

André Gustavo Carlon

**Bayesian-based optimization of experiments  
using stochastic gradient methods**

Florianópolis/SC, Brazil

2019



-

André Gustavo Carlon

**Bayesian-based optimization of experiments using  
stochastic gradient methods**

In Partial Fulfillment of the Requirements for the Degree of Doctor in Engineering

Universidade Federal de Santa Catarina – UFSC

Programa de Pós-Graduação em Engenharia Civil – PPGEC

Supervisor: Rafael Holdorf Lopez, Prof.

Florianópolis/SC, Brazil

2019

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Carlton, André

Bayesian-based optimization of experiments using  
stochastic gradient methods / André Carlton ;  
orientador, Rafael Holdorf Lopez, 2019.  
135 p.

Tese (doutorado) - Universidade Federal de Santa  
Catarina, Centro Tecnológico, Programa de Pós  
Graduação em Engenharia Civil, Florianópolis, 2019.

Inclui referências.

1. Engenharia Civil. 2. Projeto ótimo de  
experimentos. 3. Otimização estocástica. 4.  
Quantificação de incertezas. 5. Gradiente  
estocástico. I. Lopez, Rafael Holdorf. II.  
Universidade Federal de Santa Catarina. Programa de  
Pós-Graduação em Engenharia Civil. III. Título.

André Gustavo Carlon

**Bayesian-based optimization of experiments using  
stochastic gradient methods**

In Partial Fulfillment of the Requirements for the Degree of Doctor in Engineering

Approved dissertation. Florianópolis/SC, Brazil, 4<sup>th</sup> of July, 2019:



Poliana Dias de Moraes, Prof.  
Graduate program coordinator



Rafael Holdorf Lopez, Prof.  
Supervisor

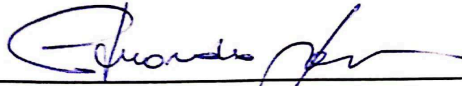


~~Prof. Poliana Dias de Moraes, Dr.~~

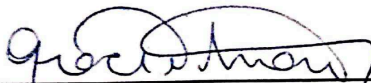
p/ Hélcio Rangel Barreto O ~~Coordenador~~ **PPGEC/UFSC**  
Member (video conference)



p/ Roberta de Queiroz Lima, Prof.  
Member (video conference)



Eduardo Alberto Fancello, Prof.  
Member



Gracieli Dienstmann, Prof.  
Member



# Acknowledgments

This thesis would not be possible without the support of my advisor, prof. Rafael Holdorf Lopez, and the support from prof. Raul Tempone and Luis Espath. Much of this thesis is the result of research developed as a visiting student at the Stochastic Numerics Research Group at KAUST (King Abdullah University of Science and Technology) under the supervision of prof. Raul Tempone. Moreover, I gratefully acknowledge the financial support of CNPq (National Counsel of Technological and Scientific Development) and CAPES (Coordination of Superior Level Staff Improvement).





# Resumo

Os experimentos desempenham um papel importante na ciência, sendo a maneira como observamos o mundo real. No entanto, em muitos casos, os experimentos podem ser caros, demorados ou não fornecer informação o suficiente sobre as quantidades de interesse. Nesses casos, pode ser interessante ajustar o experimento para melhorar sua eficiência. O campo da ciência cujo o objetivo é encontrar a melhor configuração para experimentos é chamado de projeto ótimo de experimento (OED). Utilizamos conceitos de inferência bayesiana para modelar a realização do experimento e, assim, podemos avaliar o desempenho do experimento utilizando o ganho esperado de informação de Shannon (EIG). Quanto mais informativo um experimento for quanto às quantidades de interesse, maior será seu EIG. Assim, o EIG é nossa função de utilidade no OED; a quantidade que queremos maximizar.

Para o caso geral, estimar o EIG pode ser caro. A abordagem direta é aproximar as integrais do EIG usando integração de Monte Carlo (MCI). O estimador resultante é o Monte Carlo de laço duplo (DLMC). O DLMC é caro e é conhecido por ter problemas numéricos, assim, buscamos outro estimador de EIG que possa ter melhor desempenho que o DLMC. Uma alternativa é aproximar a informação posterior ao experimento por uma função gaussiana, resultando no estimador de Monte Carlo com Laplace (MCLA). Este estimador tem um viés da aproximação de Laplace e é, portanto, inconsistente. Como alternativa para estimar o EIG, apresentamos o Monte Carlo de laço duplo com amostragem por importância. A amostragem por importância usa uma estimativa Gaussiana da posterior para obter amostras mais informativas quanto às quantidades de interesse.

Sendo o custo uma questão principal em OED, é importante tornar a otimização o mais eficiente possível. Para otimizar o projeto de experimentos, usamos o gradiente estocástico de descida, reduzindo o custo de cada iteração ao evitar o custo de um MCI. A diminuição no tamanho do passo para o SGD torna a convergência sensível ao tamanho

do passo escolhido. Usamos técnicas no estado-da-arte em otimização estocástica para obter uma otimização robusta e rápida. Uma técnica é a média de Polyak–Ruppert, consistindo de uma média móvel do caminho de otimização. Como a média é mais suave do que o caminho, a diminuição do tamanho do passo necessário para o SGD pode ser relaxada, fornecendo, assim, um algoritmo mais robusto. Para acelerar a convergência, combinamos o SGD com um método de momento, a aceleração do Nesterov. O algoritmo resultante, o ASGD-restart, é robusto e mantém a convergência acelerada em algumas situações.

Para usar os métodos do gradiente estocástico, são necessários estimadores não-enviesados do gradiente verdadeiro. Assim, deduzimos os gradientes dos estimadores DLMC, MCLA e DLMCIS em função do modelo direto e suas derivadas. Além disso, as complexidades dos estimadores de gradiente são apresentadas e o pseudocódigo de seus algoritmos é mostrado para permitir a reprodução.

Para testar o desempenho dos métodos, nós os usamos para resolver quatro exemplos numéricos. O primeiro é um problema de otimização estocástica pura com vinte dimensões tanto para o projeto quanto para as quantidades de interesse. O segundo exemplo é um problema de OED sem um significado físico que criamos para testar as combinações entre os estimadores EIG e os métodos de otimização. Na média de cem execuções, o ASGD-restart com o MCLA resolveu este exemplo com menos de 300 avaliações de modelo, enquanto o DLMC usando a descida de gradiente determinístico precisou de  $2.99 \times 10^7$  avaliações. O terceiro exemplo é o posicionamento ideal de um extensômetro em uma viga para inferir propriedades mecânicas do material do qual a viga é feita. Este exemplo é usado para mostrar que o ótimo encontrado é consistente com a intuição de engenharia. O quarto e último exemplo é a otimização das correntes em um experimento de tomografia por impedância elétrica (EIT) para inferir os ângulos nas camadas de um material laminado composto. A simulação do EIT requer o uso do método de elementos finitos, sendo, portanto, um modelo avançado caro para avaliar. O ASGD-restart usando o gradiente do estimador MCLA

convergiu para soluções ótimas nos quatro casos testados. Até onde vai o conhecimento dos autores, esta é a primeira pesquisa para resolver efetivamente um problema OED com um modelo baseado na análise de elementos finitos.

Nos testes numéricos, usando o gradiente do estimador MCLA para otimização estocástica resultou em convergência rápida em relação ao custo. Além disso, o reinício do ASGD acoplado ao MCLA provou ser uma opção viável para a otimização de experimentos com modelos caros.

**Keywords:** Projeto ótimo de experimentos. Inferência Bayesiana. Otimização estocástica. Método de Laplace.



# Abstract

Experiments play an important role in science, being the way we observe the real world. However, in many cases experiments can be expensive, time-consuming or not provide enough information about the quantities of interest. In such cases, it might be interesting to tune the experiment up as to improve its efficiency. The field of science concerned with finding the best set-up for experiments is called optimal experiment design (OED). We use Bayesian inference concepts to model the experiment evaluation and, thus, are able to evaluate the experiment performance using the Shannon's expected information gain (EIG). The more informative an experiment is about the quantities of interest, the larger is its EIG. Thus, the EIG is our utility function in OED; the quantity we want to maximize.

Estimating the EIG can be expensive. The straightforward approach is to approximate the integrals in the EIG by Monte Carlo Integration (MCI). The resulting estimator is the double-loop Monte Carlo (DLMC). The DLMC is expensive and is known to have numerical issues, thus, we seek other EIG that can have better performance than DLMC. One estimator arises from approximating the posterior by a Gaussian function, the Monte Carlo with Laplace approximation (MCLA). This estimator has a bias from the Laplace approximation, thus, it is inconsistent. As an alternative to estimate the EIG, we present the double-loop Monte Carlo with importance sampling. The importance sampling uses a Gaussian estimate of the posterior to draw more informative samples about the quantities of interest.

Being the cost a main issue in OED, it is important to make the optimization as efficient as possible. To optimize the design of experiments, we use the stochastic gradient descent, reducing the cost of each iteration by the cost of a MCI at the cost of a decreasing step-size. The decrease in step-size for SGD makes the convergence sensible to the step-size chosen. We use state-of-the-art techniques in stochastic optimization to get a robust and fast framework. One technique is the

Polyak–Ruppert averaging, consisting of a moving average of the optimization path. Since the average is smoother than the path, the decrease of the step-size required for SGD can be relaxed, thus, furnishing a more robust algorithm. To accelerate the convergence, we combine the SGD with a momentum method, the Nesterov’s acceleration. The resulting algorithm, the ASGD-restart is robust and maintains the accelerated convergence under some situations.

To use the stochastic gradient methods, unbiased estimators of the true gradient are needed. Thus, we devise the gradients of the DLMC, MCLA, and DLMCIS estimators as a function of the forward model and its derivatives. Moreover, the complexities of the gradient estimators are presented and pseudocode of their algorithms is shown to allow reproduction.

To test the performance of the methods, we use them to solve four numerical examples. The first is a pure stochastic optimization problem with twenty dimensions for both the design and quantities of interest spaces. The second example is an OED problem without a physical meaning that we created to test the combinations between the EIG estimators and the optimization methods. In the average of a hundred runs, the ASGD-restart with MCLA solved this example with less than 300 model evaluations, whereas DLMC using full-gradient descent took  $2.99 \times 10^7$  evaluations. The third example is the optimal positioning of strain-gauges on a beam to infer mechanical properties of the material the beam is made. This example is used to show that the optimum found is consistent with engineering intuition. The fourth and last example is the optimization of the currents in an electrical impedance tomography experiment to infer the angles in the plies of a composite laminate material. The EIT simulation requires the use of finite elements method, thus, being an expensive forward model to evaluate. The ASGD-restart using the gradient of the MCLA estimator converged to optimum solutions in the four cases tested. To best of the author’s knowledge, this is the first research to effectively solve an OED problem with a model based on finite elements analysis.

In the numerical tests, using the gradient of the MCLA estimator for stochastic optimization resulted in fast convergence with respect to cost. Moreover, the ASGD-restart coupled with MCLA has proven to be a viable option for the optimization of experiments with expensive forward models.

**Keywords:** Optimal experimental design. Bayesian inference. Stochastic optimization. Laplace method.





# List of Figures

Figure 1 – Example of fracture toughness testing. Source: Wiki- media commons (1) . . . . .	32
Figure 2 – Electrical impedance tomography example. Source: adapted from Beck et al. (2) . . . . .	32
Figure 3 – Example of the setup for a three-point flexural test.	40
Figure 4 – Prior and posterior pdfs for the three-point flexural experiment. . . . .	41
Figure 5 – For the three-point flexural test: prior and posterior distributions (a), and the integrand of the $D_{KL}$ (b).	43
Figure 6 – The pdfs for prior, posterior, and optimized posterior (a) and the $D_{KL}$ integrand for the non-optimized and for the optimized cases (b). . . . .	44
Figure 7 – The EIG integrand. . . . .	45
Figure 8 – Bias and variance of an estimator . . . . .	47
Figure 9 – Numerical underflow illustrated for the three-point flexural test. . . . .	51
Figure 10 – Avoiding numerical underflow using importance sam- pling illustrated for the three-point flexural test. . .	56
Figure 11 – Path of full-gradient descent and stochastic gradient descent. . . . .	61
Figure 12 – Convergence of FGD and SGD. . . . .	64
Figure 13 – Example of SGD with Polyak–Ruppert averaging. . .	67
Figure 14 – Convergence of vanilla SGD, SGD with Polyak–Ruppert and its average. . . . .	68
Figure 15 – Example of Nesterov’s acceleration. . . . .	69
Figure 16 – Representation of how accelerated steps are taken. .	70
Figure 17 – Convergence of the Nesterov acceleration for different values of $q$ . . . . .	71
Figure 18 – Illustration of restart technique for Nesterov’s accel- eration. . . . .	72

Figure 19 – Convergence of the Nesterov acceleration with the restart technique. . . . .	73
Figure 20 – (Example 1): Convergence of the methods with standard deviations $\sigma_\theta = 0.1$ (a) and $\sigma_\theta = 0.01$ (b). . . .	90
Figure 21 – (Example 2) Convergence to the optimum in relation to iterations for SGD, ASGD, and ASGD-restart with MCLA. . . . .	93
Figure 22 – (Example 2) Convergence to the optimum in relation to iterations for the ASGD-restart with MCLA and with DLMCIS. . . . .	93
Figure 23 – (Example 2) Contour of EIG and optimization ascent paths for the ASGD-restart with MCLA and with DLMCIS. . . . .	94
Figure 24 – (Example 3) Geometry of the Timoshenko beam. . .	95
Figure 25 – From top to bottom, cases 1 to 4 from Example 3 (summarized in Table 4). Expected information gain contours computed with MCLA and optimization ascent paths using SGD, ASGD, and ASGD-restart with MCLA. . . . .	98
Figure 26 – Convergences from cases 1 (a) and 2 (b) using ASGD-restart with tolerance of 1 mm, or relative tolerance of $10^{-4}$ ) (Example 3). . . . .	101
Figure 27 – Prior, posterior, and optimized posterior pdfs for the Young modulus $E$ and the shear modulus $G$ for cases 1 (a), 2 (b), 3 (c), and 4 (d) from Example 3. . . . .	102
Figure 28 – Experimental configuration for EIT with two plies and four electrodes (Example 4). . . . .	104
Figure 29 – Current streamlines (a), optimization path (b), and pdfs of both the initial and optimized configurations (c) for case 1 (Example 4). . . . .	106
Figure 30 – (Example 4) Contour of $\mathcal{I}_{\text{MCLA}}$ with optimization paths for EIT test case 2. . . . .	107
Figure 31 – Current streamlines for guess 2 (a) and pdfs for both guess 1 (b) and guess 2 (c) (Example 4). . . . .	108

Figure 32 – Current streamlines (a), pdfs of initial and optimized configurations (b), and self-convergence to the optimum (c) for case 3. . . . . 110



# List of Tables

Table 1	– Number of evaluations of $\mathbf{g}$ required for each estimator.	86
Table 2	– Mean NCFM over a hundred independent runs to achieve $\ \boldsymbol{\xi}_k - \boldsymbol{\xi}^*\ _2 \leq 0.01$ .	92
Table 3	– Parameters for the Timoshenko beam problem (Example 3).	97
Table 4	– Results from the Timoshenko beam problem (Example 3).	100
Table 5	– Expected information gain using MCLA with $N = 1000$ in Example 4.	109



# List of abbreviations and acronyms

EIT	Electrical impedance tomography
OED	Optimal experimental design
EIG	Expected information gain
MCI	Monte Carlo integration
DLMC	Double loop Monte Carlo
MCLA	Monte Carlo with Laplace approximation
DLMCIS	Double loop Monte Carlo importance sampling
SGD	Stochastic gradient descent
ASGD	Accelerated stochastic gradient descent
FEM	Finite element method
QoI	Quantity of interest
pdf	Probability density function
$D_{KL}$	Kullback Leibler divergence
MAP	Maximum a posteriori
FGD	Full gradient descent
AGD	Accelerated gradient descent
ASGD-restart	Accelerated stochastic gradient descent with restart technique
NCFM	Number of calls of the forward model





# List of symbols

$\theta$	Quantities of interest
$\xi$	Experiment setup
$\mathbf{y}$	Experimental observations
$N_e$	Number of repetitive Experiments
$\mathbf{Y}$	Set of all $N_e$ observations $\mathbf{y}$
$\epsilon$	Instrument error in measurements $\mathbf{y}$
$\Theta$	Space of quantities of interest
$\Xi$	Space of design parameters
$\mathcal{Y}$	Space of experiment observations
$\theta_t$	True value of the quantities of interest
$\Sigma_\epsilon$	Covariance matrix of the error in observations
$\mathbf{g}$	Foward model of the experiment
$I$	Shannon's expected information gain
$h$	Mesh discretization parameter
$\eta$	Rate of convergence of the discretization error in finite element analysis
$\rho$	Cost constant for finite element method
$\mathcal{I}$	Estimator of $I$
$N$	Monte Carlo sample size for the outer-loop
$M$	Monte Carlo sample size for the inner-loop

$f$	function whose expected value is to be minimized
$\hat{\theta}$	Maximum a posteriori
$\Sigma(\cdot)$	Posterior covariance matrix evaluated at $(\cdot)$
$\alpha$	Step-size in optimization
$\mathcal{G}$	Estimator of the gradient of $I$ .
$k$	Optimization iteration
$\bar{\xi}$	Polyak–Ruppert averaging
$L$	Lipschitz constant
$\mu$	Strong-convexity constant
$q$	Acceleration constant
$\lambda, \gamma$	Nesterov’s sequences
$\mathcal{L}$	Importance sampling for the likelihood
TOL	Tolerance for optimum estimate

# Summary

<b>1</b>	<b>INTRODUCTION . . . . .</b>	<b>31</b>
<b>1.1</b>	<b>Motivation . . . . .</b>	<b>31</b>
<b>1.2</b>	<b>Notation . . . . .</b>	<b>35</b>
<b>1.3</b>	<b>Aim and objectives . . . . .</b>	<b>35</b>
<b>1.4</b>	<b>Original contributions . . . . .</b>	<b>36</b>
<b>1.5</b>	<b>Outline of the thesis . . . . .</b>	<b>36</b>
<b>2</b>	<b>OPTIMAL EXPERIMENTAL DESIGN . . . . .</b>	<b>39</b>
<b>2.1</b>	<b>Experiment model . . . . .</b>	<b>39</b>
2.1.1	Bayesian Inference . . . . .	40
2.1.2	Kullback–Leibler Divergence . . . . .	42
2.1.3	Shannon’s Expected Information Gain . . . . .	44
<b>2.2</b>	<b>Expected information gain estimators . . . . .</b>	<b>46</b>
2.2.1	Error analysis . . . . .	46
2.2.2	Monte Carlo integration . . . . .	47
2.2.3	Double-loop Monte Carlo estimator . . . . .	48
2.2.4	Monte Carlo estimator with Laplace approximation . . . . .	52
2.2.5	Double-loop Monte Carlo estimator with Laplace-based importance sampling . . . . .	54
<b>2.3</b>	<b>Chapter summary . . . . .</b>	<b>56</b>
<b>3</b>	<b>OPTIMIZATION . . . . .</b>	<b>59</b>
<b>3.1</b>	<b>Stochastic optimization problem . . . . .</b>	<b>60</b>
<b>3.2</b>	<b>Stochastic gradient descent . . . . .</b>	<b>60</b>
3.2.1	Polyak–Ruppert averaging . . . . .	65
<b>3.3</b>	<b>Nesterov’s accelerated gradient descent . . . . .</b>	<b>68</b>
3.3.1	Restart method . . . . .	70
<b>3.4</b>	<b>Accelerated stochastic gradient descent . . . . .</b>	<b>74</b>
3.4.1	Restart method for the stochastic case . . . . .	76
<b>3.5</b>	<b>Gradients of Monte Carlo estimators for OED . . . . .</b>	<b>77</b>

3.5.1	Gradient of the double-loop Monte Carlo estimator . . .	79
3.5.2	Gradient of the Monte Carlo estimator with Laplace approximation . . . . .	80
3.5.3	Gradient of the double-loop Monte Carlo estimator with importance sampling . . . . .	82
3.5.4	Pseudocode of the complete stochastic optimization framework for OED . . . . .	83
<b>3.6</b>	<b>Chapter summary . . . . .</b>	<b>84</b>
<b>4</b>	<b>EXAMPLES . . . . .</b>	<b>87</b>
<b>4.1</b>	<b>Example 1: Stochastic quadratic function . . . . .</b>	<b>87</b>
<b>4.2</b>	<b>Example 2: OED with quadratic model . . . . .</b>	<b>90</b>
<b>4.3</b>	<b>Example 3: Strain gauge positioning on Timoshenko beam . . . . .</b>	<b>95</b>
4.3.1	Bayesian formulation . . . . .	96
4.3.2	Test cases . . . . .	96
<b>4.4</b>	<b>Example 4: Electrical impedance tomography . . .</b>	<b>101</b>
4.4.1	Bayesian setting . . . . .	103
4.4.2	Numerical tests for EIT . . . . .	104
4.4.2.1	Test case 1 (Configuration with four electrodes and one design variable) . . . . .	105
4.4.2.2	Test case 2 (Configuration with three electrodes and two design variables) . . . . .	107
4.4.2.3	Test case 3 (Configuration with ten electrodes and ten design variables) . . . . .	109
<b>5</b>	<b>CONCLUSION . . . . .</b>	<b>111</b>
<b>5.1</b>	<b>Future research . . . . .</b>	<b>114</b>
	<b>BIBLIOGRAPHY . . . . .</b>	<b>117</b>
	<b>APPENDIX A – DEDUCTION OF MAP AND CO-VARIANCE MATRIX FOR LAPLACE METHODS . . . . .</b>	<b>123</b>

**APPENDIX B – PROOF OF EQ. 3.8 . . . . . 127**

**APPENDIX C – TIMOSHENKO BEAM MODEL . 129**



# 1 Introduction

## 1.1 Motivation

Many fields of science rely heavily on information obtained through experiments. Statistics about quantities of interest are inferred from data resulting from experiment observations, therefore, it is important that experiments provide informative data. For example, in structural engineering, it is useful to have statistical information about the properties of materials, e.g., Young modulus, Poisson modulus, yield stress; so that the engineer can take the best decisions. However, to obtain statistically relevant data, experiments must have large enough samples. For example, if a three-point flexural test is performed on concrete beams to evaluate the fracture toughness of concrete, a significant number of beams must be built beforehand and let to cure for 28 days. Even if a hundred beams are built, cured, and tested, the standard error in the fracture toughness estimate is still the standard deviation of one of the samples divided by ten. To reduce the standard error of the mean by one digit requires increasing the sample size by a hundred times. A three-point bending experiment for fracture toughness determination and its design parameters are presented in Figure 1. Properly defining design parameters can reduce dispersion of the quantity of interest estimation or reduce the cost required to achieve the same precision.

Another example of practical interest is the one of verifying the quality of composite laminates. Some composite materials can have improved mechanical properties in some directions by alternating plies of orthotropic materials with some specific angles between them. One way of testing these laminates is by electrical impedance tomography (EIT), however, the information obtained depends heavily on the currents imposed on the electrodes, as can be seen in our results obtained by Beck et al. (2). The set-up of an EIT experiment is presented in Figure

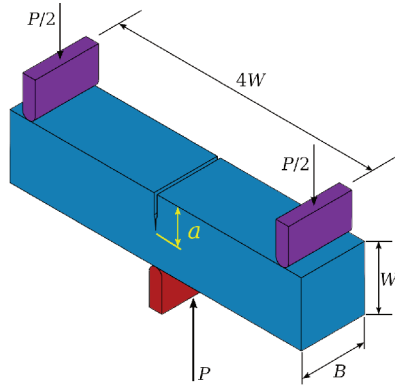


Figure 1 Example of fracture toughness testing. Source: Wikimedia commons (1)

2, where the electrodes are illustrated in black, the first ply in blue and the second ply in red. We optimize the currents in an EIT experiment in Section 4.4.

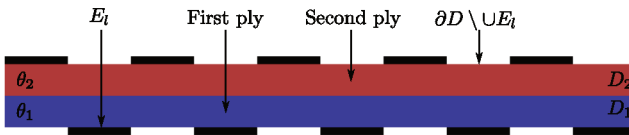


Figure 2 Electrical impedance tomography example. Source: adapted from Beck et al. (2)

Given the difficulties and the costs involved in the experimental process, it might be interesting to optimally tune experiments parameters to maximize their efficiency, the main concern of optimal experimental design (OED). Thus, in OED, we seek the optimum design parameters that maximize some measure of efficiency of an experiment.

The classical optimality criteria for experiments are known as the alphabetic optimality criteria. However, they require the model to be linear with respect to the random parameters. The alphabetic optimality criteria use the Fisher information matrix, a measure of information



inversely proportional to the covariance matrix, to qualify an experiment. The better known of these criteria, D-optimality, consists in minimizing the determinant of the Fisher information matrix (3). According to Chaloner and Verdineli (3), D-optimality is the best optimality criterion to obtain the maximum of information about the quantities of interest. Another popular approach is to minimize the trace of the inverse of the information matrix, known as A-optimality. This is equivalent to minimize the average of the variances of the estimates. In some cases where is the E-optimality; it maximizes the minimum eigenvalue of the Fisher information matrix. However, these methods require knowledge about the model of the experiment. We want to build an OED framework that can optimize experiments with nonlinear black-box forward model. Thus, we opt to measure the performance of an experiment by its Shannon's expected information gain (EIG) (4). The EIG is related to the relative entropy of information and, in the OED context, is a measure on how much information an experiment provides (5). Lindley (6) was the first to use EIG as an utility function for OED. However, for a general experiment with nonlinear model, estimating EIG is cumbersome.

Ryan (7) develop an EIG estimator based on Monte Carlo integration (MCI), however, his estimator requires the evaluation of two nested integrals. We refer to Ryan's estimator as the double-loop Monte Carlo (DLMC). The DLMC estimator has the disadvantages of being expensive to evaluate and numerically unstable. The evaluation of the two nested MCI requires a large number of experiment simulations; if the two outer and inner MCI have sample size of, respectively,  $N$  and  $M$ , the DLMC estimator requires  $N(M + 1)$  forward model evaluations to approximate the EIG. Moreover, for some cases,  $M$  needs to be large to avoid numerical underflow (2). Long et al. (8) propose a method to estimate the EIG using a Laplace approximation, furnishing the Monte Carlo with Laplace approximation (MCLA) estimator. The MCLA estimator does not require the evaluation of one of the two nested integrals, thus, being less expensive. The disadvantage of MCLA is that the Laplace approximation introduces a bias that might not

be acceptable depending on the situation. Beck et al. (2) propose an importance sampling for DLMC that uses Laplace approximation to draw more informative samples, reducing the cost of DLMC without adding the bias of MCLA. The DLMC with the importance sampling is called double-loop Monte Carlo with importance sampling (DLMCIS). In the present thesis we compare the performance of DLMC, MCLA, and DLMCIS in OED.

Even MCLA and DLMCIS being cheaper than DLMC, they still require the evaluation of MCI. Moreover, for DLMCIS, a maximum a posteriori must be found for each outer loop. In the present thesis, we use gradient-based methods to perform optimization. Indeed, gradient based methods have been successfully applied to several engineering fields such as structural optimization (9), multibody dynamics (10), oil well placement (11), among other. Therefore, the gradients of EIG estimators are needed. However, evaluating these gradients each iteration of an optimization procedure might not be feasible.

To perform optimization, we propose the use the stochastic gradient descent (SGD) and some of its variations. The main idea of SGD, proposed by Robbins and Monro (12), is to converge with noisy estimates of the gradient. The effect of the noise is mitigated by a reducing step-size. The SGD algorithm was developed by Robbins and Monro (12) to solve regression problems over large data-sets. With the rise of machine learning methods, SGD became prominent as the main algorithm in the training process (13). In the context of OED, Huan and Marzouk (14) use the Robbins-Monro algorithm to solve OED problems, however, their approach is not efficient; they do not see much improvement in comparison to a BFGS with sample average approximation. To develop on the work of Huan and Marzouk (14), we apply state-of-the-art stochastic optimization techniques to solve OED problems, namely, Polyak–Ruppert averaging, Nesterov’s acceleration, and a restart scheme.

The Polyak–Ruppert averaging is a technique developed indepen-

dently by both Polyak and Juditsky (15), and Ruppert (16). It consists of using a moving average of the optimization path, thus, relaxing the step-size decrease constraint in SGD. The SGD with Polyak–Ruppert averaging is more robust to step-size tuning, converging in cases where *vanilla* SGD would not. On top of that, we use Nesterov’s acceleration (17), resulting in the accelerated stochastic gradient descent (ASGD) algorithm. Nesterov’s accelerated gradient descent is a momentum method that achieves optimal linear convergence for deterministic convex optimization. Its stochastic counterpart does not maintain its linear convergence, however, it still improves on SGD. Moreover, to further improve ASGD we use a restart technique developed for deterministic optimization.

To use the stochastic optimization framework we developed, we need unbiased estimators of the gradient of EIG with respect to the design variables. Thus, we devise the gradients of DLMC, MCLA, and DLMCIS with respect to the design parameters as a function of the forward model and its derivatives. We present the cost of evaluating each EIG gradient estimator.

## 1.2 Notation

The following notation is used throughout this thesis:  $\|\mathbf{a}\|$  is the  $l^2$ -norm of  $\mathbf{a}$ ,  $\det(\mathbf{A})$  is the determinant of the matrix  $\mathbf{A}$ ,  $\langle \mathbf{a}, \mathbf{b} \rangle$  is the inner product between vectors  $\mathbf{a}$  and  $\mathbf{b}$ :  $\mathbf{a}^T \mathbf{b}$ ,  $\|\mathbf{a}\|_{\mathbf{A}}$  is the matrix  $\mathbf{A}$  norm of  $\mathbf{a}$ :  $\sqrt{\mathbf{a}^T \mathbf{A} \mathbf{a}}$ ,  $\mathbb{E}[\cdot]$  is the expected value,  $\mathbb{V}[\cdot]$  is the variance, and  $\dim(\cdot)$  is the dimension.

## 1.3 Aim and objectives

The aim of this thesis is to study numerical methods in both stochastic optimization and uncertainty quantification in order to efficiently solve OED problems.

The objectives to achieve are:

- to implement the EIG estimators DLMC, MCLA, and DLMCIS;
- to devise the gradients of the EIG estimators;
- to develop a state-of-the-art stochastic optimization framework that is robust and efficient;
- to combine the stochastic optimization framework with the EIG gradients devised;
- to evaluate the performance of the methods in numerical examples.

## 1.4 Original contributions

The contributions of the present thesis are both theoretical and practical. From the theoretical perspective, we devised the gradients of the EIG estimators for both deterministic (full-gradient) and stochastic optimization and discussed their numerical complexity. The gradient of the MCLA for stochastic optimization, the main EIG estimator used in the present thesis, does not require any MCI in its evaluation. In the practical sense, as far as the author's knowledge goes, this is the first research to perform OED in an experiment simulated with finite element method (FEM) with nonlinear forward model, what could only be achieved by the efficient OED machinery we devised. Thus, the combination of the optimization and uncertainty quantification methods we propose allows the solution of expensive problems with practical interest that would be unsolvable otherwise.

