Computer model calibration as a method of design

Carl Ehrett

March 28, 2018

1 Introduction

1.1 Computer experiments

1.2 Computer model calibration

1.2.1 Gaussian processes

Background

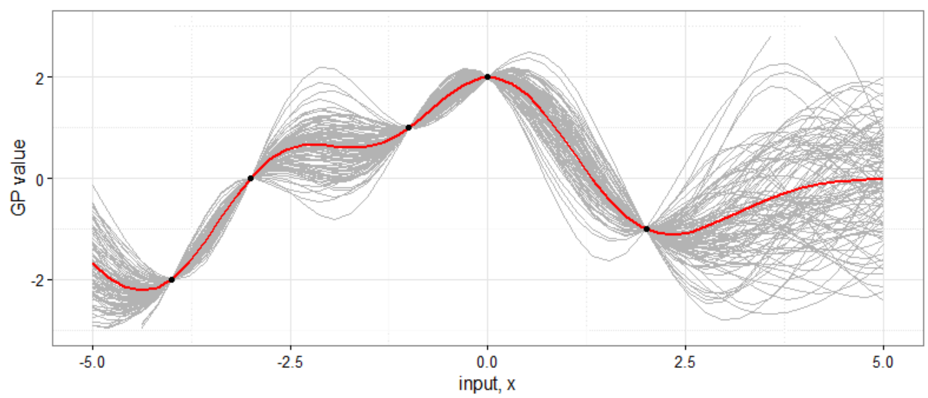

Figure 1: Example of a Gaussian process trained to interpolate five data points (black dots).

Gaussian process regression

Gaussian processes in computer model calibration

1.2.2 Markov chain Monte Carlo methods

Background

Metropolis-Hastings algorithm

Elimination of boundary constraints

1.2.3 Normalization of inputs and standardization of outputs

Blah

1.2.4 Computational difficulties

Blah

Likelihoods Blah

Ill-conditioned covariance matrices

2 Calibration for design

3 Application

3.1 Project background

3.2 Emulation of finite element simulator

Blah

3.2.1 Wind turbine blade simulator

Blah

3.2.2 Mathematical basis for the emulator

Blah

3.2.3 Experimental design

Blah

3.2.4 Covariance parameters

Blah

Finding covariance parameters via MCMC Blah

Grid optimization Blah

Gradient method Blah

4 MCMC using the emulator

Blah

4.1 MCMC methods

Blah

4.2 The model

Blah

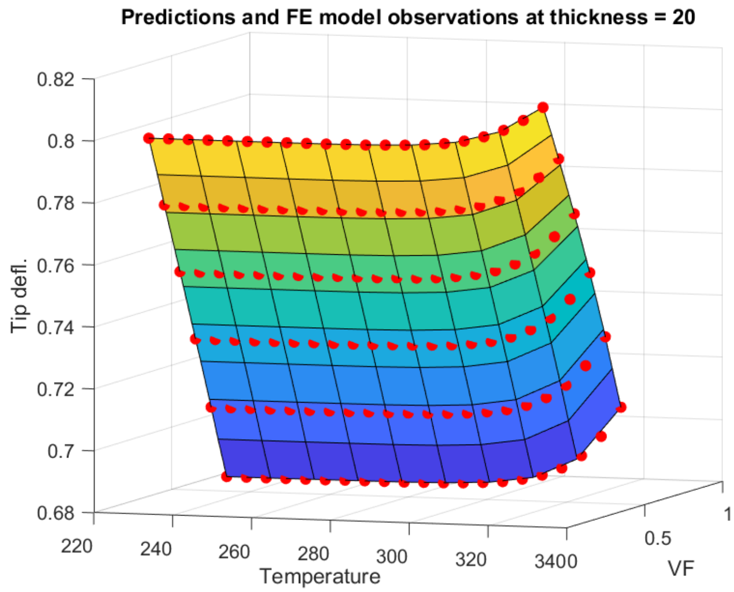

Figure 2: A slice of the GP emulator (restricted to the output for tip deflection) at thickness =20mm. Red dots are observations from the simulator.

