Fall 2021

Multi-Fidelity Surrogate and Reduced-Order Model-Based Microfluidic Concentration Gradient Generator Design

Haizhou Yang

MULTI-FIDELITY SURROGATE AND REDUCED-ORDER MODEL-BASED MICROFLUIDIC CONCENTRATION GRADIENT GENERATOR DESIGN

by

Haizhou Yang

Bachelor of Science
Dalian University of Technology, 2015

Master of Science
Dalian University of Technology, 2018

Submitted in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy in

Mechanical Engineering

College of Engineering and Computing

University of South Carolina

2022

Accepted by:

Yi Wang, Major Professor

Yu Qian, Committee Member

Andrew Gross, Committee Member

Xiaomin Deng, Committee Member

Tracey L. Weldon, Interim Vice Provost and Dean of the Graduate School

DEDICATION

To my parents Wenwu Yang and Jinyan Wang, and my wife Jiaxuan Ma

for their love and support.

ACKNOWLEDGEMENTS

There are many people I would like to thank for their help, both in my research and personal life during my Ph.D. study. First, I would like to thank my advisor, Professor Yi Wang. Professor Wang provided comprehensive and detailed guidance on my research, and let me know what is research and how to do research. I learned a lot from his knowledge, skills, insight, patience, and enthusiasm for research. Besides, Professor Wang also shared his extensive experiences in his life and gave me suggestions for my future career. I would also like to thank all the committee members, including my co-advisor Professor Yu Qian, Professor Andrew Gross, and Professor Xiaomin Deng for their great guidance and support on my research. Their valuable suggestions and feedback help me complete this dissertation with improved quality.

I would like to thank Dr. Seong Hyeon Hong, Dr. Jung IL Shu and all group members from iMSEL. Dr. Hong provided tremendous useful guidance and suggestions for my research. Dr. Shu shared his knowledge on CFD with me and also provided help in my personal life. I would like to thank all group members who gave me suggestions during our group meeting, which are helpful to improve my research. Furthermore, I would like to thank my friends Yulin Huang, Dr. Feng Guo, Junlin Ou, Dr. Guanghan Huang, Dr. Wenming Li, Kai Luo, Dr. Chenfei Zhang, Dr. Congcong Ren, and all my friends at the University of South Carolina and in China. Life becomes more colorful because of them.

I would like to appreciate my school, University of South Carolina, provided great utilities for my research. The sponsorships from Samsung, National Aeronautics and Space

Administration (NASA), and Federal Railroad Administration (FRA) are greatly appreciated for their significant financial support.

This dissertation is dedicated to my parents, my wife, and all my family members. My parents, Wenwu Yang and Jinyan Wang, always stand behind me and give me encouragement. My wife Jiaxuan Ma, takes care of me as well as gives me boundless love. My cat, Jian, accompanies me and her loveliness is a relief for the pressure. Besides, all my family members expressed their concern and support during my Ph.D. study, especially during the pandemic. For me, they ALL mean POWER.

ABSTRACT

The microfluidic concentration gradient generator (μCGG) is an important device to generate and maintain concentration gradients (CGs) of biomolecules for understanding and controlling biological processes. However, determining the optimal operating parameters of μCGG is still a significant challenge, especially for complex CGs in cascaded networks. To tackle such a challenge, this study presents multi-fidelity surrogate- and reduced-order model-based optimization methodologies for accurate and computationally efficient design of μCGGs.

The surrogate-based optimization (SBO) method is first proposed for the design optimization of μCGGs based on an efficient physics-based component model (PBCM). Various combinations of regression and correlation functions in Kriging and different adaptive sampling (infill) techniques are examined to establish the design process with refined model structures. In order to combine the simulation data from different sources with varying fidelities and computational costs for improved design efficiency and accuracy, a novel multi-fidelity surrogate-based optimization (MFSBO) method is presented. For the first time, a new computation-aware adaptive sampling strategy based on expected improvement reduction (EIR) is proposed to accelerate the convergence of MFSBO. EIR-based infill determines the data source and infill location by hypothetically interrogating the effect of samples and simulation fidelities on the reduction of the expected improvement. It also enables low-fidelity batch infills within a dynamically varying trust region to improve exploration on the fly. Subsequently, a new data sparsification technique

based on the reduced design space and data filtering (RDS&DF) is investigated to eliminate redundant data and reduce the modeling time for improved optimization efficiency, hence addressing the long-standing “big data” issues associated with MFSBO. RDS&DF is also combined with EIR-based infill technique, enabling both parsimony and computational awareness for MFSBO. Finally, a multi-fidelity reduced-order modeling (MFROM) method is developed to enable model reusability and completely replace the CFD simulation when different μCGGs need to be designed. The key innovation of MFROM is using the proper orthogonal decomposition to obtain the low-dimensional representation of the high-fidelity CFD data and the low-fidelity PBCM data and a kriging model to bridge the fidelity gap between them in the modal subspace, yielding compact MFROM applicable within the broad trade space. As a result, MFROM is highly compatible with GPU-enabled optimization by utilizing its massively parallelized computing threads. The excellent agreement between the designed CGs and the prescribed CGs demonstrates the unprecedented accuracy and efficiency of the proposed multi-fidelity modeling and optimization methodologies.

In conclusion, given their non-intrusive, data-driven natures, both (MF)SBO and MFROM are versatile and can serve as a new paradigm for μCGG design.

TABLE OF CONTENTS

Dedication iii

Acknowledgements iv

Abstract vi

List of Tables xi

List of Figures xiii

Chapter 1 Introduction 1

1.1 Background 2

1.2 Motivation 5

1.3 Scope 6

1.4 Dissertation Organization 10

Chapter 2 Research Fundamentals 12

2.1 Microfluidic Concentration Gradient Generators and Data Sources 13

2.2 Surrogate and Reduced-order Models 20

2.3 Adaptive Sampling and Infill Strategies 26

2.4 Genetic Algorithm 29

Chapter 3 Surrogate-based Optimization for Microfluidic Concentration Gradient Generator Design 30

3.1 Methodology 31

3.2 Problem Formulation and Case Studies 34

3.3 Results and Discussion 38

3.4 Summary 56

Chapter 4 Multi-fidelity Surrogate-based Optimization for Microfluidic Concentration Gradient Generator Design 58

4.1 Methodology 61

4.2 Problem Formulation and Performance Verification 63

4.3 Results and Discussion 66

4.4 Summary 76

Chapter 5 A Sequential Multi-Fidelity Surrogate-based Optimization Methodology based on Expected Improvement Reduction 78

5.1 Proposed Methodology 80

5.2 Results and Discussion 87

5.3 Summary 104

Chapter 6 A Sparse Multi-fidelity Surrogate-based Optimization method with Computational Awareness 106

6.1 Problem Formulation 110

6.2 Proposed Methodology 111

6.3 Case Studies 118

6.4 Summary 131

Chapter 7 Multi-Fidelity Reduced-order Model for GPU-Enabled Microfluidic Concentration Gradient Design 133

7.1 Problem Description 136

7.2 Proposed Methodology 137

7.3 Results and Discussion 144

7.4 Summary 160

Chapter 8 Conclusion and Future Work 163

8.1 Conclusion 164

8.2 Future Work 167

References 169

LIST OF TABLES

Table 3.1 Relative percentage errors of surrogate models built by different combinations of regression and correlation models in the case study: design of inlet concentrations. 42

Table 3.2 Comparison of different infill strategies and prescribed CGs in terms of J_d of optimum design for the case study: design of inlet concentrations. 43

Table 3.3 Comparison of the number of PBCM evaluation/simulation between SBO with adaptive sampling and gradient-based optimization for the case study: design of Inlet Concentrations. 48

Table 3.4 Relative percentage errors of surrogate models built by different combinations of regression and correlation models in the case study: design of inlet concentrations and pressure differences. 49

Table 3.5 Comparison of different infill strategies and prescribed CGs in terms of J_d of optimum design for the case study: design of inlet concentrations and pressure differences. 50

Table 3.6 J_ds of CGs predicted by PBCM and CFD using the optimal designs found by SBO with adaptive sampling. 53

Table 3.7 Evaluation comparisons between the SBO with adaptive sampling and the gradient-based optimization methods. 55

Table 4.1 J_d of the design corresponding to the minimum of the surrogate model at the 20^{th} iteration and the optimal design for the three prescribed CG. 68

Table 4.2 The number and cost of simulation runs for the three optimization methods. 69

Table 4.3 J_d of the design corresponding to the minimum of the surrogate model at 20^{th} iteration and the optimal design for the three prescribed CG. 72

Table 4.4 The number and cost of simulation runs for three optimization methods. 73

Table 4.5 J_d of the design corresponding to the minimum of the surrogate model at 100th iteration and the optimal design for the three prescribed CG. 74

Table 4.6 The number and cost of simulation runs for three optimization methods. 76

Table 5.1 The minimum of MFSM and corresponding objective values at the 10th and 20th iteration. 92

Table 5.2 The minimum of MFSM and corresponding objective at 10th iteration and 20th iteration. 97

Table 5.3 Pressures at inlet reservoirs for three prescribed CGs. 99

Table 5.4 J_d at the minimum of the MFSM at the 15th and 30th iteration for the three prescribed CG. 102

Table 6.1 The computational time of the optimization and the minimum of CoKriging found by the optimization. 122

Table 6.2 The computational time of MFSBO, min. J_d corresponding to the found minimum, and the ratio of reduction in computation time. 129

Table 7.1. NRMSE of the 4 selected testing samples. 150

Table 7.2 Mean and standard deviation of design parameters and optimization time. 152

Table 7.3 NRMSE of predicted CGs. 153

Table 7.4 NRMSE of the four selected testing samples. 157

Table 7.5 Mean and standard deviation of design parameters and optimization time. 159

Table 7.6 NRMSE of predicted CGs. 159

LIST OF FIGURES

Figure 2.1 Schematic of the Triple-Y μCGG. 13

Figure 2.2 Component model of the Triple-Y μCGG. 17

Figure 2.3 Example results and comparison between PBCM and CFD model given the same simulation parameters. 19

Figure 3.1 Flowchart of the SBO with adaptive sampling for μCGG. 34

Figure 3.2 Illustration of the backflow issue and reformulation of the design problem to use the pressure difference as the design variables rather than the inlet pressure for the Triple-Y μCGG. 36

Figure 3.3 Prescribed CGs in the first case study: design of inlet concentrations. 37

Figure 3.4 Prescribed CGs in the second case study: design of both inlet concentrations and pressure differences. 38

Figure 3.5 The procedure for design verification and performance benchmarking. 41

Figure 3.6 Convergence of Min. J_d using different infill strategies for prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) linear in the case study: design of inlet concentrations. 43

Figure 3.7 Response surface plots of the surrogate models in 3D that vary with c_1 and c_2 while keeping the other design variables constant for different numbers of sample infills: (1) 0, (2) 15, and (3) 30 in the case study: design of inlet concentrations. 44

Figure 3.8 CFD contour plots and predicted CGs relative to the prescribed CG for the case study: design of inlet concentrations. 46

Figure 3.9 Comparison of results between SBO with random sampling and adaptive sampling for the case study: design of Inlet Concentrations. 47

Figure 3.10 Convergence of Min. J_d of surrogate model using different infill strategies for prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped for the case study: design of inlet concentrations and pressure differences. 50

Figure 3.11 Response surface plots of the surrogate models in 3D that vary with c_1 and c_2 while keeping the other design variables constant for different numbers of sample infills: (1) 0, (2) 300, and (3) 700 in the case study: design of inlet concentrations and pressure differences. 51

Figure 3.12 CFD contour plots and predicted CGs relative to the prescribed CG for the case study: design of inlet concentrations and pressure differences. 53

Figure 3.13 Comparison between PBCM and CFD simulation at fully developed region for the sawtooth-shaped CG. 53

Figure 3.14 Results of SBO with random sampling for the design of inlet concentration and pressure difference. 54

Figure 4.1 Flowchart of the MFSBO for µCGG. 61

Figure 4.2 Prescribed CGs including asymmetric constituent components of CGs. 64

Figure 4.3 The procedure for design verification and performance benchmarking. 65

Figure 4.4 Convergence of Min. J_d using different optimization methods for prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped. 67

Figure 4.5 Comparison of predicted CGs to the prescribed CG: (1) the design corresponding to the minimum of the surrogate model at 20^{th} iteration and (2) optimal design at the end of the optimization, and (3) the CFD contour plots at optimal designs attained using MFSBO-4 for three prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped. 67

Figure 4.6 Convergence of Min. J_d using different optimization methods for prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped. 70

Figure 4.7 Comparison of predicted CGs to the prescribed CG: (1) the design corresponding to the minimum of the surrogate model at 20th iteration and (2) optimal design at the end of the optimization, and (3) the CFD contour plots at optimal designs using MFSBO-4 for three prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped. 71

Figure 4.8 Convergence of Min. J_d using different optimization methods for prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped. 74

Figure 4.9 Comparison of predicted CGs to the prescribed CG: (1) the design corresponding to the minimum of the surrogate model at 100th iteration and (2) optimal design at the end of the optimization, and (3) the CFD contour plots at optimal designs attained using MFSBO-4 for three prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped. 75

Figure 5.1 Flowchart of MFSBO with EIR-based Infill. 83

Figure 5.2 2D example flowchart of EIR-based infill. 84

Figure 5.3 Response surfaces of HF and LF models for the Brainin function. 88

Figure 5.4 Convergence of MFSBO to the global minimum using different infill strategies. 89

Figure 5.5 Response surface plots of the MFSM for the four infill strategies: (1) H infill, (2) alternating Infill, (3) random Infill, and (4) EIR-based Infill. 90

Figure 5.6 Selection of data source (fidelity level) at each iteration for the EIR-based Infill for the Brainin function. 93

Figure 5.7 Response surfaces of the HF and LF model for the Hartmann 4 function. 94

Figure 5.8 Convergence of the minimum of MFSM using different infill strategies. 94

Figure 5.9 Response surface plots of MFSM for the four infill strategies: (1) H infill, (2) alternating Infill, (3) random Infill, and (4) EIR-based Infill. 96

Figure 5.10 Selection of data source (fidelity level) at each iteration for the EIR-based Infill for the Hartmann 4-dimensional Function. 98

Figure 5.11 Prescribed CGs. 99

Figure 5.12 Convergence of Min. J_d using different infill strategies for prescribed CGs: (a) trapezoidal, (b) valley-shaped, and (c) sawtooth-shaped. 100

Figure 5.13 Predicted CGs relative to the prescribed CG based on (1) the minimum of MFSM at the 15th and (2) at the 30th iteration, and (3) the CFD contour plots at the 30th iteration for the three prescribed CGs: (a) trapezoidal, (b) valley-shaped, and (c) sawtooth-shaped. 101

Figure 5.14 Data source selection at each infill iteration for EIR-based Infill. 104

Figure 6.1 Flowchart of MFSBO including both EIR-based infill and RDS&DF. 113

Figure 6.2 Flowchart of reduced design space and data filtering. 116

Figure 6.3 Convergence of the minimum of CoKriging using EIR with and without RDS&DF. 119

Figure 6.4 Selection of data source (fidelity level) at each iteration for the EIR-based infill with and without RDS&DF for the POWELL function. 120

Figure 6.5 (a) RDS application, (b) DF application, (c) volume ratio ξ of RDS, and (d) number of HF and LF data at each iteration. 121

Figure 6.6 Prescribed CGs: (a) trapezoida; (b) vally-shaped; and (c) sawtooth-shaped. 124

Figure 6.7 Convergence of the minimum of CoKriging with and without RDS&DF. 125

Figure 6.8 Selection of data source (fidelity level) at each iteration for the EIR-based Infill with and without RDS&DF for μCGG design. 126

Figure 6.9 (1) RDS application, (2) DF application, (3) volume ratio ξ of RDS, and (4) number of HF and LF data at each iteration for three prescribed CGs: (a) trapezoidal, (b) valley-shaped, and (c) sawtooth-shaped. 128

Figure 6.10 Predicted CGs relative to the prescribed CG based on (1) the minimum of CoKriging at the 75th and (2) at the

150^{th} iteration, and (3) the CFD contour plots at the 150^{th} iteration for the three prescribed CGs: (a) trapezoidal, (b) valley-shaped, and (c) sawtooth-shaped. 129

Figure 7.1 Flowchart of offline MFROM construction for μCGG design. 139

Figure 7.2 Flowchart of offline optimization of the MFROM for μCGG design. 141

Figure 7.3 GPU-enabled optimization flowchart. 144

Figure 7.4 Average NRMSE of 20 testing inputs. 147

Figure 7.5 Singular values comparison between PBCM and CFD snapshots. 147

Figure 7.6 POD basis vectors/modes comparison between PBCM and CFD snapshots. 149

Figure 7.7 CGs comparison for four selected testing samples 150

Figure 7.8 MFROM predicted CGs relative to the true prescribed CGs. 153

Figure 7.9 Average NRMSE of 20 testing inputs. 154

Figure 7.10 Singular values comparison between PBCM and CFD snapshots. 155

Figure 7.11 POD basis vectors/modes comparison between PBCM and CFD snapshots. 156

Figure 7.12 CGs comparison for four selected testing samples 157

Figure 7.13 MFROM predicted CGs relative to the true prescribed CGs. 160

CHAPTER 1 INTRODUCTION

1.1 Background

Formation of complex concentration gradients (CGs) of biomolecules, which regulate the signaling pathway between cells, plays an important role in biological and chemical processes, such as immune response, wound healing, embryogenesis, cancer metastasis, drug screening, and others (Irimia, Geba, and Toner 2006; Cabaleiro 2020; C. G. Yang et al. 2011; X. Wang, Liu, and Pang 2017). To achieve a better understanding of the influence of CGs on biomolecules, various devices and methods have been developed to generate CGs within an in vitro environment. Early work focused on establishing the platforms, such as the Boyden chamber, Dunn slide chamber, Zigmond chamber, agarose dish and the micro-aspirator to create the CG for enhanced understanding of cell response through a signaling pathway (Toh et al. 2014). However, these early platforms, typically with the characteristic lengths of a few millimeters to centimeters, were difficult to realize the length scale of actual biological cells, which range from 1 to 100 μm (Sadava, Hillis, and Heller 2009). Therefore, the microfluidic concentration gradient generators (μCGG) were proposed, which operate on nanolitres or microlitres volumes scale of reagents and provide higher gradient resolutions to generate and maintain concentration gradients, such as linear, parabolic, exponential, sawtooth, and hybrid profiles (Dertinger et al. 2001; Tang et al. 2018; Jeon et al. 2019). In contrast to their counterparts at the macroscale, the μCGG features several unique merits, including short transportation time, fast analysis speed, simple operation, precise manipulation of locations and quantities of biomolecule delivery, and excellent physiological capability to cellular assays at spatiotemporal scales (C. G. Yang et al. 2011; X. Wang, Liu, and Pang 2017; B. Hong et al. 2016; Mulholland et al.

2018). Therefore, a variety of μCGGs are designed, microfabricated, and demonstrated in the field of cell biology and biochemistry, including tree-shaped, altered tree-shaped, Y-shaped, pressure-balanced, incomplete mixing-based, and membrane μCGGs for various CGs (X. Wang, Liu, and Pang 2017; Höving, Janasek, and Novo 2018).

In general, methods for generating CGs can be classified into two categories: static and dynamic (C. G. Yang et al. 2011). The static method forms continuous CGs by molecular diffusion in static liquid. No external force is applied to accelerate the diffusion process, and thus, the diversity of CGs obtained by this method is usually limited. On the other hand, the dynamic method generates CGs by taking advantage of convection in laminar flow driven by external forces. Therefore, the CGs generated by the dynamic method are diverse and have large-range gradients, and hence, widely employed. Besides, μCGGs also can be classified into two categories according to the carrier medium for CG generation: mono-phase and droplet-based (X. Wang, Liu, and Pang 2017). The mono-phase method can be further divided into two groups by distinguishing the presence of shearing. The μCGGs in the mono-phase with shearing form CGs based on the fluid splitting and combining at the junction, and mixing in channels, and include the tree-shaped, altered tree-shaped, Y-shaped μCGGs (Höving, Janasek, and Novo 2018; Y. Wang, Mukherjee, and Lin 2006; Cabaleiro 2020). The tree-shaped network is one of the earliest μCGG designs and widely used due to its simple geometry, which successively splits, mixes, and recombines biologically relevant chemical solution to form digitalized CGs across channel widths (Gorman and Wikswo 2008; Rismanian, Saidi, and Kashaninejad 2019). In order to generate more complex CGs with higher resolutions, the number of stages of tree-shaped μCGG needs to be increased, which however may be more prone to

clogging or leakage (Y. Wang, Mukherjee, and Lin 2006; X. Wang, Liu, and Pang 2017). Therefore, an altered tree-shaped device was developed, which is able to reduce the number of stages of the tree-shaped network and simplify the structure by delicately designed splitting-and-combining patterns (Hattori, Sugiura, and Kanamori 2009). Moreover, a Y-shaped generator is designed to simplify the structure compared to conventional and altered tree-shaped networks by reducing the mixing channel length (Höving, Janasek, and Novo 2018). It can generate the monotonous CG by supplying the fluid with different chemical concentrations at two inlets. In contrast to these complete mixing-based μCGGs, μCGGs utilizing partial mixing were also proposed by our coauthor that manipulates species transport within microchannels and juxtaposes constituent CGs to form complex ones, leading to simple network topology and salient device reliability (Y. Wang, Mukherjee, and Lin 2006; Y. Zhou et al. 2009). Typically, fluid flows fast in these devices, and the CGs are generated within a short period of time. Thus, the shearing effect in the fluid can be significant. To mitigate the issue, membrane systems is developed, that utilizes the unique property of the membrane for flow and species transport, which only allows specific molecules to diffuse through, to form the CG in multiple parallel channels (Zhang et al. 2015). It effectively reduces the flow rate of fluid, increase the residence time, and thus reduce the shearing effect and yield shear free CGs (Zhang et al. 2015). On the other hand, droplet-based methods have been extensively studied and applied recently. Droplet generation, coalescence, and mixing are major techniques to generate CGs for this method (Seemann et al. 2011). Protected by the interface, the droplet is isolated, and the enclosed internal environment meets the requirement of chemical and biochemical processes.

Moreover, the droplet-based μCGG can handle small volumes of sample and perform quantitative analysis using the generated CGs.

1.2 Motivation

Research efforts above mostly focused on demonstrating μCGGs that were fabricated with known operating parameters, such as inlet concentrations and pressures/flow rates. Nonetheless, determining these design parameters is challenging, a trial-and-error process entailing iterative modeling, simulation, and experiments under the guidance of prior experiences, especially when complex CGs are desired. Wang et al. (Y. Wang, Mukherjee, and Lin 2006) proposed a physics-based component model (PBCM) for partial mixing-based μCGGs, which runs orders of magnitude faster than high-fidelity (HF) computational fluid dynamics (CFD) simulations, although less accurate, especially when asymmetric flows from multiple branches are combined (H. Yang et al. 2020). Due to the computational efficiency of the PBCM, multiple simulations were performed iteratively within the design space to find the operational parameters that can generate CGs matching the prescribed ones(Y. Zhou et al. 2009). However, this is a trial-and-error process that needs to be performed manually and requires a large number of simulations, leading to a long design cycle, even though the simulation itself is computationally efficient. Such a challenge can be resolved by an automated design optimization approach that can combine computationally efficient models and optimization algorithms within a single framework. Friedrich et al. (Friedrich, Please, and Melvin 2012) generated the prescribed CG using a customed μCGG, which consists of a single microfluidic channel and oblique-angled grooves. The angle was determined by an optimization loop wrapping around CFD

simulations, and therefore, the desired CGs were generated by designing grooves with different angles. Nonetheless, optimization of CFD simulation requires demanding resources, and can be computationally prohibitive for a resource-limited environment. An efficient μCGGs design automation method based on physics-based models and simulation to rapidly determine operating parameters that accurately generate prescribed CGs is indeed scarce and strongly needed.

1.3 Scope

Therefore, in this context, we developed design optimization algorithms using surrogate and reduced-order models to address such a challenge, where optimization is undertaken on the model and searches within the design space for optimal parameters that can generate CGs matching the prescribed ones. Surrogate models, also known as response surface models and metamodels are used to approximate the behavior of physics-based models through direct mapping between input-output data pairs produced by the latter, and is more computationally efficient to evaluate. Therefore, they are widely used to minimize the number of evaluations by physics-based computer simulation, such as the CFD or the computational structural dynamics (CSD) (I. Couckuyt et al. 2010; Singh et al. 2016) for accelerated optimization and design process. It is well known that high-fidelity, physics-based simulation can be computationally prohibitive for optimization in high-dimensional design parameter space (Forrester and Keane 2009; Bhosekar and Ierapetritou 2018). The surrogate model, constructed by a small number of selected physics-based simulations, and expressed in a compact form comprised of elementary functions. Therefore, the surrogate model is extremely fast-to-evaluate and enables a cost-effective and rapid exploration of

the design space, which is often used to approximate and replace the high-fidelity, physics-based model in computationally intensive processes, such as optimization for enhanced efficiency and resource utilization (Forrester, Sóbester, and Keane 2007; H. Yang et al. 2020; Park, Haftka, and Kim 2017; Z.-H. Han and Zhang 2012; Haftka, Villanueva, and Chaudhuri 2016).

The multi-fidelity surrogate model (MFSM) is one special type of surrogate models, and is built by making use of the data from multiple sources with different fidelities and the correlations between them (Marques et al. 2019; Park, Haftka, and Kim 2017; Fernández-Godino et al. 2016). Because the low-fidelity (LF) model is cheap to evaluate, the global trend of the response surface can be quickly explored and estimated by running a large number of LF simulations. On the other hand, HF simulations are accurate but computationally expensive. As a result, their availability is limited, and they are more appropriate for further accuracy enhancement of the global trend captured by the LF model (Kuya et al. 2011; B. Liu, Koziel, and Zhang 2016). Therefore, LF and HF simulation data play complementary roles in MFSM, and proper management of both could contribute to significant improvement in the computational performance of the analysis. MFSM methods can be classified into several categories (Park, Haftka, and Kim 2017; Fernández-Godino et al. 2016; Peherstorfer, Willcox, and Gunzburger 2018), including adaption, fusion, and filtering. The adaption method enhances the LF model with the information from the HF model through different correction strategies, such as additive, multiplicative, and comprehensive (Fernández-Godino et al. 2016). The fusion method combines the information from HF and LF data, and formulate one MFSM for prediction. Co-Kriging is an example of fusion methods and has been widely used in various engineering fields. The

filtering method performs HF evaluations only when the LF model becomes inaccurate. In other words, the LF model serves as a filter to select candidate points to perform HF evaluations.

Proper orthogonal decomposition (POD), also known as principal component analysis or Karhunen-Loeve expansion, is a powerful method of low-rank matrix approximation to reduce the dimension of complex data for the reduced-order model (ROM) (Bai and Wang 2021; Robertson et al. 2018). It extracts L2 optimal orthonormal basis vectors spanning the subspace of the data, and reveals the hidden underlying structure of the data (Guénot et al. 2013; Kato and Funazaki 2014; Gutierrez-Castillo and Thomases 2019; Quesada, Villon, and Salsac 2021). Therefore, POD is often used to construct the ROM of low dimensions that represents the solution in the modal subspace spanned by the extracted orthogonal basis.

An adaptive sampling and infill strategy is utilized to determine new sample points at the most important but under-explored regions for the next round of physics-based simulation to progressively improve surrogate model accuracy, especially near the region of the global optimum by analyzing its underlying response surface. The infill is undertaken with respect to a criterion that balances between exploitation and exploration (Forrester, Sóbester, and Keane 2008). Examples of traditional infill strategies for surrogate-based optimization (SBO) include minimization of the predicted objective function (MP), statistical lower-bound (LB), probability of improvement (PI), expected improvement (EI), and others (J. Liu, Han, and Song 2012). Recently, several novel infill strategies are proposed for both the global multi-fidelity surrogate modeling that aims to accurately capture the response surface within the entire parameter space through MFSM,

and the multi-fidelity surrogate-based optimization (MFSBO). For example, Zhou et al. proposed a sequential multi-fidelity metamodeling approach to determine the location of the sample point that reduces the maximum mean squared error (MSE) of the predictor within the parameter space (Q. Zhou et al. 2017). Huang et al. proposed an extension of the sequential kriging optimization method for global prediction (Huang et al. 2006). The location and the fidelity level (or source) of the next evaluation are selected by maximizing an augmented expected EI function. Chen et al. proposed a new fusion-based sequential optimization approach, which identifies the sample location based on the EI criterion and which model to evaluate at the chosen sample location by incorporating one additional hypothetical data into the initial data set and comparing the reduction in predicted uncertainty for each model with different fidelities (S. Chen et al. 2017). Kaya et al. presented a sample refinement strategy that adds samples to the location where the difference between the MFSM and the HF surrogate model is largest (Kaya et al. 2019). In all these methods, one sample is determined for each infill iteration regardless of the fidelity level of the evaluation. Thus, computational resources may not be fully utilized, in particular, when LF evaluation is performed which in general is less demanding. Therefore, batch sampling, i.e., identify multiple infill samples at one iteration to allow parallelized evaluation of the LF model is highly desirable, which will improve not only resource utilization, but also exploration for global optimum localization in MFSBO. Le Gratiet and Cannamela presented a co-Kriging-based sequential design method, which evaluates both the MSE from the predictor and the errors from the leave-one-out cross-validation procedure, forming a modified co-Kriging variance. It then determines the infill samples based on the combined errors above (Le Gratiet and Cannamela 2015). It also infills a batch

of samples and determines the fidelity of simulation for infill. Similarly, Shi et al. proposed a novel dual-sampling-based co-Kriging method for optimization (Shi et al. 2018). The dual-sampling approach consists of trust-region- and MSE prediction-based sampling modules. Trust-region-based sampling identifies a sub-region in the design space according to the minimum of the surrogate model in the current iteration, and sampling is then undertaken within this region. MSE prediction-based sampling infills a sample where the MSE is the largest.

1.4 Dissertation Organization

The rest of the study is organized as follows. The fundamentals of design optimization for μCGG are presented in Chapter 2, including the μCGG and its data generation engine, data-driven modeling techniques, adaptive sampling strategies, and optimization algorithms. The SBO methodology and corresponding results and discussion of μCGG design are introduced in Chapter 3. In Chapter 4, in order to combine the data from different sources with varying fidelities and computational costs to accelerate the optimization process and improve design accuracy, an MFSBO methodology for μCGG design is proposed. Chapter 5 elucidates a novel computation-aware MFSBO methodology and a new sequential and adaptive sampling strategy based on expected improvement reduction (EIR). Chapter 6 introduces a new data sparsification method for MFSBO to balance between exploration and exploitation during optimization and reduce surrogate modeling complexity and time for improved efficiency. In Chapter 7, a multi-fidelity reduced-order model (MFROM) is presented for rapid and accurate simulation and thus

utilized for the design of μCGGs. Finally, the study concludes with a summary and future work in Chapter 8.

CHAPTER 2 RESEARCH FUNDAMENTALS

In this chapter, the fundamentals for design optimization are introduced, including the structure of the Triple-Y μCGG, simulations (PBCM and CFD) for data generation, surrogate modeling methods, infill strategies, and genetic algorithm (GA).

2.1 Microfluidic Concentration Gradient Generators and Data Sources

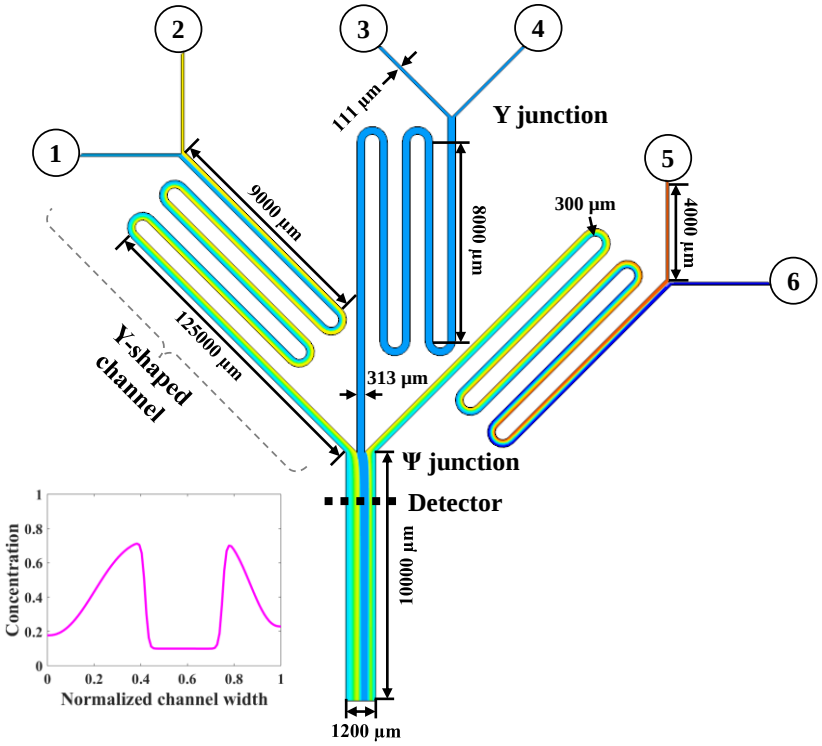


Figure 2.1 Schematic of the Triple-Y μCGG.

The proposed methods in the following chapters are implemented and demonstrated for a triple-Y μCGG (Y. Wang, Mukherjee, and Lin 2006; Y. Zhou et al. 2009). The configuration of the triple-Y μCGG is shown in Figure 2.1, which is comprised of six inlets and one outlet. All channels have a depth of h = 60 μm with the aspect ratio of 5 to 20, and other geometric parameters of the μCGG are given in Figure 2.1. Phosphate-buffered saline (PBS) with a viscosity of 0.001 kg m^{-1} s^{-1} is used as the carrier fluid. A chemical with a diffusivity of 1×10^{-10} m^2 s^{-1} is used to form various CGs. In each Y-shaped mixer, two

streams containing chemicals of different concentrations enter the μCGG via the two inlets, and then merge together and diffuse transversely within the mixing channel following the Y-junction. At the end of the mixing channel, a monotonically increasing or decreasing linear CG is generated. Subsequently, constituent CGs emanating from all the three Y-shaped mixers are concatenated along the width direction in the Ψ-shaped junction to form an even more complex CG at the entrance of the main output channel. Likewise, the chemicals carried by the three streams will also diffuse within the main output channel, and the extent of mixing depends on the location relative to the entrance. Both the chemical concentrations at the inlets and the pressure (or equivalently the flow rates) can be used to tune precisely the generated CGs. For example, a large flow rate driven by a large pressure head applied to the inlet will reduce the residence time of the chemical and inter-stream diffusion within the microchannels, resulting in a sharp gradient of the chemical concentration. On the other hand, a small flow rate and pressure head leads to milder CGs. In addition, unequal pressure or flow rates among the three Y-shaped mixers will also give rise to different widths of the constituent CGs in the concatenated one. In this study, two models with different levels of accuracy and fidelity are used to simulate this triple-Y μCGG: PBCM and CFD, which are detailed below.

2.1.1 Physics-based Component Model

The PBCM, treated as the LF model in this study, decomposes the triple-Y μCGG network of complex topology into a set of constituent components: microchannels (straight or curved), Y junctions, inlet reservoirs, and outlet reservoirs as shown in Figure 2.2. Due to the simple geometries of these components, analytical solutions of the species transport equation within them that considers the species convection and diffusion given the

background flow, can be obtained analytically. The component models are then connected in correspondence to the desired μCGG topology to form a network model that can typically run orders of magnitude faster than high-fidelity CFD simulation because of its analytical, closed-form nature, but less accurate due to the assumptions used in the model.

PBCM considers the fluid flow and the species transport separately within each constituent component above. Since the full set of the models were reported previously (Y. Wang, Mukherjee, and Lin 2006; Y. Zhou et al. 2009), the important ones for the microchannel and the Y-junction are described here briefly for the sake of completeness of the work. The microchannel is used for mixing and diffusion of chemicals along the channel width to form desired CGs. The fluid flow within the microchannel is modeled using the electric analogy and its hydrodynamic resistance is given in our previous work (Y. Wang, Mukherjee, and Lin 2006). To model the species transport, two assumptions are taken, that is, the channel is flat with a large aspect ratio and long. With a flat channel, the effect on the chemical transportation due to nonuniform velocity distribution along the channel cross-section is negligible and the convection term in the transport equation can be approximated by the cross-sectionally averaged velocity. Within a long channel, the axial diffusion is also negligible (Y. Wang, Mukherjee, and Lin 2006). The simplification allows analytical solution to the convection-diffusion equation, in which the chemical concentration is represented by a Fourier series, and the relationship of the Fourier coefficients (d_n) between the inlet and the outlet is given by

$$\begin{aligned} d_n^{(out)} &= d_n^{(in)} e^{-(n\pi)^2 \tau} \ , \\ \tau &= \frac{lD}{Vw^2} \end{aligned} \tag{2.1}$$

where l is the length of the channel, w is the width of the channel, D is the molecular diffusivity of the chemical, and V is the average flow velocity along the cross-section.

For the Y junction, two streams enter from the inlets, and are combined as a single stream exiting through the outlet. The flow resistance between the inlets and the outlet of the Y-shaped junction is assumed zero, that is, it is treated as a point-wise component without the physical size. The relationship between Fourier coefficients $d_n^{(in)}$ and $d_n^{(out)}$ of the concentration profile at the inlets and the outlet is

$$d_n^{(out)} = \begin{cases} d_0^{(L)} s + d_0^{(R)} (1-s), & n = 0 \\ s \sum_{m=0}^{\infty, m \neq ns} d_m^{(L)} \dfrac{f_1 \sin(f_2) + f_2 \sin(f_1)}{f_1 f_2} + s \sum_{m=0}^{\infty, m=ns} d_m^{(L)} + (1-s) \sum_{m=0}^{\infty, m=n(1-s)} (-1)^{n-m} d_m^{(R)} & \\ +2(-1)^n (1-s) \sum_{m=0}^{\infty, m \neq n(1-s)} d_m^{(R)} \left(\dfrac{\cos(F_2/2)\sin(F_1/2)}{F_1} + \dfrac{\cos(F_1/2)\sin(F_2/2)}{F_2} \right), & n \geq 1 \end{cases} \quad (2.2)$$

where L, R, and out denote the left inlet, right inlet, and outlet, respectively; $s = q^{(L)}/(q^{(L)}+q^{(R)})$ denotes the flow ratio of the left stream to the combined stream at the Y-junction, and also the normalized position of the stream interface; q is the flow rate. $f_1 = (m\text{-}ns)\pi$, $f_2 = (m+ns)\pi$, $F_1 = (m+n\text{-}ns)\pi$, and $F_2 = (m+\mathrm{n}+ns)\pi$. A Ψ-shaped junction consisting of three inlets and one outlet can be treated as a cascade concatenation of two Y-shaped junctions as shown in Figure 2.2, and the Fourier coefficients are obtained by computing Eq. (2.2) twice. That is, the Fourier coefficients at the outlet of the first Y-shaped junction is supplied to the left inlet of the second Y-shaped junction.

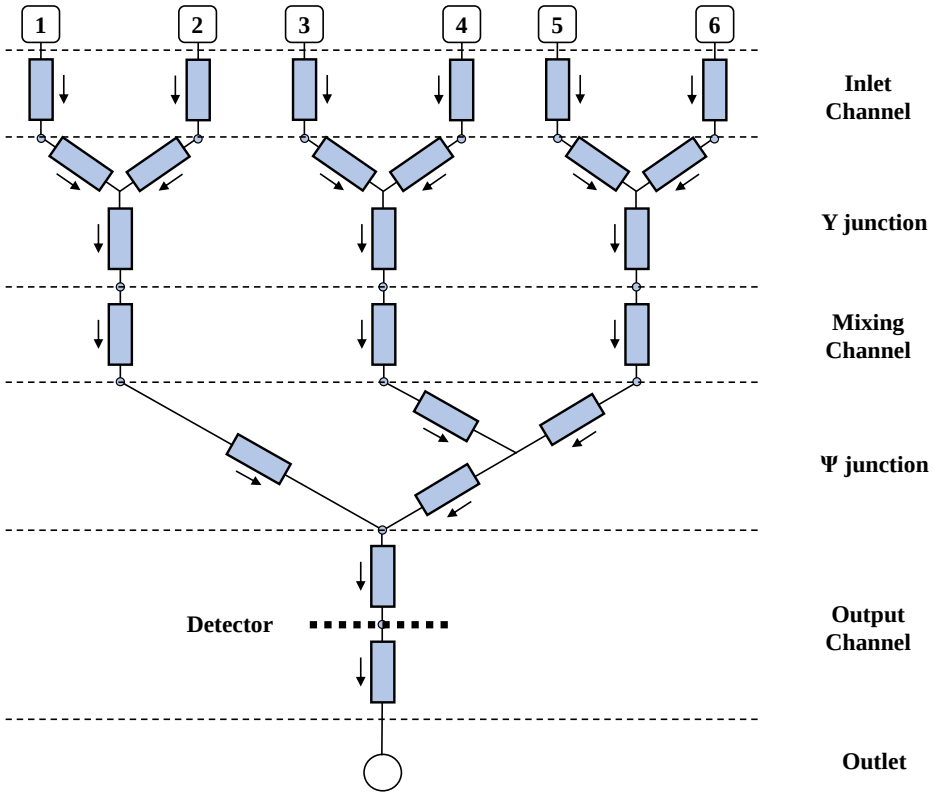


Figure 2.2 Component model of the Triple-Y μCGG.

All the PBCMs above in this study are developed in MATLAB (www.mathworks.com), and the simulation is carried out in two serial steps. First, the pressure and the flow distribution within the μCGG network is simulated following the Kirchhoff's law given the boundary conditions, i.e., the pressure and/or flow rate specified at the inlet and outlet reservoirs. Next, the Fourier coefficients of the concentration profiles are calculated along the flow direction determined in the previous step, and the calculation is initiated from inlet reservoirs where constant concentrations of the chemical are specified as the design variables in SBO. The coefficients $\left\{d_n^{(out)}\right\}^j$ at the outlet of the j^{th} component are computed using those at its inlet(s), and then assigned to those at the inlet of the component immediately downstream. It should be noted that PBCM above is applicable to both the partial mixing- and the complete mixing-based μCGG (Y. Wang, Mukherjee, and

Lin 2006), while in this study only demonstrated for the former that involves species transport along the width of each component and is more challenging to design (Gorman and Wikswo 2008). The PBCM was previously verified by both CFD simulations (Y. Wang, Mukherjee, and Lin 2006) and experiments (Y. Zhou et al. 2009), and it exhibited good agreement when the branch streams entering the Y and Ψ junction have the same flow rate. However, it was found in our recent study that distinctly different flow rates of the branch streams cause appreciable errors of PBCM prediction immediately after the junctions (H. Yang et al. 2020).

2.1.2 Computational Fluid Dynamic Model

CFD simulation is performed with the commercial package STAR-CCM+ based on the finite volume method, and serves as the HF model in this study. The computational domain is discretized into 785,664 structured cells, and fine cells are applied at all the junctions to capture steep velocity fields and concentration gradients. CFD is based on the discretized numerical formulation of the PDEs with fewer assumptions, and hence, is more accurate to represent a higher-level fidelity. The steady, laminar, passive scalar models are selected in STAR-CCM+, which solves the continuity equation, the Navier–Stokes equation, and the convection-diffusion equation for flow and species transport in the three-dimensional μCGGs. The equations are given as follows:

$$\nabla \cdot \left(\rho V \right) = 0 \tag{2.3}$$

$$\rho \frac{dV}{dt} = \rho g - \nabla p + \mu \nabla^2 V \tag{2.4}$$

$$\frac{\partial c}{\partial t} = \nabla \cdot \left(D \nabla c \right) - \nabla \cdot \left(Vc \right) + R \tag{2.5}$$

where ρ is the fluid density, V is the flow velocity, g is the gravitational acceleration, p is the pressure, μ is the dynamic viscosity, c is the chemical species concentration, and R is the source term. The segregated flow solver is applied and the algebraic multigrid (AMG) iterative method is used for solving the linearized algebraic equations. The computational time for CFD simulation is 1,325 s per simulation, and much longer than that of PBCM. Figure 2.3 illustrates the example results of CGs obtained by PBCM and CFD models given the same simulation parameters, and an appreciable difference between them is clearly observed.

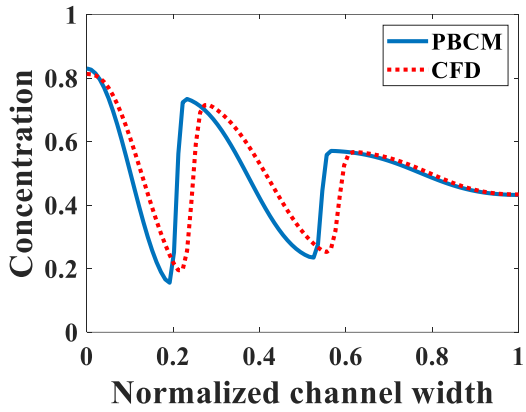


Figure 2.3 Example results and comparison between PBCM and CFD model given the same simulation parameters.

2.1.3 Automated Simulations

Performing design optimization for μCGG requires automated simulation (S. H. Hong et al. 2021; Shu et al. 2020), that is, all the simulations need to be invoked and controlled by the optimization process, which in the present work is implemented in the MATLAB environment. The details of the automated simulation are given as follows. First, the design algorithm supplies the input samples to the simulation solver, i.e., PBCM and

CFD. Second, automated simulations will be performed either with PBCM or by STAR-CCM+ computation. The PBCM was also developed within MATLAB, and hence, can be directly simulated in the optimization environment. CFD simulation in commercial software STAR-CCM+ can be started in the batch mode through JavaScript that also can be invoked by MATLAB. Third, the CG at the detector is extracted and new samples for the next round of simulation will be determined by the design algorithm. This fully automated process is repeated till the end design optimization iteration.

