SMAT USER MANUAL

Version 0.9.0

By
Shunlai Zang

XI'AN JIAOTONG UNIVERSITY

April 05, 2014
Shaanxi, CHINA

Contents

1 Preface

What is SMAT? SMAT (**S**imulate **MAT**erials) is a Matlab toolbox for parameter identification of constitutive models, which containing the following features: i) Identify the material parameters based on inverse method (Currently, SMAT only supports Abaqus software for the simulation of mechanical behavior); ii) Create publication ready plots.

Contact Information:

Dr. Shun-lai ZANG
School of Mechanical Engineering
Xi'an Jiaotong University
No.28, Xianning West Road, Xi'an, Shaanxi, 710049, P.R. China
E-mail: shawn@mail.xjtu.edu.cn; shunlai.zang@gmail.com
Homepage: http://shawn.gr.xjtu.edu.cn/

2 Installation and startup

2.1 System requirements

The system requirements for the running of SMAT toolbox are listed as below.

- Linux operating systems, for example, OpenSUSE, Fedora, Ubuntu, etc.
- Abaqus commercial software. If the default command is not *abaqus*, the command variable in the global setting of SMAT toolbox should be changed to the suitable values. The global setting file, smat_setting.m, locates in the root directory of SMAT toolbox. The variables' name is ABACMD.
- Matlab commercial software with version larger than 7.5. If the default command is not *matlab*, the command variable in the global setting of SMAT toolbox should be changed to the suitable values. The global setting file, smat_setting.m, locates in the root directory of SMAT toolbox. The variables' name is MATCMD.
- Intel Fortran compiler has been successfully configured in order to call the user subroutines in command line, as shown in Fig. 1.

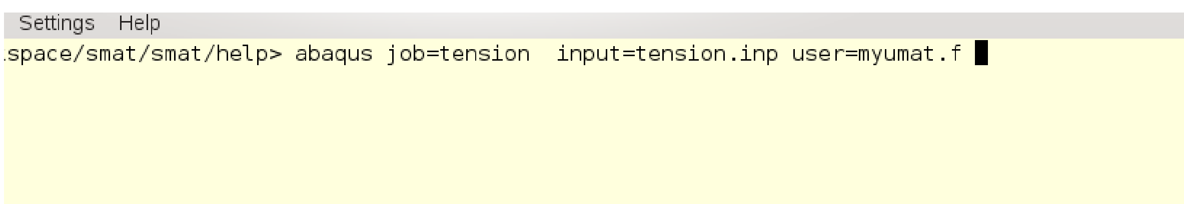

Figure 1: Submitting an abaqus job in command line on linux system.

2.2 Installation

The installation on Linux system is,

- Extract the smat from the smat.tar.gz package
- Move smat directory to Matlab_RootDirectory/toolbox/
- Backup /toolbox/local/startup.m if this file exists
- Move /toolbox/smat/startup.m to /toolbox/local/

2.3 Startup

To start an optimization task, the following steps should be performed.

- Copy the optimization task to any directory with a writable permission.
- Start Matlab, you will see a hint, which is *startup_smat*. Then enter *startup_smat* in the command windows of Matlab software. After this, the toolbox will be loaded. Please note that you need to load the toolbox at each time of starting Matlab.
- Change the current directory to the optimization task directory in Matlab software. The task directory includes all files and settings related to user's optimization. For the details on optimization task, please refer the Section: Files structure of optimization task in this manual file.
- Run the optimization task by entering command $smat\ *.config$ in Matlab windows.

3 Logical structure of SMAT toolbox

In solid mechanics, the inverse identification of material parameters is usually required when the constitutive model and experimental tests are complicated. For the SMAT toolbox, a cost function $\mathscr{L}(\boldsymbol{A})$ is defined in the least square sense by Eq.(1), and starting from an initial guess of material parameters $\boldsymbol{A}_0$ in order to optimize the material parameters of constitutive model. The cost function is minimized with a Levenberg-Marquardt algorithm.

$$\mathscr{L}(\boldsymbol{A}) = \sum_{n=1}^{N} \mathscr{L}_n(\boldsymbol{A})$$

with N the number of tests in the database. For each test, the gap between experiments and model is given by

$$\mathscr{L}_n(\boldsymbol{A}) = \frac{1}{M_n} \sum_{i=1}^{M_n} \left(D_n \left| Z(\boldsymbol{A}, t_i) - Z^*(t_i) \right|^2 \right) \tag{1}$$

where M_n is the number of experimental points of the n-th test, $Z(\boldsymbol{A}, t_i) - Z^*(t_i)$ is the normalized gap between experimental $Z(\boldsymbol{A}, t_i)$ and simulated results $Z^*(t_i)$ at time t_i, and D_n is a weighting coefficient for the n-th test.

