

Probabilistic and Statistical Learning Models for Error Modeling and Uncertainty Quantification

Azam Moosavi

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Adrian Sandu, Chair
Calvin J. Ribbens
Bert Huang
Serkan Gugercin
Archibald K. Archibald

February 14, 2018
Blacksburg, Virginia

Keywords: Uncertainty Quantification, Uncertainty Reduction, Stochastic Simulation of
Chemical Reactions, Reduced-Order Models, Structural Uncertainty, Data Assimilation,
Numerical Weather Prediction Models, Machine Learning
Copyright 2018, Azam Moosavi

Probabilistic and statistical learning models for error modeling and uncertainty quantification

Azam Moosavi

(ABSTRACT)

Simulations and modeling of large-scale systems are vital to understanding real world phenomena. However, even advanced numerical models can only approximate the true physics. The discrepancy between model results and nature can be attributed to different sources of uncertainty including the parameters of the model, input data, or some missing physics that is not included in the model due to a lack of knowledge or high computational costs. Uncertainty reduction approaches seek to improve the model accuracy by decreasing the overall uncertainties in models.

Aiming to contribute to this area, this study explores uncertainty quantification and reduction approaches for complex physical problems. This study proposes several novel probabilistic and statistical approaches for identifying the sources of uncertainty, modeling the errors, and reducing uncertainty to improve the model predictions for large-scale simulations. We explore different computational models. The first class of models studied herein are inherently stochastic, and numerical approximations suffer from stability and accuracy issues. The second class of models are partial differential equations, which capture the laws of mathematical physics; however, they only approximate a more complex reality, and have uncertainties due to missing dynamics which is not captured by the models. The third class are low-fidelity models, which are fast approximations of very expensive high-fidelity models. The reduced-order models have uncertainty due to loss of information in the dimension reduction process. We also consider uncertainty analysis in the data assimilation framework, specifically for ensemble based methods where the effect of sampling errors is alleviated by localization. Finally, we study the uncertainty in numerical weather prediction models coming from approximate descriptions of physical processes.

Probabilistic and statistical learning models for error modeling and uncertainty quantification

Azam Moosavi

(GENERAL AUDIENCE ABSTRACT)

Computational models are used to understand the behavior of the natural phenomenon. Models are used to approximate the evolution of the true phenomenon or reality in time. We obtain more accurate forecast for the future by combining the model approximation together with the observation from reality. Weather forecast models, oceanography, geoscience, etc. are some examples of the forecasting models. However, models can only approximate the true reality to some extent and model approximation of reality is not perfect due to several sources of error or uncertainty. The noise in measurements or in observations from nature, the uncertainty in some model components, some missing components in models, the interaction between different components of the model, all cause model forecast to be different from reality.

The aim of this study is to explore the techniques and approaches of modeling the error and uncertainty of computational models, provide solution and remedies to reduce the error of model forecast and ultimately improve the model forecast. Taking the discrepancy or error between model forecast and reality in time and mining that error provide valuable information about the origin of uncertainty in models as well as the hidden dynamics that is not considered in the model. Statistical and machine learning based solutions are proposed in this study to identify the source of uncertainty, capturing the uncertainty and using that information to reduce the error and enhancing the model forecast. We studied the error modeling, error or uncertainty quantification and reduction techniques in several frameworks from chemical models to weather forecast models. In each of the models, we tried to provide proper solution to detect the origin of uncertainty, model the error and reduce the uncertainty to improve the model forecast.

Dedication

To my family for their love and support.

Acknowledgments

First and foremost I would like to express my sincerest gratitude to my advisor Prof. Adrian Sandu for his invaluable encouragement, patience and excessive support during my graduate studies. I was always feeling so blessed and honored to do my Ph.D program having Prof. Sandu as my mentor and advisor.

I would like to thank my advisory committee Prof. Bert Huang, Dr. Richard K. Archibald, Prof. Serkan Gugercin and Prof. Calvin Ribbens for their time, advice and support toward finishing my PhD.

I would like to acknowledge my friends at Computational Science Laboratory (CSL) who have affected immensely to my personal and professional time at Virginia Tech. Collaborating and interacting with them made me grow into a person of dedication, integrity and determination.

I gratefully acknowledge the funding sources that made my Ph.D. work possible. They include the awards NSF DMS-1419003, NSF CCF-1218454, AFOSR FA9550-12-1-0293-DEF, AFOSR 12-2640-06, AFOSR DDDAS 15RT1037, NSF CCF 1218454, NSF CCF-1613905, the AFOSR Computational Mathematics, and the Computational Science Laboratory (CSL) in Department of Computer Science at Virginia Tech.

