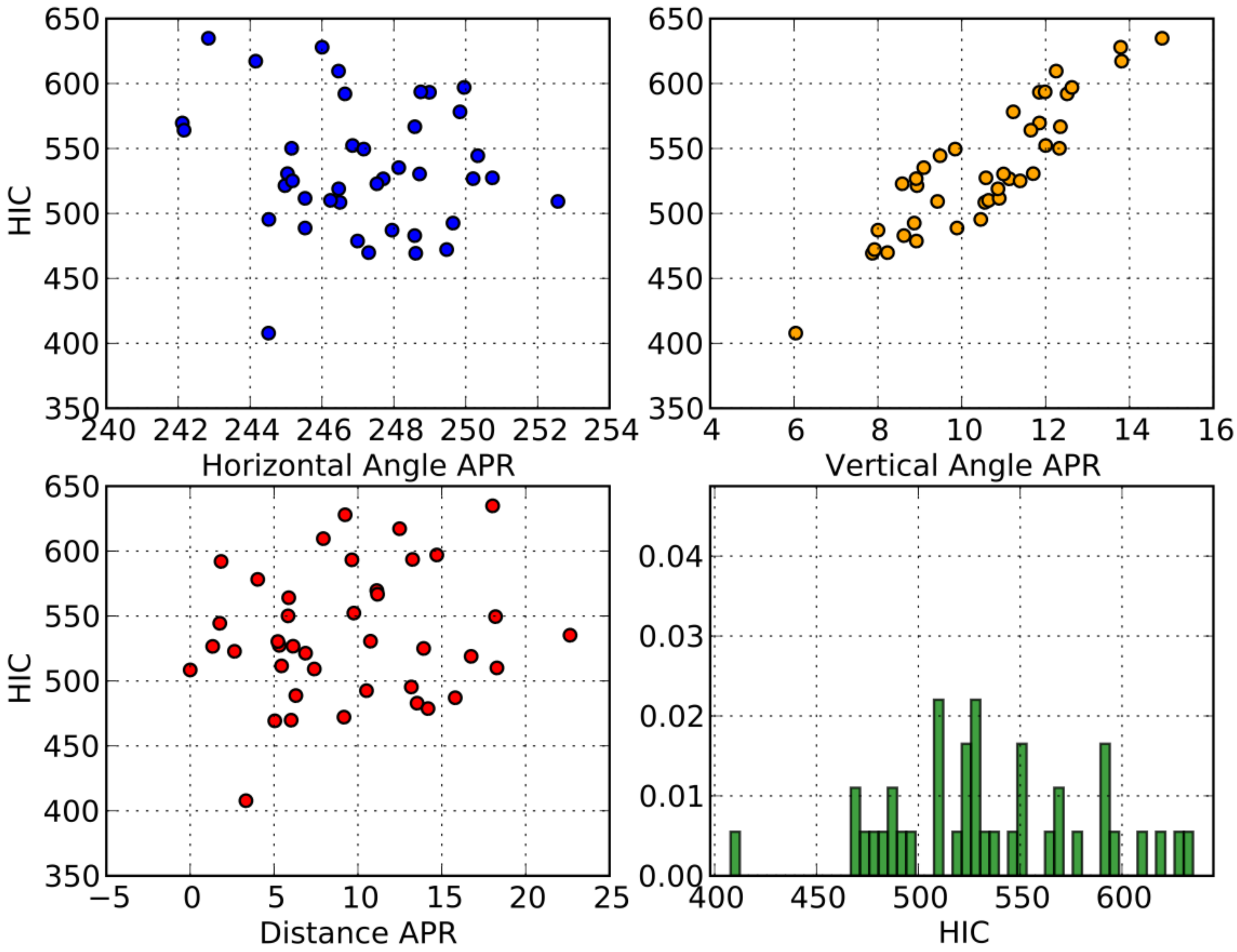
META Script: META_201U.ses

Results Report: ResultsReport.pdf

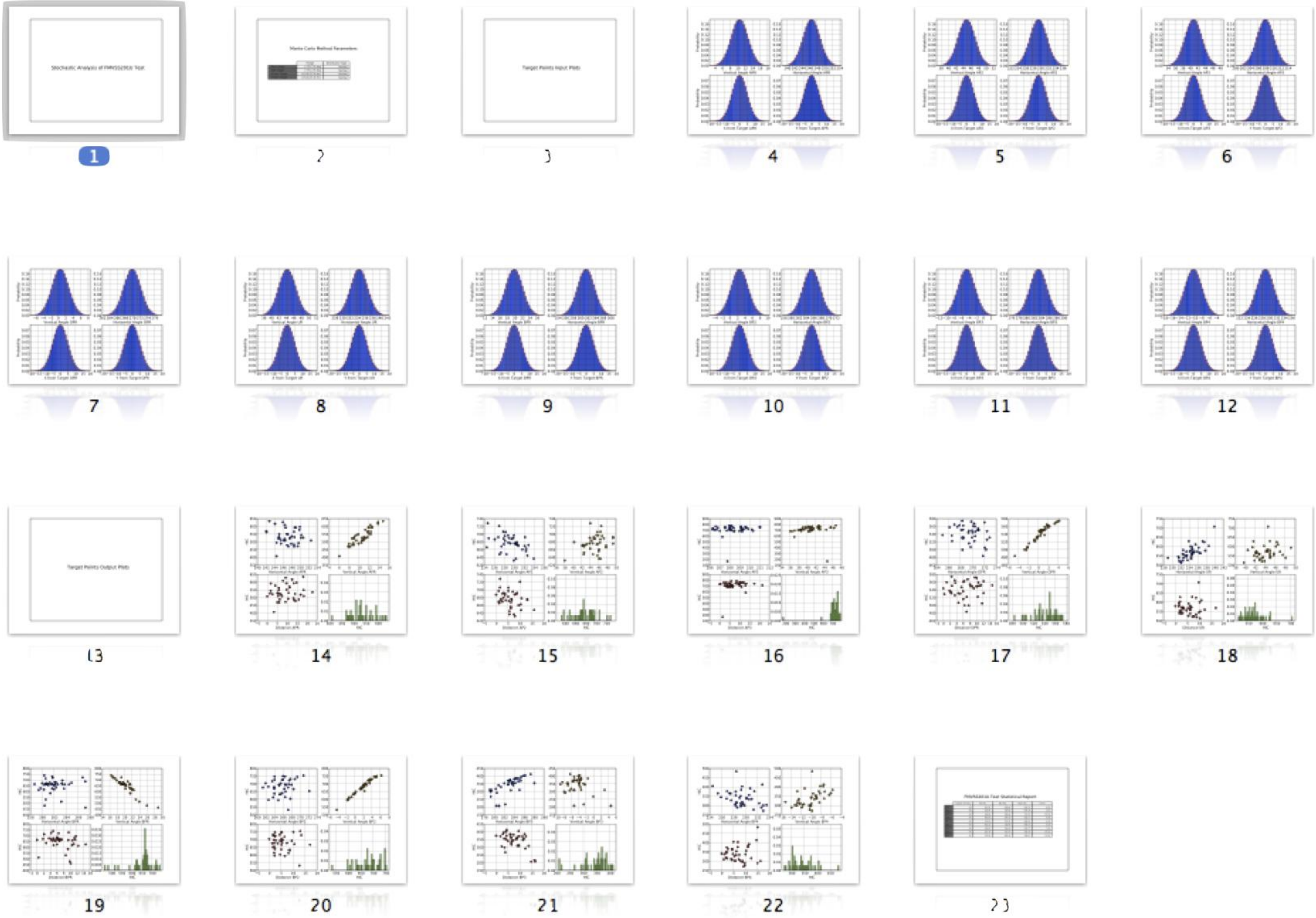


- Automatically create a robustness analysis for a safety loadcase
- Automatically manage solver runs and compile a report