4 Files structure of optimization task

For an optimization task, the SMAT toolbox needs three kinds of files which are categorized as shown in Fig. 2.

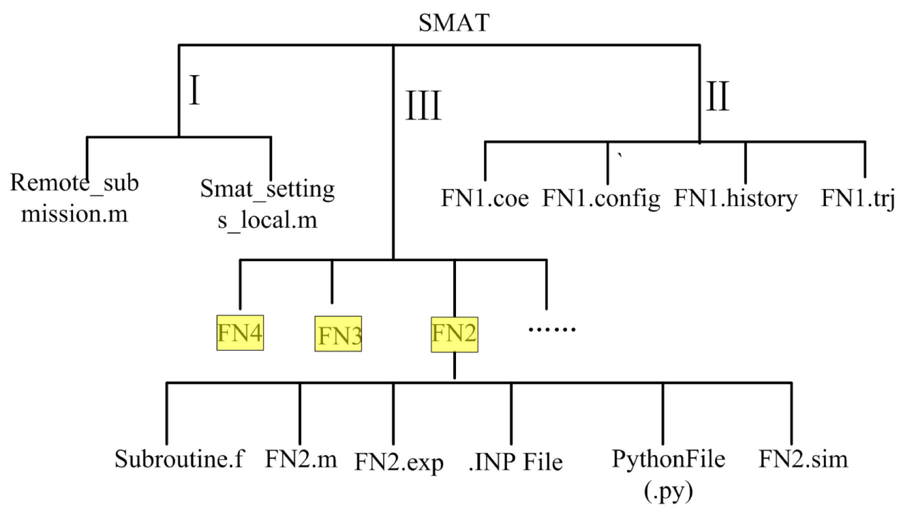

Figure 2: Files structure of optimization task.

For the type I files, there are two files named as remote_submission.m and smat_settings_local.m. 1) For the remote_submission.m file, it is usually required when the user tries to submit the optimization task without Matlab graphics interface. This is the case that you might want to connect a remote linux server where the SMAT toolbox has been installed by SSH secure connection. If it is not your case, please ignore it. 2) For the smat_settings_local.m file, it is used to modify the values of setting variables in the SMAT toolbox, which will be detailed explained later.

For the type II files, there are four files named as FN1.coe, FN1.config, FN1.history and FN1.trj, where the latter two files are automatically generated by the SMAT toolbox. 1) For the FN1.coe file, it includes the initial and current values of material coefficients. 2) For the FN1.config file, it is the configuration file for an optimization task. 3) For the FN1.history file, this is just the log file. 4) For the FN1.trj file, it includes the total error norm, i.e. the total cost function for each iteration.

For the type III files, actually, they are stored in one folder. This folder can be called as the optimization job, for example, the T0 folder as shown in Fig. 3. The optimization job includes one or several curves which are obtained from one test, for instance, longitudinal strain vs. Cauchy stress and longitudinal strain vs. transverse strain from uni-axial tension at 0° to rolling direction.

HINTS: What is the difference between optimization task and job in SMAT toolbox? One task can include many jobs. One job represents one kind of test. For example, the user might

want to optimize the material parameters of constitutive model from the true stress vs. strain curves of uni-axial tension at 0°, 45° and 90° to the rolling direction of metallic sheets. Here we call this work is optimization task, each uni-axial test is called optimization job.