## 1.5 Outline of the thesis

**Chapter 2:** In this chapter, we introduce key concepts of OED. First, we define the experiment model we use throughout the thesis. Then, we introduce Bayes' theorem and show how it applies to our experiment model. Using the concepts from Bayesian inference, we introduce the Kullback–Leibler divergence and the EIG. To estimate

the EIG, we present three alternatives: DLMC, MCLA, or DLMCIS estimators.

**Chapter 3:** This chapter tackles the optimization part of OED. We formulate the OED as an stochastic optimization problem and introduce the SGD method. Then, we discuss improvements made over the SGD, namely, Polyak–Rupert averaging, Nesterov’s acceleration, and a restart technique to improve acceleration. We combine these features in the main algorithm of this thesis, the ASGD-restart. To use stochastic gradient methods, we devise the gradients of the DLMC, MCLA, and DLMCIS estimators.

**Chapter 4:** In this chapter, we solve four numerical examples. The first example is a 20-dimensional quadratic function with 20 random parameters we devised to compare the performance of the optimization methods presented. The second example is an OED with a forward model quadratic with respect to both the design parameters and the quantities of interest. The idea of the second example is to test the combinations between EIG estimators and optimization methods. The third example is an engineering example used to test the consistency of the OED framework we use; we find the optimal placement of strain-gauges on a beam to infer mechanical properties of its material. This example is used to illustrate that the solution found in OED is consistent with engineering intuition. The fourth and last example is an engineering problem with a FEM-based forward model. In this example, we find the optimal currents to be imposed on the electrodes in an EIT experiment to infer the angles of the plies a composite laminate material.

**Chapter 5:** This chapter is the conclusion. We revise the development of the thesis and discuss the main difficulties of OED. Moreover, we highlight our achievements and discuss the conclusions from the results obtained. We finish the chapter with recommendations for future research.

**Appendix A:** In this Appendix, we show how to obtain the equation for the maximum a posteriori and the covariance matrix to be

used in Laplace approximation and importance sampling.

**Appendix B:** Here, we present a proof that, for our experiment model, the gradients of the EIG estimators are estimators of the gradients of EIG. Thus, the gradients of the estimators presented can be used for OED.

**Appendix C:** The governing equations for the Timoshenko beam model, used in the example in Section 4.3, are deduced from the elasticity equations in this chapter.

## 2 Optimal Experimental Design

The principal goal of the present thesis is the optimization of experiments, which is OED (3). From the perspective of optimization, it is important to define a criterion to measure the efficiency of experiments: the objective function to be maximized. On the present thesis, we use the Shannon's EIG (4) of an experiment to evaluate its performance. Thus, on the next section, we introduce EIG and other concepts related to it.

### 2.1 Experiment model

We model the evaluation of  $N_e$  repetitive experiments as

$$\mathbf{y}_i(\boldsymbol{\xi}, \boldsymbol{\theta}_t, \boldsymbol{\epsilon}_i) = \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}_t) + \boldsymbol{\epsilon}_i, \quad i = 1, \dots, N_e, \quad (2.1)$$

where  $\mathbf{y}_i \in \mathbb{R}^q$  is the vector of experiment observations,  $\boldsymbol{\theta}_t \in \mathbb{R}^d$  is the vector of quantities of interest (QoI) to be recovered,  $\boldsymbol{\xi}$  is the vector with experiment parameters to be optimized,  $\mathbf{g}$  is the experiment model, and  $\boldsymbol{\epsilon}$  is the additive noise from measurements. For  $N_e$  experiments performed with the same setup  $\boldsymbol{\xi}$ , the set of observed data is  $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N_e}$ . Since we cannot observe  $\boldsymbol{\theta}_t$  directly, we use the random variable  $\boldsymbol{\theta}: \Theta \rightarrow \mathbb{R}^m$  with prior distribution  $\pi(\boldsymbol{\theta})$  in lieu of  $\boldsymbol{\theta}_t$ . Thus, through observations  $\mathbf{Y}$ , we calculate statistics about  $\boldsymbol{\theta}$ . In the present thesis, our goal is to find the optimal experimental design  $\boldsymbol{\xi} \in \Xi \subset \mathbb{R}^n$  that provides more information about  $\boldsymbol{\theta}$ , where  $\Xi$  is the space of experiment designs. Moreover, we consider  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\epsilon)$  to be an additive noise, i.e. independent of  $\mathbf{g}$ ,  $\boldsymbol{\xi}$  and  $\boldsymbol{\theta}$ , for some positive-definite and symmetric matrix  $\boldsymbol{\Sigma}_\epsilon$ . This model for experiments is the same used in (8, 2, 18).

For example, consider the case of a single three-point test where the stiffness modulus (E) of some material is estimated, as illustrated in Figure 3. In this case, the QoI ( $\theta$ ) is  $E$ , the measurement  $y$  is the observed

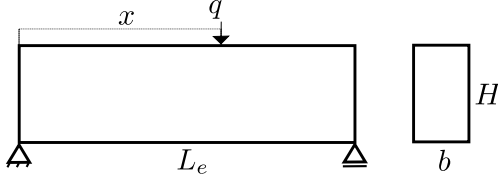


Figure 3 – Example of the setup for a three-point flexural test.

deflection in mm, and  $\epsilon$  is the error in the observation of  $y$ . The design parameter of the experiment to be optimized is  $\xi = x$ , the position where the point load  $q$  is applied, and where the deflection is measured. The model relating  $E$  and  $x$  to the observation  $y$  is  $g(x, E) = \frac{qx}{6L_e EI_e}(2L_e - x)$ , where  $I_e$  is the moment of inertia of the cross-section of the beam. This simple problem is used to illustrate the concepts introduced in the present chapter; the proof that the optimum is at  $x^* = L_e/2$  is trivial.

### 2.1.1 Bayesian Inference

To evaluate the quality of experiments, we use a Bayesian framework of analysis. Thus, in this section, we introduce essential concepts of Bayesian inference. The main idea behind Bayesian inference is to, given some previously known information about a parameter, use new data to update statistics about it. This update is done using Bayes' theorem:

$$\pi(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\xi}) = \frac{p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})\pi(\boldsymbol{\theta})}{p(\mathbf{Y}|\boldsymbol{\xi})}, \quad (2.2)$$

where  $\pi(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\xi})$  is the posterior distribution, the updated probability density function (pdf) of some random variable  $\boldsymbol{\theta}$  given observations  $\mathbf{Y}$ ;  $\pi(\boldsymbol{\theta})$  is the prior distribution, the pdf of  $\boldsymbol{\theta}$  before the experiment;  $p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})$  is the likelihood, the probability of observing  $\mathbf{Y}$  given the previously known prior  $\pi(\boldsymbol{\theta})$ ; and  $p(\mathbf{Y}|\boldsymbol{\xi})$  is the evidence, the probability of  $\mathbf{Y}$  being observed.

On the context of experiments, Bayesian inference is used to estimate a posterior pdf of a QoI given a prior pdf and some observations  $\mathbf{Y}$  provided by the experiment. For example, consider the previously



mentioned three-point flexural test; in Figure 4 we present the pdf for the stiffness both before and after the flexural experiment, i.e., the prior and posterior pdfs of Eq. 2.2. It can be seen that the dispersion is reduced after the experiment, meaning that the experiment provided useful information about the QoI.

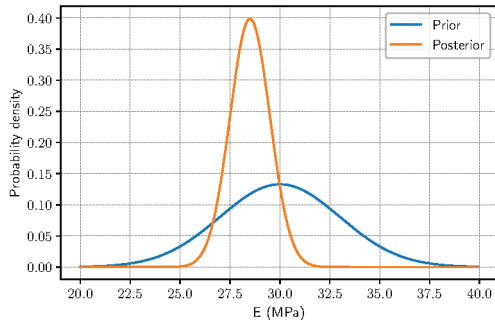


Figure 4 Prior and posterior pdfs for the three-point flexural experiment.

In this example, we modeled the prior knowledge about the QoI as a Gaussian function, however, other distributions can be used as well. The only requirement is that one can sample  $\boldsymbol{\theta}$  from the prior distribution. Using the Bayes' theorem in Eq. 2.2, the posterior can be obtained by multiplying the prior by the division between the likelihood and the evidence. For our experiment model with additive noise in Eq. 2.1, the likelihood of observing  $\mathbf{Y}$  given  $\boldsymbol{\theta}$  and  $\boldsymbol{\xi}$  is a multivariate Gaussian with mean  $\mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta})$  and covariance matrix  $\boldsymbol{\Sigma}_\epsilon$ ,

$$p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) = \det(2\pi\boldsymbol{\Sigma}_\epsilon)^{-\frac{N_e}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N_e} \|\mathbf{y}_i - \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta})\|_{\boldsymbol{\Sigma}_\epsilon^{-1}}^2\right), \quad (2.3)$$

where  $\boldsymbol{\theta}$  is not necessarily the same used to calculate  $\mathbf{Y}$ . Using Eq. 2.3, the likelihood in Eq. 2.2 can be evaluated, yet, the evidence  $p(\mathbf{Y}|\boldsymbol{\xi})$  is not known. To estimate the evidence, we follow the same procedure as Ryan (7). We introduce a new variable  $\boldsymbol{\theta}^*$  that is independent and identically distributed with respect to  $\boldsymbol{\theta}$  and marginalize its likelihood

with respect to  $\boldsymbol{\theta}^*$  as

$$p(\mathbf{Y}|\boldsymbol{\xi}) = \int_{\Theta} p(\mathbf{Y}|\boldsymbol{\theta}^*, \boldsymbol{\xi})\pi(\boldsymbol{\theta}^*)d\boldsymbol{\theta}^*. \quad (2.4)$$

The likelihood in Eq. 2.4 can be calculated from Eq. 2.3 as

$$p(\mathbf{Y}|\boldsymbol{\theta}^*, \boldsymbol{\xi}) = \det(2\pi\Sigma_{\epsilon})^{-\frac{N_{\epsilon}}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N_{\epsilon}} \|\mathbf{y}_i(\boldsymbol{\xi}, \boldsymbol{\theta}, \epsilon_i) - \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}^*)\|_{\Sigma_{\epsilon}^{-1}}^2\right). \quad (2.5)$$

Thus, substituting in the Bayes' equation in Eq. 2.2 the likelihood and the evidence respectively presented in Eqs. 2.3 and 2.4, one can obtain the posterior distribution of the experiment, i.e., the statistics for the QoI after the experiment.

To estimate how much information an experiment provides, we use the Kullback–Leibler divergence ( $D_{KL}$ ) between the prior and posterior pdfs.

### 2.1.2 Kullback–Leibler Divergence

According to Cover and Thomas (19), entropy is a measure of uncertainty of a random variable. Cover and Thomas define the differential entropy of a random variable  $\boldsymbol{\theta}$  with pdf  $f_{\boldsymbol{\theta}}$  as

$$-\int_{\Theta} f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \log f_{\boldsymbol{\theta}}(\boldsymbol{\theta})d\boldsymbol{\theta}. \quad (2.6)$$

The larger the entropy of  $\pi(\boldsymbol{\theta})$  is, the larger its uncertainty with respect to  $\boldsymbol{\theta}$  is. The  $D_{KL}$  is the entropy of a probability measure with respect to another (20), and, for two probability measures  $f_{\boldsymbol{\theta}}$  and  $g_{\boldsymbol{\theta}}$  on  $\boldsymbol{\theta}$  with the same support  $\Theta$ , the  $D_{KL}$  is defined as

$$\begin{aligned} D_{KL}(f_{\boldsymbol{\theta}}(\boldsymbol{\theta})\|g_{\boldsymbol{\theta}}(\boldsymbol{\theta})) &= -\int_{\Theta} f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \log g_{\boldsymbol{\theta}}(\boldsymbol{\theta})d\boldsymbol{\theta} + \int_{\Theta} f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \log f_{\boldsymbol{\theta}}(\boldsymbol{\theta})d\boldsymbol{\theta} \\ &= -\int_{\Theta} f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \log \left(\frac{g_{\boldsymbol{\theta}}(\boldsymbol{\theta})}{f_{\boldsymbol{\theta}}(\boldsymbol{\theta})}\right)d\boldsymbol{\theta} \\ &= \int_{\Theta} \log \left(\frac{f_{\boldsymbol{\theta}}(\boldsymbol{\theta})}{g_{\boldsymbol{\theta}}(\boldsymbol{\theta})}\right) f_{\boldsymbol{\theta}}(\boldsymbol{\theta})d\boldsymbol{\theta}. \end{aligned} \quad (2.7)$$

The larger the  $D_{KL}$  of  $f_{\theta}(\theta)$  with respect to  $g_{\theta}(\theta)$  is, the more informative  $f_{\theta}(\theta)$  is with respect to  $g_{\theta}(\theta)$ . As a way of measuring the efficiency of an experiment, we use the  $D_{KL}$  of the posterior with respect of the prior:

$$D_{KL}(\pi(\theta|\mathbf{Y}, \xi) || \pi(\theta)) = \int_{\Theta} \log \left( \frac{\pi(\theta|\mathbf{Y}, \xi)}{\pi(\theta)} \right) \pi(\theta|\mathbf{Y}, \xi) d\theta. \quad (2.8)$$

For the sake of simplicity, throughout this thesis, the Kullback Leibler divergence between the prior and posterior pdfs is referred simply as Kullback Leibler divergence and denoted as  $D_{KL}$ . In Figure 5, at the left, we present the prior and posterior pdf of the three-point flexural test. The regions where the probability density of the posterior distribution is greater than the one of the prior are shaded in green, whereas the regions where the posterior pdf is less than the prior pdf are shaded in red. In Figure 5, at the right, we present the integrand in Eq. 2.8 over the domain of  $\theta$  for the three-point flexural test. The regions where the integrand is positive are shaded in green, and the regions where the integrand is negative are shaded in red. The more

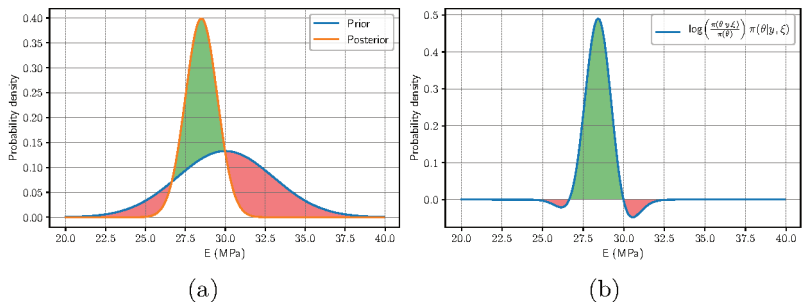


Figure 5 For the three-point flexural test: prior and posterior distributions (a), and the integrand of the  $D_{KL}$  (b).

informative an experiment is, the larger is the integral of the function in Figure 5. Thus, we aim to find the  $\xi^*$  that provides the more informative observations  $\mathbf{Y}^*$ ; the  $\mathbf{Y}^*$  that maximize difference between the green area and the red area in Figure 5.

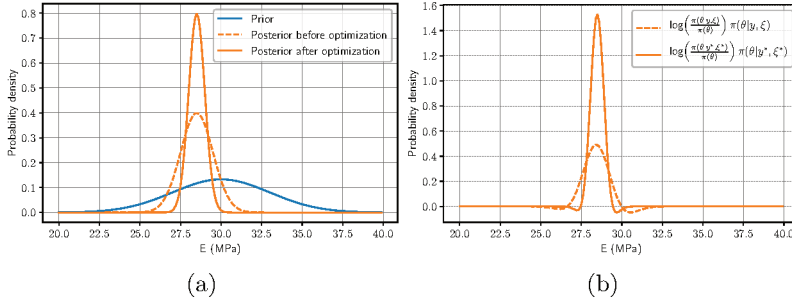


Figure 6 The pdfs for prior, posterior, and optimized posterior (a) and the  $D_{KL}$  integrand for the non-optimized and for the optimized cases (b).

In Figure 6, we present a comparison between the optimized and non-optimized cases for the three-point flexural test. The non-optimized case is evaluated with  $x = L_e/4$ , whereas, in the optimized case, the load and the measurement are in the middle of the beam, i.e.,  $x = x^* = L_e/2$ . In the left plot of Figure 6, it can be observed that the posterior pdf for the optimized configuration is more concentrated than the posterior pdf before before optimization. In the right plot of Figure 6, the integrand of the  $D_{KL}$  is presented for both the optimized and non-optimized configurations. It can be observed that the  $D_{KL}$  for the optimized case is greater than the  $D_{KL}$  for the non-optimized configuration, as can be inferred from the areas under the integrands.

When modeling a real experiment, one might need to consider the noise inherent to experiment observations. To estimate the information gain considering the noise in observations, we use the EIG (4).

### 2.1.3 Shannon's Expected Information Gain

The  $D_{KL}$  does not consider the noise in observations due to measurement uncertainties. To estimate the information gain on this context, we need to marginalize the  $D_{KL}$  with respect to  $\mathbf{Y}$ , thus,

obtaining the Shannon's expected information gain (4) as

$$I = \int_{\mathcal{Y}} \int_{\Theta} \log \left( \frac{\pi(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\xi})}{\pi(\boldsymbol{\theta})} \right) \pi(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\xi}) d\boldsymbol{\theta} p(\mathbf{Y}|\boldsymbol{\xi}) d\mathbf{Y}. \quad (2.9)$$

Lindley (6) is the first to use EIG as an utility function for optimal experimental design. An example of the integrand in Eq. 2.9 with respect to both the QoI and the observations for the three-point flexural test is presented in Figure 7. Estimating  $I$  requires the solution of a double

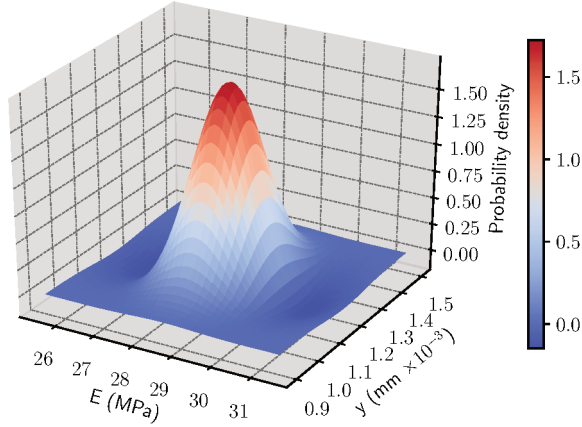


Figure 7 The EIG integrand.

integral over the sample space of the QoI ( $\Theta$ ), and the observations ( $\mathcal{Y}$ ).

Using the Bayes' equation presented in Eq. 2.2, we can rewrite Eq. 2.9 as

$$I = \int_{\Theta} \int_{\mathcal{Y}} \log \left( \frac{p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})}{p(\mathbf{Y}|\boldsymbol{\xi})} \right) p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) d\mathbf{Y} \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}. \quad (2.10)$$

Presenting EIG as in Eq. 2.10 has the advantage that, for our experiment model, the likelihood can be calculated directly using Eq. 2.3. Moreover, using Eq.2.4 to estimate the evidence results in

$$I = \int_{\Theta} \int_{\mathcal{Y}} \log \left( \frac{p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})}{\int_{\Theta} p(\mathbf{Y}|\boldsymbol{\theta}^*, \boldsymbol{\xi}) \pi(\boldsymbol{\theta}^*) d\boldsymbol{\theta}^*} \right) p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) d\mathbf{Y} \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}. \quad (2.11)$$

On this thesis, we use Equations. 2.10 and 2.11 instead of Eq. 2.9 to calculate EIG.

## 2.2 Expected information gain estimators

Evaluating  $I$  in Eq. 2.11 requires the solution of the double integral over  $\boldsymbol{\theta}$  and  $\mathbf{Y}$ , moreover, it requires the solution of the integral used to marginalize the evidence. In most cases, these integrals do not have closed form, thus, numerical methods are needed to approximate them. When the number of random parameters is small, quadrature methods can be used to approximate the integrals in Eq. 2.11, however, as pointed out by Robert and Casella (21), these methods suffer from the curse of dimensionality. Hamada et al. (22) note that using deterministic integration methods to approximate 2.11 becomes unfeasible if the dimensionality of  $\boldsymbol{\theta}$  exceeds three.

### 2.2.1 Error analysis

In this thesis, we focus on the case where the integrals in Eq. 2.11 do not have closed form solution and Monte Carlo integration (MCI) is needed to approximate them. In this section, we introduce Monte Carlo-based estimators for EIG that can deal with high dimensional parameter spaces. We present the error analysis and complexity of the EIG estimators with respect to the evaluation of  $\mathbf{g}$  by FEM with a mesh discretization parameter  $h$ . As  $h \rightarrow 0$ , the discretization bias asymptotically converges as

$$\mathbb{E} [\|\mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}) - \mathbf{g}_h(\boldsymbol{\xi}, \boldsymbol{\theta})\|_2] = \mathcal{O}(h^\eta), \quad (2.12)$$

where  $\mathbf{g}_h$  is the FEM approximation of  $\mathbf{g}$  using  $h$ , and  $\eta > 0$  is the rate of convergence of the discretization error. For the computational effort analysis, we assume that the cost of evaluating  $\mathbf{g}_h$  is  $\mathcal{O}(h^{-\varrho})$ , for the constant  $\varrho > 0$ . Both  $\eta$  and  $\varrho$  are constants that depend only on the numerical approach used in approximating  $\mathbf{g}$  by  $\mathbf{g}_h$ .

The error of estimators can be decomposed in two terms: their bias and variances. The variance of the estimator is a measure of the dispersion from different and independent estimations of the same quantity, e.g., for an estimator  $\mathcal{I}$ , its variance is  $\mathbb{V}[\mathcal{I}] \stackrel{\text{def}}{=} \mathbb{E}[(\mathcal{I} - \mathbb{E}[\mathcal{I}])^2]$ . The bias of an estimator is the difference between its expected value and the true value that is being estimated, e.g., for an estimator  $\mathcal{I}$  of a quantity  $I$ , the bias of  $\mathcal{I}$  is  $|I - \mathbb{E}[\mathcal{I}]|$ . Figure 8 illustrates the bias and variance of an hypothetical estimator of  $I$ , where the curve in blue is the probability density that a value of  $\mathcal{I}$  is estimated. Moreover, the distance between the expected value of  $\mathcal{I}$  and  $I$ , i.e., the bias, is presented, as is the square root of the variance,  $\sigma$ , the standard error of the estimator. The bias of the estimator includes the bias from numerical

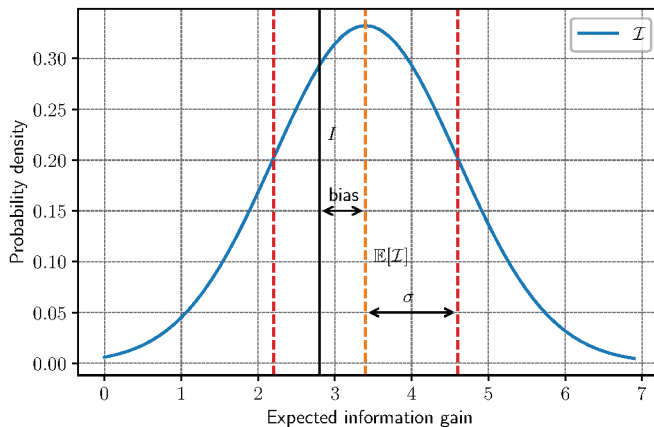


Figure 8 Bias and variance of an estimator

approximation of the model. For further information about the bias and variance of each EIG estimator, the reader is referred to Beck et al.(2).

### 2.2.2 Monte Carlo integration

The MCI is a method for approximating integrals by sampling the integrand and averaging the samples. For example, for  $\theta \in \Theta \subset \mathbb{R}^n$ ,

and  $f : \boldsymbol{\theta} \rightarrow \mathbb{R}$ , the integral of  $f$  over  $\boldsymbol{\theta}$  can be approximated as

$$\int_{\Theta} f(\boldsymbol{\theta}) d\boldsymbol{\theta} \approx \int_{\Theta} d\boldsymbol{\theta} \frac{1}{N} \sum_{i=1}^N f(\boldsymbol{\theta}_i), \quad (2.13)$$

where the  $\boldsymbol{\theta}_i$  are sampled uniformly from  $\Theta$ . If  $f(\boldsymbol{\theta})$  is being integrated over some measure, say  $\pi(\boldsymbol{\theta})$ , its integral can be approximated as

$$\int_{\Theta} f(\boldsymbol{\theta}) \pi(\boldsymbol{\theta}) d\boldsymbol{\theta} \approx \int_{\Theta} \pi(\boldsymbol{\theta}) d\boldsymbol{\theta} \frac{1}{N} \sum_{i=1}^N f(\boldsymbol{\theta}_i), \quad (2.14)$$

where  $\boldsymbol{\theta}$  is sampled from  $\pi(\boldsymbol{\theta})$ . Note that, if  $\pi(\boldsymbol{\theta})$  is a pdf, the integral  $\int_{\Theta} \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}$  evaluates to one. From the strong law of large numbers, the Monte Carlo estimator converges to the real value of the integral as  $N \rightarrow \infty$  (23). Moreover, from the central limit theorem, as  $N \rightarrow \infty$ , the error in approximating an integral by MCI converges to zero with rate  $1/\sqrt{N}$  (23).

### 2.2.3 Double-loop Monte Carlo estimator

The double integral in Eq. 2.10 can be approximated using MCI by sampling  $\boldsymbol{\theta}$  from the prior pdf and  $\mathbf{Y}$  from the likelihood (the measures that the integrands are being integrated over in Eq. 2.10). Then, for  $N$  samples, the Monte Carlo estimator for OED is defined as

$$\mathcal{I}_{MC}(\boldsymbol{\xi}) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N \log \left( \frac{p(\mathbf{Y}_n | \boldsymbol{\theta}_n, \boldsymbol{\xi})}{p(\mathbf{Y}_n | \boldsymbol{\xi})} \right). \quad (2.15)$$

Also, we use Eq. 2.4 and approximate the evidence integral by a MCI as

$$\int_{\Theta} p(\mathbf{Y} | \boldsymbol{\theta}^*, \boldsymbol{\xi}) \pi(\boldsymbol{\theta}^*) d\boldsymbol{\theta}^* \approx \frac{1}{M} \sum_{m=1}^M p(\mathbf{Y} | \boldsymbol{\theta}_m^*, \boldsymbol{\xi}). \quad (2.16)$$

Thus, the DLMC estimator for EIG is defined as

$$\mathcal{I}_{DLMC}(\boldsymbol{\xi}) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N \log \left( \frac{p(\mathbf{Y}_n | \boldsymbol{\theta}_n, \boldsymbol{\xi})}{\frac{1}{M} \sum_{m=1}^M p(\mathbf{Y}_n | \boldsymbol{\theta}_m^*, \boldsymbol{\xi})} \right). \quad (2.17)$$

The first to use DLMC in OED is Ryan (7).



Monte Carlo estimators are generally not biased, however, the DLMC estimator has a bias resulting from the logarithm of the inner loop. However, DLMC is a consistent estimator because the bias goes to zero as the number of inner loop samples  $M$  goes to infinity. The DLMC estimator has bias and variance respectively given by

$$|I - \mathbb{E}[\mathcal{I}_{\text{DLMC}}]| \leq C_{DL,1}h^\eta + \frac{C_{DL,2}}{M} + o(h^\eta) + \mathcal{O}\left(\frac{1}{M^2}\right), \quad \text{and} \quad (2.18)$$

$$\mathbb{V}[\mathcal{I}_{\text{DLMC}}] = \frac{C_{DL,3}}{N} + \frac{C_{DL,4}}{NM} + \mathcal{O}\left(\frac{1}{NM^2}\right) \quad (2.19)$$

for the constants  $C_{DL,1}$ ,  $C_{DL,2}$ ,  $C_{DL,3}$ , and  $C_{DL,4}$  (cf. (2)).

To evaluate the DLMC estimator in Eq. 2.17, one needs the likelihood of observing  $\mathbf{Y}_n$  given  $\boldsymbol{\theta}_n$  and the likelihoods of observing  $\mathbf{Y}_n$  given each  $\boldsymbol{\theta}_m^*$ . Using Eq. 2.3 to estimate the likelihood of observing  $\mathbf{Y}_n$  given  $\boldsymbol{\theta}_n$  furnishes

$$p(\mathbf{Y}_n | \boldsymbol{\theta}_n, \boldsymbol{\xi}) = \det(2\pi\boldsymbol{\Sigma}_\epsilon)^{-\frac{N_\epsilon}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N_\epsilon} \left\| \mathbf{y}_i^{(n)}(\boldsymbol{\xi}) - \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}_n) \right\|_{\boldsymbol{\Sigma}_{\epsilon^{-1}}}^2\right), \quad (2.20)$$

$$= \det(2\pi\boldsymbol{\Sigma}_\epsilon)^{-\frac{N_\epsilon}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N_\epsilon} \left\| \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}_n) + \boldsymbol{\epsilon}_i - \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}_n) \right\|_{\boldsymbol{\Sigma}_{\epsilon^{-1}}}^2\right), \quad (2.21)$$

$$= \det(2\pi\boldsymbol{\Sigma}_\epsilon)^{-\frac{N_\epsilon}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N_\epsilon} \left\| \boldsymbol{\epsilon}_i \right\|_{\boldsymbol{\Sigma}_{\epsilon^{-1}}}^2\right). \quad (2.22)$$

Since  $\mathbf{Y}$  and  $\mathbf{g}$  are evaluated using the same  $\boldsymbol{\theta}$ , the model evaluation  $\mathbf{g}$  in Eq. 2.21 cancels out. Thus, evaluating the likelihood of observing  $\mathbf{Y}_n$  given  $\boldsymbol{\theta}_n$  does not require any model evaluation. However, for the evidence evaluation, we must evaluate the model  $\mathbf{g}$ . For example, consider the likelihood in Eq. 2.3, where  $\mathbf{Y}_n$  is evaluated from  $\boldsymbol{\theta}_n$  and  $\epsilon$  (sampled in the outer loop), and  $\boldsymbol{\theta}_m^*$  is sampled in the inner loop

(independently from  $\theta_n$ ),

$$p(\mathbf{Y}_n | \theta_m^*, \xi) = \det(2\pi \Sigma_\epsilon)^{-\frac{N_e}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N_e} \left\| \mathbf{y}_i^{(n)}(\xi) - \mathbf{g}(\xi, \theta_m^*) \right\|_{\Sigma_\epsilon^{-1}}^2\right). \quad (2.23)$$

Each observation of  $p(\mathbf{Y}_n | \theta_m^*, \xi)$  requires an evaluation of  $\mathbf{g}(\xi, \theta_m^*)$ .

In Algorithm 1, we present the pseudocode for DLMC, where the inputs are the experiment setup,  $\xi$ , the sample-size for the outer loop,  $N$ , and the sample-size for the inner loop,  $M$ . Problem parameters, e.g.,  $\Sigma_\epsilon$ ,  $\pi(\theta)$ ,  $\mathbf{g}$ ,  $N_e$ , are considered to be known. The DLMC returns the estimation of  $I$ ,  $\mathcal{I}_{DLMC}$ .

