



CZECH TECHNICAL UNIVERSITY IN PRAGUE  
Faculty of Civil Engineering  
Department of mechanics  
Thákurova 7  
166 29 Praha 6

TF05000040-2019V010

---

Identification of material model parameters for  
concrete:  
Software Documentation

---

ELIŠKA JANOUCHOVÁ  
KAREL MIKEŠ  
ANNA KUČEROVÁ

# SOFTWARE MANUAL

ELIŠKA JANOUCHOVÁ, KAREL MIKEŠ, ANNA KUČEROVÁ

*Department of Mechanics, Faculty of Civil Engineering, CTU in Prague*

*email: eliska.janouchova@fsv.cvut.cz, karel.mikes.1@fsv.cvut.cz, Anna.Kucerova@cvut.cz*

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Methodology . . . . .	3
1.2	Prerequisites . . . . .	3
<b>2</b>	<b>Application</b>	<b>4</b>
2.1	Model parameters . . . . .	4
2.2	Results and software outputs . . . . .	5
<b>3</b>	<b>Functions</b>	<b>12</b>
3.1	Software for identification and sensitivity analysis . . . . .	13
3.2	mfiles . . . . .	15
3.3	oofem_data . . . . .	19
	<b>References</b>	<b>20</b>

# 1 Introduction

The presented manual describes the basic usage of the software for material model parameter identification delivered by the CTU team as a partial result of the project CeSTaR - Computer simulation and experimental validation - complex service for flexible and efficient design of pre-cast concrete columns with innovative multi-spiral reinforcement. There are two main MATLAB programs: IDENTIFICATION and SENSITIVITY\_ANALYSIS, which provides information about relations between inputs and outputs of the investigated material model for concrete and estimation of the model inputs' values together with the associated uncertainties for given experimental data.

## 1.1 Methodology

The theoretical methodology of the implemented parameter identification is based on the Bayesian inference [2]. This probabilistic approach enables to efficiently combine prior knowledge about the unknown values of parameters and experimental data in order to obtain estimation of parameters' values together with corresponding uncertainties. The inference provides the updated a posteriori parameter probability distribution for combination of uniform prior parameter distribution and likelihood function based on Gaussian distribution. However, the formulation of the identified distribution involves the finite element material model, which makes analytical work with the distribution impossible. In order to get samples from the defined distribution, the Metropolis algorithm is utilized to generate Markov chain with the corresponding stationary distribution [6]. The implemented sampling method is generally suitable for nonlinear models and its convergence is mathematically proved. The computational effort of the method is primarily given by computational demands of the model simulations.

The second part of this software package allows to investigate the relationship between the model inputs and model outputs. It provides some information about importance of particular observations or significance of each parameter for a system behaviour. In parameter identification, it enables to discover which parameters play an important role in the model response so they are identifiable from the corresponding experimental data. Specifically in the software, the global sensitivity analysis is based on evaluations of Spearman's rank correlation coefficient.

An illustrative example of the software application is presented in the next section. The software functions, the appropriate setting and inputs/outputs information are in Section 3.

## 1.2 Prerequisites

The software package delivered by the CTU team requires following external software:

- finite element package OOFEM v. 2.5 and above (<http://www.oofem.org/>),
- MATLAB v. 2019a (<https://www.mathworks.com/products/matlab.html>).

## 2 Application

In order to demonstrate the software in practise, the identification procedure was used for calibration of the Concrete damage plasticity model CDPM2 [3], which was previously calibrated in a deterministic way [1]. Since computational demands of the CDPM2 simulation for single finite element have been reduced, corresponding parameter identification can be performed in probabilistic setting for the full numerical model without any model response approximation.

### 2.1 Model parameters

The CDPM2 parameters are a priori considered as uniformly distributed. This prior distribution is uninformative and it is adequate in the case where not much information is available. Specifically, the unknown values of the CDPM2 parameters are a priori prescribed to be from the intervals given in Table 1 and determined by a combination of literature data and the previous results obtained by the

Parameter	Units	Minimum	Maximum
$ecc$	[-]	0.5	0.6
$k_{init}$	[-]	0.2	0.4
$A_{hard}$	[-]	$10^{-2}$	$10^0$
$B_{hard}$	[-]	$10^{-5}$	0.01
$C_{hard}$	[-]	1	10
$A_{soft}$	[-]	5	50
$del$	[-]	0.85	1.2
$H_p$	[-]	0	0.25
$efc$	[-]	$10^{-4}$	$10^{-3}$
$E$	[GPa]	28.5	31.5
$\nu$	[-]	0.17	0.22
$f_c$	[MPa]	42.5	52

Table 1: Bounds for single element model parameters.

deterministic optimisation, see [1]. In the current version, the total number of the parameters increase from 9 to 12, because the Young's modulus  $E$ , the Poisson ratio  $\nu$ , and compressive strength  $f_c$  are unfixed and subject to identification.

Since values of parameters  $A_{hard}$  and  $efc$  vary accros different orders, the parameters themselves are replaced by their logarithms, i.e., instead of  $A_{hard}$  and  $efc$ , we work with  $a = \log(A_{hard})$  and  $e = \log(efc)$ , where  $a$  is from  $\langle -2; 0 \rangle$  and  $e$  is from  $\langle -4; -3 \rangle$ . Furthermore, there is a relation between parameters  $A_{hard}$  and  $B_{hard}$  defined as  $B_{hard} = A_{hard}/b$ . In the identification procedure, we estimate coefficient  $b$  instead of model parameter  $B_{hard}$ . The possible values of  $b$  are on the interval  $\langle 4; 100 \rangle$ . The parameters to be identified together with their ranges of values are listed in Table 2.