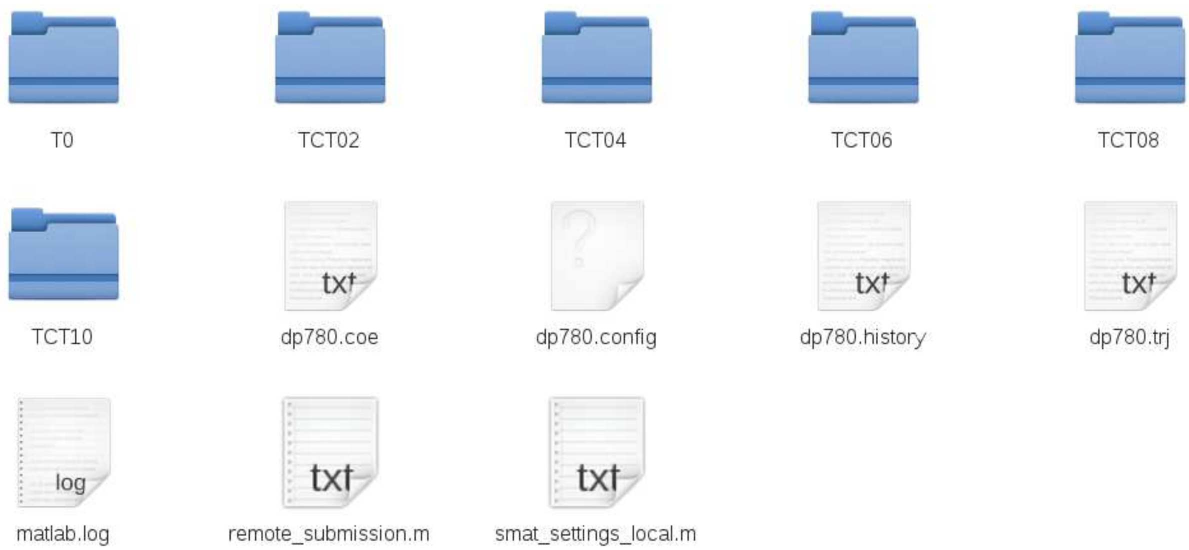

Figure 3: An example of optimization task.

4.1 Remote_submission.m

Usually, we have to keep the Matlab windows open when an optimization task is running, however the user might want to submit the optimization task from a remote computer without Matlab graphics interface. For instance, login on the remote server by a SSH secure connection. And after the submission of optimization task, the user wants to login off the server. With this remote_submission.m file, it can be achieved.

```
%% Running smat without output for remote submission.
startup_smat
smat dp780
```

Figure 4: An example of remote_submission file.

Usually, there are two command lines in this matlab file. The first one is to load the SMAT toolbox in Matlab software, which is reserved and should never be modified. The second line is to start the optimization task. For the task name (for example, dp780 or dp780.config), it can be changed to the corresponding one based on user's optimization.

To submit the optimization task without Matlab graphics interface, the following linux command should be used as shown in Fig. 5. The option of -nodisplay will start the Matlab software without graphics interface.

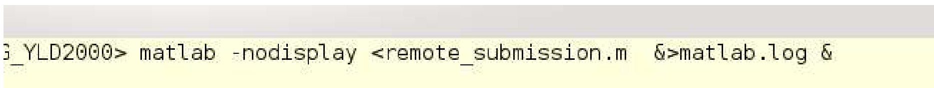

Figure 5: The command for the remote submission.

4.2 Smat_settings_local.m

The smat_setting_local.m is a copy of global setting file (smat_settings.m) which locates in the root directory of SMAT toolbox. Fig. 6 is an example of this local setting file. Here, several variables can be redefined.

4.2.1 Variable ABACMD

This variable is used to define the default name of abaqus command. For some users, there might be several abaqus software with different version. In such a case, it will be easy to call different codes by redefining this variable. The default value is **abaqus**.

4.2.2 Variable MATCMD

Same as the variable of ABACMD. The default value is **matlab**.

4.2.3 Variable FIGURE_DISPLAY

This variable is used to turn on or off the figure plotting function during optimization. The default value is **on**.

4.2.4 Variable LEGEND_DISPLAY

This variable is used to turn on or off the legend display during figure plotting. The default value is **on**.

4.2.5 Variable PLOT_LINEWIDTH

This variable is used to set the line width of figures.