---

**Algorithm 1** Pseudocode for the DLMC estimator for EIG.

---

```

1: function DLMC( $\xi, N, M$ )
2:   for  $n = 1, 2, \dots, N$  do ▷ Outer loop
3:     Sample  $\theta_n$  from  $\pi(\theta)$ 
4:     Evaluate  $\mathbf{g}(\xi, \theta_n)$ 
5:     for  $i = 1, 2, \dots, N_e$  do
6:       Sample  $\epsilon_i$  from  $\mathcal{N}(0, \Sigma_\epsilon)$ 
7:        $\mathbf{y}_i \leftarrow \mathbf{g}(\xi, \theta_n) + \epsilon_i$ 
8:     end for
9:      $\mathbf{Y}_n \leftarrow \{\mathbf{y}_i\}_{i=1}^{N_e}$ 
10:     $p(\mathbf{Y}_n | \theta_n, \xi) \leftarrow \det(2\pi \Sigma_\epsilon)^{-\frac{N_e}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N_e} \|\epsilon_i\|_{\Sigma_\epsilon^{-1}}^2\right)$ 
11:    for  $m = 1, 2, \dots, M$  do ▷ Inner loop
12:      Sample  $\theta_m^*$  from  $\pi(\theta)$ 
13:      Evaluate  $\mathbf{g}(\xi, \theta_m^*)$ 
14:       $p(\mathbf{Y}_n | \theta_m^*, \xi) \leftarrow \det(2\pi \Sigma_\epsilon)^{-\frac{N_e}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N_e} \left\| \mathbf{y}_i^{(n)}(\xi) - \mathbf{g}(\xi, \theta_m^*) \right\|_{\Sigma_\epsilon^{-1}}^2\right)$ 
15:    end for
16:     $p(\mathbf{Y}_n | \xi) \leftarrow \frac{1}{M} \sum_{m=1}^M p(\mathbf{Y}_n | \theta_m^*, \xi)$ 
17:  end for
18:   $\mathcal{I}_{DLMC}(\xi) \leftarrow \frac{1}{N} \sum_{n=1}^N \log\left(\frac{p(\mathbf{Y}_n | \theta_n, \xi)}{p(\mathbf{Y}_n | \xi)}\right)$ 
19:  return  $\mathcal{I}_{DLMC}(\xi)$ 
20: end function

```

---

From Algorithm 1, it can be seen that the cost of evaluating the DLMC estimator is  $N(M + 1)$  forward model evaluations. Considering that each model evaluation requires the solution of a PDE using FEM

with a mesh of size  $h$ , the computational cost of evaluating the DLMC estimator is of order  $N(M + 1)h^{-e}$ .

The DLMC estimator can suffer from numerical instabilities, namely, *numerical underflow* (2). The marginalization of the likelihood of observing  $\mathbf{Y}$  (evaluated using  $\boldsymbol{\theta}^*$  sampled in the outer loop) given  $\boldsymbol{\theta}$  sampled in the inner loop can result in numerical underflow. If the likelihood is zero for all the  $M$  inner samples, the evidence also becomes zero. If  $g(\boldsymbol{\xi}, \boldsymbol{\theta}_m^*)$  is too distant to each  $\mathbf{y}^{(n)}$ , or if  $\boldsymbol{\Sigma}_\epsilon$  has small eigenvalues, the likelihood can get smaller than floating-point precision. Thus, to avoid numerical underflow,  $M$  needs to be large enough as to guarantee that at least one of the  $M$  likelihoods is not numerically evaluated to zero. If the evidence is evaluated to zero, then Eq. 2.15 cannot be evaluated. Figure 9 illustrates the numerical underflow for the flexural test for five inner loop samples; for each loop the model is evaluated and the likelihood of  $y$  being observed is drawn in red. It can be observed that, for all  $g$  evaluated at each inner loop, the likelihood of observing the  $y^{(n)}$  evaluated at the outer loop is near to zero. In the next sections

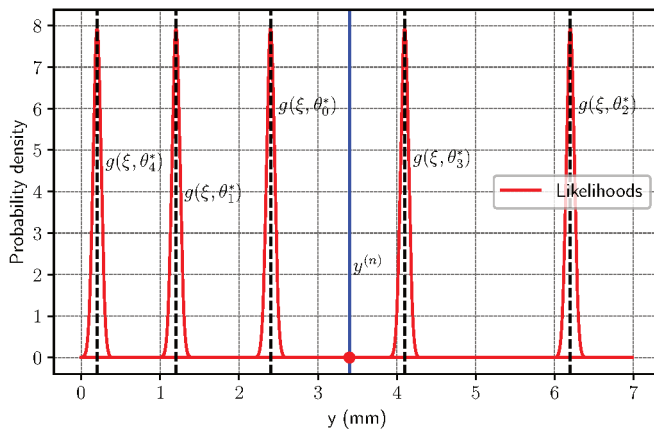


Figure 9 Numerical underflow illustrated for the three-point flexural test.

we present two estimators of EIG that are able to overcome the high cost and numerical instabilities of DLMC.

### 2.2.4 Monte Carlo estimator with Laplace approximation

The cost of solving the two-loop Monte Carlo required for DLMC can be expensive. Since MCI is a method for approximating integrals, one might think of alternative methods to approximate one of the integrals. Long et al. (8) propose the use of the Laplace method to approximate the the logarithm of the posterior distribution by a second-order Taylor expansion, thus, avoiding the evaluation of the evidence. We follow the same approach as Long et al. (8) to devise the Monte Carlo with Laplace method (MCLA) estimator. One advantage of the MCLA is that the approximated posterior pdf is a Gaussian function

$$\pi(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\xi}) \approx \det(2\pi\Sigma(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}}))^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\boldsymbol{\xi}, \mathbf{Y})\|_{\Sigma^{-1}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}})}^2\right), \quad (2.24)$$

where  $\hat{\boldsymbol{\theta}}$  is the maximum a posteriori (MAP) of  $\boldsymbol{\theta}$ , and  $\Sigma(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}})$  is the covariance matrix of the posterior at the MAP. The MAP is the  $\boldsymbol{\theta}$  that maximizes the posterior pdf, i.e., the  $\boldsymbol{\theta}$  more likely to be  $\boldsymbol{\theta}_t$  after the experiment data is considered. Thus,  $\hat{\boldsymbol{\theta}}$  is the  $\boldsymbol{\theta}$  that solves

$$\hat{\boldsymbol{\theta}}(\boldsymbol{\xi}, \mathbf{Y}) \stackrel{\text{def}}{=} \arg \min_{\boldsymbol{\theta} \in \Theta} \left[ \frac{1}{2} \sum_{i=1}^{N_e} \|\mathbf{y}_i - \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta})\|_{\Sigma_e^{-1}}^2 - \log(\pi(\boldsymbol{\theta})) \right]. \quad (2.25)$$

Long et al. (8) show that

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_t + \mathcal{O}_{\mathbb{P}}\left(\frac{1}{\sqrt{N_e}}\right). \quad (2.26)$$

The covariance matrix of the posterior  $\Sigma(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}})$  is the Hessian matrix of the negative logarithm of the posterior pdf evaluated at  $\boldsymbol{\xi}$  and  $\hat{\boldsymbol{\theta}}$ ,

$$\Sigma^{-1}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}}) = N_e \nabla_{\boldsymbol{\theta}}(\mathbf{g}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}}))^T \Sigma_e^{-1} \nabla_{\boldsymbol{\theta}} \mathbf{g}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}}) - \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \log(\pi(\hat{\boldsymbol{\theta}})) + \mathcal{O}_{\mathbb{P}}(\sqrt{N_e}). \quad (2.27)$$

A detailed deduction of  $\hat{\boldsymbol{\theta}}$  and  $\Sigma(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}})$  is presented in Appendix A.

According to Long et al. (8), using the approximation  $\hat{\boldsymbol{\theta}} \approx \boldsymbol{\theta}_t$  yields the Laplace-approximated EIG as

$$I(\boldsymbol{\xi}) = \int_{\Theta} \left[ -\frac{1}{2} \log(\det(2\pi\Sigma(\boldsymbol{\xi}, \boldsymbol{\theta}_t))) - \frac{\dim(\boldsymbol{\theta})}{2} - \log(\pi(\boldsymbol{\theta}_t)) \right] \pi(\boldsymbol{\theta}_t) d\boldsymbol{\theta}_t + \mathcal{O}\left(\frac{1}{N_e}\right). \quad (2.28)$$

Compared to Eq. 2.9, the approximated EIG in Eq. 2.28 has only one integral, not requiring the integration over  $\mathbf{Y}$  nor the integral to calculate the evidence. This EIG with Laplace approximation is consistent with Chaloner and Verdinelli (3), according to whom, maximizing EIG is equivalent to minimizing the determinant of the posterior covariance matrix.

Estimating  $I$  in Eq. 2.28 by MCI results in the MCLA:

$$\mathcal{I}_{\text{MCLA}}(\boldsymbol{\xi}) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N \left[ -\frac{1}{2} \log(\det(2\pi \boldsymbol{\Sigma}(\boldsymbol{\xi}, \boldsymbol{\theta}_n))) - \frac{\dim(\boldsymbol{\theta})}{2} - \log(\pi(\boldsymbol{\theta}_n)) \right]. \quad (2.29)$$

According to Beck et al. (2), the bias and variance of the MCLA estimator are, respectively,

$$|I - \mathbb{E}[\mathcal{I}_{\text{MCLA}}]| \leq C_{LA,1} h^\eta + \frac{C_{LA,2}}{N_e} + o(h^\eta), \quad \text{and} \quad (2.30)$$

$$\mathbb{V}[\mathcal{I}_{\text{MCLA}}] = \frac{C_{LA,3}}{N} \quad (2.31)$$

for the constants  $C_{LA,1}$ ,  $C_{LA,2}$ , and  $C_{LA,3}$ . Although DLMC is a consistent estimator, MCLA is not, because the bias from the Laplace approximation does not vanish as the number of samples increases, being dependent on the number of repetitive experiments  $N_e$ . Hence, in the numerical example section we investigate whether the bias of MCLA affects the optimization.

Algorithm 2 presents the pseudocode for the MCLA estimator. Since MCLA does not have an inner loop, it only needs the outer loop sample-size,  $N$ . The evaluation of Eq. 2.27 requires a Jacobian of the model  $\mathbf{g}$  with respect to the parameters  $\boldsymbol{\theta}$ ; the main cost of MCLA evaluation.

If forward-Euler is used for the estimation of the Jacobian in Eq. 2.27, the evaluation of the MCLA estimator cost has order  $N(\dim(\boldsymbol{\theta}) + 1)h^{-\ell}$ . Thus, in comparison to DLMC, MCLA requires lower computational effort if  $\dim(\boldsymbol{\theta})$  is less than  $M$ . For most cases, the  $M$  required

---

**Algorithm 2** Pseudocode for the MCLA estimator for EIG.

---

```

1: function MCLA( $\xi, N$ )
2:   for  $n = 1, 2, \dots, N$  do ▷ Outer loop
3:     Sample  $\theta_n$  from  $\pi(\theta)$ 
4:     Evaluate  $\nabla_{\theta} g(\xi, \theta_n)$ 
5:     Use  $\nabla_{\theta} g(\xi, \theta_n)$  to evaluate  $\Sigma(\xi, \theta_n)$  using Eq. 2.27
6:   end for
7:    $\mathcal{I}_{MCLA}(\xi) \leftarrow \frac{1}{N} \sum_{n=1}^N \left[ -\frac{1}{2} \log(\det(2\pi\Sigma(\xi, \theta_n))) - \frac{\dim(\theta)}{2} - \log(\pi(\theta_n)) \right]$ 
8:   return  $\mathcal{I}_{MCLA}(\xi)$ 
9: end function

```

---

to achieve a certain precision with DLMC is large, therefore, for these cases, MCLA is computationally more efficient than DLMC.

### 2.2.5 Double-loop Monte Carlo estimator with Laplace-based importance sampling

Based on the idea proposed by Beck et al. (2), we use an importance sampling in the MCI of the evidence. Instead of sampling  $\theta^*$  from  $\pi(\theta)$ , we sample  $\theta^*$  from  $\tilde{\pi}(\theta)$ , where  $\tilde{\pi}(\theta) \sim \mathcal{N}(\hat{\theta}, \Sigma(\hat{\theta}))$ , and  $\hat{\theta}$  and  $\Sigma$  are given in Eqs. 2.25 and 2.27. Therefore, we are using a Gaussian approximation of the posterior resulting from a second-order Taylor expansion of the logarithm of the posterior at its MAP to draw more informative samples for the inner loop. The advantage of using the importance sampling is that it avoids the approximation error from the Laplace approximation and the numerical underflow problem from DLMC. The DLMCIS estimator is defined as

$$\mathcal{I}_{DLMCIS}(\xi) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N \log \left( \frac{p(\mathbf{Y}_n | \theta_n, \xi)}{\frac{1}{M} \sum_{m=1}^M \mathcal{L}(\mathbf{Y}_n; \xi; \theta_m^*)} \right), \quad (2.32)$$

where

$$\mathcal{L}(\mathbf{Y}; \xi; \theta) = \frac{p(\mathbf{Y} | \theta, \xi) \pi(\theta)}{\tilde{\pi}(\theta)}. \quad (2.33)$$

The bias and variance of the DLMCIS estimator are deduced on the original paper (2) and are proven to be the same as of DLMC for a given tolerance, thus the DLMCIS estimator is also consistent. However, to achieve the desired tolerance, the inner loop sample-size  $M$  is

significantly smaller for DLMCIS than for DLMC. Like in MCLA, the evaluation of Eq. 2.27 requires a Jacobian of the model  $\mathbf{g}$  with respect to the parameters  $\boldsymbol{\theta}$ , moreover, finding  $\hat{\boldsymbol{\theta}}$  requires solving the optimization problem in Eq. 2.25. We use a steepest descent search to find  $\hat{\boldsymbol{\theta}}$ . For that we use the gradient of the function to be minimized in Eq. 2.25,

$$\hat{\nabla}_{\boldsymbol{\theta}} \stackrel{\text{def}}{=} -(\nabla_{\boldsymbol{\theta}} \mathbf{g}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}}))^T \boldsymbol{\Sigma}_{\epsilon}(\mathbf{Y} - \mathbf{g}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}})) - \nabla_{\boldsymbol{\theta}} \log \pi(\hat{\boldsymbol{\theta}}). \quad (2.34)$$

---

**Algorithm 3** Pseudocode for finding MAP using steepest descent.

---

```

1: function FINDMAP( $\boldsymbol{\xi}, \boldsymbol{\theta}, \mathbf{Y}, \alpha_{\boldsymbol{\theta}}, \text{TOL}$ )
2:    $\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}$ 
3:   for  $j = 1, 2, 3..$  do
4:      $\hat{\nabla}_{\boldsymbol{\theta}} \leftarrow -(\nabla_{\boldsymbol{\theta}} \mathbf{g}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}}))^T \boldsymbol{\Sigma}_{\epsilon}(\mathbf{Y} - \mathbf{g}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}})) - \nabla_{\boldsymbol{\theta}} \log \pi(\hat{\boldsymbol{\theta}})$ 
5:      $\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} - \alpha_{\boldsymbol{\theta}} \hat{\nabla}_{\boldsymbol{\theta}}$ 
6:     if  $|\hat{\nabla}_{\boldsymbol{\theta}}|_2 > \text{TOL}$  then
7:       Break
8:     end if
9:   end for
10:  return  $\hat{\boldsymbol{\theta}}$ 
11: end function

```

---

If forward-Euler is used for the estimation of the Jacobians in Eq. 2.27 and Algorithm 3, the evaluation of the DLMCIS estimator costs  $N((C_{MAP} + 1)(\dim(\boldsymbol{\theta}) + 1) + M)h^{-\rho}$ , where  $C_{MAP}$  is the number of iterations to estimate  $\hat{\boldsymbol{\theta}}$  in Eq. in Algorithm 3.

Besides the advantage of reducing the number of samples of the inner loop,  $M$ , the DLMCIS estimator is more robust to numerical underflow than the DLMC estimator. The change of measure in the sampling of  $\boldsymbol{\theta}^*$  guarantees that  $\mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}^*)$  is close to  $\mathbf{Y}$ , evaluated using the  $\boldsymbol{\theta}$  sampled in the outer loop. Thus, it is not likely that, for all inner loops, the likelihood of observing  $\mathbf{Y}$  given  $\boldsymbol{\theta}^*$  is numerically evaluated to zero. Figure 10 illustrates how the importance sampling mitigates the numerical underflow problem for the three-point flexural point; the likelihoods of observing  $y^{(n)}$  for each  $g$  are larger than zero.

A pseudocode for the DLMCIS is presented in Algorithm 4, where the line where importance sampling happens is shaded in gray.

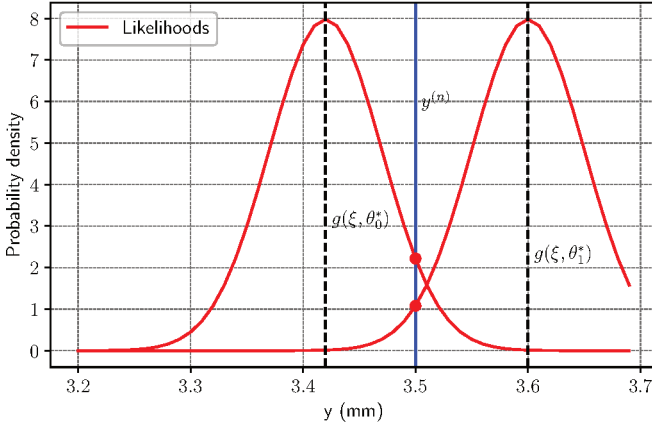


Figure 10 Avoiding numerical underflow using importance sampling illustrated for the three-point flexural test.

### 2.3 Chapter summary

In this chapter, we introduced key concepts related to Bayesian inference and OED. In Section 2.1, we presented the model of experiments with additive noise used in this thesis. Also, in Section 2.1, we defined a measure of efficiency of experiments:  $D_{KL}$  between the prior and posterior pdfs. Moreover, we presented, from an experiment with additive noise, the EIG, the quantity we want to maximize.

In Section 2.2, we presented EIG estimators based in MCI. Applying MCI in EIG furnishes the DLMC estimator, introduced by Ryan (7). The DLMC estimator is consistent and can estimate EIG for any general case, however, depending on the problem it can be expensive and unstable (2). As an alternative to DLMC, we presented the MCLA estimator proposed by Long et al. (8). This estimator uses the Laplace approximation of the posterior, thus, avoiding the inner-loop in DLMC. The resulting algorithm, MCLA, is an inexpensive estimator for EIG, at the expense of the introduction of a bias. This bias resulting from the Laplace approximation does not vanish as sample-sizes go to infinity, thus, MCLA is an inconsistent estimator. This fact is then investigated



---

**Algorithm 4** Pseudocode for the DLMCIS estimator for EIG.
 

---

```

1: function DLMCIS( $\xi, N, M$ )
2:   for  $n = 1, 2, \dots, N$  do ▷ Outer loop
3:     Sample  $\theta_n$  from  $\pi(\theta)$ 
4:     for  $i = 1, 2, \dots, N_e$  do
5:       Sample  $\epsilon_i$  from  $\mathcal{N}(0, \Sigma_\epsilon)$ 
6:        $\mathbf{y}_i \leftarrow \mathbf{g}(\xi, \theta_n) + \epsilon_i$ 
7:     end for
8:      $\mathbf{Y}_n \leftarrow \{\mathbf{y}_i\}_{i=1}^{N_e}$ 
9:      $p(\mathbf{Y}_n | \theta_n, \xi) \leftarrow \det(2\pi\Sigma_\epsilon)^{-\frac{N_e}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N_e} \|\epsilon_i\|_{\Sigma_\epsilon}^2\right)$ 
10:    Find  $\hat{\theta}_n(\xi, \mathbf{Y}_n)$  using Algorithm 3
11:    Evaluate  $\nabla_{\theta} \mathbf{g}(\xi, \hat{\theta}_n)$ 
12:    Use  $\nabla_{\theta} \mathbf{g}(\xi, \hat{\theta}_n)$  to evaluate  $\Sigma(\xi, \hat{\theta}_n)$  using Eq. 2.27
13:    for  $m = 1, 2, \dots, M$  do ▷ Inner loop
14:      Sample  $\theta_m^*$  from  $\tilde{\pi}(\theta) \sim \mathcal{N}(\hat{\theta}_n, \Sigma(\xi, \hat{\theta}_n))$  ▷
15:      Importance sampling
16:      Evaluate  $\mathbf{g}(\xi, \theta_m^*)$ 
17:       $p(\mathbf{Y}_n | \theta_m^*, \xi) \leftarrow \det(2\pi\Sigma_\epsilon)^{-\frac{N_e}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N_e} \|\mathbf{y}_i^{(n)}(\xi) - \mathbf{g}(\xi, \theta_m^*)\|_{\Sigma_\epsilon}^2\right)$ 
18:       $\mathcal{L}(\mathbf{Y}_n; \xi; \theta_m^*) \leftarrow p(\mathbf{Y}_n | \theta_m^*, \xi) \pi(\theta_m^*) / \tilde{\pi}(\theta_m^*)$ 
19:    end for
20:     $p(\mathbf{Y}_n | \xi) \leftarrow \frac{1}{M} \sum_{m=1}^M \mathcal{L}(\mathbf{Y}_n; \xi; \theta_m^*)$ 
21:  end for
22:   $\mathcal{I}_{DLMCIS}(\xi) \leftarrow \frac{1}{N} \sum_{n=1}^N \log\left(\frac{p(\mathbf{Y}_n | \theta_n, \xi)}{p(\mathbf{Y}_n | \xi)}\right)$ 
23:  return  $\mathcal{I}_{DLMCIS}(\xi)$ 
24: end function

```

---

in the numerical example section. For the cases where the bias of MCLA is not acceptable, we introduced an importance sampling based on the Laplace approximation, proposed by Beck et al (2), resulting in the DLMCIS estimator.



### 3 Optimization

We formulate the optimization problem of OED as

$$\text{find } \boldsymbol{\xi}^* = \arg \max_{\boldsymbol{\xi} \in \Xi} (I(\boldsymbol{\xi})). \quad (3.1)$$

In the present thesis, we assume that  $I$  is continuous and differentiable, thus, we focus on gradient-based methods to search for local maxima. The main idea behind gradient methods is to use information on the derivatives of the objective function to iteratively converge to local optima. Considering the problem in Eq. 3.1, if the function  $I$  is continuous, smooth, and convex, we can state that the iterative process

$$\boldsymbol{\xi}_k = \boldsymbol{\xi}_{k-1} + \alpha \nabla_{\boldsymbol{\xi}} I(\boldsymbol{\xi}_{k-1}) \quad (3.2)$$

produces a sequence  $\{I(\boldsymbol{\xi}_k)\}_{k \geq 0}$  that monotonically increases to  $I(\boldsymbol{\xi}^*)$ , given that  $\alpha$  is small enough (24). Moreover, if  $I$  is convex with respect to  $\boldsymbol{\xi}$ , the sequence  $\{|\boldsymbol{\xi}_k - \boldsymbol{\xi}^*|\}_{k \geq 0}$  converges to zero in the Cauchy sense independently of the starting point  $\boldsymbol{\xi}_0$ . We refer to Eq. 3.2 as the full-gradient descent (FGD) method. Note that, since we are maximizing  $I$ , the optimizer performs steps in the direction of the gradient, therefore, it is actually ascending. However, for the sake of simplicity, we refer to gradient-based algorithms as descent algorithms, even when they are employed for maximization.

To perform gradient-based optimization for the OED problem, each iteration requires the evaluation of the gradient of  $I$ ,

$$\nabla_{\boldsymbol{\xi}} I(\boldsymbol{\xi}) = \nabla_{\boldsymbol{\xi}} \int_{\Theta} \int_{\mathcal{Y}} \log \left( \frac{p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})}{p(\mathbf{Y}|\boldsymbol{\xi})} \right) p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) d\mathbf{Y} \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}. \quad (3.3)$$

One aspect of Eq. 3.1 is that each evaluation of  $\nabla_{\boldsymbol{\xi}} I$  requires the solution of an uncertainty quantification task: the evaluation of the two integrals in Eq. 3.3 and the integral in Eq. 2.4, needed to estimate the evidence. In cases where these integrals cannot be directly evaluated,

one might use the DLMC, MCLA, or DLMCIS estimators presented in Chapter 2. In these cases, the cost of evaluating these estimations every iteration can be prohibitive. For example, if  $K$  iterations are needed to find an optimum using the DLMC estimator, and forward finite differences are used to approximate the gradient, the total cost of performing optimization is  $K(\dim(\boldsymbol{\xi}) + 1)NMh^{-e}$ , where  $K$ ,  $N$ , and  $M$  can be large.

### 3.1 Stochastic optimization problem

To alleviate the computational burden of the optimization, we reformulate the optimization problem in Eq. 3.1 as a stochastic optimization problem,

$$\text{find } \boldsymbol{\xi}^* = \arg \max_{\boldsymbol{\xi} \in \Xi} (\mathbb{E}_{\boldsymbol{\theta}, \mathbf{Y}}[f(\boldsymbol{\xi}, \boldsymbol{\theta}, \mathbf{Y})]), \quad (3.4)$$

where

$$f(\boldsymbol{\xi}, \boldsymbol{\theta}, \mathbf{Y}) = \log \left( \frac{p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})}{p(\mathbf{Y}|\boldsymbol{\xi})} \right). \quad (3.5)$$

Note that  $I(\boldsymbol{\xi}) = \mathbb{E}_{\boldsymbol{\theta}, \mathbf{Y}}[f(\boldsymbol{\xi}, \boldsymbol{\theta}, \mathbf{Y})]$ . The problem of finding minima of the expected value of functions, i.e., stochastic optimization, is well-known, and much has been done to reduce the cost of solving it (13, 25, 26). One of the most popular methods for the solution of Eq. 3.4 is the Robbins–Monro algorithm (12), also known as the stochastic gradient descent.

### 3.2 Stochastic gradient descent

The SGD is a method of finding minima (or maxima) of the expected value of functions. In our case, the SGD for OED is

$$\boldsymbol{\xi}_k = \boldsymbol{\xi}_{k-1} + \alpha_k \nabla_{\boldsymbol{\xi}} f(\boldsymbol{\xi}_{k-1}, \boldsymbol{\theta}_k, \mathbf{Y}_k), \quad (3.6)$$

where the step-size  $\alpha$  at iteration  $k$  is a decreasing sequence that satisfies

$$\sum_{i=0}^{\infty} \alpha_i = \infty \quad \text{and} \quad \sum_{i=0}^{\infty} \alpha_i^2 < \infty. \quad (3.7)$$

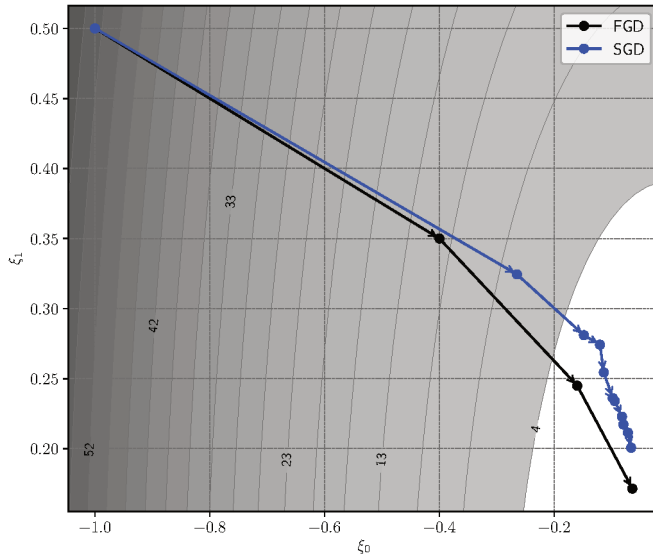


Figure 11 Path of full-gradient descent and stochastic gradient descent.

Robbins and Monro (12) suggest the use of  $\alpha_k = \alpha_0/k$  as the step-size sequence.

The rationale behind SGD is that, instead of spending computational effort in estimating the true gradient, noisy gradients are used in the optimization process. Then, the decreasing step-size guarantees that the effect of the noise in the optimization also decreases. Figure 11 presents the optimization path for both FGD (deterministic steepest descent) and SGD. It can be observed that SGD walks erratically, however, it still moves in the direction of the optimum. Note that, even though FGD approximates the optimum faster than SGD, the cost of each FGD iteration might be several orders of magnitude larger than the one of SGD.

The SGD can be used as long as  $\nabla_{\xi} f$  is an unbiased estimator of  $\mathbb{E}_{\theta, \mathbf{Y}}[\nabla_{\xi} f]$  (12). Throughout the present thesis, we assume that gradients with respect to  $\xi$  and integrals with respect to  $\theta$  and  $\mathbf{Y}$  are

interchangeable, that is,

$$\begin{aligned}
 \nabla_{\boldsymbol{\xi}} \mathbb{E}_{\boldsymbol{\theta}, \mathbf{Y}} [f(\boldsymbol{\xi}, \boldsymbol{\theta}, \mathbf{Y})] &= \nabla_{\boldsymbol{\xi}} \int_{\Theta} \int_{\mathcal{Y}} \log \left( \frac{p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})}{p(\mathbf{Y}|\boldsymbol{\xi})} \right) p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) d\mathbf{Y} \pi(\boldsymbol{\theta}) d\boldsymbol{\theta} \\
 &= \int_{\Theta} \int_{\mathcal{Y}} \nabla_{\boldsymbol{\xi}} \log \left( \frac{p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})}{p(\mathbf{Y}|\boldsymbol{\xi})} \right) p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) d\mathbf{Y} \pi(\boldsymbol{\theta}) d\boldsymbol{\theta} \\
 &= \mathbb{E}_{\boldsymbol{\theta}, \mathbf{Y}} [\nabla_{\boldsymbol{\xi}} f(\boldsymbol{\xi}, \boldsymbol{\theta}, \mathbf{Y})].
 \end{aligned} \tag{3.8}$$

Indeed, for the experiment model in Eq. 2.1, Eq. 3.8 is true. A proof is presented in Appendix B.

Robbins and Monro (12) propose the SGD method and prove that it converges given that the function  $f$  is convex, continuous, and differentiable with respect to  $\boldsymbol{\xi}$ , and that conditions in Eq. 3.7 are satisfied. In comparison to Eq. 3.3, the iterative procedure in Eq. 3.6 does not require the evaluation of the expectation over  $\boldsymbol{\theta}$  and  $\mathbf{Y}$ . For example, applying the SGD to the OED problem with DLMC and forward finite-differences reduces the cost from  $K(\dim(\boldsymbol{\xi}) + 1)N M h^{-\epsilon}$  to  $K(\dim(\boldsymbol{\xi}) + 1)M h^{-\epsilon}$ , i.e., the optimization cost is  $N$  times lower than the cost for the full-gradient case.

We denote the estimators of the gradient of  $f$  with respect to  $\boldsymbol{\xi}$  as  $\mathcal{G}$ , thus, the SGD iterative equation for the OED problem is written as

$$\boldsymbol{\xi}_k = \boldsymbol{\xi}_{k-1} + \alpha_k \mathcal{G}(\boldsymbol{\xi}_{k-1}, \boldsymbol{\theta}_k, \mathbf{Y}_k) \tag{3.9}$$

where  $\boldsymbol{\theta}_k$  is sampled independently from the prior each iteration and  $\mathbf{Y}_k$  is evaluated from Eq. 2.1 using  $N_e$  independent samples of  $\epsilon$ . Note that, since we are trying to find the maximizer of  $I$ , we move in the direction of  $\mathcal{G}$ , and not in its opposite direction. In Eq. 3.5, the evidence still needs to be evaluated. Thus, we have an equivalent gradient estimator  $\mathcal{G}$  for each of the three expected information gain estimators presented in Section 2.2 (DLMC, MCLA, DLMCIS). In DLMC and DLMCIS the evidence is estimated by another MCI, whereas, in MCLA, because of the approximation that the posterior is a Gaussian, the evidence need not be evaluated.

