1. Install calculix, gfortran (for UMAT) and python 3 with numpy and scipy module;

2. Generate microstructure using matlab
   - Open generate_synthetic_microstructures.m in the folder “Microstructure_Generation_Code”
   - Input specimen size, fiber radius, fiber volume fraction in generate_synthetic_microstructures.m ;
   - Run generate_synthetic_microstructures.m and it will automatically generate the folder “fibercentroid_r***x***vf***y***domain***z”***
     - x,y,z respectively stand for the value of fiber radius, volume fraction and domain size; The output information like fiber centroid coordinates, amount of fibers are stored in fibercentroid_info.dat and fibercoord.dat within the folder;

3. Generate inp. file for calculix solver
   - Copy the folder “fibercentroid_r***x***vf***y***domain***z”*** to the folder “abaquscae_pythonscripting”;
   - Run casefiles_edit.py and generate composites_cae_forrun.py
   - Open abaqus cae module:
     1) For creating geometry, copy the scripts in composites_cae_forrun.py from the first line to the line “del modelObject.sketches['__profile__’]” to Kernel Command Line Interface in abaqus cae module and run.
     2) Generate the mesh.
     3) For creating the sets of fiber centroid nodes, copy the scripts from the line “allNodes = mdb.models[model].parts[part].nodes” to the last line in composites_cae_forrun.py;
     4) Create the element sets for matrix, fiber; Create node sets for boundaries;
     5) The inp. File format should be modified for calculix usage; The format can be referenced based on the existing inp. file;

3. Run the reference simulation using calculix

- Select a home directory for optimization. Create the folder jobname_results ("jobname" is the name of the case) and subdirectory "jobname_results/ref" and "jobname_results/opt" in the home directory.
  - "jobname_results/ref" will store the measurement data, which in this case is generated by numerical simulation. "jobname_results/opt" stores the information for optimization;

- Open fem2Dmechload_expmod.py and input the parameters;
  - Input jobname, solver_name, solver_directory1, parent_dir.
    - The whole solver directory for executing calculix program is "solver_directory1 + solver_name + solver_directory2". solver_directory1 is the home directory of calculix solver. solver_directory2 is fixed. Since calculix solver cannot deal with multiple UMAT tasks simultaneously, different copies of calculix solver (contained in the folder named "solver_name") are made in the folder solver_directory1.
    - parent_dir is the home directory where reference simulation and optimization locate;
  - Input measind1, measind2 (measurement number from measind1 to measind2), errorbnd (measurement error bound), Em_bar, Em_inter, alpha,num(reference material parameters) and fiber_centroid_number (number of fiber centroid , extracted from number of nodes in the nodes set fibercentroid in inp. file)

- Copy the folder "source" (containing umat file), compile.sh (employed for compiling UMAT for calculix), initial.sh (for submitting job to the cluster), fibercoord.dat and inp. file in the folder "fibercentroid_r***x***vf***y***domain***z***" to the home directory;

- Run fem2Dmechload_expmod.py
  - Reference reaction force files (reactionforce_e***x***_m***y***.dat), fiber centroid displacement files (fiber_centroid_disp_e***x***_m***y***.dat), error_reactionforce_e***x***_m***y***.dat, error_ux_e***x***_m***y***.dat, error_uy_e***x***_m***y***.dat stand for exact value of error added to the reference reaction force, ux and uy;
  - x stands for the error bound (x=1 for 1% errorbound), y stands for sequence number of measurement data (y=30 for measurement no. 37);

4. Execute optimization using scipy;

- Make multiple copies of calculix solver with the name "Calculix_***X***" in the directory solver_directory1, where ***X*** values from 1 to the total number of copies;

- Open initializationsetting_rferr.py and input parameters:
  - job_name, solver_directory1, parent_dir, errorbnd, measind1, measind2, fiber_centroid_number;
  - level (total levels of stratified sampling for initialization, One set of optimization is executed using multiple initialization, which is set using stratified sampling);
  - num_cpu (total number of cpus employed for the same set of optimizations with multiple initializations);

- Run initializationsetting_rferr.py and create all the files for optimization settings;
  - The optimization setting files, including the bash files (initial.sh) for submitting the jobs in the cluster are stored under the directory e***x***_cpu***y***, where ***x*** stands for error bound, y stands for sequence number of cpu;

- Run submit.sh to submit the jobs, the results will be stored in the directory "jobname_results/opt";
  - The results file name format is objfunc_e***x***Ei***j***a***k***num***l***.dat, where ***x*** stands for error bound, j, k, l stand for number of stratified sampling level;

- Running errordetect.py will detect whether there is error information (stored in the file e***x***_cpu***y***.e* under the directory e***x***_cpu***y***) occurring during the optimization;

- Run plot_optdata.m will plot the final results;

- Note:
  - collectoptinfo.sh can copy all the cluster job information files e***x***_cpu***y***.e* and e***x***_cpu***y***.o* to one place;
  - jobdeletion.sh can delete all of the cluster jobs;
  - Running errordetect.py will show the optimization which exceeds the maximum iteration step. Those optimizations needs to be run again. Running initializationsetting_supplementary.py will create the initialization for those optimizations.
  - initializationsetting.py and optmechload_expmod.py are for the optimizations in which the reaction force has no measurement error;
  - The final optimization results are stored in the directory “optresults_sample***x***_microns***y***_vf***z***_xxx”; The numerical reference measurement data are stored in microns***y***vf***z***_results, where x stands for total amount of samples, y stands for specimen size, z stands for fiber volume fraction;
  - initializationsetting*.py edit the information of line 107 to 120 in optmechload_expmod*.py, line 15 in compile.sh , line 56 in source/umat.f90 and initial.sh. Please be careful about the corresponding modifications of line number if there is any modifications in those files