4.2.6 Variable DEF_FONTSIZE

This variable is used to set the font size of figures.

```
%% Section misc
%# set default result of checking whether matlab display is on
%# if display is off, the dynamic progress bar is disabled
global MATLAB_DISPLAY_FLAG
MATLAB_DISPLAY_FLAG  = isdisplay();

%% Section for abaqus numerical simulation.
%# set default value for pause interval in function progress_bar
global PROGRESS_PAUSE_TIME
PROGRESS_PAUSE_TIME  = 2.0;

%# set default abaqus command
global ABACMD
ABACMD = 'abaqus';

%# set default matlab command
global MATCMD
MATCMD = 'matlab';

%# set default maximum waiting time for a simulation
global MAX_SIMULATION_TIME
MAX_SIMULATION_TIME = 1000;

%# set default waiting time after submit an abaqus job
%# before to check whether simulation is over!
global SUBMIT_WAITING_TIME
SUBMIT_WAITING_TIME = 5;

%% Section for figure plot
%# set whether display figures during optimisation
%# default is on
global FIGURE_DISPLAY
FIGURE_DISPLAY = 'on';

%# set whether display legend of figures
%# default is on
global LEGEND_DISPLAY
LEGEND_DISPLAY = 'on';

%# set default value for linewidth
global PLOT_LINEWIDTH
PLOT_LINEWIDTH = 1.8;

%# set default value for fontsize
global DEF_FONTSIZE
DEF_FONTSIZE = 22;

%# set default value for markersize
global DEF_MARKERSIZE
DEF_MARKERSIZE = 8.0;

%% Section for comparison of curves
%# set default x-axis name between experimental and simulated curves
global DEF_X_AXIS
DEF_X_AXIS = 'le11';

%# set default x-axis name for plot
global DEF_X_AXIS_PLOT
DEF_X_AXIS_PLOT = 'le11';
```

Figure 6: An example of smat_setting_local.m file.

4.2.7 Variable DEF_MARKERSIZE

This variable is used to set the mark size of figures.

4.2.8 Variable DEF_X_AXIS

This variable is used to set the column name of x-axis for the linear interpolation. The detailed explanation can be found in the Section: Logical structure of SMAT toolbox. The default value is **le11**. The x-axis name for linear interpolation can also be redefined by the keyword *comparison.

4.2.9 Variable DEF_AXIS_PLOT

This variable is used to set the x-axis name for figure plotting. The detailed explanation can be found in the Section: Logical structure of SMAT toolbox. The default value is **le11**. The x-axis name for figure plotting can also be redefined by the keyword *comparison.

4.2.10 Variable DEF_RESTART_FLAG

This variable is used to set the restarting flag. For large computation cost, if the optimization is interrupted by user, we have to recalculate the gradient. It is time consuming. With the help of this variable, you can recovery the error vector from previous optimization. The string ′on′ means this variable is activated, ′off′ is disabled.

4.2.11 Variable DEF_RESTART_FOLDERNAME

This variable is used to set the folder name which is related to Variable DEF_RESTART_FLAG. The history error vectors will be stored in this folder.

4.3 Coefficients file *.coe

This file includes the name and value pairs of all the material parameters in user subroutine code (UMAT or vumat), see Fig. 7 as an example. Since these names will be used in the *.config file too, they should be simple and short, especially no blank and brackets are included.

Note that the number of parameters should be equal to the value of variable CONSTANT in the material section (*USER MATERIAL,CONSTANT=X) of abaqus inp file no matter whether this parameter will be optimized, see Fig. 8 as an example. At the beginning of optimization, the initial material parameters will be given in the *.coe file, and then these values will be updated during optimization.

```
E                1.988000e+05
v                3.000000e-01
sy0              5.270000e+02
C1               2.489443e+04
gm1              6.505208e+01
C2               1.282663e-02
K                4.220000e+02
n                2.800000e+01
a1               9.276000e-01
a2               1.024300e+00
a3               9.622000e-01
a4               9.880000e-01
a5               1.004300e+00
a6               9.165000e-01
a7               1.004300e+00
a8               1.032400e+00
em               6.000000e+00
```

Figure 7: An example of coefficients file.

```
**
** MATERIALS
**
*MATERIAL, NAME=MAT
*USER MATERIAL,CONSTANT=17
**+++++++++++++++++++++++++++++++++
**
**E, v, sy0, C1, gm1, C2, ek, en,
**a1, a2, a3, a4, a5, a6, a7, a8,
**em
**+++++++++++++++++++++++++++++++++
**
1.988e+05,3.0e-01,5.27e+02,2.435e+04,6.4e+01,4.48e-05,4.22e+02,2.8e+01,
9.2e-01,1.02e+00,9.62e-01,9.88e-01,1.0e+00,9.165e-01,1.0e+00,1.0e+00,
6.0,
*DEPVAR
18
```

Figure 8: An example of material section in abaqus inp file.