In Algorithm 5, we present the pseudocode for SGD. For the sake of notation consistency, we use  $\boldsymbol{\theta}$  and  $\mathbf{Y}$  as the random parameters in the gradient estimator  $\mathcal{G}$ , however, one could use any number of random parameters as long as  $\mathcal{G}$  is an unbiased estimator of the gradient of the objective function. One of the advantages of SGD is that the

---

**Algorithm 5** Pseudocode for *vanilla* SGD.

---

```

1: procedure SGD( $\boldsymbol{\xi}_0, \alpha_0$ )
2:   for  $k = 1, 2, \dots$  do
3:     Sample random variables, e.g.,  $\boldsymbol{\theta}_k, \mathbf{Y}_k, \dots$ 
4:      $\alpha_k \leftarrow \frac{\alpha_0}{k}$ 
5:      $\boldsymbol{\xi}_k \leftarrow \boldsymbol{\xi}_{k-1} + \alpha_k \mathcal{G}(\boldsymbol{\xi}_{k-1}, \boldsymbol{\theta}_k, \mathbf{Y}_k)$ 
6:   end for
7:    $\hat{\boldsymbol{\xi}} \leftarrow \boldsymbol{\xi}_k$ 
8: end procedure

```

---

cost per iteration does not scales up with the number of dimensions in the parameters space, i.e., SGD does not suffer from the curse of dimensionality. For example, using surrogate models (e.g., polynomial chaos expansions, kriging) to approximate EIG for a 20-dimensional problem like the one in Section 4.1 is unfeasible.

In gradient-based stochastic optimization, noisy estimates of the gradient are used to perform steps. If, throughout optimization, the gradient of the objective function converges to a null vector, and the noise in the estimates does not, the relative error in gradient estimates goes to infinity. Thus, if the order of magnitude of the noise does not vary much in the optimization domain, the error in gradient estimation in early iterations is relatively small when compared to the gradient size. We divide the behavior of the SGD in two phases: the preasymptotic phase, where the true-gradient is large with respect to the error, and the asymptotic phase, where the optimizator is already close to the optimum. As optimization evolves, the error dominates the gradient estimates and step-size reduction mitigates the influence of the noise.

The SGD has proven convergence for  $\alpha_k = \alpha_0/k$ , however, in practice, the decrease in the step-size can be faster than ideal, i.e., at

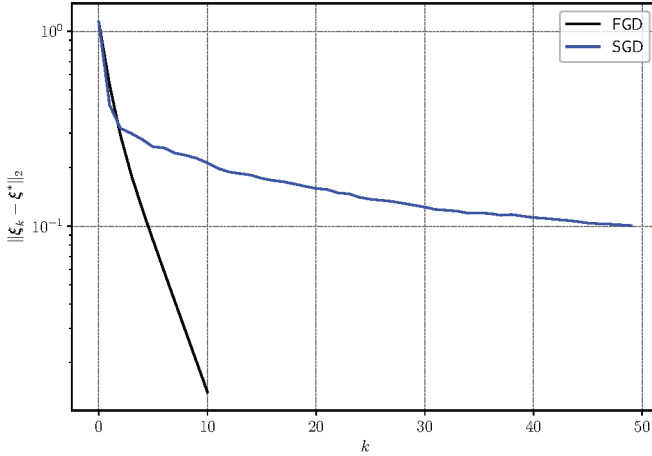


Figure 12 Convergence of FGD and SGD.

later iterations, the step-size becomes small. According to Nemirovski, (27), the convergence depends on the initial step-size being  $\alpha_0 = 1/L$  for strongly-convex objective functions with  $L$ -Lipschitz gradient. Since, for the general case,  $L$  is not known, the initial step-size must be tuned arbitrarily. Moreover, as pointed out by Moulines and Bach (28), SGD convergence is very sensible to the tuning of the initial step-size  $\alpha_0$ . If the initial step-size is too small,  $\alpha_0 \ll 1/L$ , the step-size gets too small before the algorithm gets close to the optimum, specially if the starting point is far from the optimum. If, otherwise, the step-size is too large,  $\alpha_0 > 1/L$ , the algorithm might not converge.

Figure 12 presents the convergence of the distance to the optimum for both FGD and SGD. It can be observed that SGD converges slower each iteration. After a hundred iterations the step-size is reduced by a hundred, thus, the algorithm gets slower each iteration. The SGD algorithm converges in the asymptotic phase, however, the reduction in step-size during the non-asymptotic phase does not allow SGD to get in the asymptotic phase.

Huan and Marzouk (14) use SGD in OED; however, they also increase bias by using small samples for the inner-loop sample size



$M$ . According to Beck et al. (2), DLMC can suffer from numerical underflow if  $M$  is not large enough; thus, choosing small  $M$  might cause a numerical error for some experiments. Huan and Marzouk (14) conclude that SGD converges slightly better than BFGS with a sample average approximation. Moreover, Huan and Marzouk (14) acknowledge the difficulty of finding a step-size for the SGD algorithm. Thus, the problem of efficiently using SGD in OED is still unsolved. To tackle this issue, we devise a robust and efficient stochastic optimization framework for OED, which we consider the main contribution of this thesis.

In the next sections, we present a discussion on how to improve over Robbins–Monro algorithm and get globally-convergent algorithms that are robust to sub-optimal tuning of the initial step-size. Namely, we use the Polyak–Ruppert averaging to relax the step-size decrease condition, improving the convergence in the asymptotic phase, and a Nesterov’s acceleration that improves the convergence in the non-asymptotic phase.

### 3.2.1 Polyak–Ruppert averaging

Polyak and Juditsky (15) and Ruppert (16) independently developed an averaging method that relaxes the step-size constraint. The method consists in having a moving-average of the stochastic gradient as an optimum estimate as, for example,

$$\bar{\xi}_k \stackrel{\text{def}}{=} \frac{1}{k} \sum_{i=0}^k \xi_i \quad (3.10)$$

or

$$\bar{\xi}_k \stackrel{\text{def}}{=} \left( \sum_{\frac{k}{2} \leq i \leq k} \alpha_i \right)^{-1} \sum_{\frac{k}{2} \leq i \leq k} \alpha_i \xi_i. \quad (3.11)$$

Nemirovski (27) proves convergence of  $\bar{\xi}_k$  to  $\xi^*$  as  $k$  goes to infinity, for the averaging in Eq. 3.11. Moreover, he proves that it converges with any step-size sequence

$$\alpha_k = \frac{\alpha_0}{k^c} \quad (3.12)$$

for  $c$  in  $(1/2, 1)$ . In the present thesis, we use SGD with Polyak–Ruppert averaging as defined in Eq. 3.11 using step-size  $\alpha_k = \alpha_0/\sqrt{k}$ .

In Algorithm 6, we present the pseudocode for SGD with the Polyak–Ruppert averaging and the stop-criterion based on the Polyak–Ruppert average. Using Algorithm 6 has the advantage that the step-size does not decrease as fast as in vanilla SGD in Algorithm 5,  $\alpha_k = \alpha_0/k$ , thus, SGD with averaging can better explore the search space while convergence is still guaranteed.

---

**Algorithm 6** Pseudocode for SGD with Polyak–Ruppert averaging.

---

```

1: procedure SGD( $\xi_0, \alpha_0, \text{TOL}$ )
2:    $\bar{\xi}_0 \leftarrow \xi_0$ 
3:   for  $k = 1, 2, \dots$  do
4:     Sample random variables, e.g.,  $\theta_k, \mathbf{Y}_k, \dots$ 
5:      $\alpha_k \leftarrow \frac{\alpha_0}{\sqrt{k}}$ 
6:      $\xi_k \leftarrow \xi_{k-1} + \alpha_k \mathcal{G}(\xi_{k-1}, \theta_k, \mathbf{Y}_k)$ 
7:      $\bar{\xi}_k \leftarrow \left( \sum_{\frac{k}{2} \leq i \leq k} \alpha_i \right)^{-1} \sum_{\frac{k}{2} \leq i \leq k} \alpha_i \xi_i$   $\triangleright$  Polyak–Ruppert
       averaging
8:     if  $|\bar{\xi}_k - \bar{\xi}_{k-1}|_2 < \text{TOL}$  then  $\triangleright$  Stop Criterion
9:       Break
10:    end if
11:  end for
12:   $\hat{\xi} \leftarrow \bar{\xi}_k$ 
13: end procedure

```

---

In Figure 13, we present SGD with Polyak–Ruppert (with step-size  $\alpha_k \leftarrow \frac{\alpha_0}{\sqrt{k}}$ ) in red, and its average in black. The SGD walks erratically, whereas its average converges smoothly to the optimum in  $(0, 0)$ . In Figure 14, the convergence of the distance to the optimum per iteration is shown for vanilla SGD, SGD with Polyak–Ruppert, and its average. The SGD with Polyak–Ruppert behaves like a stochastic process, achieving the error of  $10^{-2}$  in 25 iterations and not improving much after that. The SGD averaging, however, continues to converge as the distance to the optimum decrease monotonically. Even though

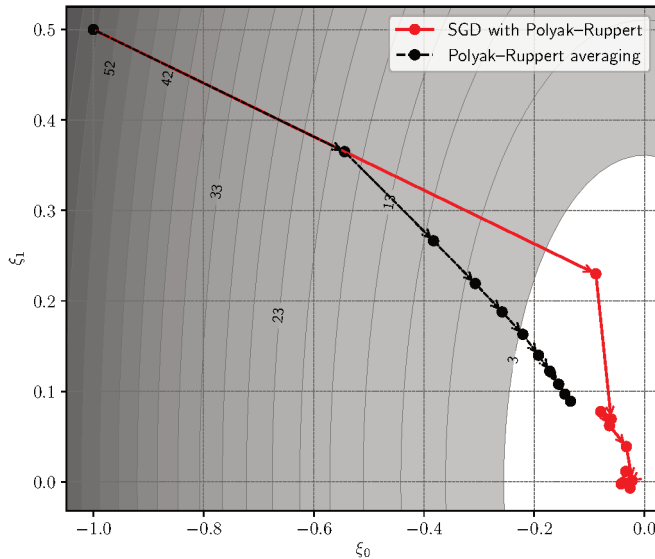


Figure 13 Example of SGD with Polyak Ruppert averaging.

both Polyak Ruppert averaging and vanilla SGD converge sublinearly, Polyak Ruppert averaging converges faster.

The SGD algorithm with Polyak Ruppert averaging is globally-convergent, converging almost-surely to the optimum of convex functions from any starting point, given that the step-size is properly tuned (27). However, since the step-size does not decrease as fast as in vanilla SGD, the resulting algorithm is less sensible to the step-size tuning, numerically converging for any  $\alpha_0$  of around the same order of magnitude of  $1/L$  (27).

In this thesis, we use Polyak Ruppert averaged SGD instead of vanilla SGD, given the robustness of the former and its global convergence properties. We also employ the change in the Polyak Ruppert averaging at iteration  $k$  as a stopping criterion for our algorithm, i.e., if  $|\bar{\xi}_k - \bar{\xi}_{k-1}|_2 < \text{TOL}$ , for a tolerance TOL defined by the user, then stop the optimization procedure.

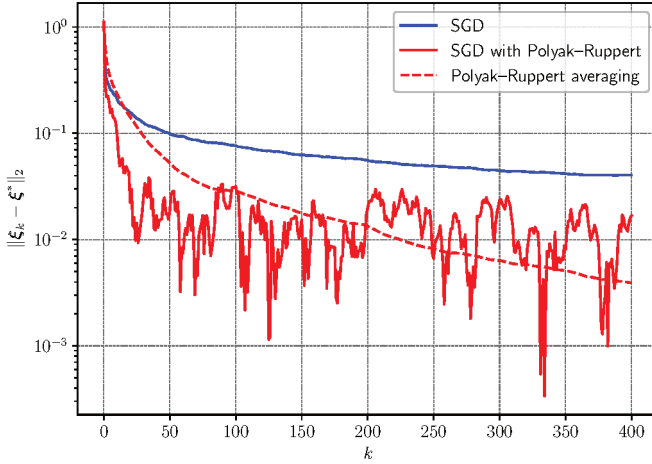


Figure 14 Convergence of vanilla SGD, SGD with Polyak Ruppert and its average.

### 3.3 Nesterov's accelerated gradient descent

The steepest descent method converges linearly to local minimum functions that are convex and smooth, however, its convergence rate is suboptimal (29). Nesterov (17) proposes an optimal first-order method that uses the information from previous iterations to achieve accelerated convergence. The Nesterov's accelerated gradient descent (AGD) is a gradient-based iterative method to find the local minimum of convex deterministic functions (17). The AGD is a momentum method in the sense that it adds a fraction of the last step taken to the current step, in an analogy to the linear momentum in physics. For a deterministic function  $f : \xi \rightarrow \mathbb{R}$ , we write the AGD to find a minimizer of  $f$  as

$$\begin{cases} \mathbf{z}_k = \xi_{k-1} - \alpha \nabla_{\xi} f(\xi_{k-1}) \\ \xi_k = \mathbf{z}_k + \gamma_k (\mathbf{z}_k - \mathbf{z}_{k-1}) \end{cases} \quad (3.13)$$

where the sequence  $(\gamma_k)_{k \geq 0}$  is given by

$$\gamma_k = \frac{\lambda_{k-1}(1 - \lambda_{k-1})}{\lambda_{k-1}^2 + \lambda_k}, \quad (3.14)$$

the sequence  $(\lambda_k)_{k \geq 0}$  solves

$$\lambda_k^2 = (1 - \lambda_k)\lambda_{k-1}^2 + q\lambda_k, \quad \lambda_0 = 1, \quad (3.15)$$

and  $q$  is a positive real number smaller than one ( $q \in (0, 1)$ ). In Figure 15, we present the optimization path for AGD for a quadratic function with optimum in  $(0, 0)$ . The red arrows illustrate the steepest descent steps from  $\xi_k$  to  $z_{k+1}$ . The blue arrows represent the accelerated parts of the steps, from  $z_{k+1}$  to  $\xi_{k+1}$ , calculated as in Eq. 3.13. It can be observed that  $\xi_2$  is closer to the optimum than  $z_2$ , i.e., acceleration furnishes a better approximation of the optimum with the same gradient information. Figure 16 shows how the Nesterov's accelerated step,

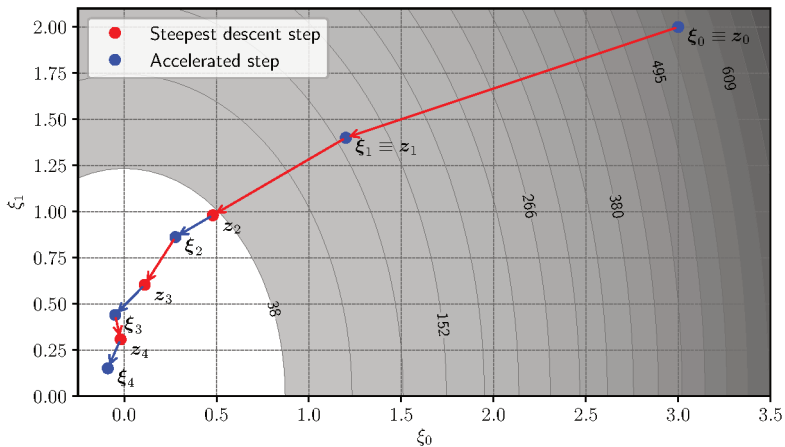


Figure 15 Example of Nesterov's acceleration.

presented in Eq. 3.13, is performed. The accelerated part of the optimization step is built as the scaled difference between  $z_{k-1}$  and  $z_k$ , where the scaling constant is  $\gamma_k$ .

The constant  $q$  defines how much acceleration is applied in the optimization; when  $q = 0$  the acceleration is maximum whereas  $q = 1$  furnishes the steepest descent method. Nesterov (29) proves that the optimal value for  $q$  is  $q = \mu/L$ , where  $L$  is the first-order  $L$ -Lipschitz constant of the objective function, and  $\mu$  is a strong-convexity constant,

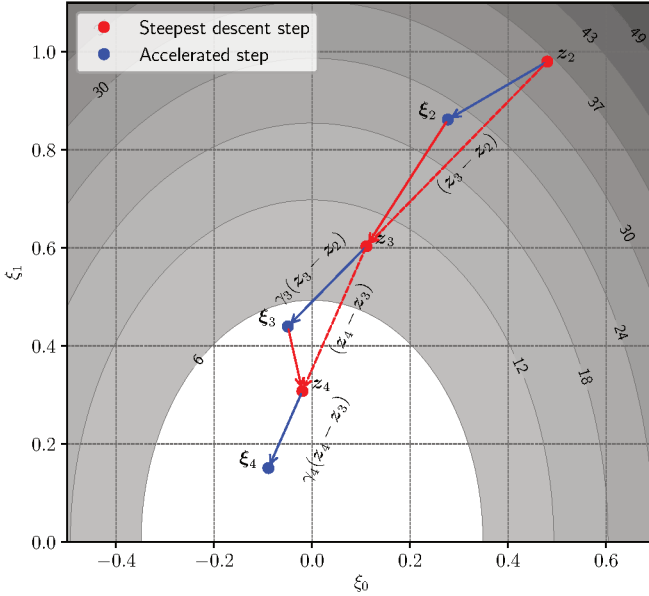


Figure 16 Representation of how accelerated steps are taken.

i.e.,  $\mu \preceq \nabla \nabla f(\xi) \preceq L \forall \xi \in \Xi$ , being  $f$  the objective function. As proven in (29), the use of the optimal  $q$  results in the optimal linear convergence for first-order methods. Figure 17 shows the convergence of Nesterov's acceleration for different values of  $q$ . It can be observed that using the optimal  $q$  results in the fastest convergence to the optimum. The convergences of the methods with  $q < q^*$  have bumps indicating that the algorithms are accelerating too much, resulting in the provable sublinear convergence of  $\mathcal{O}(1/k^2)$ . Cases where  $q > q^*$  have sub-optimal linear convergence (30), as can be observed in Figure 17.

In Algorithm 7 we present the Nesterov's accelerated gradient descent for the maximization of some deterministic function  $I$ .

### 3.3.1 Restart method

The use of  $q^*$  furnishes the optimal linear convergence for first-order methods (29), however, the constants  $\mu$  and  $L$  are difficult to

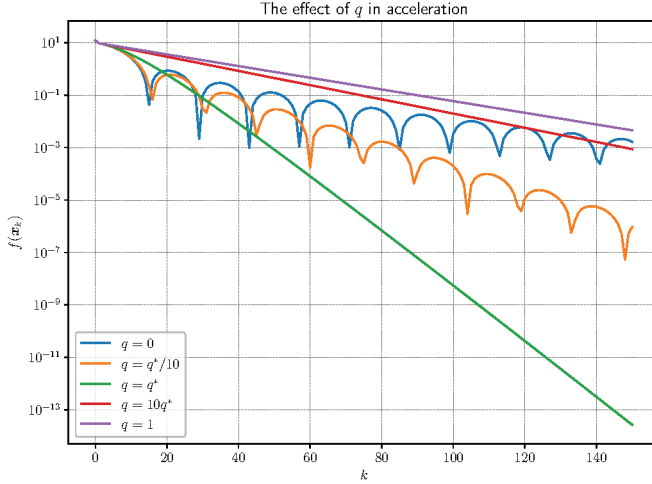


Figure 17 Convergence of the Nesterov acceleration for different values of  $q$ .

---

**Algorithm 7** Pseudocode for AGD.

---

```

1: procedure AGD( $\xi_0, \alpha, q$ )
2:    $z_i \leftarrow \xi_0, \lambda_0 \leftarrow 1$ 
3:   for  $k = 1, 2, \dots$  do
4:     Solve  $\lambda_k$  for  $\lambda_k^2 = (1 - \lambda_k)\lambda_{k-1}^2 + q\lambda_k$ 
5:      $\gamma_k \leftarrow \frac{\lambda_{k-1}(1 - \lambda_{k-1})}{\lambda_{k-1}^2 + \lambda_k}$ 
6:      $z_k \leftarrow \xi_{k-1} + \alpha \nabla_{\xi} I(\xi_{k-1})$ 
7:      $\xi_k \leftarrow \xi_{k-1} + \gamma_k(z_k - z_{k-1})$   $\triangleright$  Nesterov's accelerated step
8:   end for
9:    $\hat{\xi} \leftarrow \xi_k$ 
10: end procedure

```

---

estimate. O'Donoghue and Candès (30) propose an alternative method for achieving the same linear rate as with  $q^*$ , however without evaluating  $\mu$  and  $L$ . Their method consists of restarting the acceleration whenever the optimizer moves in an unwanted direction, e.g., for the maximization of  $I$ , when

$$\langle \nabla_{\xi} \mathbb{E}_{\theta, \mathbf{Y}} [f(\xi_{k-1}, \theta, \mathbf{Y})], \xi_k - \xi_{k-1} \rangle < 0. \quad (3.16)$$

This simple restart technique improves the convergence of Nesterov's acceleration without needing to tune  $q$ , i.e.,  $q$  can be set to 0.

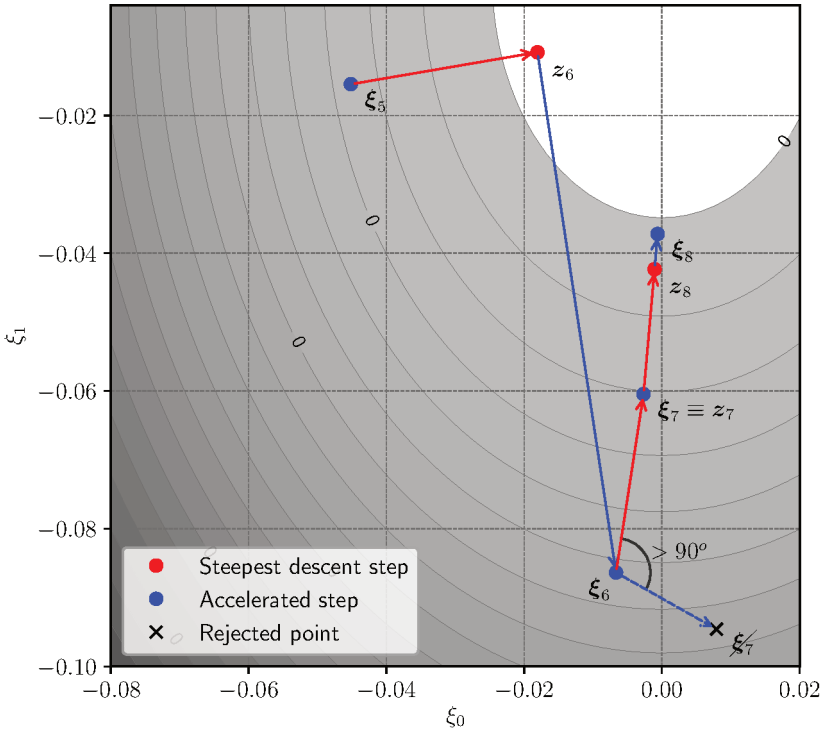


Figure 18 Illustration of restart technique for Nesterov's acceleration.

Figure 18 illustrates the restart technique for Nesterov's acceleration. The update candidate  $\xi_7$  is moving upwards the objective function, with respect to  $\xi_6$ , hence, it is rejected,  $\xi_7$  is set as  $z_7$ , and  $\lambda_k$  is reset to 1. Note that the angle between the steepest descent direction, illustrated with a red arrow, and the accelerated step, illustrated as a dashed blue arrow, have negative inner product, thus, as the condition in Eq. 3.16 is satisfied, the acceleration is restarted.

Figure 19 shows the convergence of Nesterov's accelerated gradient for two different values of  $q$ ,  $q = 0$  and  $q = 1$ , and for the accelerated gradient with restart. It can be observed that the restart technique



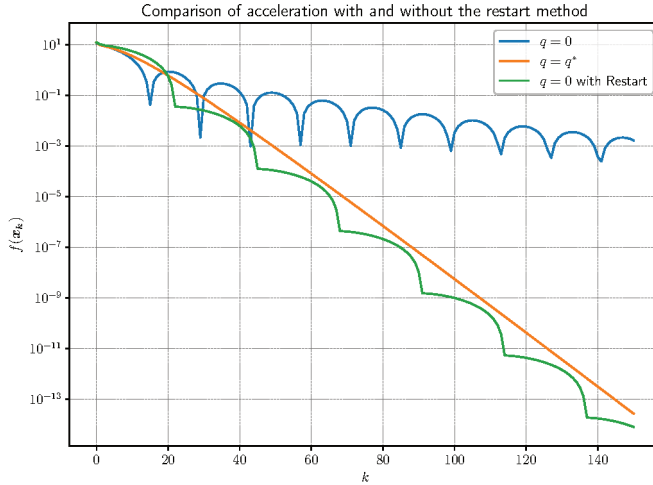


Figure 19 Convergence of the Nesterov acceleration with the restart technique.

improved the convergence of AGD with  $q = 0$  to the linear convergence of AGD with  $q = q^*$ .

O'Donoghue and Candès (30) also propose a third, equally efficient method based on verifying whether or not the objective function is decreasing, however, this method requires the evaluation of the objective function for each step. Su, Boyd and Candès (31) propose another criterion for the restart based on the increase of speed, i.e., restart if  $\|\xi_k - \xi_{k-1}\| < \|\xi_{k-1} - \xi_{k-2}\|$ , however, the gradient-based restart performs significantly better in their numerical evaluations.

**Algorithm 8** Pseudocode for AGD-restart.

---

```

1: procedure AGD( $\xi_0, \alpha, q$ )
2:    $z_0 \leftarrow \xi_0, \lambda_0 \leftarrow 1$ 
3:   for  $k = 1, 2, \dots$  do
4:     Solve  $\lambda_k$  for  $\lambda_k^2 = (1 - \lambda_k)\lambda_{k-1}^2 + q\lambda_k$ 
5:      $\gamma_k \leftarrow \frac{\lambda_{k-1}(1-\lambda_{k-1})}{\lambda_{k-1}^2 + \lambda_k}$ 
6:      $z_k \leftarrow \xi_{k-1} + \alpha \nabla_{\xi} I(\xi_{k-1})$ 
7:      $\xi_k \leftarrow \xi_{k-1} + \gamma_k(z_k - z_{k-1})$   $\triangleright$  Nesterov's accelerated step
8:     if  $\langle \nabla_{\xi} I(\xi_{k-1}), \xi_k - \xi_{k-1} \rangle < 0$  then  $\triangleright$  Restart technique
9:        $\xi_k \leftarrow z_k$ 
10:       $\lambda_{k+1} = 1$ 
11:     end if
12:   end for
13:    $\hat{\xi} \leftarrow \xi_k$ 
14: end procedure

```

---

### 3.4 Accelerated stochastic gradient descent

Using the Nesterov's accelerated step of AGD in Section 3.3 to the SGD in Section 3.2 furnishes the ASGD algorithm. We adapt the AGD for the stochastic optimization problem of OED by substituting the gradient in Eq. 3.13 for  $\mathcal{G}$  and the constant step-size by a decreasing one:

$$\begin{cases} z_k = \xi_{k-1} + \alpha_k \mathcal{G}(\xi_{k-1}, \theta_k, Y_k) \\ \xi_k = z_k + \gamma_k(z_k - z_{k-1}). \end{cases} \quad (3.17)$$

We couple ASGD with Polyak–Ruppert averaging to allow the use of the step-size sequence  $\alpha_k = \alpha_0/\sqrt{k}$ . Consequently, since the step-size does not reduce as fast as with  $\alpha_k = \alpha_0/k$ , ASGD with Polyak–Ruppert can better explore the search space, improving its global convergence properties. Moreover, in the asymptotic region, where the convergence is dominated by term  $\mathcal{O}(k^{-1/2})$ , the Nesterov acceleration does not affect the convergence. Thus, in the preasymptotic regime, if the ratio

between the noise and the true gradient is small, Nesterov’s acceleration achieves the optimal linear convergence for first order methods. As the optimization approaches the optimum and the true gradient norm converges to a null vector, i.e., as we get into the asymptotic phase, the ratio between the true gradient and the noise gets small and acceleration is degenerated. Results in the numerical section confirm the efficiency of ASGD for both a pure stochastic optimization problem and OED. In example 2, Nesterov’s acceleration reduced the total cost of optimization by a factor of ten in an average of a hundred independent runs.

A variant of ASGD using Polyak–Ruppert averaging is proposed by Lan (32) for the minimization of the sum of non-smooth functions, however, with an increasing step-size, instead of a decreasing one. Even though Lan presents a theoretical proof of ASGD convergence, he does not solve any numerical example using his method. Cotter et al. (33) use a variant of Lan’s ASGD for binary classification problems. According to Cotter et al. (33), the ASGD algorithm does not have the same accelerated convergence of AGD, being bounded by the same sublinear convergence rate as of SGD,  $\mathcal{O}(1/\sqrt{k})$ . Many papers have been recently published using Nesterov’s technique to accelerate stochastic optimization problems using variance reduction techniques to avoid losing the accelerated convergence (33, 34, 35). Recently, Allen-Zhu developed an accelerated algorithm called Katyusha (26) for which he proves accelerated convergence even for non-strongly convex stochastic problems.

In Algorithm 9, we present the pseudocode for ASGD, where the accelerated step is painted in blue and Polyak–Ruppert averaging and its respective step-size are painted in green.

The ASGD algorithm is the main optimization method of the present thesis.

**Algorithm 9** Pseudocode for ASGD.

---

```

1: procedure ASGD( $\xi_0, \alpha_0, q$ )
2:    $\bar{\xi}_0 \leftarrow \xi_0, z_0 \leftarrow \xi_0, \lambda_0 \leftarrow 1$ 
3:   for  $k = 1, 2, \dots$  do
4:     Sample random variables, e.g.,  $\theta_k, \mathbf{Y}_k, \dots$ 
5:     Solve  $\lambda_k$  for  $\lambda_k^2 = (1 - \lambda_k)\lambda_{k-1}^2 + q\lambda_k$ 
6:      $\gamma_k \leftarrow \frac{\lambda_{k-1}(1-\lambda_{k-1})}{\lambda_{k-1}^2 + \lambda_k}$ 
7:      $\alpha_k = \frac{\alpha_0}{\sqrt{k}}$  ▷ Decreasing step scheme
8:      $z_k \leftarrow \xi_{k-1} + \alpha_k \nabla_{\xi} \mathcal{G}(\xi_{k-1}, \theta_k, \mathbf{Y}_k)$ 
9:      $\xi_k \leftarrow \xi_{k-1} + \gamma_k (z_k - z_{k-1})$  ▷ Nesterov's accelerated step
10:     $\bar{\xi}_k = \left( \sum_{\frac{k}{2} \leq i \leq k} \alpha_i \right)^{-1} \sum_{\frac{k}{2} \leq i \leq k} \alpha_i \xi_i$  ▷ Polyak–Ruppert
    averaging
11:   end for
12:    $\hat{\xi} \leftarrow \xi_k$ 
13: end procedure

```

---

## 3.4.1 Restart method for the stochastic case

The restart method proposed by O’Donoghue and Candès (30) for the deterministic case uses information from the gradient of the objective function to decide whether or not to restart the Nesterov’s acceleration. Since we cannot observe the true gradient, we use the stochastic approximation of the gradient as the criterion to perform the restart, i.e.,

$$\langle \mathcal{G}(\xi_{k-1}, \theta_k, \mathbf{Y}_k), \xi_k - \xi_{k-1} \rangle < 0. \quad (3.18)$$

Nitanda (35) uses a restart scheme like this to improve the acceleration of the minimization of a finite number of sums. However, as previously mentioned, Nitanda uses control variates and minibatches to reduce the variance of the estimator, resulting in a near-deterministic estimation of the gradient.