Parameter	Units	Minimum	Maximum
$ecc$	[-]	0.5	0.6
$k_{init}$	[-]	0.2	0.4
$a$	[-]	-2	0
$b$	[-]	4	100
$C_{hard}$	[-]	1	10
$A_{soft}$	[-]	5	50
$del$	[-]	0.85	1.2
$H_p$	[-]	0	0.25
$e$	[-]	-4	-3
$E$	[GPa]	28.5	31.5
$\nu$	[-]	0.17	0.22
$f_c$	[MPa]	42.5	52

Table 2: Parameters to be identified with corresponding ranges of values.

In the Metropolis algorithm itself, all the parameters are linearly transformed into the interval  $\langle 0; 1 \rangle$  in order to get rid of the scales. Prior distribution is determined as uninformative multivariate uniform distribution of independent parameters on  $\langle 0; 1 \rangle$ .

## 2.2 Results and software outputs

The presented software has been verified for pseudo-experimental data and validated for experimental data available in [4]; namely samples with water-cement ration 0.55 measured on dried specimens with confinement stress level 2.15 MPa. The response of the single element model consists of two quantities, particularly they are axial stress  $\sigma$  and lateral strain  $\epsilon_2$ . The control variable is axial strain  $\epsilon_1$  which takes values on the interval  $(0; -8.6 \cdot 10^{-3})$  and is discretised into 100 uniformly distributed steps. The corresponding experimental errors are assumed to be normally distributed random variables with zero mean values and standard deviations 1.0 MPa in the case of measured stress and 0.1 for lateral strain. The likelihood function is then formulated as multivariate normal distribution of independent experimental errors of all measured values.

Posterior parameters' distribution is obtained in the form of 120.000 MCMC samples. For the case of verification, the resulting histograms of particular parameters are depicted in Figure 1 together with marked true values of model parameters. Parameters  $a$  and  $f_c$  are identified very precisely with small remaining uncertainties. A good match is obtained also for parameters  $b$ ,  $del$  and  $E$  but the estimations are burdened with higher uncertainties. The other parameters are not identified well, which is probably caused by mutual correlations among the parameters, see Figure 2, and low sensitivity of the model outputs to these parameters, see Figure 3.

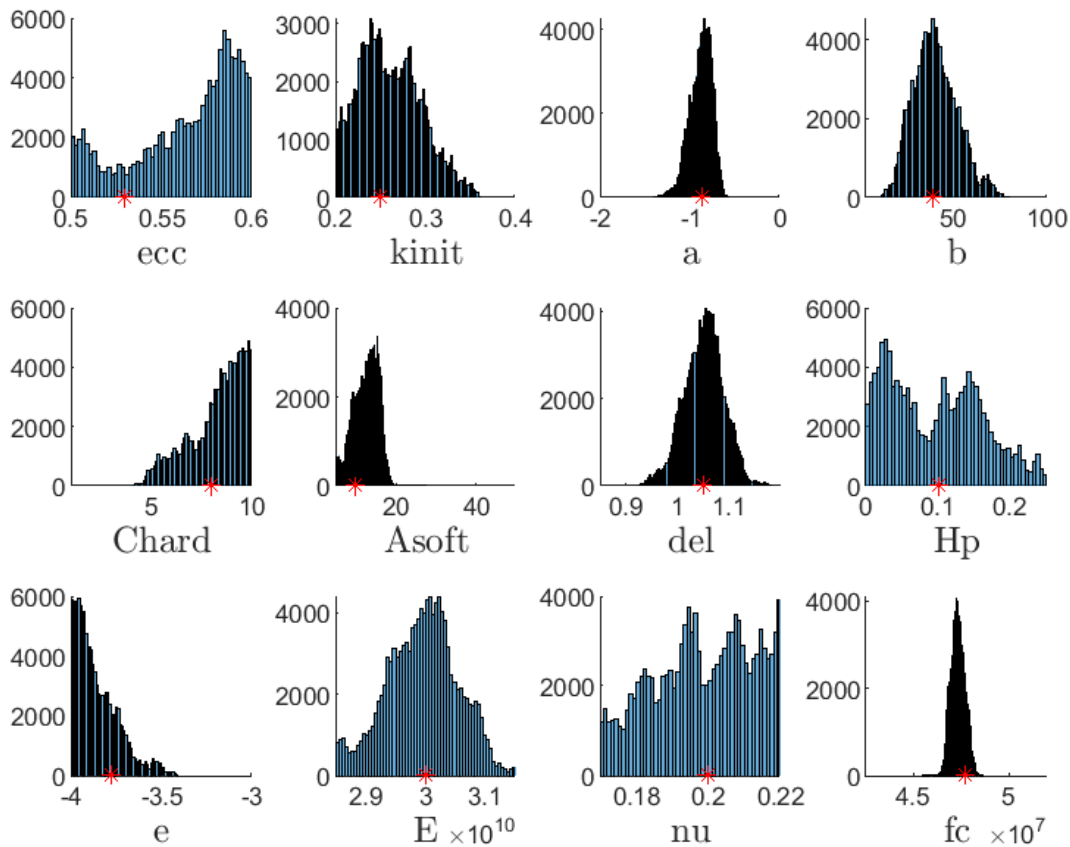


Figure 1: Verification: Histograms of generated MCMC samples and the true model inputs (\*).

Figure 2 presents two-dimensional marginal posterior distributions estimated from the obtained MCMC samples together with corresponding values of correlations. The strongest relationship is between parameters  $a$  and  $b$  which arises from the parameters definition itself as mentioned above. Another almost fully correlated parameters are  $e$  and  $A_{soft}$ . Significant relations are also between parameters  $ecc$  and  $f_c$ ,  $k_{init}$  and  $a$ ,  $C_{hard}$  and  $a$ ,  $C_{hard}$  and  $b$ ,  $H_p$  and  $del$ .