4.4 Configuration file *.config

It is the most important file for an optimization task, which includes several keywords started with double asterisks ** or single asterisk *. The notion /* means that this line is comment one. Fig. 9 is an example of configuration file, the functions and usages of these keywords are explained as below.

```
/*============================
**definition
*source .
/*============================
**coefficient file
*coefficient  dp780.coe
/*============================
**run T0
**run TCT02
**run TCT04
**run TCT06
**run TCT08
**run TCT10
/*============================
*parameters
/*sy0
C1
gm1
C2
/*K
/*n
/*============================
*constraint
/*sy0   min   0.0
C1      min   0.0
gm1     min   0.0
C2      min   0.0
/*K     min   0.0
/*n     min   0.0
/*============================
*comparison T0
  sig11 1.0
*comparison TCT02
  sig11 1.0  time  le11
*comparison TCT02  TCT02a
  sig11 0.5  time  le11
*comparison TCT04
  sig11 1.0  time  le11
*comparison TCT04  TCT04a
  sig11 0.5  time  le11
*comparison TCT04  TCT04b
  sig11 0.8  time  le11
*comparison TCT06
  sig11 1.0  time  le11
*comparison TCT06  TCT06a
  sig11 0.5  time  le11
*comparison TCT06  TCT06b
  sig11 0.8  time  le11
*comparison TCT08
  sig11 1.0  time  le11
*comparison TCT08  TCT08a
  sig11 0.5  time  le11
*comparison TCT08  TCT08b
  sig11 0.8  time  le11
*comparison TCT10
  sig11 1.0  time  le11
*comparison TCT10  TCT10a
  sig11 0.5  time  le11
*comparison TCT10  TCT10b
  sig11 0.8  time  le11
/*============================
**optimisation local file
*optimisation  optimisation_local.m
/*============================
**setting local file
*setting  smat_settings_local.m
**return
```

Figure 9: An example of configuration file.

4.4.1 Keyword *coefficient

This keyword is used to define the name of material coefficient file. For the Fig. 9, the coefficients file name is **dc04.coe**.

4.4.2 Keyword **run

This keyword is used to define which experimental test is adopted to optimize the material parameters. For the Fig. 9, the experimental data obtained from uni-axial tensile tests at 0°, 45° and 90° to the rolling direction are all used to calibrate the material parameters. If some lines are commented by notion /*, it means that these experimental tests will not be used in the optimization.

4.4.3 Keyword *parameters

This keyword is used to define which parameter will be optimized. The names of these parameters should be consistent with those in coefficients file *.coe. If the parameters started with notion /*, it means that these parameters will not be optimized.

4.4.4 Keyword *constraint

This keyword is used to define the bounds of material parameters during optimization. The names of these parameters should be consistent with those in coefficients file *.coe. If the parameters started with notion /* or weren't appeared in this section, it means that the default bounds will be used. The default lower (MINBOUND) and upper (MAXBOUND) bounds can be found in the global optimization setting file in the root directory of SMAT toolbox, i.e. optimisation_global.m.

Here, three methods can be used to modify the default bounds.

- Redefine the lower bound with the notion min, and the default upper bound is used.
- Redefine the upper bound with the notion max, and the default lower bound is used.
- Redefine the lower and upper bounds with 0.0 100.0.

4.4.5 Keyword *comparison

The keyword **comparison* is used to define which experimental and simulated data will be compared. The default setting is,

**comparison T0*

sig11 1.0

le22 1.0e5

where *T0* is the optimization job name, *sig11* and *le22* are the columns to be compared, *1.0* and *1.0e5* are their weights. Since the experimental and simulated file names are same as the optimization job name. It means that there should be two files, *T0.exp* and *T0.sim*, in the *T0* directory. When the numerical simulation is completed, SMAT will open these two files and find the corresponding columns in order to calculate their difference.

