# Experimental Planning - Particular Study

Allan Moreira de Carvalho (allan.carvalh@ufabc.edu.br)

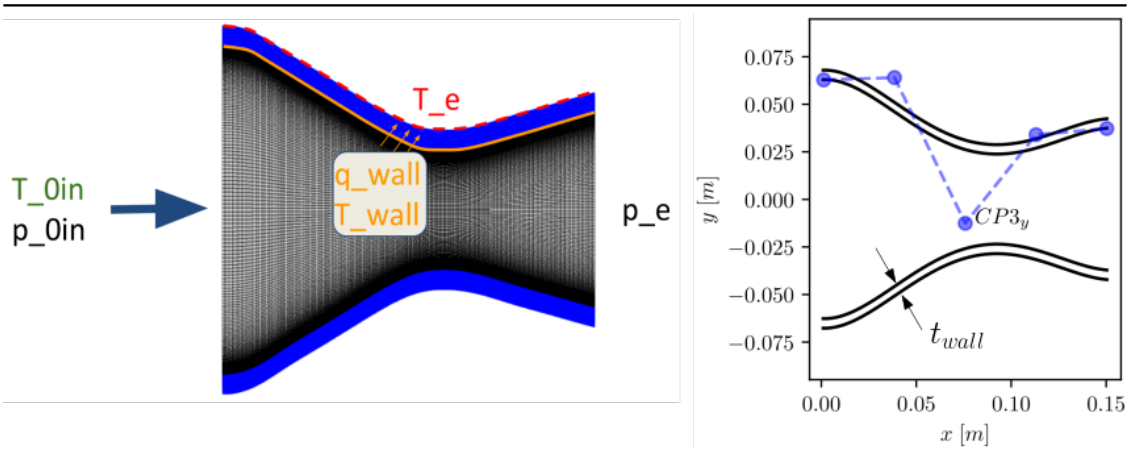
August 16, 2023

## 1 Proposal Study

This proposal employs a factorial design approach to construct a surrogate model. The aim is the reconstruction of 2D fields acquired through Reynolds Averaged Navier Stokes (RANS) solutions, involving conjugate heat transfer of the nozzle flow.

### 1.1 Nozzle Flow and Heat Transfer

Rocket nozzles hold paramount importance in aerospace propulsion. The optimal aerodynamic design of these nozzles is inherently limited by the need to maintain structural integrity, as they are subjected to intense heat transfer from the hot gases to the inner walls.




Representation of Nozzle numerical domain and boundary conditions

## 2 Main Hypotesis

The central hypothesis poses that the factorial design could be effective in acquiring a model with the capacity to predict the principal components of a reduced-order model of fluid flow and heat transfer. This model, once established, can then be used to reconstruct complete two-dimensional flow fields from the designated independent variables.

### 2.1 Experimental Samples

To test this hypothesis, a series of numerical experiments will be conducted following the Central Composite Design (CCD)

- Inlet Total Temperature, $T_{0in}$ [K]
- Inlet Total Pressure, $p_{0in}$ [Pa]
- Wall Thickness, $t_{wall}$ [m]
- Nozzle Shape Control Point, $CP3_y$ [m]
- External Nozzle Wall Temperature, $T_e$ [K]

The SU2 software (Economon et al., 2016) was employed to address the conjugate heat transfer interfaces between the fluid and solid in the nozzle. In the fluid domain, the Reynolds Averaged Navier-Stokes equations were solved using the finite volume method and the SST (Shear Stress Transport) turbulence model. To obtain the steady-state solution, the implicit Euler integration method was utilized in conjunction with time marching. On the other hand, for the solid domain, the energy equation was solved.

## 3 Central Composite Design

The range of variables were choosen in order to be within real experiments made by (Back et al., 1964) . Another factor taking into account to limit variable ranges is the numerical stability of conjugate heat transfer CFD.

[1]:
```
from explann.doe import CentralCompositeDesign
import pandas as pd

variables = {
    't_wall': [0.001, 0.010],               #[m]
    'CP3_y': [-0.01255805, 0.0],            #[m]
    'T_0in': [400.0, 650.0],                #[K]
    'p_0in': [4.0e5, 1.0e6],                #[Pa]
    'T_e': [290.0, 400.0],                  #[K]
}

# variable ranges
pd.DataFrame(variables)
```

[1]:

| | t_wall | CP3_y | T_0in | p_0in | T_e |
|---|---|---|---|---|---|
| 0 | 0.001 | -0.012558 | 400.0 | 400000.0 | 290.0 |
| 1 | 0.010 | 0.000000 | 650.0 | 1000000.0 | 400.0 |

A CCD (Central Composite Desig) was choosen in order to be able to capture any high order interaction fo the design variables.

[2]:
```
ccd = CentralCompositeDesign(
    variables=variables,
    center=(1,0),
    alpha='r',
    face='ccc',
)
ccd.doe
```

[2]:

| Index | t_wall | CP3_y | T_0in | p_0in | T_e |
|---|---|---|---|---|---|
| 1 | -1.000000 | -1.000000 | -1.000000 | -1.000000 | -1.000000 |
| 2 | 1.000000 | -1.000000 | -1.000000 | -1.000000 | -1.000000 |
| 3 | -1.000000 | 1.000000 | -1.000000 | -1.000000 | -1.000000 |
| 4 | 1.000000 | 1.000000 | -1.000000 | -1.000000 | -1.000000 |
| 5 | -1.000000 | -1.000000 | 1.000000 | -1.000000 | -1.000000 |
| 6 | 1.000000 | -1.000000 | 1.000000 | -1.000000 | -1.000000 |
| 7 | -1.000000 | 1.000000 | 1.000000 | -1.000000 | -1.000000 |
| 8 | 1.000000 | 1.000000 | 1.000000 | -1.000000 | -1.000000 |
| 9 | -1.000000 | -1.000000 | -1.000000 | 1.000000 | -1.000000 |
| 10 | 1.000000 | -1.000000 | -1.000000 | 1.000000 | -1.000000 |
| 11 | -1.000000 | 1.000000 | -1.000000 | 1.000000 | -1.000000 |
| 12 | 1.000000 | 1.000000 | -1.000000 | 1.000000 | -1.000000 |
| 13 | -1.000000 | -1.000000 | 1.000000 | 1.000000 | -1.000000 |
| 14 | 1.000000 | -1.000000 | 1.000000 | 1.000000 | -1.000000 |
| 15 | -1.000000 | 1.000000 | 1.000000 | 1.000000 | -1.000000 |
| 16 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | -1.000000 |
| 17 | -1.000000 | -1.000000 | -1.000000 | -1.000000 | 1.000000 |
| 18 | 1.000000 | -1.000000 | -1.000000 | -1.000000 | 1.000000 |
| 19 | -1.000000 | 1.000000 | -1.000000 | -1.000000 | 1.000000 |
| 20 | 1.000000 | 1.000000 | -1.000000 | -1.000000 | 1.000000 |
| 21 | -1.000000 | -1.000000 | 1.000000 | -1.000000 | 1.000000 |
| 22 | 1.000000 | -1.000000 | 1.000000 | -1.000000 | 1.000000 |
| 23 | -1.000000 | 1.000000 | 1.000000 | -1.000000 | 1.000000 |
| 24 | 1.000000 | 1.000000 | 1.000000 | -1.000000 | 1.000000 |
| 25 | -1.000000 | -1.000000 | -1.000000 | 1.000000 | 1.000000 |
| 26 | 1.000000 | -1.000000 | -1.000000 | 1.000000 | 1.000000 |
| 27 | -1.000000 | 1.000000 | -1.000000 | 1.000000 | 1.000000 |
| 28 | 1.000000 | 1.000000 | -1.000000 | 1.000000 | 1.000000 |
| 29 | -1.000000 | -1.000000 | 1.000000 | 1.000000 | 1.000000 |
| 30 | 1.000000 | -1.000000 | 1.000000 | 1.000000 | 1.000000 |
| 31 | -1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| 32 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| 33 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 34 | -2.378414 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 35 | 2.378414 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 36 | 0.000000 | -2.378414 | 0.000000 | 0.000000 | 0.000000 |
| 37 | 0.000000 | 2.378414 | 0.000000 | 0.000000 | 0.000000 |
| 38 | 0.000000 | 0.000000 | -2.378414 | 0.000000 | 0.000000 |
| 39 | 0.000000 | 0.000000 | 2.378414 | 0.000000 | 0.000000 |
| 40 | 0.000000 | 0.000000 | 0.000000 | -2.378414 | 0.000000 |
| 41 | 0.000000 | 0.000000 | 0.000000 | 2.378414 | 0.000000 |
| 42 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | -2.378414 |
| 43 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 2.378414 |

[3]:
```
ccd.levels
```

[3]:

| Levels | t_wall | CP3_y | T_0in | p_0in | T_e |
|---|---|---|---|---|---|
| -2.378414 | 0.001000 | -0.012558 | 400.000000 | 400000.000000 | 290.000000 |
| -1.000000 | 0.003608 | -0.008919 | 472.443974 | 573865.537712 | 321.875349 |
| 0.000000 | 0.005500 | -0.006279 | 525.000000 | 700000.000000 | 345.000000 |
| 1.000000 | 0.007392 | -0.003639 | 577.556026 | 826134.462288 | 368.124651 |
| 2.378414 | 0.010000 | 0.000000 | 650.000000 | 1000000.000000 | 400.000000 |

This DoE variables was then used as boundary conditions for the CFD(Computational Fluid Dynamics) numerical experiments. The 43 runs accounts for accounts for 429.6MB of data stored in a `HDF5` file, from wich, the seleceted fields were choosen to build snapshot matrices

`'Heat_Flux_UPPER_WALL', 'Mach', 'Pressure', 'Temperature', 'Temperature_Solid', 'Temperature_Solid_INNERWALL'`

For each experiment a snapshot vector $\mathbf{S_i}$, corresponding to the concatation of fluid and solid domain 2D numerical solutions were obtained.

$$\mathbf{S_i}(T_{0in}, p_{0in}, t_{wall}, CP3_y) = \begin{bmatrix} \mathbf{q_{upperWall}} \\ \mathbf{p_{fluid}} \\ \mathbf{T_{fluid}} \\ \mathbf{M_{fluid}} \\ \mathbf{T_{solid}} \\ \mathbf{T_{solidInner}} \end{bmatrix}^{252840\times 1}$$

, the snapshots $\mathbf{S_i}$ were then concatenated to form the snapshot matrix $\mathbf{A}$ of the Design of Experimen (DoE).

$$\mathbf{A} = \begin{bmatrix} \mathbf{S_1^T} \\ \mathbf{S_2^T} \\ \vdots \\ \mathbf{S_N^T} \end{bmatrix}^{43\times 252840}$$

[4]:
```
import sys
sys.path.append('/home/ppiper/Dropbox/local/github/frog')

from frog.datahandler import HDF5Handler

hfdh = HDF5Handler(
    datapath='/home/ppiper/Dropbox/local/github/explann/data/
 ↪experimental_planning_T0in_limit_clear/hf_transpose.h5',
    datasets = ['Heat_Flux_UPPER_WALL', 'Mach', 'Pressure', 'Temperature',␣
 ↪'Temperature_Solid', 'Temperature_Solid_INNERWALL'],
)
```

# 4 Dependent variables (Order Reduction)

The high dimensional space of samples will subsequently submited to an order reduction, giveng by Principal Component Analysis (PCA), resulting into a more manageable, low-dimensional orthogonal basis. Subsequently, a factorial model will be fitted to predict the principal components as a function of the independent variables. The resultant model will then be applied to reconstruct flow field and heat transfer aspects for a designated test set of independent variables. This reconstructed output will be compared against the CFD solutions for error analysis.