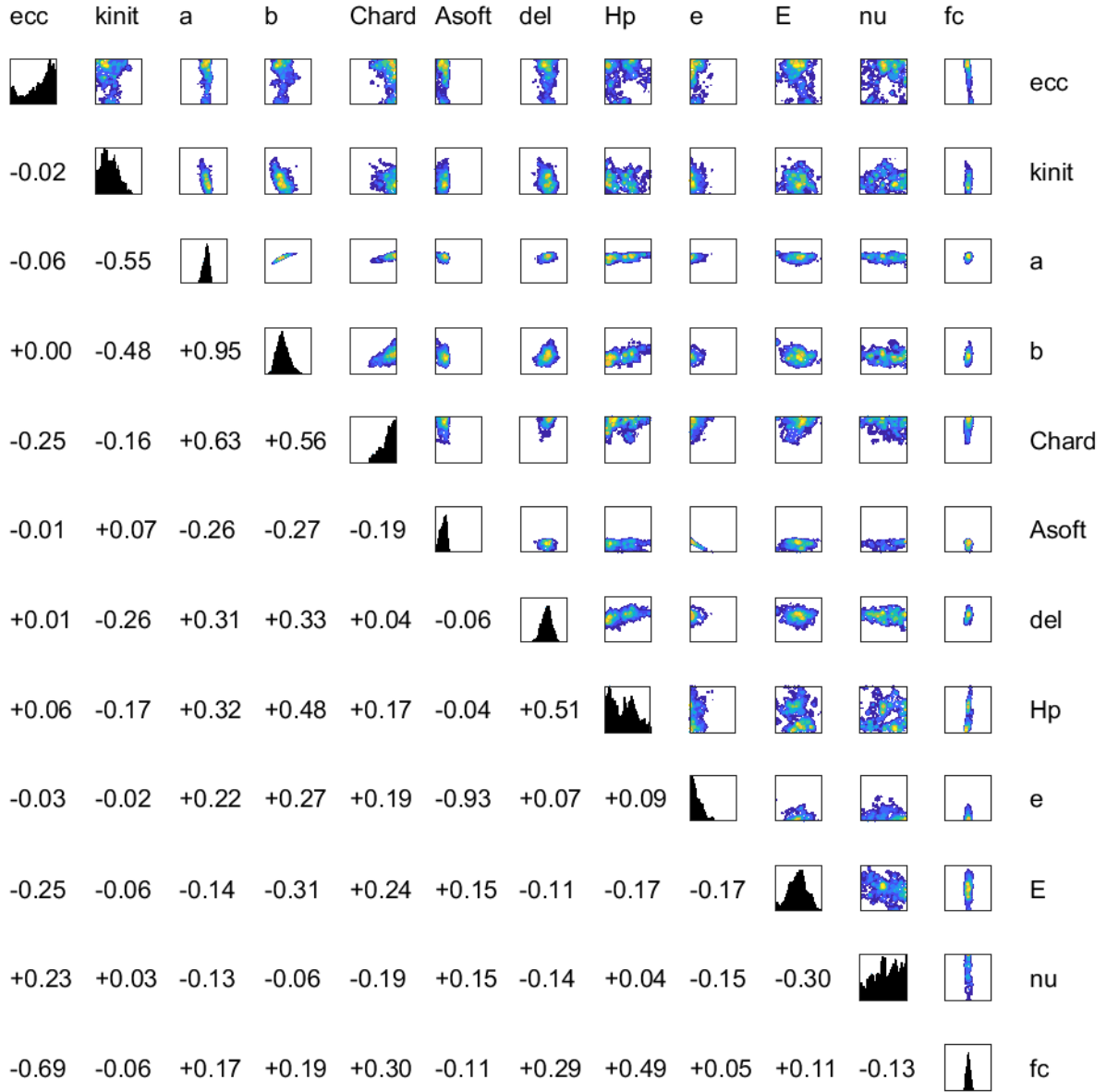


Figure 2: Verification: Two-dimensional marginal distributions estimated from generated MCMC samples and corresponding correlations.

Results of sensitivity analysis are depicted in Figure 3. Only 5 of total 12 unknown parameters play significant role in the model output on the defined prior intervals. The parameters with almost no influence to the model output cannot be identified from this type of measurement, therefore another experiment should be employed in the identification process to activate the parameters in order to make them identifiable.