If more than one experimental or simulated file needs to be compared in one optimization job, these experimental and simulated file names should be explicitly given as,

**comparison T0 expfile1 simfile1*

sig11 1.0

where *T0* is the file name of the optimization job, *expfile1* is the experimental filename, *simfile1* the simulated filename. In the above definition, *expfile1.exp* and *simfile1.sim* will be opened, and the column of *sig11* is compared. The reason to design such a function is that, for instance, material parameters of the combined isotropic-kinematic hardening model is usually calibrated from the cyclic loading curves. In such case, the range of transient behavior is quite small comparing with that of monotonic loading. But the material parameters of kinematic hardening model is mainly determined by the experimental data in range of transient behavior. To enlarge the contribution of the errors in range of transient behavior to the total cost function, a higher weight need to be defined for this region. Then the whole experimental cyclic loading curve can be divided into several parts, for example, loading, unloading (including transient behavior region) and reverse loading curves, named as *exploading*, *expunloading* and *expreloading*. If the job directory is *TC*, for the above case, the configuration should be like,

**comparison TC*

sig11 0.0

**comparison TC exploading TC*

sig11 1.0

**comparison TC expunloading TC*

sig11 2.0

**comparison TC expreloading TC*

sig11 1.0

where the last term for each comparison line after the second one, *TC*, means that the same simulated file is used for the interpolation. For the first comparison, it does not contribute to final cost function due to zero weight. It just is used for figure plotting.

In the option of keyword **comparison*, the xaxis for interpolation or figure plotting is usually neglected. But in some case, the xaxis needs to be explicitly defined. For instance, for cyclic loading stress-strain curve, there might be two stress values at the same strain. Then it will result in mathematical difficulty to interpolate such data. To overcome it, a time column can be introduced in experimental and simulated results. Then, for the time-stress curve, only one stress can be found at certain time. However, for the convenience of plotting figure, strain is still considered as x-axis. In the SMAT toolbox, the users can define these two xaxis by

**comparison T0*

sig11 1.0 xaxis-int xaxis-plot

where *xaxis-int* is used to define xaxis for linear interpolation, *xaxis-plot* the xaxis for figure plotting. The default settings for these two variables can be found in the global or local setting file, smat_settings.m or smat_settings_local.m.

4.4.6 Keyword *optimisation

This keyword is used to redefine the settings for the Matlab optimization function. The local optimization file name should be followed with this keyword. If the file does not exist in the optimization task folder, the default settings will be used.

The local optimization file is a copy of the global optimization setting file in the root directory of SMAT toolbox, i.e. optimisation_global.m. For the meaning of each variable, the user can refer the Matlab function lsqnonlin().

WARNING: Be sure what are you doing if you want to modify the default optimization settings. Unsuitable values might cause the optimization failed.

```
%# set default bound for constraint in function constraint_bound
global MINBOUND
global MAXBOUND

MINBOUND = -1.0e8;
MAXBOUND =  1.0e8;

%# set default TolFun
global TolFun
TolFun = 0.001;

%# set default TolX
global TolX
TolX = 1.0e-7;

%# set default MaxIter
global MaxIter
MaxIter = 200;

%# set default DiffMinChange
global DiffMinChange
DiffMinChange = 1.0e-3;
```

Figure 10: Smat global optimisation file optimisation_global.m.

4.4.7 Keyword *setting

This keyword is used to redefine the settings for the SMAT toolbox. The local settings file name should be followed with this keyword. If the file does not exist in the optimization task folder, the default settings will be used. For the details, please refer the previous explanation on smat_settings_local.m.

4.4.8 Keyword **return

If the keyword **run is followed by **return as shown in Fig. 11, it means that all the optimization jobs will be performed only one time. Usually, this keyword should be put at the end of the *.config file.

```
/*===========================
**run T0
**run TCT02
**run TCT04
**run TCT06
**run TCT08
**run TCT10
**return
/*===========================
```

Figure 11: The position of keyword **return.

4.5 Trajectory file *.trj

This file includes the total error norm (cost function value) during optimization, as shown in Fig. 12 as an example, where the first column is the iteration number.

```
0           5.391657e-07
1           5.391657e-07
2           9.746533e-02
3           1.040129e-01
4           1.502105e-02
5           1.591344e+02
6           2.474205e+01
7           2.371726e+01
8           5.391657e-07
9           9.746533e-02
10          1.040129e-01
11          1.502105e-02
12          1.591344e+02
13          2.474205e+01
14          2.371726e+01
15          5.391657e-07
16          9.746533e-02
```

Figure 12: An example of trajectory file.