Robustness Analysis Tool



Robustness Analysis Tool

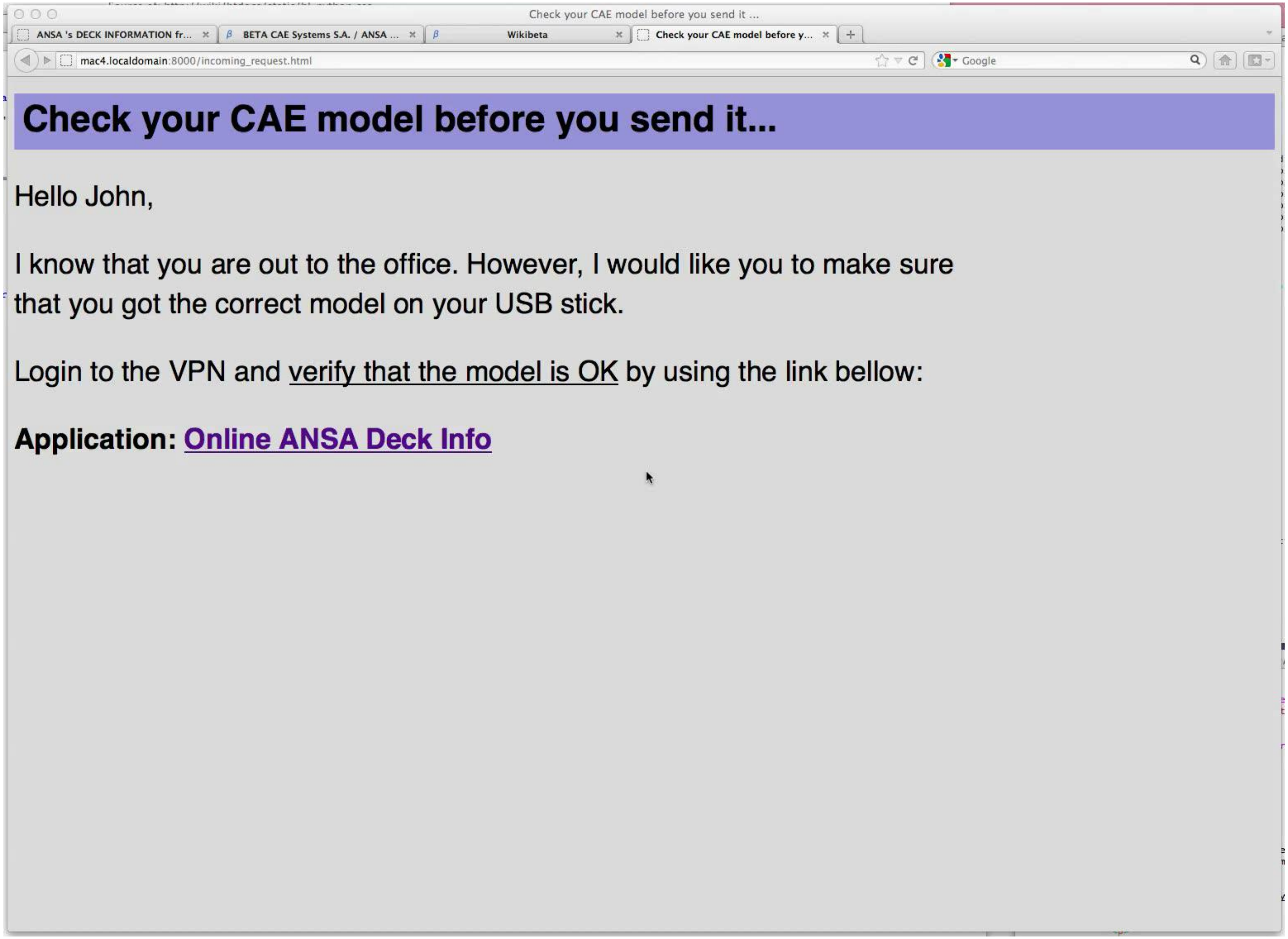

Stochastic Analysis of FMVSS[illegible] Test
Monte Carlo Method Parameters
Target Points Input Plots
Target Points Output Plots
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