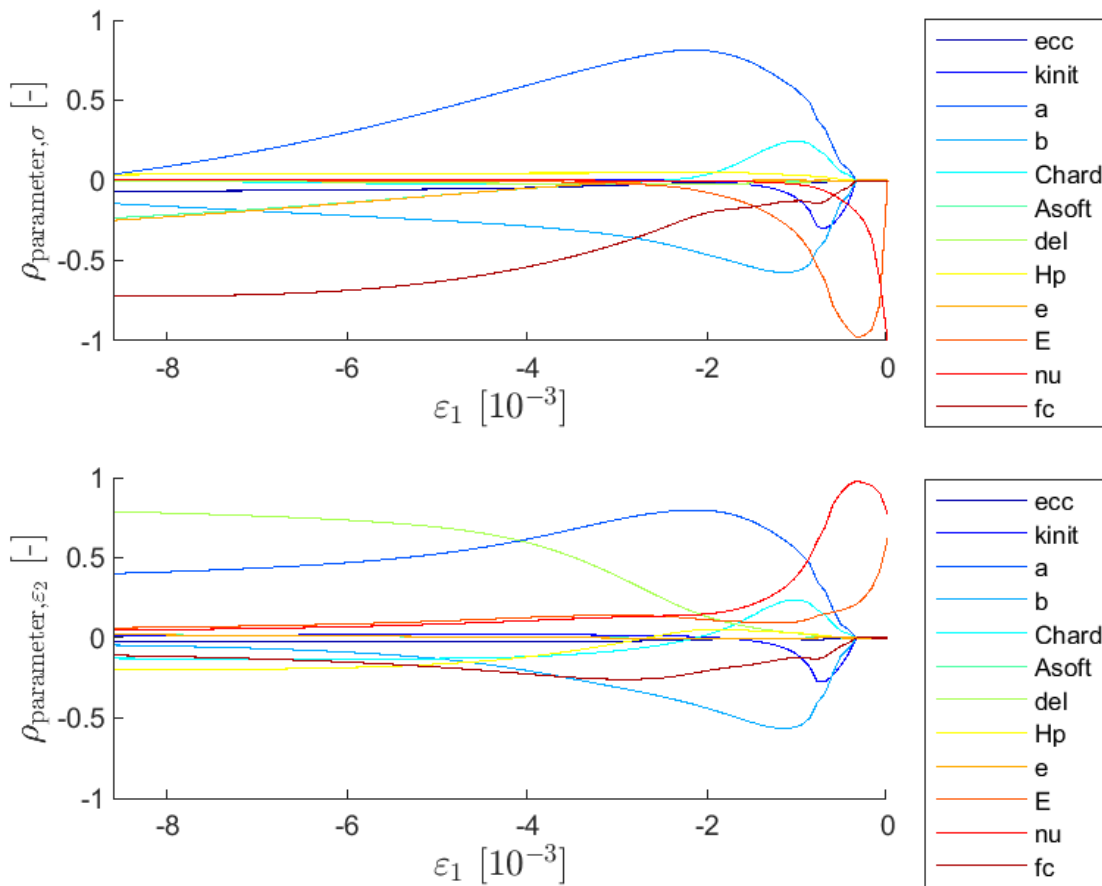


Figure 3: Sensitivity analysis based on Spearman’s rank correlation coefficients between model outputs and particular model input.



The important point of view for evaluation of method success is comparison of pseudo-experimental data and model response obtained for identified model parameters' values. Figure 4 shows a perfect match of these two curves.

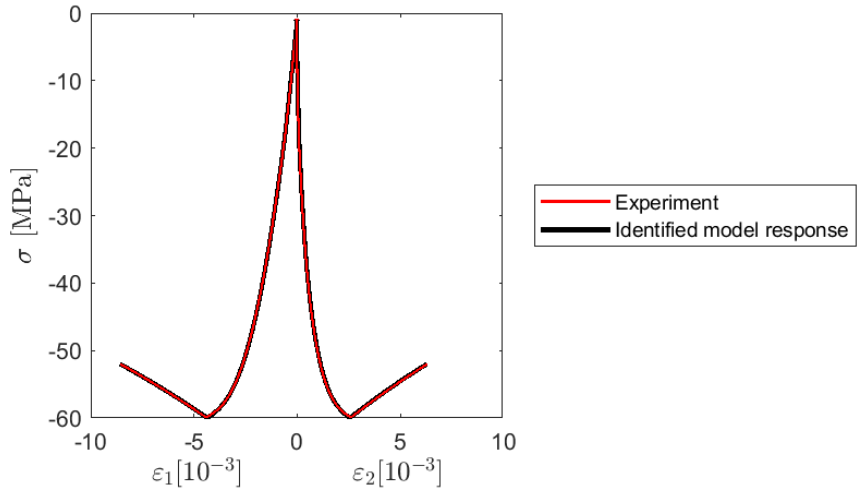


Figure 4: Verification: Comparison of pseudo-experimental data and model response for maximum a posteriori estimate of model parameters.

In the case of method validation, the same procedure was performed for experimental data from literature and the obtained results are very satisfactory. The comparison of experimental curve and model response for identified model parameters' values is presented in Figure 5. The identified parameters' distributions are shown in Figures 6 and 7.

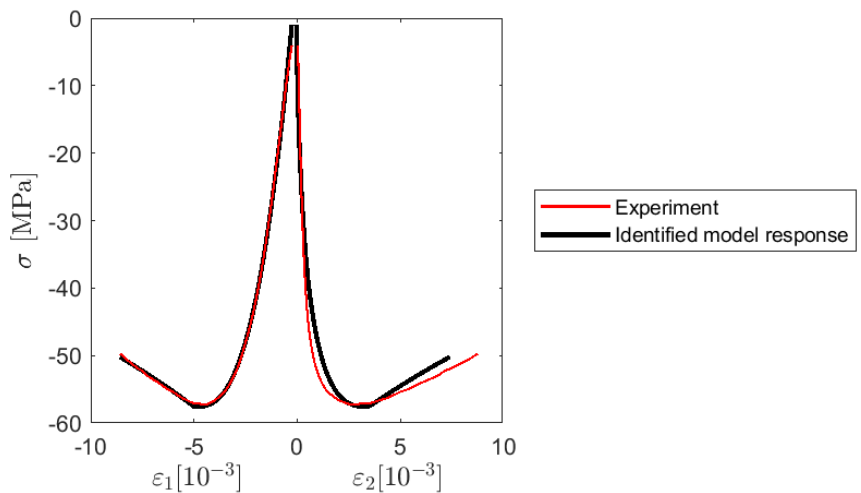


Figure 5: Validation: Comparison of experimental data and model response for maximum a posteriori estimate of model parameters.