The dependent variables are principal componentes of a POD(Proper Orthogonal Decompostion) of the snapshots matrix $\mathbf{A}$

$$\mathbf{A} = \mathbf{U}\ \mathbf{V^T}$$

by truncanting the diagonal matrix of singular values  , the snapshot matrix can be represented in terms o a reduced basis $\tilde{\mathbf{U}}$ and principal components$\tilde{}$

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}}$$

The data were then compressed using 10 principal components of the orthogonal projection, resulting in 24.3MB of data. An approximated 20-fold reduction in storage requirments.

[5]:
```
from sklearn.pipeline import Pipeline
from sklearn.decomposition import PCA
from frog.normalization import Normalization, DataHandlerNormalization
import pickle

rom = Pipeline([
    ('scaler', Normalization(bounds=[-1,1])),
    ('pca', PCA(n_components=10)),
])

rom.fit(hfdh.data)

results_pc = rom.transform(hfdh.data)

with open('/home/ppiper/Dropbox/local/github/explann/data/
 ↪experimental_planning_T0in_limit_clear/rom.pkl','wb') as f:
    pickle.dump(rom,f)

with open('/home/ppiper/Dropbox/local/github/explann/data/
 ↪experimental_planning_T0in_limit_clear/rom.pkl','rb') as f:
    rom = pickle.load(f)
```

The principal components are then joined to the CCD DoE to form full tables for factorial model construction.

[6]:
```
import pandas as pd
results = {f'L{i}': results_pc[:,i] for i in range(results_pc.shape[1])}

ccd.append_results(results)
ccd.doe
```

[6]:

| Index | t_wall | CP3_y | T_0in | p_0in | T_e | L0 | L1 |
|---|---|---|---|---|---|---|---|
| 1 | -1.000000 | -1.000000 | -1.000000 | -1.000000 | -1.000000 | 47.655208 | -2.574525 |
| 2 | 1.000000 | -1.000000 | -1.000000 | -1.000000 | -1.000000 | 47.657308 | -2.584284 |
| 3 | -1.000000 | 1.000000 | -1.000000 | -1.000000 | -1.000000 | 47.482853 | 2.523185 |
| 4 | 1.000000 | 1.000000 | -1.000000 | -1.000000 | -1.000000 | 47.484867 | 2.512161 |
| 5 | -1.000000 | -1.000000 | 1.000000 | -1.000000 | -1.000000 | 47.629272 | -2.501821 |
| 6 | 1.000000 | -1.000000 | 1.000000 | -1.000000 | -1.000000 | 47.632879 | -2.517772 |
| 7 | -1.000000 | 1.000000 | 1.000000 | -1.000000 | -1.000000 | 47.456260 | 2.599557 |
| 8 | 1.000000 | 1.000000 | 1.000000 | -1.000000 | -1.000000 | 47.459997 | 2.581783 |
| 9 | -1.000000 | -1.000000 | -1.000000 | 1.000000 | -1.000000 | -47.418552 | -3.711689 |
| 10 | 1.000000 | -1.000000 | -1.000000 | 1.000000 | -1.000000 | -47.415844 | -3.725914 |
| 11 | -1.000000 | 1.000000 | -1.000000 | 1.000000 | -1.000000 | -47.666183 | 3.624649 |
| 12 | 1.000000 | 1.000000 | -1.000000 | 1.000000 | -1.000000 | -47.663805 | 3.609438 |
| 13 | -1.000000 | -1.000000 | 1.000000 | 1.000000 | -1.000000 | -47.451135 | -3.618356 |
| 14 | 1.000000 | -1.000000 | 1.000000 | 1.000000 | -1.000000 | -47.446400 | -3.641930 |
| 15 | -1.000000 | 1.000000 | 1.000000 | 1.000000 | -1.000000 | -47.698985 | 3.721687 |
| 16 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | -1.000000 | -47.694650 | 3.696741 |
| 17 | -1.000000 | -1.000000 | -1.000000 | -1.000000 | 1.000000 | 47.655208 | -2.574525 |
| 18 | 1.000000 | -1.000000 | -1.000000 | -1.000000 | 1.000000 | 47.657308 | -2.584284 |
| 19 | -1.000000 | 1.000000 | -1.000000 | -1.000000 | 1.000000 | 47.482853 | 2.523185 |
| 20 | 1.000000 | 1.000000 | -1.000000 | -1.000000 | 1.000000 | 47.484867 | 2.512161 |
| 21 | -1.000000 | -1.000000 | 1.000000 | -1.000000 | 1.000000 | 47.629272 | -2.501821 |
| 22 | 1.000000 | -1.000000 | 1.000000 | -1.000000 | 1.000000 | 47.632879 | -2.517772 |
| 23 | -1.000000 | 1.000000 | 1.000000 | -1.000000 | 1.000000 | 47.456260 | 2.599557 |
| 24 | 1.000000 | 1.000000 | 1.000000 | -1.000000 | 1.000000 | 47.459997 | 2.581783 |
| 25 | -1.000000 | -1.000000 | -1.000000 | 1.000000 | 1.000000 | -47.418552 | -3.711689 |
| 26 | 1.000000 | -1.000000 | -1.000000 | 1.000000 | 1.000000 | -47.415844 | -3.725914 |
| 27 | -1.000000 | 1.000000 | -1.000000 | 1.000000 | 1.000000 | -47.666183 | 3.624649 |
| 28 | 1.000000 | 1.000000 | -1.000000 | 1.000000 | 1.000000 | -47.663805 | 3.609438 |
| 29 | -1.000000 | -1.000000 | 1.000000 | 1.000000 | 1.000000 | -47.451135 | -3.618356 |
| 30 | 1.000000 | -1.000000 | 1.000000 | 1.000000 | 1.000000 | -47.446400 | -3.641930 |
| 31 | -1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | -47.698985 | 3.721687 |
| 32 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | -47.694650 | 3.696741 |
| 33 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.025936 | -0.038532 |
| 34 | -2.378414 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.022612 | -0.015342 |
| 35 | 2.378414 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.029698 | -0.056069 |
| 36 | 0.000000 | -2.378414 | 0.000000 | 0.000000 | 0.000000 | 0.137596 | -7.218266 |
| 37 | 0.000000 | 2.378414 | 0.000000 | 0.000000 | 0.000000 | -0.381774 | 7.591583 |
| 38 | 0.000000 | 0.000000 | -2.378414 | 0.000000 | 0.000000 | 0.059251 | -0.135482 |
| 39 | 0.000000 | 0.000000 | 2.378414 | 0.000000 | 0.000000 | -0.009178 | 0.056877 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 40 | 0.000000 | 0.000000 | 0.000000 | -2.378414 | 0.000000 | 113.128140 | -0.016841 |
| 41 | 0.000000 | 0.000000 | 0.000000 | 2.378414 | 0.000000 | -113.070333 | -0.076683 |
| 42 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | -2.378414 | 0.025936 | -0.038532 |
| 43 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 2.378414 | 0.025936 | -0.038532 |