2.2 Surrogate and Reduced-order Models

2.2.1 Kriging

The kriging interpolation method first proposed by Krige and Sacks is mainly used to predict the unknown response based on existing samples by minimizing prediction's MSE (Z.-H. Han and Zhang 2012). To construct a Kriging model that maps the relationship between input and output, two data sets corresponding to inputs and responses are needed

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}^{(1)} \\ \vdots \\ \mathbf{x}^{(np)} \end{pmatrix}, \mathbf{Y} = \begin{pmatrix} \mathbf{y}^{(1)} \\ \vdots \\ \mathbf{y}^{(np)} \end{pmatrix} \tag{2.6}$$

where $\mathbf{X}$ is input samples; $\mathbf{Y}$ is the responses of the samples; np is the total number of data pairs. Kriging model consists of two components: a regression model $f(\mathbf{x})$ to represent the global trend of the input-response mapping, and a Gaussian process model $Z(\mathbf{x})$ with zero mean and variance σ^2 to capture the residuals from the data points to the surface of the global trend function (Ivo Couckuyt, Dhaene, and Demeester 2014). Mathematically, the Kriging model reads

$$y(\mathbf{x}) = f(\mathbf{x}) + Z(\mathbf{x}), \mathbf{x} \in R^k \tag{2.7}$$

In general, the regression model can be either a constant or low-order polynomials (first-order polynomial in this study)

$$f(\mathbf{x}) = \sum_{i=1}^{p} \gamma_i b_i(\mathbf{x}) \tag{2.8}$$

where b_i is the polynomial basis functions and γ_i is the corresponding coefficients. Therefore, in order to calculate the γ_i in Eq. (2.8), the regression matrix **F** is assembled first

$$\mathbf{F} = \begin{pmatrix} b_1(\mathbf{x}^{(1)}) & \cdots & b_p(\mathbf{x}^{(1)}) \\ \vdots & \ddots & \vdots \\ b_1(\mathbf{x}^{(np)}) & \cdots & b_p(\mathbf{x}^{(np)}) \end{pmatrix} \tag{2.9}$$

The correlation matrix of the Gaussian process *Z* model is defined as

$$\mathbf{\Psi} = \begin{pmatrix} \psi(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}) & \cdots & \psi(\mathbf{x}^{(1)}, \mathbf{x}^{(\mathrm{np})}) \\ \vdots & \ddots & \vdots \\ \psi(\mathbf{x}^{(\mathrm{np})}, \mathrm{x}^{(1)}) & \cdots & \psi(\mathbf{x}^{(\mathrm{np})}, \mathbf{x}^{(\mathrm{np})}) \end{pmatrix} \tag{2.10}$$

where ψ represents a correlation function that measures the relationship between two samples based on their Euclidean distance. The smaller distance between two points results in higher correlation, and therefore, the closer function values. In this study, a widely used form of the correlation function is adopted

$$\psi(\mathbf{x}, \mathbf{x}') = \exp\left(-\sum_{j=1}^{k} \theta_j \left| \mathbf{x}_j - \mathbf{x}'_j \right|^{p_j} \right) \tag{2.11}$$

where θ and p are hyperparameters to be determined and subscript j = 1, 2, …, k denotes the j^{th} dimension of the input and k is the number of dimensions of the input. Typically, p can be specified as a constant, e.g., p = 2 corresponding to a Gaussian correlation function (adopted in this study) and p = 1 for an exponential correlation function. The natural log of the marginal likelihood is given by

$$-\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\ln|\mathbf{\Psi}| - \frac{(\mathbf{Y} - \mathbf{F}\boldsymbol{\gamma})^{\mathrm{T}} \mathbf{\Psi}^{-1} (\mathbf{Y} - \mathbf{F}\boldsymbol{\gamma})}{2\sigma^2} \tag{2.12}$$

where $\boldsymbol{\gamma} = [\gamma_1\ \gamma_2\ \ldots\ \gamma_p]^T$. Maximum Likelihood Estimation (MLE) is applied to Eq. (2.12) to compute the hyperparameters in the Kriging model ($\boldsymbol{\gamma}$, $\boldsymbol{\theta}$, σ^2). Then, the prediction $\hat{y}(\mathbf{x})$ should follow the Gaussian process governed by the same correlation parameters identified in the training process. Subsequently, the predicted mean is obtained by maximizing the augmented likelihood function, i.e.,

$$\hat{y}(\mathbf{x}) = \mathbf{m}^T\boldsymbol{\gamma} + \mathbf{r}(\mathbf{x})^T\,\boldsymbol{\Psi}^{-1}(\mathbf{Y} - \mathbf{F}\boldsymbol{\gamma}) \tag{2.13}$$

where

$$\mathbf{m} = \begin{pmatrix} b_1(\mathbf{x}) & b_2(\mathbf{x}) & \cdots & b_p(\mathbf{x}) \end{pmatrix}^T \tag{2.14}$$

$$\boldsymbol{\gamma} = \left(\mathbf{F}^T\boldsymbol{\Psi}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}^T\boldsymbol{\Psi}^{-1}\mathbf{Y} \tag{2.15}$$

$$\mathbf{r}(\mathbf{x}) = \left[\psi(\mathbf{x},\mathbf{x}^{(1)}) \quad \cdots \quad \psi(\mathbf{x},\mathbf{x}^{(np)})\right]^T \tag{2.16}$$

The estimated mean-squared error by the predictor is

$$s^2(\mathbf{x}) = \sigma^2\left(1 + \mathbf{u}^T\left(\mathbf{F}^T\boldsymbol{\Psi}^{-1}\mathbf{F}\right)^{-1}\mathbf{u} - \mathbf{r}(\mathbf{x})^T\,\boldsymbol{\Psi}^{-1}\mathbf{r}(\mathbf{x})\right) \tag{2.17}$$

where $u = F^T\Psi^{-1}r(\mathbf{x})-m$.

2.2.2 Multi-fidelity Surrogate Model (Cokriging)

The MFSM fits the data generated by simulation of different fidelities and analyzes the correlation between them (Marques et al. 2019; Giselle Fernández-Godino et al. 2019; Park, Haftka, and Kim 2017). It is able to reduce the computational cost while preserving model accuracy by judiciously assigning the computational resource to models/simulations of different fidelities. Cokriging is one of the widely used MFSM methods, and is also adopted in this work (Forrester, Sóbester, and Keane 2008; Shi et al. 2018; Le Gratiet and Cannamela 2015). As an extension of Kriging, the co-Kriging method is also a Gaussian process model that establishes the correlation between two data sets of different natures and fidelities (Shi et al. 2018; Forrester, Sóbester, and Keane 2008). To construct a co-

Kriging-based MFSM, two data sets with sampled inputs and the responses are needed, which are given by

$$\mathbf{X}=\begin{pmatrix}\mathbf{X}_L\\ \mathbf{X}_H\end{pmatrix}=\begin{pmatrix}X_L^{(1)}\\ \vdots\\ X_L^{(n_L)}\\ X_H^{(1)}\\ \vdots\\ X_H^{(n_H)}\end{pmatrix},\mathbf{Y}=\begin{pmatrix}\mathbf{Y}_L\\ \mathbf{Y}_H\end{pmatrix}=\begin{pmatrix}Y_L^{(1)}\\ \vdots\\ Y_L^{(n_L)}\\ Y_H^{(1)}\\ \vdots\\ Y_H^{(n_H)}\end{pmatrix} \tag{2.18}$$

where **X** is the inputs of the sample points; **Y** is the corresponding responses/outputs of the samples; *n* is the number of sample points; subscripts *H* and *L* represent the data from HF (computationally expensive) and LF (computationally cheap) models, respectively. Then, the relationship among three Gaussian processes Z_H, Z_L, and Z_d, for the HF model, the LF model, and the difference between them, need to be established. That is, the Gaussian process of the HF model Z_H is represented by the Gaussian process of the LF model Z_L multiplied by a constant ρ plus another Gaussian process Z_d (Forrester, Sóbester, and Keane 2007):

$$Z_H(x)=\rho Z_L(x)+Z_d(x) \tag{2.19}$$

where x is the input in the parameter space. The covariance matrix governing the entire process in Eq. (2.19) is expressed as follows

$$\begin{aligned}\mathbf{C}&=\begin{bmatrix}\operatorname{cov}\{\mathbf{Y}_L(\mathbf{X}_L),\mathbf{Y}_L(\mathbf{X}_L)\} & \operatorname{cov}\{\mathbf{Y}_L(\mathbf{X}_L),\mathbf{Y}_H(\mathbf{X}_H)\}\\ \operatorname{cov}\{\mathbf{Y}_H(\mathbf{X}_H),\mathbf{Y}_L(\mathbf{X}_L)\} & \operatorname{cov}\{\mathbf{Y}_H(\mathbf{X}_H),\mathbf{Y}_H(\mathbf{X}_H)\}\end{bmatrix}\\ &=\begin{bmatrix}\sigma_L^2\boldsymbol{\Psi}_L(\mathbf{X}_L,\mathbf{X}_L) & \rho\sigma_L^2\boldsymbol{\Psi}_L(\mathbf{X}_L,\mathbf{X}_H)\\ \rho\sigma_L^2\boldsymbol{\Psi}_L(\mathbf{X}_H,\mathbf{X}_L) & \rho^2\sigma_L^2\boldsymbol{\Psi}_L(\mathbf{X}_H,\mathbf{X}_H)+\sigma_d^2\boldsymbol{\Psi}_d(\mathbf{X}_H,\mathbf{X}_H)\end{bmatrix}\end{aligned} \tag{2.20}$$

where σ_L and σ_d represent the standard deviations of Z_L and Z_d, respectively. $\boldsymbol{\Psi}_L$ and $\boldsymbol{\Psi}_d$ represent the correlations of Z_L and Z_d, respectively. Applying the MLE for LF data yields

$$-\frac{n}{2}\ln\left(\sigma_c^2\right)-\frac{1}{2}\ln\left|\boldsymbol{\Psi}_L\left(\mathbf{X}_L,\mathbf{X}_L\right)\right|-\frac{\left(\mathbf{Y}_L-\mathbf{1}\mu_L\right)^T\boldsymbol{\Psi}_L\left(\mathbf{X}_L,\mathbf{X}_L\right)^{-1}\left(\mathbf{Y}_L-\mathbf{1}\mu_L\right)}{2\sigma_L^2} \quad (2.21)$$

where μ_L is the mean of Z_L. The hyperparameters for the mean μ_L and the variance σ_L of Z_L are obtained:

$$\begin{aligned}\mu_L&=\frac{\mathbf{1}^{\mathrm{T}}\boldsymbol{\Psi}_L\left(\mathbf{X}_L,\mathbf{X}_L\right)^{-1}\mathbf{Y}_L}{\mathbf{1}^{\mathrm{T}}\boldsymbol{\Psi}_L\left(\mathbf{X}_L,\mathbf{X}_L\right)^{-1}\mathbf{1}},\\ \sigma_L^2&=\frac{\left(\mathbf{Y}_L-\mathbf{1}\mu_L\right)^T\boldsymbol{\Psi}_L\left(\mathbf{X}_L,\mathbf{X}_L\right)^{-1}\left(\mathbf{Y}_L-\mathbf{1}\mu_L\right)}{n_L}\end{aligned} \quad (2.22)$$

Similarly, the MLEs for HF data are performed, leading to

$$-\frac{n}{2}\ln\left(\sigma_d^2\right)-\frac{1}{2}\ln\left|\boldsymbol{\Psi}_L\left(\mathbf{X}_L,\mathbf{X}_L\right)\right|-\frac{\left(\mathbf{d}-\mathbf{1}\mu_d\right)^T\boldsymbol{\Psi}_d\left(\mathbf{X}_H,\mathbf{X}_H\right)^{-1}\left(\mathbf{d}-\mathbf{1}\mu_d\right)}{2\sigma_d^2} \quad (2.23)$$

where μ_d is the mean of Z_d, where $\mathbf{d} = \mathbf{Y}_H$-$\rho\hat{y}_L(\mathbf{X}_H)$, and $\hat{y}_L$ is the LF surrogate model prediction. Then, the hyperparameters for the mean μ_d and the variance σ_d of Z_d are given as

$$\begin{aligned}\mu_d&=\frac{\mathbf{1}^{\mathrm{T}}\boldsymbol{\Psi}_d\left(\mathbf{X}_H,\mathbf{X}_H\right)^{-1}\mathbf{d}}{\mathbf{1}^{\mathrm{T}}\boldsymbol{\Psi}_d\left(\mathbf{X}_H,\mathbf{X}_H\right)^{-1}\mathbf{1}},\\ \sigma_d^2&=\frac{\left(\mathbf{d}-\mathbf{1}\mu_d\right)^T\boldsymbol{\Psi}_d\left(\mathbf{X}_H,\mathbf{X}_H\right)^{-1}\left(\mathbf{d}-\mathbf{1}\mu_d\right)}{n_H}\end{aligned} \quad (2.24)$$

The unknown parameters in the Gaussian process and constant ρ are also determined by MLEs (Forrester, Sóbester, and Keane 2008). Finally, the predictor for MFSM is given by

$$\hat{y}\left(x\right)=\hat{\mu}+\mathbf{c}^{\mathrm{T}}\mathbf{C}^{-1}\left(\mathbf{Y}-\mathbf{1}\hat{\mu}\right) \quad (2.25)$$

where

$$\mathbf{c} = \begin{pmatrix} \hat{\rho}\hat{\sigma}_L^2 \boldsymbol{\Psi}_L\left(\mathbf{X}_L, x\right) \\ \hat{\rho}^2\hat{\sigma}_L^2 \boldsymbol{\Psi}_L\left(\mathbf{X}_H, x\right) + \hat{\sigma}_d^2 \boldsymbol{\Psi}_d\left(\mathbf{X}_H, x\right) \end{pmatrix} \tag{2.26}$$

$$\hat{\mu} = \mathbf{1}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{Y} / \mathbf{1}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{1} \tag{2.27}$$

The estimated mean-squared error by the predictor is

$$s^2\left(x\right) = \hat{\rho}^2\hat{\sigma}_L^2 + \hat{\sigma}_d^2 - \mathbf{c}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{c} \tag{2.28}$$

2.2.3 Proper Orthogonal Decomposition

Consider a set of data $\mathbf{C} \in \mathbb{R}^{mp\times np}$, where each column of **C** represents a snapshot/date instance, that is $\mathbf{C}^{(k)} \in \mathbb{R}^{mp}$, where $k = 1, 2, \ldots, np$, and np is the number of snapshots. In this chapter, one snapshot corresponds to a CG profile/vector generated by the simulation (PBCM or CFD) with one set of design parameter values. Although there exist different approaches to compute POD basis vectors and coefficients, in this work singular value decomposition (SVD) is adopted and applied to matrix **C**,

$$\mathbf{C} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}} \tag{2.29}$$

where $\mathbf{U} \in \mathbb{R}^{mp\times np}$, $\mathbf{V} \in \mathbb{R}^{np\times np}$, and $\boldsymbol{\Sigma} \in \mathbb{R}^{np\times np}$. $\boldsymbol{\Sigma}$ is a diagonal matrix with singular values on its diagonal and sorted in the descending order, viz., $\sigma_1 > \sigma_2 > \cdots > \sigma_{\mathrm{np}}$. Each column of **U** represents the POD orthonormal basis vector $\mathbf{u}_i$ and $i = 1, 2, \ldots, np$. A salient feature of SVD is that the first several leading POD basis vectors, corresponding to large singular values, represent the most important structure underlying the data and contribute most when the data needs to be reconstructed. Therefore, a low-rank approximation of the data matrix **C** can be obtained through truncation, i.e., only keeping the leading POD basis vectors (also called modes)

$$\mathbf{C} \approx \mathbf{U}_{\mathrm{R}}\boldsymbol{\Sigma}_{\mathrm{R}}\mathbf{V}_{\mathrm{R}}^{\mathrm{T}} \tag{2.30}$$

where $\mathbf{U}_R \in \mathbb{R}^{mp\times k}$, $\mathbf{V}_R \in \mathbb{R}^{np\times k}$, and $\mathbf{\Sigma}_R \in \mathbb{R}^{k\times k}$. k is the number of the truncated POD basis vectors. The accuracy of POD approximation can be estimated using the energy ratio, which is defined as

$$\gamma = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{np} \sigma_i^2} \geq \gamma_{\text{lim}} \tag{2.31}$$

where γ_{lim} is the threshold of energy level to determine the number of the POD basis vectors to keep and the accuracy of truncation for data approximation. In this study, an empirical value $\gamma_{\text{lim}} = 0.9999$ is adopted. In addition to the POD basis vectors $\mathbf{U}_R$, the POD modal coefficient matrix can also be obtained by $\mathbf{A} = \mathbf{\Sigma}\mathbf{V}^T \in \mathbb{R}^{k\times np}$ and $\mathbf{A} = \begin{bmatrix} \boldsymbol{\alpha}_1 & \dots & \boldsymbol{\alpha}_i & \dots & \boldsymbol{\alpha}_{np} \end{bmatrix}$, where $i = 1, 2, \ldots, np$. Applying the POD onto both the PBCM and the CFD simulation yields, respectively, the POD basis matrix $\mathbf{U}_R^{\text{PBCM}}$ and $\mathbf{U}_R^{\text{CFD}}$. Correspondingly, $\boldsymbol{\alpha}_i^{\text{PBCM}}$ and $\boldsymbol{\alpha}_i^{\text{CFD}}$ are the POD modal coefficient of the i^{th} snapshot/data instance of all the *np* PBCM and CFD simulations, respectively.

2.3 Adaptive Sampling and Infill Strategies

Adaptive sampling and infill, is a key technique that exploits response surface information of the existing surrogate model and adds new samples and information at critical regions within the design space to further refine the surrogate model for optimization (Parr et al. 2012). Through a discreet selection of infill points, accurate surrogate models can be constructed with a small number of samples (Cozad, Sahinidis, and Miller 2014). Normally the infill process is repeated until stopping criteria are satisfied, such as the number of maximum iterations and error tolerance. The infill process is

embedded in the optimization loop, the choice of infill techniques and criteria is critical for optimization performance. In this chapter, four different infill techniques: MP, LB, PI, and EI are introduced.

MP determines the infill location at the minimum of the current surrogate model, which is defined as:

$$\text{Min} \quad \hat{y}(x) \tag{2.32}$$

where $\hat{y}(x)$ is the prediction of the surrogate model. MP is solely based on exploitation, that is, MP believes the surrogate model is globally accurate and only infills at the location where the minimum of the surrogate model is present. The MP works well when the surrogate model is generally accurate. However, if the surrogate model is not accurate enough and the minimum is not trustworthy, the infill samples from MP may not identify the good sample location or benefit the optimization.

LB is defined as:

$$\text{Min} \quad \text{LB}(x) = \hat{y}(x) - A\hat{s}(x) \tag{2.33}$$

where $\hat{y}(\text{x})$ and $\hat{s}(\text{x})$ are the prediction and MSE of the surrogate model at the input variable x, respectively. A is a constant that balances between the exploitation term $\hat{y}(\text{x})$ and exploration term $\hat{s}(\text{x})$ (Forrester, Sóbester, and Keane 2008) for sample selection, and in this study, an empirical value of $A = 2$ is accepted. Exploitation uses the information of the optimum in the current iteration, and select infill points close to it. It is well suited for the scenario when the global optimum is relatively easy to find. Exploration focuses on the regions that are unknown or under-sampled, and then adds infill points where the surrogate model exhibits the large MSE, and hence, is more suitable for complex problems where multiple optimums appear in the response surface, such as the multi-modal response

surface. In this chapter, the infill point is obtained by finding the sample location that minimizes the statistical lower-bound.

PI selects an infill point that leads to a maximum probability of an improvement, $I = y_{\min} - \hat{y}(x)$ (Ivo Couckuyt, Deschrijver, and Dhaene 2014), where $y_{\min}$ is the minimum observation of existing samplings. PI is expressed as an error function as shown in Eq. (2.34), and maximizing it yields the infill point

$$\text{Max} \quad P\left[I(x)\right] = \frac{1}{\hat{s}\sqrt{2\pi}} \int_{-\infty}^{0} e^{\left(-(I-\hat{y}(x))^2\right)/2s^2} \mathrm{d}I \tag{2.34}$$

EI, which is one of the widely used sample infill strategies for surrogate-based optimization problems, is utilized in this study. EI measures the integral of the PI (Forrester, Sóbester, and Keane 2008). The first term determines the infill based on the exploitation of the current minimum y_{min}, and the second term prioritizes exploration and samples the under-sampled region. The balance of exploitation and exploration is a typical consideration for acquisition functions of adaptive sampling. The infill is essentially to maximize the amount of improvement we expect (Forrester, Sóbester, and Keane 2008). The equation of EI is shown in Eq. (2.35).

$$\text{Max} \quad E\left[I(x)\right] = \begin{cases} \left(y_{\min} - \hat{y}(x)\right)\Phi\left(\dfrac{y_{\min} - \hat{y}(x)}{\hat{s}(x)}\right) + \hat{s}(x)\varphi\left(\dfrac{y_{\min} - \hat{y}(x)}{\hat{s}(x)}\right) & \text{if } s > 0 \\ 0 & \text{if } s = 0 \end{cases} \tag{2.35}$$

where $\Phi(x)$ and $\varphi(x)$ are the cumulative distribution function and the probability density function, respectively. This equation represents the area enclosed by the Gaussian distribution below the minimum of surrogate model found at the current iteration (Forrester

and Keane 2009). Similar to PI, the infill point is attained by maximizing the EI in Eq. (2.35).

2.4 Genetic Algorithm

For the design optimization, global optimization needs to be performed for sample infill and search for the minimum of the surrogate model. Despite various global optimization methods, the GA (Taylor 1994) is selected for use. GA evolves an initial population of random gene sequences, through many generations, and toward a final population of "fit" gene sequences that demonstrate optimal performance on a fitness function used to assess the performance of a given gene sequence. There are three basic genetic operations, reproduction, cross over, and mutation (Jahed Armaghani et al. 2018), which need to be undertaken to generate the next generation with consideration of both exploration and exploitation. The GA process will be terminated by evaluating certain stop criteria, e.g., meeting the desired tolerance in the fitness value or reaching the maximum number of generations.

CHAPTER 3 SURROGATE-BASED OPTIMIZATION FOR MICROFLUIDIC CONCENTRATION GRADIENT GENERATOR DESIGN[1]

[1] Yang, Haizhou, Seong Hyeon Hong, Rei ZhG, and Yi Wang. *RSC Advances* 10, no. 23 (2020): 13799-13814.

In this chapter, we propose a SBO with adaptive sampling framework to do design optimization for μCGG. In contrast to existing efforts of μCGG modeling and design, this chapter presents several novelties. First, to the best of our knowledge, it is an initial effort to establish SBO with adaptive sampling/infill method for μCGG design. Second, a comparative analysis is carried out to thoroughly investigate the effects of various combinations of correlation functions, regression functions, and infill strategies on surrogate model accuracy and SBO convergence for μCGG design. Last, a new formulation for SBO of μCGGs is proposed to avoid the backflow issue, that is, liquid solution unexpectedly exits through inlets of the μCGG network due to overly large difference of the pressure head among inlets. In this formulation, instead of the inlet pressures, the pressure differences between branch points within the μCGG network are used as design variables, which facilitates surrogate modeling and adaptive sampling. Note that the new formulation can potentially be extended to microfluidic electrokinetic flow driven by the electric field (Y. Wang, Lin, and Mukherjee 2005; Biddiss and Li 2005).

3.1 Methodology

Figure 3.1 illustrates the SBO process with adaptive sampling (Forrester and Keane 2009; Forrester, Sóbester, and Keane 2008; Z. H. Han et al. 2018; X. Chen et al. 2014), specifically for designing inlet operating parameters of μCGGs that allow generating user-desired/prescribed CGs. It includes initial sampling, model selection, surrogate modeling, surrogate model optimization, adaptive sampling (or infill), and iterative surrogate model update to gradually identify the global optimum parameters within the design space. The detailed procedure is given as follows: first, Latin hypercube sampling (LHS) (block

labeled ‘1’ in Figure 3.1), one kind of the one-shot space-filling techniques for the design of experiments (DoE), is used to generate initial samples in the multi-dimensional design space (Cozad, Sahinidis, and Miller 2014; Etikan 2016), which includes chemical concentrations at the inlet reservoirs and pressures (or flow rates). Second, the aforementioned PBCM (Y. Wang, Mukherjee, and Lin 2006)(labeled ‘2’ in Figure 3.1) representing the designated μCGG network is then simulated as the main engine to predict corresponding CGs at each sample obtained in the previous step for surrogate model construction and SBO. The discrepancy J_d between the generated CG C_o at the sampled point and the user-prescribed CG C_s, i.e., the Normalized Root Mean Squared Error (NRMSE) (Rai and Campbell 2007) is used as the output of the surrogate model. Next, the existing sampled points and their corresponding discrepancies J_ds relative to the user-prescribed CG are utilized as the input-out data pairs to construct the surrogate model (labeled ‘3’ in Figure 3.1). Despite a variety of surrogate model techniques available to establish the input-output mapping relationship (Forrester, Sóbester, and Keane 2008), the Kriging interpolation method that is comprised of a trend regression model and a correlation model is adopted in this chapter. Because of multiple choices of the regression model and the correlation model, the best combination of them needs to be selected and will be used for subsequent infill and SBO. Therefore, a model selection process (labeled ‘4’ in Figure 3.1) will be executed using the initial sampling data during the first iteration. That is, the data of initial sampling is divided into two subsets, and the first subset is used to construct the surrogate model, while the second to evaluate its accuracy.

Since the surrogate model is an approximation of the physics-based model, an adaptive sampling technique (also known as an infill) (labeled ‘5’ in Figure 3.1) will be

incorporated into SBO, which during each iteration will add a new sampled point and its corresponding discrepancy J_d computed by PBCM (labeled '2') into the data set to update the surrogate model (labeled '3') for enhanced accuracy. Essentially infill is a sub-optimization process to identify a new sample within the design space that minimizes or maximizes a specific infill criterion, and hence, providing more information than randomly selected samples for SBO. In addition, the surrogate model is very computationally efficient, and each evaluation only costs milli- to centi-second. As a result, it can be used to find the global optimum, e.g., using the genetic algorithm that entails a large number of model evaluations. The infill, PBCM simulation, and optimization will be repeated until the minimum of the surrogate model (labeled '6' in Figure 3.1) converges with respect to a predefined tolerance or the maximum number of iterations defined by the user is reached. Once converged, the optimum design (labeled '7' in Figure 3.1), selected from the minimum of the surrogate model and all existing samples in the last iteration, will be supplied to PBCM and CFD simulation to predict corresponding CGs, which then will be compared with prescribed CGs to verify SBO-based design of μCGGs. The detailed verification process is elucidated in section 3.3.

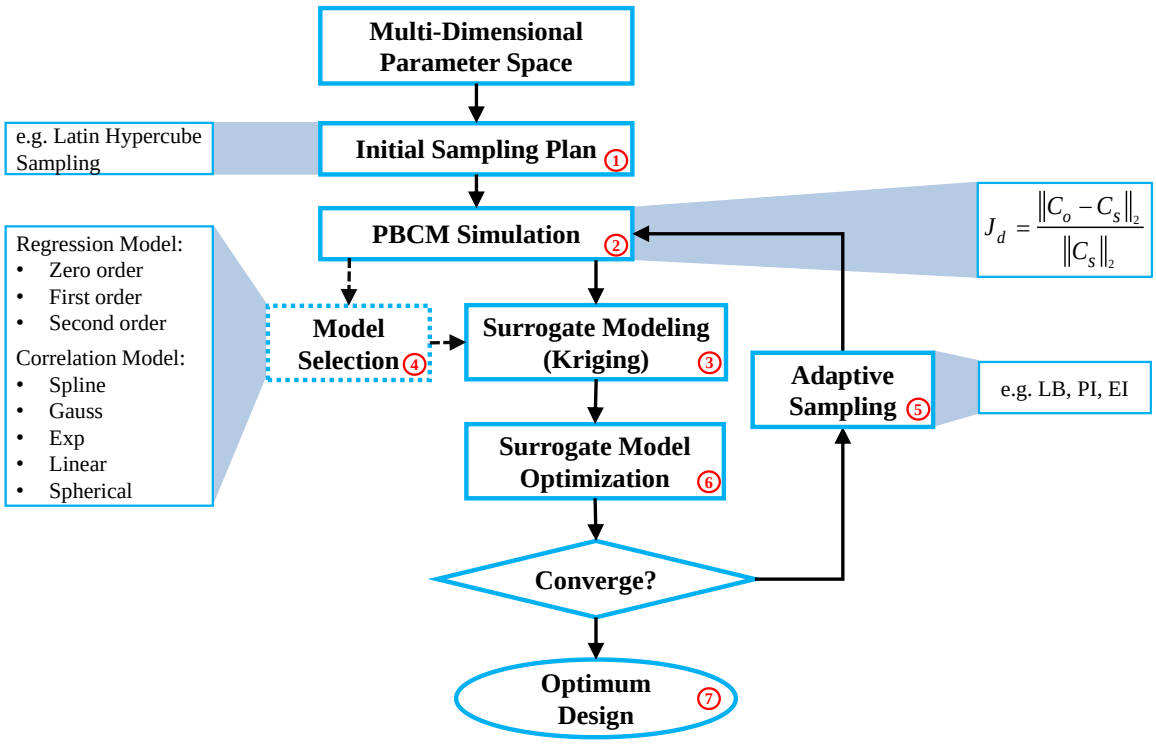


Figure 3.1 Flowchart of the SBO with adaptive sampling for µCGG.

3.2 Problem Formulation and Case Studies

In this section, the problem of µCGG design will be formulated, including the design variables and the cost function used in SBO, and case studies used to verify the proposed method will also be described. The first step of our SBO design formulation is to prescribe a desired CG C_s, such as the linear, sawtooth, trapezoidal, and others. The cost function is then defined as the l^2 norm of the discrepancy J_d between the CG created by the candidate design C_o at the detector location and the prescribed one C_s. It seems that the inlet parameters of the triple-Y µCGG should be used as the design variables to minimize J_d, such as inlet concentrations C_i and pressures p_i (or equivalently the flow rate q_i) at the inlet, where i denotes the i^{th} inlet. However, as shown in Figure 3.2, when the pressure at junction 7 is greatly higher than those at junction 8 or (and) 9 and the outlet 0, the pressure

at junction 10 may also be higher than that at Junction 8 or (and) 9. Thus a fraction of fluid from the first Y-shaped mixer will be diverted towards the second and the third Y-shaped mixer, and unexpectedly exit through inlet 3, 4, 5 and 6, viz., backflow, although the inlets are originally intended for inflow. To eliminate such an issue, the pressure difference between junctions in each Y-shaped mixer, instead of the inlet pressure is proposed as the design variables, which is one of the novelties of the present work. The flow conservation at junction 10, 7, 8, and 9, is written as

$$\begin{cases} \dfrac{\Delta p_{7,10}}{R_{7,10}} + \dfrac{\Delta p_{8,10}}{R_{8,10}} + \dfrac{\Delta p_{9,10}}{R_{9,10}} = \dfrac{\Delta p_{10,0}}{R_{10,0}} \\ \dfrac{(p_1 - (\Delta p_{10,0} + \Delta p_{7,10}))}{R_{1,7}} + \dfrac{(p_2 - (\Delta p_{10,0} + \Delta p_{7,10}))}{R_{2,7}} = \dfrac{\Delta p_{7,10}}{R_{7,10}} \\ \dfrac{(p_3 - (\Delta p_{10,0} + \Delta p_{8,10}))}{R_{3,8}} + \dfrac{(p_4 - (\Delta p_{10,0} + \Delta p_{8,10}))}{R_{4,8}} = \dfrac{\Delta p_{8,10}}{R_{8,10}} \\ \dfrac{(p_5 - (\Delta p_{10,0} + \Delta p_{9,10}))}{R_{5,9}} + \dfrac{(p_6 - (\Delta p_{10,0} + \Delta p_{9,10}))}{R_{6,9}} = \dfrac{\Delta p_{9,10}}{R_{9,10}} \end{cases} \tag{3.1}$$

where as shown in Figure 3.2, p, Δp, and R are the pressure, pressure difference, and resistance, respectively; the subscript with a single number or two numbers, respectively, denote the quantity at the junction or the quantity across the channel, e.g., pressure difference and resistance between two junctions. The pressure at the outlet is assumed zero in Eq. (3.1), i.e., grounded, and the pressure at junction 10, i.e., p_{10} is equal to $\Delta p_{10,0}$. Therefore, the inlet pressures p_1, p_2…, p_6 can be expressed using pressure differences $\Delta p_{7,10}$, $\Delta p_{8,10}$, and $\Delta p_{9,10}$. In our formulation, the incoming branch channels of each Y-shaped mixer and the pressures at their inlets are set the same, while the pressures could be different from one Y-shaped mixer to another, that is, $p_1 = p_2$, $p_3 = p_4$, $p_5 = p_6$, and $p_1 \neq p_3$

$\neq p_5$. By simply constraining the values of these pressure differences to be larger than zero, the backflow can be effectively eliminated because positive pressure differences of junction 7, 8, and 9 relative to junction 10 imply that all fluid streams enter the main output channel.

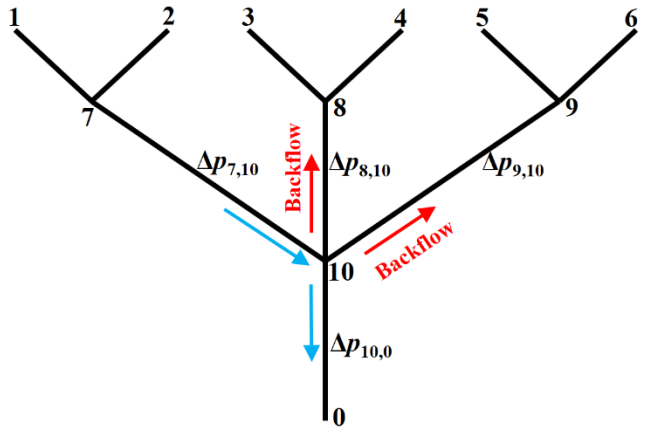


Figure 3.2 Illustration of the backflow issue and reformulation of the design problem to use the pressure difference as the design variables rather than the inlet pressure for the Triple-Y μCGG.

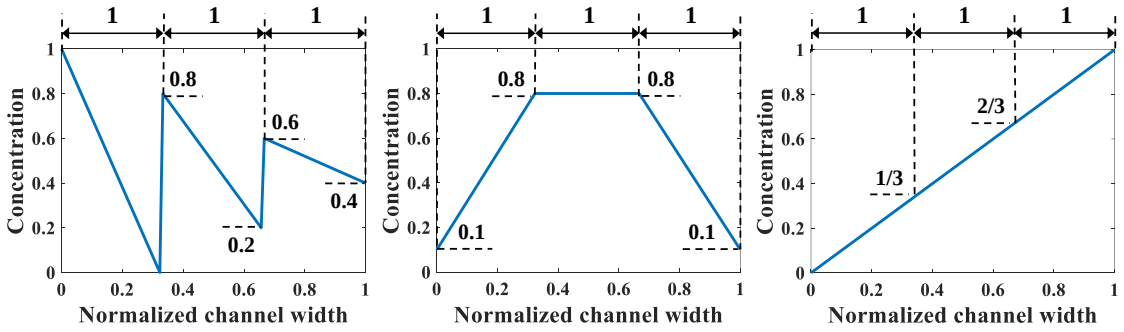
Mathematically, the SBO-based design of µCGGs can be summarized as follows:

$$\min_{x} \qquad J_d = \frac{\|C_o - C_s\|_2}{\|C_s\|_2} \tag{3.2}$$

where x is the vector of the design variables; recall that C_s and C_o are, respectively, the prescribed CG and the CG of candidate design extracted at the detector location; and C_o depends on the values of the design variables x. The CG is measured with 100 uniformly distributed probes along the channel width direction, and thus, both C_s and C_o are a 100-dimensional vector.

In this chapter, two case studies following the formulation above are investigated to verify SBO-based design of µCGGs. In the first one, only normalized chemical

concentrations at the six inlets are included as the design variables, i.e., $x = [c_1, c_2, \ldots, c_6]$ with c_i being a scalar, and the pressure difference applied across the merging channel of all the Y-shaped mixers is the same, i.e., $\Delta p_{7,10} = \Delta p_{8,10} = \Delta p_{9,10}$. This reduces the problem to six dimensions and is called design of inlet concentrations hereafter. In this case study, $p_1 = p_2 = p_3 = p_4 = p_5 = p_6 = 382.44$ pa is used, and correspondingly the flow rate through each inlet channel is fixed as 864 *nl*/min (Y. Wang, Mukherjee, and Lin 2006). Figure 3.3 illustrates three prescribed CGs, i.e., C_s that need to be generated by selecting appropriate inlet concentrations c_i, which include the sawtooth-shaped, trapezoidal, and linear CGs. The normalized concentration is in the range of [0 1]. Note that due to the same flow rate through each inlet channel, the three linear segments in the prescribed CGs have the same width.

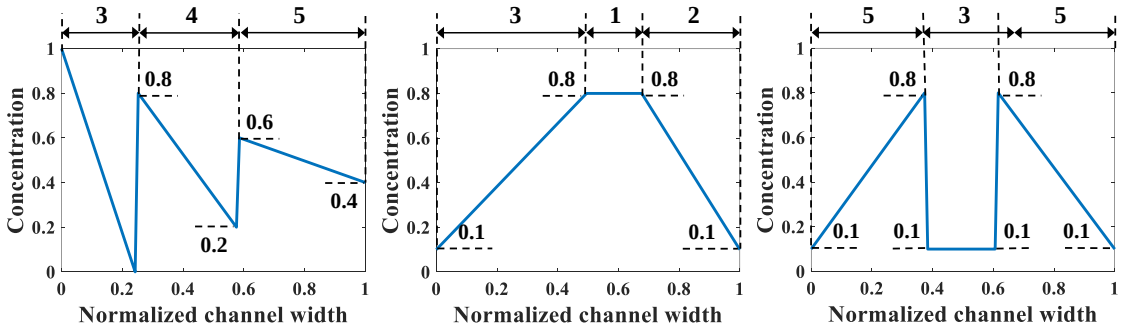


(a) Sawtooth-shaped **(b) Trapezoidal** **(c) Linear**

Figure 3.3 Prescribed CGs in the first case study: design of inlet concentrations.

In the second case study, the pressure difference across the merging channel of the three Y-shaped mixers ($\Delta p_{7,10}$, $\Delta p_{8,10}$, and $\Delta p_{9,10}$ in Figure 3.2), are introduced as three additional design variables to generate more complex CGs. This will increase the design dimension to 9, viz., $x = [c_1, \ldots, c_6, \Delta p_{7,10}, \Delta p_{8,10}, \Delta p_{9,10}]$, making it more challenging to

construct the surrogate model and to search optimal parameters corresponding to minimum J_d. Note that once the optimal values of the pressure difference are found, they can be converted to the inlet pressures at the reservoirs using Eq. (3.1). Because of the identical size of the inlet branch channels in all Y-shaped mixers, the two inlet pressures and the two flow rates through each Y-shaped mixer are equal, that is $q_1 = q_2$, $q_3 = q_4$, and $q_5 = q_6$. However, the flow rate can be different among them, i.e., $q_1 \neq q_3 \neq q_5$. Therefore, the μCGG in this design can generate CGs comprised of three segments with different widths as shown in Figure 3.4, which include the sawtooth-shaped, trapezoidal, and valley-shaped CGs.



(a) Sawtooth-shaped **(b) Trapezoidal** **(c) Valley-shaped**

Figure 3.4 Prescribed CGs in the second case study: design of both inlet concentrations and pressure differences.

3.3 Results and Discussion

In this section, we will first describe a process to verify the optimum design obtained by SBO with adaptive sampling. Then the details of the SBO design solutions for both case studies above will be presented. Specifically, in each case study, the model selection step is first undertaken to compare various combinations of the regression model

and the correlation model and select the best one for surrogate model construction. The adaptive sampling is then carried out to update the surrogate model and the response surface will be refined with infills for enhanced approximation, in which various infill criteria are also compared in terms of convergence rate. The performance of SBO with adaptive sampling will also be benchmarked with other relevant optimization methods, including SBO with random sampling and gradient-based optimization.

The procedure to verify the design obtained by SBO with adaptive sampling and benchmark its performance with the other optimization methods is illustrated in Figure 3.5. Starting with a minimum number of initial samples, SBO with adaptive sampling (within the black box) consisting of model selection and infill will be performed, and eventually yields an optimum design when convergence criterion is reached. For verification, the optimum design parameters are then entered to CFD simulation or PBCM simulation, producing a CG that is then compared against the prescribed CG, i.e., C_s. The process will be repeated for several prescribed CGs as presented in Section 3.3. To quantitatively characterize the performance of the proposed design method, two performance criteria are defined, including discrepancy of PBCM J_d^{PBCM} and discrepancy of CFD J_d^{CFD}

$$J_d^{PBCM} = \frac{\left\| C_s - g_{PBCM}(\mathbf{x}_{opt}) \right\|_2}{\left\| C_s \right\|_2}, \qquad J_d^{CFD} = \frac{\left\| C_s - g_{CFD}(\mathbf{x}_{opt}) \right\|_2}{\left\| C_s \right\|_2} \tag{3.3}$$

where C_s again is the prescribed CG; x_{opt} is the optimal design parameters; g_{PBCM} and g_{CFD} represent PBCM and CFD simulation, respectively, which takes x_{opt} as inputs and predicts the generated CGs. In this chapter, CGs produced by CFD simulation is treated as the ground truth. J_d^{PBCM} and J_d^{CFD} are used to inspect different aspects of the design process.

J_d^{PBCM} compares the prescribed CG and the CG computed by PBCM using optimal design parameters, and therefore, it characterizes not only design performance, but also feasibility of generating the prescribed CG. It should be noted that it is almost impossible to generate prescribed CGs in Figure 3.3 and Figure 3.4 exactly using μCGGs due to the physical limitation that CGs will be bent at all channel walls due to their impermeability to species transport (Y. Wang, Mukherjee, and Lin 2006). More broadly, J_d^{CFD} will also examine the discrepancy between PBCM and high-fidelity CFD arising from the assumptions used in PBCM. CFD simulation is performed to verify the optimal parameters and corresponding CGs obtained by various design methods above. The details regarding CFD simulation is presented in our prior work (Y. Wang, Mukherjee, and Lin 2006).

As discussed above SBO with adaptive sampling is also compared with two other design methods, i.e., SBO with random sampling and gradient-based optimization as shown by the gray dashed lines in Figure 3.5. In the former, the one-shot random sampling is used, and the surrogate model is only constructed once before the design process using simulation data at these randomly sampled parameters. In the latter, Matlab's built-in function, *fmincon*, a gradient-based optimization method to find the minimum of a constrained nonlinear multivariable function, is used to search for the optimum given prescribed CGs. The design performance of these methods, including accuracy and the numbers of evaluations, i.e., design costs are also compared.

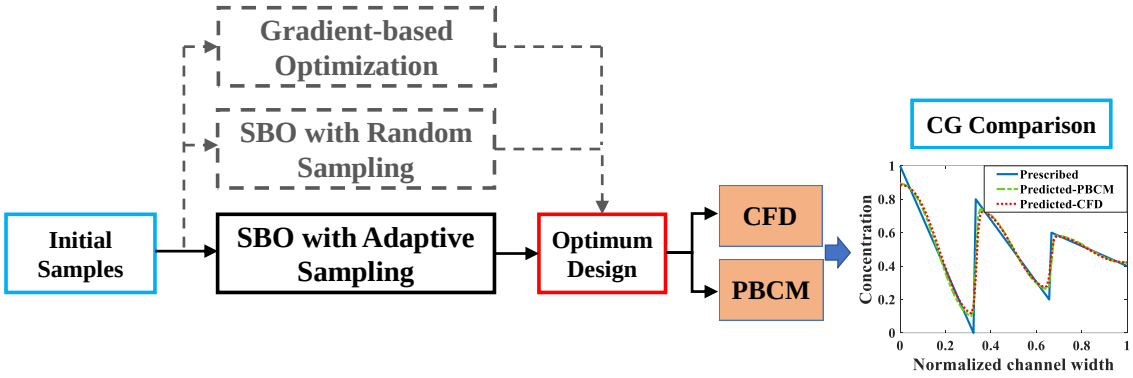


Figure 3.5 The procedure for design verification and performance benchmarking.

3.3.1 Case Study 1: Design of Inlet Concentrations

For each prescribed CG given in Figure 3.3, a comparative analysis is performed to compare the performance of various combinations of the regression model and the correlation model in order to construct a surrogate model of salient accuracy for SBO design. 28 training data, the minimum number of required samples to build a surrogate model in the 6-dimensional design space is adopted (Z.-H. Han and Zhang 2012). For each prescribed CG, a total of 15 surrogate models are constructed by the full-factorial combination of three regression models and five correlation models. The regression models under consideration include the 0^{th}, 1^{st}, and 2^{nd} order polynomial. The correlation models include spline, Gauss, exponential, linear, and spherical, and their mathematical expressions are described in (Forrester, Sóbester, and Keane 2008). Then surrogate model-predicted values are compared with true values of 9 validation data, and the relative error between them is defined as

$$\varepsilon = \frac{1}{n}\sum_{1}^{n}\left|\frac{J_d - J_d{}'}{J_d}\right| \times 100\% \tag{3.4}$$

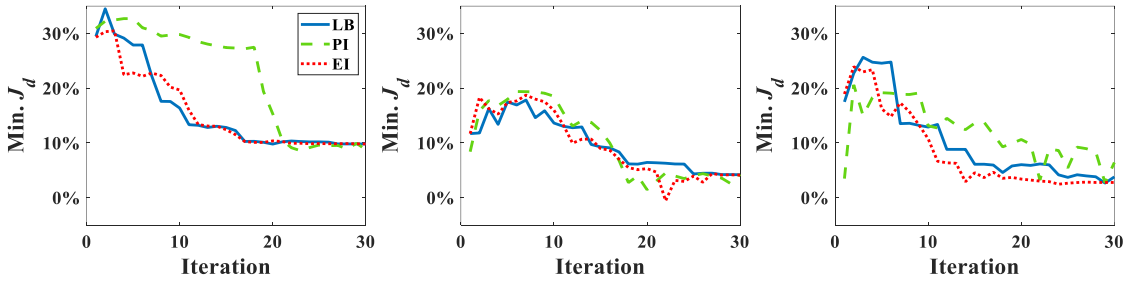
where J_d is the true value of the discrepancy between the prescribed CG C_s and the CG generated at the validation samples C_o, while J_d' is the discrepancy predicted by the surrogate model, n is the number of validation samples and is 9 in this case study. Table 3.1 lists the relative error for various combinations of the regression model and the correlation model, according to which the most accurate one is selected for adaptive sampling and SBO.

Table 3.1 Relative percentage errors of surrogate models built by different combinations of regression and correlation models in the case study: design of inlet concentrations.

Regression Model	Correlation Model	Relative (Percentage) Errors ε		
		Sawtooth-shaped	Trapezoidal	Linear
Zero Order Polynomial	Spline	20.60%	22.56%	20.69%
	Gauss	17.63%	17.45%	12.78%
	Exp	19.59%	21.18%	13.07%
	Linear	20.60%	22.56%	20.69%
	Spherical	20.60%	22.56%	20.69%
First Order Polynomial	Spline	15.20%	10.52%	10.10%
	Gauss	13.25%	9.48%	8.89%
	Exp	15.20%	10.31%	10.59%
	Linear	15.20%	10.52%	10.10%
	Spherical	15.20%	10.52%	10.10%
Second Order Polynomial	Spline	16.14%	26.08%	9.60%
	Gauss	16.14%	26.08%	9.60%
	Exp	16.14%	26.08%	9.60%
	Linear	16.14%	26.08%	9.60%
	Spherical	16.14%	26.08%	9.60%

It was found that the first-order polynomial regression model combined with the Gauss correlation model reveals the smallest relative error ε for all three prescribed CGs, and thus, is selected for SBO with adaptive sampling. In this case study, 30 adaptive samples/infill (corresponding to 58 in total) are allowed to find the optimum design for each prescribed CG. For comparison, three infill techniques above are applied separately. Figure 3.6 shows the convergence of Min. J_d of the surrogate model for different infill

strategies for each prescribed CG. Multiple runs of the same optimization configuration are repeated for each prescribed CG, and all converge to the global optimum, and therefore, the same results are not duplicated here for the sake of conciseness. For sawtooth-shaped CGs, LB exhibits a faster convergence rate compared to the other two infill strategies. For the trapezoidal CG, three infill strategies have a similar convergence rate. For the linear CG, EI converges to a better solution, i.e., lower J_d at a faster rate.



(a) Sawtooth-shaped **(b) Trapezoidal** **(c) Linear**

Figure 3.6 Convergence of Min. J_d using different infill strategies for prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) linear in the case study: design of inlet concentrations.

Table 3.2 Comparison of different infill strategies and prescribed CGs in terms of J_d of optimum design for the case study: design of inlet concentrations.

Infill	J_d		
	Sawtooth-shaped	Trapezoidal	Linear
LB	9.83%	4.07%	2.97%
PI	10.24%	4.54%	7.04%
EI	9.91%	4.20%	2.60%

Table 3.2 lists the numerical values of J_d of optimum design in all cases above. The J_d achieved for each infill strategy represents the closeness between the CG generated by the candidate design and the prescribed CG. A better infill strategy constructs a more accurate surrogate model near the region around the optimum to capture the true response

surface of J_d there, but not necessarily the entire design space. This is because the goal of our infill is to improve the accuracy of the optimal design, rather than building a global surrogate model to represent the input and output relationship across the entire domain. The results in Table 3.2 confirm that LB outperforms the other two for the sawtooth-shaped CG, all the three are equivalent for the trapezoidal CG and LB is used for the analysis below, and EI is the best the linear CG.

