INVERSION: SURROGATE MODEL ACCELERATED CALIBRATION

Table of Contents

In this example, it is shown how a surrogate model can be constructed and then used instead of the original forward model in a Bayesian inversion analysis. For a computationally expensive forward model, this approach can yield considerable time savings in the analysis.

The problem considered here is similar to the one in `uq_Example_Inversion_01_Beam`. The inversion analysis is rerun and compared to an analysis using a polynomial chaos expansion (PCE) surrogate of the full computational model.

1 - INITIALIZE UQLAB

Clear all variables from the workspace, set the random number generator for reproducible results, and initialize the UQLab framework:

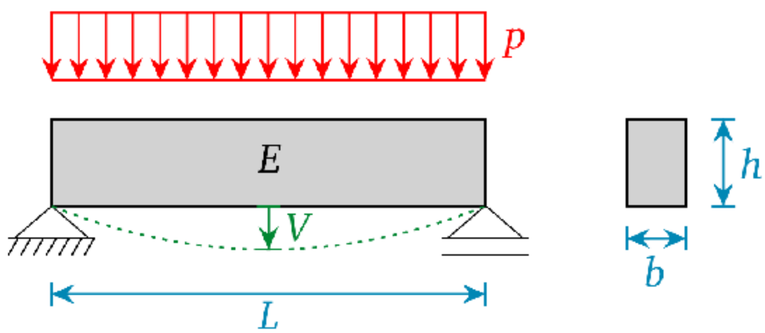
```
clearvars
rng(100,'twister')
uqlab
```

```
Copyright 2013-2019, Stefano Marelli and Bruno Sudret, all rights
 reserved.
This is UQLabCore, version 1.3.0
UQLabCore may not be reproduced, displayed, modified or redistributed
without the express prior written permission of the copyright
 holder(s).
To request special permissions, please contact:
  - Stefano Marelli (<a
 href="mailto:marelli@ibk.baug.ethz.ch">marelli@ibk.baug.ethz.ch</a>).

Useful commands to get started with UQLab:
<a href="matlab:uqlab -doc">uqlab -doc</a>         - Access the
 available documentation
<a href="matlab:uqlab -help">uqlab -help</a>        - Additional help
 on how to get started with UQLab
<a href="matlab:uq_citation help">uq_citation help</a>    - Information
 on how to cite UQLab in publications
<a href="matlab:uq_sendFeedback">uq_sendFeedback</a>     - Report bugs,
 suggestions, enhancement requests, etc.
```

2 - FORWARD MODEL

The simply supported beam problem is shown in the following figure:

```
uq_figure
[I,~] = imread('SimplySupportedBeam.png');
image(I)
axis equal
set(gca, 'visible', 'off')
```



Define the forward model as a MODEL object using the function `uq_SimplySupportedBeam(X)`:

```
ModelOpts.mFile = 'uq_SimplySupportedBeam';
ModelOpts.isVectorized = true;

myForwardModel = uq_createModel(ModelOpts);
```

For more information about the function `uq_SimplySupportedBeam(X)`, refer to `uq_Example_Inversion_01_Beam`.

3 - PRIOR DISTRIBUTION OF THE MODEL PARAMETERS

The prior information about the model parameters is gathered in a probabilistic model that includes both known (constant) and unknown parameters.

```
PriorOpts.Marginals(1).Name = 'b';               % beam width
PriorOpts.Marginals(1).Type = 'Constant';
PriorOpts.Marginals(1).Parameters = [0.15];      % (m)

PriorOpts.Marginals(2).Name = 'h';               % beam height
PriorOpts.Marginals(2).Type = 'Constant';
PriorOpts.Marginals(2).Parameters = [0.3];       % (m)

PriorOpts.Marginals(3).Name = 'L';               % beam length
PriorOpts.Marginals(3).Type = 'Constant';
PriorOpts.Marginals(3).Parameters = 5;           % (m)

PriorOpts.Marginals(4).Name = 'E';               % Young's modulus
PriorOpts.Marginals(4).Type = 'LogNormal';
PriorOpts.Marginals(4).Moments = [30000 4500];   % (MPa)

PriorOpts.Marginals(5).Name = 'p';               % uniform load
PriorOpts.Marginals(5).Type = 'Gaussian';
PriorOpts.Marginals(5).Moments = [0.012 0.012*0.05]; % (kN/m)

myPriorDist = uq_createInput(PriorOpts);
```