| Index | L2 | L3 | L4 | L5 | L6 | L7 | L8 \ |
|---|---|---|---|---|---|---|---|
| 1 | 1.054864 | 0.116466 | 0.006886 | 0.026542 | -0.019255 | 0.012261 | -0.003517 |
| 2 | 1.705080 | -0.057877 | 0.004462 | 0.046149 | -0.004435 | -0.013983 | 0.008446 |
| 3 | 1.070312 | 0.115771 | 0.007093 | 0.020386 | -0.025199 | 0.005370 | -0.005530 |
| 4 | 1.748841 | -0.031342 | 0.003471 | 0.026343 | -0.011534 | -0.007418 | 0.008019 |
| 5 | -1.897897 | 0.122485 | -0.010489 | -0.061067 | 0.013869 | 0.004764 | 0.001125 |
| 6 | -0.854666 | -0.157822 | -0.009886 | -0.029965 | 0.015498 | -0.008033 | 0.008307 |
| 7 | -1.962213 | 0.058714 | -0.004065 | -0.022829 | -0.001661 | -0.012595 | -0.013704 |
| 8 | -0.875525 | -0.179893 | -0.005059 | -0.016389 | -0.003456 | -0.008670 | -0.003721 |
| 9 | 1.184190 | 0.148468 | 0.003752 | -0.001800 | 0.000860 | 0.003282 | -0.012650 |
| 10 | 2.135199 | -0.085025 | -0.001124 | -0.016591 | 0.008043 | -0.013796 | -0.002655 |
| 11 | 1.288358 | 0.170554 | -0.002338 | -0.021060 | 0.008651 | -0.001266 | -0.003357 |
| 12 | 2.268540 | -0.022280 | -0.010885 | -0.053166 | 0.020144 | 0.008198 | 0.006091 |
| 13 | -2.496137 | 0.145204 | 0.006485 | 0.028797 | 0.001590 | -0.001518 | 0.004297 |
| 14 | -0.984372 | -0.229059 | 0.005923 | 0.006421 | -0.017370 | 0.012474 | -0.000379 |
| 15 | -2.469012 | 0.095408 | 0.006392 | 0.063492 | 0.008184 | -0.021897 | 0.007073 |
| 16 | -0.914726 | -0.216737 | 0.001333 | 0.009757 | -0.010209 | 0.027114 | 0.000185 |
| 17 | 1.054864 | 0.116466 | 0.006886 | 0.026542 | -0.019255 | 0.012261 | -0.003517 |
| 18 | 1.705080 | -0.057877 | 0.004462 | 0.046149 | -0.004435 | -0.013983 | 0.008446 |
| 19 | 1.070312 | 0.115771 | 0.007093 | 0.020386 | -0.025199 | 0.005370 | -0.005530 |
| 20 | 1.748841 | -0.031342 | 0.003471 | 0.026343 | -0.011534 | -0.007418 | 0.008019 |
| 21 | -1.897897 | 0.122485 | -0.010489 | -0.061067 | 0.013869 | 0.004764 | 0.001125 |
| 22 | -0.854666 | -0.157822 | -0.009886 | -0.029965 | 0.015498 | -0.008033 | 0.008307 |
| 23 | -1.962213 | 0.058714 | -0.004065 | -0.022829 | -0.001661 | -0.012595 | -0.013704 |
| 24 | -0.875525 | -0.179893 | -0.005059 | -0.016389 | -0.003456 | -0.008670 | -0.003721 |
| 25 | 1.184190 | 0.148468 | 0.003752 | -0.001800 | 0.000860 | 0.003282 | -0.012650 |
| 26 | 2.135199 | -0.085025 | -0.001124 | -0.016591 | 0.008043 | -0.013796 | -0.002655 |
| 27 | 1.288358 | 0.170554 | -0.002338 | -0.021060 | 0.008651 | -0.001266 | -0.003357 |
| 28 | 2.268540 | -0.022280 | -0.010885 | -0.053166 | 0.020144 | 0.008198 | 0.006091 |
| 29 | -2.496137 | 0.145204 | 0.006485 | 0.028797 | 0.001590 | -0.001518 | 0.004297 |
| 30 | -0.984372 | -0.229059 | 0.005923 | 0.006421 | -0.017370 | 0.012474 | -0.000379 |
| 31 | -2.469012 | 0.095408 | 0.006392 | 0.063492 | 0.008184 | -0.021897 | 0.007073 |
| 32 | -0.914726 | -0.216737 | 0.001333 | 0.009757 | -0.010209 | 0.027114 | 0.000185 |
| 33 | -0.035311 | -0.020620 | -0.031631 | 0.018992 | -0.033009 | 0.004036 | -0.009210 |
| 34 | -1.611487 | 0.417478 | -0.036272 | -0.035432 | -0.028328 | 0.022534 | 0.024267 |
| 35 | 1.020713 | -0.226146 | -0.035953 | -0.007553 | -0.032945 | 0.004386 | 0.014950 |
| 36 | -0.059227 | -0.014131 | 0.189392 | -0.011965 | 0.017133 | 0.001690 | 0.000903 |
| 37 | 0.028908 | -0.014403 | 0.190904 | -0.011537 | 0.015566 | 0.002630 | -0.000132 |
| 38 | 3.737966 | 0.092381 | -0.044229 | -0.044475 | -0.002733 | -0.021055 | 0.002082 |
| 39 | -3.682362 | -0.139727 | -0.048415 | -0.052666 | -0.000609 | -0.014789 | -0.004364 |
| 40 | 0.340592 | -0.011511 | -0.054779 | 0.059510 | 0.105344 | 0.029156 | -0.003415 |

```
41     0.329156 -0.028152 -0.069653  0.037104  0.058162 -0.005234 -0.002726
42    -0.035311 -0.020620 -0.031631  0.018992 -0.033009  0.004036 -0.009210
43    -0.035311 -0.020620 -0.031631  0.018992 -0.033009  0.004036 -0.009210
```