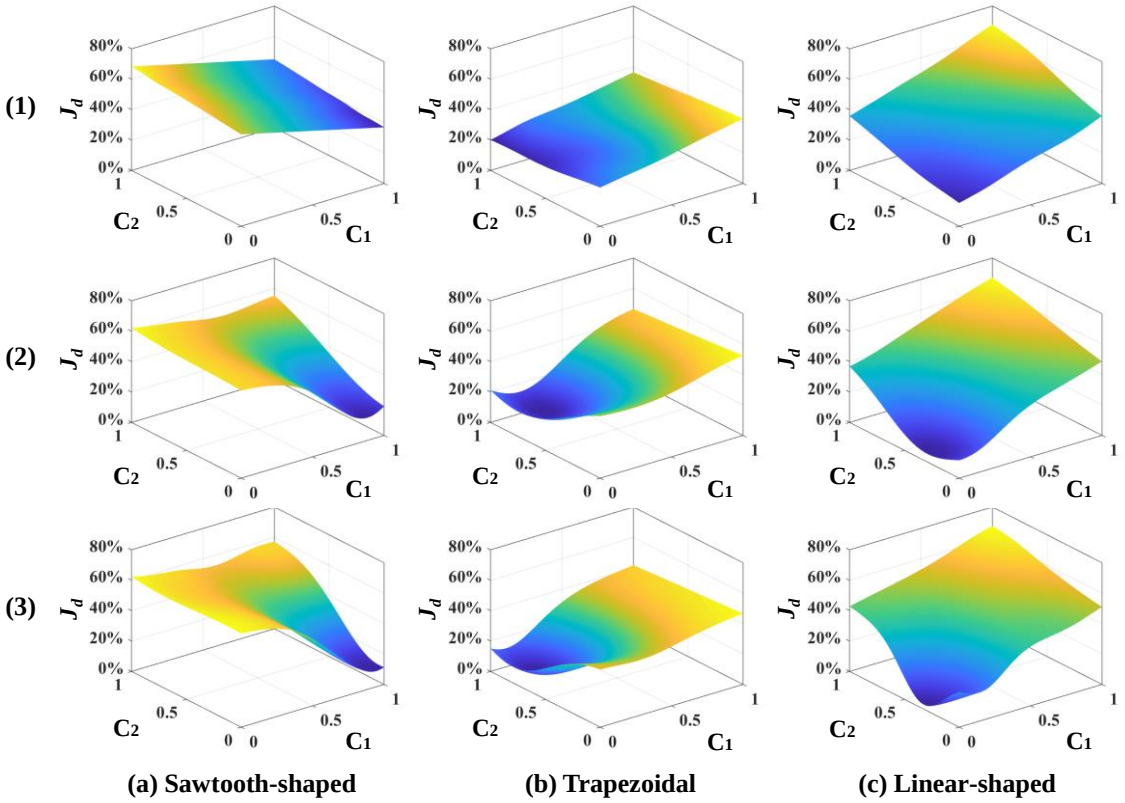


Figure 3.7 Response surface plots of the surrogate models in 3D that vary with c_1 and c_2 while keeping the other design variables constant for different numbers of sample infills: (1) 0, (2) 15, and (3) 30 in the case study: design of inlet concentrations.

Figure 3.7 shows the response surface of J_d predicted by the surrogate model as more samples are added by the best infill strategy identified above for each prescribed CG.

To facilitate visualization, the surface is portrayed in 3D that only varies with c_1 and c_2 while keeping the other design variables constant. 0, 15, and 30 sample infills are employed in the surface plot from the top to the bottom, respectively. It clearly shows that for each prescribed CG, the minimum value of the surrogate model becomes smaller and converges to a single point as more infills are added. Besides, the infill points are mostly distributed within the region close to the minimum, and impact the response surface shape there, which improves the accuracy of the optimal design solution and confirms that adaptive sampling effectively accelerates the process of search.

CFD simulation results and the comparison between the prescribed and predicted CGs in terms of the normalized chemical concentration are shown in Figure 3.8. The concentration contour near the Ψ-shaped junction is displayed in the top row, and the CGs across the channel width are observed at the detector, which is located 400 μm downstream of the Ψ-shaped junction. The PBCM- and CFD-predicted CGs match well with prescribed CGs, which as illustrated in Figure 3.8, are obtained by supplying the optimal design variables to PBCM and CFD simulations. The excellent agreement of PBCM- and CFD-predicted CGs with respect to prescribed CGs verifies the accuracy of SBO with adaptive sampling for the μCGG design. However, minor differences are also observed at the stream interface and the side walls, which can be attributed to the fact that the prescribed CGs are created by concatenating three linear profiles while in actual μCGGs, CGs will be bent near all channel walls resulting from their impermeability to chemical species. In other words, the μCGG is not able to generate exactly the same prescribed CGs if the latter are artificial and do not fully match the solution of the underlying species transport equation. In addition, there is also an excellent match of the CG results predicted by PBCM and CFD,

which implies that PBCM although with assumptions to allow analytical solution, is an accurate approximation of computationally demanding CFD, and can be used in place of the latter for design.

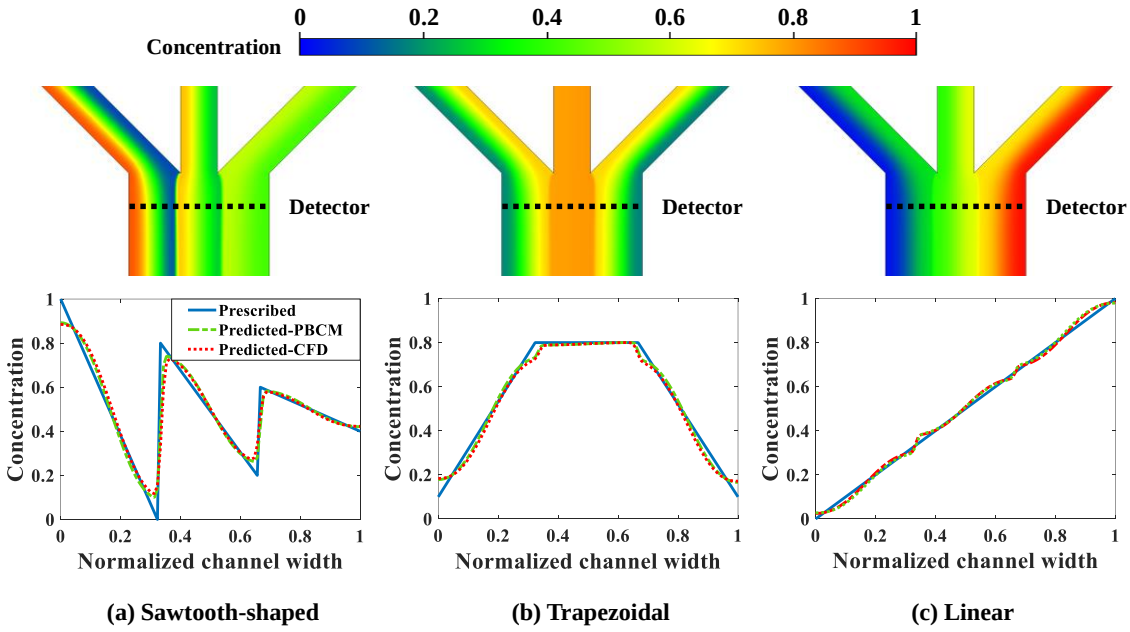
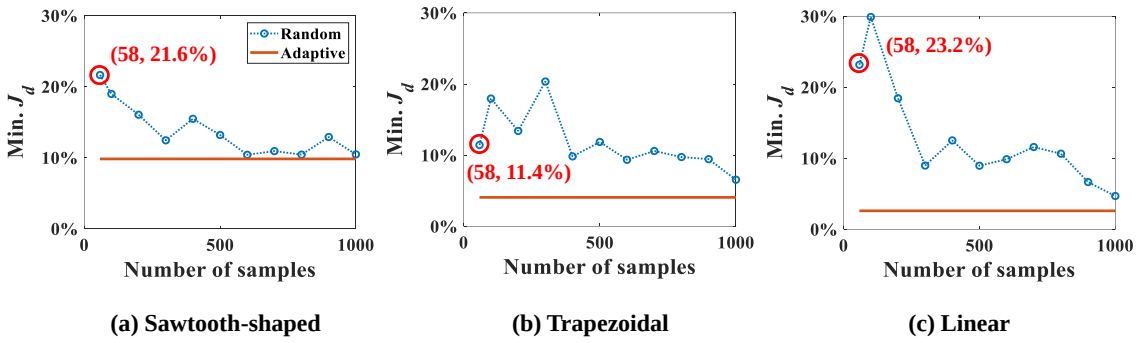


Figure 3.8 CFD contour plots and predicted CGs relative to the prescribed CG for the case study: design of inlet concentrations.

The optimum design found by SBO with adaptive sampling is also compared with two other design methods, i.e., SBO with random sampling and gradient-based optimization (enclosed in the dashed boxes in gray in Figure 3.5), for all three prescribed CGs. In the former, one surrogate model is constructed using training data produced at parameters selected by one-shot random sampling before the design, and it is then used in the design process without infill or model update. Figure 3.9 shows Min. J_d found by SBO with random sampling that uses the different number of samples, and compares it with the reference line in orange, viz., adaptive sampling results using 58 samples in total. The red circle represents the result of random sampling when 58 samples in total are used. It clearly

reveals that the optimal solution determined by the one-shot random sampling and corresponding surrogate model exhibit a much larger value of Min J_d for all cases. The adaptive sampling at least improves the accuracy of random sampling by two times, that is, Min. J_d drops from 21.6% to 9.83% in the sawtooth-shaped CG, from 11.4% to 4.07% in the trapezoidal CG, and from 23.2% to 2.60% in the linear CG. Even if the number of randomly selected samples is increased to 1000, the accuracy of random sampling-based design cannot reach that by adaptive sampling. Besides, as we can see from the figures, the oscillation present in the curve of the random sampling is due to the insufficient number of samples. Therefore, given a fixed simulation budget, adaptive sampling is more computationally efficient and desired for global optimum search.



(a) Sawtooth-shaped **(b) Trapezoidal** **(c) Linear**

Figure 3.9 Comparison of results between SBO with random sampling and adaptive sampling for the case study: design of Inlet Concentrations.

Next, SBO with adaptive sampling is compared with the gradient-based optimization method in terms of the total number of PBCM simulations, and the latter uses Matlab's built-in function, *fmincon*. It is well known that the number of model evaluations (viz., PBCM simulations herein) in gradient-based optimization heavily depends on the selection of initial start points. Therefore, ten runs with different initial points, which are

selected from initial samples in SBO with adaptive sampling, are undertaken and examined for the gradient-based optimization. The number of PBCM evaluations required to reach the same accuracy as SBO with adaptive sampling in the ten runs is averaged, and the average number is then compared with that of the proposed method. Table 3.3 shows that for all three prescribed CGs, SBO with adaptive sampling uses a smaller number of PBCM evaluations/simulations (~30 less on the average) for this case study involving 6 design variables.

Table 3.3 Comparison of the number of PBCM evaluation/simulation between SBO with adaptive sampling and gradient-based optimization for the case study: design of Inlet Concentrations.

Method	Sawtooth-shaped	Trapezoidal	Linear
Gradient-based	100	87	109
SBO	58	58	58

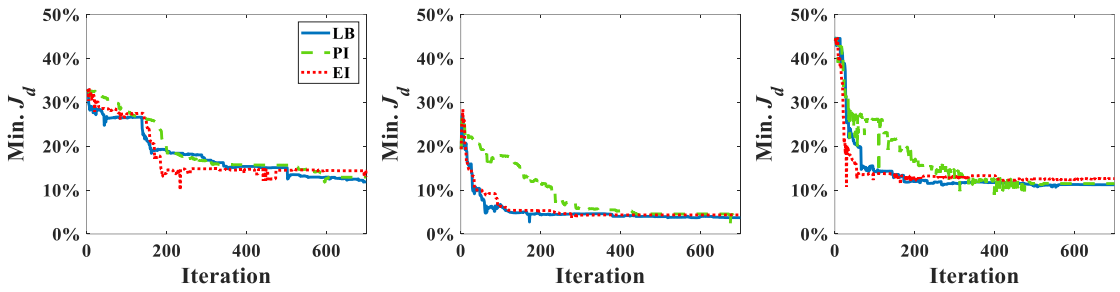
3.3.2 Case Study 2: Design of Inlet Concentrations and Pressure Differences

Next, we extend the study to the 9-dimensional design space encompassing six inlet concentrations of chemicals and three pressure differences in all the Y-shaped mixers. Similarly, the best combination of the regression model and the correlation model is first selected through a comparative analysis. Each combination uses 55 samples in the training data, which is the minimum number of samples required to build a surrogate model in the 9-dimensional design space. Subsequently, 17 validation samples are utilized to evaluate performance of the 15 combinations of the regression model and the correlation model for all three prescribed CGs in Figure 3.4. The relative percentage errors are listed in Table 3.4, which clearly indicates that the first-order polynomial regression model and the Gauss correlation model yield the smallest relative errors ε and is the best one for all three cases.

Table 3.4 Relative percentage errors of surrogate models built by different combinations of regression and correlation models in the case study: design of inlet concentrations and pressure differences.

Regression Model	Correlation Model	Relative (Percentage) Error ε		
		Sawtooth-shaped	Trapezoidal	Valley-shaped
Zero Order Polynomial	Spline	17.87%	28.90%	18.54%
	Gauss	17.57%	25.21%	17.57%
	Exp	17.87%	28.90%	18.54%
	Linear	17.87%	28.90%	18.54%
	Spherical	17.87%	28.90%	18.54%
First Order Polynomial	Spline	16.55%	20.46%	11.64%
	Gauss	16.48%	20.44%	11.10%
	Exp	16.55%	20.46%	11.64%
	Linear	16.55%	20.46%	11.64%
	Spherical	16.55%	20.46%	11.64%
Second Order Polynomial	Spline	83.02%	56.99%	83.66%
	Gauss	83.02%	56.99%	83.66%
	Exp	83.02%	56.99%	83.66%
	Linear	83.02%	56.99%	83.66%
	Spherical	83.02%	56.99%	83.66%

With the best surrogate model structure selected, SBO design with adaptive sampling is then carried out subject to a budget of 700 infill samples that are selected by three different infill strategies. Figure 3.10 portrays convergence curves of Min. J_d of the surrogate model using the three infill strategies for each prescribed CG. LB outperforms the other two in terms of the convergence rate, and constructs more accurate surrogate models that find inlet concentrations and pressure differences. This is quantitatively confirmed by J_d of optimum design listed in Table 3.5, which clearly shows that LB achieves the lowest J_d for all three prescribed CGs, yielding better designs than PI and EI.



(a) Sawtooth-shaped **(b) Trapezoidal** **(c) Valley-shaped**

Figure 3.10 Convergence of Min. J_d of surrogate model using different infill strategies for prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped for the case study: design of inlet concentrations and pressure differences.

Table 3.5 Comparison of different infill strategies and prescribed CGs in terms of J_d of optimum design for the case study: design of inlet concentrations and pressure differences.

Infill	J_d		
	Sawtooth-shaped	Trapezoidal	Valley-shaped
LB	11.90%	3.75%	11.23%
PI	12.97%	4.36%	11.57%
EI	14.61%	4.43%	12.77%

The response surface of J_d predicted by the surrogate model with more samples added by the LB infill is shown in Figure 3.11. Similarly, for the sake of visualization, only c_1 and c_2 vary while the other design variables are held constant. The surface plots from the top to the bottom are generated by surrogate models with 0, 300, and 700 infills. It shows that without infills, the response surfaces appear almost linear because of the use of the first-order regression model and extremely insufficient training data. As more infill points are added, e.g., 300 infills, the nonlinearity of the response surface for each prescribed CG is observed, and Min. J_d becomes evident. The profiles of the response surfaces only change slightly at 700 infills along with the converged solution of Min. J_d. Again, this confirms that the surrogate model can be improved and the optimal design in

9-dimensional space can be found in a reliable manner given adequate infill points. More importantly, adaptive sampling based on the LB infill strategy successfully assigns most of the infill points within the region close to optimum (not shown to avoid data clustering and facilitate visualization) that provides more topological information to speed up the search process and improve the solution accuracy.

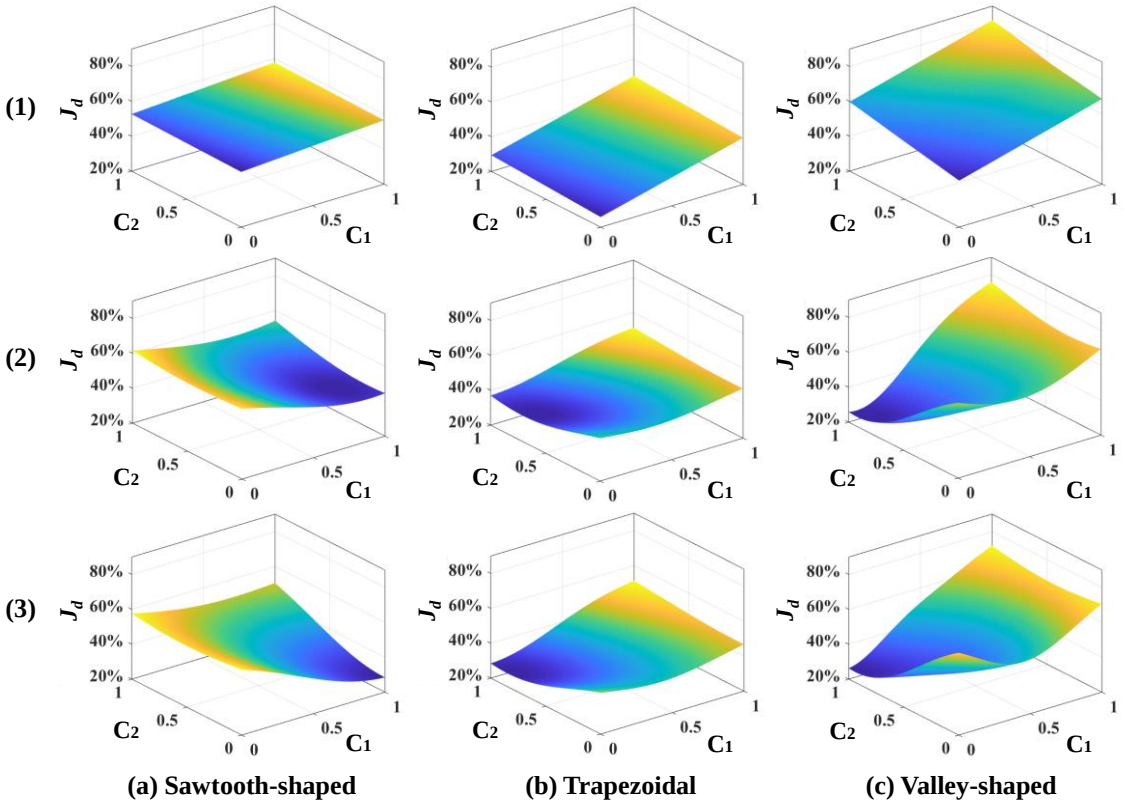


Figure 3.11 Response surface plots of the surrogate models in 3D that vary with c_1 and c_2 while keeping the other design variables constant for different numbers of sample infills: (1) 0, (2) 300, and (3) 700 in the case study: design of inlet concentrations and pressure differences.

Figure 3.12 shows the CFD contour plots of the normalized chemical concentrations and the comparison between the prescribed and predicted CGs extracted at the detector location. Table 3.6 lists the numerical values of J_ds of CGs predicted by PBCM and CFD simulation using the optimal design parameters found above by SBO with

adaptive sampling. There are several points of note. First, the PBCM-predicted CGs match well with prescribed CGs in all cases, although their shapes are more complex in this case study with 9 design variables. The discrepancies between prescribed CGs and PBCM-predicted CGs are mostly due to bending of the concentration distributions at the side walls that are impermeable to chemical transport. Second, CFD-predicted CGs exhibit noticeable discrepancy from the prescribed and PBCM-predicted CGs in the sawtooth-shaped and trapezoidal CGs, and therefore, the values of J_d by CFD is appreciably higher than that of PBCM in both CGs. This is caused by appreciable transverse flow immediately downstream the Ψ-shaped junction before the flow is fully developed, and the presence of the detector within the flow entrance region of the main output channel, as revealed by the concentration contours in Figure 3.12 (a) and (b). However, as discussed above, our PBCM is not able to take into account such an effect because of its modeling assumptions. To confirm the interpretation, Figure 3.13 illustrates the comparison between PBCM- and CFD-predicted CGs when the detector is located further downstream (2000 μm from the Ψ-shaped junction), and we can see that both match very well. It is also noticed that for the valley-shaped CG, the transverse flow is relatively weak due to the pressure symmetry in the streamwise direction, which again is apparent in CFD contour plot of Figure 3.12 (c). Therefore, PBCM- and CFD-predicted CGs are almost identical with negligible differences.

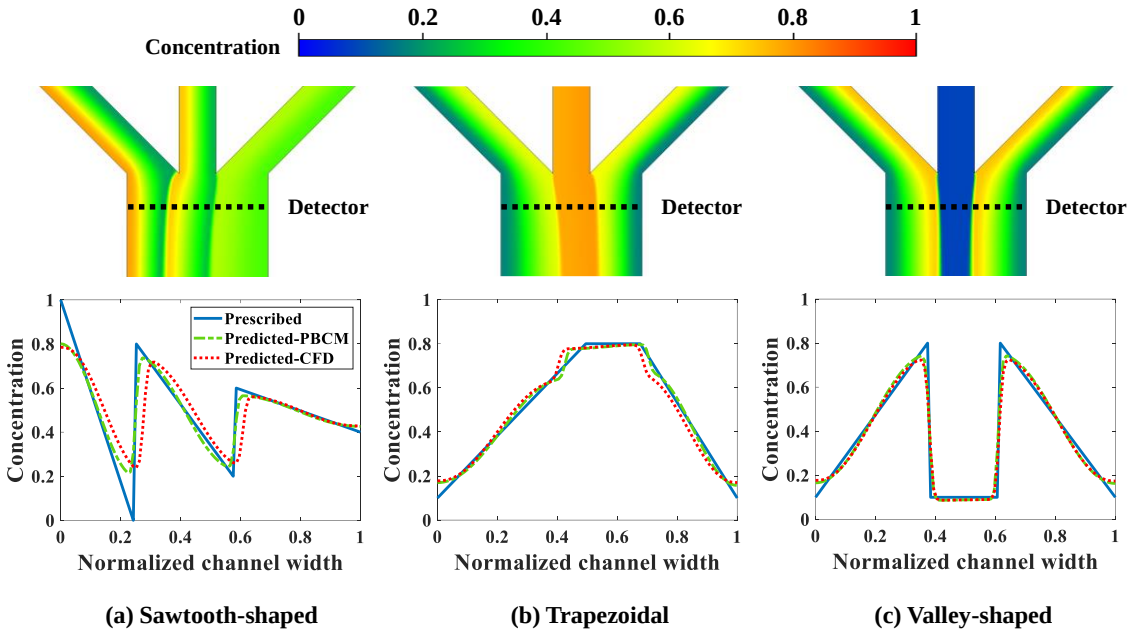


Figure 3.12 CFD contour plots and predicted CGs relative to the prescribed CG for the case study: design of inlet concentrations and pressure differences.

Table 3.6 J_ds of CGs predicted by PBCM and CFD using the optimal designs found by SBO with adaptive sampling.

	Sawtooth-shaped	Trapezoidal	Valley-shaped
PBCM	11.90%	3.75%	11.23%
CFD	22.98%	6.42%	12.34%

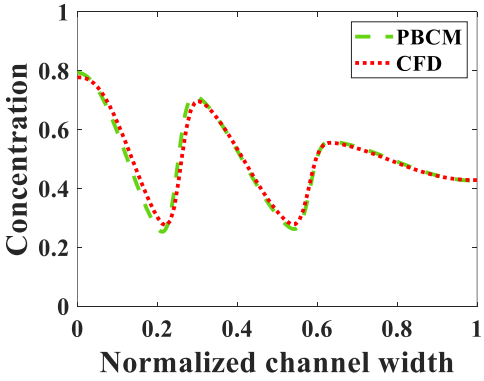
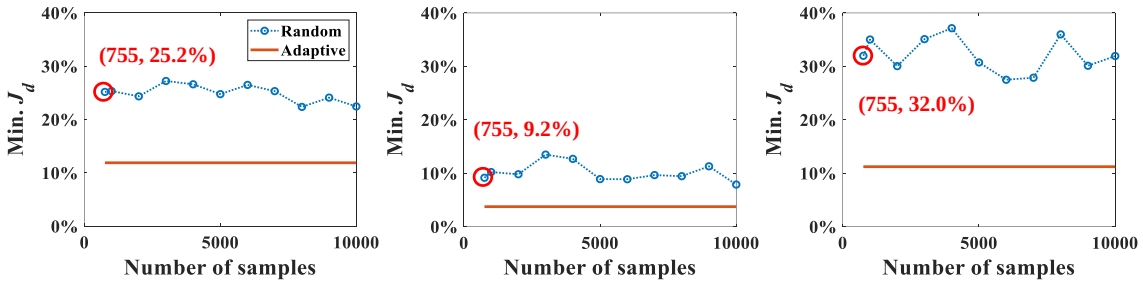


Figure 3.13 Comparison between PBCM and CFD simulation at fully developed region for the sawtooth-shaped CG.

Min. J_d for the different numbers of samples obtained by SBO with random sampling is shown in Figure 3.14, and compared with shows the reference line in orange,

viz., adaptive sampling results using 755 samples in total. Again, Min. J_d achieved by the random sampling is at least 2 times larger than that by adaptive sampling with 755 samples in total. Even if the number of samples and evaluations is 13 times larger than adaptive sampling, it still cannot reach the same level of accuracy. This is because even 10,000 samples for 9 dimensions are still too small for constructing an accurate surrogate model in the entire design space, in particular, around the region of optimum. It turns out that adaptive sampling effectively accelerates the process of searching minimum. Besides, the uniform distribution of samples contributes to the oscillation of the curve as the correlation matrix in Kriging can vary dramatically and randomly. However, the general trend of random sampling error is to decrease as more samples are added.



(a) Sawtooth-shaped **(b) Trapezoidal** **(c) Valley-shaped**

Figure 3.14 Results of SBO with random sampling for the design of inlet concentration and pressure difference.

Table 3.7 shows the number of evaluations required by the gradient-based optimization to reach the optimum solution at the same level of accuracy as SBO with adaptive sampling. Again, 10 different start points are selected from the initial samples used in SBO design to initiate 10 gradient-based optimization runs. It shows that 50% of the runs fail to converge to the optimum solution for the sawtooth-shaped CG, 10% failure

for the trapezoidal, and 50% for the valley-shaped. It seems caused by being trapped at local optima because Min. J_ds achieved in these failed runs are notably higher than those at the global optimum. This implies that although gradient-based optimization can be more computationally efficient for high-dimensional design problems, there is a risk of missing the global optimum, in particular, for generating complex CGs. However, for the trapezoidal CG, gradient-based optimization outperforms SBO with adaptive sampling in efficiency for 9 runs, which may be attributed to the simpler topology of its response surface. Generally speaking, because of its exploratory nature, SBO with adaptive sampling is a more feasible and reliable method to search for the global optimum, while the gradient-based approach is more computationally efficient and requires fewer PBCM evaluation in high-dimensional design space if not trapped at the local optimum. The large variation in the number of PBCM evaluations further reveals that the gradient-based optimization is indeed dependent on the initial start point.

Table 3.7 Evaluation comparisons between the SBO with adaptive sampling and the gradient-based optimization methods.

Run No.	Sawtooth-shaped		Trapezoidal		Valley-shaped	
	Number of Evaluation	Min. J_d	Number of Evaluation	Min. J_d	Number of Evaluation	Min. J_d
1	247	11.42%	448	3.74%	Fail	22.37%
2	280	11.17%	207	3.73%	Fail	22.36%
3	Fail	31.11%	246	3.70%	307	10.83%
4	484	11.12%	255	3.69%	260	10.93%
5	Fail	31.11%	204	3.75%	Fail	21.99%
6	279	10.63%	195	3.72%	229	11.18%
7	297	11.06%	219	3.74%	Fail	21.93%
8	Fail	31.11%	195	3.73%	255	11.15%
9	Fail	31.11%	216	3.71%	351	10.96%
10	Fail	31.11%	Fail	4.62%	Fail	22.28%

3.4 Summary

In this chapter, a new method based on SBO with adaptive sampling is developed for efficient and reliable design of μCGGs. The key rationale of the proposed method is to construct the surrogate model, i.e., Kriging, using physics-based simulation data, update the model using incrementally and adaptively added data, viz., infill, and then utilize continuously enriched topological information provided by the surrogate model to guide the search of global optimum. New aspects of the proposed research include: First, the feasibility of applying SBO with adaptive sampling to complex μCGG design is systematically examined. Second, the PBCM of μCGGs in the closed-form is employed to generate data for surrogate model construction to further reduce the computational cost. Third, a comparative analysis is performed to identify the best combinations of the regression model and the correlation model, and determine the infill strategies for improved surrogate modeling and design performance. Last, the use of pressure differences rather than the native inlet pressures as the design variables to eliminate the backflow issue.

Two case studies are undertaken on the partial mixing- and species transport-governed triple Y-shaped μCGG to evaluate design performance of the proposed method. Key technical findings are obtained, including

1. Our comparative analysis indicates that combining the first-order polynomial regression model and the Gauss correlation model in Kriging yields the highest surrogate model accuracy.
2. In general, three infill strategies all allow the design to converge to the global optimum, while on average LB exhibits faster convergence rate than EI and PI.
3. All CGs predicted by PBCM using the optimal design parameters match prescribed

CGs, which verifies feasibility, robustness, and accuracy of the proposed method.

4. Both PBCM-predicted and CFD-predicted CGs match very well in the first case study, which validates the accurate design of SBO with adaptive sampling, while an appreciable difference (average J_d difference 4.95%) between them is observed in the second case study. It is attributed to asymmetric flow rates through Y-shaped mixers in the second case study that give rise to significant transverse flow in the entrance region, the effect of which on species transport can be captured by CFD but not PBCM.
5. The proposed method is at least two times more accurate than SBO with random sampling for all prescribed CGs, which confirms that adaptive sampling-based infill is necessary for SBO design of complex CGs.
6. The gradient-based optimization method requires at least 30 more evaluations compared to the method of SBO with adaptive sampling in the first case study. In the second case, approximately 1/3 of the runs of gradient-based optimization fail to find the global optimal solution, although on average it uses less simulation than SBO with adaptive sampling. In short, SBO with adaptive sampling is preferred as a robust method to find the global optimal design.

CHAPTER 4 MULTI-FIDELITY SURROGATE-BASED OPTIMIZATION FOR MICROFLUIDIC CONCENTRATION GRADIENT GENERATOR DESIGN[2]

[2] Yang, Haizhou, Seong Hyeon Hong, Yu Qian, and Yi Wang. Submitted to *Engineering Computations*, 01/20/2022

In the previous chapter, we developed an SBO technique (H. Yang et al. 2020), in which the time-consuming CFD simulations were replaced with PBCM simulations. More importantly, SBO significantly reduced the number of total simulations required to reach the global optimum. In a complementary work, we also developed a deep learning-based inverse design method that used the PBCM to generate a large number of data samples by parallel computing, and the samples are then used to train artificial neural network models to instantly compute the operational parameters to match the desired CGs (S. H. Hong, Yang, and Wang 2020). However, the PBCM model used in both methods above was based on assumptions that could be violated for complex μCGGs, leading to inaccurate design. First, only the one-dimensional flow (averaged along the channel's cross-section) in the longitudinal direction was considered in the model, and the velocity nonuniformity across the cross-section was neglected. Second, the developing flow at the entrance of a junction was not taken into account, which could result in appreciable errors when the flows from multiple branch channels entering the merging junction are not equal (i.e., asymmetric) giving rise to significant transverse flow.

To address this limitation, a novel approach for the μCGG design that combines the HF CFD model and the LF PBCM model in the optimization is highly desirable, as both phenomena above can be effectively captured by the CFD simulation. This will not only improve the accuracy, but also retain the computational efficiency. Therefore, an MFSBO framework is presented in this chapter. The contributions of the present effort include: (1) the MFSM is used to combine PBCM and CFD, two models with distinctly different natures, fidelities, accuracies, and computational efficiencies. The underlying principle of MFSM is to improve the prediction accuracy and modeling efficiency by establishing the

correlation between HF and LF data. In other words, the objective function of the HF model can be learned more quickly by adding the cheap LF function evaluations (Marques et al. 2019; Fernández-Godino et al. 2016; Peherstorfer, Willcox, and Gunzburger 2018; Forrester, Sóbester, and Keane 2007). With such a consideration, we propose to use the PBCM as the LF model, which is based on the system decomposition method and the analytical solution of the convection-diffusion equation for species transport. Note that the PBCM takes the aforementioned assumptions to allow the closed-form solution and is extremely fast to evaluate, but less accurate (Y. Wang, Mukherjee, and Lin 2006). On the other hand, CFD relies on the direct discretization of the governing equation with fewer assumptions compared to PBCM. Although more time-consuming, it is more accurate to evaluate involved transport behavior in complex μCGGs, and hence, serving as the HF model in this work. Our MFSM combining such two complementary sources of data is able to generate accurate surrogate model and optimization results with salient computational efficiency (Kuya et al. 2011; Fernández-Godino et al. 2016); (2) Due to incorporation of HF CFD data, MFSBO allows the automated design of CGs with asymmetric flows, which is otherwise not achievable through SBO only with PBCM (H. Yang et al. 2020); (3) Because of its negligible computational cost, the parallel multi-fidelity adaptive infill strategy, infilling one HF and multiple LF samples, is utilized to determine new samples for sequential evaluation to gradually improve model accuracy, especially near the region of the global optimum. Parallel infill relies on multiple infill criteria and principles from different statistical perspectives to balance sample exploration and exploitation, and hence, is more robust and computationally efficient; and (4) to the best of our knowledge, the

present research represents an initial effort to investigate the feasibility of applying MFSBO to μCGG design.

4.1 Methodology

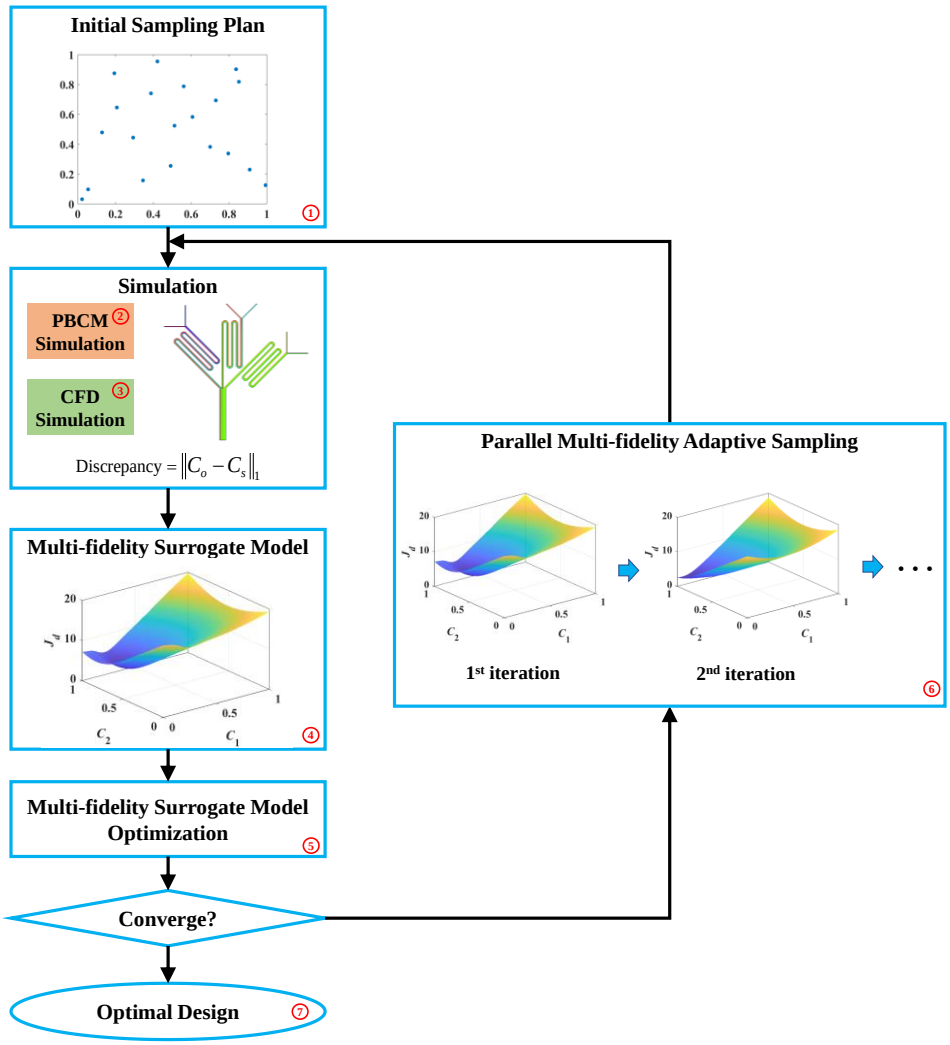


Figure 4.1 Flowchart of the MFSBO for μCGG.

Figure 4.1 illustrates the MFSBO process for finding the optimal operational parameters of the μCGG at its inlets that allow generating prescribed CGs. It includes initial sampling, MFSM, surrogate model optimization, and parallel multi-fidelity adaptive

sampling. The detailed process is given as follows: First, the LHS (labeled "1" in Figure 4.1) is applied for generating initial samples in multi-dimensional design parameter space (Etikan 2016; Ferrari et al. 2019), which in the present study are chemical species concentrations at the inlet reservoirs and pressures applied (or flow rates). Second, the PBCM and CFD simulations (labeled "2" and "3" in Figure 4.1), representing the LF model and the HF model, respectively, are performed at the sample points/locations generated in the previous step. The discrepancy, J_d, which is defined as the L_1 norm of the difference between the CG generated at the sampled point (C_o) and the user-prescribed CG (C_s), is computed to directly measure the difference between them, and is used as the output of MFSM (and also the objective function for optimization). Next, MFSM (labeled "4" in Figure 4.1) is constructed to capture the mapping relationship between the input design variables and their corresponding outputs (discrepancies J_ds) using the HF and LF data at the sample locations. Specifically, the Cokriging method, an MFSM technique based on stochastic process modeling, is adopted in this study. The constructed MFSM is then used for global optimization to search for the best design corresponding to the minimum discrepancy J_d (labeled "5" in Figure 4.1). Since the initial samples are not enough to build an accurate MFSM, parallel multi-fidelity adaptive sampling (labeled "6" in Figure 4.1) will be performed to infill additional samples in the design space, which are then simulated again using the PBCM and CFD to compute the corresponding discrepancies at these new sample locations and update the MFSM following the same procedure. Two different infill strategies are applied and compared in terms of convergence rate, computational cost, and design accuracy in the present effort. The first strategy is infilling one HF and one LF samples at the same location that could be determined by one of the following infill criteria:

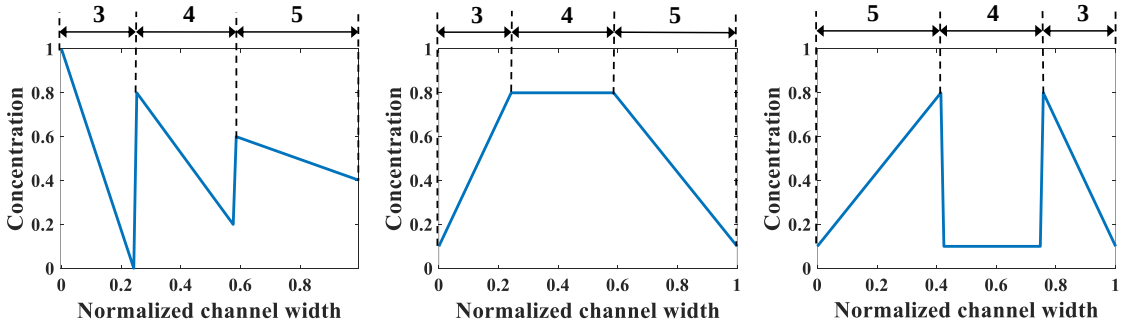
MP, LB, PI, and EI. The other strategy infills one HF sample generated by one of the infill criteria above, and four LF samples determined by all four criteria listed above. The whole process is repeated until the minimum of the MFSM converges or the maximum number of global optimization iterations is reached. Finally, the HF CFD simulation at the minimum of the MFSM in the last iteration is performed and compared with all existing data for verification. The set of operational parameters with the smallest discrepancy is selected as the optimal design (labeled "7" in Figure 4.1).

4.2 Problem Formulation and Performance Verification

4.2.1 Problem Formulation

The MFSBO design optimization for μCGGs is essentially to determine operational parameters that allow μCGGs to generate a CG to match a prescribed CG. Design variables are the chemical concentrations at the inlets (c_1, c_2, …, c_6) and the pressure differences across the mixing channels (Δp_1, Δp_2, Δp_3). The reason of optimizing the pressure differences rather than the pressures at the inlets is to avoid the backflow issue (H. Yang et al. 2020). Significantly unbalanced pressures at the inlets, i.e., pressures within different branches have large differences, may cause the reversed flow out of the inlets. Therefore, replacing the pressures at inlets with the pressure differences across the mixing channels guarantees positive pressures in the inlet and mixing channels, and ensures the solution flows out via the outlet. The discrepancy J_d (H. Yang et al. 2020), which is defined as the L_1-norm of the difference between the generated CG (C_o) at the sampled design variables and the user-prescribed CG (C_s), serves as the objective function of this μCGG design problem. In this chapter, three different prescribed CGs are considered, which are shown

in Figure 4.2, and include sawtooth-shaped, trapezoidal, and valley-shaped. Note that, all the prescribed CGs are asymmetric, i.e., the width of each constituent segment is not equal, which introduces transverse flow and associated species transport at channel junctions that cannot be captured by PBCM, but can be by the three-dimensional CFD simulation.



(a) Sawtooth-shaped **(b) Trapezoidal** **(c) Valley-shaped**

Figure 4.2 Prescribed CGs including asymmetric constituent components of CGs.

To verify the proposed MFSBO method, three μCGG design cases with increasing levels of complexities are devised and investigated, although the same prescribed CGs above are used in these studies. In the first case study, only the pressure differences across three mixing channels are treated as design variables, i.e., $x = [\Delta p_1, \Delta p_2, \Delta p_3]$, and the chemical concentration at inlets are kept as constant, which makes a three-dimensional optimization problem. In the second case study, only normalized chemical concentrations at six inlets are considered as design variables, i.e., $x = [c_1, c_2, \ldots, c_6]$ with c_i being a scalar, and the pressure differences across three mixing channels are kept constant, yielding a six-dimensional optimization problem. In the third case study, the design variables include both normalized chemical concentrations at the six inlets and the pressure differences across the three mixing channels i.e., $x = [\Delta p_1, \Delta p_2, \Delta p_3, c_1, c_2, \ldots, c_6]$, viz., a nine-dimensional optimization problem.

4.2.2 Performance Verification

Our proposed MFSBO method will be compared with the SBO method, which performs optimization only using HF simulations. The SBO method has been proven effective for generating faster, more stable, and accurate designs than gradient-based optimization methods in our previous work (H. Yang et al. 2020). In the present MFSBO work, two different multi-fidelity adaptive sampling strategies are evaluated and compared. 'MFSBO-1' infills one HF and one LF samples at the same location for each iteration, which is determined by the LB infill criterion. LB exhibits a faster convergence rate for μCGG design in our previous research (H. Yang et al. 2020). 'MFSBO-4' adds one HF and four LF in each iteration. The HF infill location is determined by the LB infill criterion, while four LF samples are determined by the four different infill criteria presented in the previous section. For SBO, one HF sample is determined by the LB infill criterion for each iteration.

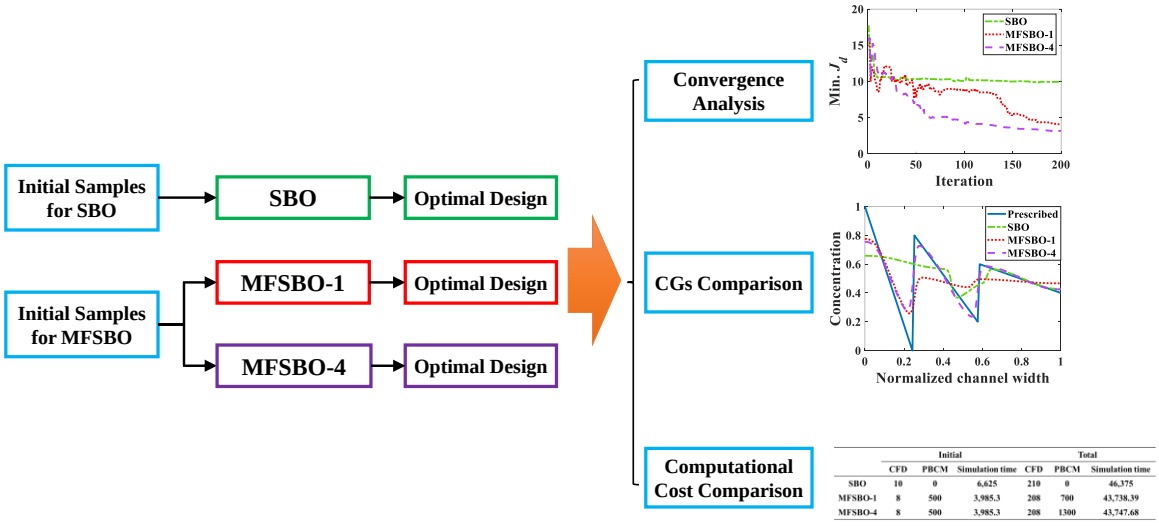


Figure 4.3 The procedure for design verification and performance benchmarking.

The details of the procedure for design verification and performance benchmarking are illustrated in Figure 4.3. First, the initial samples for SBO and MFSBO are generated LHS separately. Note that, the initial computational cost of MFSBO will be less than SBO's. Second, SBO, MFSBO-1, and MFSBO-4 are performed respectively to update the surrogate model following the optimization iteration process discussed above, and yield the optimal designs. Last, the optimal designs from these three methods are compared in terms of the convergence, accuracy of optimal design, and computational cost.

4.3 Results and Discussion

4.3.1 Case Study 1: Design of Pressure Differences

For each prescribed CG in Figure 4.2, initial samples are generated first for SBO and MFSBO, respectively. Five HF samples are generated for SBO, while 3 HF and 100 LF samples are produced within the design space for both MFSBO-1 and MFSBO-4. The computational cost of the initial samples of MFSBO is less than that of SBO as shown in Table 4.2. In this case study, 30 infill iterations are performed to find the optimal design for each prescribed CG. For comparison, the convergence analysis of minimum J_d (Min. J_d) of the surrogate model for these three different optimization methods is shown in Figure 4.4. For all prescribed CGs, two MFSBO methods converge faster, i.e., a lower J_d within fewer iterations. SBO does not fully converge, and therefore, more infills are necessary to reach the global minimum. Besides, two MFSBO methods have a similar convergence rate. For this particular case study, MFSBO-4 does not dramatically exceed MFSBO-1 because optimizing three design parameters is a relatively easy task, and the four parallel LF infills may not be able to exhibit their full utilities.

