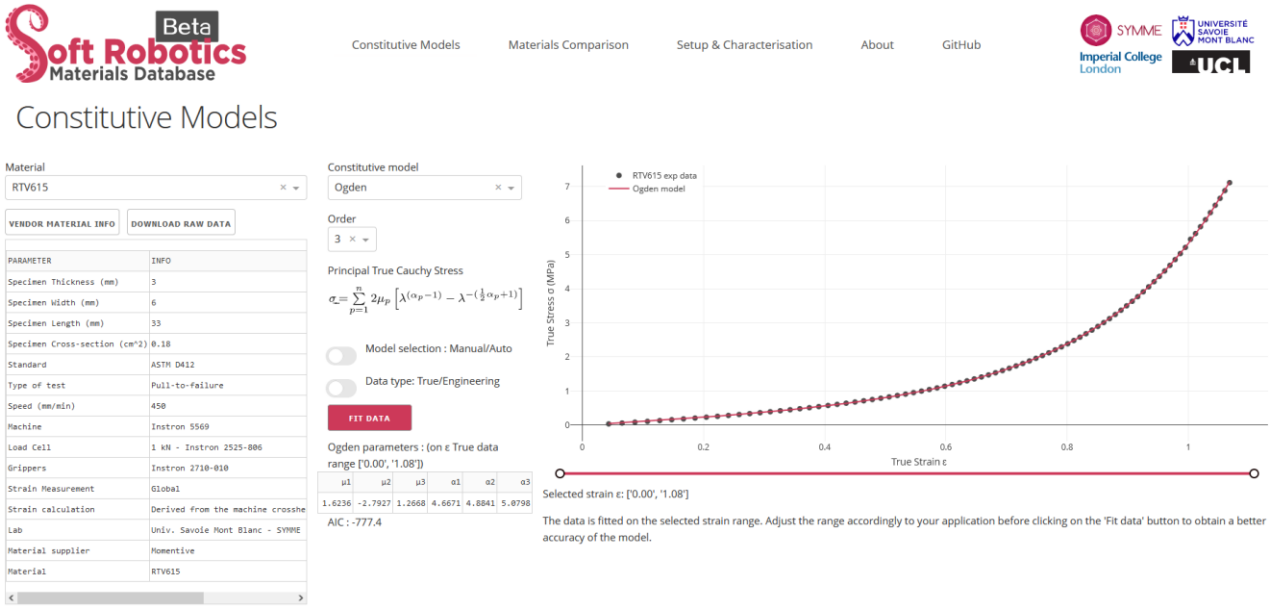
Soft Robotics Materials Database and experimental tensile test data fitting tutorial

Abstract: This tutorial will provide an overview of the open source Soft Robotics Materials Database and its web App designed to aid material selection for soft Robotics application. It will also include an introduction to modelling constitutive models for hyperelastic materials from experimental tensile test data. Attendees will be able to practice experimental data fitting using a non linear optimization technique by coding and testing their own Python script under a Jupyter notebook.




```
[1]: # Pandas and table
import pandas as pd
# Opimization
from scipy.optimize import least_squares
import numpy as np
```

```
[*]: ##############################################################################
#  OPTIMIZATION
##############################################################################

def OgdenModel(trueStrain, parameters, order):
    """Ogden hyperelastic model (incompressible material under uniaxial tension)
    Uses true strain and true stress data"""

    # parameter is a 1D array : [mu0,mu1,...,mun,alpha0,alpha1,...,alphan]
    muVec = parameters.reshape(2, order)[0]
    alphaVec = parameters.reshape(2, order)[1]
    lambd = np.exp(trueStrain)
    # broadcasting method to speed up computation
    lambd = lambd[np.newaxis, :]
    muVec = muVec[:order, np.newaxis]
    alphaVec = alphaVec[:order, np.newaxis]

    trueStress = np.sum(2*muVec*(lambd**(alphaVec - 1) - lambd**(-((1/2)*alphaVec + 1))), axis=0)
    return trueStress


# cost function to calculate the residuals. The fitting function holds the parameter values.
def objectiveFun_Callback(parameters, exp_strain, exp_stress):
    theo_stress = OgdenModel(exp_strain, parameters, order)
    residuals = theo_stress - exp_stress
    return residuals
```