4 - SURROGATE MODEL

Use polynomial chaos expansions (PCE) to construct a surrogate model of `myForwardModel`:

```
MetaOpts.Type = 'Metamodel';
MetaOpts.MetaType = 'PCE';
MetaOpts.Method = 'LARS';

MetaOpts.ExpDesign.Sampling = 'Sobol';
MetaOpts.ExpDesign.NSamples = 10;
MetaOpts.Degree = 10;

MetaOpts.Input = myPriorDist;
MetaOpts.FullModel = myForwardModel;
mySurrogateModel = uq_createModel(MetaOpts);
```

```
---   Calculating the PCE coefficients with least-squares.   ---
The estimation of PCE coefficients stopped at polynomial degree 10 and
 qNorm 1.00 for output variable 1
Final LOO error estimate: 2.221250e-02
---                   Calculation finished!
   ---
```

With just 50 model evaluations, the PCE is extremely accurate and has a leave-one-out cross-validation error of $\varepsilon_{\mathrm{LOO}} \approx 10^{-7}$.

5 - MEASUREMENT DATA

The measurement data consists of $N = 5$ independent measurements of the beam mid-span deflection. The data is stored in the column vector `y`:

```
myData.y = [12.84; 13.12; 12.13; 12.19; 12.67]/1000;  % (m)
```

```
myData.Name = 'Mid-span deflection';
```

6 - BAYESIAN ANALYSIS

The options of the Bayesian inversion analysis are specified with the following structure:

```
BayesOpts.Type = 'Inversion';
BayesOpts.Data = myData;
BayesOpts.Solver.Type = 'MCMC';
BayesOpts.Solver.MCMC.Sampler = 'MH';
BayesOpts.Solver.MCMC.Steps = 1e3;
BayesOpts.Solver.MCMC.NChains = 1e2;
BayesOpts.Solver.MCMC.T0 = 1e1;
% %%
% % To use the original forward model |myForwardModel| in the
 analysis,
% % set the following option:
 BayesOpts.ForwardModel.Model = myForwardModel;

% %%
% % Run the Bayesian inversion analysis:
 myBayesianAnalysis_fullModel = uq_createAnalysis(BayesOpts);

% %%
% % Print out a report of the results:
 uq_print(myBayesianAnalysis_fullModel)
```

```
The discrepancy was not specified,
using unknown i.i.d. Gaussian discrepancy...

Starting Metropolis Hastings Algorithm...

|##############################| 100.00%

Finished Metropolis Hastings Algorithm!

%----------------------- Inversion output ----------------------%
   Number of calibrated model parameters:         2
   Number of non-calibrated model parameters:     3

   Number of calibrated discrepancy parameters:   1

%------------------- Data and Discrepancy
%  Data-/Discrepancy group 1:
   Number of independent observations:            5

   Discrepancy:
      Type:                                       Gaussian
      Discrepancy family:                         Scalar
      Discrepancy parameters known:               No

   Associated outputs:
      Model 1:
         Output dimensions:                       1
```

```
%------------------- Solver
   Solution method:                               MCMC

   Algorithm:                                     MH
   Duration (HH:MM:SS):                           00:00:09
   Number of sample points:                       1.00e+05

%------------------- Posterior Marginals
---------------------------------------------------------------------
| Parameter | Mean    | Std     | (0.05-0.95) Quant.  | Type        |
---------------------------------------------------------------------
| E         | 2.4e+04 | 1.9e+03 | (2.2e+04 - 2.7e+04) | Model       |
| p         | 0.012   | 0.00054 | (0.011 - 0.013)     | Model       |
| Sigma2    | 2.7e-06 | 8.6e-06 | (1.1e-07 - 1e-05)   | Discrepancy |
---------------------------------------------------------------------

%------------------- Point estimate
----------------------------------------
| Parameter | Mean    | Parameter Type |
----------------------------------------
| E         | 2.4e+04 | Model          |
| p         | 0.012   | Model          |
| Sigma2    | 2.7e-06 | Discrepancy    |
----------------------------------------

%------------------- Correlation matrix (Model Parameters)
----------------------
|   |  E       p     |
----------------------
| E |  1       0.46  |
| p |  0.46    1     |
----------------------
```