In Algorithm 10, we present the general pseudocode for ASGD with restart technique (ASGD-restart).

**Algorithm 10** Pseudocode for ASGD-restart.

---

```

1: procedure ASGD-RESTART( $\xi_0, \alpha_0, q$ )
2:    $\bar{\xi}_0 \leftarrow \xi_0, z_0 \leftarrow \xi_0, \lambda_0 \leftarrow 1$ 
3:   for  $k = 1, 2, \dots$  do
4:     Sample random variables, e.g.,  $\theta_k, \mathbf{Y}_k, \dots$ 
5:     Solve  $\lambda_k$  for  $\lambda_k^2 = (1 - \lambda_k)\lambda_{k-1}^2 + q\lambda_k$ 
6:      $\gamma_k \leftarrow \frac{\lambda_{k-1}(1-\lambda_{k-1})}{\lambda_{k-1}^2 + \lambda_k}$ 
7:      $\alpha_k = \frac{\alpha_0}{\sqrt{k}}$  ▷ Decreasing step scheme
8:      $z_k = \xi_{k-1} + \alpha_k \mathcal{G}(\xi_{k-1}, \theta_k, \mathbf{Y}_k)$ 
9:      $\xi_k = z_k + \gamma_k(z_k - z_{k-1})$  ▷ Nesterov's accelerated step
10:    if  $\langle \mathcal{G}(\xi_{k-1}, \theta_k, \mathbf{Y}_k), \xi_k - \xi_{k-1} \rangle < 0$  then ▷ Restart
11:       $\lambda_k = 1$  technique
12:    end if
13:     $\bar{\xi}_k = \left( \sum_{\frac{k}{2} \leq i \leq k} \alpha_i \right)^{-1} \sum_{\frac{k}{2} \leq i \leq k} \alpha_i \xi_i$  ▷ Polyak–Ruppert
14:    end for averaging
15:     $\hat{\xi} \leftarrow \xi_{k-1}$ 
16: end procedure

```

---

## 3.5 Gradients of Monte Carlo estimators for OED

In this section, we deduce the gradients of the estimators presented in Section 2.2, for both full-gradient and stochastic optimization. From Eq. 3.3, and applying Eq. 3.8 we get

$$\nabla_{\xi} I(\xi) = \int_{\Theta} \int_{\mathcal{Y}} \nabla_{\xi} \log \left( \frac{p(\mathbf{Y}|\theta, \xi)}{p(\mathbf{Y}|\xi)} \right) p(\mathbf{Y}|\theta, \xi) d\mathbf{Y} \pi(\theta) d\theta, \quad (3.19)$$

the full-gradient of  $I$ . To use SGD, we need the gradient of  $f$  presented in Eq. 3.5, used to approximate the gradient in Eq. 3.19, i.e.,

$$\nabla_{\xi} f(\xi, \theta, \mathbf{Y}) = \nabla_{\xi} \log \left( \frac{p(\mathbf{Y}|\theta, \xi)}{p(\mathbf{Y}|\xi)} \right). \quad (3.20)$$

Developing the gradients in Eq. 3.20:

$$\nabla_{\xi} f(\xi, \theta, \mathbf{Y}) = \frac{\nabla_{\xi} p(\mathbf{Y}|\theta, \xi)}{p(\mathbf{Y}|\theta, \xi)} - \frac{\nabla_{\xi} p(\mathbf{Y}|\xi)}{p(\mathbf{Y}|\xi)}. \quad (3.21)$$

From Eq. 2.3,

$$\begin{aligned}
\nabla_{\boldsymbol{\xi}} p(\mathbf{Y}(\boldsymbol{\xi}, \boldsymbol{\theta}, \boldsymbol{\epsilon}) | \boldsymbol{\theta}, \boldsymbol{\xi}) &= p(\mathbf{Y} | \boldsymbol{\theta}, \boldsymbol{\xi}) \nabla_{\boldsymbol{\xi}} \left( -\frac{1}{2} \sum_{i=1}^{N_{\boldsymbol{\epsilon}}} \|\mathbf{y}_i - \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta})\|_{\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1}}^2 \right) \\
&= p(\mathbf{Y} | \boldsymbol{\theta}, \boldsymbol{\xi}) \nabla_{\boldsymbol{\xi}} \left( -\frac{1}{2} \sum_{i=1}^{N_{\boldsymbol{\epsilon}}} \|\mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}) + \boldsymbol{\epsilon}_i - \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta})\|_{\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1}}^2 \right) \\
&= p(\mathbf{Y} | \boldsymbol{\theta}, \boldsymbol{\xi}) \nabla_{\boldsymbol{\xi}} \left( -\frac{1}{2} \sum_{i=1}^{N_{\boldsymbol{\epsilon}}} \|\boldsymbol{\epsilon}_i\|_{\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1}}^2 \right) \\
&= 0,
\end{aligned} \tag{3.22}$$

thus, the first term in Eq. 3.21 disappears. To estimate the evidence and its gradient, we marginalize them with respect to  $\boldsymbol{\theta}^*$ , as in Eq. 2.4. Thus,

$$\nabla_{\boldsymbol{\xi}} p(\mathbf{Y} | \boldsymbol{\xi}) = \int_{\Theta} \nabla_{\boldsymbol{\xi}} p(\mathbf{Y} | \boldsymbol{\theta}^*, \boldsymbol{\xi}) \pi(\boldsymbol{\theta}^*) d\boldsymbol{\theta}^*. \tag{3.23}$$

Substituting Eq. 3.23 in Eq. 3.21 furnishes

$$\begin{aligned}
\nabla_{\boldsymbol{\xi}} f(\boldsymbol{\xi}, \boldsymbol{\theta}, \mathbf{Y}) &= -\frac{\nabla_{\boldsymbol{\xi}} p(\mathbf{Y} | \boldsymbol{\xi})}{p(\mathbf{Y} | \boldsymbol{\xi})}, \\
&= -\frac{\int_{\Theta} \nabla_{\boldsymbol{\xi}} p(\mathbf{Y} | \boldsymbol{\theta}^*, \boldsymbol{\xi}) \pi(\boldsymbol{\theta}^*) d\boldsymbol{\theta}^*}{\int_{\Theta} p(\mathbf{Y} | \boldsymbol{\theta}^*, \boldsymbol{\xi}) \pi(\boldsymbol{\theta}^*) d\boldsymbol{\theta}^*}
\end{aligned} \tag{3.24}$$

The gradient of the likelihood of observing  $\mathbf{Y}$  given  $\boldsymbol{\theta}^*$  different than  $\boldsymbol{\theta}$  is

$$\begin{aligned}
\nabla_{\boldsymbol{\xi}} p(\mathbf{Y}(\boldsymbol{\xi}, \boldsymbol{\theta}, \boldsymbol{\epsilon}) | \boldsymbol{\theta}^*, \boldsymbol{\xi}) &= p(\mathbf{Y} | \boldsymbol{\theta}^*, \boldsymbol{\xi}) \nabla_{\boldsymbol{\xi}} \left( -\frac{1}{2} \sum_{i=1}^{N_{\boldsymbol{\epsilon}}} \|\mathbf{y}_i - \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}^*)\|_{\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1}}^2 \right) \\
&= -p(\mathbf{Y} | \boldsymbol{\theta}^*, \boldsymbol{\xi}) \left[ \sum_{i=1}^{N_{\boldsymbol{\epsilon}}} (\nabla_{\boldsymbol{\xi}} \mathbf{y} - \nabla_{\boldsymbol{\xi}} \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}^*))^T \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1} (\mathbf{y}_i - \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}^*)) \right] \\
&= -p(\mathbf{Y} | \boldsymbol{\theta}^*, \boldsymbol{\xi}) \left[ (\nabla_{\boldsymbol{\xi}} \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}) - \nabla_{\boldsymbol{\xi}} \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}^*))^T \sum_{i=1}^{N_{\boldsymbol{\epsilon}}} \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-1} (\mathbf{y}_i - \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}^*)) \right]
\end{aligned} \tag{3.25}$$

In the next section we present estimators of the gradient of  $f$  using MCI and Laplace approximation.

### 3.5.1 Gradient of the double-loop Monte Carlo estimator

Similarly to the derivation of the DLMC estimator for the EIG in Section 2.2.3, we approximate the nested integrals by Monte Carlo integration and the evidence by another Monte Carlo integration, as in Eq. 2.4,

$$\nabla_{\xi} \mathcal{I}_{DLMC}(\xi) = \frac{1}{N} \sum_{n=1}^N \nabla_{\xi} \log \left( \frac{p(\mathbf{Y}_n | \boldsymbol{\theta}_n, \xi)}{\frac{1}{M} \sum_{m=1}^M p(\mathbf{Y}_n | \boldsymbol{\theta}_m^*, \xi)} \right). \quad (3.26)$$

Given that Eq. 3.8 holds, the gradient in Eq. 3.26 is exactly the gradient of the DLMC estimator in Eq. 2.17, i.e., the gradient in Eq. 3.26 is an approximation of the full-gradient of  $I$ . To perform SGD, we need an estimator for  $\nabla_{\xi} f$ . Marginalizing the evidence in Eq. 3.26 with respect to  $\boldsymbol{\theta}^*$ , and applying MCI, furnishes the approximation of the gradient of the DLMC estimator,

$$\mathcal{G}_{DLMC}(\xi, \boldsymbol{\theta}, \mathbf{Y}) \stackrel{\text{def}}{=} \nabla_{\xi} \log \left( \frac{p(\mathbf{Y} | \boldsymbol{\theta}, \xi)}{\frac{1}{M} \sum_{m=1}^M p(\mathbf{Y} | \boldsymbol{\theta}_m^*, \xi)} \right). \quad (3.27)$$

The Eq. 3.27 does not have two Monte Carlo loops, however, to keep the consistency between the naming of  $\mathcal{I}$  and  $\mathcal{G}$  estimators, we keep the same naming standard. Developing the gradient in Eq. 3.27 and noting that  $\nabla_{\xi} p(\mathbf{Y} | \boldsymbol{\theta}, \xi) = \mathbf{0}$ , we have

$$\mathcal{G}_{DLMC}(\xi, \boldsymbol{\theta}, \mathbf{Y}) = - \frac{\sum_{m=1}^M \nabla_{\xi} p(\mathbf{Y} | \boldsymbol{\theta}_m^*, \xi)}{\sum_{m=1}^M p(\mathbf{Y} | \boldsymbol{\theta}_m^*, \xi)}. \quad (3.28)$$

Using MCI on the two integrals in Eq. 3.24 results in a similar estimator, however, we opt to use the same set  $\{\boldsymbol{\theta}_m^*\}_{m>0}$  for both integrals, reducing computational complexity. Haun and Marzouk (14) use the gradient of the DLMC estimator in Eq. 3.28.

In Algorithm 11, we present the the pseudocode for the evaluation of estimator  $\mathcal{G}_{DLMC}$ .

Note that the cost of evaluating  $\mathcal{G}_{DLMC}$ , is  $N$  times smaller than the cost of evaluating  $\nabla_{\xi} \mathcal{I}_{DLMC}$ . For example, if forward finite differences are used to calculate the gradient of  $\mathbf{g}$ , the cost of evaluating

---

**Algorithm 11** Pseudocode for the gradient of the DLMC estimator.

---

```

1: function GRADIENT( $\xi, \theta, \mathbf{Y}$ )
2:   Evaluate  $\mathbf{g}(\xi, \theta)$ 
3:   Evaluate  $\nabla_{\xi} \mathbf{g}(\xi, \theta)$ 
4:   for  $i = 1, 2, \dots, N_e$  do
5:     Sample  $\epsilon_i$  from  $\mathcal{N}(0, \Sigma_{\epsilon})$ 
6:      $\mathbf{y}_i \leftarrow \mathbf{g}(\xi, \theta) + \epsilon_i$ 
7:   end for
8:    $\mathbf{Y} \leftarrow \{\mathbf{y}_i\}_{i=1}^{N_e}$ 
9:   for  $m = 1, 2, \dots, M$  do
10:    Sample  $\theta_m^*$  from  $\pi(\theta)$ 
11:    Evaluate  $\mathbf{g}(\xi, \theta_m^*)$ 
12:     $p(\mathbf{Y}_n | \theta_m^*, \xi) \leftarrow \det(2\pi \Sigma_{\epsilon})^{-\frac{N_e}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N_e} \|\mathbf{y}_i^{(n)}(\xi) - \mathbf{g}(\xi, \theta_m^*)\|_{\Sigma_{\epsilon}^{-1}}^2\right)$ 
13:    Evaluate  $\nabla_{\xi} \mathbf{g}(\xi, \theta_m^*)$ 
14:     $\nabla_{\xi} p(\mathbf{Y} | \theta_m^*, \xi) \leftarrow -p(\mathbf{Y} | \theta_m^*, \xi) \left[ (\nabla_{\xi} \mathbf{g}(\xi, \theta) - \nabla_{\xi} \mathbf{g}(\xi, \theta_m^*))^T \right.$ 
       $\left. \sum_{i=1}^{N_e} \Sigma_{\epsilon}^{-1} (\mathbf{y}_i - \mathbf{g}(\xi, \theta_m^*)) \right]$ 
15:   end for
16:    $\mathcal{G}_{DLMC} \leftarrow -\frac{\sum_{m=1}^M \nabla_{\xi} p(\mathbf{Y} | \theta_m^*, \xi)}{\sum_{m=1}^M p(\mathbf{Y} | \theta_m^*, \xi)}$ 
17:   return  $\mathcal{G}_{DLMC}$ 
18: end function

```

---

$\nabla_{\xi} \mathcal{I}_{DLMC}$  is  $N(M+1)(\dim(\xi) + 1)h^{-e}$ , whereas, for  $\mathcal{G}_{DLMC}$ , the cost is  $(M+1)(\dim(\xi) + 1)h^{-e}$ .

### 3.5.2 Gradient of the Monte Carlo estimator with Laplace approximation

From Eq. 2.28, we have the Laplace approximation of the EIG. We derive the full-gradient of the Laplace approximation of  $I$  as

$$\nabla_{\xi} I(\xi) \approx \int_{\Theta} \left[ -\frac{1}{2} \nabla_{\xi} \log(\det(2\pi \Sigma(\xi, \theta_t))) \right] \pi(\theta_t) d\theta_t. \quad (3.29)$$

Approximating the integral in Eq. 3.29 by a MCI results in the full-gradient of the MCLA estimator,

$$\nabla_{\xi} \mathcal{I}_{MCLA}(\xi) = \frac{1}{N} \sum_{n=1}^N \left[ -\frac{1}{2} \nabla_{\xi} \log(\det(2\pi \Sigma(\xi, \theta_n))) \right]. \quad (3.30)$$



The gradient of  $f$  using the Laplace approximation, needed for stochastic optimization, is

$$\begin{aligned}\nabla_{\boldsymbol{\xi}} \mathcal{G}_{MCLA}(\boldsymbol{\xi}, \boldsymbol{\theta}) &= -\frac{1}{2} \nabla_{\boldsymbol{\xi}} \log(\det(2\pi \boldsymbol{\Sigma}(\boldsymbol{\xi}, \boldsymbol{\theta}))) \\ &= -\frac{1}{2} \nabla_{\boldsymbol{\xi}} \log(\det(\boldsymbol{\Sigma}(\boldsymbol{\xi}, \boldsymbol{\theta}))) \\ &= -\frac{1}{2} \frac{\nabla_{\boldsymbol{\xi}} \det(\boldsymbol{\Sigma}(\boldsymbol{\xi}, \boldsymbol{\theta}))}{\det(\boldsymbol{\Sigma}(\boldsymbol{\xi}, \boldsymbol{\theta}))}.\end{aligned}\quad (3.31)$$

Using Jacobi's formula (and suppressing the dependencies of  $\boldsymbol{\Sigma}$ ) furnishes

$$\nabla_{\boldsymbol{\xi}} \det(\boldsymbol{\Sigma}) = \text{tr}(\det(\boldsymbol{\Sigma}) (\boldsymbol{\Sigma}^{-1})^T \nabla_{\boldsymbol{\xi}} \boldsymbol{\Sigma}), \quad (3.32)$$

thus,

$$\begin{aligned}\nabla_{\boldsymbol{\xi}} \mathcal{G}_{MCLA}(\boldsymbol{\xi}, \boldsymbol{\theta}) &= -\frac{1}{2} \frac{\text{tr}(\det(\boldsymbol{\Sigma}) (\boldsymbol{\Sigma}^{-1})^T \nabla_{\boldsymbol{\xi}} \boldsymbol{\Sigma})}{\det(\boldsymbol{\Sigma})} \\ &= -\frac{1}{2} \text{tr}((\boldsymbol{\Sigma}^{-1})^T \nabla_{\boldsymbol{\xi}} \boldsymbol{\Sigma})\end{aligned}\quad (3.33)$$

Using Eq. 2.27, and remembering that  $\boldsymbol{\Sigma}_e$  is symmetric,

$$\nabla_{\boldsymbol{\xi}} \boldsymbol{\Sigma}^{-1} = 2N_e \nabla_{\boldsymbol{\xi}} \nabla_{\boldsymbol{\theta}} \mathbf{g}^T \boldsymbol{\Sigma}_e^{-1} \nabla_{\boldsymbol{\theta}} \mathbf{g}, \quad (3.34)$$

and, using the identity  $\nabla \boldsymbol{\Sigma} = -\boldsymbol{\Sigma}^T \nabla \boldsymbol{\Sigma}^{-1} \boldsymbol{\Sigma}$ ,

$$\nabla_{\boldsymbol{\xi}} \boldsymbol{\Sigma} = -2N_e \boldsymbol{\Sigma} \nabla_{\boldsymbol{\xi}} \nabla_{\boldsymbol{\theta}} \mathbf{g} \boldsymbol{\Sigma}_e^{-1} \nabla_{\boldsymbol{\theta}} \mathbf{g} \boldsymbol{\Sigma}. \quad (3.35)$$

Using Eq. 3.33,

$$\begin{aligned}\nabla_{\boldsymbol{\xi}} \mathcal{G}_{MCLA}(\boldsymbol{\xi}, \boldsymbol{\theta}) &= -\frac{1}{2} \text{tr}((\boldsymbol{\Sigma}^{-1})^T (-2N_e \boldsymbol{\Sigma} \nabla_{\boldsymbol{\xi}} \nabla_{\boldsymbol{\theta}} \mathbf{g} \boldsymbol{\Sigma}_e^{-1} \nabla_{\boldsymbol{\theta}} \mathbf{g} \boldsymbol{\Sigma})) \\ &= N_e \text{tr}(\nabla_{\boldsymbol{\xi}} \nabla_{\boldsymbol{\theta}} \mathbf{g} \boldsymbol{\Sigma}_e^{-1} \nabla_{\boldsymbol{\theta}} \mathbf{g} \boldsymbol{\Sigma}),\end{aligned}\quad (3.36)$$

or, in index notation,

$$\{\mathcal{G}_{MCLA}\}_s = (N_e) \Sigma_{lm} \left( \frac{\partial^2 g_i}{\partial \xi_s \partial \theta_l} (\Sigma_e^{-1})_{ij} \frac{\partial g_j}{\partial \theta_m} \right). \quad (3.37)$$

This estimator has the advantage of not needing the solution of any integral, i.e., it does not need MCI.

---

**Algorithm 12** Pseudocode for the gradient of the MCLA estimator.

---

- 1: **function** GRADIENT( $\xi, \theta, \mathbf{Y}$ )
  - 2:     Evaluate  $\nabla_{\theta} g(\xi, \theta)$  and  $\nabla_{\xi} \nabla_{\theta} g(\xi, \theta)$
  - 3:     Use  $\nabla_{\theta} g(\xi, \theta)$  to evaluate  $\Sigma(\xi, \theta)$  using Eq. 2.27
  - 4:      $\mathcal{G}_{MCLA}(\xi) \leftarrow N e^{\text{tr}(\nabla_{\xi} \nabla_{\theta} g(\xi, \theta) \Sigma_{\epsilon}^{-1} \nabla_{\theta} g(\xi, \theta) \Sigma(\xi, \theta))}$
  - 5:     **return**  $\mathcal{G}_{MCLA}(\xi)$
  - 6: **end function**
- 

Evaluating the gradient of the MCLA estimator requires both  $\nabla_{\theta} g$ , and  $\nabla_{\xi} \nabla_{\theta} g$ . If we use forward Euler to approximate the derivatives, the cost of evaluating  $\mathcal{G}_{MCLA}$  is  $(\dim(\xi) + 1)(\dim(\theta) + 1)h^{\eta}$ ,  $N$  times less than the cost of evaluating  $\nabla_{\xi} \mathcal{L}_{MCLA}$ , which is  $(\dim(\xi) + 1)(\dim(\theta) + 1)Nh^{\eta}$ .

### 3.5.3 Gradient of the double-loop Monte Carlo estimator with importance sampling

The full-gradient of the DLMC estimator in Eq. 3.26 can be improved by using an importance sampling to estimate the evidence,

$$\nabla_{\xi} \mathcal{L}_{DLMCIS}(\xi) = \frac{1}{N} \sum_{n=1}^N \nabla_{\xi} \log \left( \frac{p(\mathbf{y}_n | \theta_n, \xi)}{\frac{1}{M} \sum_{m=1}^M \mathcal{L}(\mathbf{y}_n; \xi; \theta_m^*)} \right), \quad (3.38)$$

where  $\mathcal{L}$  is presented in Eq. 2.33. Similarly, the approximation of Eq. 3.38, i.e., the gradient of  $f$  using importance sampling, is

$$\mathcal{G}_{DLMCIS}(\xi, \theta, \mathbf{Y}) = \nabla_{\xi} \log \left( \frac{p(\mathbf{y} | \theta, \xi)}{\frac{1}{M} \sum_{m=1}^M \mathcal{L}(\mathbf{y}; \xi; \theta_m^*)} \right). \quad (3.39)$$

The development of Eq. 3.39 is similar to the one of Eq. 3.27.

$$\mathcal{G}_{DLMCIS}(\xi, \theta, \mathbf{Y}) = \frac{\nabla_{\xi} p(\mathbf{Y} | \theta, \xi)}{p(\mathbf{Y} | \theta, \xi)} - \frac{\sum_{m=1}^M \nabla_{\xi} \mathcal{L}(\mathbf{y}; \xi; \theta_m^*)}{\sum_{m=1}^M \mathcal{L}(\mathbf{y}; \xi; \theta_m^*)}. \quad (3.40)$$

The gradient of the likelihood in the first term of Eq. 3.40 vanishes, as shown in Eq. 3.22. However, the evaluation of the gradient of  $\mathcal{L}$  is cumbersome. Remember that the importance sampling pdf  $\tilde{\pi}$  is Gaussian-distributed with mean  $\hat{\theta}(\xi, \theta)$  and covariance matrix  $\Sigma(\xi, \hat{\theta})$ ,

thus, evaluating its gradient requires the evaluation of the gradient of  $\hat{\boldsymbol{\theta}}$  with respect to  $\boldsymbol{\xi}$ . We assume  $\nabla_{\boldsymbol{\xi}}\tilde{\pi} \approx \mathbf{0}$ , thus, taking the gradient of Eq. 2.33 furnishes

$$\nabla_{\boldsymbol{\xi}}\mathcal{L}(\mathbf{Y}; \boldsymbol{\xi}; \boldsymbol{\theta}) \approx \frac{\nabla_{\boldsymbol{\xi}}p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})\pi(\boldsymbol{\theta})}{\tilde{\pi}(\boldsymbol{\theta})}. \quad (3.41)$$

Indeed, for any  $\tilde{\boldsymbol{\xi}} \stackrel{\text{def}}{=} \lim_{\Delta_{\boldsymbol{\xi}} \rightarrow 0}(\boldsymbol{\xi} + \Delta_{\boldsymbol{\xi}})$ , the MAP of  $\boldsymbol{\theta}$  at  $\tilde{\boldsymbol{\xi}}$  can be approximated as  $\hat{\boldsymbol{\theta}}(\tilde{\boldsymbol{\xi}}) \approx \hat{\boldsymbol{\theta}}(\boldsymbol{\xi})$ , hence, this simplification should not affect importance sampling effectiveness in drawing informative samples. Therefore, we opt to calculate  $\hat{\boldsymbol{\theta}}$  and  $\boldsymbol{\Sigma}(\hat{\boldsymbol{\theta}})$  only once per  $\mathcal{G}_{DLMCIS}$  evaluation. Moreover, we sample  $\boldsymbol{\theta}^*$  only once per  $\mathcal{G}_{DLMCIS}$  and use it to estimate the gradient of  $\mathcal{L}$ . The gradient of the likelihood in Eq. 3.41 is evaluated as in Eq. 3.25.

If we use forward finite differences to estimate the gradients with respect to  $\boldsymbol{\xi}$  and  $\boldsymbol{\theta}$ , the cost for evaluating  $\mathcal{G}_{DLMCIS}$  is  $[(M+1)(\dim(\boldsymbol{\xi})+1) + (C_{MAP}+1)(\dim(\boldsymbol{\theta})+1)]h^{-e}$ , where  $C_{MAP}$  is the number of iterations needed to find  $\hat{\boldsymbol{\theta}}$  in Algorithm 3. Moreover, the cost of evaluating  $\mathcal{G}_{DLMCIS}$  is already  $N$  times smaller the cost of evaluating  $\nabla_{\boldsymbol{\xi}}\mathcal{I}_{DLMCIS}$ , which is of  $N(\dim(\boldsymbol{\xi})+1)[(1+C_{MAP})(\dim(\boldsymbol{\theta})+1)+M+1]h^{-e}$ .

Algorithm 13 presents the pseudocode for the gradient of the DLMCIS estimator, where the line where importance sampling happens is shaded in gray.

Given that the cost of estimating  $\nabla_{\boldsymbol{\theta}}\mathbf{g}$  and  $\nabla_{\boldsymbol{\xi}}\mathbf{g}$  by finite differences are respectively  $C_{J,\boldsymbol{\theta}}$  and  $C_{J,\boldsymbol{\xi}}$ , and that the cost of evaluating  $\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\xi}}\mathbf{g}$  is  $C_{J,\boldsymbol{\theta}}C_{J,\boldsymbol{\xi}}$ , the number of evaluations of  $\mathbf{g}$  needed by each estimator  $\mathcal{I}$ , their full gradients, and gradient estimators  $\mathcal{G}$  are presented in Table 1.

### 3.5.4 Pseudocode of the complete stochastic optimization framework for OED

In Algorithm 14, we present the pseudocode for the ASGD method for Bayesian OED. The Polyak–Ruppert averaging at the last iteration is taken as  $\hat{\boldsymbol{\xi}} \approx \boldsymbol{\xi}^*$ .

---

**Algorithm 13** Pseudocode for the gradient of the DLMCIS estimator.

---

```

1: function GRADIENT( $\xi, \theta, \mathbf{Y}$ )
2:   Evaluate  $\nabla_{\xi} g(\xi, \theta)$ 
3:   for  $i = 1, 2, \dots, N_e$  do
4:     Sample  $\epsilon_i$  from  $\mathcal{N}(0, \Sigma_{\epsilon})$ 
5:      $\mathbf{y}_i \leftarrow g(\xi, \theta) + \epsilon_i$ 
6:   end for
7:    $\mathbf{Y} \leftarrow \{\mathbf{y}_i\}_{i=1}^{N_e}$ 
8:   Find  $\hat{\theta}(\xi, \mathbf{Y})$  using Algorithm 3
9:   Evaluate  $\nabla_{\theta} g(\xi, \hat{\theta})$ 
10:  Use  $\nabla_{\theta} g(\xi, \hat{\theta})$  to evaluate  $\Sigma(\xi, \hat{\theta})$  using Eq. 2.27
11:  for  $m = 1, 2, \dots, M$  do ▷ Inner loop
12:    Sample  $\theta_m^*$  from  $\tilde{\pi}(\theta) \sim \mathcal{N}(\hat{\theta}, \Sigma(\xi, \hat{\theta}))$  ▷ Importance
    sampling
13:     $p(\mathbf{Y}_n | \theta_m^*, \xi) \leftarrow \det(2\pi\Sigma_{\epsilon})^{-\frac{N_e}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N_e} \left\| \mathbf{y}_i^{(n)}(\xi) - g(\xi, \theta_m^*) \right\|_{\Sigma_{\epsilon}^{-1}}^2\right)$ 
14:     $\mathcal{L}(\mathbf{Y}_n; \xi; \theta_m^*) \leftarrow p(\mathbf{Y}_n | \theta_m^*, \xi) \pi(\theta_m^*) / \tilde{\pi}(\theta_m^*)$ 
15:    Evaluate  $\nabla_{\xi} g(\xi, \theta_m^*)$ 
16:     $\nabla_{\xi} p(\mathbf{Y} | \theta_m^*, \xi) \leftarrow -p(\mathbf{Y} | \theta_m^*, \xi) \left[ (\nabla_{\xi} g(\xi, \theta) - \nabla_{\xi} g(\xi, \theta_m^*))^T \right.$ 
     $\left. \sum_{i=1}^{N_e} \Sigma_{\epsilon}^{-1} (\mathbf{y}_i - g(\xi, \theta_m^*)) \right]$ 
17:     $\nabla_{\xi} \mathcal{L}(\mathbf{Y}_n; \xi; \theta_m^*) \leftarrow \nabla_{\xi} p(\mathbf{Y}_n | \theta_m^*, \xi) \pi(\theta_m^*) / \tilde{\pi}(\theta_m^*)$ 
18:  end for
19:   $\mathcal{G}_{DLMCIS}(\xi, \theta, \mathbf{Y}) \leftarrow -\frac{\sum_{m=1}^M \nabla_{\xi} \mathcal{L}(\mathbf{Y}; \xi; \theta_m^*)}{\sum_{m=1}^M \mathcal{L}(\mathbf{Y}; \xi; \theta_m^*)}$ 
20:  return  $\mathcal{G}_{DLMCIS}(\xi)$ 
21: end function

```

---

ASGD-restart is the main optimization algorithm of this thesis and its efficiency and robustness is assessed in the numerical example section.