Robustness Analysis Tool

Python Benefits

- Use of the NumPy library to run MonteCarlo Method
- Use of the MatPlot lib to plot
- Use of a pdf library to create the pdf report
- Allow user to implement the desired statistical analysis or modify process

Using the Web



Check your CAE model before you send it ...
ANSA 's DECK INFORMATION fr...
BETA CAE Systems S.A. / ANSA ...
Wikibeta
Check your CAE model before y...
mac4.localdomain:8000/incoming_request.html
Google

Check your CAE model before you send it...

Hello John,

I know that you are out to the office. However, I would like you to make sure that you got the correct model on your USB stick.

Login to the VPN and verify that the model is OK by using the link bellow:

Application: Online ANSA Deck Info

Powered by Firefox and WebGL

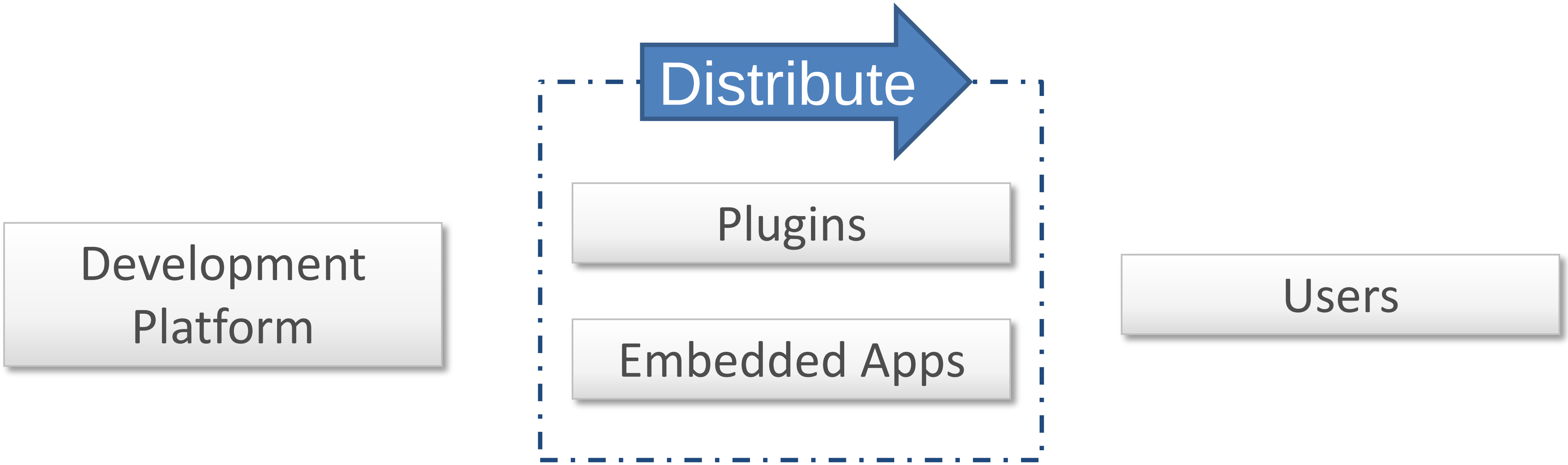
Next Steps

Future Direction

- Enrich the existing ANSA and μETA API
- Allow access to lower level functionality (i.e User Defined Entities)
- Move to a more object oriented design
- Help more people to join the community

Future Direction

Plugin Manager



Thank You

ANSA and μETA as a CAE Software Development Platform

Michael Giannakidis, Yianni Kolokythas

BETA CAE Systems SA