For comparison, the analysis is now rerun using the surrogate model `mySurrogateModel` in lieu of the original `myForwardModel`:

```
BayesOpts.ForwardModel.Model = mySurrogateModel;
```

Run the Bayesian inversion analysis:

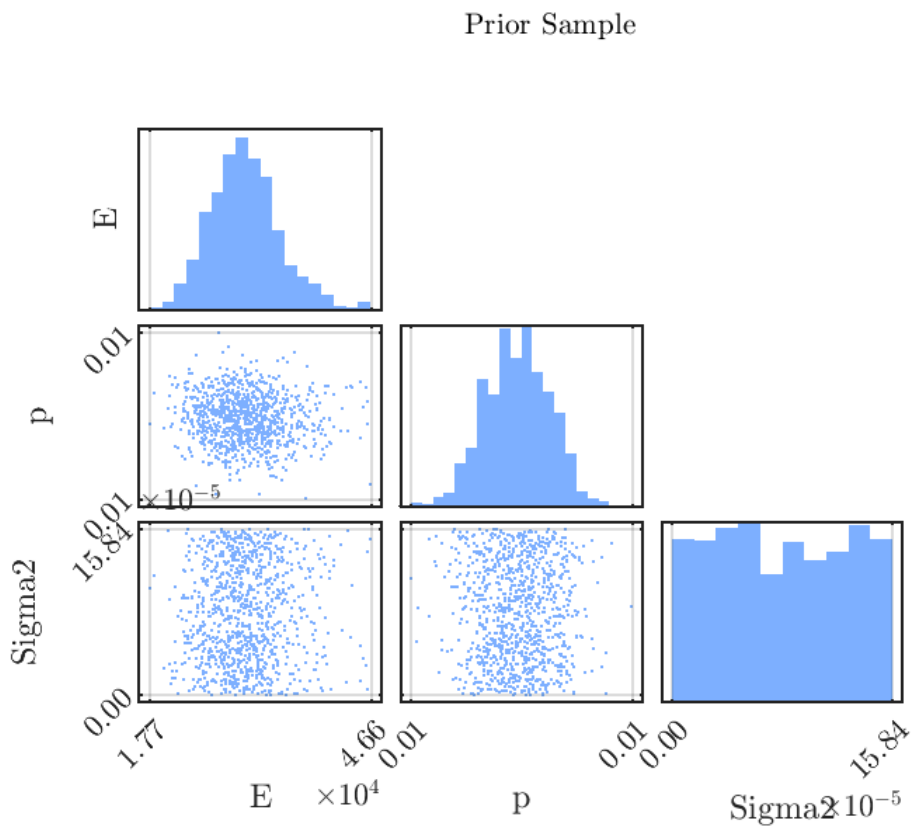
```
myBayesianAnalysis_surrogateModel = uq_createAnalysis(BayesOpts);
```

