# Appendix – Analysis Code Files and Description

## Step by step solution process:

This code **EUC\_PYTHON\_1.py** calls Abaqus Model of EUC (Microscopic Model) for evaluation of equivalent materials properties and update them in EUC code of Macroscopic Model. Following that the iterative solution process (called Job in Abacus) of the modified macroscopic and microscopic models is initiated.

The script **EUC\_PYTHON\_1.py** is designed to process three specific input models, namely **1\_EUC\_alfa11.inp**, **1\_EUC\_alfa12.inp**, and **1\_EUC\_alfa22.inp**. These models represent Microscopic Equivalent Uniform Condition (EUC) Models that are subjected to a Unit Temperature Load. The primary function of this script is to compute the equivalent material properties based on the data provided by these models. Upon successful calculation, **EUC\_PYTHON\_1.py** proceeds to modify the subsequent Macroscopic INP file Model found in **Macro.inp** to generate the modified model **Macro\_new.inp**. This modification involves the integration of the newly calculated properties, but it is restricted to user-defined elements within the model. The script ensures that only these specified elements are updated with the new material properties, maintaining the integrity of the remaining components of the macroscopic model.

As a separate offline procedure prior to job application, a NI Labview app is processed to identify the corresponding boundary nodes to apply the MPC boundary conditions and modify the input file of the microscopic model.

The file **EUC\_PYTHON\_1.py** initiates a computation process for the **Macro\_new.inp** model, utilizing **EUC\_UMAT.for**. The **EUC\_UMAT.for** is a user-defined subroutine within the Abaqus finite element software, specifically designed to calculate the strain-stress response. This calculation is a result of the concurrent processing of micro and macro models. The subroutine **EUC\_UMAT.for** plays a crucial role in facilitating the transfer of strain and stress information between these Microscopic and Macroscopic Models. This process is directed through an interface with another Python script, namely **EUC\_PYTHON\_2.py**, which is accessed via the UMAT subroutine. The **EUC\_PYTHON\_2.py** is responsible for preparing the required input parameters and submitting the job for the evaluation of Microscopic model and the Multi-Point Constraints (MPC) code.

This occurs concurrently with the running and calculation of the Macroscopic model. The file **3\_EUC\_Micro.inp** represents an EUC model with a micro crack. This model is loaded by strain derived from the Macro model, and in return, it provides the output of the microscopic stress response.

The calculation process described above is initiated with the Abaqus software, a specialized Finite Element Analysis (FEA) tool. Here's a detailed breakdown of the workflow:

1. **Model construction:** Construction of both macroscopic and microscopic models according to the relevant application or benchmark case.
2. **Boundary Nodes Finder:** Processing the external app that analyzes the microscopic model to identify the corresponding nodes in order to apply special boundary conditions.
3. **Abaqus Software Launch**: The process begins by launching the Abaqus software. Abaqus is known for its robust capabilities in engineering simulations, particularly in handling complex models and computation-intensive tasks.
4. **Script Execution**: Once Abaqus is active, the user navigates to the 'Run Script' option within the software interface. This feature in Abaqus allows users to execute custom Python scripts which can automate or perform specific tasks that are not readily available through the standard Abaqus toolset.
5. **Running the main script**: Through the 'Run Script' function, **EUC\_PYTHON\_1.py** is executed. This script, as previously discussed, handles the computation of equivalent material properties from microscopic EUC models and updates the macroscopic INP file model with these new properties in user-defined elements. This is a top level step in the process, as it sets the stage for subsequent calculations involving both micro and macro-level models and interactions.

## List of Code Files:

* EUC\_PYTHON\_1.py
* EUC\_PYTHON\_2.py
* 1\_EUC\_alfa11.inp
* 1\_EUC\_alfa12.inp
* 1\_EUC\_alfa22.inp
* 3\_EUC\_Micro.inp
* Macro.inp
* EUC\_MPC.for
* EUC\_UMAT.for
* Labview app zipped in archive

## Repository link

<https://github.com/marinagrig/EUCcrack.git>