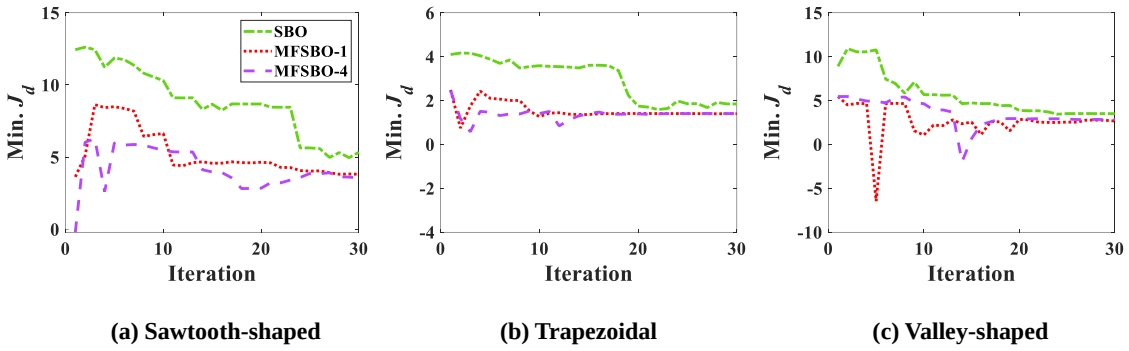


Figure 4.4 Convergence of Min. J_d using different optimization methods for prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped.

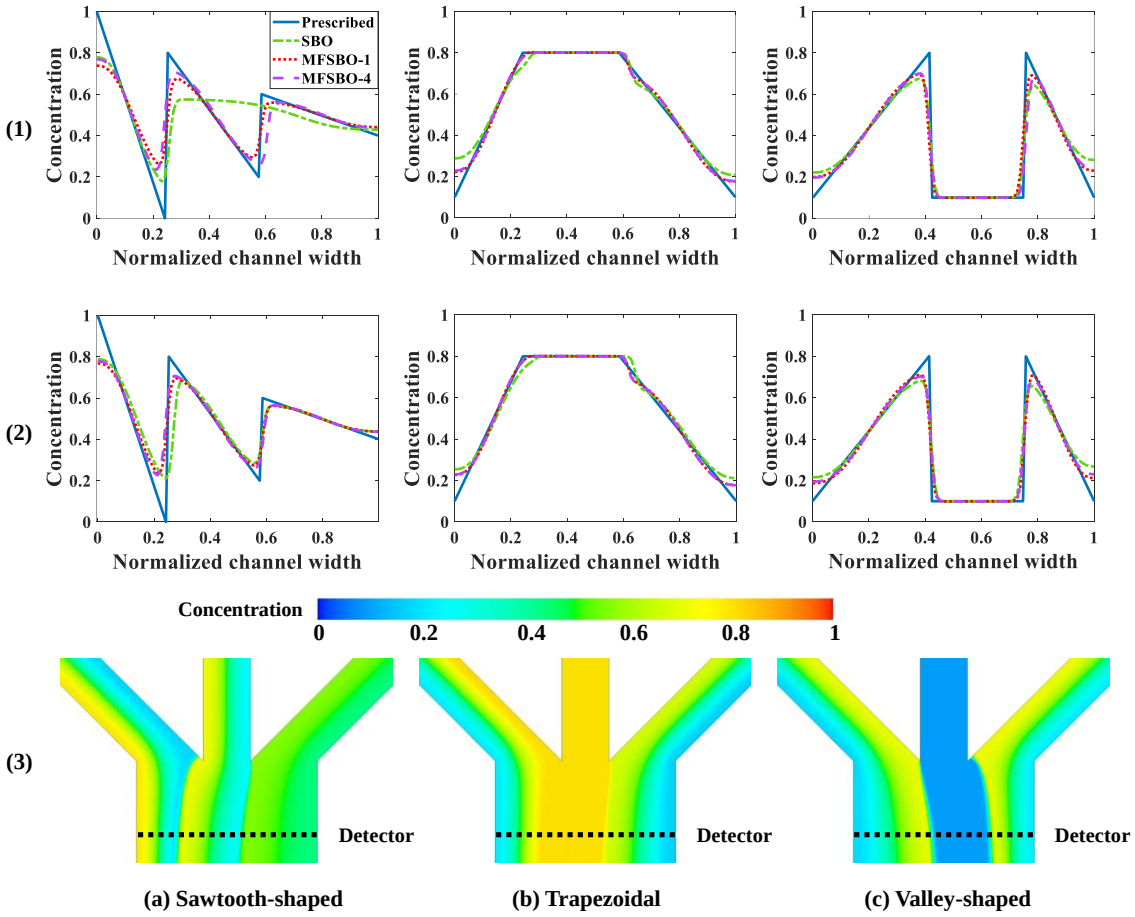


Figure 4.5 Comparison of predicted CGs to the prescribed CG: (1) the design corresponding to the minimum of the surrogate model at 20th iteration and (2) optimal design at the end of the optimization, and (3) the CFD contour plots at optimal designs attained using MFSBO-4 for three prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped.

Table 4.1 J_d of the design corresponding to the minimum of the surrogate model at the 20th iteration and the optimal design for the three prescribed CG.

	Method	J_d		
		Sawtooth-shaped	Trapezoidal	Valley-shaped
20th Iteration	SBO	8.73	2.17	3.82
	MFSBO-1	4.58	1.41	3.12
	MFSBO-4	4.47	1.41	3.03
Optimal Design	SBO	5.09	2.29	3.63
	MFSBO-1	3.79	1.40	2.79
	MFSBO-4	3.61	1.41	2.85

The comparison between the prescribed CGs and the predicted CGs at the minimum of the surrogate model at the 20th iteration and the optimal design is made in Figure 4.5 along with the CFD contour plots of predicted CGs at the optimal design attained using MFSBO-4. The top row shows the predicted CGs at the minimum of the surrogate model at the 20th iteration, which represents the intermediate results prior to reaching the optimal design, and can be used to observe the optimization process. The second row in Figure 4.5 shows the CGs comparison at the optimal design obtained at the end of the design optimization (i.e., 30th iteration), and the third row is the CFD contour plots at the optimal design by MFSBO-4 for the three prescribed CGs. The corresponding numerical values of J_ds are also computed and listed in Table 4.1. Recall that J_d represents the discrepancy between predicted CGs with respect to prescribed CGs. It is clear that all predicted CGs at the end of the optimization more resemble the prescribed CGs, indicating convergence of the optimization method. The results also show that the predicted CGs by the two MFSBO methods are closer to the prescribed CGs than that by the SBO method at the 20th iteration. According to J_ds in Table 4.1, the predicted CGs by SBO deviate more from the prescribed

than the other two methods, especially for the sawtooth-shaped CG. The results at optimal design clearly show that compared to SBO, MFSBO finds the design that could generate CGs in better agreement with all the three prescribed CGs, corresponding to 29% reduction in J_d on the average. All these confirm that the MFSBO converges to better design, i.e., lower J_d at a faster rate.

The number and cost of simulation runs for the three different optimization methods are listed in Table 4.2. Note that, the computational time of the HF simulation (CFD) is 1,325 s per simulation, and the LF simulation (PBCM) is 0.103 s per simulation (the number of the Fourier series terms used is 700). Because MFSBO uses LF simulations to reduce the number of HF simulations and the computational time of LF simulation is almost negligible, the initial and the total computational cost for MFSBO is actually less than SBO, while the former converges to a better design faster.

Table 4.2 The number and cost of simulation runs for the three optimization methods.

	Initial			Total		
	CFD	PBCM	Simulation time	CFD	PBCM	Simulation time
SBO	5	0	6,625	35	0	46,375
MFSBO-1	3	100	3,985.3	33	130	43,738.39
MFSBO-4	3	100	3,985.3	33	220	43,747.68

4.3.2 Case Study 2: Design of Inlet Concentrations

Next, the design of inlet concentrations, a six-dimensional optimization problem, is investigated for the μCGG design. Similarly, the initial samples are generated first. Five HF samples are produced for SBO, while 3 HF and 100 LF samples are generated for MFSBO. In this case study, 30 infill iterations are performed to find the optimal design for

each prescribed CG. Note that, even though the prescribed CGs are the same as the first case study, the difference in design variables makes the problem different. In general, a high-dimensional problem is more difficult to find the global optimum. The convergence comparison of Min. J_d of the surrogate model for the three different optimization methods is shown in Figure 4.6. Similar to the design of pressure differences above, both MFSBO methods outperform SBO in terms of the convergence rate. The SBO could not reach the global minimum manifested by a higher J_d for the three prescribed CGs. Besides, MFSBO-4 shows a slight advantage over MFSBO-1 for trapezoidal and valley-shaped CGs. It converges to a better design with a faster convergence rate while the added computational cost is negligible, i.e., three more LF PBCM simulations in each iteration. Therefore, the MFSBO-4 has faster convergence compared to MFSBO-1, when the complexity of the problem increases.

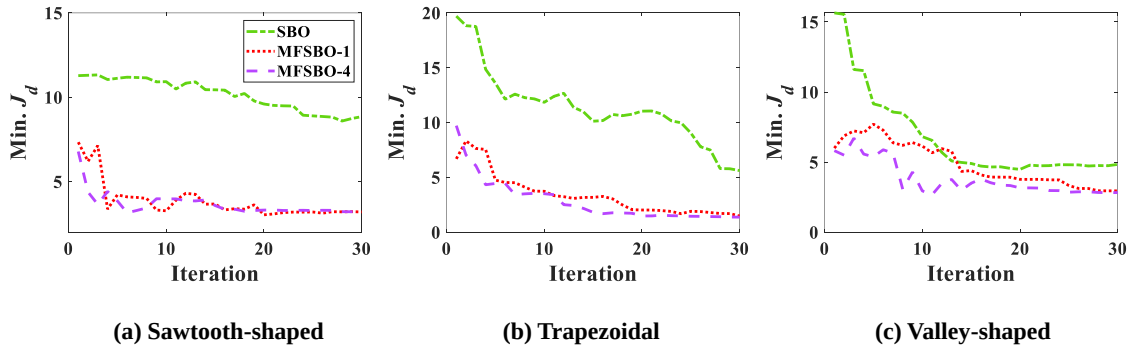


Figure 4.6 Convergence of Min. J_d using different optimization methods for prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped.

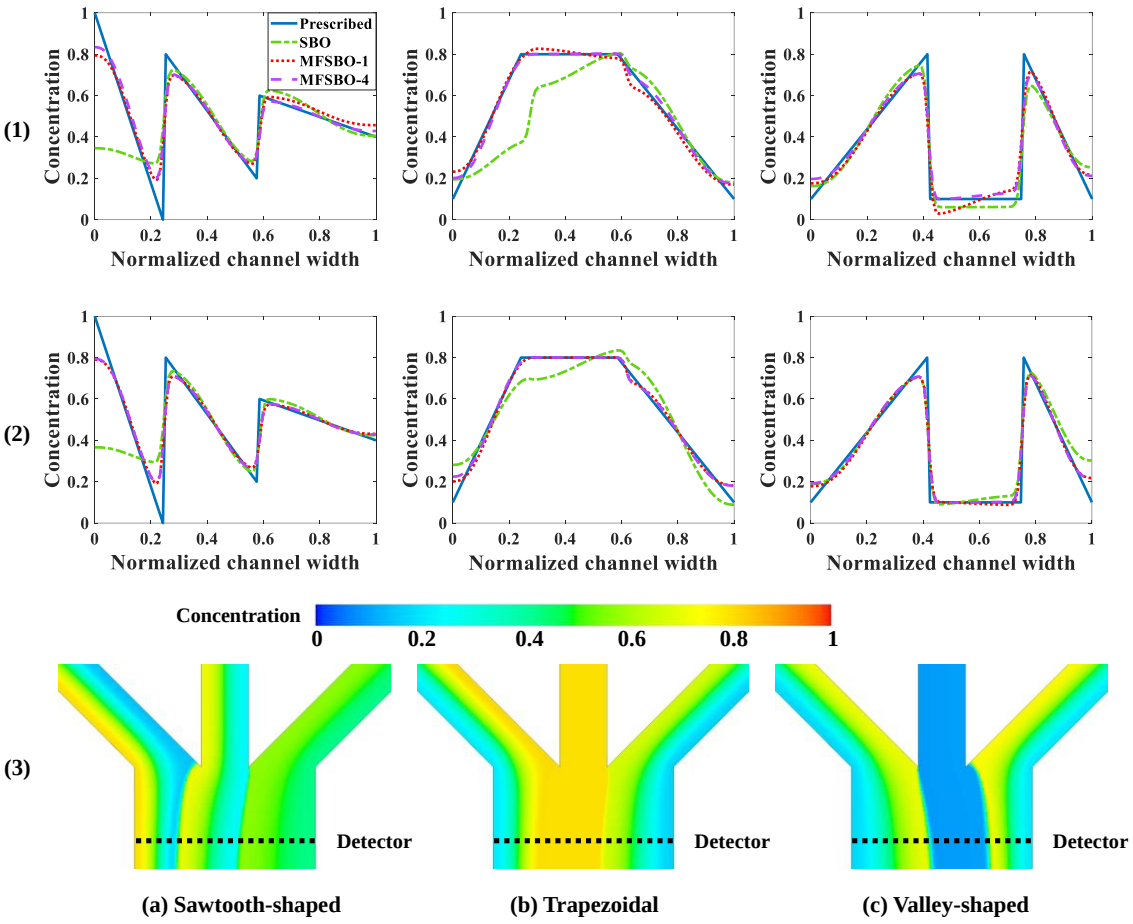


Figure 4.7 Comparison of predicted CGs to the prescribed CG: (1) the design corresponding to the minimum of the surrogate model at 20th iteration and (2) optimal design at the end of the optimization, and (3) the CFD contour plots at optimal designs using MFSBO-4 for three prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped.

Figure 4.7 portrays the comparison between the prescribed and predicted CGs corresponding to the minimum of the surrogate model at the 20th iteration and the optimal design for the three different prescribed CGs, along with the CFD contour plots of predicted CGs at the optimal design attained using MFSBO-4. The corresponding numerical values of J_ds are listed in Table 4.3. Compared to the SBO method, predicted CGs of the two MFSBO methods are closer to the three prescribed CGs at both the 20th iteration and the optimal design. The better agreements of CGs predicted by the MFSBO method at the 20th

iteration (row 1 in Figure 4.7) mean that even within the first half of the optimization process, MFSBO is already more accurate than SBO. According to the predicted CGs at optimal designs (row 2 in Figure 4.7), the SBO's predicted CGs have noticeable differences from the three prescribed CGs, indicating SBO does not converge to the global optimum. The comparisons in CGs at the optimal design further confirm that MFSBO outperforms SBO in terms of the convergence rate and accuracy with 57% additional reduction in J_d on the average. Besides, the predicted CGs of MFSBO-4 match the prescribed CG slightly better than those of MFSBO-1 with 4% more reduction in J_d. This further confirms the MFSBO-4 with four parallel LF infills accelerates the optimization process with three additional LF simulation runs whose computational time, however, is almost negligible.

Table 4.3 J_d of the design corresponding to the minimum of the surrogate model at 20th iteration and the optimal design for the three prescribed CG.

	Method	J_d		
		Sawtooth-shaped	Trapezoidal	Valley-shaped
20th Iteration	SBO	9.36	10.05	4.52
	MFSBO-1	4.02	2.26	3.81
	MFSBO-4	3.32	1.56	3.24
Optimal Design	SBO	8.62	5.87	4.25
	MFSBO-1	3.23	1.48	2.94
	MFSBO-4	3.22	1.38	2.82

Table 4.4 summarizes the simulation runs and time for the three different optimization methods. Similar to the design of pressure differences above, less HF CFD simulation and computational time is needed for MFSBO both initially and totally, because

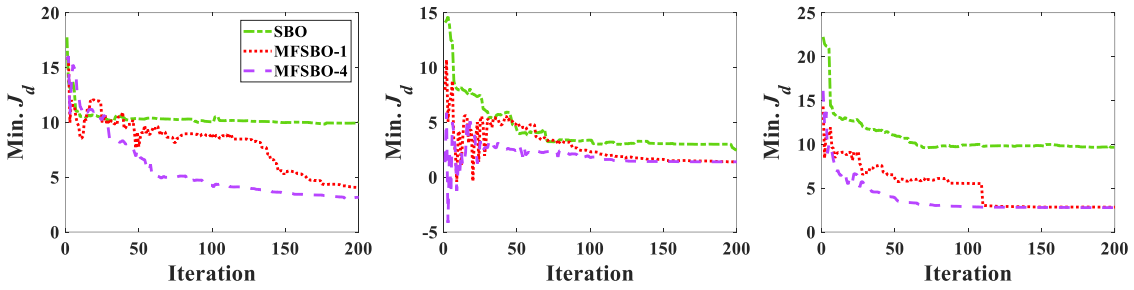
of the negligible LF simulation time. Thus, compared to SBO, MFSBO is more efficient (faster convergence) and accuracy (better agreement with the prescribed CGs).

Table 4.4 The number and cost of simulation runs for three optimization methods.

	Initial			Total		
	CFD	PBCM	Simulation time	CFD	PBCM	Simulation time
SBO	5	0	6,625	35	0	46,375
MFSBO-1	3	100	3,985.3	33	130	43,738.39
MFSBO-4	3	100	3,985.3	33	220	43,747.68

4.3.3 Case Study 3: Design of Pressure Differences and Inlet Concentrations

Furthermore, the design of pressure differences and inlet concentrations, a nine-dimensional optimization problem, is performed. Because of the increased complexity of the optimization problem, more initial samples and infill iterations are required. 10 HF samples are generated for SBO, while 8 HF and 500 LF samples are generated for MFSBO. Optimizations are then carried out with 200 infill iterations for each prescribed CG. Figure 4.8 portrays convergence curves of Min. J_d of the surrogate model using three optimization methods for each prescribed CG. MFSBO outperforms SBO in terms of the convergence rate. SBO solution seems to reach the local minimum and more iterations are needed to converge to the global minimum. MFSBO methods approach the optimum before the 200^{th} iteration, leading to a short optimization process. Compared to MFSBO-1, MFSBO-4 converges significantly faster and to a better solution for all three prescribed CGs due to the use of more LF PBCM simulations whose computational time, however, is negligible. Therefore, it substantiates that MFSBO-4 outperforms MFSBO-1 drastically for a more complex design optimization problem.



(a) Sawtooth-shaped **(b) Trapezoidal** **(c) Valley-shaped**

Figure 4.8 Convergence of Min. J_d using different optimization methods for prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped.

Table 4.5 J_d of the design corresponding to the minimum of the surrogate model at 100th iteration and the optimal design for the three prescribed CG.

	Method	J_d		
		Sawtooth-shaped	Trapezoidal	Valley-shaped
100th Iteration	SBO	10.49	3.61	10.54
	MFSBO-1	9.12	2.34	5.48
	MFSBO-4	4.44	1.71	2.88
Optimal Design	SBO	10.26	2.65	10.00
	MFSBO-1	4.08	1.43	2.83
	MFSBO-4	3.17	1.41	2.78

Figure 4.9 shows the comparison between the prescribed and the predicted CGs corresponding to the minimum of the surrogate model at the 100th iteration and the optimal design for three different prescribed CGs. Table 4.5 lists the corresponding numerical value of J_d. The CGs comparisons at the 100th iteration (row 1 in Figure 4.9) agree with the convergence analysis of Min J_d in Figure 4.8. Besides, the predicted CGs at the optimal designs (row 2 in Figure 4.9) of MFSBO-4 are extracted from CFD contour plots (row 3 in Figure 4.9). The predicted CGs from the SBO exhibit huge differences from prescribed CGs for the sawtooth-shaped and valley-shaped CGs, and also a noticeable discrepancy for

the trapezoidal CG. It turns out that SBO is not able to converge to the global optimal for such a complex problem given a small number of iterations. The CGs comparisons at the optimal design show that the predicted CGs of MFSBO are closer to the prescribed CGs than those of SBO with 61% less J_d on the average. This provides additional evidence to confirm that MFSBO converges faster than SBO. Furthermore, MFSBO-4 also exceeds MFSBO-1, and the predicted CGs of MFSBO-4 have 8% less J_d than those of MFSBO-1.

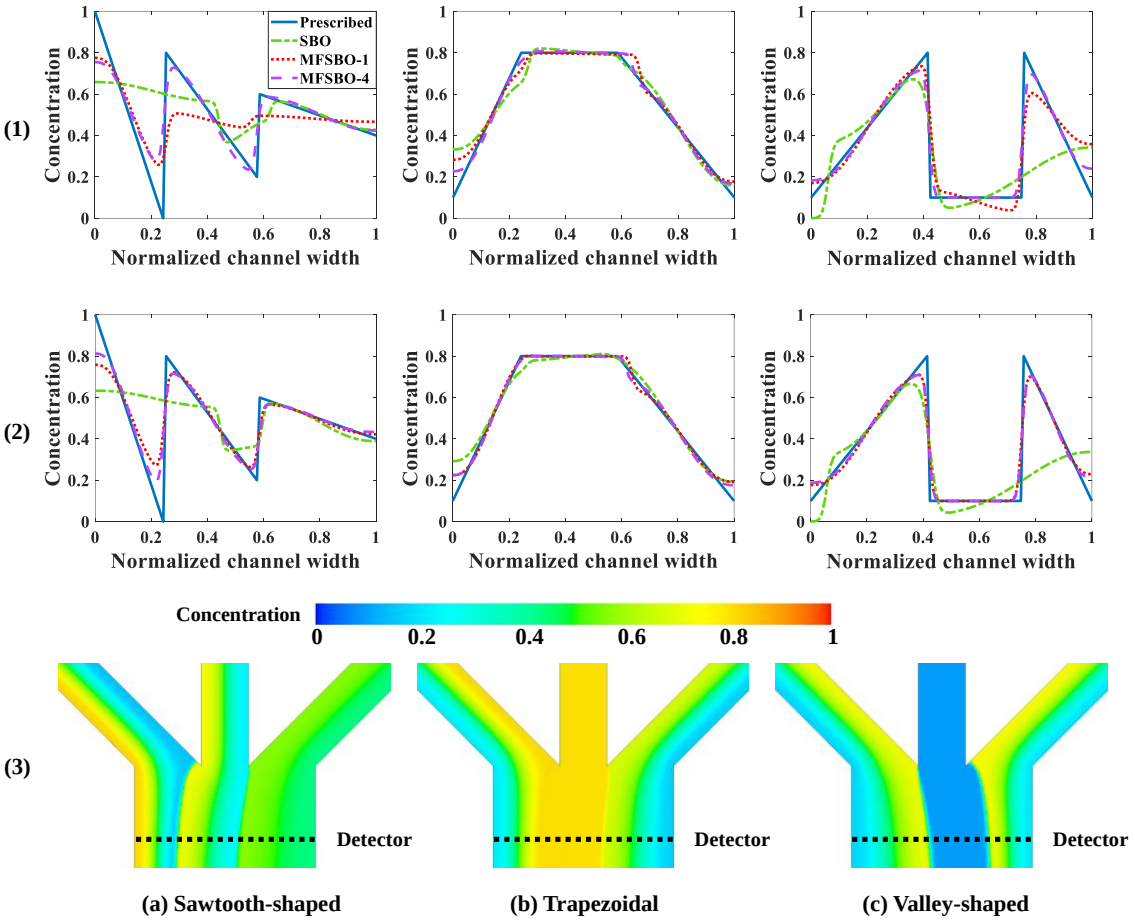


Figure 4.9 Comparison of predicted CGs to the prescribed CG: (1) the design corresponding to the minimum of the surrogate model at 100th iteration and (2) optimal design at the end of the optimization, and (3) the CFD contour plots at optimal designs attained using MFSBO-4 for three prescribed CGs: (a) sawtooth-shaped, (b) trapezoidal, and (c) valley-shaped.

Table 4.6 portrays the number and cost of simulation runs for three different optimization methods. MFSBO uses fewer HF simulation runs initially and totally, and reaches a better design at a faster rate. Besides, in contrast to MFSBO-1, MFSBO-4 achieves a better convergence rate and optimal design with almost the same simulation time due to the tiny cost of the LF PBCM simulation.

Table 4.6 The number and cost of simulation runs for three optimization methods.

	Initial			Total		
	CFD	PBCM	Simulation time (s)	CFD	PBCM	Simulation time (s)
SBO	10	0	13,250	210	0	278,250
MFSBO-1	8	500	10,651.5	208	700	275,672.1
MFSBO-4	8	500	10,651.5	208	1300	275,733.9

4.4 Summary

In this chapter, an MFSBO method is presented for the efficient and reliable global design optimization of μCGGs. Two new multi-fidelity infill strategies for MFSBO are developed and compared with surrogate-based optimization in terms of the convergence rate, optimal design accuracy, and simulation run time. New aspects of the proposed research include: first, the feasibility of applying MFSBO to complex μCGG designs is systematically examined and established. Second, data from two different sources (PBCM and CFD) and with different fidelities and computational costs are combined into MFSBO. Third, the design of CGs comprised of asymmetric constituent profiles, which cannot be obtained through SBO with PBCM only, is achieved by MFSBO. Last, a parallel infill strategy is applied, and its salient efficiency is confirmed for complex, high-dimensional optimization problems.

Three case studies are undertaken to evaluate the performance of the proposed method. Results show that all predicted CGs from MFSBO match prescribed CGs, which verifies the feasibility, robustness, and accuracy of the proposed method. Furthermore, MFSBO exhibits a faster convergence rate and a better design than SBO as indicated by 49% less J_d of the former on the average. MFSBO-4 outperforms MFSBO-1 in both convergence rate and design accuracy. MFSBO-4 reaches 4% and 8% additional reduction of J_d relative to MFSBO-1 for the last two case studies, respectively. Therefore, MFSBO-4, which applies four parallel LF infills, is more robust and efficient for complex design problems.

CHAPTER 5 A SEQUENTIAL MULTI-FIDELITY SURROGATE-BASED OPTIMIZATION METHODOLOGY BASED ON EXPECTED IMPROVEMENT REDUCTION[3]

[3] Yang, Haizhou, Seong Hyeon Hong, and Yi Wang. Submitted to *Structural and Multidisciplinary Optimization*, 10/25/2021

In this chapter, in order to enable sequential and adaptive batch sampling, and determine the evaluation fidelity (which is also called data source hereafter in this chapter) and infill samples to maximize the optimization efficiency, a computation-aware MFSBO methodology and a new infill strategy based on EIR are proposed. "Computation-aware" means that the selection of data sources and subsequent infills also depends on their computational costs. Given a fixed computational budget for each iteration, the proposed MFSBO methodology will first determine a single infill location through one of the commonly used criteria EI. Two hypothetical MFSMs will be obtained by updating the existing MFSM separately with additional hypothetical HF and LF data. The hypothetical HF data is acquired by evaluating the MFSM exactly at the infill location. The hypothetical LF data is obtained through two steps: sampled inputs are generated by space-filling sampling techniques within the trust-region centered around the infill location, and the output responses are evaluated by the low-fidelity surrogate model (LFSM). Finally, the values of EIR are computed for both hypothetically updated MFSMs to determine the data source and samples of infill for the next iteration (see detail in section 5.1.1).

The major novelties of the present chapter include: (1) a new EIR infill strategy is proposed to determine the data source (fidelity level) and samples of infill. The decision is made by examining the potential of EIR through hypothetically updated MFSMs and the computational cost ratio between HF-based and LF-based update (including simulation time and MFSM update time, shown in Eq. (5.2)) given a fixed computational budget for each iteration; (2) The MFSM update time or surrogate modeling time, typically neglected in the previous research, is considered in the present work; (3) the dynamically varying and iteration-dependent trust-region allows batch sampling for LF model evaluation and data

generation to improve exploration on a self-adaptive, as-needed basis for enhanced optimization convergence and computational resource utilization; and (4) EIR-based infill strategy is thoroughly studied by comparison with other pertinent infill strategies, and its performance is fully characterized through numerical studies of varying complexity levels.

5.1 Proposed Methodology

Figure 5.1 shows the flowchart of the proposed MFSBO with the EIR-based infill strategy. The detailed procedure is described as follows: First, it starts with an initial sampling plan, such as the LHS to generate two sets of initial samples, i.e., $\mathbf{X} = \{ \mathbf{X}^H, \mathbf{X}^L\}$, where $\mathbf{X}^H$ is the HF sample set and $\mathbf{X}^L$ is the LF sample set to fill the multi-dimensional parameter space (Block ① in Figure 5.1). Second, HF and LF models are simulated at HF and LF sample points to generate corresponding responses i.e., $\mathbf{Y} = \{\mathbf{Y}^H, \mathbf{Y}^L\}$, which then form the training data (Block ② and ③ in Figure 5.1). Third, an MFSM (denote as 'F') (Block ④ in Figure 5.1) is constructed using the training data in the previous step and the co-Kriging method above to capture the input-output mapping relationship. Fourth, the initial MFSM is inaccurate because the initial training data is limited, and hence, the optimum found by the current MFSM might not be accurate. Therefore, the proposed adaptive sampling process, EIR-based infill (Block ⑥ in Figure 5.1), is incorporated into the process to add samples at critical locations iteratively within the parameter space, which will improve the accuracy of MFSM gradually and accelerate the convergence of the optimization process. In the EIR-based infill, the infill location $\hat{\mathbf{x}}_i$ (Block (6-a) in Figure 5.1) is determined first by the EI criteria (i.e., Eq. (2.35)), where i denotes the i^{th} iteration or the current iteration. At this infill location, two hypothetical cases are interrogated and

compared to determine the data source (HF or LF) for infill with respect to the computational cost ratio of HF-based to LF-based update and the potential of EIR.

In case 1, 1 hypothetical HF infill sample $\tilde{\mathbf{x}}_i^H$ is added at the infill location $\hat{\mathbf{x}}_i$ determined in Block (6-a), and its output response $\tilde{\mathbf{y}}_i^H$ is calculated by the MFSM $F_{i\text{-}1}$ obtained in the last iteration, only an approximate value of the true response, instead of the true HF simulation. This is also the reason why it is named the "hypothetical" HF infill, i.e., Block (6-b) in Figure 5.1. Then the hypothetical HF infill sample $\tilde{\mathbf{x}}_i^H$ and corresponding output $\tilde{\mathbf{y}}_i^H$ are tentatively included in the true data set from the last iteration, i.e., $\tilde{\mathbf{X}}_i^H = \{\mathbf{X}_{i\text{-}1}, \tilde{\mathbf{x}}_i^H\}$ and $\tilde{\mathbf{Y}}_i^H = \{\mathbf{Y}_{i\text{-}1}, \tilde{\mathbf{y}}_i^H\}$, where $\mathbf{X}_{i\text{-}1}$ and $\mathbf{Y}_{i\text{-}1}$ denote the input and the response of the data set generated by the true HF and LF simulation in the previous iteration; and the tilde represents the quantities associated with the hypothetical data. The data set $\tilde{\mathbf{X}}_i^H = \{\mathbf{X}_{i\text{-}1}, \tilde{\mathbf{x}}_i^H\}$ and $\tilde{\mathbf{Y}}_i^H = \{\mathbf{Y}_{i\text{-}1}, \tilde{\mathbf{y}}_i^H\}$ are used to update MFSM, i.e., $\tilde{F}_I^H$ (Block (6-c) in Figure 5.1).

In case 2, Q LF hypothetical infill samples $\tilde{\mathbf{x}}_i^L$ (Block (6-d) in Figure 5.1) are generated within a trust-region centered around the infill location $\hat{\mathbf{x}}_i$ determined in Block (6-a), where Q is the computational cost ratio between HF-based and LF-based update. The trust-region is a reduced design space, whose center locates at the infill location $\hat{\mathbf{x}}_i$, and its size is determined by a convergence rate-related parameter (detailed in section 5.1.1). The outputs for the Q LF samples $\tilde{\mathbf{y}}_i^L$ are generated by the LFSM $f_{i\text{-}1}$ that is constructed by using only LF simulation data (including the input set $\mathbf{X}_{i\text{-}1}^L$ and the output response set $\mathbf{Y}_{i\text{-}1}^L$) and is explicitly available when MFSM is built (see Eq. (2.22)). Therefore, $\tilde{\mathbf{y}}_i^L$ is also hypothetical. Similar to case 1 above, $\tilde{\mathbf{x}}_i^L$ and $\tilde{\mathbf{y}}_i^L$ are tentatively included in the true data set in the last iteration, i.e., $\tilde{\mathbf{X}}_i^L = \{\mathbf{X}_{i\text{-}1}, \tilde{\mathbf{x}}_i^L\}$ and $\tilde{\mathbf{Y}}_i^L = \{\mathbf{Y}_{i\text{-}1}, \tilde{\mathbf{y}}_i^L\}$) to update MFSM $\tilde{F}_i^L$ as shown in Block (6-e) in Figure 5.1.

After analyzing these two hypothetical cases, the values of expected improvement reduction (EIRs) that measure the EI differences of MFSMs before and after the two hypothetical infills (i.e., EIRs of $\tilde{F}_i^H$ and $\tilde{F}_I^L$ relative to $F_{i\text{-}1}$), are computed (Block (6-f) in Figure 5.1). The one with the larger EIR implies a higher potential of improving the optimization accuracy and accelerating the design process. Thus, the data source (from either HF or LF simulation) and infill samples (either $\tilde{\mathbf{x}}_i^H$ or $\tilde{\mathbf{x}}_i^L$) are determined by comparing EIRs in both cases (Block (6-g) in Figure 5.1). Subsequently, the infill samples (either $\mathbf{x}_i^H = \tilde{\mathbf{x}}_i^H$ or $\mathbf{x}_i^L = \tilde{\mathbf{x}}_i^L$) and the corresponding true outputs (either $\mathbf{y}_i^H$ or $\mathbf{y}_i^L$) generated by HF and LF simulation (Block ② and ③ in Figure 5.1), are added to the true data set in the last iteration, viz., $\mathbf{X}_i = \{\mathbf{X}_{i\text{-}1}, \mathbf{x}_i^H \text{ or } \mathbf{x}_i^L\}$ and $\mathbf{Y}_i = \{\mathbf{Y}_{i\text{-}1}, \mathbf{y}_i^H \text{ or } \mathbf{y}_i^L\}$ to update MFSM, i.e., F_i for the current iteration. It should be reiterated that in contrast to the hypothetical evaluation above (Block ⑥ in Figure 5.1), the output response $\mathbf{y}_i^H$ or $\mathbf{y}_i^L$ are generated by HF and LF simulation rather than MFSM $F_{i\text{-}1}$ or LFSM $f_{i\text{-}1}$. This process continues until the minimum of MFSM (Block ⑤ in Figure 5.1) converges or the maximum number of iterations is reached. Finally, the minimum of MFSM in the last iteration is extracted. The details of the EIR-based infill process, i.e., Block ⑥ in Figure 5.1 are presented in the next section.

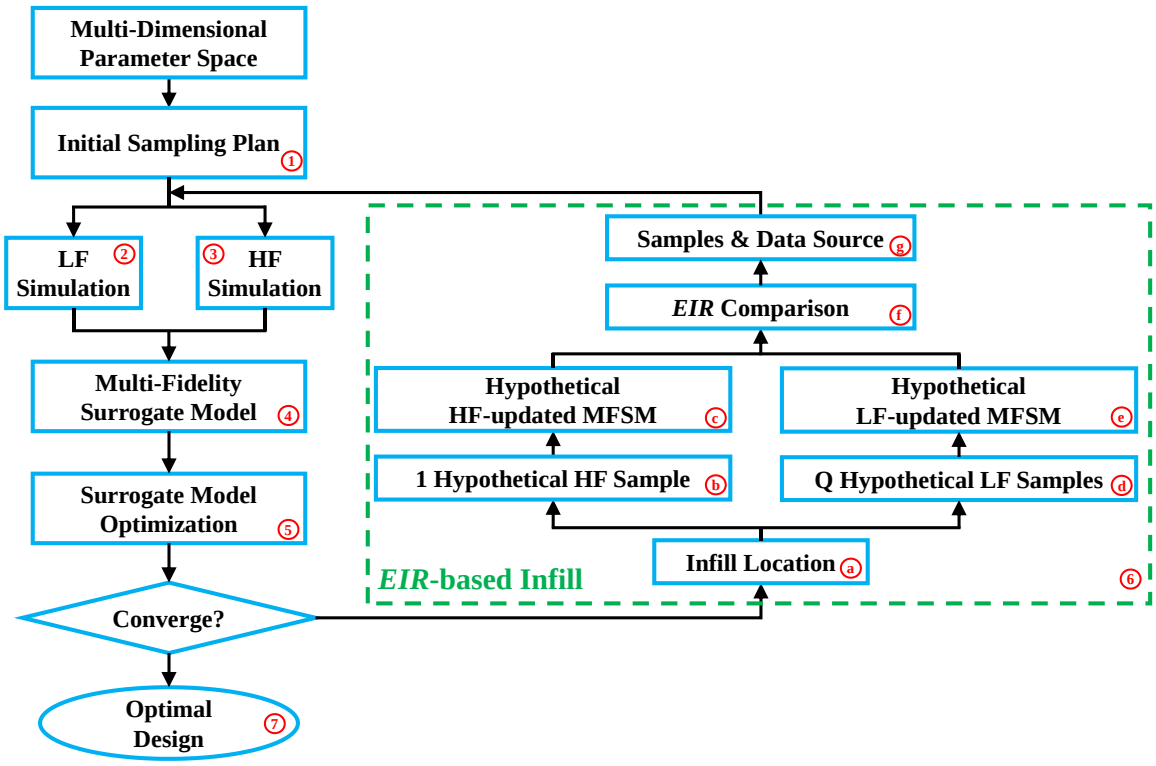


Figure 5.1 Flowchart of MFSBO with EIR-based Infill.

5.1.1 EIR-based Infill

Figure 5.2 illustrates the proposed EIR-based infill procedure using an example in 2D parameter space. First, an infill location $\hat{\mathbf{x}}_i$, i.e., Step (2) in Figure 5.2 corresponding to Block (6-a) in Figure 5.1 is determined by applying one of the widely used infill strategies to the current MFSM. Specifically, the EI infill is adopted in this study, although others can also be used in this step (H. Yang et al. 2020). The infill location will be used for generating HF and LF infill samples.

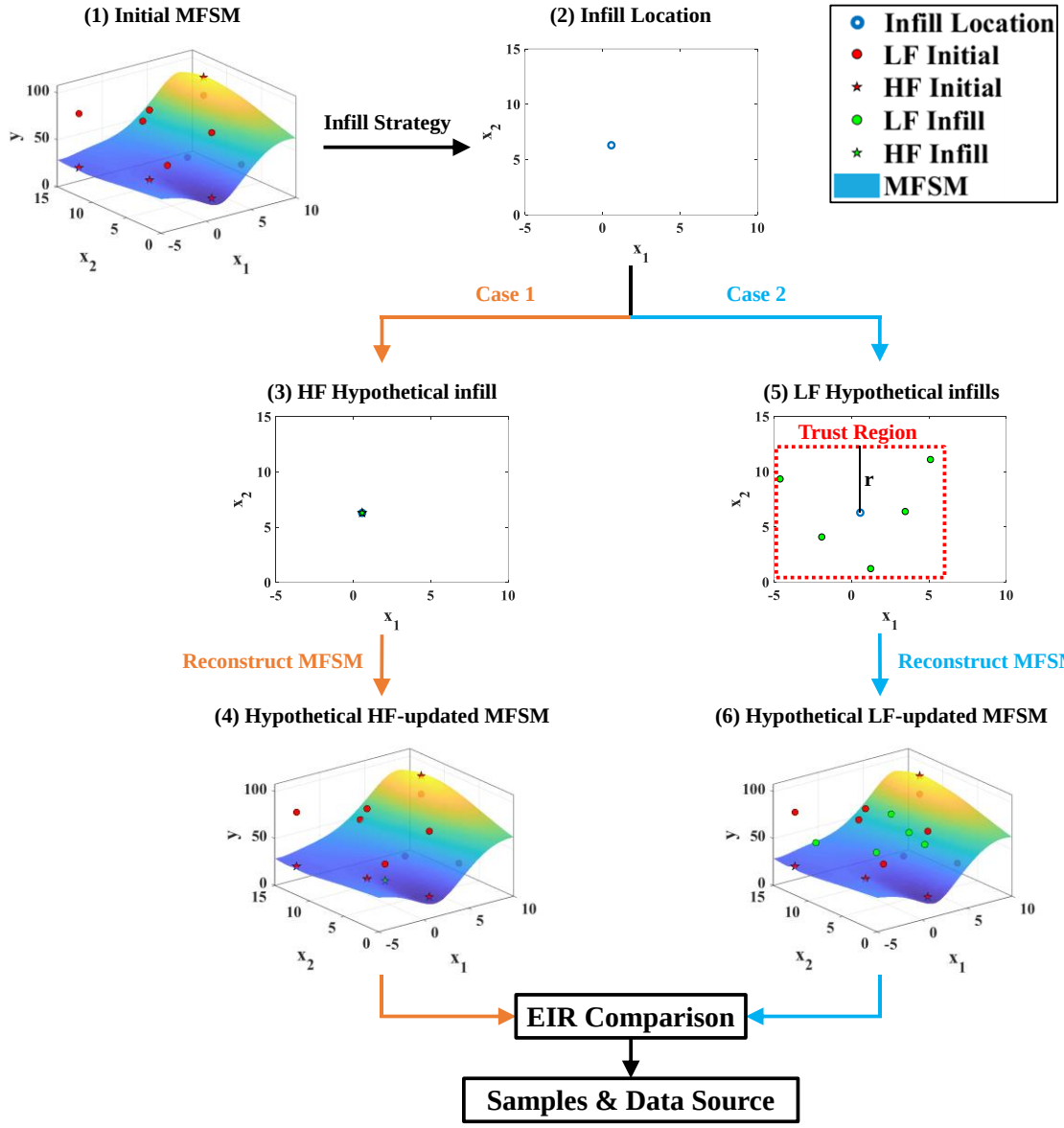


Figure 5.2 2D example flowchart of EIR-based infill.

Second, once the infill location is determined, two hypothetical cases presented above are performed independently. In the first case, 1 HF hypothetical sample $\tilde{\mathbf{x}}_i^H$, i.e., Step (3) in Figure 5.2 corresponding to Block (6-b) in Figure 5.1, located exactly at the infill location $\hat{\mathbf{x}}_i$, is selected for generating the hypothetical HF output $\tilde{\mathbf{y}}_i^H$ and updates the MFSM. Again, in order to reduce computational time, the predicted value from MFSM,

rather than the HF simulation, is used as the approximate HF output response. Note that the computational cost of MFSM is virtually negligible compared to HF simulation. The hypothetical HF sample $\tilde{\mathbf{x}}_i^H$ and corresponding response $\tilde{\mathbf{y}}_i^H$ are tentatively combined with the true data set ($\mathbf{X}_{i-1}$ and $\mathbf{Y}_{i-1}$) as described above to update MFSM $\tilde{F}_i^H$ (hypothetical updated MFSM with hypothetical HF infill data), i.e., Step (4) in Figure 5.2 and Block (6-c) in Figure 5.1. In the second case, the EIR-based infill strategy simultaneously samples Q LF hypothetical samples $\tilde{\mathbf{x}}_i^L$ (Step (5) in Figure 5.2 corresponding to Block (6-d) in Figure 5.1) by LHS in a trust-region. As described above, the center of the trust-region is at the infill location $\hat{\mathbf{x}}_i$ determined in Step (2) in Figure 5.2, and its size is dictated by a convergence rate-related parameter. The range of the trust-region is defined as:

$$
\begin{aligned}
&\mathbf{TL}_i = \max\{\mathbf{xd}_{\min}, \hat{\mathbf{x}}_i - \mathbf{r}_i\} \\
&\mathbf{TU}_i = \min\{\mathbf{xd}_{\max}, \hat{\mathbf{x}}_i + \mathbf{r}_i\} \\
&\mathbf{r}_i = \lambda_i \times \left(\mathbf{xd}_{\max} - \mathbf{xd}_{\min}\right) \\
&\lambda_i = \begin{cases} 0.1 & \lambda_i \le 0.1 \\ \left(\dfrac{\left| y_{\min}^{i-1} - y_{\min}^{i-2} \right|}{\max(\mathbf{K}) - \min(\mathbf{K})} \right)^{0.7} & 0.1 < \lambda_i < 0.7 \\ 0.7 & \lambda_i \ge 0.7 \end{cases}
\end{aligned}
\tag{5.1}
$$

where **TL** and **TU** are the lower bound and upper bound of the trust-region, respectively. $\mathbf{xd}_{\min}$ and $\mathbf{xd}_{\max}$ are the lower bound and upper bound of the design parameter space. Again $\hat{\mathbf{x}}$ is the coordinate of the infill location (Step (2) in Figure 5.2). **r** is half of the length of the trust-region, and also determines the area of the region. λ is the convergence rate parameter, and measures the convergence rate at the i^{th} iteration. $\mathbf{K} = \left[y_{\min}^{i-10}, y_{\min}^{i-9}, \cdots, y_{\min}^{i-1} \right]$ is a vector that records the minimum of the model in the previous ten iterations and is utilized to compute the convergence rate parameter λ. p is a user-defined parameter that

adjusts the effect of the convergence rate on the size of the trust-region. In this study, an empirical value $p = 0.7$ is applied. The limit for λ is empirically set to be [0.1 0.7], which works well for all the case studies in this chapter. A larger λ at the i^{th} iteration indicates that the minimum of MFSM deviates a lot from that of the last iteration, and thus, more exploration needs to be applied for infill. Therefore, a larger λ results in a larger r and a larger range of the trust-region for sampling. On the contrary, a smaller λ corresponds to a smaller r, which narrows down the trust-region for sampling to allow more exploitation. Within the trust-region, Q LF hypothetical samples $\tilde{\mathbf{x}}_i^L$ (Step (5) in Figure 5.2 and Block (6-d) in Figure 5.1) are generated with LHS and used to update MFSM $\tilde{F}_i^L$. Next, Q LF samples $\tilde{\mathbf{x}}_i^L$ and corresponding hypothetical responses $\tilde{\mathbf{y}}_i^L$ evaluated by LFSM (not LF simulation), are tentatively incorporated into the true data set to update MFSM, i.e., $\tilde{F}_i^L$ for case 2 above (Step (6) in Figure 5.2 and Block (6-e) in Figure 5.1). Q, the computational cost ratio of HF-based to LF-based update, is defined as:

$$\mathrm{Q} = \frac{T_{\mathrm{S}}^H + T_{\mathrm{M}}^H}{T_{\mathrm{S}}^L + T_{\mathrm{M}}^L} \tag{5.2}$$

where T_{S} is the simulation time; T_{M} is the average surrogate modeling time during the optimization (Block ④ in Figure 5.1). Superscript H and L represent HF and LF, respectively. Note that the ratio Q in this study considers LF or HF simulation time and surrogate modeling time as well during each iteration, which is one of the distinguishing aspects of the present work from previous research efforts that often neglected the surrogate modeling time. For some optimization problems, the surrogate modeling time could increase dramatically relative to the simulation time when the data set becomes larger and the correlation matrix calculation becomes more computationally demanding.

Third, expected improvement reductions (EIRs) obtained by the updated MFSMs in the two hypothetical cases ($\tilde{F}_i^H$ and $\tilde{F}_i^L$ relative to $F_{i\text{-}1}$) are measured by

$$EIR_M = EI_M - EI_0 \qquad M = 1, 2 \tag{5.3}$$

where EI_M is the expected improvement in case M, viz., after Step 4 or Step 6 in Figure 5.2. EI_0 is the one before infill, viz., in Step 1 in Figure 5.2. Since EI_0 is the same for both hypothetical cases, we can also directly compare the resulting EIs without resort to EIR. Last, the data source M^* (HF and LF) and corresponding infill samples (1 HF sample or Q LF samples) in the case with the higher EIR is selected for the next round of simulation and MFSM update:

$$M^* = \arg\max EIR_M \tag{5.4}$$

5.2 Results and Discussion

The proposed EIR-based infill method is examined with three case studies, including two numerical examples and one engineering example, which represent optimization problems with different dimensions and levels of complexities. It is also compared with the other three infill strategies in terms of convergence and optimization accuracy, including (1) H-infill strategy that adds 1 HF sample at the infill location in each iteration; (2) Alternating infill strategy that adds 1 HF sample and Q LF samples alternatively. For example, it infills Q LF samples at the 1st, 3rd, 5th, … iteration, and 1 HF sample at the 2nd, 4th, 6th, … iteration, and so on; and (3) Random infill strategy that adds 1 HF sample or Q LF samples randomly in each iteration, that is, the infill in each iteration will be randomly determined on-the-fly during optimization. For all infill strategies, the computational cost ratio is kept the same and constant throughout the optimization, and the

initial samples are also identical for the sake of fair comparison. Besides, due to the randomness associated with the genetic algorithm optimization and LHS in the LF infill, each numerical experiment below is repeated five times and the average is calculated and presented for objective performance benchmarking (convergence and design accuracy) of the proposed method.