## 3.6 Chapter summary

In this chapter, we introduced the stochastic optimization framework that we propose to solve OED problems. In Section 3.2, we presented the Robbins–Monro algorithm, also known as the SGD, and the rationale behind it, pointing out its main drawbacks and the literature

---

**Algorithm 14** Pseudocode for the ASGD-restart method for Bayesian OED.

---

```

1: procedure ASGD( $\xi_0, \alpha_0, q, \text{TOL}$ )
2:    $\mathbf{z}_0 = \xi_0, \lambda_0 = 1$ 
3:   for  $k = 1, 2, \dots$  do
4:     Solve  $\lambda_k$  for  $\lambda_k^2 = (1 - \lambda_k)\lambda_{k-1}^2 + q\lambda_k$ 
5:      $\gamma_k = \frac{\lambda_{k-1}(1-\lambda_{k-1})}{\lambda_{k-1}^2 + \lambda_k}$ 
6:     Sample  $\theta_k$  from  $\pi(\theta)$ 
7:     for  $i = 1, 2, \dots, N_e$  do
8:       Sample  $\epsilon_i$  from  $\mathcal{N}(0, \Sigma_\epsilon)$ 
9:        $\mathbf{y}_i = \mathbf{g}(\xi_{k-1}, \theta_k) + \epsilon_i$ 
10:    end for
11:     $\mathbf{Y}_k = \{\mathbf{y}_i\}_{i=1}^{N_e}$ 
12:     $\alpha_k = \frac{\alpha_0}{\sqrt{k}}$  ▷ Decreasing step scheme
13:     $\mathbf{z}_k = \xi_{k-1} + \alpha_k \mathcal{G}(\xi_{k-1}, \theta_k, \mathbf{Y}_k)$ 
14:     $\xi_k = \mathbf{z}_k + \gamma_k(\mathbf{z}_k - \mathbf{z}_{k-1})$  ▷ Nesterov's accelerated step
15:    if  $\langle \mathcal{G}(\xi_{k-1}, \theta_k, \mathbf{Y}_k), \xi_k - \xi_{k-1} \rangle < 0$  then ▷ Restart
16:      technique  $\lambda_k = 1$ 
17:    end if
18:     $\bar{\xi}_k = \left( \sum_{\frac{k}{2} \leq i \leq k} \alpha_i \right)^{-1} \sum_{\frac{k}{2} \leq i \leq k} \alpha_i \xi_i$  ▷ Polyak–Ruppert
19:    if  $|\bar{\xi}_k - \bar{\xi}_{k-1}|_2 < \text{TOL}$  then ▷ Stop Criterion
20:      Break
21:    end if
22:  end for
23:   $\hat{\xi} \stackrel{\text{def}}{=} \bar{\xi}_{k+1}$ 
24: end procedure

```

---

Table 1 – Number of evaluations of  $\mathbf{g}$  required for each estimator.

	DLMC	MCLA	DLMCIS
$\mathcal{I}$	$N(M + 1)$	$NC_{J,\theta}$	$N[(C_{MAP} + 1)C_{J,\theta} + M + 1]$
$\nabla_{\xi}\mathcal{I}$	$C_{J,\xi}N(M + 1)$	$C_{J,\xi}NC_{J,\theta}$	$C_{J,\xi}N[(C_{MAP} + 1)C_{J,\theta} + M + 1]$
$\mathcal{G}$	$C_{J,\xi}(M + 1)$	$C_{J,\xi}C_{J,\theta}$	$(C_{MAP} + 1)C_{J,\theta} + C_{J,\xi}(M + 1)$

limitation regarding its application to OED problems. In order to overcome these drawbacks, we presented in the following section different approaches, we presented the averaging technique, known as the Polyak–Ruppert averaging, as a way of improving SGD main weaknesses, its lack of robustness and its step-size tuning sensibility.

In Section 3.3, we introduced Nesterov’s acceleration to deterministic gradient descent (17) and a restart method proposed by O’Donoghue and Candès (30).

In Section 3.4, we apply Nesterov’s acceleration to SGD, furnishing the ASGD. In addition, we adapted the deterministic restart technique from O’Donoghue and Candès (30) to ASGD. In Section 3.5, we derived the gradients of the EIG estimators presented in Chapter 2.

The gradient of the MCLA estimator with respect to design parameters is novel, being first published in a paper from the authors (36). Moreover, we deduced the gradient of the DLMCIS with respect to design parameters. These gradient estimators do not introduce any bias for our case, thus, being suited for stochastic optimization.

## 4 Examples

To assess the efficiency of the methods presented in Chapters 2 and 3, we solve four different numerical examples. The first numerical example is a stochastic optimization problem used to compare the efficiency of SGD, ASGD, and ASGD-restart in the minimization of a quadratic function. In the second example, we solve a simple OED problem with quadratic model in order to draw comparisons among the performances of DLMC, MCLA, and DLMCIS estimators using different optimization methods (SGD, ASGD, and ASGD-restart). In the third example, we address the optimization of strain gauge positioning on a beam modeled following Timoshenko beam theory for estimating the beam's mechanical properties. The purpose of example 3 is to show that the proposed OED framework can reproduce the engineer's intuition. That is, that the optimized design found is consistent with what an engineer would intuitively expect. In the fourth and last example, we optimize an EIT experiment; we identify the optimal currents that, when imposed on the electrodes, maximize the expected information gain about ply orientations in a composite material.

The main interest in the numerical examples presented in this thesis is the evaluation of the performance and robustness of the methods discussed in Sections 2 and 3.2. Thus, the numerical examples do not contemplate practical aspects that may be important for actual experiments. Details regarding the experiment evaluation are responsibility of the experiment designer.

### 4.1 Example 1: Stochastic quadratic function

In this first example, we evaluate the performance of stochastic optimization algorithms on finding the maximum of a quadratic function. The goal of this example is to evaluate the efficiency of the Nesterov's

acceleration and the restart technique when using noisy gradient observations. Moreover, since the problem has a closed-form solution with known optimum and derivatives, we can use the optimal  $q$  tuning as Nemirovski (27), and compare its effect with the restart technique.

We analyze the problem of finding  $\boldsymbol{\xi}$  that maximizes the expected value of a function  $f(\boldsymbol{\xi}, \boldsymbol{\theta})$  with respect to  $\boldsymbol{\theta}$  given as

$$f(\boldsymbol{\xi}, \boldsymbol{\theta}) = - \left( \frac{1}{2} \boldsymbol{\xi}^T \mathbf{A} \boldsymbol{\xi} + \boldsymbol{\xi}^T \mathbf{A} \boldsymbol{\theta} \right), \quad (4.1)$$

where  $\mathbf{A}$  is a diagonal  $n \times n$  matrix with elements  $A_{jj} = j$  for  $j = 1, \dots, n$ . The random vector  $\boldsymbol{\theta}$  is Gaussian-distributed with null mean and covariance matrix  $\boldsymbol{\Sigma}_{\boldsymbol{\theta}} = \text{diag}(\{\sigma_{\theta}^2\}_{i=1}^n)$ . The vector  $\boldsymbol{\xi}$  is a design variable, belonging to  $\Xi$ , a subset of  $\mathbb{R}^n$ . The objective function to be maximized is

$$\mathbb{E}_{\boldsymbol{\theta}} [f(\boldsymbol{\xi}, \boldsymbol{\theta})] = -\mathbb{E}_{\boldsymbol{\theta}} \left[ \frac{1}{2} \boldsymbol{\xi}^T \mathbf{A} \boldsymbol{\xi} + \boldsymbol{\xi}^T \mathbf{A} \boldsymbol{\theta} \right] \quad (4.2)$$

$$= -\frac{1}{2} \boldsymbol{\xi}^T \mathbf{A} \boldsymbol{\xi}. \quad (4.3)$$

Taking the derivative of the objective function with respect to  $\boldsymbol{\xi}$  furnishes

$$\nabla_{\boldsymbol{\xi}} \mathbb{E}_{\boldsymbol{\theta}} [f(\boldsymbol{\xi}, \boldsymbol{\theta})] = -\mathbf{A} \boldsymbol{\xi}, \quad (4.4)$$

hence, the maximum of Eq. 4.2 has closed form

$$\boldsymbol{\xi}^* = \mathbf{0}. \quad (4.5)$$

To approximate  $\nabla_{\boldsymbol{\xi}} \mathbb{E}_{\boldsymbol{\theta}} [f(\boldsymbol{\xi}, \boldsymbol{\theta})]$  for stochastic optimization, we use the gradient of  $f$ ,

$$\mathcal{G}(\boldsymbol{\xi}, \boldsymbol{\theta}) = -\mathbf{A}(\boldsymbol{\xi} + \boldsymbol{\theta}). \quad (4.6)$$

Note that  $\nabla_{\boldsymbol{\xi}} \mathbb{E}_{\boldsymbol{\theta}} [f(\boldsymbol{\xi}, \boldsymbol{\theta})] = \mathbb{E}_{\boldsymbol{\theta}} [\mathcal{G}(\boldsymbol{\xi}, \boldsymbol{\theta})]$ , thus,  $\mathcal{G}$  is an unbiased estimator for the gradient of the objective function. Since  $\mathbf{A}$  is diagonal with elements  $A_{ii} = i$ , the variance of the  $i$ -th element of the estimator  $\mathcal{G}$  is calculated as

$$\mathbb{V}[\mathcal{G}_i(\boldsymbol{\xi}, \boldsymbol{\theta})] = i^2 \mathbb{V}[\theta_i] \quad (4.7)$$

$$= i^2 \sigma_{\theta}^2. \quad (4.8)$$



The variance in the gradient estimation does not depend on  $\boldsymbol{\xi}$  and does not vanish in the optimum. Thus, as  $\nabla_{\boldsymbol{\xi}} \mathbb{E}_{\boldsymbol{\theta}} [f(\boldsymbol{\xi}, \boldsymbol{\theta})]$  converges to zero, the relative error in gradient estimation goes to infinity.

The SGD in Eq. 3.9 for example 1 becomes

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \alpha_k \mathcal{G}(\boldsymbol{\xi}_k, \boldsymbol{\theta}_k) \quad (4.9)$$

$$= \boldsymbol{\xi}_k - \alpha_k \mathbf{A}(\boldsymbol{\xi} + \boldsymbol{\theta}_k), \quad (4.10)$$

where each  $\boldsymbol{\theta}_k$  is independently sampled from the multivariate Gaussian  $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$  each iteration. The accelerated gradient formulation is similarly obtained by replacing  $\mathcal{G}$  in Eq. 3.17.

The estimation of the conditioning number  $L/\mu$  is straightforward in this case, since the Hessian of the objective function is constant and equal to  $\mathbf{A}$ , whose maximum and minimum eigenvalues are, respectively,  $L = n$ , and  $\mu = 1$ . Thus, for this example, the optimal value for the parameter  $q$  is  $q^* = 1/n$ . Similarly, the step-size is set to  $\alpha_0 = 2/(L + \mu) = 2/(n + 1)$ .

We solve the stochastic optimization problem in example 1 using SGD, ASGD, ASGD employing optimal  $q$ , and ASGD with the restart technique. In the numerical tests for example 1, we use  $n = 20$ , thus,  $\alpha_0 = 2/21$  and  $q^* = 1/20$ . Note that  $\boldsymbol{\theta}$  has is  $n$ -dimensional, thus, this example has 20 random parameters. One advantage of SGD is that it can solve high-dimensional problems without any increase in the cost per iteration. Figure 20 presents the convergence of each method towards the optimum using different standard deviations for the prior pdf  $\pi(\boldsymbol{\theta})$ ; on the left,  $\sigma_{\boldsymbol{\theta}} = 0.1$ , and on the right,  $\sigma_{\boldsymbol{\theta}} = 0.01$ . It can be seen that the algorithms behave similarly to their deterministic counterparts up to a certain point, where they start converging sublinearly. As the noise in the gradient estimation becomes large in comparison to its magnitude, the convergence gets dominated by the sublinear term of stochastic gradient. Moreover, when the variance  $\sigma_{\boldsymbol{\theta}}$  is increased to 0.1, the sublinear phase starts sooner.

As discussed in Section 3.4.1, in ASGD the Nesterov's acceleration

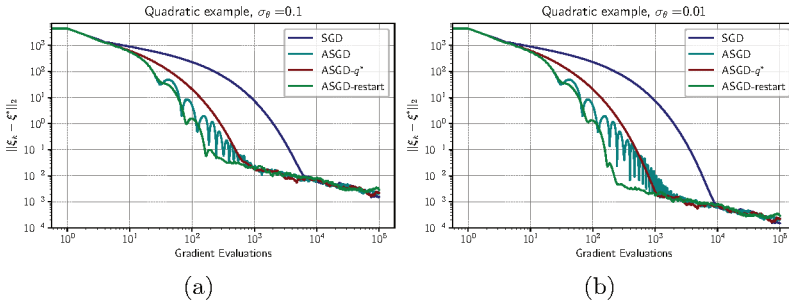


Figure 20 (Example 1): Convergence of the methods with standard deviations  $\sigma_\theta = 0.1$  (a) and  $\sigma_\theta = 0.01$  (b).

imposes an excessive momentum that generates oscillations over the optimum. For this example, the optimal tuning of  $q$  do not improve on ASGD; however, the restart technique speed up the convergence without the need for any prior knowledge about the Hessian of the objective function. Observing Figure 20, it can be seen that ASGD-restart achieves the asymptotic phase at around 300 gradient evaluations, whereas SGD takes almost 10000 gradients to get to the sublinear regime.

## 4.2 Example 2: OED with quadratic model

Here, we consider an OED problem based on a quadratic forward model we devised to perform a comparative analysis between the gradients of EIG estimators presented in Chapter 3.5. We are interested in assessing the efficiency of the gradients of DLMC, MCLA, and DLMCIS estimators on the solution of OED problems. Moreover, we are interested in evaluating the performance of the optimization methods presented, FGD, SGD, ASGD, and ASGD-restart, when coupled with the gradients of DLMC, MCLA, and DLMCIS estimators. Since  $q^*$  is hard to estimate, and ASGD-restart performed better than ASGD- $q^*$  in Example 1, we opt to focus on ASGD-restart as our main optimization algorithm. In FGD, we use the full-gradients of the OED estimators, whereas, in the other (stochastic) optimization methods, we use the gradients derived for stochastic optimization in Section 3.5, noted as  $\mathcal{G}$ .

In this example, the forward model is

$$g(\boldsymbol{\xi}, \theta) = \boldsymbol{\xi}^T \mathbf{A} \boldsymbol{\xi} \theta - \boldsymbol{\xi}^T \mathbf{A} \mathbf{1} \theta^2 - 8 \mathbf{1} \theta - 1, \quad \text{where } \mathbf{A} = \begin{bmatrix} 1 & -0.2 \\ -0.2 & 0.5 \end{bmatrix}, \quad (4.11)$$

the scalar random variable  $\theta$  is sampled from the prior pdf  $\pi(\theta) = \mathcal{N}(0, 10^{-4})$ , and  $\boldsymbol{\xi} \in \Xi = [-2, 2]^2 \subset \mathbb{R}^2$ . For DLMC and DLMCIS gradients,  $\nabla_{\boldsymbol{\xi}} \mathbf{g}$  need to be evaluated, and MCLA gradient requires  $\nabla_{\theta} \mathbf{g}$  and  $\nabla_{\boldsymbol{\xi}} \nabla_{\theta} \mathbf{g}$ . For this problem, closed forms of the model derivatives can be obtained as

$$\nabla_{\boldsymbol{\xi}} \mathbf{g}(\boldsymbol{\xi}, \theta) = 2 \mathbf{A} \boldsymbol{\xi} \theta - \mathbf{A} \mathbf{1} \theta^2, \quad (4.12)$$

$$\nabla_{\theta} \mathbf{g}(\boldsymbol{\xi}, \theta) = \boldsymbol{\xi}^T \mathbf{A} \boldsymbol{\xi} - 2 \boldsymbol{\xi}^T \mathbf{A} \mathbf{1} \theta - 8 \mathbf{1}, \quad (4.13)$$

$$\nabla_{\boldsymbol{\xi}} \nabla_{\theta} \mathbf{g}(\boldsymbol{\xi}, \theta) = 2 \mathbf{A} \boldsymbol{\xi} - 2 \mathbf{A} \mathbf{1} \theta. \quad (4.14)$$

The additive error is assumed to be Gaussian  $\epsilon \sim \mathcal{N}(0, 10^{-4})$  and the number of experiments is  $N_e = 1$ . The initial step-size is chosen as  $\alpha_0 = 1.00$ .

The efficiency criterion we use to compare different methods is defined as the average number of calls of the forward model (NCFM) required to approximate  $\boldsymbol{\xi}^*$  for a given tolerance. We compute NCFM as the mean value of ten independent runs (because of the randomness of SGD), where we aim for an error tolerance of 0.01, i.e.,  $\|\boldsymbol{\xi}_k - \boldsymbol{\xi}^*\|_2 \leq 0.01$ .

To approximate the inner loop in DLMC and DLMCIS, we use the optimal sampling from Beck et al. (2), which we evaluate at the starting point of the optimization and keep constant during the process. To achieve the tolerance of 0.01 in the error of the optimum estimation, the optimal numbers of MC samples are  $N_{\text{DLMC}}^* = 2447$  and  $M_{\text{DLMC}}^* = 80$  for DLMC,  $N_{\text{DLMCIS}}^* = 2402$  and  $M_{\text{DLMCIS}}^* = 7$  for DLMCIS, and  $N_{\text{MCLA}}^* = 966$  for MCLA. We use the same values for their SG estimators, except that  $N = 1$  is used. We use the Algorithm 3 to estimate  $\hat{\theta}$  in (2.25) for DLMCIS.

We solve this problem for each combination of the optimization methods (FGD, SGD, ASGD, ASGD-restart), and gradients of EIG estimators (DLMC, MCLA, DLMCIS) to compare their performances. Table 2 presents mean NCFM for each different combinations of the optimization methods and gradient estimators for a hundred independent runs, except for FGD using DLMC, where, because of the high cost, only ten runs were performed. The optimization methods are indicated at the top of each column, and the gradient estimators in Section 2.2 are listed by row.

Table 2 – Mean NCFM over a hundred independent runs to achieve  $\|\xi_k - \xi^*\|_2 \leq 0.01$ .

	FGD	SGD	ASGD	ASGD-restart
DLMC	$2.99 \times 10^7$	$1.68 \times 10^5$	$9.94 \times 10^3$	$1.18 \times 10^4$
DLMCIS	$6.57 \times 10^6$	$3.18 \times 10^4$	$3.17 \times 10^3$	$2.56 \times 10^3$
MCLA	$2.80 \times 10^5$	$4.06 \times 10^3$	$2.87 \times 10^2$	<b><math>2.75 \times 10^2</math></b>

By analyzing the first line of Table 2, we see that the two methods using Nesterov’s acceleration (ASGD and ASGD-restart) reduce the computational burden by three to four orders of magnitude compared to FGD. Moreover, when using MCLA, ASGD-restart estimates  $\xi^*$  in fewer than 300 calls of the forward model. For this example, the Laplace method and the Nesterov acceleration both performed as expected and reduced the cost to solve the OED problem. Moreover, the coupling of the methods worked; the Laplace approximation did not affect the Nesterov’s acceleration. The convergence to the optimum of SGD, ASGD, and ASGD-restart using the gradient of MCLA estimator is presented in Figure 21. It can be seen that the acceleration speeds up the convergence, and, moreover, that the restart technique results in smoother convergence.

To further evaluate Nesterov’s acceleration combined with the EIG estimators, we solve the problem using ASGD-restart with MCLA and with DLMCIS using three different values for the inner sample-size,

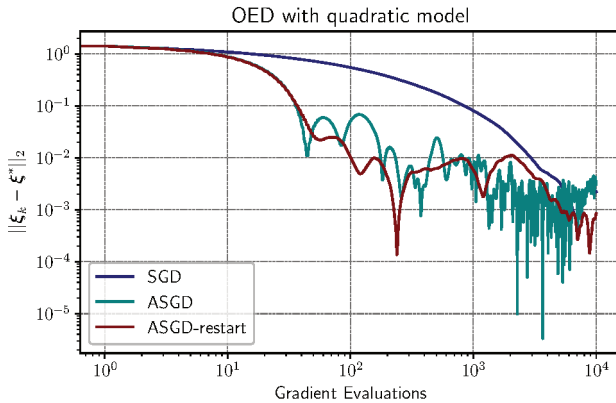


Figure 21 (Example 2) Convergence to the optimum in relation to iterations for SGD, ASGD, and ASGD-restart with MCLA.

1, 10, and 100. The convergence of the distance to the optimum per model evaluation is presented in Figure 22. Even for  $M = 1$ , DLMCIS

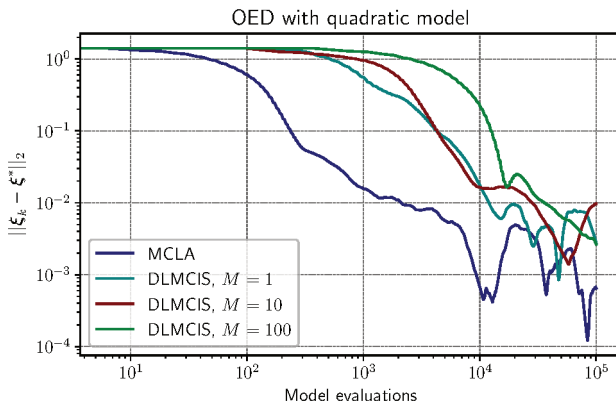


Figure 22 (Example 2) Convergence to the optimum in relation to iterations for the ASGD-restart with MCLA and with DLMCIS.

does not suffer with numerical underflow. However, for this example, using MCLA resulted in a faster convergence than DLMCIS with any

of the sample-sizes used.

Figure 23 shows the contour of EIG and the optimization path for ASGD-restart using MCLA and DLMCIS with the different inner-loop sample sizes. Each method in Figure 23 has only a thousand gradient evaluations. It can be seen that MCLA is the only one to be close to the optimum.

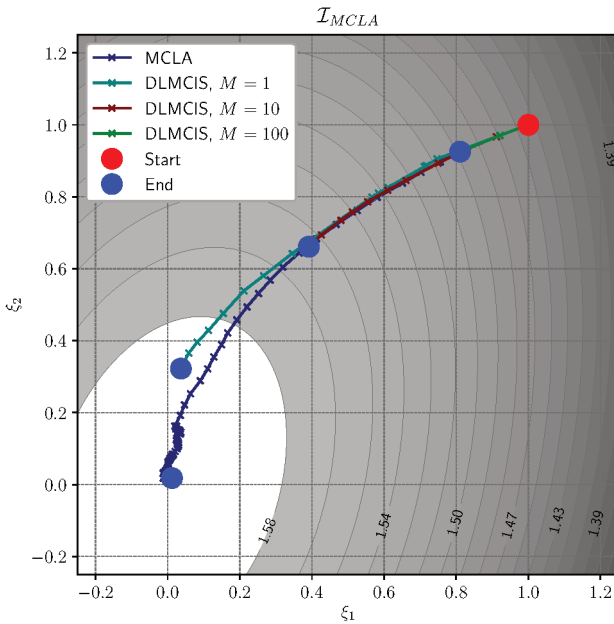


Figure 23 (Example 2) Contour of EIG and optimization ascent paths for the ASGD-restart with MCLA and with DLMCIS.

As a sanity check to estimate the intrinsic bias of the Laplace approximation in the optimization carried out with the estimator MCLA, we compute the expected value of gradient using DLMCIS at the optimum found. Using  $N = 10^5$  and  $M = 10^3$  in DLMCIS, we obtain a gradient with a norm of  $10^{-6}$ , thus, the bias introduced by the Laplace approximation is negligible in this case.

### 4.3 Example 3: Strain gauge positioning on Timoshenko beam

To test the OED machinery developed on the present thesis on a problem with physical meaning, we study the purely academic engineering problem of finding the optimal placement of a strain-gauge on a beam to estimate Young and shear deformation moduli. In this problem, we opt to use the gradient of MCLA, given its lower cost in comparison to DLMCIS. We compare SGD, ASGD, and ASGD-restart in the maximization of  $\mathcal{I}_{MCLA}$ .

We consider that the strain-gauges provide (noisy) strains observations in the vertical and longitudinal axes for a given point of the domain of the beam. We characterize the beam's mechanical properties, namely the Young modulus  $E$  and the shear modulus  $G$ , given measurements obtained from the strain gauge using the Timoshenko beam model. The beam to be studied has 10 m length, 2 m height, and 0.1 m base width. The beam's geometry is not consistent with engineering practice and is devised to furnish an interesting OED problem on the stochastic optimization perspective: we want an optimization problem where the optimum is not in a vertex of the beam. A uniform load  $q_o$  of 1.00 kN/mm is imposed on the beam's vertical axis and distributed along its main axis. The geometry of the beam, the load, and the position of the strain gauge are illustrated in Figure 6. We aim to locate a

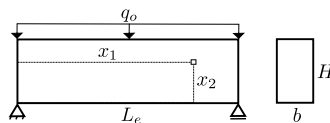


Figure 24 – (Example 3) Geometry of the Timoshenko beam.

strain gauge on the beam that maximizes the information on  $E$  and  $G$ . We model the beam following Timoshenko's theory (37), a mechanical model that captures the strains resulting from both normal and shear

stresses. The Timoshenko beam model for our case is

$$\begin{cases} K_s G A_r \varepsilon_{12} = \frac{q_o L_e}{2} - q_o x_1, \\ E I_n \varepsilon_{11} = \frac{q_o x_1 (L_e - x_1)}{2} x_2, \end{cases} \quad (4.15)$$

where  $\varepsilon_{11}$  is the normal strain,  $\varepsilon_{12}$  is the shear strain,  $x_1$  and  $x_2$  are the positions of the strain gauge on the horizontal and vertical axes respectively,  $q_o$  is the uniform load,  $L_e$  is the length of the beam,  $I_n$  is the inertia moment of the cross section,  $K_s$  is the Timoshenko constant ( $K_s = 5/6$  in all test cases), and  $A_r$  is the cross-section area. A deduction of the Timoshenko beam equations from the elasticity equations is presented in Appendix C.

### 4.3.1 Bayesian formulation

The optimal position for the strain gauge that provides the maximum information about  $\mathbf{E}$  and  $\mathbf{G}$  is denoted by  $\boldsymbol{\xi}^* = (x_1^*, x_2^*)$ . The longitudinal strain on the main axis of the beam, denoted by  $\varepsilon_{11}$ , together with the transverse strain  $\varepsilon_{12}$ , compose the output of the forward model. Therefore, based on (4.15), we find that

$$\begin{aligned} \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}) &= (\varepsilon_{11}(\boldsymbol{\xi}, \boldsymbol{\theta}), \varepsilon_{12}(\boldsymbol{\xi}, \boldsymbol{\theta})) \\ &= \left( \frac{\xi_2 (q_o L_e \xi_1 - q_o \xi_1^2)}{2\theta_1 I_n}, \frac{\frac{L_e}{2} q_o - q_o \xi_1}{K_s \theta_2 A_r} \right), \end{aligned} \quad (4.16)$$

where  $(x_1, x_2)$  and  $(E, G)$  are replaced by  $(\xi_1, \xi_2)$  and  $(\theta_1, \theta_2)$ , respectively. The additive error of the measurement is Gaussian  $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_\epsilon)$ , where the noise covariance matrix is  $\boldsymbol{\Sigma}_\epsilon = \text{diag} \{ \sigma_{\epsilon_1}^2, \sigma_{\epsilon_2}^2 \}$ .

### 4.3.2 Test cases

We assess the robustness of the proposed methods in four test cases, in which we attempt to locate the optimal strain-gauge placement on a beam. We test all the different cases, changing the variance of the prior pdf of  $\boldsymbol{\theta}$ , the dispersion of the measurement noise, and the number of experiments. The prior pdf of  $\boldsymbol{\theta}$  is Gaussian with the distribution



$\pi(\boldsymbol{\theta}) \sim \mathcal{N}((\mu_{pr}^E, \mu_{pr}^G)^T, \text{diag}\{(\sigma_{pr}^G)^2, (\sigma_{pr}^E)^2\})$ , where  $\mu_{pr}^E = 30.00$  GPa and  $\mu_{pr}^G = 11.54$  GPa.

Table 3 presents the parameters used in each of the four cases.

Table 3 – Parameters for the Timoshenko beam problem (Example 3).

Parameter	$N_e$	$\sigma_{pr}^E$ (GPa)	$\sigma_{pr}^G$ (GPa)	$\sigma_{\epsilon_1}$ ( $\times 10^{-4}$ )	$\sigma_{\epsilon_2}$ ( $\times 10^{-4}$ )
Case 1	3	9.00	3.46	6.25	1.30
Case 2	1	6.00	2.31	3.75	0.78
Case 3	1	6.00	0.46	3.75	0.78
Case 4	1	1.20	2.31	3.75	0.78

We devised the parameters of the four cases with the intention of having four different optimization problems. In the first two cases, both  $E$  and  $G$  have standard deviation of, respectively, 30% and 20% of their means. Also, case 1 has larger observation noise than case 2, and more experiments, 3. The third and fourth cases are exactly like case 2, except that case 3 has significantly less dispersion in  $G$ , a coefficient of variation of 4%, and case 4 has the same coefficient of variation for  $E$ .

To evaluate the efficiency of each optimization method in the solution of each case, we run 10000 iterations of each method for each case. Since MCLA is used for this problem, the cost per iteration is  $(\dim(\boldsymbol{\theta}) + 1)(\dim(\boldsymbol{\xi}) + 1)$  forward model evaluations. Thus, since the dimension of  $\boldsymbol{\theta}$  is two, and the dimension of  $\boldsymbol{\xi}$  is also two, the NCFM each method uses is 90000. We compare how close to each optimum each method can get with this fixed cost. The optimization paths for the placement of the strain gauges on the beam are drawn against contour plots of the expected information gain across the optimization domain in Figure 25.

In cases 1 and 2, the optima are similarly located near the bottom of the beam, between the middle and the end. In case 3, the optimum is located in the bottom-middle of the beam; in case 4, the optimum is located on the supports. These placements are expected, as the Young

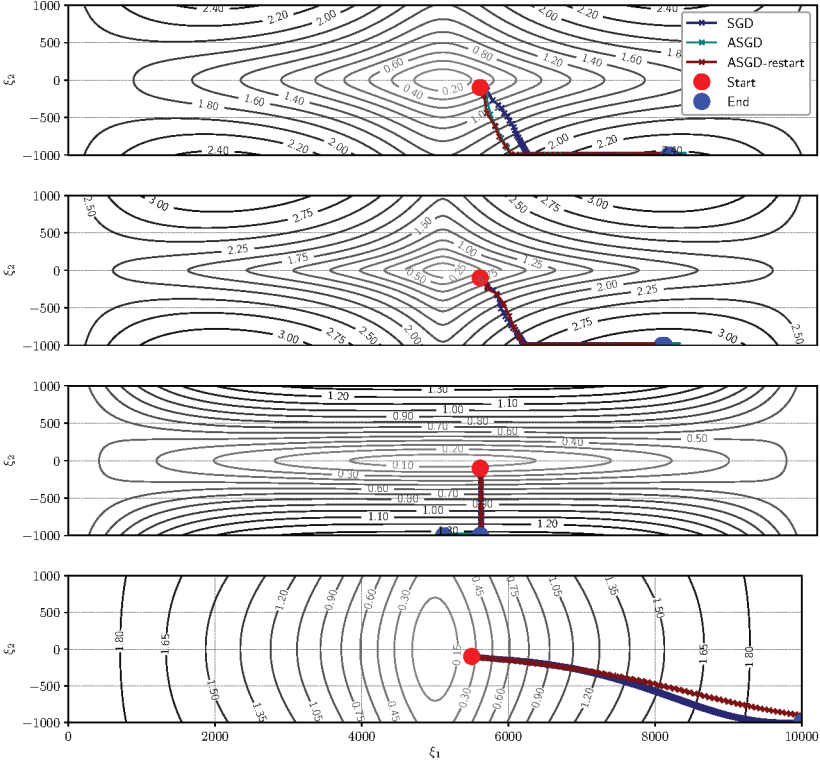


Figure 25 From top to bottom, cases 1 to 4 from Example 3 (summarized in Table 4). Expected information gain contours computed with MCLA and optimization ascent paths using SGD, ASGD, and ASGD-restart with MCLA.

modulus depends on the bending moment (that is maximum at the middle of the beam ( $x_1 = L_e/2$ )), and the shear modulus depends on the shear stress (that is maximum at the beam supports ( $x_1 = 0$  and  $x_1 = L_e$ )). In case 3, the prior information about  $G$  is more accurate; consequently, the algorithm converges to the middle of the beam where more information about  $E$  can be collected. Conversely, in case 4, the algorithm converges to the beam supports, where data is more informative about  $G$ . In Table 4, we present the initial guesses, the optimized setups, the respective expected information gains in relation

to the prior, and the standard deviations of the posterior pdfs of the parameters  $E$  and  $G$  for the four cases. The posteriors are evaluated at  $\hat{\theta} = (\mu_{pr}^E, \mu_{pr}^G)$  for the four cases are presented in Figure 27. We observe a reduced variance in the optimized experiment, compared to the original, reflecting the importance of an informative experiment. In cases 3 and 4, no information is acquired about  $G$  and  $E$ , respectively, since the variances in the axes are not reduced, compared to the prior.