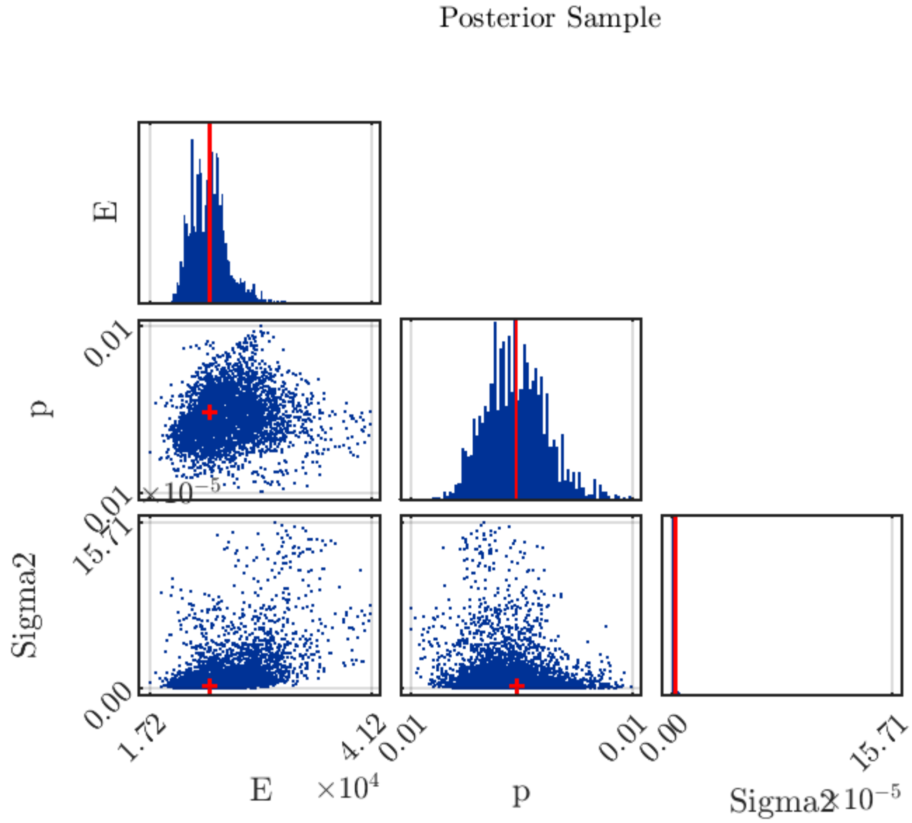
```
The discrepancy was not specified,
using unknown i.i.d. Gaussian discrepancy...

Starting Metropolis Hastings Algorithm...

|###############################| 100.00%

Finished Metropolis Hastings Algorithm!
```