| Index | L9 |
|---|---|
| 1 | -0.003013 |
| 2 | -0.001133 |
| 3 | -0.004666 |
| 4 | -0.007382 |
| 5 | -0.008234 |
| 6 | 0.003685 |
| 7 | 0.005997 |
| 8 | 0.009605 |
| 9 | 0.009544 |
| 10 | 0.007812 |
| 11 | 0.001258 |
| 12 | -0.006560 |
| 13 | -0.008894 |
| 14 | -0.001259 |
| 15 | 0.003627 |
| 16 | -0.000820 |
| 17 | -0.003013 |
| 18 | -0.001133 |
| 19 | -0.004666 |
| 20 | -0.007382 |
| 21 | -0.008234 |
| 22 | 0.003685 |
| 23 | 0.005997 |
| 24 | 0.009605 |
| 25 | 0.009544 |
| 26 | 0.007812 |
| 27 | 0.001258 |
| 28 | -0.006560 |
| 29 | -0.008894 |
| 30 | -0.001259 |
| 31 | 0.003627 |
| 32 | -0.000820 |
| 33 | -0.002065 |
| 34 | 0.016765 |
| 35 | 0.010513 |
| 36 | -0.000792 |
| 37 | 0.000395 |
| 38 | -0.007991 |
| 39 | -0.012773 |
| 40 | 0.004612 |
| 41 | -0.003670 |

```
42    -0.002065
43    -0.002065
```

## 5 Fitting Factorial Model

The CCD factorial model accounts for second order terms in each $\mathbf{L_i}$ componenent of the  matrix.

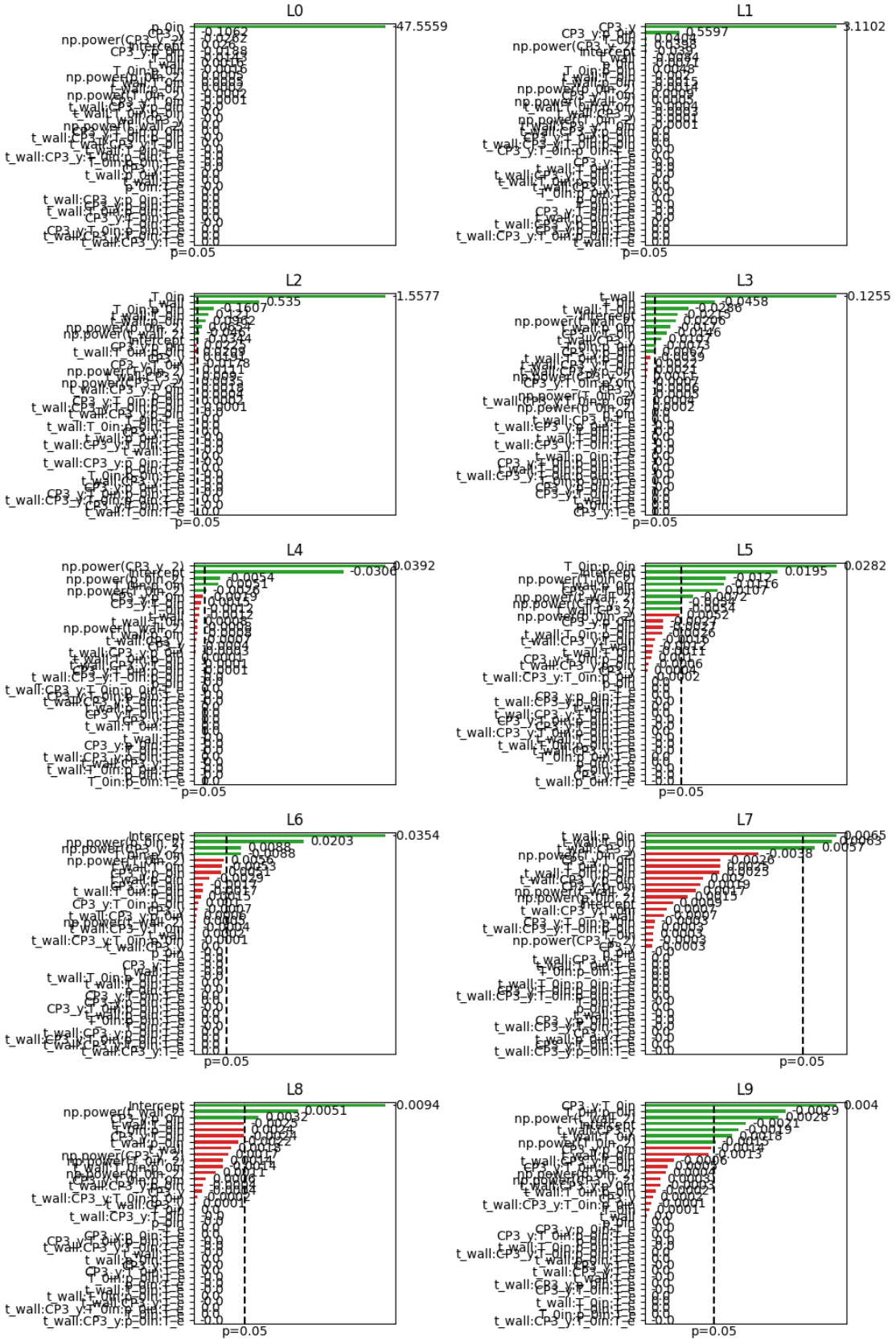
[7]:
```
from explann.models import FactorialModel
from explann.dataio import ImportString, ImportXLSX
from explann.plot import ParetoPlot
import matplotlib.pyplot as plt

expressions = {f'{key}': f'{key} ~ 1 + t_wall * CP3_y * T_0in * p_0in * T_e +␣
 ↪np.power(t_wall, 2) + np.power(T_0in, 2) + np.power(p_0in, 2) + np.
 ↪power(CP3_y,2) + np.power(CP3_y, 2)' for key in results.keys()}

fm_ccd = FactorialModel(
    data = ccd.doe,
    functions = expressions,
    levels = ccd.levels,
)
expressions
```