Table 4 – Results from the Timoshenko beam problem (Example 3).

		$x_1^*$ (mm)	$x_2^*$ (mm)	$\mathcal{I}_{\text{MCLA}}$	$\sigma_{post}^E$ (GPa)	$\sigma_{post}^G$ (GPa)
Case 1	Non-Opt.	5500.00	-100	0.14	8.00	2.40
	Opt.	8022.59	-1000.00	2.43	2.48	0.54
Case 2	Non-Opt.	5500.00	-100	0.23	2.38	1.38
	Opt.	7962.77	-1000.00	3.35	1.60	0.74
Case 3	Non-Opt.	5500.00	-100	0.06	5.70	0.46
	Opt.	5004.47	-1000.00	1.28	1.72	0.46
Case 4	Non-Opt.	5500.00	-100	0.22	1.20	1.93
	Opt.	10000.00	-1000.00	1.94	1.20	0.33

Because we use the biased and inconsistent MCLA estimator of the gradient, as a sanity check, we evaluate the gradient at the optima we found (the first two cases) using the full gradient of the DLMCIS estimator with  $N = 10^3$  and  $M = 10^2$ . In both cases, the gradient norm is below  $10^{-3}$ , meaning that the bias of the Laplace approximation is considerably small at the optima. We conclude that the biased solutions found are not significantly distant to the real optima. To plot the convergence, we estimate the real optima using DLMCIS with FGD from the optima found using the MCLA. The convergences from the first two cases are presented in Figure 26.

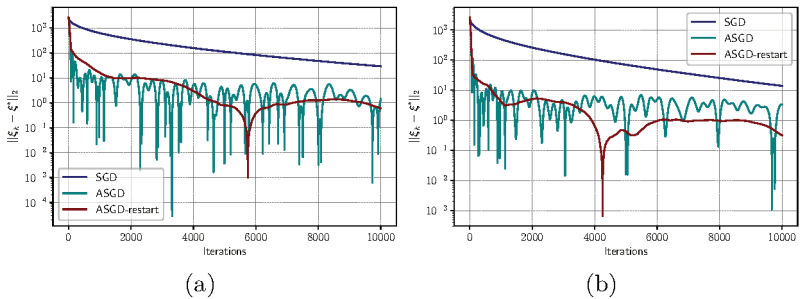


Figure 26 Convergences from cases 1 (a) and 2 (b) using ASGD-restart with tolerance of 1 mm, or relative tolerance of  $10^{-4}$  (Example 3).

The main goal of this example is to visualize how the optimum found is consistent with the input data given. Moreover, to show that the ASGD-restart with the gradient of MCLA is able to solve the OED problem proposed.

#### 4.4 Example 4: Electrical impedance tomography

EIT is an imaging technique that infers the conductivity of a closed body from potential measurements obtained from electrodes placed on the boundary surface of the body. Here, we consider the optimal design of an EIT experiment conducted on two orthotropic plies, in which the potential field is assumed to be quasi-static. The

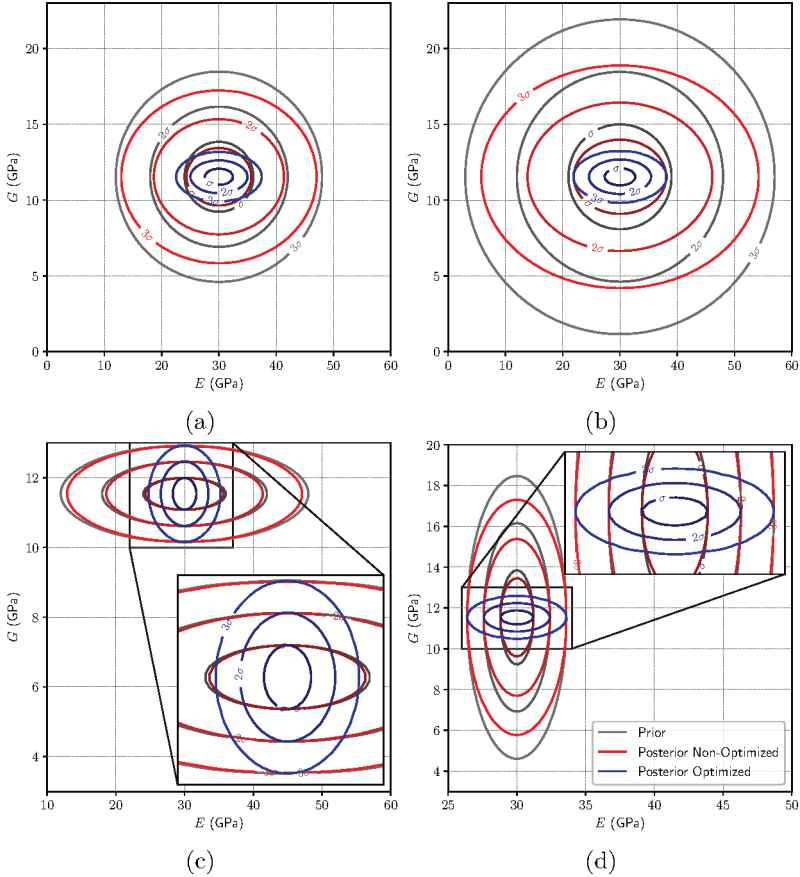


Figure 27 Prior, posterior, and optimized posterior pdfs for the Young modulus  $E$  and the shear modulus  $G$  for cases 1 (a), 2 (b), 3 (c), and 4 (d) from Example 3.

physical phenomenon is governed by a second-order partial differential equation combined with the complete electrode boundary model (38). Beck et al. (2) prove that the bias of the Laplace approximation for this problem is negligible. Therefore, in this example, we use MCLA combined with ASGD-restart.

#### 4.4.1 Bayesian setting

We consider a body  $D$  that is 20 cm long and composed of two plies that are each 1 cm thick, resulting in a total thickness of 2 cm. Both plies are made of the same material, but are oriented at different angles. The conductivity of each ply is  $\bar{\sigma}(\boldsymbol{\theta}, \mathbf{x}) = \mathbf{Q}^T(\theta_i) \cdot \boldsymbol{\sigma} \cdot \mathbf{Q}(\theta_i)$ , where  $\boldsymbol{\sigma} = \text{diag}\{10^{-2}, 10^{-3}, 10^{-3}\}$ , and  $\mathbf{Q}(\theta_i)$  is an orthogonal matrix that rules the rotation of the unknown orientation angle  $\theta_i$  of ply  $i$ , counting from bottom to top. The objective is to infer  $\theta_1$  and  $\theta_2$ , about which we assume the prior information to be  $\pi(\theta_1) \sim \mathcal{U}(\frac{\pi}{4.5}, \frac{\pi}{3.5})$  and  $\pi(\theta_2) \sim \mathcal{U}(-\frac{\pi}{3.5}, -\frac{\pi}{4.5})$ . During the EIT experiment, low-frequency electrical currents are injected through the electrodes  $E_l$  (with  $l = 1, \dots, N_{el}$ ) attached to the boundary of the body, with  $N_{el}$  being the number of electrodes. The potentials at the electrodes are calculated as

$$\mathbf{y}_i(\boldsymbol{\xi}) = \mathbf{g}_h(\boldsymbol{\xi}, \boldsymbol{\theta}_t) + \boldsymbol{\epsilon}_i \stackrel{\text{def}}{=} \mathbf{U}_h(\boldsymbol{\xi}, \boldsymbol{\theta}_t) + \boldsymbol{\epsilon}_i, \quad \text{for } i = 1, \dots, N_e, \quad (4.17)$$

where  $\mathbf{y}_i \in \mathbb{R}^{N_{el}-1}$ , and  $\boldsymbol{\theta}_t = (\theta_{t,1}, \theta_{t,2})$  are the true orientation angles that we intend to infer. In the Bubnov–Galerkin sense,  $\mathbf{U}_h = (U_1, \dots, U_{N_{el}-1})$  is the finite elements approximation (i.e., the potential at the electrodes) of  $\mathbf{U}$  from the following variational problem: find  $(u, \mathbf{U}) \in L_{\mathbb{P}}^2(\Theta; \mathcal{H})$  such that

$$\mathbb{E}[B((u, \mathbf{U}), (v, \mathbf{V}))] = \mathbf{I}_e \cdot \mathbb{E}[\mathbf{U}], \quad \text{for all } (v, \mathbf{V}) \in L_{\mathbb{P}}^2(\Theta; \mathcal{H}). \quad (4.18)$$

where  $\mathbf{I}_e$  represents the values of injected current at  $N_{el} - 1$  electrodes  $\mathbf{I}_e = (I_{e_1}, \dots, I_{e_{N_{el}-1}})^T$ . Let the constitutive relation for the current flux be  $\mathbf{j}(\boldsymbol{\theta}, \mathbf{x}) = \bar{\boldsymbol{\sigma}}(\boldsymbol{\theta}, \mathbf{x}) \cdot \nabla u(\boldsymbol{\theta}, \mathbf{x})$ . Then, the bilinear form  $B : \mathcal{H} \times \mathcal{H} \rightarrow \mathbb{R}$  is

$$B((u, \mathbf{U}), (v, \mathbf{V})) = \int_D \mathbf{j} \cdot \nabla v dD + \sum_{l=1}^{N_{el}} \frac{1}{z_l} \int_{E_l} (U_m - u)(V_m - v) dE_l, \quad (4.19)$$

where  $z_l$  is the surface contact impedance between the electrode  $l$  and the surface of the body. The space of the solution for the potential field  $(u(\boldsymbol{\theta}), \mathbf{U}(\boldsymbol{\theta}))$  is  $\mathcal{H} \stackrel{\text{def}}{=} H^1(D) \times \mathbb{R}_{\text{free}}^{N_{el}}$  for a given random event  $\boldsymbol{\theta} \in \Theta$ ,

where  $H^1$  is the Sobolev space of functions that belong to  $L^2$ , and whose first-order partial derivatives also belong to  $L^2$ . Then,  $L_{\mathbb{P}}^2(\Theta; \mathcal{H})$  is the Bochner space given by

$$L_{\mathbb{P}}^2(\Theta; \mathcal{H}) \stackrel{\text{def}}{=} \left\{ (u, \mathbf{U}) : \Theta \rightarrow \mathcal{H} \text{ s.t. } \int_{\Theta} \|(u(\boldsymbol{\theta}), \mathbf{U}(\boldsymbol{\theta}))\|_{\mathcal{H}}^2 d\mathbb{P}(\boldsymbol{\theta}) < \infty \right\}. \quad (4.20)$$

The measurement-error distribution is  $\epsilon \sim \mathcal{N}(0, 100.0)$ . We note that, by imposing the Kirchhoff law on  $\mathbf{I}_e$  and the zero-potential law on  $\mathbf{U}_h$ , the model output  $\mathbf{g}$  is projected to a suitable space for the optimization.

The optimization parameters are defined as the current intensity to be injected through the electrodes, i.e.,  $\boldsymbol{\xi} = (\{I_e\}_{i=1}^{N_{el}})$ , where each  $I_e$  is the normalized current intensity applied to the  $i$ -th electrode such that  $I_e \in [-1, 1]$ . A schematic of the experimental setup showing the laminated material with four electrodes is depicted in Figure 28.



Figure 28 Experimental configuration for EIT with two plies and four electrodes (Example 4).

#### 4.4.2 Numerical tests for EIT

To evaluate the efficiency of MCLA combined with ASGD-restart in solving the EIT problem, we solve four different cases using different numbers of electrodes of different lengths and positions. In all cases, the number of experiments is  $N_e = 1$ . In all cases, a fixed number of iterations is defined as 1000. Since the gradient of the MCLA estimator for stochastic optimization is used, the cost of each iteration is  $(\dim(\boldsymbol{\theta})+1)(\dim(\boldsymbol{\xi})+1)$  forward model evaluations, thus, for each case



the cost of iteration is different. The third case has ten design variables, thus, each iteration costs 33 NCFM. Thus, in the most expensive case, 33000 finite element analysis are used. The evaluation of the optimization for each case takes less than half an hour in a personal computer, however, the running time can change depending on the computer used.

#### 4.4.2.1 Test case 1 (Configuration with four electrodes and one design variable)

We aim to find the most informative current intensity to inject through three out of the four electrodes attached to the two-ply composite material described above and shown in Figure 28. The current at the fourth is defined by Kirchoff's law. The electrodes are 1 cm long and have fixed positions.

We approximate the covariance of the posterior pdf for each  $\xi$  by  $\Sigma_{post}(\xi)$ , as presented in (2.27). Thus, the approximated covariances at the initial guess and the optimum solution are

$$\begin{aligned} \Sigma_{post}(\xi_0) &= \begin{bmatrix} 7.21 \times 10^{-3} & 9.73 \times 10^{-4} \\ 9.73 \times 10^{-4} & 1.35 \times 10^{-4} \end{bmatrix}, \\ \Sigma_{post}(\xi^*) &= \begin{bmatrix} 5.39 \times 10^{-6} & 3.21 \times 10^{-6} \\ 3.21 \times 10^{-6} & 3.39 \times 10^{-6} \end{bmatrix}. \end{aligned} \quad (4.21)$$

The optimization reduces the terms in the covariance matrices by two orders of magnitude, meaning that the optimized experiment provides more precise estimates of QoI. Due to the symmetry of the problem, there are two local maxima, one with  $\xi_1 = -1$  and one with  $\xi_1 = 1$ . However, the local maximum where  $\xi_1 = 1$  is also the global maximum, with a larger expected information gain. Therefore, we conclude that we can obtain more information about the angles of the plies from the optimized configuration than from the non-optimized configuration.

In Figure 29, we present the electric potential and the current

streamlines both before and after the optimization. We also present the expected information gain when using the MCLA estimator with the optimization path and the posteriors evaluated at  $\hat{\theta} = (\frac{\pi}{3.9375}, -\frac{\pi}{3.9375})$ . The initial guess provides less information about  $\theta_1$  than about  $\theta_2$ .

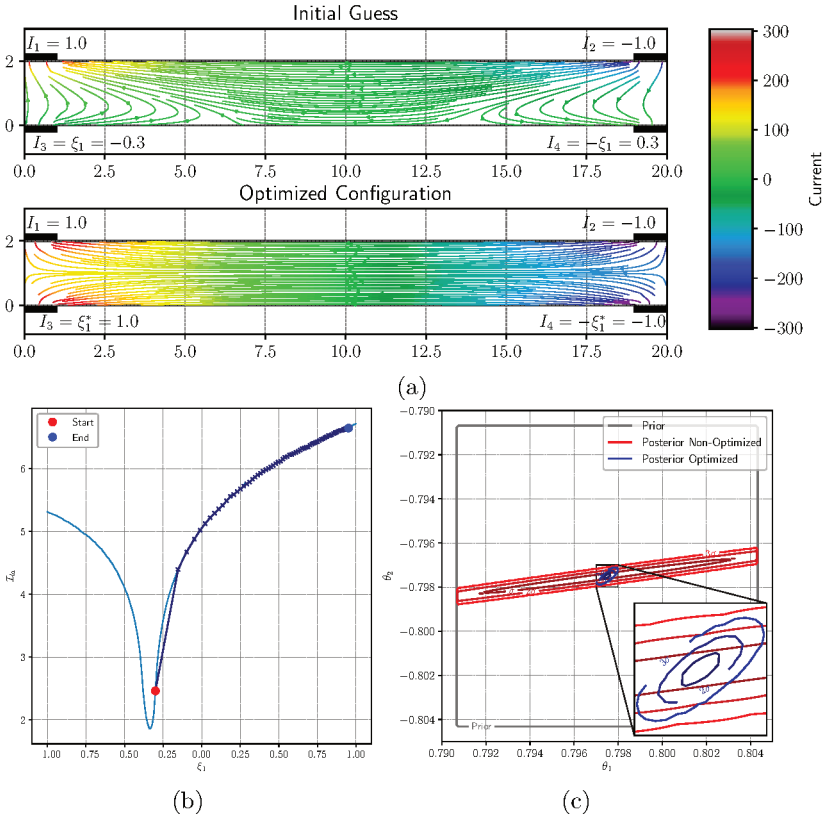


Figure 29 Current streamlines (a), optimization path (b), and pdfs of both the initial and optimized configurations (c) for case 1 (Example 4).

However, the optimized position significantly reduces the variance of the  $\theta_1$  estimation and provides insightful information on both parameters  $\theta_1$  and  $\theta_2$  with almost the same uncertainty.

#### 4.4.2.2 Test case 2 (Configuration with three electrodes and two design variables)

Here, we consider a configuration for the EIT experiment with two electrodes on the top of the two-ply composite body, and one at the bottom, each 4 cm long. We allow the current applied to the two top electrodes to vary from  $-1$  to  $1$ , i.e., the optimization variables are  $\xi = (I_e^1, I_e^2)$ . We enforce Kirchoff's law by letting  $I_e^3 = -I_e^1 - I_e^2$ , and force box constraints on  $I_e^3$  by imposing the constraint  $|I_e^1 + I_e^2| < 1$  by projection. The contour plot of the expected information gain and the ascent paths of two different initial guesses are presented in Figure 30, in which the regions illustrated in blue are where the box constraint on  $I_e^3$  are violated. The optimization is presented for the two initial guesses over the contour lines of the expected information gain. The region shaded in gray indicates where the experiment does not provide any information gain, i.e., where  $\mathcal{I} = 0$ .

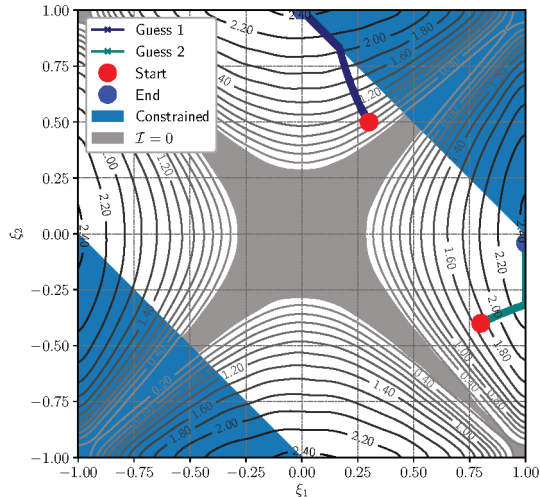


Figure 30 (Example 4) Contour of  $\mathcal{I}_{\text{MCLA}}$  with optimization paths for EIT test case 2.

Figure 30 shows that the optimization converges to local optima for the two initial guesses, arriving at solutions where the expected

information gain is around 2.4. Figure 31 presents the current streamlines for one of the initial guesses,  $\xi = (0.8, -0.4)$  and the posteriors from both guesses. The two optimized posteriors look alike. As shown in

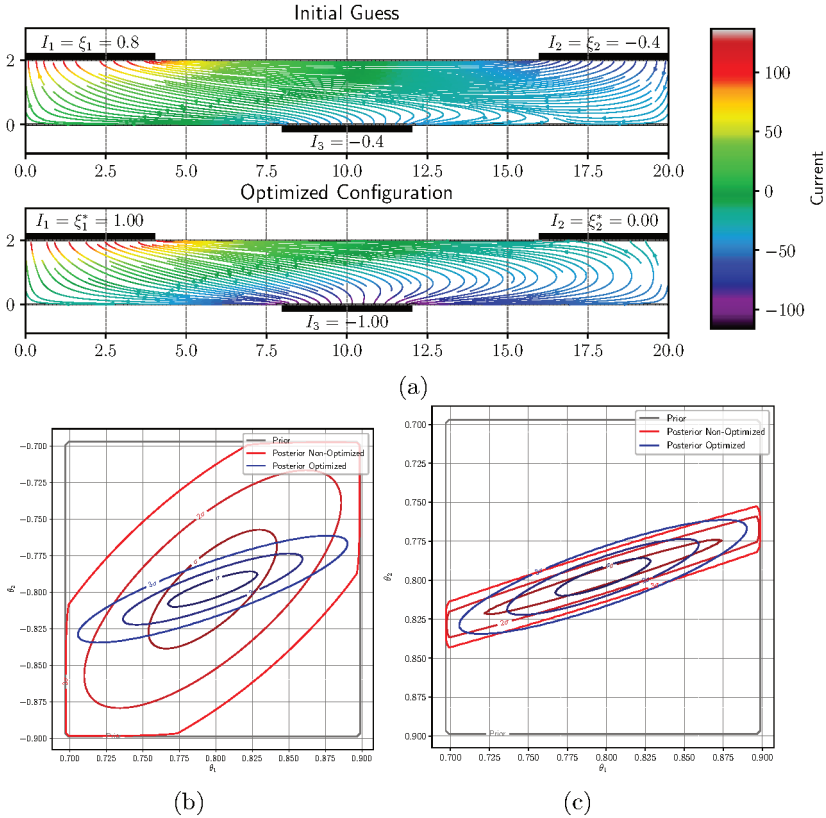


Figure 31 Current streamlines for guess 2 (a) and pdfs for both guess 1 (b) and guess 2 (c) (Example 4).

Figure 30, this problem has four optima:  $(0, 1)$ ,  $(1, 0)$ ,  $(0, -1)$ , and  $(-1, 0)$ . These optima have in common the fact that one of the two electrodes in the top ply ( $\xi_1$  or  $\xi_2$ ) has null-current while the other two electrodes have current 1 or -1.

#### 4.4.2.3 Test case 3 (Configuration with ten electrodes and ten design variables)

We now consider a more complex EIT experiment with ten 2 cm long electrodes. The intensity of the initial current applied is 0.5 at the inlet electrodes (on top of the two-ply composite body) and  $-0.5$  for the outlet electrodes (on the bottom). It is worth to highlight that this problem has ten design variables, thus, each gradient by forward finite differences costs 11 model evaluations, i.e. FEM is evaluated 11 times. Therefore, it is a case where the stochastic optimization framework devised can show its performance by efficiently utilizing the gradient information.

The current streamlines, before and after the optimization, are depicted at the top of Figure 32. The optimization converges to a setup with both positive and negative currents applied on both the top and the bottom electrodes. This optimal setup provides an expected information gain of 7.18. For the sake of comparison, the expected information gain from the setup with currents of 1.0 and -1.0 applied to the top and bottom electrodes, respectively, is only 2.95. On the bottom-left of Figure 32, the posteriors show that the variance of QoI for the optimized configuration is remarkably smaller than for the initial guess. On the bottom-right of the figure, we present the self-convergence test where we see that using Nesterov’s acceleration resulted in an accelerated convergence of the optimizer to the optimum found. The expected information gains for the three cases presented in Example 4 are listed in Table 5.

Table 5 – Expected information gain using MCLA with  $N = 1000$  in Example 4.

	Initial Guess	Optimized
Case 1	2.26	6.72
Case 2, Guess 1	0.64	2.46
Case 2, Guess 2	1.74	2.47
Case 3	1.57	7.18

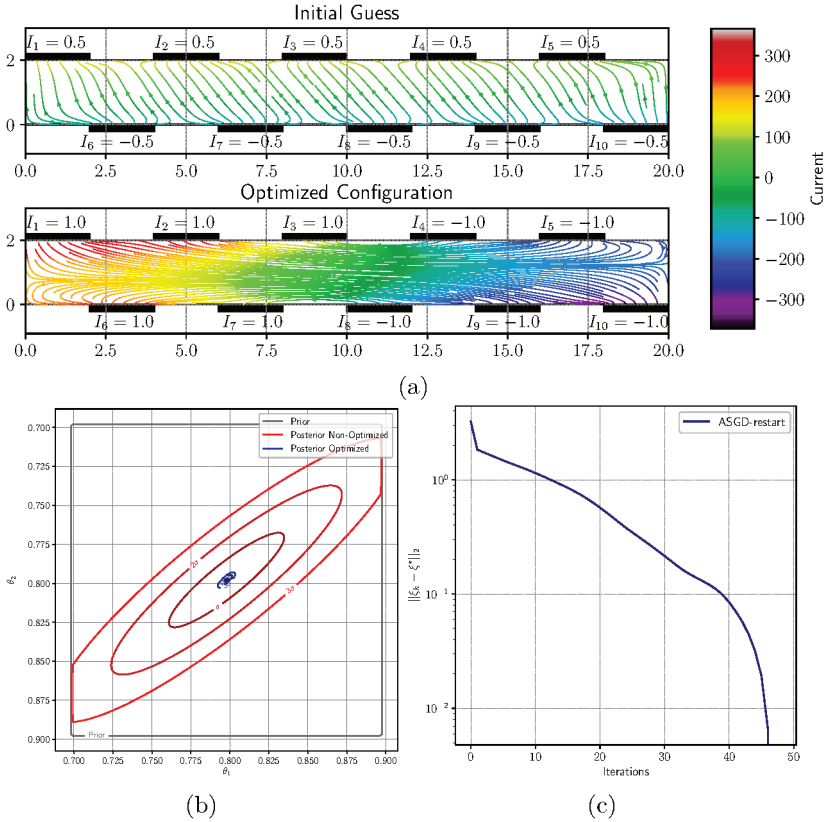


Figure 32 Current streamlines (a), pdfs of initial and optimized configurations (b), and self-convergence to the optimum (c) for case 3.

## 5 Conclusion

Experiments are an important part of science, and, as such, it is interesting to tune the experiments parameters to obtain the most informative observations. However, for many cases, finding the optimum tuning for experiments is not a trivial task. We opted to use Shannon's expected information gain (EIG), a measure based on Bayes' theorem, to estimate the performance of an experiment, in what is called Bayesian optimal experimental design (OED). The Bayesian OED is a very general methodology that can be applied to a wide range of experiments. The price of generality is often high; the standard method to estimate EIG requires the solution of a double-loop Monte Carlo (DLMC) method. The cost of evaluating DLMC to achieve a desired precision can be of the order of millions of physical model simulations. Moreover, DLMC is sensible to the dispersion of the parameters, being susceptible to numerical instabilities, e.g., numerical underflow. Thus, the goal of the present thesis was to study numerical methods in both stochastic optimization and uncertainty quantification in order to solve OED problems efficiently.

We devised a robust stochastic optimization framework to solve OED problems. We followed the same path as Huan and Marzouk (14), and used stochastic gradient descent (SGD) for OED, however, we employed state-of-the-art techniques in stochastic optimization to improve the convergence: Polyak–Ruppert averaging, Nesterov acceleration, and a restart scheme. The SGD does not require the evaluation of the outer-loop of the EIG estimators, consequently reducing the cost of each iteration. In comparison to the full-gradient descent approach, the reduction to the iteration cost is of the order of the outer-loop Monte Carlo sample size, thus, the optimization cost can be dramatically reduced. We also used Nesterov's acceleration with a restart technique to speed up the convergence in stochastic optimization. The restart technique is

proposed by Candès and O’Donoghue (30) for the deterministic case. We adapted the restart method for stochastic optimization by using the gradient approximation instead of the true gradient to decide whether or not to restart the acceleration. The accelerated stochastic gradient descent (ASGD) with the restart technique performs well in the numerical examples evaluated in this thesis, thus, being a viable alternative for stochastic optimization problems. The rationale behind ASGD efficiency is that it takes advantage of acceleration in the starting iterations, before the asymptotic regime. As the optimization progresses, the acceleration is degenerated by the noise in gradient estimation, however, ASGD does not perform worst than SGD in asymptotic regime. Thus, ASGD takes advantage of Nesterov’s acceleration in its preasymptotic phase, arriving faster in the neighborhood of the optimum.

To further improve the efficiency of the proposed OED machinery, we coupled the stochastic optimization framework with a Laplace approximation of the posterior pdf, thus, avoiding the evaluation of the two nested Monte Carlo integrations needed for the gradient of EIG. Long et al. (8) proposed the use of the Laplace method to approximate the integral of the posterior in the context of OED. The Laplace approximation consists in approximating the log of a pdf by a second-order Taylor expansion at its mode, thus, the pdf is approximated by a Gaussian. By using the Laplace approximation of the posterior, we avoided the need of evaluating the evidence, consequently only one of the two nested integrals, the outer integral, needed to be evaluated. The Monte Carlo with Laplace approximation (MCLA) is can be orders of magnitude cheaper than the DLMC estimator, however it is a biased and inconsistent estimator. As a viable alternative for the cases where the Laplace approximation bias is not acceptable, Beck et al. (2) use an importance sampling based on the Laplace method to improve the performance on the MC approximation of the evidence. Beck et al. (2) prove that the double-loop Monte Carlo with Laplace-based importance sampling (DLMCIS) is a consistent estimator and that the reduction in comparison to DLMC in cost can be of several orders of magnitude. We



---

derived the gradients for stochastic optimization of the DLMC, MCLA, and DLMCIS estimators and estimated their costs. These gradients do not have the outer-loop, thus, are cheaper than their deterministic counterparts.

To test the performance of the methods presented in this thesis, we solved four numerical examples. The first example was a non-OED stochastic optimization problem we devised to test the optimization methods and compare their performances. The second was a simple OED problem without any physical meaning; we used a model that is quadratic with respect to both the design parameters and the quantities of interest. The third example was an engineering problem built to be an interesting OED case: we optimized the positioning of strain-gauges on a beam to infer mechanical properties of the material which the beam is made. The fourth and last example was an engineering problem with a model that requires a finite element analysis; we optimized the currents to be imposed on the electrodes during an electrical impedance tomography (EIT) experiment to infer the angles of the plies of laminated composite material.

The general conclusion from the numerical examples was that the stochastic optimization framework for OED is efficient, being able to consistently solve the proposed problems. For the second example, a hundred independent runs were performed with each combination of the optimization algorithms (FGD, SGD, ASGD, ASGD-restart), and the EIG estimators (DLMC, MCLA, DLMCIS). ASGD-restart with MCLA solved the problem with an average of 275 calls of the forward model, whereas DLMC with FGD needed  $2.99 \times 10^7$  model evaluations. These results were obtained for optimized sample-sizes for EIG estimators derived in Beck et al. (2). However, the gradient of the MCLA for stochastic optimization does not require any Monte Carlo integration. The iteration without Monte Carlo integration results in an efficient and cheap optimization process. Consequently, OED problems with expensive models can be optimized. The combination of ASGD-restart and MCLA solved the expensive problem of EIT that would, otherwise,

be unfeasible. The main weakness of the MCLA estimator is the bias introduced from the Laplace approximation, however, for the last two problems, we evaluated the full-gradient of the DLMCIS estimator in the solution found with MCLA as a sanity check. This is a way of measuring if the true gradient is indeed null at the optimum candidate.