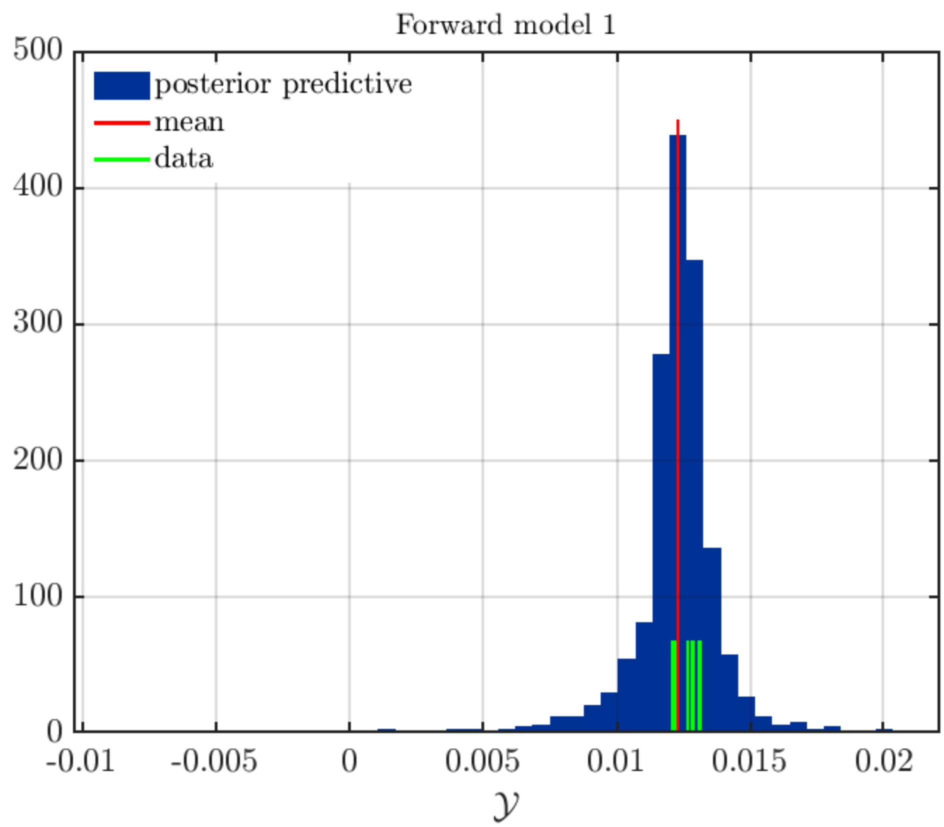
Print out a report of the results:

```
uq_print(myBayesianAnalysis_surrogateModel)
uq_display(myBayesianAnalysis_surrogateModel)


%----------------------- Inversion output -----------------------%
   Number of calibrated model parameters:         2
   Number of non-calibrated model parameters:     3

   Number of calibrated discrepancy parameters:   1

%------------------- Data and Discrepancy
%  Data-/Discrepancy group 1:
   Number of independent observations:            5

   Discrepancy:
      Type:                                       Gaussian
      Discrepancy family:                         Scalar
      Discrepancy parameters known:               No

   Associated outputs:
      Model 1:
         Output dimensions:                       1

%------------------- Solver
   Solution method:                               MCMC

   Algorithm:                                     MH
   Duration (HH:MM:SS):                           00:00:16
   Number of sample points:                       1.00e+05

%------------------- Posterior Marginals
---------------------------------------------------------------------
| Parameter | Mean    | Std     | (0.05-0.95) Quant.  | Type        |
---------------------------------------------------------------------
| E         | 2.4e+04 | 2.1e+03 | (2.1e+04 - 2.8e+04) | Model       |
| p         | 0.012   | 0.00052 | (0.012 - 0.013)     | Model       |
| Sigma2    | 2.7e-06 | 8.8e-06 | (1.2e-07 - 1e-05)   | Discrepancy |
---------------------------------------------------------------------

%------------------- Point estimate
----------------------------------------
| Parameter | Mean    | Parameter Type |
----------------------------------------
| E         | 2.4e+04 | Model          |
| p         | 0.012   | Model          |
| Sigma2    | 2.7e-06 | Discrepancy    |
----------------------------------------

%------------------- Correlation matrix (Model Parameters)
----------------------
|   |  E       p     |
----------------------
| E |  1       0.51  |
| p |  0.51    1     |
```

- -


Prior Sample
E
p
Sigma2
0.01
0.01
$\times10^{-5}$
15.84
0.00
1.77
4.66
0.01
0.01
0.00
15.84
E
$\times10^{4}$
p
Sigma2
$\times10^{-5}$


Posterior Sample
E
p
Sigma2
0.01
0.01
×10⁻⁵
15.71
0.00
1.72
4.12
0.01
0.01
0.00
15.71
E ×10⁴
p
Sigma2 ×10⁻⁵


Forward model 1
posterior predictive
mean
data
500
400
300
200
100
0
-0.01
-0.005
0
0.005
0.01
0.015
0.02
𝒴

Comparing `myBayesianAnalysis_fullModel` and `myBayesianAnalysis_surrogateModel` it can be seen that the results are practically identical. The small differences come from the randomness of the MCMC algorithms.

The number of original forward model calls in MCMC with the full model was $N = 90{,}000$ compared to the $N = 50$ model evaluations necessary to compute the PCE surrogate. In cases where the original forward model is computationally expensive, accelerating MCMC with surrogate models result in significant reduction of the total computational costs.

In this example, a PCE surrogate was used, but generally any surrogate model available in UQLab (e.g., Kriging, LRA, SVR) can be used.

Published with MATLAB® R2020a