[7]:
```
{'L0': 'L0 ~ 1 + t_wall * CP3_y * T_0in * p_0in * T_e + np.power(t_wall, 2) +
np.power(T_0in, 2) + np.power(p_0in, 2) + np.power(CP3_y,2) + np.power(CP3_y,
2)',
 'L1': 'L1 ~ 1 + t_wall * CP3_y * T_0in * p_0in * T_e + np.power(t_wall, 2) +
np.power(T_0in, 2) + np.power(p_0in, 2) + np.power(CP3_y,2) + np.power(CP3_y,
2)',
 'L2': 'L2 ~ 1 + t_wall * CP3_y * T_0in * p_0in * T_e + np.power(t_wall, 2) +
np.power(T_0in, 2) + np.power(p_0in, 2) + np.power(CP3_y,2) + np.power(CP3_y,
2)',
 'L3': 'L3 ~ 1 + t_wall * CP3_y * T_0in * p_0in * T_e + np.power(t_wall, 2) +
np.power(T_0in, 2) + np.power(p_0in, 2) + np.power(CP3_y,2) + np.power(CP3_y,
2)',
 'L4': 'L4 ~ 1 + t_wall * CP3_y * T_0in * p_0in * T_e + np.power(t_wall, 2) +
np.power(T_0in, 2) + np.power(p_0in, 2) + np.power(CP3_y,2) + np.power(CP3_y,
2)',
 'L5': 'L5 ~ 1 + t_wall * CP3_y * T_0in * p_0in * T_e + np.power(t_wall, 2) +
np.power(T_0in, 2) + np.power(p_0in, 2) + np.power(CP3_y,2) + np.power(CP3_y,
2)',
 'L6': 'L6 ~ 1 + t_wall * CP3_y * T_0in * p_0in * T_e + np.power(t_wall, 2) +
np.power(T_0in, 2) + np.power(p_0in, 2) + np.power(CP3_y,2) + np.power(CP3_y,
2)',
 'L7': 'L7 ~ 1 + t_wall * CP3_y * T_0in * p_0in * T_e + np.power(t_wall, 2) +
np.power(T_0in, 2) + np.power(p_0in, 2) + np.power(CP3_y,2) + np.power(CP3_y,
```

```
2)',
 'L8': 'L8 ~ 1 + t_wall * CP3_y * T_0in * p_0in * T_e + np.power(t_wall, 2) +
np.power(T_0in, 2) + np.power(p_0in, 2) + np.power(CP3_y,2) + np.power(CP3_y,
2)',
 'L9': 'L9 ~ 1 + t_wall * CP3_y * T_0in * p_0in * T_e + np.power(t_wall, 2) +
np.power(T_0in, 2) + np.power(p_0in, 2) + np.power(CP3_y,2) + np.power(CP3_y,
2)'}
```

As can be viewed in pareto plots, not all terms are significant, so we can retaing fewer terms.