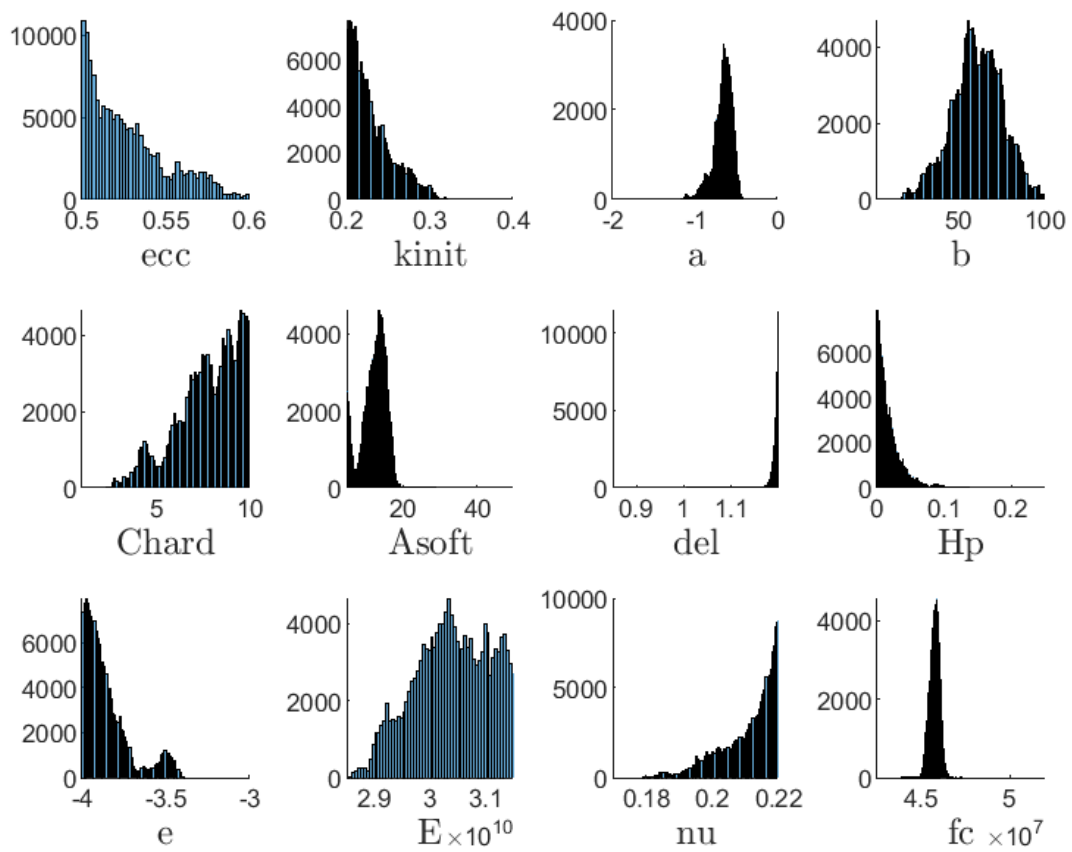


Figure 6: Validation: Histograms of generated MCMC samples.

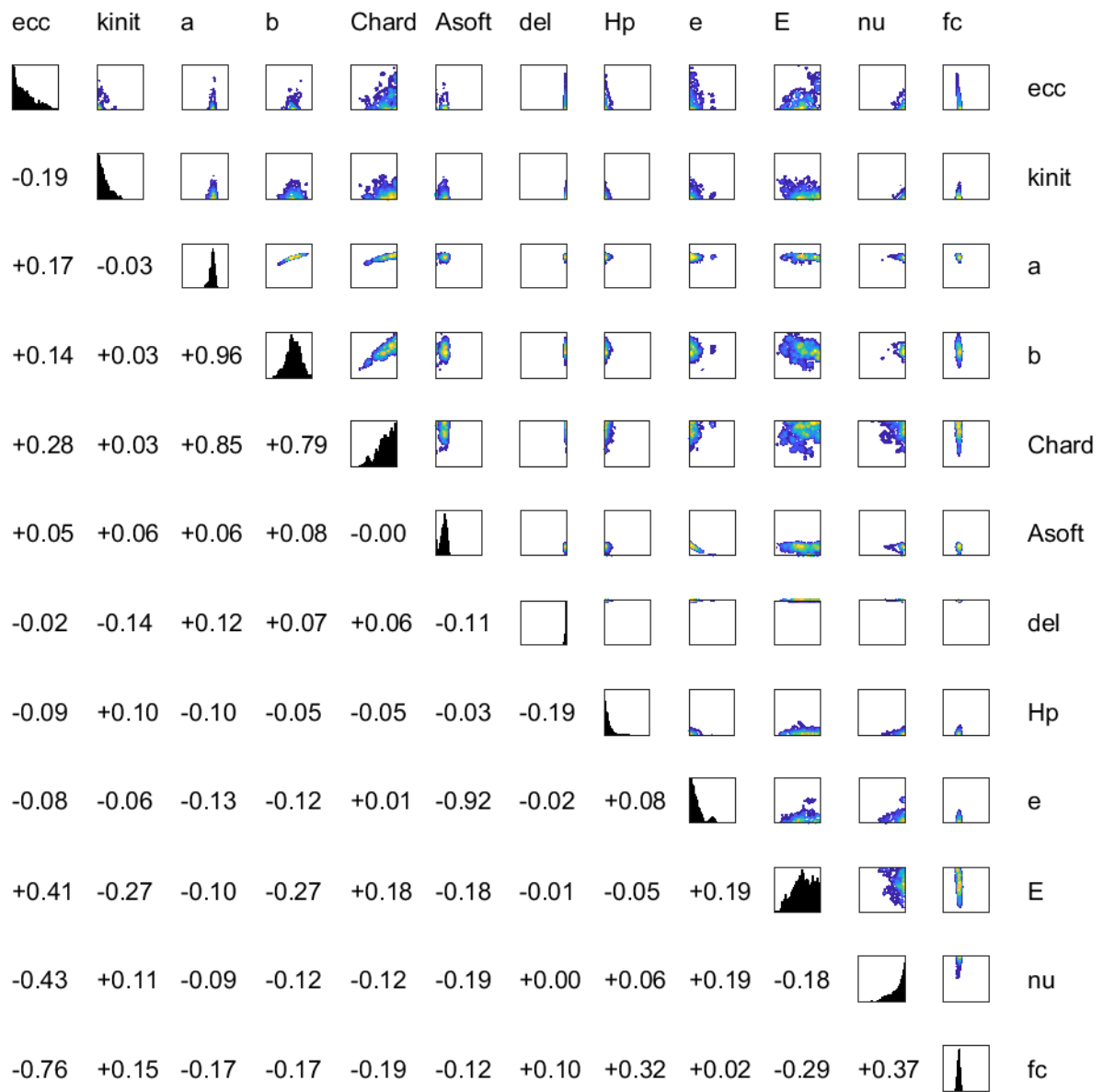


Figure 7: Validation: Two-dimensional marginal distributions estimated from generated MCMC samples and corresponding correlations.