4.6 Log file *.history

It is the log file for the SMAT toolbox, as shown in Fig. 13 as an example. Here, the iteration number, current values and variation percentage of material parameters, time cost for each optimization job, error norm for each curve and the total error norm are respectively saved.

```
ITERATION: 2

    New Material Parameters:

      Param        CurValue            Variation [%]
        E          [198800.00000]            [       0.00000]
        v          [       0.30000]          [       0.00000]
        sy0        [     527.00000]          [       0.00000]
        C1         [  24919.32443]           [       0.10000]
        gm1        [      65.05208]          [       0.00000]
        C2         [       0.01283]          [       0.00000]
        K          [     422.00000]          [       0.00000]
        n          [      28.00000]          [       0.00000]
        a1         [       0.92760]          [       0.00000]
        a2         [       1.02430]          [       0.00000]
        a3         [       0.96220]          [       0.00000]
        a4         [       0.98800]          [       0.00000]
        a5         [       1.00430]          [       0.00000]
        a6         [       0.91650]          [       0.00000]
        a7         [       1.00430]          [       0.00000]
        a8         [       1.03240]          [       0.00000]
        em         [       6.00000]          [       0.00000]

    T0 tension00 is completed!        14.9598 s
    TCT02 tension00 is completed!        20.9793 s
    TCT04 tension00 is completed!        21.0358 s
    TCT06 tension00 is completed!        20.9248 s
    TCT08 tension00 is completed!        20.8933 s
    TCT10 tension00 is completed!        20.9061 s

    Each_Error_Norm =

        Task            Exp          CompVar         ErrNorm
        T0              T0           sig11       [    175.38628]
        TCT02           TCT02        sig11       [   3169.54898]
        TCT02           TCT02a       sig11       [   1364.36527]
        TCT04           TCT04        sig11       [   1379.63898]
        TCT04           TCT04a       sig11       [    681.32418]
        TCT04           TCT04b       sig11       [   1072.06862]
        TCT06           TCT06        sig11       [   2051.66453]
        TCT06           TCT06a       sig11       [    997.89994]
        TCT06           TCT06b       sig11       [   1976.18260]
        TCT08           TCT08        sig11       [   1882.74930]
        TCT08           TCT08a       sig11       [   1246.41600]
        TCT08           TCT08b       sig11       [   1668.59915]
        TCT10           TCT10        sig11       [   1599.01268]
        TCT10           TCT10a       sig11       [   1139.63287]
        TCT10           TCT10b       sig11       [   1598.90547]


    Total_Error_Norm = 22003.3948
```

Figure 13: An example of log file.

4.7 Optimization job files

For each optimization job, the related files are organized as one folder. Fig. 14 is an example of optimization job, which is the uni-axial tensile test at 0° to the rolling direction. Here, it should note that the folder name of optimization job should be same as the value after keyword **run presented in the *.config file. For the Fig. 14, the name of optimization job is T0. In each folder of optimization job, three important files are required, experimental result file *.exp, simulated result file *.sim, and job configuration file *.m. For these files, their names should be same as that of optimization job, for the current example, they are T0.

Figure 14: Files structure of optimization job.

4.7.1 Experimental data file *.exp

This file includes all the experimental data from the test, which is stored in ASCII format. Fig. 15 is an example of experimental file, which is the uni-axial tensile test at 0° to the rolling direction. Here, the first line is the column name, and the others are data lines.

WARNING: This experimental file name should be strictly same as the folder name of the optimization job.