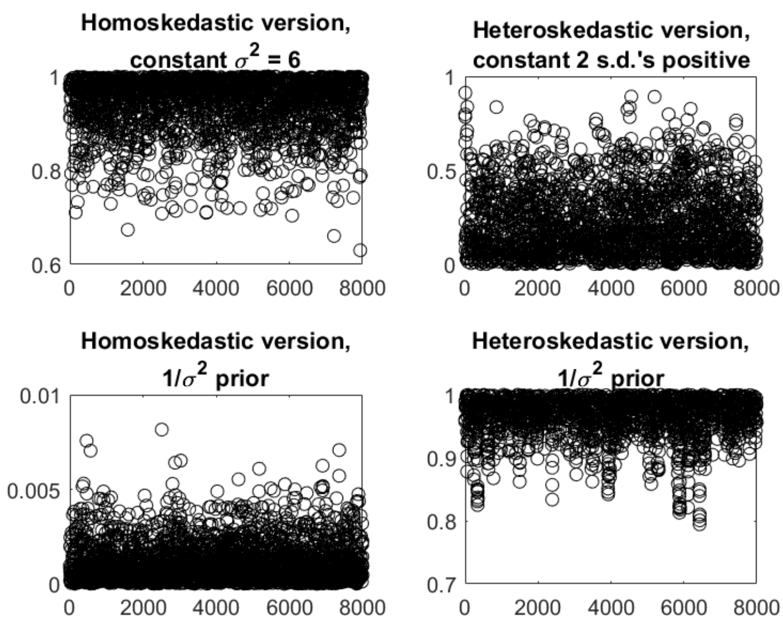

Figure 3: MCMC results at various observation variance settings.

	Heteroskedastic, constant	Homoskedastic, constant	Heteroskedastic, prior	Homoskedastic, prior
Deflection	0.749	0.729	0.659	0.709
Rotation	0.0904	0.0865	0.0773	0.0843
Cost	276.16	236.11	350.80	233.95

Table 1: Comparison of model outputs, where the desired data outputs are assumed to be either homoskedastic or heteroskedastic, with either a specified constant variance or a $1/\sigma^2$ prior.

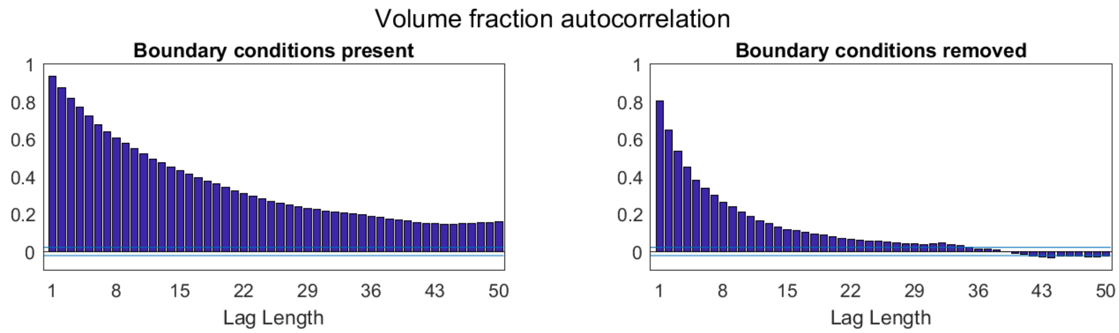

Figure 4: Auto-correlation for draws both with and without the elimination of boundary conditions.