5.2.1 2D Numerical Example: Brainin

The Brainin benchmark function is a two-dimensional, widely used function for the optimization problem. The HF and LF models are given as follows (Perdikaris et al. 2017):

$$
\begin{aligned}
f_H &= \left(\frac{-1.275x_1^2}{\pi^2} + \frac{5x_1}{\pi} + x_2 - 6\right)^2 + \left(10 - \frac{5}{4\pi}\right)\cos\left(x_1\right) + 10 \\
f_L &= 10\sqrt{f_H\left(x-2\right)} + 2\left(x_1 - 0.5\right) - 3\left(3x_2 - 1\right) - 1 \\
x_1 &\in [-5,10], x_2 \in [0,15]
\end{aligned}
\tag{5.5}
$$

where f_H represents the HF model, and f_L is the LF model. Their response surfaces are depicted in Figure 5.3. There are three global optima, $x^* = (-\pi, 12.28)$, $(\pi, 2.28)$, and $(9.42, 2.48)$, and their corresponding function value is $f_H(x^*) = 0.40$.

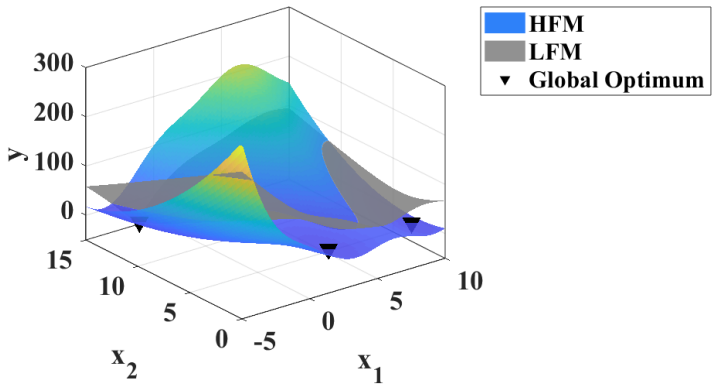


Figure 5.3 Response surfaces of HF and LF models for the Brainin function.

The optimization process starts with two HF and two LF samples, which are the same for all infill strategies in comparison. In this case study, 20 infill iterations are performed to add samples to update MFSM and conduct MFSBO. The computational cost ratio Q for this case study is set to be 10, that is, the computational time of a HF-based update (including simulation time and MFSM update time) is 10 times longer than that of a LF-based update. For comparison, the convergence of the minimum of MFSM for four different infill strategies is shown in Figure 5.4. Note that average results of five repeated runs that take the exactly same optimization parameters (initial samples, number of infill iterations, cost ratio, etc.) are presented for a fair comparison. The results show that EIR, alternating, and random infill converge faster than the H-infill. EIR, alternating, and random infill strategies converge at the 12^{th}, 18^{th}, and 18^{th} iteration, respectively, while H-infill has not converged yet at the 20^{th} iteration. Comparing to the other three infill strategies, the EIR-based infill strategy converges faster and to a minimum closer to the global optimum.

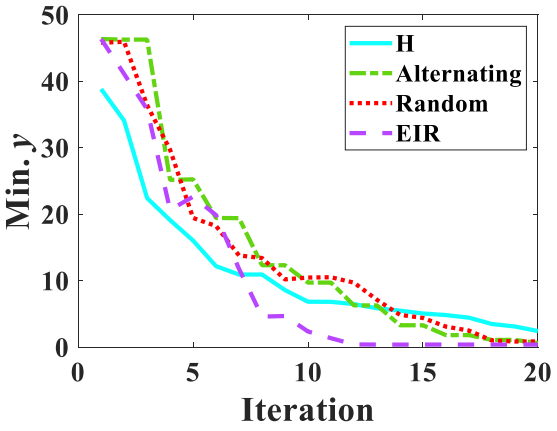


Figure 5.4 Convergence of MFSBO to the global minimum using different infill strategies.

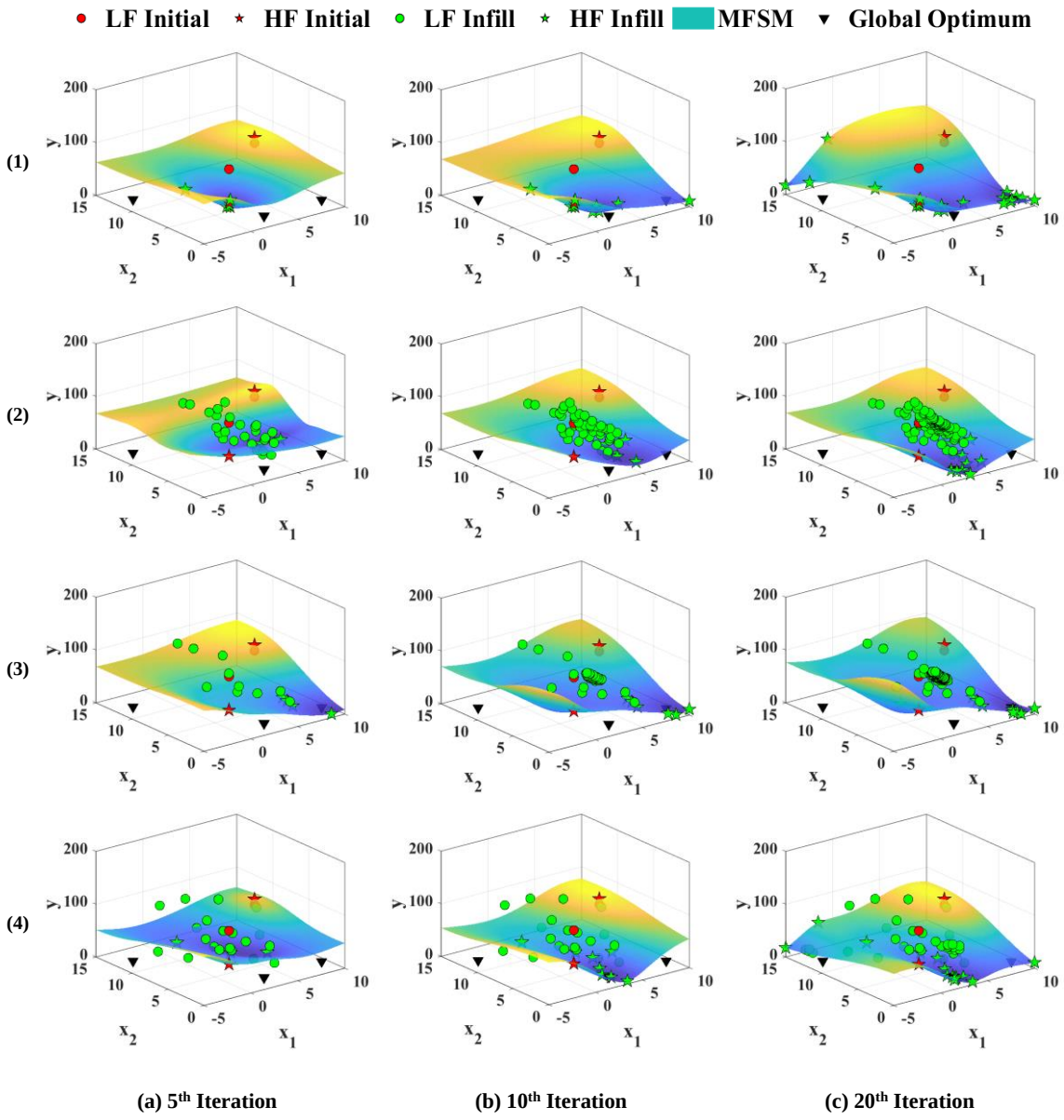


Figure 5.5 Response surface plots of the MFSM for the four infill strategies: (1) H infill, (2) alternating Infill, (3) random Infill, and (4) EIR-based Infill.

To further confirm the observation, the surface plots for H-only, alternating, random and EIR-based infill strategies are portrayed in Figure 5.5 from the first row to the fourth row. Their variations at the 5th, 10th, and 20th iteration are shown from the left to the right. Note that among the five repeated optimization runs, the one with median

performance in design accuracy is selected for these plots because average surface plots are physically meaningless. It clearly shows that the minimum of MFSM during iterations keeps decreasing and converges to the optimum as more infills are added for all infill strategies. For H-only, alternating and random infill, the MFSM gradually exhibits the true response surface around one global optimum. However, their convergence rates are relatively slow because many samples are infilled into non-targeted regions that have large objective values and are far away from the global optima. For the EIR-based infill, MFSM finds one global optimum at a relatively faster rate, and the region around the global optimum is under intensive search. Besides, the infill samples are mostly distributed in the region around the global optima. These results confirm that the EIR-based infill strategy outperforms the other three in terms of convergence rate and optimization accuracy, which agrees with the convergence analysis in Figure 5.4.

The minimum of MFSM and the corresponding objective value obtained from the average of the five repeated runs at the 10^{th} and the 20^{th} iteration (i.e., in the middle and at the end of the optimization) are listed in Table 5.1 for comparing the four infill strategies. At the 10^{th} iteration, the objective value of the minimum of MFSM using the EIR-based infill is closer to one of the global optima with a smaller relative error, while the relative errors of the other three are 1675.98%, 2335.84%, and 2706.37%. Furthermore, the results at the 20^{th} iteration show that the EIR-based infill identifies a more accurate optimum with 0.08% relative error only, which is in distinct contrast to the H-only, alternating, and random infill strategies that, respectively, have relative errors of 401.63%, 80.24%, and 177.64%. Note that although the Brainin function has three optima, they share the same objective value. The relative error presented above is based on any optimum found by the

individual method. The numerical results agree with the convergence analysis above, and EIR-based infill indeed outperforms the other three in terms of convergence rate and optimization accuracy.

Table 5.1 The minimum of MFSM and corresponding objective values at the 10^{th} and 20^{th} iteration.

	10^{th} Iteration			20^{th} Iteration		
	x_{min}	$\hat{y}(x_{min})$	Relative Error	x_{min}	$\hat{y}(x_{min})$	Relative Error
H	(2.64, 4.98)	7.07	1675.98%	(4.52, 3.81)	2.00	401.63%
Alternating	(6.86, 1.58)	9.69	2335.84%	(6.79, 2.04)	0.72	80.24%
Random	(6.68, 1.68)	11.17	2706.37%	(8.15, 2.24)	1.10	177.64%
EIR	(3.10, 2.96)	1.87	369.84%	(3.14, 2.27)	0.40	0.08%
Global Optima (x^*)	(-π, 12.28), (π, 2.28), (9.42, 2.48)					
$f_H(x^*)$	0.40					

Figure 5.6 illustrates the data source selection during each iteration for the EIR-based infill. Again, among the five runs, the one with median performance is presented. In the figure, 'H' represents choosing one HF sample for the infill, while 'L' means Q LF samples. It clearly shows that four iterations are selected to use LF data for infill to update MFSM, while the other 16 iterations are performed with HF simulations. Since each LF infill iteration runs 10 LF simulations that generates sufficient LF data, the EIR-based infill uses more HF iterations to obtain sensible contributions from both HF and LF data sources. In short, the proposed EIR-based infill judiciously manages the fidelity level (or data source) and samples to accelerate the convergence, i.e., 20% of the computational time allocated to LF simulation and 80% to HF simulation.

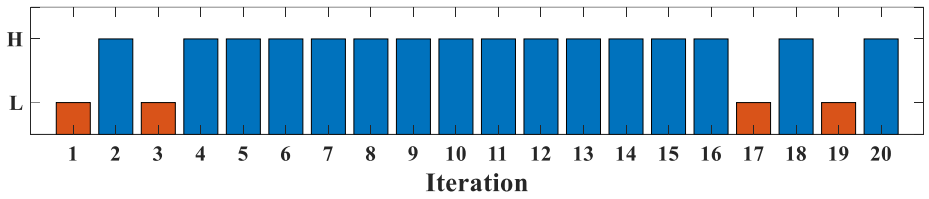


Figure 5.6 Selection of data source (fidelity level) at each iteration for the EIR-based Infill for the Brainin function.

5.2.2 4D Numerical Example: Hartmann 4-dimensional Function

In this example, we will evaluate the proposed method using the Hartmann 4-dimensional model. The HF model given in Eq. (5.6) is provided by (Picheny, Wagner, and Ginsbourger 2013) and the LF model is obtained by changing the optimum location and the scale of the functional value. Both are given as follows

$$
\begin{aligned}
f_H &= \frac{1}{0.839}\left[1.1 - \sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{4} A_{ij}\left(x_j - P_{ij}\right)^2\right)\right] \\
f_L &= 0.9 f_H\left(x - [-0.03,\ 0.05,\ -0.1,\ 0.07]\right) + 0.37 \\
&\text{where } \alpha = \left(1.0,\ 1.2,\ 3.0,\ 3.2\right)^T \\
&\mathrm{A} = \begin{pmatrix} 10 & 3 & 17 & 3.5 \\ 0.05 & 10 & 17 & 0.1 \\ 3 & 3.5 & 1.7 & 10 \\ 17 & 8 & 0.05 & 10 \end{pmatrix} \\
&\mathrm{P} = 10^{-4}\begin{bmatrix} 1312 & 1696 & 5569 & 124 \\ 2329 & 4135 & 8307 & 3736 \\ 2348 & 1451 & 3522 & 2883 \\ 4047 & 8828 & 8732 & 5743 \end{bmatrix} \\
&x_i \in [0,\ 1] \text{ for all } i = 1, 2, 3, 4
\end{aligned}
\tag{5.6}
$$

The global optimum of the Hartmann 4-dimensional function is x^* = (0.19, 0.19, 0.56, 0.26) and the corresponding function value $f_H(x^*)$ = -3.14. Figure 5.7 portrays the response surfaces of the HF and LF model. To facilitate visualization, the surfaces are shown in a 3D space and only vary with two design variables while keeping the other two

constants at their corresponding global optimum values x^*. The global optimum denoted by the triangular symbol in black can also be visualized in Figure 5.7.

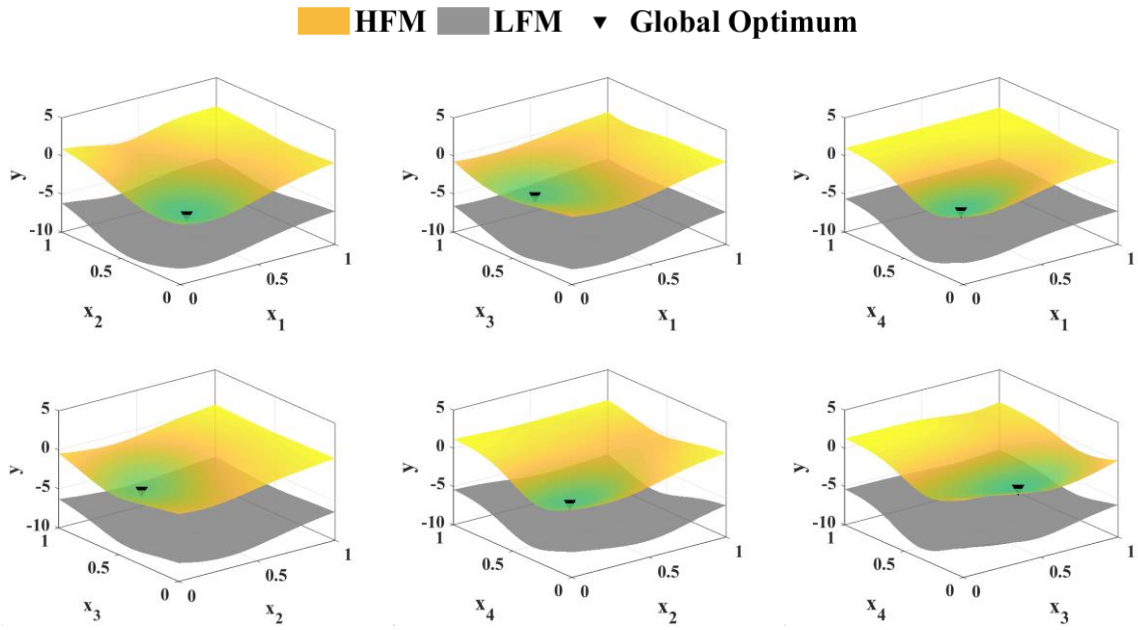


Figure 5.7 Response surfaces of the HF and LF model for the Hartmann 4 function.

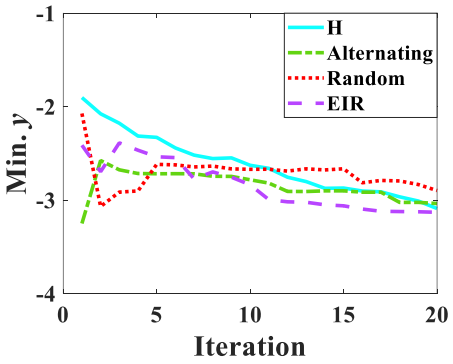


Figure 5.8 Convergence of the minimum of MFSM using different infill strategies.

Similar to the first case study, the initial samples are generated first, including two HF and two LF samples for all infill methods. In this case study, 20 iterations are performed to search for the global minimum. Since this numerical example is a four-dimensional

optimization problem with an elevated level of complexity, a larger computational cost ratio Q = 20 is set. The convergence comparison among the four infill strategies is shown in Figure 5.8. Again, the average results of the five repeated optimization runs are presented. At the 2nd iteration, the random infill strategy exhibits a sudden drop, which is attributed to the fact that the initial MFSM is not sufficiently accurate, and predicts a wrong minimum. This problem can be resolved by the adaptive sampling process, which adds more data into MFSM to improve its accuracy, and the minimum of MFSM will gradually converge to the global optimum. According to the convergence analysis, the EIR-based infill reaches the global minimum around the 16th iteration, while the other three could not converge until the 20th iteration. Thus, the EIR-based infill exhibits a faster convergence rate.

Figure 5.9 shows the surface plots of MFSM at the 5th,10th, and 20th iteration for the four infill strategies. Again, to facilitate visualization, the surface plot is portrayed in 3D by fixing x_3 and x_4 at their corresponding values of the global optimum x*. Similar to the above, the plots are extracted from the one with median performance. We can see that surface plots for the H-infill do not show a clear trend of convergence at the early iterations, although the profile of the response surface around the global optimum becomes more obvious with more infill samples. The minimum of MFSM with the alternating and random infill converge to a single point as more infill iterations are performed. However, it seems like the minimum does not fully reach the global minimum, and more iterations are required to improve the prediction accuracy. For the EIR-based infill, the minimum of MFSM progressively converges to the global optimum as more infill iterations are performed. Besides, EIR is able to find the optimum much faster than the other three methods.

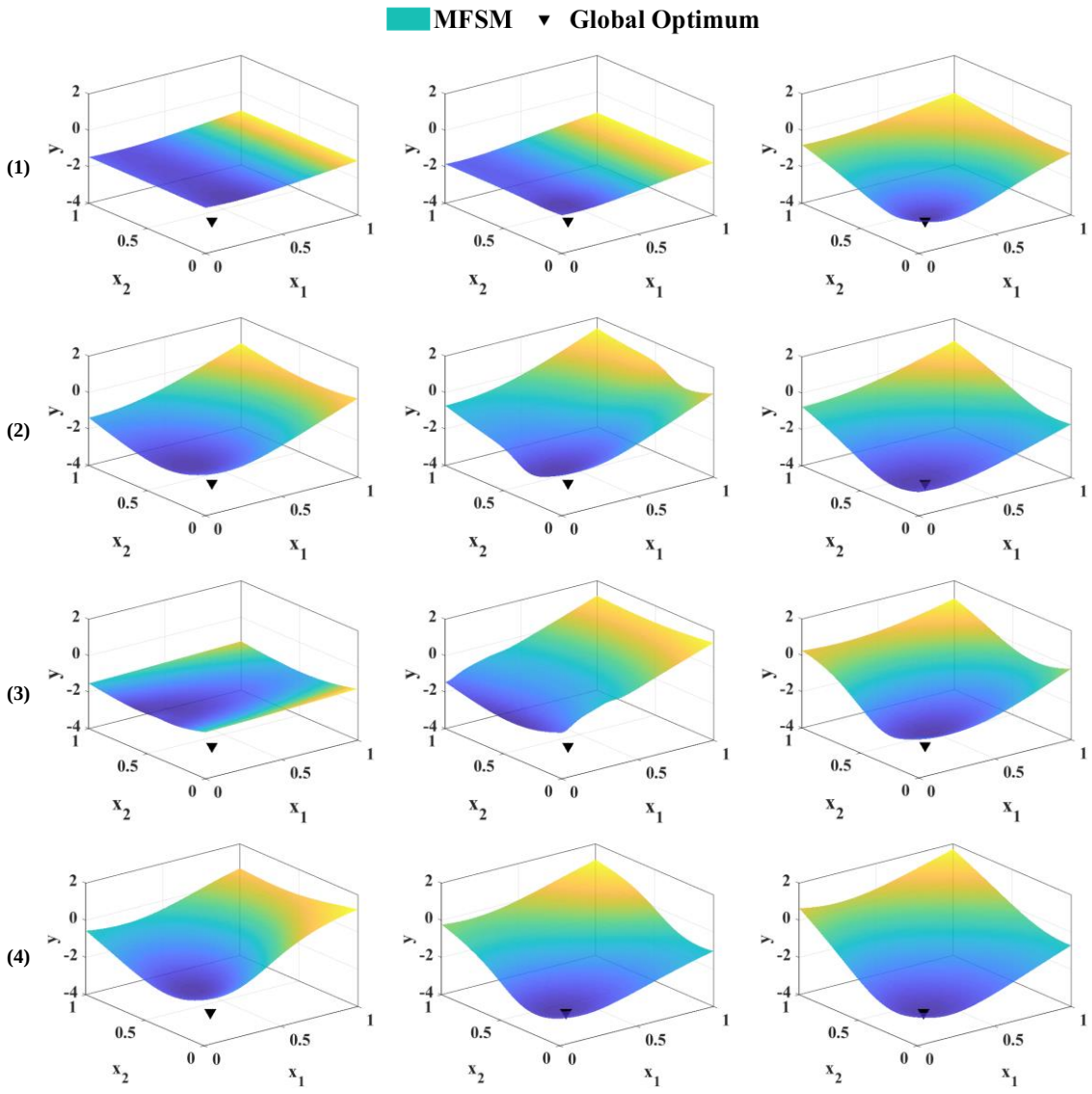


Figure 5.9 Response surface plots of MFSM for the four infill strategies: (1) H infill, (2) alternating Infill, (3) random Infill, and (4) EIR-based Infill.

Table 5.2 lists the minimum of MFSM and the corresponding objective value averaged across the five repeated runs at the 10th iteration and the 20th iteration. At the 10th iteration, the minimum of MFSM with the EIR-based infill is closer to the global optimum with only 8.21% relative error. The objective value of the 20th iteration for the EIR-based

infill is also closer to global optimal with only 0.15% relative error. It further confirms that the EIR-based infill accelerates the optimization process and produces more accurate optimization results.

Table 5.2 The minimum of MFSM and corresponding objective at 10^{th} iteration and 20^{th} iteration.

	10^{th} Iteration			20^{th} Iteration		
	x_{min}	$\hat{y}(x_{min})$	Relative Error	x_{min}	$\hat{y}(x_{min})$	Relative Error
H	(0.20, 0.12, 0.33, 0.29)	-2.57	18.12%	(0.20, 0.19, 0.53, 0.27)	-3.10	1.03%
Alternating	(0.14, 0.33, 0.61, 0.33)	-2.72	13.30%	(0.15, 0.21, 0.60, 0.28)	-3.04	3.01%
Random	(0.25, 0.39, 0.69, 0.34)	-2.48	20.81%	(0.22, 0.33, 0.65, 0.28)	-2.78	11.28%
EIR	(0.17, 0.24, 0.49, 0.30)	-2.88	8.21%	(0.19, 0.19, 0.55, 0.26)	-3.13	0.15%
Global Optimum (x^*)	(0.19, 0.19, 0.56, 0.26)					
$f_H(x^*)$	-3.14					

Figure 5.10 shows the selection of the data source (fidelity level) at each iteration for the EIR-based infill, which again is extracted from the one with median performance in the five optimization runs. Three iterations are assigned to LF simulation and infill, and 17 iterations to HF simulation during the adaptive sampling process. In other words, more computational time, i.e. 85% is assigned to HF infill, while 15% is left to LF infill. Besides, it also shows that the EIR-based infill chooses LF infills at the beginning for more exploration, while performing HF infills for exploitation to improve optimization accuracy in the latter half of the optimization process. This confirms that the EIR-based infill can select the data source adaptively based on EI reduction to benefit MFSBO most.

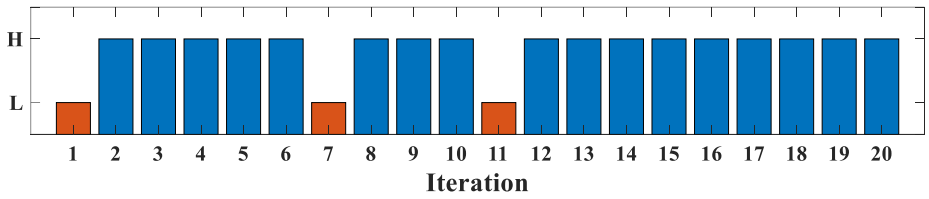


Figure 5.10 Selection of data source (fidelity level) at each iteration for the EIR-based Infill for the Hartmann 4-dimensional Function.

5.2.3 Engineering Example: Design of Inlet Concentrations for Microfluidic Concentration Gradient Generator

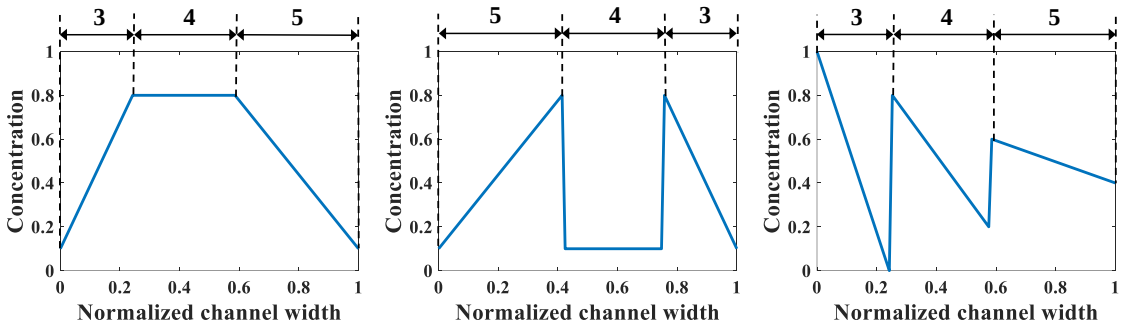
In this example, an engineering example that utilizes the proposed EIR and MFSBO methodology for the design of the μCGG is carried out. μCGG is an important device in biological and chemical processes to generate stable CGs (Irimia, Geba, and Toner 2006; Cabaleiro 2020; C. G. Yang et al. 2011; X. Wang, Liu, and Pang 2017). Numerous μCGGs are proposed to generate CGs of various shapes to meet application requirements. Searching for operating parameters that could yield a user-prescribed CG using a given μCGG device is a formidable challenge (H. Yang et al. 2020; S. H. Hong, Yang, and Wang 2020), which, therefore, will be examined by the proposed method.

In both PBCM and CFD models, the design variables of the optimization problem are the chemical concentrations at the six inlets $x = [c_1, c_2, \ldots, c_6]$. The pressures at the inlet reservoirs are fixed as constants, yielding a six-dimensional problem, whose objective function is defined as J_d - the L_1 norm of the difference between the generated CG ($C_o = f(x)$) and the user-prescribed CG (C_s) in Eq. (5.7) (H. Yang et al. 2020)

$$\min_{x} \qquad J_d = \left\| C_o - C_s \right\|_1 \tag{5.7}$$

Figure 5.11 shows three prescribed CGs, which are trapezoidal, valley-shaped, and sawtooth-shaped. Pressures at inlet reservoirs for these three prescribed CGs are listed in

Table 5.3, and extracted from the previous study (H. Yang et al. 2020). Since these three prescribed CGs consist of segments with different slopes and widths, the pressure magnitudes are also different. Similarly, the four infill strategies above are implemented with two HF (CFD) and two LF (PBCM) initial samples for each prescribed CG. To update MFSM and search for the optimal parameters of inlet concentrations, 30 infill iterations are performed. The computational cost ratio Q is set to be 20, which is computed based on the computational cost ratio of HF-based and LF-based update for this case study using Eq.(5.2). It should be noted that the modeling time during the optimization process varies, because the amount of the LF and HF data increases with the iterations. Therefore, the computational cost ratio value Q above is an estimated average.

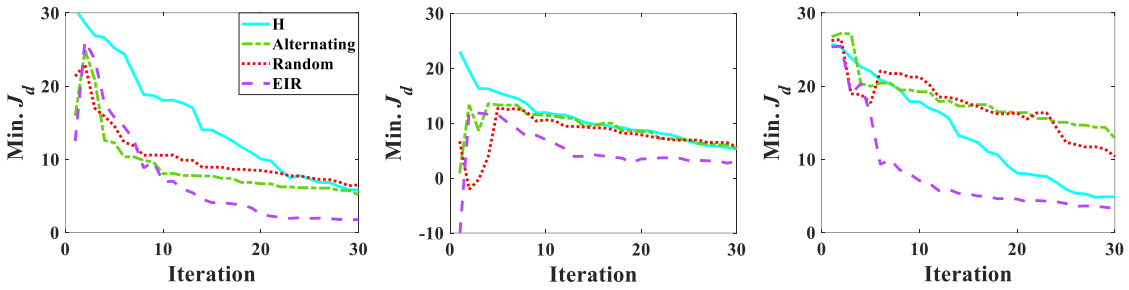


(a) Trapezoidal **(b) Valley-shaped** **(c) Sawtooth-shaped**

Figure 5.11 Prescribed CGs.

Table 5.3 Pressures at inlet reservoirs for three prescribed CGs.

	P_1 (Pa)	P_2 (Pa)	P_3 (Pa)	P_4 (Pa)	P_5 (Pa)	P_6 (Pa)
Trapezoidal	193.99	193.99	257.33	257.33	300.20	300.20
Valley-shaped	352.50	352.50	259.53	259.53	178.62	178.62
Sawtooth-shaped	239.09	239.09	360.60	360.60	478.02	478.02



(a) Trapezoidal **(b) Valley-shaped** **(c) Sawtooth-shaped**

Figure 5.12 Convergence of Min. J_d using different infill strategies for prescribed CGs: (a) trapezoidal, (b) valley-shaped, and (c) sawtooth-shaped.

Figure 5.12 shows the averaged results for the convergence analysis of the minimum J_d (Min. J_d) of MFSM for the four infill strategies. For the trapezoidal and valley-shaped CG, the alternating and random infill converges slightly faster than H-infill. For the sawtooth-shaped, H-infill converges faster than the alternating and random infill. This may be ascribed to the fact that the LF model is relatively less accurate for the sawtooth-shaped CG, and hence, alternating and random infill converges slowly due to inordinate use of the LF data. Under this circumstance, the H-infill, which only adopts HF infill for each iteration, preserves necessary MFSM accuracy and achieves a faster convergence rate. Furthermore, the EIR-based infill properly selects the data source and samples of infill on the as-needed basis by following the EIR criteria, and therefore, it is able to converge even faster to a lower J_d. Quantitatively, the EIR-based infill obtains its full convergence at the 21st, 24th, and 26th iteration for all the three prescribed CGs, respectively, while the other three infill strategies mostly have not converged at the 30th iteration, and more iterations are needed.

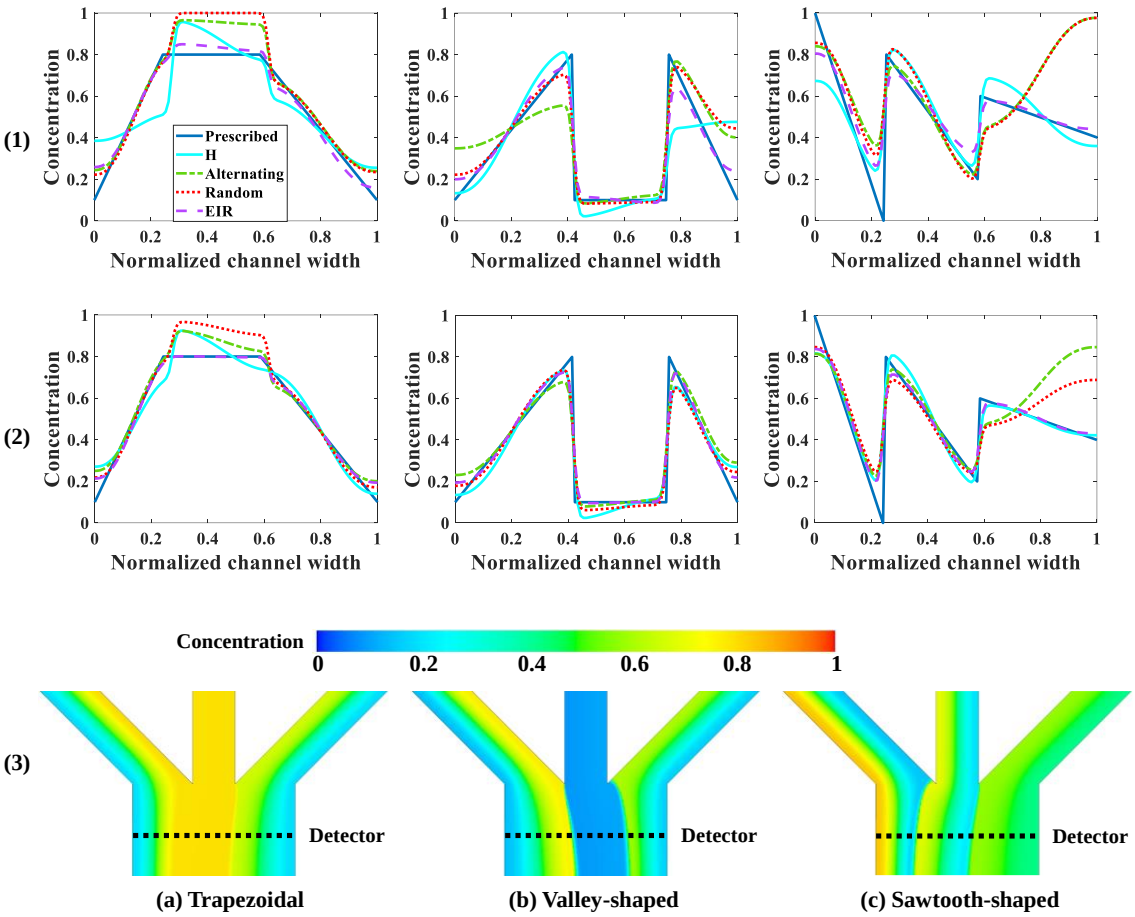


Figure 5.13 Predicted CGs relative to the prescribed CG based on (1) the minimum of MFSM at the 15th and (2) at the 30th iteration, and (3) the CFD contour plots at the 30th iteration for the three prescribed CGs: (a) trapezoidal, (b) valley-shaped, and (c) sawtooth-shaped.

Figure 5.13 illustrates the comparison between the prescribed CGs and the predicted CGs at the 15th and 30th iteration for different infill strategies. The predicted CGs are obtained by supplying the found minimum of MFSM as the inlet concentrations to CFD simulation. Again, each predicted CG is the average of those from the five repeated runs. The bottom row of the figure presents the CFD contour plots at the 30th iteration using the EIR-based infill for three different prescribed CGs. Similarly, the contour plots are extracted from the run with the median performance. The averaged numerical values of J_ds of predicted CGs at the 15th and the 30th iteration (extracted from row 1 and row 2 in Figure

5.13) are listed in Table 5.4. Recall that J_d is the measure of the difference between the predicted CG and the prescribed CG, and a smaller J_d corresponds to better agreement between them. Results clearly show that predicted CGs of the EIR-based infill are closer to the prescribed CGs than those of the other three infill strategies at the 15th iteration with 63.04% J_d reduction on average, indicating a faster convergence rate of the former. Similar findings of optimization performance are observed at the 30th iteration that predicted CGs obtained by the EIR-based infill match the prescribed CGs better with 59.42% J_d reduction on average, viz., faster convergence to lower J_ds.

Table 5.4 J_d at the minimum of the MFSM at the 15th and 30th iteration for the three prescribed CG.

	Infill Strategy	J_d		
		Trapezoidal	Valley-shaped	Sawtooth-shaped
15th Iteration	H	14.19	9.88	12.67
	Alternating	7.71	11.11	17.81
	Random	9.08	9.58	18.57
	EIR	4.56	4.74	5.43
30th Iteration	H	5.84	5.32	4.89
	Alternating	5.48	5.38	13.38
	Random	6.50	5.61	10.34
	EIR	1.65	3.07	3.48

Figure 5.14 shows the selection of the data source (fidelity level) at each iteration using the EIR-based infill for three prescribed CGs. Similarly, the one with the median performance among the five runs is presented. The results show that the EIR-based infill assigns five iterations to LF simulations and 15 iterations to HF simulations. For the trapezoidal CG, a relatively small number of iterations are allocated to the LF infill. This

is because the design for the trapezoidal CG is relatively easy, and fewer LF infills that aim at improving exploration for MFSM are needed. However, it should also be pointed out that even if only a few LF infill iterations are selected, the number of LF samples is actually more than that of HF samples because each LF infill adds Q = 20 LF samples. Take the trapezoidal CG in Figure 5.11 (a) as an example, there are totally 42 LF data and 30 HF data after performing the EIR-based MFSBO. On average, 25% of the computational time is used to run LF infills, while 75% to perform HF infills for the designs of the three prescribed CGs. Besides, similar to previous case studies, LF infills are mostly selected at the beginning of the optimization to explore the design space, and HF infills are performed during the latter half of the optimization to identify the optimum from the results of five runs. For the valley-shaped profile, LF infills are also selected at the latter half of the optimization, which may be attributed to a more complex shape of the local response surface, which, hence, needs more LF samples for enhanced exploration. In conclusion, the EIR-based infill strategy that maximizes the reduction of EI selects the data source and samples in a more adaptive manner and accelerates the optimization process.

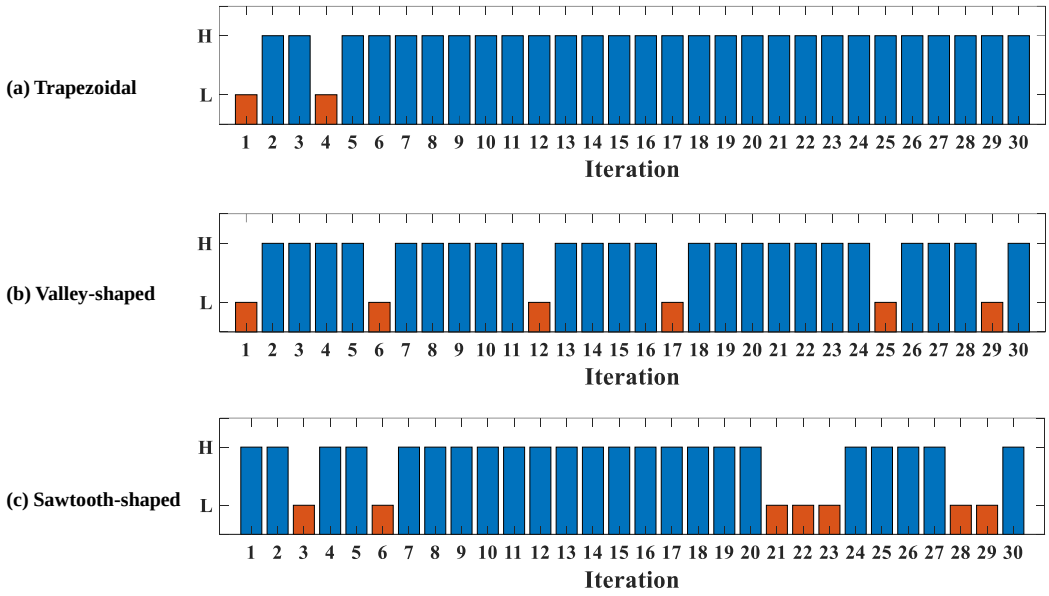


Figure 5.14 Data source selection at each infill iteration for EIR-based Infill.

5.3 Summary

In this chapter, an EIR-based infill strategy is proposed to accelerate the convergence of MFSBO. New aspects of the proposed method include: first, the EIR-based infill method selects the data source (i.e., simulation fidelity level) and samples of infill automatically and judiciously based on the potential of reducing EI through hypothetical interrogation. Second, batch LF sampling is applied within a dynamically varying trust-region, which appreciably improves sampling exploration on an as-needed basis and accelerates the optimization process. Third, given a fixed computational budget for each iteration, the proposed method allows batch LF infills to fully utilize the computational resources because of the lower computational cost of the LF model. Besides, in contrast to existing studies, the present research also takes into account the computational cost incurred for surrogate modeling. Last, the efficiency and accuracy of the proposed method

are thoroughly examined by comparing it against the other three infill strategies (H-only, alternating, and random).

Three case studies are conducted to verify the performance of the proposed method in terms of convergence rate and design accuracy. Results confirm that the proposed EIR-based infill effectively searches within the design space and finds the global minimum in all three case studies. It is able to determine both the sample location and data source in a self-adaptive manner during each iteration, and achieves faster convergence and more accurate design than other infill strategies for all case studies. Specifically, compared to the other three infill strategies, the relative errors of optimal designs found by EIR-based infill decrease by 219.80% and 5.02% on average for the first two numerical case studies. Min. J_d decreases by 59.42% on average for all CGs design in the third engineering case study. Therefore, the EIR-based infill is a more feasible and computation-aware approach for multi-fidelity surrogate-based optimization problems.

CHAPTER 6 A SPARSE MULTI-FIDELITY SURROGATE-BASED OPTIMIZATION METHOD WITH COMPUTATIONAL AWARENESS[4]

[4] Yang, Haizhou, and Yi Wang. To be submitted to *Structural and Multidisciplinary Optimization*.

The CoKriging method, one of the multi-fidelity Gaussian process (GP) surrogate modeling techniques, can be used to handle those data and learn the correlation between them and often utilized in various engineering fields (Marques et al. 2019; Park, Haftka, and Kim 2017; Fernández-Godino et al. 2016; Peherstorfer, Willcox, and Gunzburger 2018). Nevertheless, the GP model suffers from the big data problem, cubic time complexity of $O(n^3)$ and memory storage of $O(n^2)$ (Lawrence, Seeger, and Herbrich 2003), due to the inversion and determinant calculation of the matrix (H. Liu et al. 2020), where n is the number of training samples. This limits the scalability of the GP model and prevents it from the domains with a large dataset.

In order to overcome the scaling problem, various sparse GP approximations were proposed, which are divided into two main categories (H. Liu et al. 2020; Lawrence, Seeger, and Herbrich 2003; Lee, Lee, and Kim 2017; Hensman, Fusi, and Lawrence 2013; Snelson and Ghahramani 2007): global approximations and local approximations. Global approximations simplify the kernel/correlation matrix to achieve scalability, including (1) Subset of Data: the GP model can be approximated by using a subset of the training set (m data points), introducing a smaller kernel matrix, and simplifying the calculation (Chalupka, Williams, and Murray 2013; Das, Roy, and Sambasivan 2018). This method retains the similar accuracy of standard GP, and achieves a lower time complexity $O(m^3)$. (2) Sparse Kernel: the kernel matrix is simplified by removing uncorrelated entries, such that the entry becomes zero when entry exceeds a certain threshold (Melkumyan and Ramos 2009). As a result, only nonzero entries in the kernel matrix are involved in the calculation and that reduces the time complexity to $O(\alpha n^3)$ with $0<\alpha<1$ (3) Sparse Approximation: the full-rank kernel matrix can be replaced by a low-rank approximation through eigenvalue

decomposition (Wilson and Nickisch 2015). The most important features are reserved in the low-rank approximation to retain the accuracy, and the time complexity is reduced to $O(nk^3)$, where k is the number of modes kept in the low-rank approximation. On the other hand, local approximation divides the training data into several subsets (Rullière et al. 2018; Gramacy 2016; H. Wang et al. 2017). For each subset (m_0 data points), a local GP model is built and conquers a subdomain to gain scalability, capturing more local features. Then, the modeling average is employed through a mixture or product of local experts to produce a global prediction. With local approximation, the time complexity is reduced to $O(nm_0^3)$.

Moreover, MFSBO based on Cokriging may also suffer from big data issues. In previous chapter, a computation-aware MFSBO method was developed, which automatically determines both the data source (i.e., the fidelity level HF vs LF) and infill locations by hypothetically estimating the reduction of EI, hence enabling computational awareness. The EIR-based infill takes into account computational costs of both simulations and surrogate modeling, although the latter is often neglected by existing research. However, this approach also suffers from a unique big data issue. That is, for a higher-dimensional complex design problem, a large number of exploration-based LF samples are generated by batch sampling and infilled for surrogate modeling, which increases the sample size dramatically and may eventually compromise MFSBO efficiency.

In order to address the challenges above, this chapter presents a new data sparsification method that balances the exploration and exploitation during MFSBO. It consists of two key components: reduced design space (RDS) and data filtering (DF) to judiciously select data subsets of maximum values for multi-fidelity GP, more specifically, CoKriging, and EIR-based infill strategy in MFSBO. The proposed method aims to remove

the redundant data in multi-fidelity data sets for reduced surrogate modeling time and enhanced optimization efficiency. RDS identifies the region where the optimum is most likely located, and shrinks the design space into a targeted region of a smaller size (reduced design space). Then the data outside RDS, within the non-interested region, is removed, leading to reduced modeling time. On the other hand, DF directly removes redundant LF samples only within the reduced design space. This is because the CoKriging Model when applied to engineering applications, typically has a large number of LF samples and a few HF samples. The LF data is cheaper but less accurate, and may cluster significantly to provide similar or even redundant information to construct the response surface due to the infill process. Therefore, DF is designed for eliminating these LF samples based on certain criteria.

The major novelties of the chapter include: (1) To the best of our knowledge, this chapter presents an initial effort for developing a data sparsification method to address the big data issues in MFSBO, in particular, the one with computational awareness. The data sparsification is performed on an as-needed basis to shrink the modeling time and enhance the optimization efficiency; (2) a two-pronged approach based on different principles is applied before the modeling process to reduce redundant data on the fly. RDS eliminates the data outside the region of interest (the region near the global optimum) to emphasize exploitation, and DF removes the LF samples to mitigate data redundancy in exploration; (3) the proposed RDS&DF method eliminates the bottleneck of the EIR-based infill method (i.e., big data issue) and the novel combination of this data sparsification method with EIR-based infill makes the sparse MFSBO also computation-aware; and (4) previous studies mostly assume the computing load of HF or even LF simulation significantly

dominates over that of surrogate modeling, infill, and optimization, and the latter are neglected. In contrast, this chapter considers a more generic scenario where the latter is comparable to or even more demanding than the HF or LF simulations. This could occur when the aforementioned big data issues emerge, or the MFSBO and HF/LF simulations are conducted on different computing platforms (e.g., due to software, licenses, and hardware constraints). Therefore, in this chapter, the computation time of each constituent piece in MFSBO is taken into account and quantitatively analyzed with respect to the HF or LF simulation time.