### 3 Functions

In this chapter, the functions created and/or used by CTU team are briefly described. Following file folder tree is considered:

```
└─ CESTAR-Identification.Software.Package
   └─ data
   └─ figs
   └─ mfiles
      └─ Bayes_graphics.m
      └─ CDPM_tetra.m
      └─ halton_LHS.m
      └─ interpolate_model_response.m
      └─ interpolate_sig.m
      └─ OOFEM_CDPM_tetra.m
      └─ oofemInputFileFromVector_tetra.m
      └─ OOFEMout_imp.m
      └─ removeOscillations.m
      └─ run_MCMC_Metropolis_algorithm.m
      └─ run_SA.m
      └─ SA_graphics.m
      └─ transform_X.m
   └─ oofem_data
      └─ bin
      └─ lib
      └─ oofem.in
      └─ oofem.out
   └─ IDENTIFICATION.m
   └─ SENSITIVITY_ANALYSIS.m
```

### 3.1 Software for identification and sensitivity analysis

#### script IDENTIFICATION

The main script of the computer program for the identification of the model parameters. The calibrated model has to be defined as a function of a column ( $k \times 1$ ) vector of its parameters. The expected model response consists of two variables, each of them discretized into fixed discretization steps according to discretization of experimental data. The format of model response is a row vector containing all components of the first variable, then all components of the second variable. Variable `NoSamples` specifies a number of performed model simulations (equal to a number of generated samples). A priori knowledge of the model parameters is given as minimum and maximum of parameters' values, names of parameters are required in the case of demand for graphical outputs. Besides the experimental data themselves, standard deviations of the normally distributed experimental errors have to be specified. The computer program generates samples from the probability distribution defined according to Bayes' theorem for given setting. It enables to plot resulting figures and save the generated samples. For default setting, the program computes the samples of identified model parameters' probability distribution based on given pseudo-experimental data for single finite element model CDPM.

```

1  %% Author: Eliska Janouchova, Karel Mikes
2  % 11/2019, CTU in Prague, Czech Republic
3  %
4  %% Bayesian updating
5  % Identification of material model parameters
6  %
7  % Prior:      joint uniform distribution of parameters
8  % Likelihood: joint normal distribution of experimental errors in
9  %             given observations
10 %-----
11 clc
12 clear
13 RandStream.setGlobalStream(RandStream('mt19937ar', 'seed', sum(100*clock)));
14 warning('on');
15 % Add paths to folders containing required *.m files and data
16 path([pwd, '/mfiles'], path);
17 path([pwd, '/data'], path);
18
19 %% Numerical model
20 % OOFEM material model
21 % Confinement stress level
22 compression_vector = [0 2.15 4.3];
23 level = 2;
24 compression = compression_vector(level);
25 % Discretization steps of model response
26 load epsl_exp
27 steps = epsl_exp(level, :);
28 % Model
29 model = @(X) OOFEM_CDPM.tetra(X, compression, steps);
30 % Model output: [vector of model response, convergence control: 1 - ok, 0 - ko]
31
32 % Feasible number of model simulations for identification procedure
33 NoSamples = 100000;
34
35 %% Model parameters
36 % Number of parameters
37 k = 12;
38 % Names of parameters for graphical outputs
39 par_names = {'ecc', 'kinit', 'a', 'b', 'Chard', 'Asoft', 'del', 'Hp', 'e', 'E', 'nu', 'fc'};
40 % Prior bounds (minimum and maximum) of parameters' values
41 par_bounds = [0.5 0.6; 0.2 0.4; -2 0; 4 100; 1 10; 5 50; 0.85 1.2; 0 0.25; -4 -3; 28.5e9 31.5e9;

```

```

42 | 0.17 0.22; 42.5e6 52.0e6];
43 |
44 | %% Experimental data
45 | % Synthetic experimental data for verification of the identification method
46 | material_vector_test = [0.53 0.25 -0.876874 39.3098 8 10 1.05 0.1 -3.77322 30.e9 0.2 47.7e6]';
47 | Exper = model(material_vector_test);
48 | % Variable "Exper" can be replaced by real experimental data, but the discretization
49 | % has to be identical for data as well as for model response.
50 |
51 | % Experimental errors corresponding to:
52 | % 1. observed stress [MPa],
53 | epsilon1 = 1.0;
54 | % 2. observed lateral strain [-]
55 | epsilon2 = 0.1;
56 |
57 | %% Results
58 | % Name of output files
59 | filename = sprintf('Bayes.%1.1f', compression);
60 | % Save numerical results
61 | num_result = 1; %(1 - yes, 0 - no)
62 | % Save graphical results
63 | graph_result = 1; %(1 - yes, 0 - no)
64 |
65 | %% Markov chain Monte Carlo sampling of Bayesian posterior distribution for given setup
66 | run_MCMC_Metropolis_algorithm
67 |