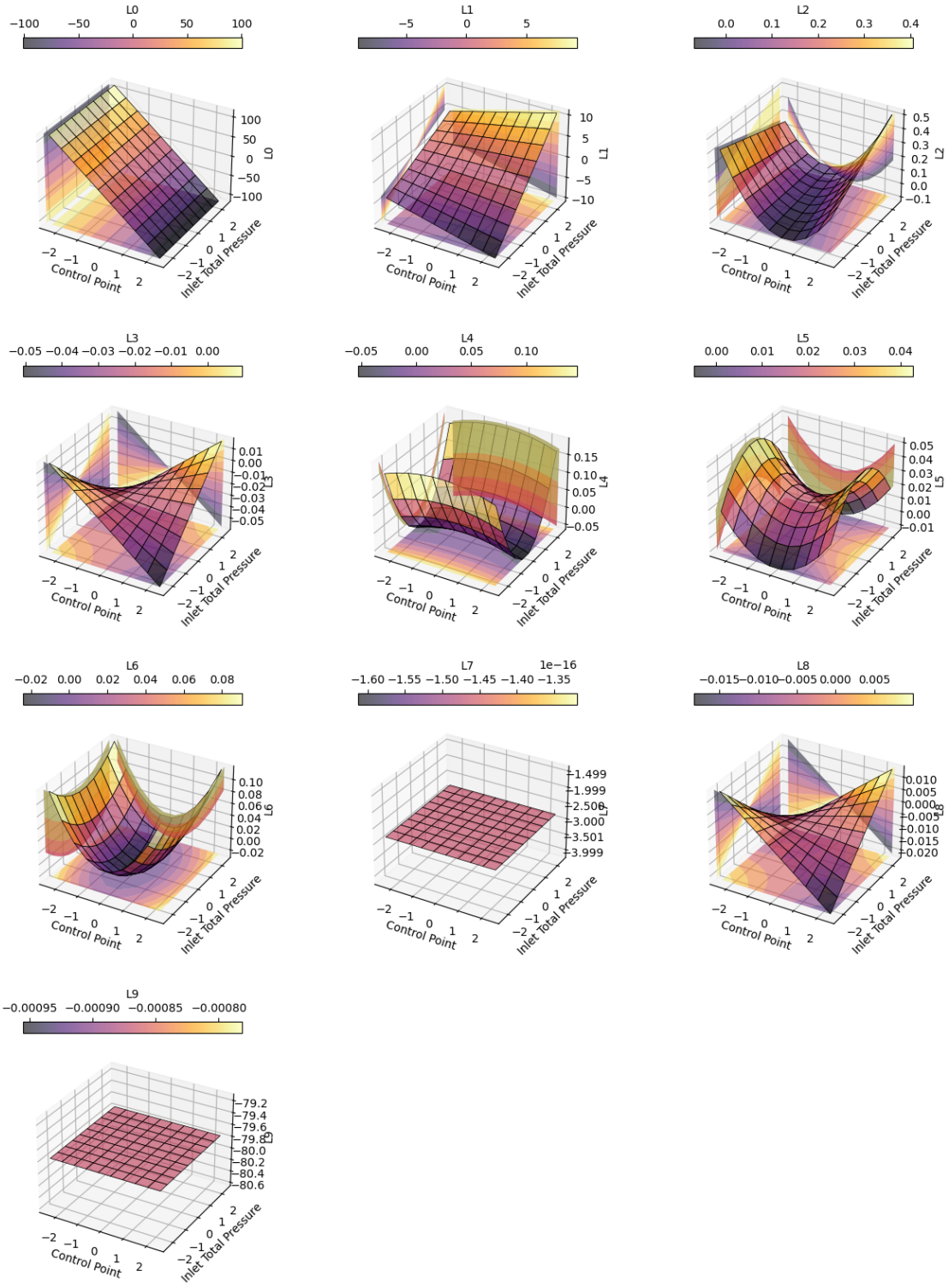
[8]:
```
from explann.plot import ParetoPlot
import matplotlib.pyplot as plt

fig, ax = plt.subplots(5,2, figsize=(10,15))
ax = ax.flatten()

pp = ParetoPlot(fm_ccd)

pp.plot(ax=ax)
plt.tight_layout()
```


L0
L1
L2
L3
L4
L5
L6
L7
L8
L9
p=0.05

[9]:
```
fm_ccd.get_significant_model_functions()
fm_ccd_full = fm_ccd
fm_ccd = fm_ccd.build_significant_models()
```

## 6 Resulting Model for principal components

[10]:
```
from explann.plot import plot_surface

fig, ax = plt.subplots(4,3, figsize=(15,20),subplot_kw={"projection": "3d"})
ax = ax.flatten()

for i,axi in enumerate(ax):#fm_ccd.functions.items()):
    if i<len(fm_ccd.functions.items()):
        key = list(fm_ccd.functions.keys())[i]
        plot_surface(x='p_0in', y='CP3_y', z=key,
             model=fm_ccd,
             n_pts=10,
             ax=ax[i],
             other_params=dict(t_wall=0,T_0in=0,T_e=0),
             labels=dict(xlabel='Control Point', ylabel='Inlet Total Pressure',␣
 ↪zlabel=key), cmap='inferno', scaled=False)
    else:
        ax[i].set_axis_off()
```


L0
−100 −50 0 50 100
L0
Control Point
Inlet Total Pressure
L1
−5 0 5
L1
Control Point
Inlet Total Pressure
L2
0.0 0.1 0.2 0.3 0.4
L2
Control Point
Inlet Total Pressure
L3
−0.05 −0.04 −0.03 −0.02 −0.01 0.00
L3
Control Point
Inlet Total Pressure
L4
−0.05 0.00 0.05 0.10
L4
Control Point
Inlet Total Pressure
L5
0.00 0.01 0.02 0.03 0.04
L5
Control Point
Inlet Total Pressure
L6
−0.02 0.00 0.02 0.04 0.06 0.08
L6
Control Point
Inlet Total Pressure
L7
1e−16
−1.60 −1.55 −1.50 −1.45 −1.40 −1.35
L7
Control Point
Inlet Total Pressure
L8
−0.015 −0.010 −0.005 0.000 0.005
L8
Control Point
Inlet Total Pressure
L9
−0.00095 −0.00090 −0.00085 −0.00080
L9
Control Point
Inlet Total Pressure