time	disp	ee11	esig11	le11	sig11
0.200000	0.002000	0.004000	506.350000	0.003992	508.375400
0.300000	0.003000	0.006000	558.410000	0.005982	561.760460
0.400000	0.004000	0.008000	588.660000	0.007968	593.369280
0.500000	0.005000	0.010000	610.850000	0.009950	616.958500
0.600000	0.006000	0.012000	630.980000	0.011929	638.551760
0.700000	0.007000	0.014000	648.750000	0.013903	657.832500
0.800000	0.008000	0.016000	664.370000	0.015873	674.999920
0.900000	0.009000	0.018000	678.110000	0.017840	690.315980
1.000000	0.010000	0.020000	690.310000	0.019803	704.116200
1.100000	0.011000	0.022000	701.170000	0.021761	716.595740
1.200000	0.012000	0.024000	710.930000	0.023717	727.992320
1.300000	0.013000	0.026000	719.710000	0.025668	738.422460
1.400000	0.014000	0.028000	727.750000	0.027615	748.127000
1.500000	0.015000	0.030000	735.190000	0.029559	757.245700
1.600000	0.016000	0.032000	742.060000	0.031499	765.805920
1.700000	0.017000	0.034000	748.380000	0.033435	773.824920
1.800000	0.018000	0.036000	754.160000	0.035367	781.309760
1.900000	0.019000	0.038000	759.450000	0.037296	788.309100
2.000000	0.020000	0.040000	764.230000	0.039221	794.799200
2.100000	0.021000	0.042000	768.760000	0.041142	801.047920
2.200000	0.022000	0.044000	773.100000	0.043059	807.116400
2.300000	0.023000	0.046000	777.160000	0.044973	812.909360

Figure 15: An example of experimental file for an optimization job.

4.7.2 Configuration file *.m

This file is the configuration file for the optimization job, which includes all the commands what we need to obtain the simulated data file *.sim.

Usually, there are two abaqus commands in this file. The first command is used to perform the abaqus numerical simulation. The second one is to used to extract the simulated results from the odb file by python script, as shown in Fig. 16. After these two commands are executed, the simulated data file *.sim should be generated.

WARNING: This file name should be strictly same as the folder name of the optimization job.

```
abaqus  job=tension00  input=tension00.inp user=E+ANK_Zang_Barlat2000_2D_PS_final.f
abaqus  python tension00.py
```

Figure 16: An example of Matlab configuration file for an optimization job.

4.7.3 Simulated data file *.sim

This file includes all the simulated data what we need to compare with experimental results. Similarly as that of experimental file *.exp, the first line is the column name. It should note that these column names should be consistent with that of experimental file in order to calculate their error norms.

WARNING: This simulated file name should be strictly same as the folder name of the optimization job.

sig11	sig22	sig33	sig12	le11
0.0000000e+00	0.0000000e+00	0.0000000e+00	0.0000000e+00	0.0000000e+00
3.9720294e+02	-1.9900477e-07	0.0000000e+00	0.0000000e+00	1.9980029e-03
5.4075848e+02	3.2304227e-01	0.0000000e+00	0.0000000e+00	3.9920211e-03
5.5967749e+02	6.3880324e-02	0.0000000e+00	0.0000000e+00	5.9820721e-03
5.7804700e+02	5.7361645e-05	0.0000000e+00	0.0000000e+00	7.9681696e-03
5.9579034e+02	4.2835170e-05	0.0000000e+00	0.0000000e+00	9.9503305e-03
6.1286865e+02	3.1846404e-05	0.0000000e+00	0.0000000e+00	1.1928570e-02
6.2927716e+02	3.1362488e-05	0.0000000e+00	0.0000000e+00	1.3902905e-02
6.4501691e+02	3.0995685e-05	0.0000000e+00	0.0000000e+00	1.5873348e-02
6.6009369e+02	3.0736996e-05	0.0000000e+00	0.0000000e+00	1.7839920e-02
6.7451697e+02	3.0571278e-05	0.0000000e+00	0.0000000e+00	1.9802626e-02
6.8829926e+02	3.0485804e-05	0.0000000e+00	9.4904748e-15	2.1761492e-02
7.0145557e+02	3.0468700e-05	0.0000000e+00	0.0000000e+00	2.3716528e-02
7.1400281e+02	3.0508842e-05	0.0000000e+00	0.0000000e+00	2.5667747e-02
7.2595935e+02	3.0595886e-05	0.0000000e+00	0.0000000e+00	2.7615167e-02
7.3734448e+02	3.0720450e-05	0.0000000e+00	0.0000000e+00	2.9558802e-02

Figure 17: An example of simulated file for an optimization job.

5 Frequently Asked Questions& Answers

5.1 What is the license of SMAT toolbox?

SMAT is not an open source software, but it might be distributed after an explicit permission from the author.

5.2 Is it possible to run SMAT toolbox on Windows platform?

Yes, it might be possible. But we have been debugging the toolbox on a Linux platform.

5.3 Is it possible to run SMAT toolbox on Scilab environment?

We haven't test it until now.