6.1 Problem Formulation

Suppose there are two different data generation sources for solving a real-world problem, respectively, denoted in the decreasing order of fidelity as $f_{\mathrm{H}}(\mathbf{x})$ and $f_{\mathrm{L}}(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^d$ is the input variable, d is the number of dimensions of the input variable. $f_{\mathrm{H}}(\mathbf{x})$ represents the HF simulation data generation source with a more accurate response but is computationally expensive. On the contrary, $f_{\mathrm{L}}(\mathbf{x})$ is the efficient LF simulation data generation source with a shorter computational time but low accuracy. The objective of the optimization is to minimize $f_{\mathrm{H}}(\mathbf{x})$ within the design space:

$$\mathbf{x}^* = \min_{\mathbf{x} \in \mathbb{R}^d} f_{\mathrm{H}}\left(\mathbf{x}\right) \tag{6.1}$$

However, direct optimizing based on $f_{\mathrm{H}}(\mathbf{x})$ can be computationally prohibitive because it involves numerous HF simulations, in particular, for the global optimization in the present chapter. Therefore, a CoKriging model $F(\mathbf{x})$ is constructed to approximate $f_{\mathrm{H}}(\mathbf{x})$, which has excellent computational efficiency and can be utilized for accelerating design optimization, i.e.,

$$\mathbf{x}^* = \min_{\mathbf{x}\in\mathbb{R}^d} f_{\mathrm{H}}(\mathbf{x}) \approx \min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) \tag{6.2}$$

6.2 Proposed Methodology

Our prior study proposed an MFSBO framework based on the EIR to enable computational awareness for accurate and efficient design optimization. More specifically, a CoKriging model is constructed to capture the input-output mapping relationship using a small amount of selected simulation data (both HF and LF data) and serves as the main engine for the optimization to reduce the computation cost. A novel EIR-based infill strategy is utilized to determine the data source (fidelity level) and infill samples in each infill iteration to update the model for enhanced accuracy. This MFSBO design optimization method and infill strategy have been verified and exhibited excellent performance in convergence rate and accuracy. However, for a more challenging, higher-dimensional design problem, a large number of iterations are needed to search and locate the global optimum because the data landscape will be more complex. Thus, many infill samples are added to the data set, and the time for surrogate modeling, infill determination, and design optimization all increases exponentially, which compromises the intent of using the surrogate models. To effectively utilize the CoKriging method for MFSBO, a novel data sparsification method that continuously shrinks the targeted design space and filters out redundant data is proposed in this chapter. As shown below, when combined with EIR-based infill, the new sparse MFSBO framework could saliently boost the computation efficiency by curtailing the surrogate modeling time while remaining the design accuracy.

Figure 6.1 illustrates the flowchart of the proposed MFSBO that includes both EIR-based infill and RDS&DF. The details of the procedure are described as follows: First, two

sets of the initial samples are generated by LHS, i.e. $\mathbf{X} = \{ \mathbf{X}^H, \mathbf{X}^L\}$, where $\mathbf{X}$ is the input samples. Subscripts *H* and *L* represent the samples for HF (computationally expensive) and LF (computationally cheap) simulations, respectively. Second, both sets of samples are supplied to corresponding HF and LF simulation engines (Block (1) in Figure 6.1) to generate their responses, i.e., $\mathbf{Y} = \{\mathbf{Y}^H, \mathbf{Y}^L\}$, forming the set of data pairs ($\mathbf{X}$, $\mathbf{Y}$) for surrogate training/modeling. Third, the sample size of the data set is examined with respect to an upper limit of the number of samples, which will determine if data sparsification by RDS&DF needs to be executed to mitigate the big data issues above. If the sample size is smaller than the upper limit, all existing data is used for training the surrogate model. Otherwise, the RDS&DF process (Block (2) in Figure 6.1 and detailed in section 6.2.1) is applied to remove redundant data in the existing data set to shorten modeling time, hence improving optimization efficiency. Typically, at the beginning of the design optimization process, the number of samples is below the limit, and the RDS&DF module will only be triggered after a certain number of iterations. Fourth, a CoKriging model (Block (3) in Figure 6.1) is constructed using the training data of both fidelities to capture the input-output relationship. Then, the EIR-based infill process (Block (4) in Figure 6.1) is invoked to determine where the infill will be and which source/fidelity of the data (HF vs. LF) will be added within the design space iteratively to improve model accuracy and accelerate optimization convergence. Next, the infill samples are supplied to the corresponding simulation engines to generate new data associated with the infill samples to update the model. This process continues until the minimum of the CoKriging converges or the maximum number of iterations is reached. Finally, the CoKriging model in the last iteration

is utilized for design optimization to search the global optimum with accuracy similar to the HF simulation.

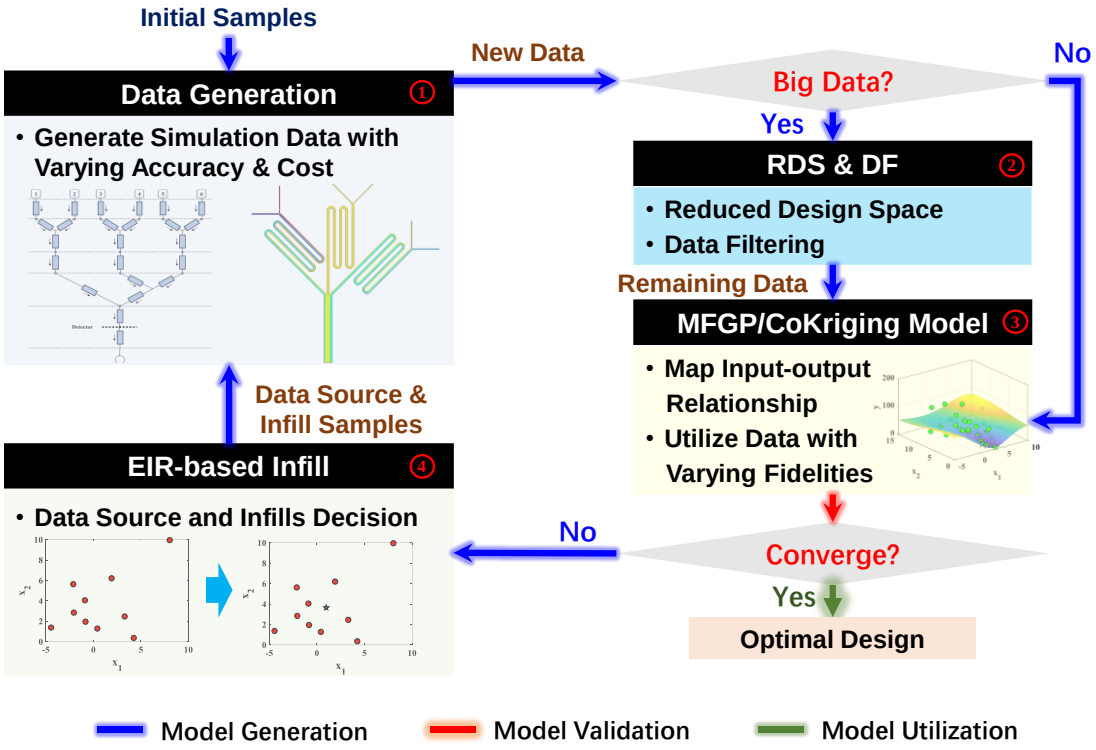


Figure 6.1 Flowchart of MFSBO including both EIR-based infill and RDS&DF.

6.2.1 Reduced Design Space & Data Filtering

The flowchart of the proposed RDS&DF is illustrated in Figure 6.2. If the total sample size (N_{total}) is larger than the user-prescribed upper bound (N_{UB}), viz., too many samples in the data set, RDS&DF will be performed to remove redundant data prior to surrogate modeling to reduce its computation cost. Otherwise, the algorithm will skip this step and execute the surrogate modeling module directly.

Once the sample size reaches the upper bound, the algorithm enters the first component of the module, viz., RDS. As discussed above, RDS shrinks the design space to eliminate the redundant data outside the targeted design space and improve exploitation

when the response surface around the global optimum is well captured and the search approaches the global optimum. RDS component is triggered only when a specific criterion is satisfied, which in the present study is when the convergence rate of the optimization reaches a user-defined threshold. The convergence rate measures the standard deviation of the normalized minimum of the surrogate model in the latest ten iterations and is mathematically defined as

$$\begin{aligned} \mathbf{K} &= \left[y_{\min}^{i-10}, y_{\min}^{i-9}, \cdots, y_{\min}^{i-1} \right] \\ \mathbf{K}' &= \frac{\mathbf{K} - \min(\mathbf{K})}{\max(\mathbf{K}) - \min(\mathbf{K})} \\ \sigma_i &= \text{std}(\mathbf{K}') \end{aligned} \tag{6.3}$$

where y_{min}^{i-1} is the minimum of the surrogate model at the $(i\text{-}1)^{\text{th}}$ iteration while the other nine follow the same notation; $\mathbf{K}$ is a vector that records the minimum of the CoKriging model in the previous ten iterations. $\mathbf{K}'$ is the normalized version of $\mathbf{K}$; σ_i is the standard deviation of $\mathbf{K}'$. If σ_i is smaller than a threshold (σ_{lim}), it is considered that the minimum of the surrogate model is converging to a specific region, and therefore, RDS is applied. Otherwise, the minimum keeps varying from one iteration to the next, indicating that the search is still far away from the global optimum, and thus, it is not appropriate to shrink the design space, and RDS should be skipped. In this study, an empirical value $\sigma_{lim} = 0.25$ is adopted. The detailed procedure of RDS is given as follows:

First, the center and the half-length of the RDS are determined to identify the location and the range of the RDS (Block (a) in Figure 6.2), which is defined as

$$
\begin{aligned}
\mathbf{RDSL}_i &= \max\{\mathbf{x}_{i-1}^{lb}, \mathbf{x}_{\min}^{i-1} - \mathbf{l}_i\} \\
\mathbf{RDSU}_i &= \min\{\mathbf{x}_{i-1}^{ub}, \mathbf{x}_{\min}^{i-1} + \mathbf{l}_i\} \\
\mathbf{l}_i &= \gamma_i \times \left(\mathbf{x}_{i-1}^{ub} - \mathbf{x}_{i-1}^{lb}\right) \\
\gamma_i &= \left(2 - \gamma_{LB}\right)^{\frac{\sigma_i}{\sigma_{\lim}}} - (1 - \gamma_{LB}) \\
\gamma_{LB} &= \sqrt[k]{\xi_{LB}}
\end{aligned}
\tag{6.4}
$$

where **RDSL** and **RDSU** are the lower bound and upper bound of the RDS, respectively. $\mathbf{x}_{i-1}^{lb}$ and $\mathbf{x}_{i-1}^{ub}$ are the lower bound and upper bound of the design parameter space in the i-1$^{\text{th}}$ iteration. $\mathbf{x}_{\min}^{i-1}$ is the coordinate of the design variables corresponding to the minimum of the surrogate model $y_{\min}^{i-1}$ in the last iteration (i-1$^{\text{th}}$ iteration). $\mathbf{l}_i$ is the half-length of RDS and also determines the area or the volume of the region. γ_i is a shrinkage ratio measuring the contracted length scale of the design space at the i^{th} iteration. A larger γ_i indicates that the minimum of the surrogate model still varies dramatically, and the search does not converge to one location, and thus, the design space cannot be shrunken much. On the other hand, a smaller γ_i represents a better convergence to the minimum of the surrogate model, and it is reliable to shrink the design space more, resulting in a shorter $\mathbf{l}_i$ and a smaller region with a high probability to accommodate the global optimum. γ_{LB} is the lower bound of the shrinkage ratio for each dimension. γ_{LB} is determined by ξ_{LB}, and ξ_{LB} prescribes the lower bound of the shrinkage ratio of the hypercube volume enclosed by RDS. A larger ξ_{LB} makes a conservative reduction of the design space, hence avoiding missing the global optimum, while a smaller one will remove more data to further reduce the surrogate modeling time and improve optimization efficiency but at the risk of missing the global optimum. A value of $\xi_{\text{LB}} = 0.7$ is appropriate to balance from both perspectives and used for all case studies in the chapter. k is the dimension of the problem. The fourth equation in Eq. (6.4) calculates the length ratio based on the standard deviation calculated in Eq.

(6.3) followed by an exponential scaling, and the former then can be used to determine the bounds of RDS in each dimension. The last equation in Eq. (6.4) transfers the lower bound of the volumetric shrinkage ratio to the lower bound of the shrinkage ratio for each dimension.

Second, all the samples out of the RDS (Block (b) in Figure 6.2), including HF and LF samples, are removed to improve the surrogate modeling efficiency and preserve the design accuracy.

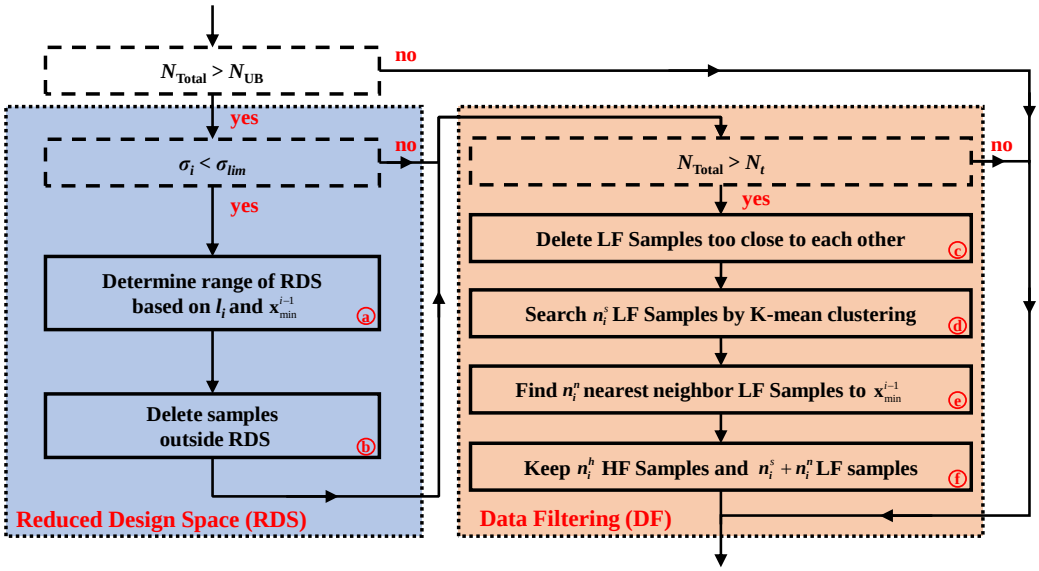


Figure 6.2 Flowchart of reduced design space and data filtering.

Following RDS, the DF step is applied to further remove redundant data within RDS when a specific criterion is satisfied, that is, the current sample size is larger than the target sample size (N_t). The target sample size (N_t) is a user-defined parameter representing the number of total samples following the RDS&DF module. N_t is determined empirically prior to the MFSBO and is subjected to the constraints of the physical memory of the computing platform and the computational time of surrogate modeling and infill relative to the HF simulation. Once the criterion is satisfied, DF filters out the potentially redundant

data. In this process, only LF samples will be removed and all the HF samples will be kept because the latter are limited in quantity, more costly to collect, and more accurate than the former. The details of the DF step are described as follows: First, the LF samples too close to each other, are removed to improve the numerical conditioning of the covariance matrix for enhanced model accuracy (Block (c) in Figure 6.2), yielding n_i^l LF samples in total. In our DF, all HF samples ($\boldsymbol{n}_i^h$) are kept. Thus, the total budget for LF samples is N_t - $\boldsymbol{n}_i^h$, of which n_i^s and $\boldsymbol{n}_i^n$ will be selected based on exploration and exploitation measures (see below), that is n_i^s and $\boldsymbol{n}_i^n$ LF samples will be drawn from the n_i^l LF samples above with better numerical conditioning. A user-defined parameter $r_1 = n_i^s /(n_i^s + \boldsymbol{n}_i^n)$ is used to determine the ratio between n_i^s and $\boldsymbol{n}_i^n$ to balance the exploration and the exploitation. In this study, an empirical value $r_1 = 0.7$ is adopted for all case studies. Next, n_i^s LF samples will be determined based on the clustering technique (Block (d) in Figure 6.2). That is, all the existing LF samples will be grouped into n_i^s clusters, and for each cluster, only one LF sample is selected, retaining totally n_i^s LF samples that still maintain sample exploration. Furthermore, the distances of all remaining LF samples ($n_i^l - n_i^s$) to $\mathbf{x}_{\min}^{i-1}$ are measured, and the top $\boldsymbol{n}_i^n$ LF samples with the shortest distance are chosen for exploitation-based sampling (Block (e) in Figure 6.2). Eventually, all $\boldsymbol{n}_i^h$ HF samples and $n_i^s + \boldsymbol{n}_i^n$ LF samples (totally $N_t = \boldsymbol{n}_i^h + n_i^s + \boldsymbol{n}_i^n$) are kept and used for surrogate modeling (Block (f) in Figure 6.2).

6.3 Case Studies

Two design optimization case studies, including one numerical example and one engineering example, are conducted, and their results are compared with solely EIR-based infill without RDS&DF. For a fair comparison, the initial samples and computational cost ratio are all kept the same. Besides, each optimization is repeated five times, and the average and standard deviation are presented in this study to evaluate the statistical characteristics.

6.3.1 Numerical Example: POWELL

In this case study, a POWELL benchmark function with eight-dimensional input variables is adopted to examine the proposed method. Both HF and LF models are given in Eq. (6.5) (Laguna and Marti 2005).

$$\begin{aligned} f_{\mathrm{H}} &= \sum_{i=1}^{2}\left[\left(\mathbf{x}_{4i-3}+10\mathbf{x}_{4i-2}\right)^2+5\left(\mathbf{x}_{4i-1}-\mathbf{x}_{4i}\right)^2+\left(\mathbf{x}_{4i-2}-2\mathbf{x}_{4i-1}\right)^4+10\left(\mathbf{x}_{4i-3}-\mathbf{x}_{4i}\right)^4\right] \\ f_{\mathrm{L}} &= 1.2\,f_{\mathrm{H}}\left(\mathbf{x}-\begin{bmatrix}0.3 & -1.1 & 1.5 & 0.8 & -2 & 0.6 & 0.1 & -1.1\end{bmatrix}\right)-1113 \\ \mathbf{x}_i &\in [-4,5] \end{aligned} \tag{6.5}$$

The global optimum of the POWELL function is at $\boldsymbol{x}_i^* = 0$ and the corresponding function value is $f_{\mathrm{H}}(\mathbf{x}_i^*) = 0$. First, 100 LF and 10 HF initial samples are generated for constructing the initial CoKriging model. 100 infill iterations are then performed using the EIR-based infill to update the CoKriging and search the global optimum continuously. The computational cost ratio is set to be 20. The prescribed upper bound of the sample size N_{UB} and the target sample size N_{t} are set to be 315 and 285, respectively. Figure 6.3 reveals the convergence of the EIR-based infill with and without RDS&DF. The curves represent the

average of the five runs, and the shaded areas indicate the 95% confidence interval. The optima found in both methods oscillate dramatically at the beginning because of the poor accuracy of the initial CoKriging model. As more samples are added, both of them gradually reach the global minimum, and the confidence intervals of the two methods continuously shrink, indicating their stable convergence. Besides, the EIR-based infill with RDS&DF exhibits a slightly faster convergence to a point closer to the global optimum.

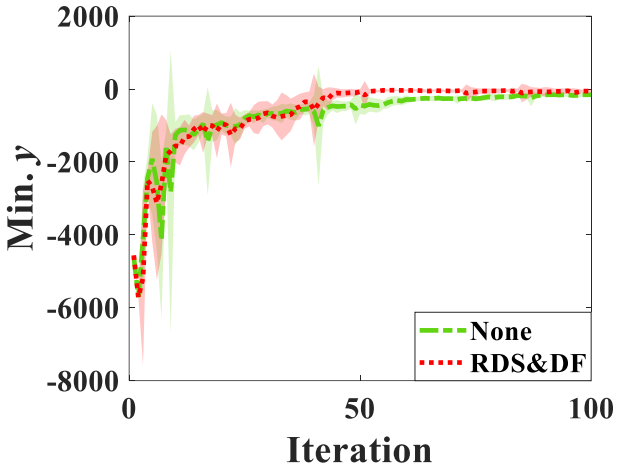


Figure 6.3 Convergence of the minimum of CoKriging using EIR with and without RDS&DF.

Figure 6.4 portrays the selection of the data source (fidelity level) at each iteration for the EIR-based infill with and without RDS&DF, and the results are collected from the run with the median performance among the five runs above. In both cases, more computational time is spent on HF simulations, which is more accurate for design optimization to locate the minimum. There are 31 infill iterations assigned to LF for the EIR-based infill without RDS&DF, and 23 LF infill iterations for the one with RDS&DF. Besides, more LF infills are performed at the beginning for enhanced exploration, while HF infills are dominant in the latter half to utilize exploitation more when it starts to

approach the optimum, which confirms that the EIR-based infill automatically allocates the computational resource as needed to accelerate design optimization maximally.

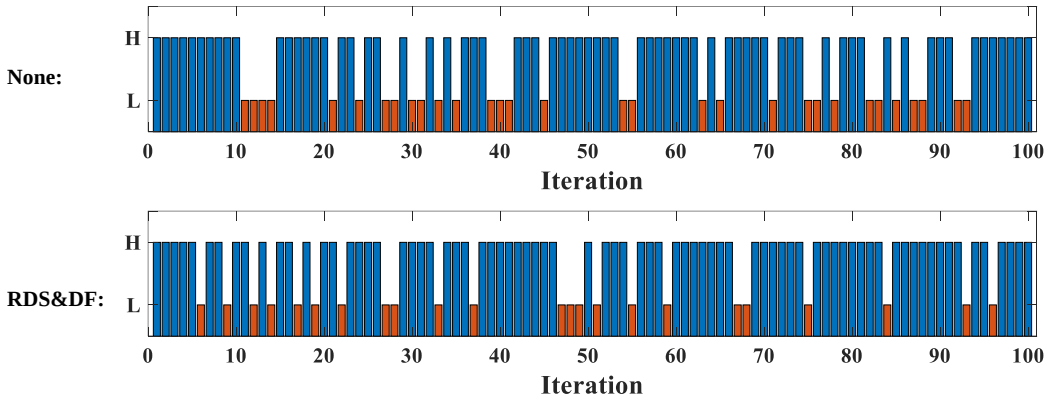


Figure 6.4 Selection of data source (fidelity level) at each iteration for the EIR-based infill with and without RDS&DF for the POWELL function.

Figure 6.5 shows RDS&DF application, the volume ratio ξ of RDS to the initial design space, and the number of HF and LF data at each iteration. Similarly, the run with the median performance among the five runs is presented. RDS&DF is triggered five times at the 33rd, 59th, 68th, 80th, and 96th iterations as shown in Figure 6.5(a) and (b), where RDS and DF are, respectively, colored in blue and purple. At the 33rd, 59th, and 80th iterations, only RDS is invoked to remove redundant samples. There are several interesting observations. First, after RDS, the sample size drops below the target sample size Nt = 285, and thus, the DF step is not necessary. Both RDS and DF are employed to eliminate redundant samples at the third and fifth use of the RDS&DF module (i.e., the 68th and 96th iteration), because even after RDS, the samples size still exceeds the target number. Therefore, DF further removes the less valuable LF data within the design space. Figure 6.5(c) shows that the volume of the RDS shrinks every time the RDS module is triggered, resulting in a smaller and smaller design space to search the global optimum and thus

accelerating the optimization. Figure 6.5(d) presents the number of both HF and LF samples at each iteration. Results show that both HF and LF redundant data are removed when the RDS&DF is invoked. In general, the numbers of both HF and LF samples fluctuate a lot at the beginning. Then, the number of LF data becomes almost saturated after the 50th iteration, while the number of HF data keeps increasing especially when it approaches the end of optimization, indicating that the ratio of the HF to LF data keeps ramping up at the latter half of the optimization. This is because the HF data provide more accurate information for surrogate modeling and is more beneficial for exploitation, it is preserved during the latter half of the optimization to locate the global optimum.

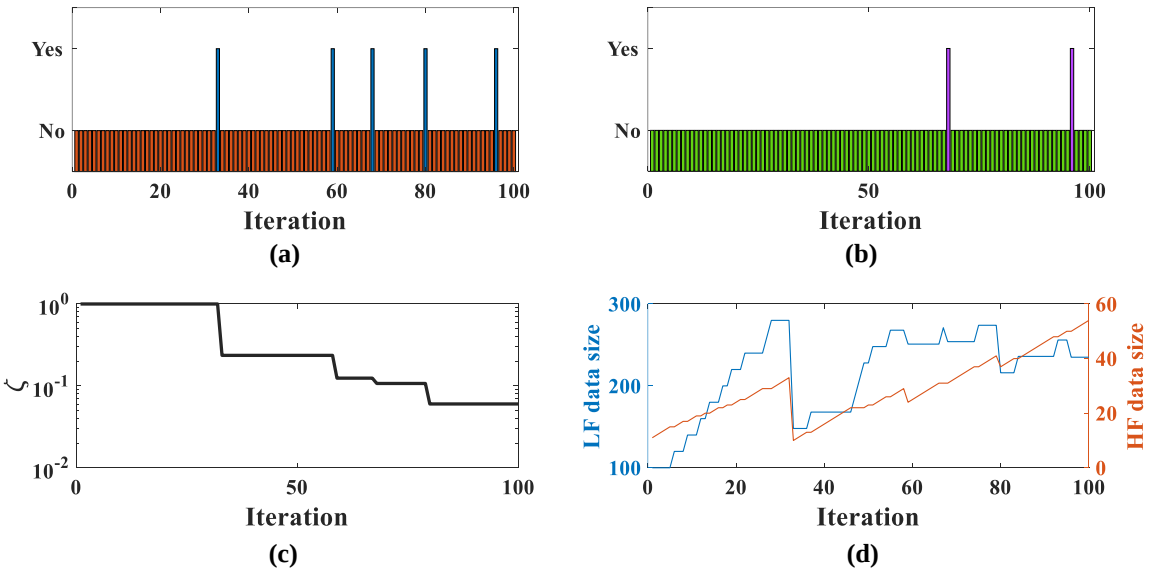


Figure 6.5 (a) RDS application, (b) DF application, (c) volume ratio ξ of RDS, and (d) number of HF and LF data at each iteration.

The average of the computational time in the five runs (including the time for surrogate modeling, EIR-based infill, both HF and LF simulations, and CoKriging

optimization), and the average of the found optimum are listed in Table 6.1. Results show that although RDS&DF reduces the data size used in surrogate modeling, it is slightly faster in optimization convergence and reaches a closer vicinity of the global minimum with the same number of iterations relative to that without RDS&DF. This may be attributed to the fact that the design space is reduced and the infill samples are used to only explore the shrunken region. Since the problem in this case study is a benchmark function, the function evaluation time is negligible, and thus, the total computation time and the computation time excluding simulation are almost the same. Due to RDS&DF, the computational time is shrunken significantly by 69.49%. This again confirms that the RDS&DF module can effectively eliminate the redundant data to accelerate the surrogate modeling and optimization process and preserve and even improve optimization efficiency.

Table 6.1 The computational time of the optimization and the minimum of CoKriging found by the optimization.

	None	RDS&DF
Computational time (s)	22007.53	6714.19
Computational time (s) (Exclude Simulation Time)	22007.47	6714.18
$f_H(\mathbf{x}_{min})$	64.75	24.71
$f_H(\mathbf{x}^*)$	0	

6.3.2 Engineering Example: Microfluidic Concentration Gradient Generator Design

In this section, an engineering problem of μCGG design optimization is investigated to further verify the feasibility of the proposed RDS&DF method. μCGG is an important biological device to generate and maintain stable CGs that are widely

employed in biological processes (Irimia, Geba, and Toner 2006; Cabaleiro 2020; C. G. Yang et al. 2011; X. Wang, Liu, and Pang 2017), such as immune response, wound healing, embryogenesis, cancer metastasis, etc. Numerous research efforts focused on fabricating novel μCGGs to stably generate various CGs for different engineering applications. However, determining values of operational parameters for μCGGs to generate a CG profile with the best agreement with the user-prescribed one is still a challenge (H. Yang et al. 2020; S. H. Hong, Yang, and Wang 2020), and MFSBO with the EIR-based infill was developed to tackle such a challenge. As mentioned before, MFSBO with the EIR-infill also suffers from the big data problem, making it forbidden from the design of μCGG with high dimensional input parameters. Therefore, the proposed RDS&DF method in convert with the MFSBO and EIR-based infill is examined on a nine-dimensional μCGG design problem in the present work.

6.3.2.1 Description of μCGG Design Optimization Problem

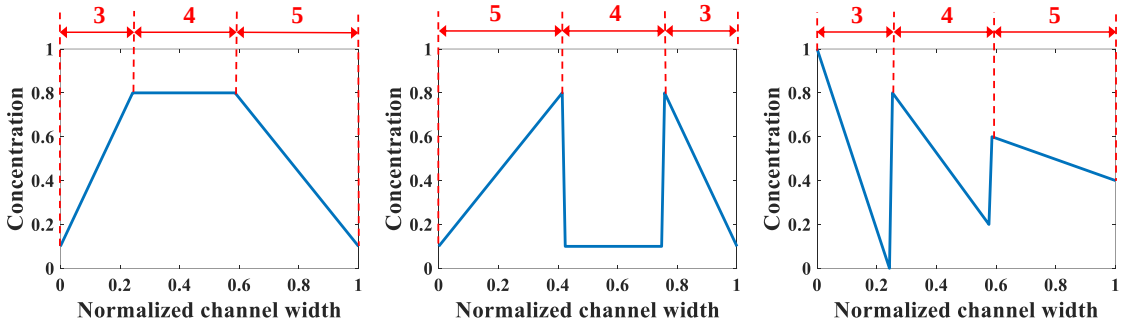
The HF and LF models for this engineering problem are defined as

$$\begin{aligned} f_{\mathrm{H}} &= \left\| \mathbf{c}_{\mathrm{H}}\left(\mathbf{x}\right) - \mathbf{c}_{\mathrm{s}} \right\|_1 \\ f_{\mathrm{L}} &= \left\| \mathbf{c}_{\mathrm{L}}\left(\mathbf{x}\right) - \mathbf{c}_{\mathrm{s}} \right\|_1 \end{aligned} \tag{6.6}$$

where $\mathbf{c}_{\mathrm{H}}(\mathbf{x})$ and $\mathbf{c}_{\mathrm{L}}(\mathbf{x})$ represent the CGs predicted by HF CFD simulation and LF PBCM simulation at sample point $\mathbf{x}$, and $\mathbf{c}_{\mathrm{s}}$ is the user-prescribed CG (H. Yang et al. 2020). The optimization problem is to minimize the discrepancy (defined as J_d) between the CFD-predicted CG and the prescribed CG following Eq. (6.1). The design variables include the normalized chemical concentrations at the six inlets and the pressure differences across the three mixing channels, i.e., $\mathbf{x} = [\Delta p_1, \Delta p_2, \Delta p_3, c_1, c_2, \ldots, c_6]$, yielding a nine-dimensional

problem. In order to visualize the design accuracy, optimal design parameters $\mathbf{x}^*_{\text{RDS\&DF}}$ or $\mathbf{x}^*_{\text{None}}$ found by MFSBO with or without RDS&DF are supplied to CFD to generate corresponding CGs, i.e., $\mathbf{c}_{\text{H}}\left(\mathbf{x}^*_{\text{RDS\&DF}}\right)$ and $\mathbf{c}_{\text{H}}\left(\mathbf{x}^*_{\text{None}}\right)$, both of which will be compared with the prescribed one $\mathbf{c}_{\text{s}}$.

Three different prescribed CGs are considered in this section as shown in Figure 6.6. Each represents a different case study, and thus, the μCGG design optimization can be thoroughly investigated.



(a) Trapezoidal **(b) Valley-shaped** **(c) Sawtooth-shaped**

Figure 6.6 Prescribed CGs: (a) trapezoida; (b) vally-shaped; and (c) sawtooth-shaped.

6.3.2.2 Result and Discussion

For each prescribed CG, 150 LF PBCM and 10 HF CFD simulations are performed to generate initial training data for CoKriging modeling. Since the initial CoKriging is not accurate, 150 EIR-based infill iterations are added during the design optimization to improve the model and design accuracy. The computational ratio is set to be Q = 20 in this case. The prescribed upper bound of the sample size N_{UB} and the target sample size N_{t} are set to be 500 and 470, respectively. Figure 6.7 depicts the convergence of the minimum of

CoKriging (Min. J_d) for two methods, which again are the averaged results from five runs. For all three prescribed CGs, both MFSBO with and without RDS&DF exhibit similar convergence rates and eventually converge to designs with similar accuracy (J_d). Moreover, the confidence intervals of both methods are also similar, confirming their consistent design performance regardless of RDS&DF.

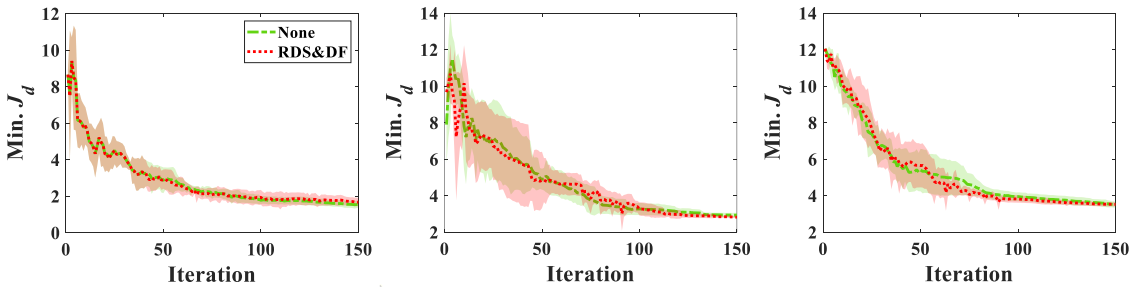


Figure 6.7 Convergence of the minimum of CoKriging with and without RDS&DF.

Figure 6.8 presents the selection of the data source (fidelity level) at each iteration for the EIR-based infill with and without RDS&DF. Similarly, the one with the median performance among the five runs is presented. In all case studies, the number of the HF infill exceeds that of LF infill during design iterations. Besides, the LF infill iterations are mostly selected at the beginning of the optimization to explore the design space, and the HF infills are utilized during the latter half of the optimization to locate the global optimum through exploitation. In summary, the EIR-based infill strategy intelligently selects the data source and sample locations to accelerate the optimization process.

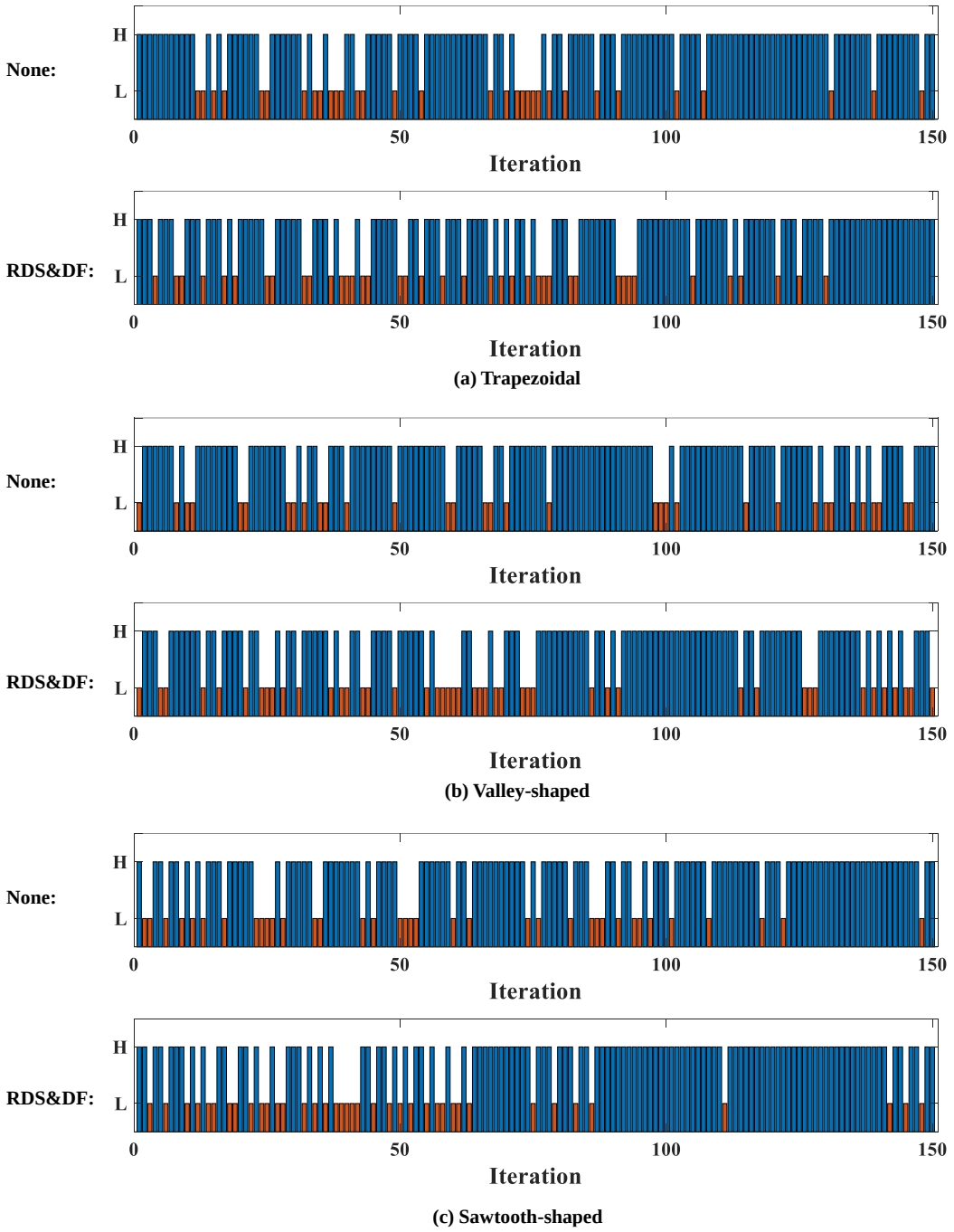


Figure 6.8 Selection of data source (fidelity level) at each iteration for the EIR-based Infill with and without RDS&DF for μCGG design.

Figure 6.9 shows RDS&DF applications, the volume ratio ξ of RDS (viz, the volume of RDS to that of the whole initial design space), and the number of HF and LF

data at each iteration. Again the one with the median performance among the five runs is presented. It clearly shows that RDS&DF is applied nine times on average for all three prescribed CGs. For the initial uses of RDS&DF, only the RDS step is executed. This is because at the beginning the sample size after RDS is less than the targeted value. Later, both RDS&DF steps are performed to further sparsify the data, including both HF and LF data outside RDS and the LF data within the RDS, to shorten the surrogate modeling time and accelerate the optimization process. Besides, through the applications of RDS, its volume decreases dramatically for all cases as shown in all subplots (3) in Figure 6.9. The contraction of the design space leads to a more focused and effective search within the small region of interest and faster localization of the global optimum. Furthermore, subplots (4) in Figure 6.9 show that the numbers of both HF and LF data have large fluctuations at the beginning due to the RDS&DF module. After the 70^{th} iteration, the optimization algorithm seems able to localize the global optimum within RDS, and the HF data is more needed for more exploitation because of higher accuracy. It is even more striking to observe that as the optimization approaches the optimum, the number of LF data actually slightly decreases, indicating that given a fixed budget of the sample number, our RDS&DF method can intelligently enrich HF data by eliminating LF data to maximize the exploitation at the later stage of MFSBO.

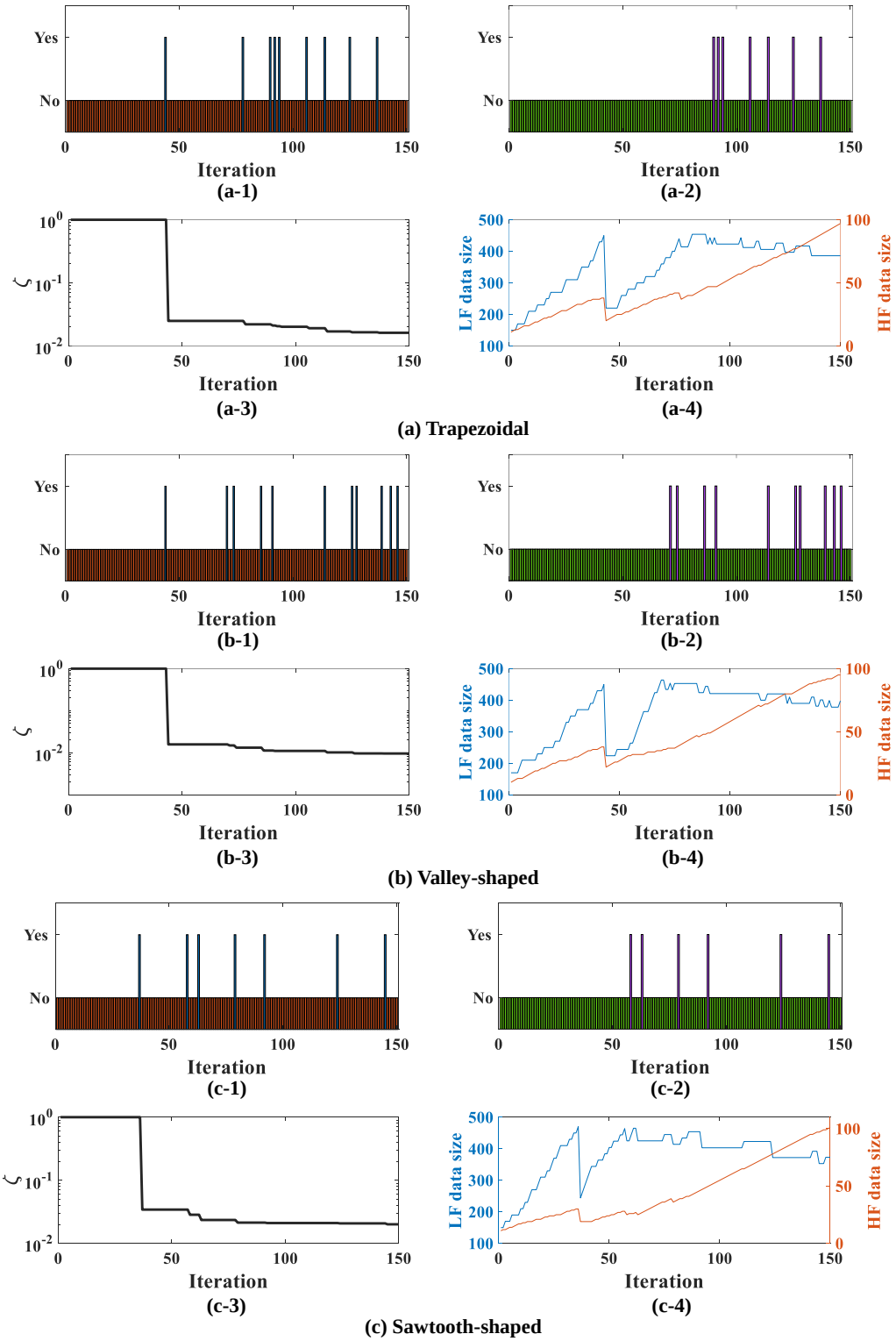


Figure 6.9 (1) RDS application, (2) DF application, (3) volume ratio ξ of RDS, and (4) number of HF and LF data at each iteration for three prescribed CGs: (a) trapezoidal, (b) valley-shaped, and (c) sawtooth-shaped.

Table 6.2 The computational time of MFSBO, min. J_d corresponding to the found minimum, and the ratio of reduction in computation time.

		Computational time (s)	Computational time (s) (Exclude Simulation Time)	Min. J_d
Trapezoidal	None	169264.70	159719.56	1.55
	RDS&DF	41980.74	31473.54	1.67
	Reduction (%)	75.20	80.29	
Valley-shaped	None	118556.58	107887.48	2.94
	RDS&DF	44362.96	33573.91	2.84
	Reduction (%)	62.58	68.88	
Sawtooth-shaped	None	111730.99	100818.88	3.52
	RDS&DF	38596.73	27631.91	3.53
	Reduction (%)	65.46	72.59	
Average Reduction (%)		**67.74**	**73.92**	

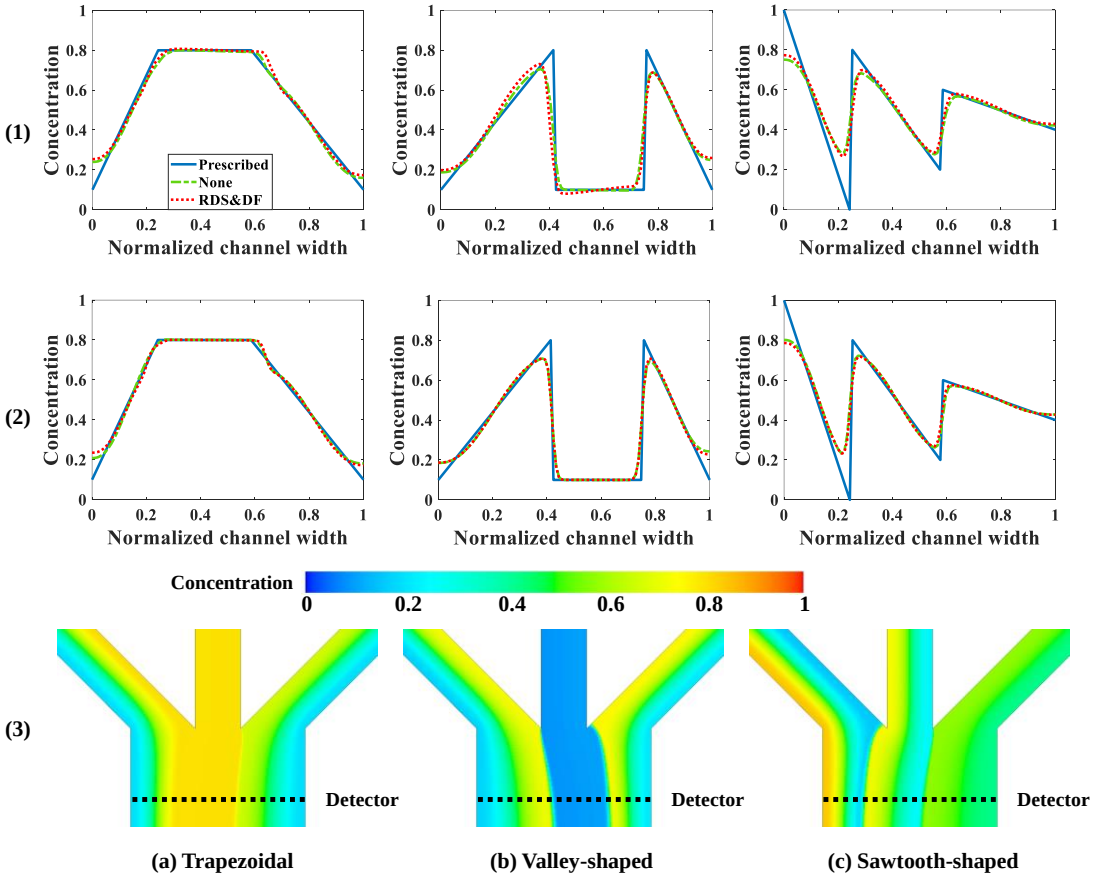


Figure 6.10 Predicted CGs relative to the prescribed CG based on (1) the minimum of CoKriging at the 75th and (2) at the 150th iteration, and (3) the CFD contour plots at the 150th iteration for the three prescribed CGs: (a) trapezoidal, (b) valley-shaped, and (c) sawtooth-shaped.