4.2.1 Desired observation variance

4.2.2 Full model and likelihood

Blah

4.2.3 Convergence difficulties

Blah

4.2.4 Implementation of the Metropolis-Hastings algorithm

Blah

4.3 Which data to desire?

Blah

4.3.1 Motivations behind the choice of desired data

Blah

4.3.2 Differing results

Desired data d	σ^2_{defl}	σ^2_{rot}	σ^2_{cost}	$\mu_{v\|d}$	$\mu_{h\|d}$	$\sigma^2_{v\|d}$	$\sigma^2_{h\|d}$
$(0,0,0)$	375.45	277.69	2.62	0.215	$4.01 \cdot 10^{-2}$	$4.41 \cdot 10^{-2}$	$1.92 \cdot 10^{-3}$
$(0.65, 0.077, 96)$	16.74	15.25	$4.62 \cdot 10^{-7}$	$1.09 \cdot 10^{-3}$	$3.36 \cdot 10^{-4}$	$1.02 \cdot 10^{-5}$	$9.97 \cdot 10^{-6}$

Table 2: Comparison of results for two different (low) values of d. Values listed are, respectively, the posterior means for the observation variance of each model output, posterior means for volume fraction (v) and thickness (h), and posterior variance of volume fraction and thickness.

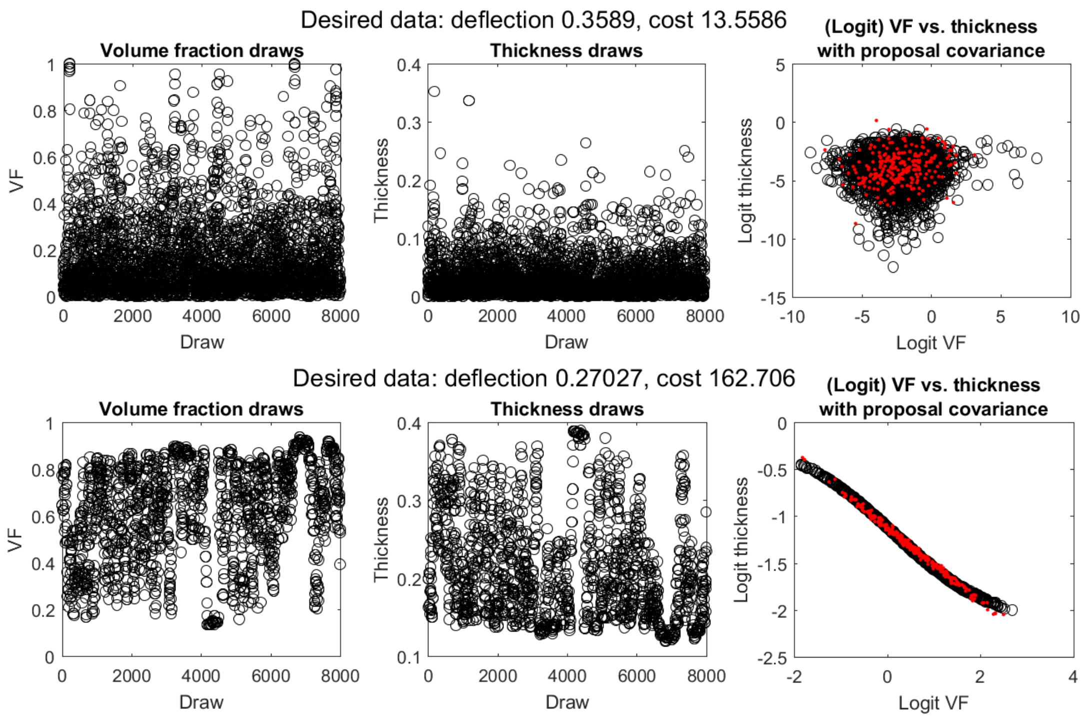

Figure 5: MCMC results for low deflection and cost (top row) and low deflection with easily achievable cost (bottom row).

4.4 Exponentially distributed desired data

Blah

4.4.1 Motivation

Blah

4.4.2 Implementation and results

Blah

4.5 Identifiability issues

Blah

5 Future work

Blah

5.1 Alternative means of handling cost

Blah

5.1.1 Removing cost from the model

Blah

5.1.2 Alternative priors for controlling cost

Blah

5.2 Building a desired data response surface

Blah

5.3 Implementing Hamiltonian Monte Carlo

Blah

5.3.1 Hamiltonian Monte Carlo

Blah

5.3.2 Benefits

Blah

5.4 Model discrepancy

Blah

6 Conclusion

Blah