# 7 Test Model Accuracy

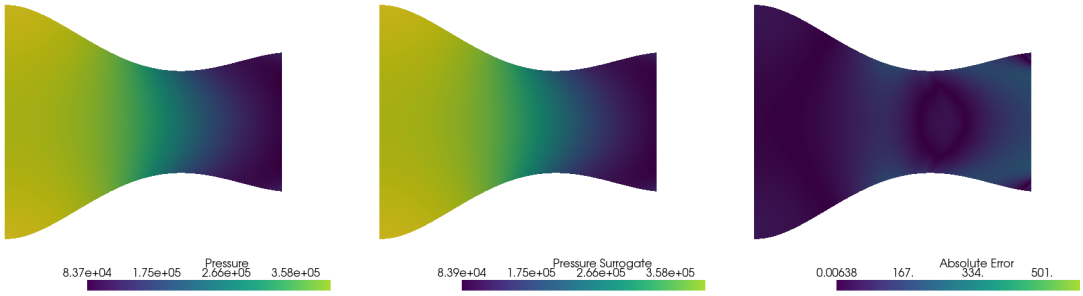
The model accuracy was tested by comparing flow reconstruction with full numerical solution for an unseeing set of independent variables

[11]:
```
import numpy as np

def surrogate(model, variables, rom=None):
    independent_vars_coded = {}
    independent_vars = variables.copy()
    for var, value in independent_vars.items():
        independent_vars_coded[var] = np.interp(value,  model.levels[f'{var}'].
 ↪values, model.levels.index.values)

    L_predicted = []
    for dependent in model.dependent_variables:
        L_predicted.append(model[dependent].predict(independent_vars_coded).
 ↪item())

    L_predicted = np.array(L_predicted)

    if rom is not None:
        L_predicted = rom.inverse_transform(L_predicted)

    return L_predicted

surrogate_solution = surrogate(
    model=fm_ccd_full,
    variables={
        't_wall': 0.006044329228014826,
        'CP3_y': -0.001277584765410799,
        'T_0in': 616.0,
        'p_0in': 456429.13834591914,
        'T_e': 318.0
    },
    rom=rom)
```

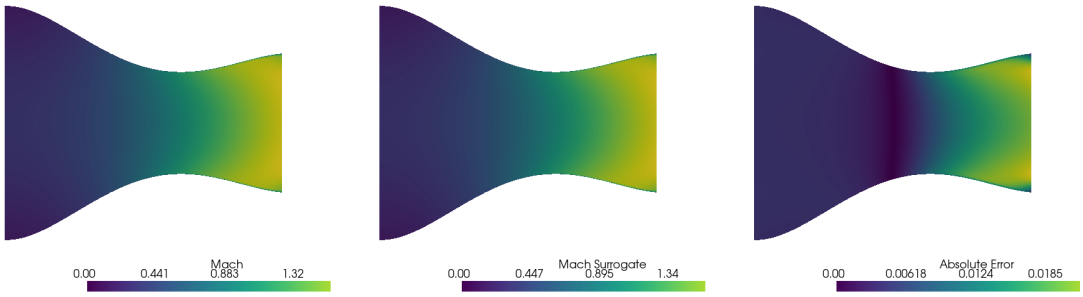
inverse_transform

[12]:
```
from plot_2d import plot_property
```

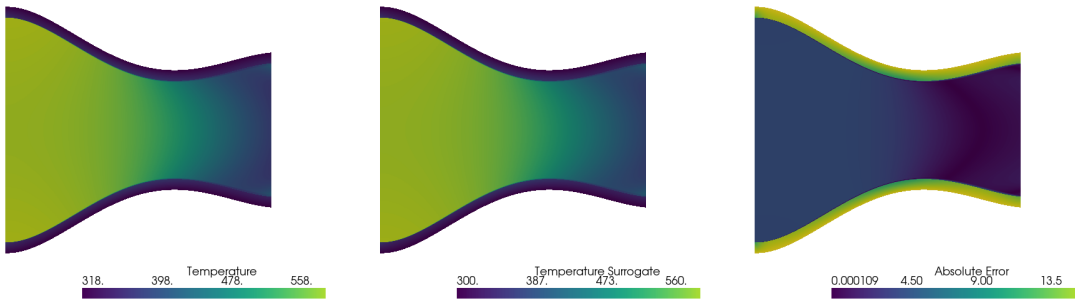
[13]:
```
plot_property(
    prop='Pressure',
    unseeing_path='/home/ppiper/Dropbox/local/github/explann/data/lhs/10/SU2/
 ↪outputs/cht_setupSU2.vtm',
    surrogate_solution=surrogate_solution,
    hfdh=hfdh,
    solid=False)
```




[14]:
```
plot_property(
    prop='Mach',
    unseeing_path='/home/ppiper/Dropbox/local/github/explann/data/lhs/10/SU2/
 ↪outputs/cht_setupSU2.vtm',
    surrogate_solution=surrogate_solution,
    hfdh=hfdh,
    solid=False)
```




[15]:
```
plot_property(
    prop='Temperature',
    unseeing_path='/home/ppiper/Dropbox/local/github/explann/data/lhs/10/SU2/
 ↪outputs/cht_setupSU2.vtm',
    surrogate_solution=surrogate_solution,
    hfdh=hfdh,
    solid=True)
```




# 8 References

Back, L.H., Massier, P.F. and Gier, H.L., 1964. "Convective heat transfer in a convergent-divergent nozzle". International Journal of Heat and Mass Transfer, Vol. 7, pp. 549–568.

Economon, T.D., Palacios, F., Copeland, S.R., Lukaczyk, T.W. and Alonso, J.J., 2016. "SU2: An open-source suite for multiphysics simulation and design". AIAA Journal, Vol. 54, No. 3, pp. 828–846. doi:10.2514/1.j053813.