We emphasize that the present work is an improvement on the state-of-the-art for simulation based OED with practical purposes for both academic and industrial research, enabling improvements on experimental processes that would not be able with current methods.

## 5.1 Future research

For future research, we suggest:

**Global convergence in multimodal case:** To improve the optimization we presented, we suggest the use of global convergence techniques that can escape local minima. Torii, Lopez, and Luersen (39) apply a probabilistic restart technique that restarts the local search from a point less likely to converge to same local optimum as others. Another approach, proposed by Pagu and Souza (40), is to create a set of random points sampled from a Gaussian distribution centered on the current iteration point and check if any of these points have better objective function value than the current one. Given that the variance of the Gaussian distribution is large enough, the algorithm can jump to regions with better local minima until the global minimum is found.

**Variance reduction techniques:** A viable way of improving the convergence of the optimization process is to use variance reduction methods. One way of reducing the variance is to use minibatches (33), which means to use a small (but larger than one) sample to estimate the gradient each iteration. Another example is the use of control variates as Johnson and Zhang (41) and Nitanda (35). Control variates take advantage of autocorrelation between the gradients in the domain to use information from previous iterations.

---

**Multilevel Monte Carlo:** Moreover, for expensive problems that require the numerical approximation of partial differential equations, multilevel Monte Carlo can be used to approximate the true gradient of EIG using a hierarchical mesh discretization, as proposed by Beck et al.(42).



# Bibliography

- 1 COMMONS, W. *Single-edge notch-bending specimen (also called three-point bending specimen) for fracture toughness testing*. 2011. Disponível em: <<https://commons.wikimedia.org/wiki/File:SingleEdgeNotchBending.svg>>. Cited 2 times on pages 15 and 32.
- 2 Beck, J.; Dia, B. M.; Espath, L. F. R.; Long, Q.; Tempone, R. Fast Bayesian experimental design: Laplace-based importance sampling for the expected information gain. *Computer Methods in Applied Mechanics and Engineering*, v. 334, p. 523–553, 2018. Cited 18 times on pages 15, 31, 32, 33, 34, 39, 47, 49, 51, 53, 54, 56, 57, 65, 91, 102, 112, and 113.
- 3 CHALONER, K.; VERDINELLI, I. Bayesian experimental design: A review. *Statistical Science*, JSTOR, p. 273–304, 1995. Cited 3 times on pages 33, 39, and 53.
- 4 SHANNON, C. E. A mathematical theory of communication, part i, part ii. *Bell Syst. Tech. J.*, v. 27, p. 623–656, 1948. Cited 4 times on pages 33, 39, 44, and 45.
- 5 HUAN, X.; MARZOUK, Y. M. Simulation-based optimal bayesian experimental design for nonlinear systems. *Journal of Computational Physics*, Elsevier, v. 232, n. 1, p. 288–317, 2013. Cited on page 33.
- 6 LINDLEY, D. V. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, JSTOR, p. 986–1005, 1956. Cited 2 times on pages 33 and 45.
- 7 RYAN, K. J. Estimating expected information gains for experimental designs with application to the random fatigue-limit model. *Journal of Computational and Graphical Statistics*, Taylor & Francis, v. 12, n. 3, p. 585–603, 2003. Cited 4 times on pages 33, 41, 48, and 56.
- 8 LONG, Q.; SCAVINO, M.; TEMPONE, R.; WANG, S. Fast estimation of expected information gains for bayesian experimental designs based on laplace approximations. *Computer Methods in Applied Mechanics and Engineering*, Elsevier, v. 259, p. 24–39, 2013. Cited 6 times on pages 33, 39, 52, 56, 112, and 124.

- 9 GHISBAIN, P. *Application of a gradient-based algorithm to structural optimization*. Tese (Doutorado) — Massachusetts Institute of Technology, 2009. Cited on page 34.
- 10 NACHBAGAUER, K.; OBERPEILSTEINER, S.; SHERIF, K.; STEINER, W. The use of the adjoint method for solving typical optimization problems in multibody dynamics. *Journal of Computational and Nonlinear Dynamics*, American Society of Mechanical Engineers, v. 10, n. 6, p. 061011, 2015. Cited on page 34.
- 11 EBADAT, A.; KARIMAGHAEI, P.; MAHDIYAR, H. Application of gradient-based control methods in efficient oil well placement through dynamic fuzzy neural network modeling. In: SPRINGER. *International Conference on Digital Enterprise and Information Systems*. [S.l.], 2011. p. 616–630. Cited on page 34.
- 12 ROBBINS, H.; MONRO, S. A stochastic approximation method. *The annals of mathematical statistics*, JSTOR, p. 400–407, 1951. Cited 4 times on pages 34, 60, 61, and 62.
- 13 BOTTOU, L. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nimes*, v. 91, n. 8, 1991. Cited 2 times on pages 34 and 60.
- 14 HUAN, X.; MARZOUK, Y. Gradient-based stochastic optimization methods in bayesian experimental design. *International Journal for Uncertainty Quantification*, Begel House Inc., v. 4, n. 6, 2014. Cited 5 times on pages 34, 64, 65, 79, and 111.
- 15 POLYAK, B. T.; JUDITSKY, A. B. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, SIAM, v. 30, n. 4, p. 838–855, 1992. Cited 2 times on pages 35 and 65.
- 16 RUPPERT, D. *Efficient estimations from a slowly convergent Robbins-Monro process*. [S.l.], 1988. Cited 2 times on pages 35 and 65.
- 17 NESTEROV, Y. A method of solving a convex programming problem with convergence rate  $o(1/k^2)$ . In: . [S.l.: s.n.], 1983. Cited 3 times on pages 35, 68, and 86.
- 18 ISSAID, C. B. *Bayesian Optimal Experimental Design Using Multilevel Monte Carlo*. Dissertação (Mestrado), 2015. Cited on page 39.

- 19 COVER, T. M.; THOMAS, J. A. *Elements of information theory*. [S.l.]: John Wiley & Sons, 2012. Cited on page 42.
- 20 KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. *The annals of mathematical statistics*, JSTOR, v. 22, n. 1, p. 79–86, 1951. Cited on page 42.
- 21 ROBERT, C.; CASELLA, G. *Monte Carlo statistical methods*. [S.l.]: Springer Science & Business Media, 2013. Cited on page 46.
- 22 HAMADA, M.; MARTZ, H.; REESE, C.; WILSON, A. Finding near-optimal bayesian experimental designs via genetic algorithms. *The American Statistician*, Taylor & Francis, v. 55, n. 3, p. 175–181, 2001. Cited on page 46.
- 23 RUBINSTEIN, R. Y.; KROESE, D. P. *Simulation and the Monte Carlo method*. [S.l.]: John Wiley & Sons, 2016. v. 10. Cited on page 48.
- 24 CAUCHY, A. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris 25, no. 1847*, p. 536–538, 1847. Cited on page 59.
- 25 SCHMIDT, M.; ROUX, N. L.; BACH, F. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, Springer, p. 1–30, 2013. Cited on page 60.
- 26 ALLEN-ZHU, Z. Katyusha: The first direct acceleration of stochastic gradient methods. *arXiv preprint arXiv:1603.05953*, 2016. Cited 2 times on pages 60 and 75.
- 27 NEMIROVSKI, A. Efficient methods in convex programming. 2005. Cited 4 times on pages 64, 65, 67, and 88.
- 28 MOULINES, E.; BACH, F. R. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2011. p. 451–459. Cited on page 64.
- 29 NESTEROV, Y. *Introductory lectures on convex optimization: A basic course*. [S.l.]: Springer Science & Business Media, 2013. Cited 3 times on pages 68, 69, and 70.
- 30 O'DONOGHUE, B.; CANDÈS, E. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, Springer, v. 15, n. 3, p. 715–732, 2015. Cited 6 times on pages 70, 71, 73, 76, 86, and 112.

- 31 SU, W.; BOYD, S.; CANDES, E. J. A differential equation for modeling Nesterov's accelerated gradient method: theory and insights. *Journal of Machine Learning Research*, v. 17, n. 153, p. 1–43, 2016. Cited on page 73.
- 32 LAN, G. An optimal method for stochastic composite optimization. *Mathematical Programming*, Springer, v. 133, n. 1-2, p. 365–397, 2012. Cited on page 75.
- 33 COTTER, A.; SHAMIR, O.; SREBRO, N.; SRIDHARAN, K. Better mini-batch algorithms via accelerated gradient methods. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2011. p. 1647–1655. Cited 2 times on pages 75 and 114.
- 34 NITANDA, A. Stochastic proximal gradient descent with acceleration techniques. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2014. p. 1574–1582. Cited on page 75.
- 35 NITANDA, A. Accelerated stochastic gradient descent for minimizing finite sums. In: *Artificial Intelligence and Statistics*. [S.l.: s.n.], 2016. p. 195–203. Cited 3 times on pages 75, 76, and 114.
- 36 CARLON, A. G.; DIA, B. M.; ESPATH, L. F.; LOPEZ, R. H.; TEMPONE, R. Nesterov-aided stochastic gradient methods using laplace approximation for bayesian design optimization. *arXiv preprint arXiv:1807.00653*, 2018. Cited on page 86.
- 37 TIMOSHENKO, S. P. Lxvi. on the correction for shear of the differential equation for transverse vibrations of prismatic bars. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Taylor & Francis, v. 41, n. 245, p. 744–746, 1921. Cited on page 95.
- 38 SOMERSALO, E.; CHENEY, M.; ISAACSON., D. Existence and uniqueness for electrode models for electric current computed tomography. *SIAM J. Appl. Math.*, v. 52, p. 1023–1040, 1992. Cited on page 102.
- 39 TORII, A. J.; LOPEZ, R. H.; LUERSEN, M. A. A local-restart coupled strategy for simultaneous sizing and geometry truss optimization. *Latin American Journal of Solids and Structures*, SciELO Brasil, v. 8, n. 3, p. 335–349, 2011. Cited on page 114.
- 40 POGU, M.; CURSI, J. S. D. Global optimization by random perturbation of the gradient method with a fixed parameter. *Journal*



of *Global Optimization*, Springer, v. 5, n. 2, p. 159–180, 1994. Cited on page 114.

41 JOHNSON, R.; ZHANG, T. Accelerating stochastic gradient descent using predictive variance reduction. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2013. p. 315–323. Cited on page 114.

42 BECK, J.; DIA, B. M.; ESPATH, L. F.; TEMPONE, R. Multilevel double loop monte carlo and stochastic collocation methods with importance sampling for bayesian optimal experimental design. *arXiv preprint arXiv:1811.11469*, 2018. Cited on page 115.

43 HARDESTY, S. S. *Optimization of shell structure acoustics*. Tese (Doutorado) — Rice University, 2010. Cited on page 129.

44 WANG, C.; REDDY, J. N.; LEE, K. *Shear deformable beams and plates: Relationships with classical solutions*. [S.l.]: Elsevier, 2000. Cited on page 132.



# APPENDIX A – Deduction of MAP and covariance matrix for Laplace methods

Consider the case where the logarithm of the posterior is approximated by a second order Taylor expansion at its MAP. In this case, the resulting approximation of the posterior is a Gaussian-distributed variable with mean  $\hat{\boldsymbol{\theta}}$  and covariance  $\boldsymbol{\Sigma}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}})$ , where  $\hat{\boldsymbol{\theta}}$  is the MAP and  $\boldsymbol{\Sigma}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}})$  is the covariance at the true posterior evaluated at  $\hat{\boldsymbol{\theta}}$ . For the sake of simplicity, we write  $\hat{\boldsymbol{\Sigma}} \stackrel{\text{def}}{=} \boldsymbol{\Sigma}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}})$ . Thus, we write the posterior approximation as

$$\pi(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\xi}) \approx \tilde{\pi}(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\xi}) \stackrel{\text{def}}{=} (2\pi)^{-\frac{\dim(\boldsymbol{\theta})}{2}} \det(\hat{\boldsymbol{\Sigma}})^{-1} \exp\left(-\frac{1}{2} \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(\boldsymbol{\xi}, \mathbf{Y})\|_{\hat{\boldsymbol{\Sigma}}^{-1}}\right) \quad (\text{A.1})$$

To approximate the posterior we need to find  $\hat{\boldsymbol{\theta}}$  and  $\hat{\boldsymbol{\Sigma}}$ . By definition, the MAP is

$$\hat{\boldsymbol{\theta}}(\boldsymbol{\xi}, \mathbf{Y}) \stackrel{\text{def}}{=} \arg \max_{\boldsymbol{\theta} \in \Theta} \pi(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\xi}). \quad (\text{A.2})$$

Since the natural logarithm is a monotonically increasing function defined on the positive real numbers, and the posterior is a strictly positive function, instead of maximizing the posterior, we can maximize its logarithm,

$$\hat{\boldsymbol{\theta}}(\boldsymbol{\xi}, \mathbf{Y}) = \arg \max_{\boldsymbol{\theta} \in \Theta} \log \pi(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\xi}). \quad (\text{A.3})$$

Using the Bayes' theorem presented in Eq. 2.2 on Eq. A.3, and noting that the evidence does not depend on  $\boldsymbol{\theta}$ , thus, can be ignored, furnishes

$$\hat{\boldsymbol{\theta}}(\boldsymbol{\xi}, \mathbf{Y}) = \arg \max_{\boldsymbol{\theta} \in \Theta} \log [p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})\pi(\boldsymbol{\theta})] \quad (\text{A.4})$$

$$= \arg \max_{\boldsymbol{\theta} \in \Theta} [\log p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) + \log \pi(\boldsymbol{\theta})] \quad (\text{A.5})$$

Substituting the likelihood from Eq. 2.3 and noting that the normalizing constant is strictly positive for any positive-definite matrix  $\Sigma_\epsilon$ , thus, can be ignored, results in

$$\hat{\boldsymbol{\theta}}(\boldsymbol{\xi}, \mathbf{Y}) = \arg \max_{\boldsymbol{\theta} \in \Theta} \left[ -\frac{1}{2} \sum_{i=1}^{N_e} \|\mathbf{y}_i(\boldsymbol{\xi}) - \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta})\|_{\Sigma_\epsilon}^2 + \log \pi(\boldsymbol{\theta}) \right], \quad (\text{A.6})$$

or, alternatively,

$$\hat{\boldsymbol{\theta}}(\boldsymbol{\xi}, \mathbf{Y}) = \arg \min_{\boldsymbol{\theta} \in \Theta} \left[ \frac{1}{2} \sum_{i=1}^{N_e} \|\mathbf{y}_i - \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta})\|_{\Sigma_\epsilon}^2 - \log(\pi(\boldsymbol{\theta})) \right]. \quad (\text{A.7})$$

The covariance matrix of the posterior evaluated at its MAP,  $\hat{\Sigma}$ , can be obtained from the second order derivative with respect to  $\boldsymbol{\theta}$  of the log-posterior evaluated at the MAP,

$$\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \log \tilde{\pi}(\hat{\boldsymbol{\theta}}|\mathbf{Y}, \boldsymbol{\xi}) = -\hat{\Sigma}^{-1}. \quad (\text{A.8})$$

From Bayes' rule,

$$\tilde{\pi}(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\xi}) = \frac{p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})\pi(\boldsymbol{\theta})}{p(\mathbf{Y}|\boldsymbol{\xi})} \quad (\text{A.9})$$

$$\log \tilde{\pi}(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\xi}) = \log(p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})) + \log(\pi(\boldsymbol{\theta})) - \log(p(\mathbf{Y}|\boldsymbol{\xi})) \quad (\text{A.10})$$

$$\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \log \tilde{\pi}(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\xi}) = \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \log(p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})) + \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \log(\pi(\boldsymbol{\theta})). \quad (\text{A.11})$$

Substituting the likelihood from Eq. 2.3 and  $\hat{\Sigma}$  from Eq. A.8 results in

$$-\hat{\Sigma}^{-1} = -N_e \nabla_{\boldsymbol{\theta}} \mathbf{g}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}})^T \Sigma_\epsilon^{-1} \nabla_{\boldsymbol{\theta}} \mathbf{g}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}}) - (\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \mathbf{g}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}}))^T \Sigma_\epsilon^{-1} \sum_{i=1}^{N_e} (\mathbf{y}_i - \mathbf{g}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}})) + \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \log(\pi(\hat{\boldsymbol{\theta}})). \quad (\text{A.12})$$

Long et al. (8) prove that ignoring the second term in the right hand side of Eq. A.12 results in an error of  $\mathcal{O}_{\mathbb{P}}(\sqrt{N_e})$ , therefore, we opt to

avoid evaluating the second order derivatives of the model. Hence, the resulting  $\hat{\Sigma}$  is

$$\Sigma^{-1}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}}) = N_e \nabla_{\boldsymbol{\theta}}(\mathbf{g}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}}))^T \Sigma_{\epsilon}^{-1} \nabla_{\boldsymbol{\theta}} \mathbf{g}(\boldsymbol{\xi}, \hat{\boldsymbol{\theta}}) - \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \log(\pi(\hat{\boldsymbol{\theta}})) + \mathcal{O}_{\mathbb{P}}\left(\sqrt{N_e}\right). \quad (\text{A.13})$$



## APPENDIX B – Proof of Eq. 3.8

Throughout the present thesis, we assume that

$$\nabla_{\boldsymbol{\xi}} \mathbb{E}_{\boldsymbol{\theta}, \mathbf{Y}} [f(\boldsymbol{\xi}, \boldsymbol{\theta}, \mathbf{Y})] = \mathbb{E}_{\boldsymbol{\theta}, \mathbf{Y}} [\nabla_{\boldsymbol{\xi}} f(\boldsymbol{\xi}, \boldsymbol{\theta}, \mathbf{Y})]. \quad (\text{B.1})$$

This assumption is consistent with the experiment model at Eq. 2.1. Consider

$$\begin{aligned} \nabla_{\boldsymbol{\xi}} \mathbb{E}_{\boldsymbol{\theta}, \mathbf{Y}} [f(\boldsymbol{\xi}, \boldsymbol{\theta}, \mathbf{Y})] &= \nabla_{\boldsymbol{\xi}} \int_{\Theta} \int_{\mathcal{Y}} \log \left( \frac{p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})}{p(\mathbf{Y}|\boldsymbol{\xi})} \right) p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) d\mathbf{Y} \pi(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \int_{\Theta} \int_{\mathcal{Y}} \nabla_{\boldsymbol{\xi}} \log \left( \frac{p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})}{p(\mathbf{Y}|\boldsymbol{\xi})} \right) p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) d\mathbf{Y} \pi(\boldsymbol{\theta}) d\boldsymbol{\theta} + \\ &\quad + \int_{\Theta} \int_{\mathcal{Y}} \log \left( \frac{p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi})}{p(\mathbf{Y}|\boldsymbol{\xi})} \right) \nabla_{\boldsymbol{\xi}} p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) d\mathbf{Y} \pi(\boldsymbol{\theta}) d\boldsymbol{\theta} \end{aligned} \quad (\text{B.2})$$

Recall that the likelihood is

$$p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) = \det(2\pi \boldsymbol{\Sigma}_{\epsilon})^{-\frac{N_{\epsilon}}{2}} \exp \left( -\frac{1}{2} \sum_{i=1}^{N_{\epsilon}} \|\mathbf{y}_i(\boldsymbol{\xi}) - \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta})\|_{\boldsymbol{\Sigma}_{\epsilon}^{-1}}^2 \right), \quad (\text{B.3})$$

thus,

$$\nabla_{\boldsymbol{\xi}} p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) = p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) \left( -\boldsymbol{\Sigma}_{\epsilon}^{-1} \sum_{i=1}^{N_{\epsilon}} (\nabla_{\boldsymbol{\xi}} \mathbf{y}_i(\boldsymbol{\xi}) - \nabla_{\boldsymbol{\xi}} \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta})) \right). \quad (\text{B.4})$$

For the experiment model  $\mathbf{y}_i = \mathbf{g}(\boldsymbol{\xi}, \boldsymbol{\theta}) + \boldsymbol{\epsilon}_i$ , since  $\mathbf{Y}$  is evaluated using  $\boldsymbol{\theta}$ , it can be observed that  $\nabla_{\boldsymbol{\xi}} \mathbf{y}_i = \nabla_{\boldsymbol{\xi}} \mathbf{g}$ , hence,

$$\nabla_{\boldsymbol{\xi}} p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\xi}) = \mathbf{0}, \quad (\text{B.5})$$

and, consequently, Eq. 3.8 holds.





# APPENDIX C – Timoshenko Beam Model

We derive the governing equations for the Timoshenko beam from 3d linear elasticity, as in (43). Considering a point  $x \in \Omega \subset \mathbb{R}^3$ , we write its position after loading is applied as  $x + u(x)$ , where  $u$  is the displacement vector. The symmetric strain tensor is defined as

$$\epsilon(u)(x) = \frac{1}{2}(\nabla u(x) + (\nabla u(x))^T), \quad (\text{C.1})$$

being linearly related to the corresponding Cauchy stress tensor

$$\sigma(x) = C : e(u)(x). \quad (\text{C.2})$$

The fourth-order tensor  $C$  elements can be deduced from constitutive equations for isotropic homogeneous materials as

$$C^{ijkl} = \frac{E}{2(1+\nu)}(\delta^{ik}\delta^{jl} + \delta^{il}\delta^{jk}) + \frac{E\nu}{(1+\nu)(1-2\nu)}\delta^{ij}\delta^{kl}, \quad (\text{C.3})$$

where  $E$  is the Young modulus, and  $\nu$  is the Poisson ratio.

We formulate the Timoshenko beam as

**Geometry** The domain  $\Omega \subset \mathbb{R}^3$  has one dimension significantly greater than the others on the  $x_1$  dimension. The line on the middle of the beam along the  $x_1$  dimension is described by the domain  $\Omega_o \equiv (0, L_e) \subset \mathbb{R}$ , the width by a function  $b : \Omega_o \mapsto (0, \infty)$  and the height by a function  $H : \Omega_o \mapsto (0, \infty)$ . The domain  $\Omega \subset \mathbb{R}^3$  is defined as

$$\Omega \equiv \{(x_1, x_2, x_3) \in \mathbb{R}^3 : x_1 \in \Omega_o, x_2 \in [-H/2, H/2], x_3 \in [-b/2, b/2]\} \quad (\text{C.4})$$

**Mechanics** Straight lines perpendicular to the line on the middle of the beam remain straight after loading. Transverse normals do not elongate, thus,  $\sigma_{33} = 0$ .

**Kinematics** The transverse normal sections can rotate in relation to the deformed line on the middle of the beam, allowing shear strains to be considered. The displacement of any point  $x \subset \Omega_o$  is given by the vector  $w(x)$ , and the rotation of the cross-section at  $x$  is  $\eta(x)$ . Since we admit that the loading is on the direction of  $w_3$  and that the displacement due to the Poisson ratio on the direction of  $w_2$  is negligible, we ignore the displacement and strain on this direction. We consider the displacement field for the beam to be

$$u(x_1, x_2) = (w_1(x_1) - x_2\zeta(x_1), w_2(x_1), 0) \quad (\text{C.5})$$

We want to find the pair  $u$  and  $\zeta$  on the domain  $\Omega$  that satisfies the conditions stated above as besides boundary conditions, cinematic conditions — relating displacement field  $u$  and strains  $\varepsilon$  — and compatibility equations between strains  $\varepsilon$  and stresses  $\sigma$ .

Modeling the beam in such a way that the rotation of the cross section  $\zeta$  can be different than  $\frac{dw_2}{dx}$  leads to shear strains on the sections, as required for our problem. Since we are ignoring the 3rd dimension, we can simplify our process by considering a 2D problem. Using the balance of linear momentum equation, we define the strong form for our problem as

$$-\text{div } \sigma = 0 \quad \text{in } \Omega \quad (\text{C.6a})$$

$$\sigma = C : \varepsilon(u) \quad \text{in } \Omega \quad (\text{C.6b})$$

$$u = 0 \quad \text{in } \Gamma_o \quad (\text{C.6c})$$

$$\sigma \cdot n = t \quad \text{in } \Gamma_1 \quad (\text{C.6d})$$

$$\sigma \cdot n = 0 \quad \text{in } \Gamma_F \quad (\text{C.6e})$$

where  $\Gamma_o$  is the region where the supports are,  $\Gamma_1$  is where traction  $q$  are applied, and  $\Gamma_F$  is the remaining part of the boundary, the free part of the boundary. To find a weak form, we introduce the test function  $v$  as

$$v(x_1, x_2) = (y_1(x_1) - x_2\eta(x_1), y_2(x_1), 0), \quad (\text{C.7})$$

and integrate the Cauchy equation in Eq. C.6d on  $\Omega$

$$W_{ext} = \int_{\Gamma_1} t v \, d\Gamma_1 = \int_{\Gamma_1} (\sigma \cdot n) v \, d\Gamma_1. \quad (\text{C.8})$$

Using the divergence theorem,

$$\int_{\Gamma_1} t v \, d\Gamma_1 = \int_{\Omega} \operatorname{div}(\sigma \cdot v) \, d\Omega, \quad (\text{C.9})$$

and noting that  $v$  is a null vector on the boundaries, furnishes

$$\int_{\Gamma_1} t v \, d\Gamma_1 = \int_{\Omega} \operatorname{div}(\sigma) \cdot v \, d\Omega + \int_{\Omega} \sigma : \nabla v \, d\Omega. \quad (\text{C.10})$$

Since  $\sigma$  is symmetrical, its double contraction with respect to the skew part of  $\nabla v$  is null, thus, we substitute  $\nabla v$  by its symmetrical part,  $\nabla v_{sym} = \frac{1}{2}(\nabla v + \nabla v^T) = \varepsilon(v)$ . We are not considering body forces, hence,  $\operatorname{div}(\sigma) = 0$ , resulting in

$$\int_{\Gamma_1} t v \, d\Gamma_1 = \int_{\Omega} \sigma : \varepsilon(v) \, d\Omega. \quad (\text{C.11})$$

Substituting the constitutive Eq. C.6b furnishes the weak form of the governing equations

$$\int_{\Gamma_1} t v \, d\Gamma_1 = \int_{\Omega} \varepsilon(u) : C : \varepsilon(v) \, d\Omega. \quad (\text{C.12})$$

From elasticity theory, we have

$$\begin{aligned} \varepsilon_{11}(u) &= \frac{\partial u_1}{\partial x_1} \\ &= x_2 \frac{d\zeta}{dx_1} \end{aligned} \quad (\text{C.13})$$

$$\gamma_{12}(u) = \varepsilon_{21}(u) + \varepsilon_{12}(u)$$

$$\gamma_{12}(u) = \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}$$

$$\gamma_{12}(u) = \frac{dw_2}{dx_1} - \zeta,$$

and, similarly,

$$\begin{aligned}\varepsilon_{11}(v) &= x_2 \frac{d\eta}{dx_1} \\ \gamma_{12}(v) &= \frac{dy_2}{dx_1} - \eta.\end{aligned}\tag{C.14}$$

Substituting the strains in Eq. C.13 and Eq. C.14 on Eq. C.12, and considering  $t = (0, q, 0)$ , a system of two equations arise

$$\begin{cases} \int_{\Omega} \left( \frac{dw_2}{dx_1} - \varsigma \right) G \frac{dy_2}{dx_1} d\Omega = \int_{\Omega} q y_2 d\Omega_o \\ \int_{\Omega} x_2 \frac{d\varsigma}{dx_1} E x_2 \frac{d\eta}{dx_1} d\Omega - \int_{\Omega} \left( \frac{dw_2}{dx_1} - \varsigma \right) G \eta d\Omega = 0,\end{cases}\tag{C.15}$$

where  $G$  is the shear modulus and is related to  $E$  and  $\mu$  as  $G = E(1 + \mu)/2$ . Integrating Eq. C.15 over the area  $A_r = H \times b$ , and noting that  $I_n = \int_{A_r} x_2^2 dA_r$  is the inertia moment, results in

$$\begin{cases} GA_r K_s \int_{\Omega_o} \left( \frac{dw_2}{dx_1} - \varsigma \right) \frac{dy_2}{dx_1} d\Omega_o = \int_{\Omega_o} q_o y_2 d\Omega_o \\ EI_n \int_{\Omega_o} \frac{d\varsigma}{dx_1} \frac{d\eta}{dx_1} d\Omega_o - GA_r K_s \int_{\Omega_o} \left( \frac{dw_2}{dx_1} - \varsigma \right) \eta d\Omega_o = 0.\end{cases}\tag{C.16}$$

The constant  $K_s$  is included on the formulation to correct the error on  $G A_r$  induced on the model by assuming that the shear stress is uniform on the cross section and is adopted as  $K_s = 5/6$  for rectangular sections. A discussion on the definition of this value can be found on the work of Wang and Reddy (44).

Summing both equations in Eq. C.16 results in

$$EI_n \int_{\Omega_o} \frac{d\varsigma}{dx_1} \frac{d\eta}{dx_1} d\Omega_o + GA_r K_s \int_{\Omega_o} \left( \frac{dw_2}{dx_1} - \varsigma \right) \left( \frac{dy_2}{dx_1} - \eta \right) d\Omega_o = \int_{\Omega_o} q_o y_2 d\Omega_o\tag{C.17}$$

where the first integral is the energy related to the strains due to the bending of the beam, the second to the shear strains and the third to the vertical uniform load, considered to be applied over  $\Omega_o$ .

Integrating the equations C.16 by parts and noting that the trial functions vanish on the boundaries vanishes

$$\begin{cases} -GA_r K_s \int_{\Omega_o} \frac{d}{dx_1} \left( \frac{dw_2}{dx_1} - \varsigma \right) y_2 d\Omega_o & = \int_{\Omega_o} q_o y_2 d\Omega_o \\ -EI_n \int_{\Omega_o} \frac{d^2 \varsigma}{dx_1^2} \eta d\Omega_o - GA_r K_s \int_{\Omega_o} \left( \frac{dw_2}{dx_1} - \varsigma \right) \eta d\Omega_o & = 0, \end{cases} \quad (\text{C.18})$$

thus,

$$\begin{cases} -GA_r K_s \frac{d}{dx_1} \left( \frac{dw_2}{dx_1} - \varsigma \right) & = q_o \\ -EI_n \frac{d^2 \varsigma}{dx_1^2} - GA_r K_s \left( \frac{dw_2}{dx_1} - \varsigma \right) & = 0. \end{cases} \quad (\text{C.19})$$

Considering the beam as simply supported with length  $L_e$ , we obtain the boundary conditions

$$\begin{cases} EI_n \frac{d\varsigma}{dx_1} \Big|_{x_1=0} & = EI_n \frac{d\varsigma}{dx_1} \Big|_{x_1=L_e} = 0 \\ K_s GA_r \left( \frac{dw_2}{dx_1} - \varsigma \right) \Big|_{x_1=0} & = K_s GA_r \left( \frac{dw_2}{dx_1} - \varsigma \right) \Big|_{x_1=L_e} = \frac{q_o L_e}{2}, \end{cases} \quad (\text{C.20})$$

and applying these conditions on equations C.19 furnishes

$$\begin{aligned} g(\xi, \theta) &= (\varepsilon_{11}(\xi, \theta), \varepsilon_{12}(\xi, \theta)) \\ &= \left( \frac{x_2(\xi) (qL_e x_1(\xi) - qx_1^2(\xi))}{2E(\theta)I_n}, \frac{L_e q - qx_1(\xi)}{K_s G(\theta)A_r} \right). \end{aligned}$$