```

### script SENSITIVITY ANALYSIS

This program serves as a tool for analysis of relations between the model response and a particular model parameter. The sensitivity of the model response to a model parameter is expressed by Spearman's rank correlation coefficients. The program allows to plot sensitivity coefficients for both response variables and save the coefficients together with model inputs and outputs employed for computing the correlations.

```

1 | %% Author: Eliska Janouchova, Karel Mikes
2 | % 11/2019, CTU in Prague, Czech Republic
3 | %
4 | %% Sensitivity analysis (SA)
5 | % Relation between model response and particular model parameters evaluated
6 | % by Spearman's rank correlation coefficient
7 | %
8 | %-----
9 | clc
10 | clear
11 | RandStream.setGlobalStream(RandStream('mt19937ar','seed',sum(100*clock)));
12 | warning('on');
13 | % Add paths to folders containing required *.m files and data
14 | path([pwd,'/mfiles'],path);
15 | path([pwd,'/data'],path);
16 |
17 |
18 | %% Numerical model
19 | % OOFEM material model
20 | % Confinement stress level
21 | compression_vector = [0 2.15 4.3];
22 | level = 2;
23 | compression = compression_vector(level);
24 | % Discretization steps of model response
25 | load eps1_exp
26 | steps = eps1_exp(level,:);
27 | % Model

```

```

28 | model = @(X)OOFEM_CDPM_tetra(X, compression, steps);
29 | % Feasible number of model simulations for SA
30 | n = 10000;
31 |
32 | %% Model parameters
33 | % Number of parameters
34 | k = 12;
35 | % Names of parameters for graphical outputs
36 | par_names = {'ecc', 'kinit', 'a', 'b', 'Chard', 'Asoft', 'del', 'Hp', 'e', 'E', 'nu', 'fc'};
37 | % Prior bounds (minimum and maximum) of parameters' values
38 | par_bounds = [0.5 0.6; 0.2 0.4; -2 0; 4 100; 1 10; 5 50; 0.85 1.2; 0 0.25; -4 -3; 28.5e9 31.5e9;
39 | 0.17 0.22; 42.5e6 52.0e6];
40 |
41 | %% Results
42 | % Name of output files
43 | filename = sprintf('SA_%1.1f', compression);
44 | % Save numerical results
45 | num_result = 1; %(1 - yes, 0 - no)
46 | % Save graphical results
47 | graph_result = 1; %(1 - yes, 0 - no)
48 |
49 | %% Sensitivity analysis for given setup
50 | run_SA

```

## 3.2 mfiles

### script Bayes\_graphics

The script provides graphical outputs of the probabilistic identification procedure:

- comparison of experimental observations and model response corresponding to the identified maximum a posteriori estimates of parameters' values,
- histograms of MCMC samples of each single parameter,
- two-dimensional marginal distributions estimated from MCMC.

**function** [eps1, eps2, sig, convergence] = CDPM\_tetra(material\_vector, compression, name)

The function computes the response of CDPM material model for given parameters and given confinement stress. First, the corresponding OOFEM input file is generated, then the OOFEM simulation is executed, and finally corresponding model response (i.e. vector of axial strains, vector of lateral strains, and vector of stress values) is provided together with information about simulation convergence.

<i>InputVar:</i>	material_vector	12 x 1 double	Values of CDPM parameters
	compression	1 x 1 double	Confinement stress
	name	1 x - char	Core name for OOFEM output file
<i>OutVar:</i>	eps1	1 x - double	Vector of axial strains
	eps2	1 x - double	Vector of lateral strains
	sig	1 x - double	Vector of stress values
	convergence	1 x 1 double	Identifier of OOFEM simulation convergence (1 - ok, 0 - ko)

**function** [X] = halton.LHS (np, dim)

The function creates a design of experiments of a defined number of samples in a defined number of dimensions. This advanced procedure created by our colleague generates Latin hypercube sampling design from the Halton sequence.

<i>InputVar:</i>	np	1 x 1 double	Number of design points
	dim	1 x 1 double	Number of variables
<i>OutVar:</i>	X	np x dim double	Design of experiments

**function** [eps1, eps2, sig] = interpolate\_model\_response(eps1, eps2, sig, steps)

The function interpolates numerical response of the model (i.e. vector of axial strains, vector of lateral strains, and vector of stress values) into the points defined by vector of experimental axial strains

<i>InputVar:</i>	eps1	1 x - double	Vector of axial strains
	eps2	1 x - double	Vector of lateral strains
	sig	1 x - double	Vector of stress values
	steps	1 x n double	Vector of experimental axial strains
<i>OutVar:</i>	eps1	1 x n double	Vector of interpolated axial strains
	eps2	1 x n double	Vector of interpolated lateral strains
	sig	1 x n double	Vector of interpolated stress values

**function** [s] = interpolate\_sig(sig, eps1, discretization)

The function interpolates computed stress values in given vector of experimental axial strains. Linear interpolation is used.

<i>InputVar:</i>	sig	1 x - double	Vector of stress values
	eps1	1 x - double	Vector of axial strains
	discretization	1 x n double	Vector of experimental axial strains
<i>OutVar:</i>	s	1 x n double	Vector of interpolated stress values



```
function [response, convergence] = OOFEM_CDPM_tetra(material_vector, compression)
```

The function defines the numerical model to be calibrated. It incorporates all functions regarding to the numerical material model, i.e. creation OOFEM input file, OOFEM simulation and post-processing of output data, and provides the final model response together with information about simulation convergence.

<i>InputVar:</i>	material_vector	12 x 1 double	Values of CDPM parameters
	compression	1 x 1 double	Confinement stress
<i>OutVar:</i>	response	1 x 2n double	Vector of stress values and lateral strains
	convergence	1 x 1 double	Identifier of OOFEM simulation convergence (1 - ok, 0 - ko)

```
function [ ] = oofemInputFileFromVector_tetra(compression, parameters, TypeOfConcrete, OutputFileName)
```

The function generates OOFEM input file (3.3) to provide compression test of a single tetrahedral element with given confinement stress and given material parameters of CDPM material model.