Table 6.2 lists the average computational time of the optimization and the average Min. J_d obtained by the design optimization. There are several points of note: (1) in the case without RDS&DF, the computational time of the surrogate modeling and design optimization (excluding CFD simulation) occupies 91.87% of the total time on average for the three prescribed CG profiles. Therefore, it is expected that removing redundant data will significantly boost the computational efficiency of the entire process, which also justifies the need for developing the proposed RDS&RF. (2) Since the time of each CFD simulation in general is fixed, it would be more meaningful to reduce the surrogate modeling through RDS&DF. As shown in the table, RDS&DF greatly curtails the surrogate modeling and design optimization time (excluding CFD simulation) and the total computational time by 73.92% and 67.74%, respectively. Besides, the optimal design found by the EIR-based infill with RDS&DF exhibits similar J_ds as the one without RDS&DF, viz., almost the same discrepancy from the prescribed CG. Figure 6.10 shows the comparison between the prescribed CGs and the predicted CGs at the 75^{th} and 150^{th} iteration for two methods. More specifically, the optimal design parameters (inlet concentration and pressure difference) found by the MFSBO with or without RDS/DF are supplied to the CFD simulation as the boundary conditions. The predicted CGs are then extracted from the CFD simulation results. The bottom row of the figure is the CFD contour plots at the 150^{th} iteration using design parameters determined by the EIR-based infill with RDS&DF. Similar to the above, the contour plots are extracted from the run with the median performance. Results clearly show that predicted CGs of both methods are almost the same and resemble the prescribed CGs at both the 75^{th} and 150^{th} iterations. These

observations verify that RDS&DF can shrink the surrogate modeling and infill time, hence improving optimization efficiency without sacrificing design accuracy.

6.4 Summary

This chapter presents a new data sparsification method utilizing exploitative RDS and explorative DF for MFSBO to reduce the complexity and time of surrogate modeling and infill, while preserving design accuracy. By combining RDS&DF with EIR-based infill, a sparse MFSBO with computational awareness is achieved, enabling not only automated determination of the data source during infill, but also intelligent enrichment or dilution of the data of varying fidelities on an as-needed basis. New aspects of the proposed method include: first, its utility to address the big data issue is examined and established for computation-aware MFSBO. Second, RDS tackles the challenge from the exploitation perspective by removing samples outside the RDS containing the global optimum. On the other hand, DF only eliminates LF samples of less accuracy that provide repetitive information while retaining both explorative and exploitative characteristics of LF data. Third, a thorough analysis is conducted to compare the design optimizations with and without RDS&DF in terms of design accuracy and efficiency (computation time and convergence rate).

Two case studies, an eight-dimensional benchmark function and a nine-dimensional μCGG design problem, are performed to examine the performance of the proposed method. Results show that the proposed RDS&DF significantly reduces the modeling time of CoKriging and thus accelerates the entire optimization process. The EIR-based infill technique developed by the authors recently to enable computation awareness

of MFSBO is rendered more applicable to higher-dimensional problems. More importantly, in combination with EIR-based infill, RDS&DF is even able to tailor the ratio of the already existing HF and LF samples on an as-needed basis during MFSBO, which to the best of our knowledge has not been reported before. Besides, the optimal designs obtained with RDS&DF exhibits similar or even better accuracy than those without RDS&DF, confirming its salient optimization efficiency and accuracy. Specifically, the overall computation time is reduced by 69.49% and 73.92% for two case studies, respectively. Therefore, data sparsification based on RDS&DF in conjunction with EIR-based infill is a promising approach for achieving efficient computational-aware MFSBO.

CHAPTER 7 MULTI-FIDELITY REDUCED-ORDER MODEL FOR GPU-ENABLED MICROFLUIDIC CONCENTRATION GRADIENT DESIGN[5]

[5] Yang, Haizhou, Seong Hyeon Hong, Gang Wang, and Yi Wang. Submitted to *Engineering with Computers*, 12/16/2021

Traditional global design optimization methods for μCGGs solely use HF CFD simulation and search for optimal parameters that can generate a CG with the best agreement with the prescribed CG. However, GA, as a global optimization method, requires a large number of CFD evaluations, and hence, is computationally prohibitive. To overcome this issue, (MF)SBO for μCGG design is proposed in previous chapters. That is, first a Kriging-based or Cokriging-based surrogate modeling method constructed with a small amount of selected simulation data, is used to approximate the response surface by capturing the relationship between the input and output data pairs. Therefore, the demand for HF CFD simulation is reduced in the (MF)SBO method. Second, once the (MF) surrogate model is built, the optimization can be performed on the surrogate model and the computation time would be largely curtailed since the surrogate model comprises elementary functions, and the time of its evaluation is extremely low. However, the issue associated with the (MF)SBO method is that the infill sampling is performed during optimization, which needs to run HF CFD simulation and train a new surrogate model. More importantly, the surrogate model for optimization and infill depends on prescribed CGs, and therefore, for different prescribed CGs, the entire optimization and infill process needs to be repeated. This is inefficient when many prescribed CGs are considered.

To overcome such an issue, this chapter proposes an MFROM framework for μCGG design. It combines both the low-fidelity PBCM and the high-fidelity CFD simulation data and makes use of their complementary merits for enhanced computation efficiency. The dimensions of PBCM and CFD data are reduced by POD, a low-rank matrix approximation technique. The fidelity gap between them is then bridged through a Kriging model within the modal subspace. The compact and portable nature of MFROM enables

ultrafast evaluation speed and global optimization of μCGG design on a highly parallelized graphics processing unit (GPU) platform.

The novelties of the present research include: (1) An MFROM is proposed to combine PBCM and CFD simulation and learn the relationship between them in POD modal subspace. It harnesses the merits of both, hence achieving an excellent balance between the computational cost and model accuracy; (2) in contrast to our prior effort (H. Yang et al. 2020) where surrogate modeling and optimization are interwoven into a single process and the model needs to be regenerated for each new prescribed profile, the present methodology separates them into two stages. The MFROM broadly applicable within the entire parameter space is constructed offline with more powerful computing resources. Global optimization that incorporates the generated MFROM as the simulation engine can be achieved very rapidly (even on low-end computing devices) due to MFROM's compact sizes and composition of elementary functions. Such a new method enables the use of different computing platforms for modeling and optimization, salient portability and deployability of the optimization module in the resource-limited environment, and repetitive simulation of MFROM with different parameters to match any prescribed CG profiles without the need for additional CFD simulations; and (3) to the best of our knowledge, the present research represents an initial effort that integrates POD for data dimensionality reduction, PBCM and CFD for multi-fidelity data generation, and Kriging model for fidelity fusion to achieve salient μCGG design in an automated manner. Owing to POD, the global optimization can be performed within the modal subspace rather than the physical CG domain.

7.1 Problem Description

The objective is to construct an MFROM for optimization to determine the operating parameters that generate a CG with the best agreement with a user-prescribed CG. The CG (**c**) at any location within the μCGG can be represented by (Y. Wang, Mukherjee, and Lin 2006)

$$\begin{aligned} &\mathbf{c} = \mathbf{F}_{\text{fs}}\mathbf{d} \\ &\mathbf{F}_{\text{fs}} = \begin{bmatrix} 1 & \cdots & \cos(n\pi \dfrac{0\Delta y}{w}) \\ \vdots & \ddots & \vdots \\ 1 & \cdots & \cos(n\pi \dfrac{m_{\text{d}}\Delta y}{w}) \end{bmatrix} \\ &\mathbf{d} = \begin{bmatrix} d_0 & d_1 & \cdots & d_n \end{bmatrix}^{\text{T}} \end{aligned} \tag{7.1}$$

where $\mathbf{F}_{\text{fs}}$ is the discretized Fourier basis matrix; $m_{\text{d}}+1$ is the number of grid points at the detector location to resolve the CG, and $n+1$ is the number of Fourier coefficients d_n (or Fourier series terms) used to reconstruct CG from the coefficient d_n.

The objective function of this optimization problem is defined as

$$\min_{x} \qquad J_d = \left\| \mathbf{c} - \mathbf{c}_{\text{s}} \right\|_2 \tag{7.2}$$

where J_{d} is the L_2 norm of the difference between the generated CG **c** = **c**(**x**) and the prescribed CG ($\mathbf{c}_{\text{s}}$) (H. Yang et al. 2020). Here **x** represents the design variables, and **c**(**x**) is the function to predict CG given any values of design variable **x**. Essentially Eq. (7.2) minimizes the difference between two CG vectors, corresponding to the best agreement of two profiles.

Two case studies with different complexities are conducted to verify the proposed methodology. In the first case study, design variables include the normalized chemical

species concentrations at the inlets (c_1, c_2, …, c_6) and a scalar u that defines the pressure magnitude, i.e. $\mathbf{x} = [u, c_1, c_2, \ldots, c_6]$, forming an optimization problem of seven dimensions. In this study, the ratio of the pressure differences across the three mixing channels (between the Y junction and the Ψ-shaped junction) is kept constant, viz., Δp_1: Δp_2: Δp_3 = 75:93:98. In the second case study, a more challenging problem involving 9 dimensions is employed to further examine the feasibility of the proposed method. The design variables include both the normalized chemical concentrations at the six inlets and the pressure differences across the three mixing channels i.e., $\mathbf{x} = [\Delta p_1, \Delta p_2, \Delta p_3, c_1, c_2, \ldots, c_6]$.

7.2 Proposed Methodology

In this section, the MFROM-based optimization methodology is presented, which includes two separate stages: offline MFROM construction and online MFROM optimization. The former constructs an MFROM based on both PBCM and CFD simulation data of CGs, and incorporates a sample infill step to iteratively update the model for enhanced accuracy. Once built, the MFROM can be used in the online optimization stage to replace CFD for predicting CG profiles. Since the MFROM can predict CGs with similar accuracy as the CFD but at a much faster evaluation speed, the optimization can be performed very efficiently. The offline MFROM construction and online MFROM optimization will be described in detail below.

Figure 7.1 illustrates the offline MFROM construction stage, and its detailed procedure is given as follows: first, the initial samples are generated in the multi-dimensional design space using LHS. Second, both CFD and PBCM, representing high-fidelity and low-fidelity μCGG simulation, are conducted to generate CGs data at these

sampled locations in the previous step. Next, the POD (detailed in section 2.2.3) is performed on CGs data (also called snapshots) of PBCM and CFD at the detector to generate the POD modal coefficients $\boldsymbol{\alpha}_i^{\mathrm{PBCM}}$, $\boldsymbol{\alpha}_i^{\mathrm{CFD}}$ and basis matrix $\mathbf{U}^{\mathrm{PBCM}}$ and $\mathbf{U}^{\mathrm{CFD}}$ of underlying subspace, achieving low-rank representation of the snapshot data. Index i denotes the i^{th} data instance. Furthermore, a Kriging-based surrogate model is constructed to establish the relationship between $\boldsymbol{\alpha}_i^{\mathrm{PBCM}}$ and $\boldsymbol{\alpha}_i^{\mathrm{CFD}}$, which are, respectively, treated as the inputs and outputs. It is essentially a multi-input multi-output surrogate model, and the dimension of input and output is therefore determined by that of $\boldsymbol{\alpha}_i^{\mathrm{PBCM}}$ and $\boldsymbol{\alpha}_i^{\mathrm{CFD}}$. Since the data obtained through initial sampling is insufficient to capture the response surface that maps between these two POD coefficients within the whole parameter space, an exploration-based adaptive sampling technique is developed to add samples and corresponding CGs simulation data to further improve the accuracy of MFROM by updating the Kriging model. Specifically, a distance-based infill is defined as

$$\mathbf{x}_j^* = \max(\min \left\| \mathbf{x}_j^* - \mathbf{x}_j^i \right\|) \tag{7.3}$$

where x_j^* represents the j^{th} element of the infill location ($\mathbf{x}^* = [x_1^*, x_2^* \ldots, x_j^* \ldots]^{\mathrm{T}}$); x_j^i is the j^{th} element of the i^{th} existing sample point $\mathbf{x}^i$. This infill criterion searches for a point in the parameter space that maximizes the smallest distance to all existing samples along each dimension, leading to a more uniform distribution of all the samples. The whole process is repeated until the maximum number of global optimization iterations is reached. Finally, the POD basis matrix of the PBCM and CFD data, i.e., $\mathbf{U}^{\mathrm{PBCM}}$, $\mathbf{U}^{\mathrm{CFD}}$ and the Kriging surrogate model that form the MFROM will be utilized for online optimization. Note that the goal of the offline stage is to build an MFROM that has similar accuracy as CFD within

subspace $\mathbf{U}^{CFD}$, and hence, can be used to replace CFD simulation in the online optimization stage.

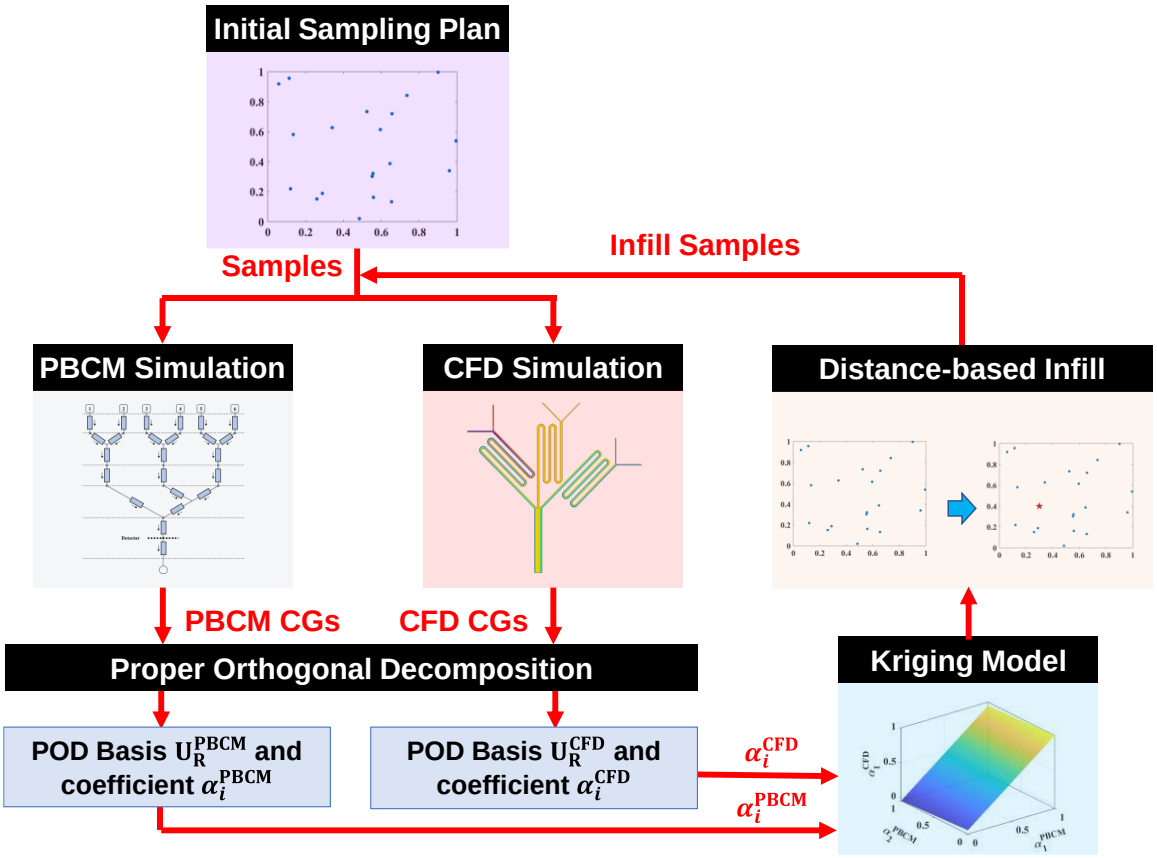


Figure 7.1 Flowchart of offline MFROM construction for µCGG design.

Figure 7.2 shows the flowchart of online optimization for µCGG design using MFROM constructed above. The user-prescribed CG ($\mathbf{c}_s$) is projected onto the POD basis matrix of CFD data, i.e., $\mathbf{U}^{CFD}$ obtained in the offline MFROM construction process, resulting in the modal coefficients ($\boldsymbol{\alpha}_s$) of the prescribed CG. Next, $\boldsymbol{\alpha}_s$ are supplied to the optimization loop to search for optimal design parameters that make the POD coefficients predicted by MFROM $\boldsymbol{\alpha}^{MFROM}$ match $\boldsymbol{\alpha}_s$. Therefore, the cost function of the optimization process in POD subspace is defined as

$$J = \left\| \boldsymbol{\alpha}^{\mathrm{MFROM}} - \boldsymbol{\alpha}_{\mathrm{s}} \right\|_2 \tag{7.4}$$

During optimization, the candidate design under evaluation (**x**) is supplied to PBCM simulation first and the PBCM-predicted CG $\mathbf{c}^{\mathrm{PBCM}}$ is projected onto POD basis $\mathbf{U}^{\mathrm{PBCM}}$ to obtain POD modal coefficients ($\boldsymbol{\alpha}^{\mathrm{PBCM}}$) of the design. $\boldsymbol{\alpha}^{\mathrm{PBCM}}$ is then provided to the Kriging model for mapping POD modal coefficients of different fidelities, yielding $\boldsymbol{\alpha}^{\mathrm{MFROM}}$. Next, the fitness value of the cost function is evaluated by comparing $\boldsymbol{\alpha}^{\mathrm{MFROM}}$ and $\boldsymbol{\alpha}_{\mathrm{s}}$ as shown in Eq. (7.4). At the end of optimization, the values of design parameters that produce the closest agreement of $\boldsymbol{\alpha}^{\mathrm{MFROM}}$ with the prescribed one $\boldsymbol{\alpha}_{\mathrm{s}}$ are obtained. Eventually, the actual CG can be reconstructed by multiplying $\mathbf{U}^{\mathrm{CFD}}$ and optimal design parameters $\boldsymbol{\alpha}^{\mathrm{MFROM}}$, and compared to the prescribed $\mathbf{c}_{\mathrm{s}}$. The key advantage of this approach is that no CFD will be used during the online design optimization, and the cost function evaluation only involves PBCM simulation, subspace projection, and surrogate model evaluation. Both PBCM and surrogate models are made up of elementary functions, and can be performed rapidly (less than 0.01 second in total), leading to salient computation efficiency. Eq. (7.4) is minimized with GA, a global optimization method to search for the design parameters that yield the minimal difference between $\boldsymbol{\alpha}^{\mathrm{MFROM}}$ and $\boldsymbol{\alpha}_{\mathrm{s}}$.

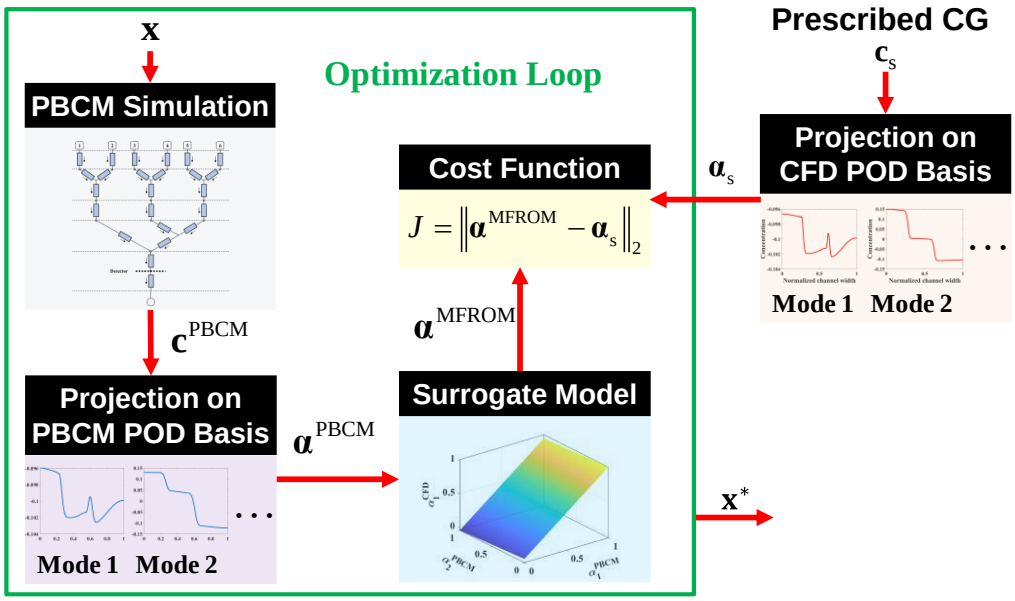


Figure 7.2 Flowchart of offline optimization of the MFROM for μCGG design.

7.2.1 Computation Acceleration by Basis Transformation

It should be noted that the explicit approach above to obtain $\boldsymbol{\alpha}^{\mathrm{MFROM}}$ is less efficient as the CG $\mathbf{c}^{\mathrm{PBCM}}$ needs to be reconstructed using the Fourier expansion, as shown in Eq. (7.1), and then projected onto POD basis $\mathbf{U}_{\mathrm{R}}^{\mathrm{PBCM}}$ to yield $\boldsymbol{\alpha}^{\mathrm{PBCM}}$. $\boldsymbol{\alpha}^{\mathrm{PBCM}}$ is then supplied to the Kriging model for multi-fidelity mapping. This process can be simplified by combining the Fourier expansion and the projection into a single, more computationally efficient matrix operation. Specifically, the CG predicted by PBCM, i.e., $\mathbf{c}^{\mathrm{PBCM}}$ can be represented as the product of Fourier basis matrix and the coefficient according to Eq. (7.1), viz., $\mathbf{c}^{\mathrm{PBCM}} = \mathbf{F}_{\mathrm{fs}}\mathbf{d}$. Next, the projection of the $\mathbf{c}^{\mathrm{PBCM}}$ on the POD basis vectors is given by

$$\boldsymbol{\alpha}^{\mathrm{PBCM}} = \left(\mathbf{U}_{\mathrm{R}}^{\mathrm{PBCM}}\right)^{\mathrm{T}} \mathbf{c}^{\mathrm{PBCM}} \tag{7.5}$$

where $\mathbf{U}_{\mathrm{R}}^{\mathrm{PBCM}}$ is the truncated POD basis matrix. Thus, computation of $\boldsymbol{\alpha}^{\mathrm{PBCM}}$ can be rewritten as

$$\boldsymbol{\alpha}^{\mathrm{PBCM}} = \mathbf{Z}\mathbf{d}, \text{ and } \mathbf{Z} = \left(\mathbf{U}_{\mathrm{R}}^{\mathrm{PBCM}}\right)^{\mathrm{T}} \mathbf{F}_{\mathrm{fs}} \tag{7.6}$$

where $\mathbf{Z}$ is a constant matrix, which can be precomputed outside the optimization loop once $\mathbf{U}_{\mathrm{R}}^{\mathrm{PBCM}}$ and the number of Fourier terms and detector resolution are determined. Eq. (7.6) essentially indicates that directly mapping from the Fourier series subspace to the POD modal subspace of PBCM can be obtained through a single matrix multiplication. This is not only computationally efficient, but also reduces the usage of physical memories. Specifically, the memory requirement of $\mathbf{Z}$ is $k \times (n+1)$, which is less than that of $\mathbf{U}_{\mathrm{R}}^{\mathrm{PBCM}}$ ($mp \times k$) and $\mathbf{F}_{\mathrm{fs}}$ ($mp \times n+1$), since k is greatly less than mp, which makes the algorithms even more amenable to GPU computing.

7.2.2 Online GPU-based Global Design Optimization

The PBCM model, the mapping matrix $\mathbf{Z}$ in Eq. (7.6) and the Kriging model in Eq. (2.13) then can be combined for online optimization as illustrated in Figure 7.2. Their compact sizes and rapid evaluation speed enable the use of heuristic global optimization methods (such as genetic algorithm, particle swarm optimization, and differential evolution) on the GPU computing platform. The massive parallel computing threads available in GPU allow for concurrent evaluation of an enormous number of candidate designs, significantly improving design exploration, convergence speed, and accuracy towards the global optimum. For this reason, in the present work the entire optimization loop built on the GA, one of the most widely used and stable global optimization algorithms, is developed in

GPU, and its flowchart for online μCGG design is depicted in Figure 7.3. At the beginning, initial populations are randomly generated in the CPU. All necessary models and information must be transferred from CPU to GPU prior to GA runs, including PBCM, Kriging model, modal coefficients of desired CG, initial population, and lower and upper bounds. In addition, memory must be allocated to store fitness values, and gene sequences of parents and offsprings. GPU then handles the entire process of fitness evaluation and GA operations until the solution converges. Fitness evaluation is broken into three stages: (1) PBCM evaluation; (2) mapping from PBCM Fourier coefficients to PBCM modal coefficients $\boldsymbol{\alpha}^{\mathrm{PBCM}}$; (3) Kriging model evaluation; and (4) cost computation, and GA operations include four steps: (1) sorting and selection; (2) crossover; (3) mutation; and (4) migration. In the first step of GA operation, gene sequences of populations are sorted according to their fitness values, and then parents are selected. In this work, the truncation selection method is implemented, which randomly selects the top 15% of the population as the parents. Once the parent genes are selected, crossover is implemented to generate the offspring. A single-point crossover is adopted, which particularly selects the pressure-related genes from one parent and the concentration-related genes from another. In the third step of GA, mutation operation is performed on the offspring with a mutation probability of 0.25. Once the gene sequence is selected for mutation, three values from the gene sequence are selected randomly and mutated by the Gaussian distribution with a deviation equal to 5% of the range. It is essential to mention that 2% of elite sequences are passed to the next generation without crossover and mutation to preserve the best designs. At the last step, migration operation is conducted, which exchanges the genes between populations, preventing pre-maturations. For every 10 generations, 2% of the best genes are passed on

to the neighboring population in the same direction, forming a circular pattern. In this work, 32 different populations, each with 64 individuals (i.e., gene sequences) are utilized for optimization. In total, 2,048 candidate designs are evaluated in parallel, which is difficult for CPU to handle but feasible on GPU. The entire optimization process is repeated for a predefined number of iterations (or generations). After the last generation, the optimum solution is transferred back to CPU. A GPU workstation is utilized for this research, which is comprised of Intel(R) Core(TM) i9-9820X CPU @ 3.30 GHz and NVIDIA GeForce RTX 2080 Ti GPU. Details regarding GA implementation can be found elsewhere (Katoch, Chauhan, and Kumar 2021; S. H. Hong et al. 2019).

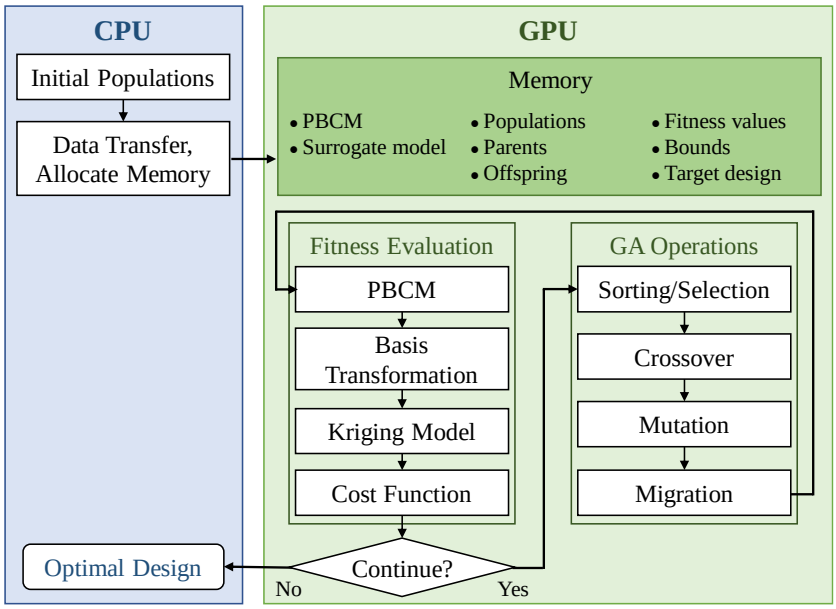


Figure 7.3 GPU-enabled optimization flowchart.

7.3 Results and Discussion

Two case studies representing different levels of complexities, respectively, including seven and nine design parameters, are presented to examine the proposed

methodology. For both cases, the MFROM is first constructed following the offline construction process in CPU above due to its need for CFD simulation and Kriging model construction. Then, given the user-prescribed CG profiles, the MFROM is used in online global optimization to search optimal design parameters that yield CGs closest to the prescribed ones. Note that the online optimization is performed within GPU to make use of its massive computing threads.

Correspondingly, two comparisons will be made in this chapter, respectively, in the offline model construction and the online optimization stages to scrutinize the model accuracy and the optimization accuracy. In the former, a set of parameters $\tilde{\mathbf{x}}$ will be provided by a random sampling technique, and then supplied to MFROM and CFD simulation to compute their CGs, i.e., $\mathbf{c}^{\mathrm{MFROM}}\left(\tilde{\mathbf{x}}\right)$ and $\mathbf{c}^{\mathrm{CFD}}\left(\tilde{\mathbf{x}}\right)$. A small difference between them suggests excellent accuracy of MFROM. To evaluate optimization accuracy, prescribed profiles $\mathbf{c}_s$ will be first provided, and the optimal design parameters $\mathbf{x}^*$ will be found by the proposed MFROM-based optimization method. $\mathbf{x}^*$ is then supplied to MFROM and CFD to generate corresponding CGs, i.e., $\mathbf{c}^{\mathrm{MFROM}}\left(\mathbf{x}^*\right)$ and $\mathbf{c}^{\mathrm{CFD}}\left(\mathbf{x}^*\right)$, both of which will be compared with the prescribed one $\mathbf{c}_s$. In this comparison, agreement of $\mathbf{c}^{\mathrm{MFROM}}\left(\mathbf{x}^*\right)$ and $\mathbf{c}^{\mathrm{CFD}}\left(\mathbf{x}^*\right)$ with $\mathbf{c}_s$ reflects salient accuracy of optimization. However, when $\mathbf{c}^{\mathrm{MFROM}}\left(\mathbf{x}^*\right)$ and $\mathbf{c}^{\mathrm{CFD}}\left(\mathbf{x}^*\right)$ match well but are very different from $\mathbf{c}_s$, it indicates that the optimization has poor performance or the μCGGs have physical limitations that render obtaining the desired CG profile difficult.

The NRMSE is adopted in this chapter to measure the difference between two CGs, which is defined as:

$$\varepsilon = \frac{\sqrt{\sum_{t=1}^{m}\left(\mathbf{c}-\mathbf{c}^{\mathrm{B}}\right)^{2}}}{\mathbf{c}_{\max}^{\mathrm{B}}-\mathbf{c}_{\min}^{\mathrm{B}}} \tag{7.7}$$

where m is the resolution (m+1 grid points) at the detector to observe CGs, and also the length of the CG vector $\mathbf{c}$. $\mathbf{c}$ and $\mathbf{c}^{\mathrm{B}}$ are, respectively, the predicted CG under comparison and the true/benchmark values. When evaluating MFROM accuracy in the offline stage, $\mathbf{c}$ and $\mathbf{c}^{\mathrm{B}}$ are, respectively, $\mathbf{c}^{\mathrm{MFROM}}\left(\tilde{\mathbf{x}}\right)$ and $\mathbf{c}^{\mathrm{CFD}}\left(\tilde{\mathbf{x}}\right)$. For optimization accuracy examination, $\mathbf{c}$ is either $\mathbf{c}^{\mathrm{MFROM}}\left(\mathbf{x}^{*}\right)$ or $\mathbf{c}^{\mathrm{CFD}}\left(\mathbf{x}^{*}\right)$ and $\mathbf{c}^{\mathrm{B}}$ is $\mathbf{c}_{\mathrm{s}}$. A smaller ε corresponds to a better agreement between the two CGs.

7.3.1 Design of Inlet Concentration and Pressure Amplitude

In this section, we will present the results of the first case study that involves the design of six inlet concentrations and one pressure magnitude, leading to an optimization problem of seven dimensions (see details in Section 7.1).

7.3.1.1 Offline MFROM Construction

First, LHS is used to generate 20 initial samples in the seven-dimensional design space. Besides, 20 testing samples are also generated with LHS, where CFD data is collected and used to evaluate the accuracy of MFROM. Figure 7.4 shows the average NRMSE (ε) of the 20 testing samples. Results show that as more infill samples are added, the average ε gradually decreases, which means that MFROM becomes more and more

accurate. After 480 infill iterations, the average ε drops below 0.55%, and more importantly, starts to saturate. Then our infill iteration is terminated.

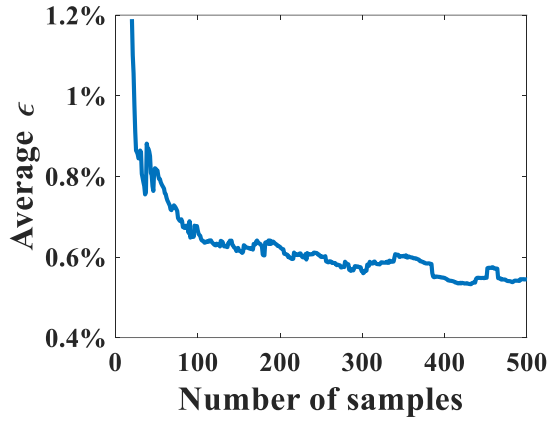


Figure 7.4 Average NRMSE of 20 testing inputs.

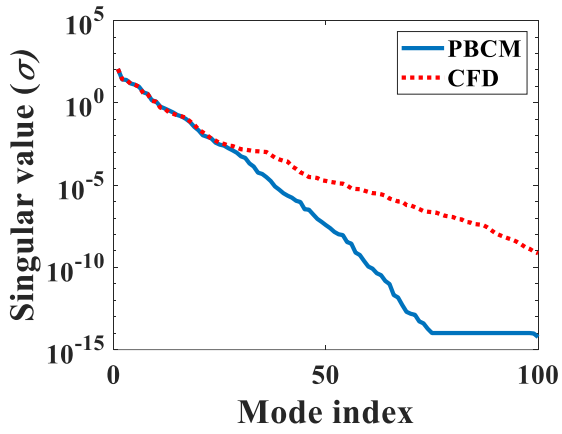


Figure 7.5 Singular values comparison between PBCM and CFD snapshots.

The singular values σ and POD basis vectors $\mathbf{U}_R$ for both PBCM and CFD snapshots after the infill process are compared in Figure 7.5 and Figure 7.6. Recall that

PBCM and CFD, respectively, represent the low-fidelity and high-fidelity data set in the present work. Figure 7.5 shows that singular values of both data sets decay rapidly, confirming that the high-order basis vectors (i.e., modes of the high index) make less contribution to data variance and representation. Thus, only the first ten modes are kept for both PBCM and CFD data sets to achieve a 0.9999 energy ratio, and the rest of them are truncated. Besides, the singular values of the first several leading modes of both data sets are also close, implying that these modes may also look similar. The comparison for the first four modes is presented in Figure 7.6. The first three modes of PBCM and CFD data are alike with small differences, which means that the most important features of CFD CGs are also present in the low-fidelity PBCM data, and more importantly, in the same order of importance. The fourth mode of PBCM and CFD, although very dissimilar, are actually mirroring of each other, which is within our expectation as the basis vectors computed numerically by singular value decomposition can take opposite spatial directions. Consequently, the observation of these POD modal characteristics also proves that PBCM is a great low-fidelity alternative to CFD for μCGG modeling and simulation (Y. Wang, Mukherjee, and Lin 2006).

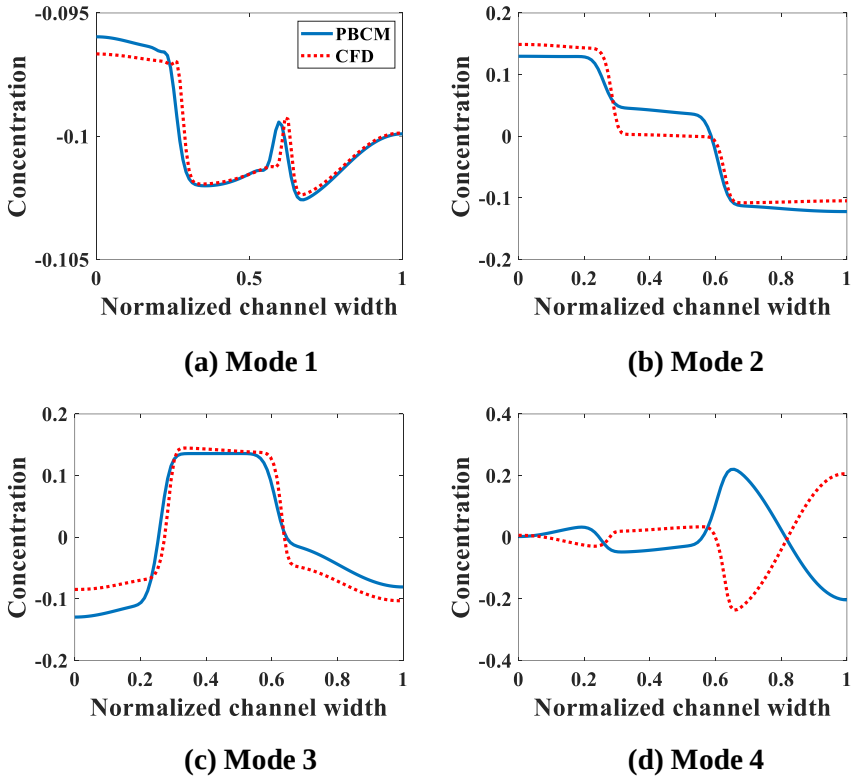


Figure 7.6 POD basis vectors/modes comparison between PBCM and CFD snapshots.

To visualize the accuracy of MFROM after the infill process, the CGs predicted by MFROM are compared with PBCM and CFD for 4 testing samples as shown in Figure 7.7. In each subfigure, the contour plots of CFD results are at the top row, and the bottom compares PBCM and MFROM with CFD on CGs. The corresponding NRMSEs of PBCM and MFROM are listed in Table 7.1. It is clearly shown that MFROM-predicted CGs match CFD-predicted CGs very well and their differences are almost negligible (ε <1% for all testing samples), and thus, MFROM is more accurate than PBCM. Besides, MFROM simulation takes only ~ 0.007s and is much faster than CFD simulation (1325s), both measured on the same CPU platform for a fair comparison.

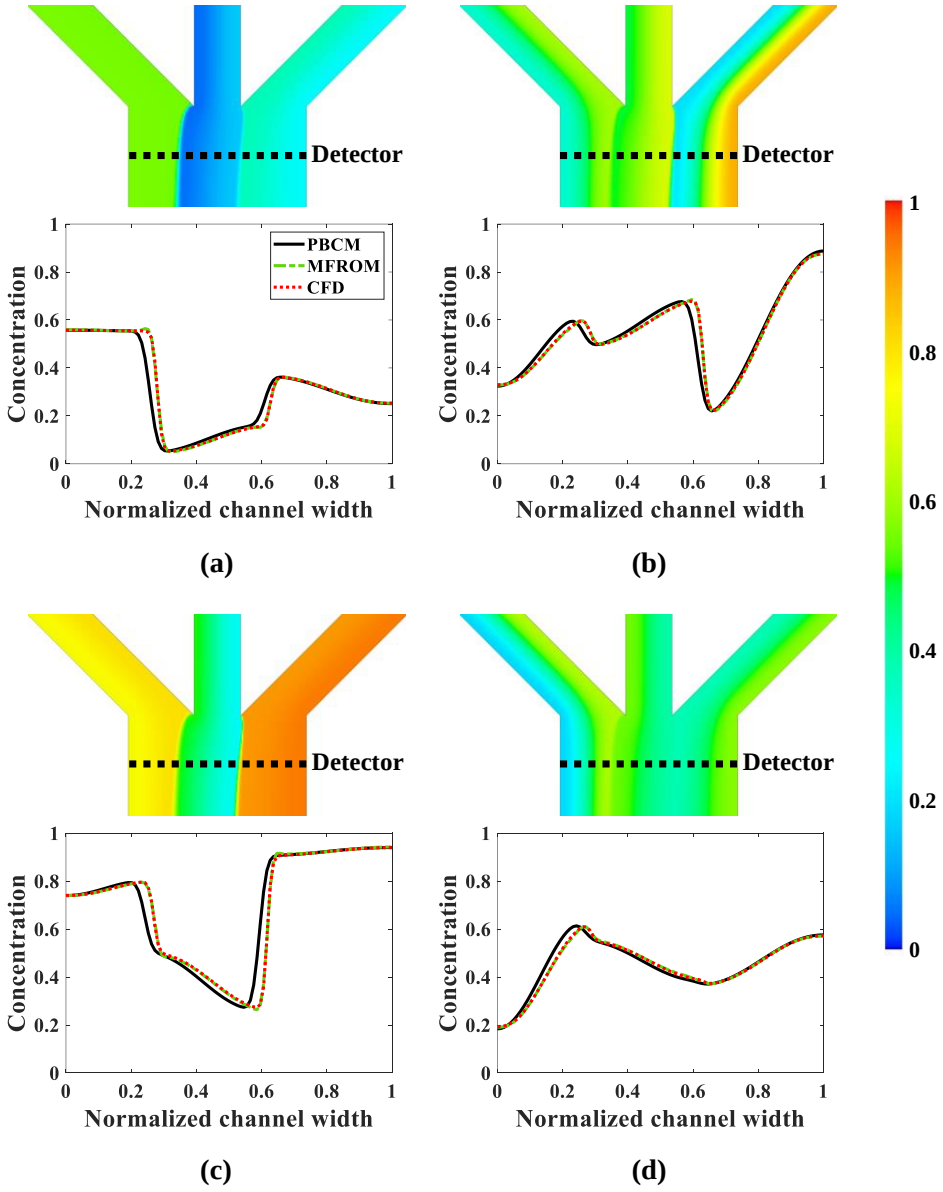


Figure 7.7 CGs comparison for four selected testing samples.

Table 7.1. NRMSE of the 4 selected testing samples.

Testing Sample	(a)	(b)	(c)	(d)
ε_{PBCM}	10.39%	6.01%	11.06%	4.52%
ε_{MFROM}	0.64%	0.45%	0.66%	0.59%

7.3.1.2 Online Optimization

Once the MFROM is built and verified, it can be used to replace CFD simulation in design optimization using GA on GPU given any prescribed CG profiles. The number of generations of the GA is set to be 40, which is found through trials and guarantees convergence of the solution. Four prescribed CGs (blue lines in Figure 7.8) are considered, each of which is obtained by connecting three line segments in different directions. For each design, ten independent runs are performed following the online optimization algorithm above. The mean (μ) and standard deviation ($\tilde{\sigma}$) of each design parameter and optimization time in these 10 runs are listed in Table 7.2, and the small standard deviations indicate consistent optimization results for all 10 runs. More importantly, all optimal design is attained rapidly and only with ~6.4 s. The design parameters extracted from the run with median performance are supplied to both MFROM and CFD simulation to evaluate the prediction of CGs. In Figure 7.8, the label 'Prescribed', 'MFROM', and 'CFD', respectively, denote the prescribed CG and CGs predicted by MFROM and CFD at the optimal design. In each subfigure (corresponding to different prescribed CGs), the top row shows the CFD contour plots of the chemical species concentration, and the comparison between the prescribed, MFROM predicted and CFD-predicted CGs is illustrated at the bottom. The corresponding NRMSE errors, which measure deviations of MFROM- or CFD-predicted CGs (viz., $\mathbf{c}$) from the prescribed CGs (viz., $\mathbf{c}^B$) using Eq. (7.7), are listed in Table 7.3. There are several interesting observations. First, the CFD-predicted CGs (red lines in the bottom plots in Figure 7.8) match the MFROM-predicted CGs excellently, which also confirms the accuracy of MFROM and validates the modeling method. Second, the MFROM-predicted CGs at the optimal design parameters can generate a CG

resembling the prescribed CGs, which confirms the feasibility of MFROM-based design optimization. However, differences between the MFROM- and CFD-predicted and prescribed CGs are also noticeable, which is attributed to the impermeability of channel walls. No species flux at the channel walls leads to zero concentration gradient at channel walls, hence making it impossible to match prescribed CGs with non-zero gradient therein. Such a disagreement is associated with the physical limitation of the μCGG device rather than the deficiency of the proposed design method. More detailed discussion can be found in (H. Yang et al. 2020).

Table 7.2 Mean and standard deviation of design parameters and optimization time.

Prescribed		(a) Sawtooth-shaped	(b) Trapezoidal	(c) Symmetric trench-shaped	(d) Asymmetric trench-shaped
u	μ_1	0.829	0.878	0.866	0.781
	$\tilde{\sigma}_1$	0.003	0.025	0.003	0.009
c_1	μ_2	0.924	0.005	0.238	0.941
	$\tilde{\sigma}_2$	0.005	0.012	0.004	0.007
c_2	μ_3	0.039	0.867	0.683	0.648
	$\tilde{\sigma}_3$	0.006	0.027	0.004	0.007
c_3	μ_4	0.736	0.797	0.298	0.423
	$\tilde{\sigma}_4$	0.003	0.003	0.005	0.004
c_4	μ_5	0.261	0.803	0.310	0.052
	$\tilde{\sigma}_5$	0.004	0.003	0.004	0.004
c_5	μ_6	0.624	0.852	0.622	0.163
	$\tilde{\sigma}_6$	0.003	0.005	0.003	0.006
c_6	μ_7	0.383	0.090	0.286	0.950
	$\tilde{\sigma}_7$	0.005	0.014	0.004	0.006
t (s)	μ_t	6.271	6.423	6.407	6.421
	$\tilde{\sigma}_t$	0.030	0.035	0.022	0.028

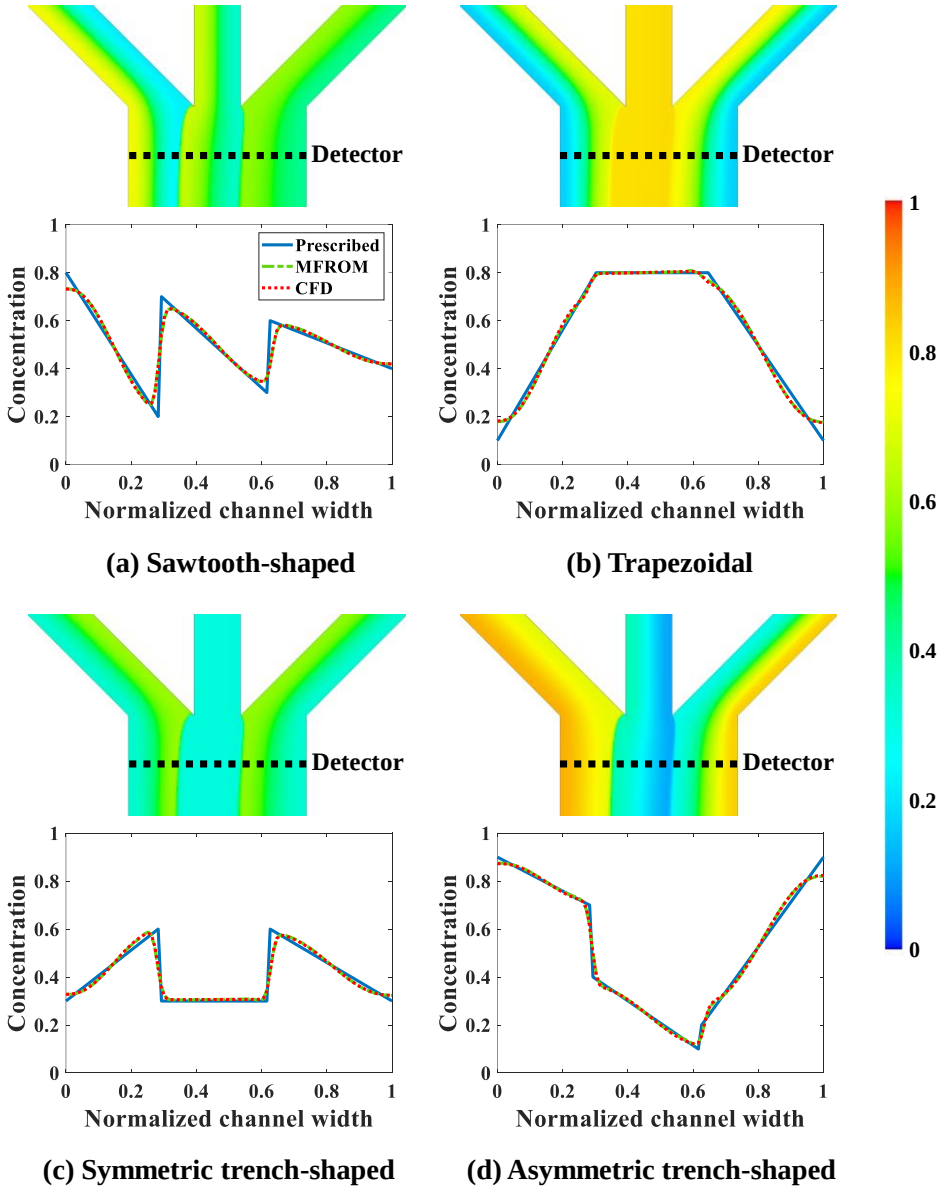


Figure 7.8 MFROM predicted CGs relative to the true prescribed CGs.