<i>InputVar:</i>	compression	1 x 1 double	Confinement stress
	parameters	12 x 1 double	Values of CDPM parameters
	TypeOfConcrete	1 x - string	Define type of concrete "55d"
	OutputFileName	1 x - string	Name of OOFEM output file
<i>OutVar:</i>			

```
function [reqData, info] = OOFEMout_imp(filename, requestedField, shortenFile, nLinesToRead)
```

The function, created by our colleagues, enables to extract required data from OOFEM output file (3.3) into MATLAB environment. A detailed description of the function is presented in the beginning of the corresponding m-file.

<i>InputVar:</i>	filename	1 x 1 string	Name of output file with extension
	requestedField	1 x - array	Labels of the requested fields
	shortenFile	1 x - boolean	shorten version on input file
	nLinesToRead	1 x 1 integer	Number of lines to be read at once
<i>OutVar:</i>	reqData	1 x - structure	Output data
	info	1 x - structure	Information details

```
function [sig_new, eps1_new, eps2_new] = removeOscillations(sig, eps1, eps2)
```

The function removes oscillations in computed stress response. In some rare cases, the computed stresses may exhibit nonphysical peaks. This function is designed to find such peaks and replace them with values interpolated from surrounding points to provide smooth stress response.

<i>InputVar:</i>	sig	1 x - double	Vector of stress values
	eps1	1 x - double	Vector of lateral strains
	eps2	1 x - double	Vector of lateral strains
<i>OutVar:</i>	sig_new	1 x - double	Vector of modified stress values
	eps1_new	1 x - double	Vector of modified lateral strains
	eps2_new	1 x - double	Vector of modified lateral strains

**script** `run.MCMC.Metropolis.algorithm`

The script creates Markov chain on a basis of the Metropolis algorithm with symmetric random walk. An efficient setup of the algorithm is problem dependent. In the beginning of the script, two main features can be changed in order to accelerate the algorithm convergence. It involves a choice of starting point and proposal standard deviation. The starting point  $x$  is set as default to be equal to a mean value of the prior distribution. Proposal distribution of a new state is normal distribution with zero mean value and standard deviation `s_p` chosen according acceptance rate `AR`, an optimal value of `AR` is between 0.1 and 0.6. The next adjustable feature is length of burn-in period, which involves initial samples until the stationary distribution of Markov chain is reached. Default value of `burnin` is 10% of `NoSamples`, but it specifically depends on the algorithm convergence rate.

Input variables and choice of outputs are specified in the main script of the program IDENTIFICATION (3.1).

**script** `run.SA`

The script provides computation of Spearman's rank correlation coefficients between model response and model parameters on a basis of Latin hypercube design of experiments generated from the Halton sequence by the function `halton_LHS` (3.2).

Inputs and choice of outputs are specified in the main script of the program SENSITIVITY\_ANALYSIS (3.1).

**script** `SA.graphics`

The script creates graphical outputs of the sensitivity analysis of the numerical model, if it is required in the main script of the program SENSITIVITY\_ANALYSIS (3.1).

**function** `[material_vector] = transform_X(X, par_bounds)`

The function transforms normalised parameters from the interval  $[0, 1]$  into material model parameters from the interval with minimum and maximum defined in `par_bounds`.

<i>InputVar:</i>	X	k x 1 double	Normalised values of model parameters
	par_bounds	k x 2 double	Minimum and maximum for parameters' values
<i>OutVar:</i>	material_vector	k x 1 double	Values of model parameters

### 3.3 `oofem_data`

This folder contains all files associated with OOFEM, which is free finite element code with object oriented architecture available at <http://www.oofem.org/>. The detail of individual folders and files are listed below.

**folder** `bin`

This folder contains the installation of OOFEM, including the main executable file `oofem.exe`.

**folder** `lib`

This folder contains dynamic library used by OOFEM.

**file** `oofem.in`

The OOFEM input file that contains all the information required to run OOFEM simulation. Each line in input file represents one input record. The order of records in file is compulsory with given structure. The individual records consist of record keyword followed by one or more attributes. Full details and examples can be found in [5]. Before each OOFEM simulation, the corresponding input file is automatically generated via function `oofemInputFileFromVectortetra`.

**file** `oofem.out`

The OOFEM output file, that contain data computed by OOFEM. This file is created/rewritten by OOFEM after each execution. After each OOFEM simulation, the required data are automatically extracted from current output file; see function `OOFEMoutimp` for more details.

## Acknowledgement

Financial support for creating this software package was provided by the Technology Agency of the Czech Republic (TAČR), project number TF05000040 (CeSTaR: Computer simulation and experimental validation - complex service for flexible and efficient design of pre-cast concrete columns with innovative multispiral reinforcement).

## References

- [1] Z. Bittnar, P. Havlásek, K. Mikeš, E. Janouchová, A. Kučerová, and P. Bittnar. Identification of material parameters, design of a new compression test. Technical report, CTU in Prague, Faculty of Civil Engineering, Department of Mechanics, 2018.
- [2] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman & Hall/CRC, 2nd edition, 2004.
- [3] Peter Grassl, Dimitrios Xenos, Ulrika Nyström, Rasmus Rempling, and Kent Gylltoft. CDPM2: A damage-plasticity approach to modelling the failure of concrete. *International Journal of Solids and Structures*, 50(24):3805 – 3816, 2013.
- [4] I. Imran and S. Pantazopoulou. Experimental study of plain concrete under triaxial stress. *ACI Materials Journal-American Concrete Institute*, 93(6):589–601, 1996.
- [5] B. Patzák. OOFEM: Input data format specification. [Online]. Available at: <http://www.oofem.org/resources/doc/oofemInput/oofemInput.pdf>, 2017.
- [6] J. C. Spall. Estimation via markov chain monte carlo. *IEEE Control Systems Magazine*, 23(2):34–45, 2003.