Table 7.3 NRMSE of predicted CGs.

Prescribed	(a) Sawtooth-shaped	(b) Trapezoidal	(c) Symmetric trench-shaped	(d) Asymmetric trench-shaped
ε_{MFROM}	6.64%	3.08%	10.13%	2.76%
ε_{CFD}	6.46%	3.13%	9.94%	2.70%

7.3.2 Design of Inlet Concentration and Pressure Differences

In this section, we will present the results of the second case study, whose optimization includes six inlet concentrations and three differential pressure values, yielding a nine-dimensional design problem.

7.3.2.1 Offline MFROM Construction

For offline MFROM construction, 60 initial samples are generated to construct the initial MFROM, and 1,440 infill samples of CFD simulations are added iteratively to enhance MFROM accuracy. Similarly, data of 20 testing samples are used to evaluate NRMSE of MFROM during the offline stage following Eq. (7.7). Figure 7.9 presents the NRMSE as infill samples are continuously added. It shows that the average ε eventually reaches a lower value below NRMSE = 3% and MFROM accuracy gradually improves given 1,500 CFD simulations in total.

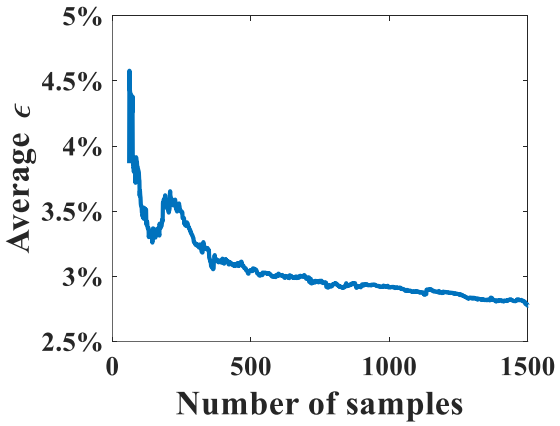


Figure 7.9 Average NRMSE of 20 testing inputs.

The singular values and the POD basis vectors extracted from the PBCM and CFD snapshots are illustrated in Figure 7.10 and Figure 7.11, respectively. Similar to the previous case study, there is a fast decrease in the singular value σ for both data sources as the mode number increases. Eventually, 21 and 25 POD basis vectors/modes (**u**) are kept for PBCM and CFD data, respectively. In Figure 7.11, we can clearly observe that the basis vectors of PBCM and CFD match well. The notice differences in mode 1 arise from the different fidelities of the simulations and their distinct model assumptions. However, the resemblances of modal characteristics between PBCM and CFD snapshots confirm both models capture key physics of flow and species transport and can be used to serve as data sources of different fidelities for MFROM construction.

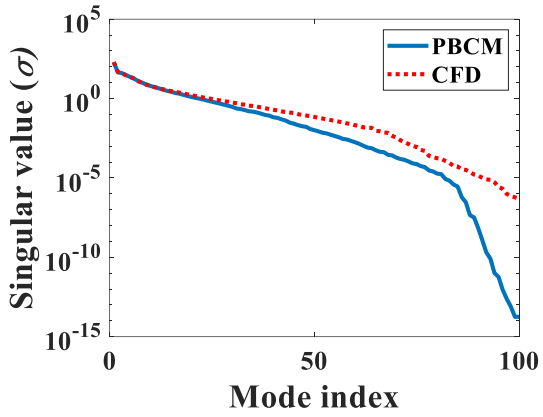


Figure 7.10 Singular values comparison between PBCM and CFD snapshots.

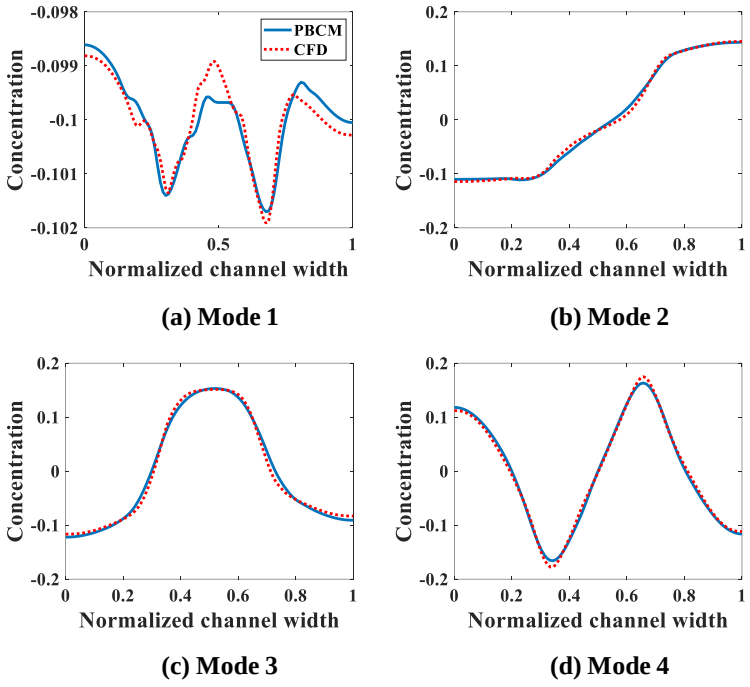


Figure 7.11 POD basis vectors/modes comparison between PBCM and CFD snapshots.

Similarly, CGs comparison among PBCM, MFROM, and CFD for four testing samples is illustrated in Figure 7.12, where CFD contour plots at the detector location are also included at the top of each subfigure. Corresponding NRMSEs of PBCM and MFROM of the four testing samples relative to CFD results are listed in Table 7.4. MFROM-predicted CGs exhibit better agreements with CFD than PBCM, indicated by lower NRMSEs, signifying that MFROM is more accurate for approximating CFD models. However, the average NRMSE of MFROM is higher than the previous case study. That is because the model complexity in this study grows dramatically as the dimension of design variables increases to 9, and thus a significant amount of data is entailed to further improve model accuracy globally within the entire 9-dimensional parameter space. In this study, the total number of simulation runs is set to be 1,500 to obtain a reasonable balance between

simulation time and model accuracy. Similar to the previous case study, MFROM only takes 0.007s to predict a CG with excellent accuracy, while CFD simulation costs 1,325s.

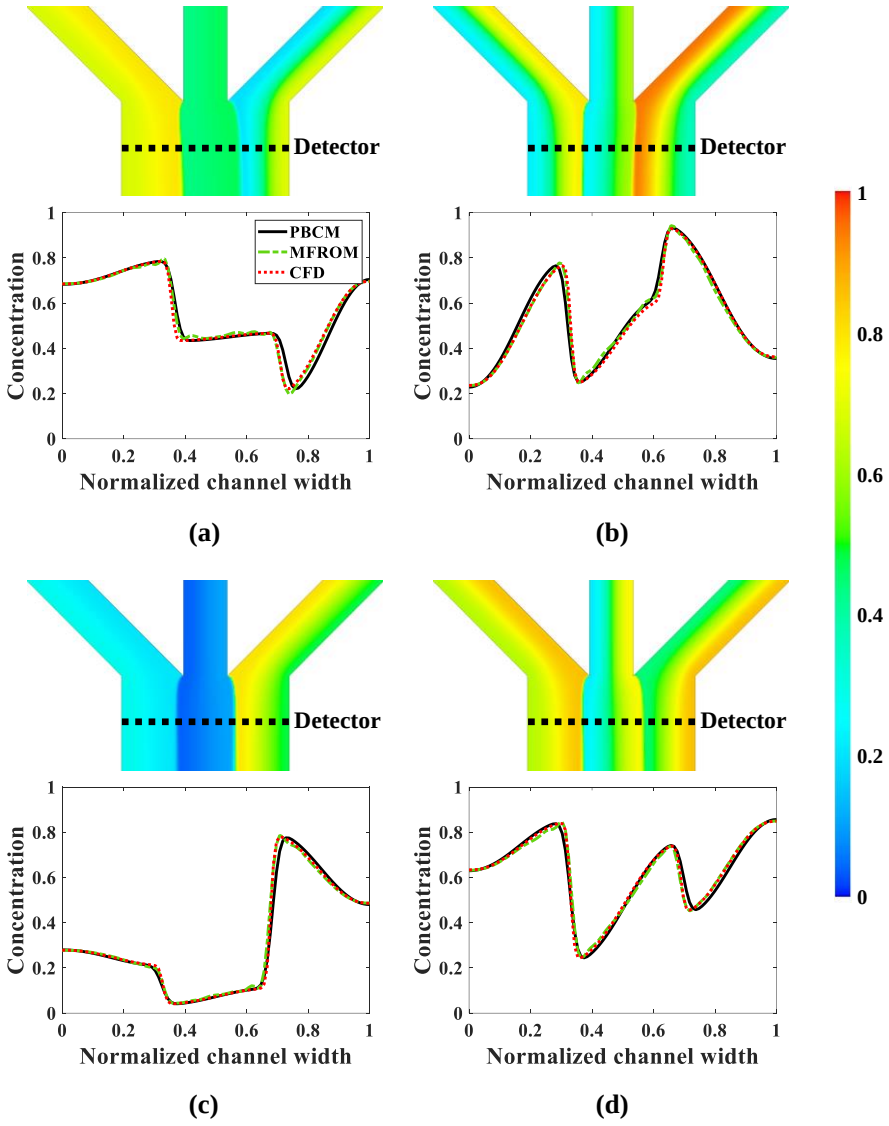


Figure 7.12 CGs comparison for four selected testing samples.

Table 7.4 NRMSE of the four selected testing samples.

Testing Sample	(a)	(b)	(c)	(d)
ε_{PBCM}	5.71%	4.89%	3.38	3.83%
ε_{MFROM}	2.68%	2.90%	1.97	2.35%

7.3.2.2 Online Optimization

Next, the verified MFROM is utilized for design optimization of μCGGs using GPU-enabled GA. Due to the increase in the dimension of the problem, 70 generations of GA are adopted for searching optimal design in this case study. Four prescribed profiles same as the above denoted by the blue lines in Figure 7.13 are used in the design optimization to inspect the proposed method. Likewise, 10 runs are repeated, and the mean (μ) and standard deviation ($\tilde{\sigma}$) of each optimal design parameter and optimization time are listed in Table 7.5, and we can see that all design variables have a relatively small deviation (2.6% relative to the mean value on average), which indicates that GA optimization on the GPU platform yields consistent results. The design parameters extracted from the run with median optimization performance are supplied to both MFROM and CFD simulations to obtain their CGs, which are represented by the green and red lines in Figure 7.13. Meanwhile, the contour plots of CFD are also included at the top to visualize the flow and species concentration distribution at the junction region. The NRMSEs of predicted CGs relative to the prescribed CGs are summarized in Table 7.6. Observations are similar to the previous case study that CGs predicted by MFROM and CFD exhibit excellent agreement both visually and quantitatively even in the 9-dimensional trade space and the proposed MFROM-based global optimization is feasible for the μCGG design. Both lead us to believe that MFROM can be potentially used in lieu of CFD simulation for microfluidic device optimization, particularly in the circumstances where design speed is more critical.

Table 7.5 Mean and standard deviation of design parameters and optimization time.

Prescribed		(a) Sawtooth-shaped	(b) Trapezoidal	(c) Symmetric trench-shaped	(d) Asymmetric trench-shaped
Δp_1	μ_1	0.846	0.757	0.851	0.727
	$\tilde{\sigma}_1$	0.021	0.025	0.021	0.017
Δp_2	μ_2	0.923	0.943	0.925	0.863
	$\tilde{\sigma}_2$	0.014	0.026	0.014	0.013
Δp_3	μ_3	0.954	0.841	0.958	0.876
	$\tilde{\sigma}_3$	0.010	0.031	0.010	0.012
c_1	μ_4	0.811	0.066	0.295	0.908
	$\tilde{\sigma}_4$	0.010	0.012	0.004	0.005
c_2	μ_5	0.157	0.747	0.616	0.708
	$\tilde{\sigma}_5$	0.010	0.036	0.005	0.004
c_3	μ_6	0.732	0.800	0.302	0.419
	$\tilde{\sigma}_6$	0.006	0.004	0.004	0.004
c_4	μ_7	0.274	0.803	0.285	0.075
	$\tilde{\sigma}_7$	0.008	0.004	0.004	0.004
c_5	μ_8	0.601	0.731	0.598	0.218
	$\tilde{\sigma}_8$	0.003	0.019	0.005	0.006
c_6	μ_9	0.414	0.094	0.319	0.890
	$\tilde{\sigma}_9$	0.002	0.012	0.004	0.006
t (s)	μ_t	11.266	11.529	11.585	11.608
	$\tilde{\sigma}_t$	0.176	0.038	0.032	0.022

Table 7.6 NRMSE of predicted CGs.

Prescribed	(a) Sawtooth-shaped	(b) Trapezoidal	(c) Symmetric trench-shaped	(d) Asymmetric trench-shaped
$\varepsilon_{\mathrm{MFROM}}$	6.09%	3.02%	8.80%	2.98%
$\varepsilon_{\mathrm{CFD}}$	6.26%	3.53%	8.50%	2.92%

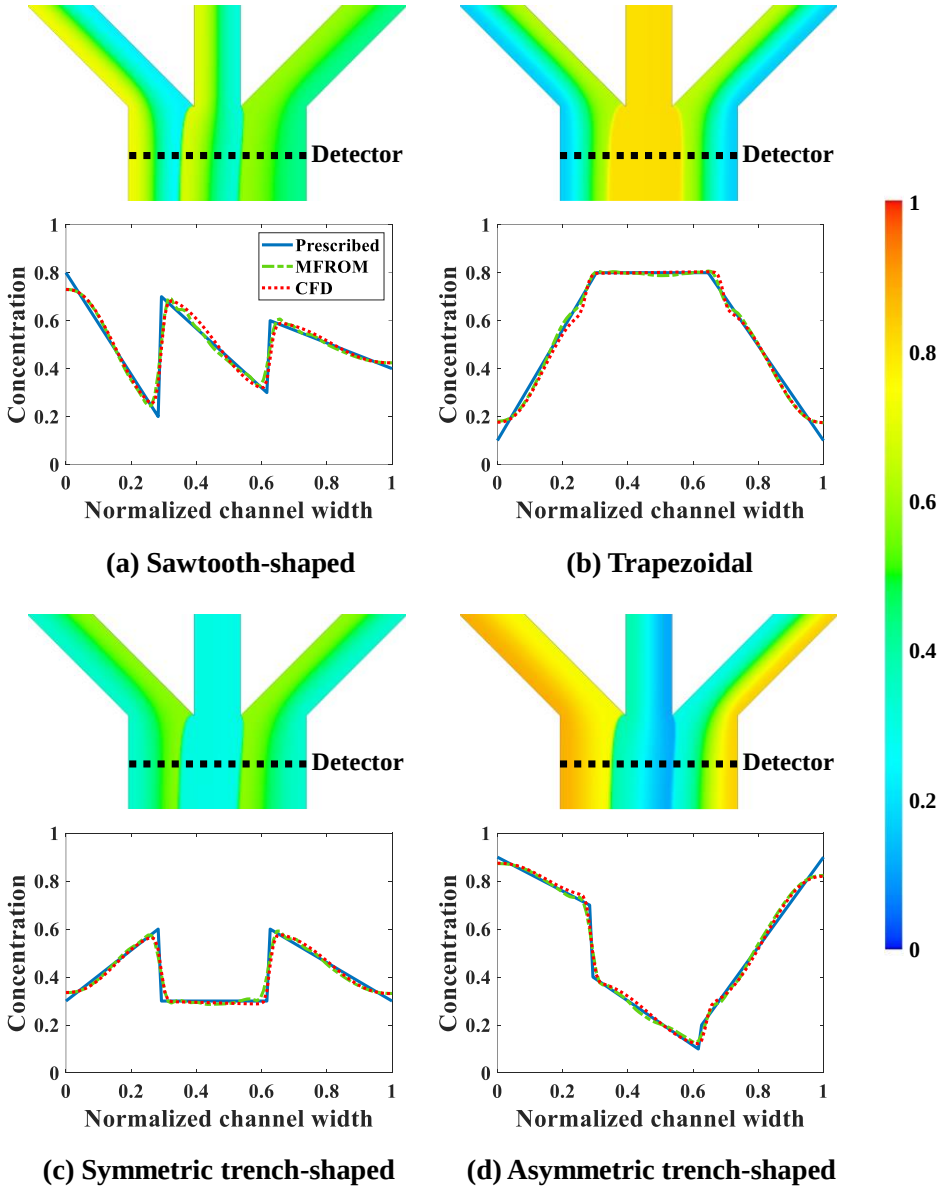


Figure 7.13 MFROM predicted CGs relative to the true prescribed CGs.

7.4 Summary

In this chapter, an MFROM method that combines both the low-fidelity PBCM and high-fidelity CFD simulation is proposed for accurate and efficient simulation and global design optimization of μCGGs. The entire process is divided into the offline MFROM

construction and the online global optimization stages. In the latter, because of its compact size and multi-fidelity nature, MFROM can be used in lieu of the high-fidelity CFD simulation to boost up the computing speed significantly while maintaining optimization accuracy, enabling an efficient and portable design platform. To construct MFROM, POD is first applied to both PBCM and CFD snapshot data, producing the low rank representation of the data in the POD modal subspace, in which the kriging-based surrogate model is utilized to learn the relationship between the modal coefficients of both data sets and bridge the fidelity gap. A simple adaptive sampling/infill technique is developed and incorporated into the MFROM construction process to continuously improve MFROM accuracy in the offline stage. In the online stage, MFROM replaces CFD simulation in design optimization of μCGGs using GPU-enabled GA by taking advantage of the massive parallel computing threads available within GPU, leading to significant computing acceleration.

Two case studies are performed on triple Y-shaped μCGGs to examine the modeling and design performance of the proposed method. In the first study, design variables only include the chemical concentrations at the inlets and the magnitude of the pressure differences (or differential pressures) across the mixing channels, resulting in an optimization problem within a 7-dimensional trade space. In the second, the individual pressure differences are also included as separate design variables, which is capable of generating more complex CG profiles and yields a 9-dimensional problem. Results show that generated MFROM exhibits excellent accuracy, and CGs predicted by MFROM are almost the same as those by CFD with only 0.55% and 2.78% average NRMSE in the two case studies, respectively (about 0.007 second). Due to its salient accuracy and speed,

MFROM is utilized in global design optimization of μCGGs to eliminate the need for CFD in the online optimization process, which is an important enabler for design optimization on different computing platforms. Four different prescribed profiles are supplied in both case studies for design optimization. The excellent agreement of both MFROM- and CFD-predicted CGs with prescribed CGs is observed. Specifically, 5.65% and 5.56% NRMSE for the first case study and 5.22% and 5.30% NRMSE for the second case study are attained, which verifies the design accuracy achieved by the proposed method. In conclusion, MFROM is a feasible, efficient, and accurate modeling approach for global optimization, which allows repetitive simulations with different design variables to match different prescribed CGs and potentially enables portable design in a resource-limited environment.

CHAPTER 8 CONCLUSION AND FUTURE WORK

8.1 Conclusion

This study investigates and develops optimization methodologies based on surrogate and reduced-order models for efficient and reliable design of μCGGs. The PBCM model and CFD simulation embedded in the methodology and framework serve as the data generation engines with varying fidelities. Surrogate models (including Kriging and Cokriging) and ROM are constructed using the training data generated by the engines to approximate the mapping relationship between input and output, and then used for μCGG design optimization that traditionally has been performed by a large number of function evaluations with physics-based models and is computationally prohibitive. The adaptive sampling (also termed infill) strategy is incorporated into the optimization algorithm to determine the samples at the critical location for the next round simulation to generate new data for updating the model. The intent of adaptive sampling is to continuously improve the model accuracy, hence accelerating optimization convergence. Furthermore, an RDS&DF strategy is proposed to remove the redundant data to address the big data issue for high-dimensional, complex optimization problems, where many iterations are needed to resolve the design space and search the global optimum. Several important technical findings are obtained from the case studies performed for the proposed methodologies and summarized as follows:

1. The comparative analysis proves that the combination of the first-order polynomial regression model and the Gauss correlation model forms the most accurate Kriging for modeling μCGG design using the PBCM model. It is also found that LB exhibits a faster convergence rate than other infill strategies, such as EI and PI, leading to a more accurate design given the same number of iterations.

2. For μCGG design based on SBO, both CFD-predicted and PBCM-predicted CGs match the prescribed CGs very well in the case study with the symmetric flow, which indicates the feasibility and design accuracy of the SBO method.
3. The SBO method requires at least 30 fewer function evaluations than the gradient-based optimization when only inlet concentrations are treated as design variables. For the design of inlet concentrations and pressure differences, 1/3 of gradient-based optimization has failed to search for the global optimum, while the SBO method can successfully localize the global optimum. Therefore, as a global optimization method, SBO is more robust to search the global optimum than gradient-based optimization.
4. The MFSBO method allows the utilization of data from two (or more) different sources with different natures, fidelities and computational costs (such as PBCM and CFD in this study) for μCGG design. The predicted CGs generated by MFSBO exhibit excellent agreement with the prescribed CGs, which verifies the feasibility, robustness, and accuracy of MFSBO for μCGG design.
5. MFSBO converges faster than SBO, resulting in a better design with 49% less discrepancy relative to the prescribed CGs on the average in the case study. Furthermore, MFSBO-4 achieves more accurate designs than MFSBO-1, indicating that MFSBO-4 with four parallel LF infills converges faster and to better designs for complex design problems.
6. The proposed EIR-based infill effectively searches the global optimum and accelerates the optimization process by determining both the sample location and the data source in a computation-aware and self-adaptive manner. The relative errors of the optimum designs found by the EIR-based infill reduce by 219.80%, 5.02%, and 59.42%

compared to the other three infill methods (H-only, alternating, and random) for two numerical case studies and one engineering case study, indicating the EIR-based infill improves both the optimization efficiency and accuracy.

7. The proposed RDS&DF effectively removes the redundant samples during the optimization process, and thus largely shrinks the surrogate modeling time. More importantly, in combination with the EIR-based infill, RDS&DF is even able to tailor the ratio of the already existing HF and LF samples on an as-needed basis during MFSBO and leads to sparse MFSBO with computational awareness. Without sacrificing the design accuracy, RDS&DF improves the design efficiency by reducing the optimization time by 69.49% and 73.92% for two case studies, respectively, relative to those without RDS&DF.
8. MFROM exhibits the salient evaluation speed, viz., 0.007 second/simulation, and CGs predicted by MFROM show excellent agreement with those by CFD with only 0.55% and 2.78% average NRMSE in the two case studies, respectively. Due to its accuracy and efficiency, MFROM is utilized to replace CFD simulations for the computationally intensive μCGGs design.
9. MFROM- and CFD-predicted CGs exhibit excellent agreement with the prescribed CGs with only about 5% NRMSE. It is proven that MFROM is a feasible, efficient, and accurate modeling approach and can be utilized for μCGGs design.

8.2 Future Work

This study presents several methodologies for μCGGs design, and the excellent results accomplished in the analysis shed light on several topics that can be further investigated and expanded to improve the design optimization for future research:

1. When SBO is applied to μCGG design, the regression model and the correlation model are defined *a priori* for the optimization and fixed through the optimization. This can be improved by an adaptive selection of the regression model and the correlation model strategy for improved model accuracy and infill efficiency.
2. This study employs the Cokriging model to combine the data from two different sources to reduce the computation load for improved efficiency and accuracy. However, this can be further extended to more data sources and build a model with multiple fidelities. Additional data sources that can be taken into account include experiments, CFD simulation of varying mesh densities, among others.
3. The computational budget of the EIR-based infill presented in this study is the time of one HF simulation. When there are sufficient computing resources, a larger computational budget might be used to fully utilize them. Therefore, the EIR-based infill can be extended to the scenario with multiple HF simulations or equivalent LF simulations during each infill iteration and perform an intelligent selection for both data source and infill samples in a batch mode.
4. The procedure of RDS&DF contains several user-specified parameters, which may not be optimal for different problems. Therefore, further investigation of automated parameter selection will be pursued.

5. The adaptive sampling technique applied to MFROM construction is a distance-based infill method for global modeling. Since MFROM contains the POD procedure, further investigation of adaptive sampling techniques will take into account the effect of the POD on modeling and optimization accuracy.
6. The stopping criteria in this study is the maximum number of iterations or function evaluation. In future work, the tolerance parameter of the convergence will be incorporated as the stopping criteria for automated termination of the optimization process.
7. Although RDS&DF is developed to remove redundant samples for MFSBO, big data issues could still appear when there is an extremely high-dimensional problem. Therefore, an alternative modeling technique better-suited for the higher dimensional problem and the large data sets will be considered, such as a machine learning model.

REFERENCES

Bai, Feng, and Yi Wang. 2021. “Reduced-Order Modeling Based on Hybrid Snapshot Simulation.” *International Journal of Computational Methods* 18 (01): 2050029.

Bhosekar, Atharv, and Marianthi Ierapetritou. 2018. “Advances in Surrogate Based Modeling, Feasibility Analysis, and Optimization: A Review.” *Computers and Chemical Engineering* 108: 250–67. https://doi.org/10.1016/j.compchemeng.2017.09.017.

Biddiss, Elaine, and Dongqing Li. 2005. “Electrokinetic Generation of Temporally and Spatially Stable Concentration Gradients in Microchannels.” *Journal of Colloid and Interface Science* 288 (2): 606–15. https://doi.org/10.1016/j.jcis.2005.03.037.

Cabaleiro, Juan Martin. 2020. “Flowrate Independent 3D Printed Microfluidic Concentration Gradient Generator.” *Chemical Engineering Journal* 382: 122742.

Chalupka, Krzysztof, Christopher K I Williams, and Iain Murray. 2013. “A Framework for Evaluating Approximation Methods for Gaussian Process Regression.” *Journal of Machine Learning Research* 14: 333–50.

Chen, Shishi, Zhen Jiang, Shuxing Yang, and Wei Chen. 2017. “Multimodel Fusion Based Sequential Optimization.” *AIAA Journal* 55 (1): 241–54. https://doi.org/10.2514/1.J054729.

Chen, Xiqun, Lei Zhang, Xiang He, Chenfeng Xiong, and Zhiheng Li. 2014. “Surrogate-Based Optimization of Expensive-to-Evaluate Objective for Optimal Highway Toll Charges in Transportation Network.” *Computer-Aided Civil and Infrastructure Engineering* 29 (5): 359–81. https://doi.org/10.1111/mice.12058.

Couckuyt, I., F. Declercq, T. Dhaene, H. Rogier, and L. Knockaert. 2010. “Surrogate-Based Infill Optimization Applied to Electromagnetic Problems.” *International Journal of RF and Microwave Computer-Aided Engineering* 20 (5): 492–501. https://doi.org/10.1002/mmce.20455.

Couckuyt, Ivo, Dirk Deschrijver, and Tom Dhaene. 2014. “Fast Calculation of Multiobjective Probability of Improvement and Expected Improvement Criteria for Pareto Optimization.” *Journal of Global Optimization* 60 (3): 575–94. https://doi.org/10.1007/s10898-013-0118-2.

Couckuyt, Ivo, Tom Dhaene, and Piet Demeester. 2014. “OoDACE Toolbox: A Flexible Object-Oriented Kriging Implementation.” *Journal of Machine Learning Research*

15 (1): 3183–86.

Cozad, Alison, Nikolaos V. Sahinidis, and David C. Miller. 2014. “Learning Surrogate Models for Simulation-Based Optimization.” *AIChE Journal* 60 (6): 2211–27. https://doi.org/10.1002/aic.14418.

Das, Sourish, Sasanka Roy, and Rajiv Sambasivan. 2018. “Fast Gaussian Process Regression for Big Data.” *Big Data Research* 14: 12–26. https://doi.org/10.1016/j.bdr.2018.06.002.

Dertinger, S. K.W., D. T. Chiu, Noo Li Jeon, and G. M. Whitesides. 2001. “Generation of Gradients Having Complex Shapes Using Microfluidic Networks.” *Analytical Chemistry* 73 (6): 1240–46. https://doi.org/10.1021/ac001132d.

Etikan, Ilker. 2016. “Comparison of Convenience Sampling and Purposive Sampling.” *American Journal of Theoretical and Applied Statistics* 5 (1): 1. https://doi.org/10.11648/j.ajtas.20160501.11.

Fernández-Godino, M Giselle, Chanyoung Park, Nam-Ho Kim, and Raphael T Haftka. 2016. “Review of Multi-Fidelity Models.” *ArXiv Preprint ArXiv:1609.07196*.

Ferrari, Rosalba, Diego Froio, Egidio Rizzi, Carmelo Gentile, and Eleni N Chatzi. 2019. “Model Updating of a Historic Concrete Bridge by Sensitivity-and Global Optimization-Based Latin Hypercube Sampling.” *Engineering Structures* 179: 139–60.

Forrester, Alexander I. J., András Sóbester, and Andy J. Keane. 2008. *Engineering Design via Surrogate Modelling*. *Engineering Design via Surrogate Modelling*. https://doi.org/10.1002/9780470770801.

Forrester, Alexander I.J., and Andy J. Keane. 2009. “Recent Advances in Surrogate-Based Optimization.” *Progress in Aerospace Sciences* 45 (1–3): 50–79. https://doi.org/10.1016/j.paerosci.2008.11.001.

Forrester, Alexander I.J., András Sóbester, and Andy J. Keane. 2007. “Multi-Fidelity Optimization via Surrogate Modelling.” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 463 (2088): 3251–69. https://doi.org/10.1098/rspa.2007.1900.

Friedrich, Daniel, Colin P. Please, and Tracy Melvin. 2012. “Design of Novel Microfluidic Concentration Gradient Generators Suitable for Linear and Exponential Concentration Ranges.” *Chemical Engineering Journal* 193–194: 296–303. https://doi.org/10.1016/j.cej.2012.04.041.

Giselle Fernández-Godino, M., Chanyoung Park, Nam H. Kim, and Raphael T. Haftka. 2019. “Issues in Deciding Whether to Use Multifidelity Surrogates.” *AIAA Journal* 57 (5): 2039–54. https://doi.org/10.2514/1.J057750.

Gorman, Bryan R., and John P. Wikswo. 2008. “Characterization of Transport in Microfluidic Gradient Generators.” *Microfluidics and Nanofluidics* 4 (4): 273–85. https://doi.org/10.1007/s10404-007-0169-0.

Gramacy, Robert B. 2016. “LaGP: Large-Scale Spatial Modeling via Local Approximate Gaussian Processes in R.” *Journal of Statistical Software* 72 (1). https://doi.org/10.18637/jss.v072.i01.

Gratiet, Loic Le, and Claire Cannamela. 2015. “Cokriging-Based Sequential Design Strategies Using Fast Cross-Validation Techniques for Multi-Fidelity Computer Codes.” *Technometrics* 57 (3): 418–27. https://doi.org/10.1080/00401706.2014.928233.

Guénot, Marc, Ingrid Lepot, Caroline Sainvitu, Jordan Goblet, and Rajan Filomeno Coelho. 2013. “Adaptive Sampling Strategies for Non-Intrusive POD-Based Surrogates.” *Engineering Computations (Swansea, Wales)* 30 (4): 521–47. https://doi.org/10.1108/02644401311329352.

Gutierrez-Castillo, Paloma, and Becca Thomases. 2019. “Proper Orthogonal Decomposition (Pod) of the Flow Dynamics for a Viscoelastic Fluid in a Four-Roll Mill Geometry at the Stokes Limit.” *Journal of Non-Newtonian Fluid Mechanics* 264: 48–61.

Haftka, Raphael T., Diane Villanueva, and Anirban Chaudhuri. 2016. “Parallel Surrogate-Assisted Global Optimization with Expensive Functions – a Survey.” *Structural and Multidisciplinary Optimization* 54 (1): 3–13. https://doi.org/10.1007/s00158-016-1432-3.

Han, Zhong-Hua, and Ke-Shi Zhang. 2012. “Surrogate-Based Optimization.” *Real-World Applications of Genetic Algorithms*, 343–62.

Han, Zhong Hua, Jing Chen, Ke Shi Zhang, Zhen Ming Xu, Zhen Zhu, and Wen Ping Song. 2018. “Aerodynamic Shape Optimization of Natural-Laminar-Flow Wing Using Surrogate-Based Approach.” *AIAA Journal* 56 (7): 2579–93. https://doi.org/10.2514/1.J056661.

Hattori, Koji, Shinji Sugiura, and Toshiyuki Kanamori. 2009. “Generation of Arbitrary Monotonic Concentration Profiles by a Serial Dilution Microfluidic Network Composed of Microchannels with a High Fluidic-Resistance Ratio<.” *Lab on a Chip* 9 (12): 1763–72. https://doi.org/10.1039/b816995k.

Hensman, James, Nicolo Fusi, and Neil D Lawrence. 2013. “Gaussian Processes for Big Data.” *ArXiv Preprint ArXiv:1309.6835*.

Hong, Bo, Peng Xue, Yafeng Wu, Jingnan Bao, Yon Jin Chuah, and Yuejun Kang. 2016. “A Concentration Gradient Generator on a Paper-Based Microfluidic Chip Coupled with Cell Culture Microarray for High-Throughput Drug Screening.” *Biomedical Microdevices* 18 (1): 1–8. https://doi.org/10.1007/s10544-016-0054-2.

Hong, Seong Hyeon, Jackson Cornelius, Yi Wang, and Kapil Pant. 2019. “Fault Compensation by Online Updating of Genetic Algorithm-Selected Neural Network Model for Model Predictive Control.” *SN Applied Sciences* 1 (11): 1–16.

Hong, Seong Hyeon, Jung-Il Shu, Yi Wang, and Oktay Baysal. 2021. “Automated Optimization of Double Heater Convective Polymerase Chain Reaction Devices Based on CFD Simulation Database and Artificial Neural Network Model.” *Biomedical Microdevices* 23 (2): 1–14.

Hong, Seong Hyeon, Haizhou Yang, and Yi Wang. 2020. “Inverse Design of Microfluidic Concentration Gradient Generator Using Deep Learning and Physics-based Component Model.” *Microfluidics and Nanofluidics* 24 (6).

Höving, Stefan, Dirk Janasek, and Pedro Novo. 2018. “Flow Rate Independent Gradient Generator and Application in Microfluidic Free-Flow Electrophoresis.” *Analytica Chimica Acta* 1044: 77–85. https://doi.org/10.1016/j.aca.2018.04.066.

Huang, D., T. T. Allen, W. I. Notz, and R. A. Miller. 2006. “Sequential Kriging Optimization Using Multiple-Fidelity Evaluations.” *Structural and Multidisciplinary Optimization* 32 (5): 369–82. https://doi.org/10.1007/s00158-005-0587-0.

Irimia, Daniel, Dan A. Geba, and Mehmet Toner. 2006. “Universal Microfluidic Gradient Generator.” *Analytical Chemistry* 78 (10): 3472–77. https://doi.org/10.1021/ac0518710.

Jahed Armaghani, Danial, Mahdi Hasanipanah, Amir Mahdiyar, Muhd Zaimi Abd Majid, Hassan Bakhshandeh Amnieh, and Mahmood M.D. Tahir. 2018. “Airblast Prediction through a Hybrid Genetic Algorithm-ANN Model.” *Neural Computing and Applications* 29 (9): 619–29. https://doi.org/10.1007/s00521-016-2598-8.

Jeon, Jinhyeok, Namhyun Choi, Hao Chen, Joung Il Moon, Lingxin Chen, and Jaebum Choo. 2019. “SERS-Based Droplet Microfluidics for High-Throughput Gradient Analysis.” *Lab on a Chip* 19 (4): 674–81. https://doi.org/10.1039/C8LC01180J.

Kato, Hiromasa, and Ken-ichi Funazaki. 2014. “POD-Driven Adaptive Sampling for Efficient Surrogate Modeling and Its Application to Supersonic Turbine Optimization.” In *Turbo Expo: Power for Land, Sea, and Air*, 45615:V02BT45A023. American Society of Mechanical Engineers.

Katoch, Sourabh, Sumit Singh Chauhan, and Vijay Kumar. 2021. “A Review on Genetic Algorithm: Past, Present, and Future.” *Multimedia Tools and Applications* 80 (5): 8091–8126.

Kaya, Halil, Hakan Tiftikçi, Ümit Kutluay, and Evren Sakarya. 2019. “Generation of Surrogate-Based Aerodynamic Model of an UCAV Configuration Using an Adaptive Co-Kriging Method.” *Aerospace Science and Technology* 95: 105511. https://doi.org/10.1016/j.ast.2019.105511.

Kuya, Yuichi, Kenji Takeda, Xin Zhang, and Alexander I.J. Forrester. 2011. "Multifidelity Surrogate Modeling of Experimental and Computational Aerodynamic Data Sets." *AIAA Journal* 49 (2): 289–98. https://doi.org/10.2514/1.J050384.

Laguna, Manuel, and Rafael Marti. 2005. "Experimental Testing of Advanced Scatter Search Designs for Global Optimization of Multimodal Functions." *Journal of Global Optimization* 33 (2): 235–55.

Lawrence, Neil, Matthias Seeger, and Ralf Herbrich. 2003. "Fast Sparse Gaussian Process Methods: The Informative Vector Machine." *Advances in Neural Information Processing Systems*.

Lee, Byung Jun, Jongmin Lee, and Kee Eung Kim. 2017. "Hierarchically-Partitioned Gaussian Process Approximation." *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017* 54.

Liu, Bo, Slawomir Koziel, and Qingfu Zhang. 2016. "A Multi-Fidelity Surrogate-Model-Assisted Evolutionary Algorithm for Computationally Expensive Optimization Problems." *Journal of Computational Science* 12: 28–37. https://doi.org/10.1016/j.jocs.2015.11.004.

Liu, Haitao, Yew-soon Ong, Xiaobo Shen, and Jianfei Cai. 2020. "When Gaussian Process Meets Big Data: A Review of Scalable GPs." *IEEE Transactions on Neural Networks and Learning Systems*, 1–19. https://doi.org/10.1109/tnnls.2019.2957109.

Liu, Jun, Zhonghua Han, and Wenping Song. 2012. "Comparison of Infill Sampling Criteria in Kriging-Based Aerodynamic Optimization." *28th Congress of the International Council of the Aeronautical Sciences 2012, ICAS 2012* 2: 1625–34.

Marques, Alexandre N., Remi R. Lam, Anirban Chaudhuri, Max M.J. Opgenoord, and Karen E. Willcox. 2019. "A Multifidelity Method for Locating Aeroelastic Flutter Boundaries." *AIAA Scitech 2019 Forum*, no. January: 1–16. https://doi.org/10.2514/6.2019-0438.

Melkumyan, Arman, and Fabio Tozeto Ramos. 2009. "A Sparse Covariance Function for Exact Gaussian Process Inference in Large Datasets." In *Twenty-First International Joint Conference on Artificial Intelligence*.

Mulholland, Theresa, Milly McAllister, Samantha Patek, David Flint, Mark Underwood, Alexander Sim, Joanne Edwards, and Michele Zagnoni. 2018. "Drug Screening of Biopsy-Derived Spheroids Using a Self-Generated Microfluidic Concentration Gradient." *Scientific Reports* 8 (1): 1–12. https://doi.org/10.1038/s41598-018-33055-0.

Park, Chanyoung, Raphael T. Haftka, and Nam H. Kim. 2017. "Remarks on Multi-Fidelity Surrogates." *Structural and Multidisciplinary Optimization* 55 (3): 1029–50. https://doi.org/10.1007/s00158-016-1550-y.

Parr, J. M., A. J. Keane, A. I.J. Forrester, and C. M.E. Holden. 2012. “Infill Sampling Criteria for Surrogate-Based Optimization with Constraint Handling.” *Engineering Optimization* 44 (10): 1147–66. https://doi.org/10.1080/0305215X.2011.637556.

Peherstorfer, Benjamin, Karen Willcox, and Max Gunzburger. 2018. “Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization.” *SIAM Review* 60 (3): 550–91. https://doi.org/10.1137/16M1082469.

Perdikaris, Paris, Maziar Raissi, Andreas Damianou, Neil D Lawrence, and George Em Karniadakis. 2017. “Nonlinear Information Fusion Algorithms for Data-Efficient Multi-Fidelity Modelling.” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473 (2198): 20160751.

Picheny, Victor, Tobias Wagner, and David Ginsbourger. 2013. “A Benchmark of Kriging-Based Infill Criteria for Noisy Optimization.” *Structural and Multidisciplinary Optimization* 48 (3): 607–26.

Quesada, Carlos, Pierre Villon, and Anne-Virginie Salsac. 2021. “Real-Time Prediction of the Deformation of Microcapsules Using Proper Orthogonal Decomposition.” *Journal of Fluids and Structures* 101: 103193.

Rai, Rahul, and Matthew I. Campbell. 2007. “Q2S2: Qualitative and Quantitative Sequential Sampling - A Novel Approach to Exploit Qualitative Design Information.” In *Proceedings of ICED 2007, the 16th International Conference on Engineering Design*.

Rismanian, Milad, Mohammad Said Saidi, and Navid Kashaninejad. 2019. “A New Non-Dimensional Parameter to Obtain the Minimum Mixing Length in Tree-like Concentration Gradient Generators.” *Chemical Engineering Science* 195: 120–26. https://doi.org/10.1016/j.ces.2018.11.041.

Robertson, Eric D, Yi Wang, Kapil Pant, Matthew J Grismer, and José A Camberos. 2018. “A Flow Feature Detection Framework for Large-Scale Computational Data Based on Incremental Proper Orthogonal Decomposition and Data Mining.” *International Journal of Computational Fluid Dynamics* 32 (6–7): 261–77.

Rullière, Didier, Nicolas Durrande, François Bachoc, and Clément Chevalier. 2018. “Nested Kriging Predictions for Datasets with a Large Number of Observations.” *Statistics and Computing* 28 (4): 849–67.

Sadava, David E, David M Hillis, and H Craig Heller. 2009. *Life: The Science of Biology*. Vol. 2. Macmillan.

Seemann, Ralf, Martin Brinkmann, Thomas Pfohl, and Stephan Herminghaus. 2011. “Droplet Based Microfluidics.” *Reports on Progress in Physics* 75 (1): 16601.

Shi, Renhe, Li Liu, Teng Long, Yufei Wu, and Yifan Tang. 2018. “Dual-Sampling Based Co-Kriging Method for Design Optimization Problems with Multi-Fidelity Models.”

2018 Multidisciplinary Analysis and Optimization Conference, 1–14. https://doi.org/10.2514/6.2018-3747.

Shu, Jung-Il, Seong Hyeon Hong, Yi Wang, and Oktay Baysal. 2020. "Surrogate-and Possibility-Based Design Optimization for Convective Polymerase Chain Reaction Devices." *Microsystem Technologies*, 1–16.

Singh, Prashant, Ivo Couckuyt, Khairy Elsayed, Dirk Deschrijver, and Tom Dhaene. 2016. "Shape Optimization of a Cyclone Separator Using Multi-Objective Surrogate-Based Optimization." *Applied Mathematical Modelling* 40 (5–6): 4248–59. https://doi.org/10.1016/j.apm.2015.11.007.

Snelson, Edward, and Zoubin Ghahramani. 2007. "Local and Global Sparse Gaussian Process Approximations." *Journal of Machine Learning Research* 2: 524–31.

Tang, Minghui, Xinyu Huang, Qian Chu, Xinghai Ning, Yuye Wang, Siu Kai Kong, Xuping Zhang, Guanghui Wang, and Ho Pui Ho. 2018. "A Linear Concentration Gradient Generator Based on Multi-Layered Centrifugal Microfluidics and Its Application in Antimicrobial Susceptibility Testing." *Lab on a Chip* 18 (10): 1452–60. https://doi.org/10.1039/c8lc00042e.

Taylor, Charles E. 1994. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. Complex Adaptive Systems. John H. Holland . The Quarterly Review of Biology*. Vol. 69. MIT press. https://doi.org/10.1086/418447.

Toh, Alicia G.G., Z. P. Wang, Chun Yang, and Nam Trung Nguyen. 2014. "Engineering Microfluidic Concentration Gradient Generators for Biological Applications." *Microfluidics and Nanofluidics* 16 (1–2): 1–18. https://doi.org/10.1007/s10404-013-1236-3.

Wang, Hao, Michael Emmerich, Bas Van Stein, and Thomas Back. 2017. "Time Complexity Reduction in Efficient Global Optimization Using Cluster Kriging." *GECCO 2017 - Proceedings of the 2017 Genetic and Evolutionary Computation Conference*, 889–96. https://doi.org/10.1145/3071178.3071321.

Wang, Xiang, Zhaomiao Liu, and Yan Pang. 2017. "Concentration Gradient Generation Methods Based on Microfluidic Systems." *RSC Advances* 7 (48): 29966–84. https://doi.org/10.1039/c7ra04494a.

Wang, Yi, Qiao Lin, and Tamal Mukherjee. 2005. "A Model for Laminar Diffusion-Based Complex Electrokinetic Passive Micromixers." *Lab on a Chip* 5 (8): 877–87. https://doi.org/10.1039/b500010f.

Wang, Yi, Tamal Mukherjee, and Qiao Lin. 2006. "Systematic Modeling of Microfluidic Concentration Gradient Generators." *Journal of Micromechanics and Microengineering* 16 (10): 2128–37. https://doi.org/10.1088/0960-1317/16/10/029.

Wilson, Andrew, and Hannes Nickisch. 2015. “Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP).” In *International Conference on Machine Learning*, 1775–84. PMLR.

Yang, Chun Guang, Ying Fan Wu, Zhang Run Xu, and Jian Hua Wang. 2011. “A Radial Microfluidic Concentration Gradient Generator with High-Density Channels for Cell Apoptosis Assay.” *Lab on a Chip* 11 (19): 3305–12. https://doi.org/10.1039/c1lc20123a.

Yang, Haizhou, Seong Hyeon Hong, Rei ZhG, and Yi Wang. 2020. “Surrogate-Based Optimization with Adaptive Sampling for Microfluidic Concentration Gradient Generator Design.” *RSC Advances* 10 (23): 13799–814.

Zhang, Zhen, Xiang Yu Kong, Kai Xiao, Qian Liu, Ganhua Xie, Pei Li, Jie Ma, Ye Tian, Liping Wen, and Lei Jiang. 2015. “Engineered Asymmetric Heterogeneous Membrane: A Concentration-Gradient-Driven Energy Harvesting Device.” *Journal of the American Chemical Society* 137 (46): 14765–72. https://doi.org/10.1021/jacs.5b09918.

Zhou, Qi, Yan Wang, Seung Kyum Choi, Ping Jiang, Xinyu Shao, and Jiexiang Hu. 2017. “A Sequential Multi-Fidelity Metamodeling Approach for Data Regression.” *Knowledge-Based Systems* 134 (January 2019): 199–212. https://doi.org/10.1016/j.knosys.2017.07.033.

Zhou, Yao, Yi Wang, Tamal Mukherjee, and Qiao Lin. 2009. “Generation of Complex Concentration Profiles by Partial Diffusive Mixing in Multi-Stream Laminar Flow.” *Lab on a Chip* 9 (10): 1439–48. https://doi.org/10.1039/b818485b.