Contents

1	Introduction	1
1.1	Scientific context	1
1.2	Research objectives	2
1.3	Research accomplishments	4
1.3.1	Approximate exponential algorithms to solve the chemical master equation	5
1.3.2	Metropolis-Hastings sampling approaches to solve stochastic chemical kinetics	5
1.3.3	Structural uncertainty analysis in physical models	5
1.3.4	Error prediction of local parametric reduced order models	6
1.3.5	Adaptive localization in data assimilation	6
1.3.6	Uncertainty analysis in numerical weather prediction models	7
1.4	Dissertation layout	7
2	Approximate Exponential Algorithms to Solve the Chemical Master Equation	9
2.1	Introduction	9
2.2	Simulation of stochastic chemical kinetics	10
2.2.1	Chemical Master Equation	11
2.2.2	Approximation to Chemical Master Equation	12

2.2.3	Tau-leaping method	13
2.3	Approximations to the exponential solution	13
2.3.1	Strang splitting	13
2.3.2	Column based splitting	14
2.3.3	Accelerated tau-leaping	15
2.3.4	Symmetric accelerated tau-leaping	16
2.4	Numerical experiments	16
2.4.1	Isomer reaction	17
2.4.2	Schlogl reaction	19
2.5	Conclusions	20
Appendix A. Examples		23
3 Solving Stochastic Chemical Kinetics by Metropolis-Hastings Sampling		26
3.1	Introduction	26
3.2	Markov Chain Monte Carlo methods	27
3.2.1	Metropolis methods	27
3.2.2	Metropolis-Hastings algorithm	28
3.3	Metropolis-Hastings for stochastic simulation	29
3.3.1	Target distribution	29
3.3.2	Proposal distribution	30
3.3.3	Markov process	31
3.4	Matrix exponential	31
3.4.1	Rational approximation methods for full matrix exponential	32
3.4.2	Krylov based approximation for matrix vector products	33
3.4.3	Faster approximation techniques for a single element	34

3.5	Numerical experiments	35
3.5.1	Schlogl reaction	35
3.5.2	Isomer reaction	37
3.5.3	Lotka Volterra reaction	37
3.6	Conclusions	39
4	A State-Space Approach to Analyze Structural Uncertainty in Physical Models	40
4.1	Introduction	40
4.2	Structural uncertainty in computer simulation models	42
4.2.1	Global models for structural uncertainty	44
4.2.2	Local models for structural uncertainty	47
4.3	Numerical experiments	48
4.3.1	Lorenz-96 model	48
4.3.2	Stratospheric chemistry model	59
4.4	Conclusions	62
5	Multivariate Predictions of Local Reduced-Order-Model errors and Dimensions	64
5.1	Introduction	64
5.2	Parametric reduced-order modeling	67
5.3	Multivariate prediction of local reduced-order models characteristics (MP-LROM)	68
5.3.1	Error Model	69
5.3.2	Dimension of the reduced basis	69
5.3.3	Supervised Machine Learning Techniques	70
5.4	Numerical experiments	74

5.4.1	One-dimensional Burgers' equation	75
5.4.2	Multivariate prediction of local reduced-order models characteristics (MP-LROM) using regression machine learning methods . . .	77
5.5	Conclusions	87
6	Parametric Domain Decomposition for Accurate Reduced Order Models: Applications of MP-LROM Methodology	95
6.1	Introduction	95
6.2	MP-LROM error model and its application to parametric domain decomposition	98
6.2.1	Designing the decomposition of the parametric domain	98
6.2.2	Combining available information for accurate local ROMs at arbitrary parametric configurations	101
6.3	Numerical experiments	107
6.3.1	Designing the decomposition of the parametric domain	108
6.3.2	Combining available information for accurate local ROMs at arbitrary parametric configurations	113
6.4	Conclusions	117
7	A Machine Learning Approach to Adaptive Covariance Localization	119
7.1	Introduction	119
7.2	Background	122
7.2.1	Ensemble Kalman filters	122
7.2.2	Inbreeding, filter divergence, and spurious correlations	123
7.2.3	Covariance localization	124
7.2.4	Machine learning	125
7.3	Machine Learning Approach for Adaptive Localization	126
7.3.1	Features and decision criteria	127

7.3.2	ML-based adaptive localization algorithm	129
7.3.3	Adaptive localization in time	129
7.3.4	Adaptive localization in time and space	130
7.3.5	Computational considerations	131
7.4	Setup of the Numerical Experiments	131
7.4.1	Forward operator	131
7.5	Numerical Results	132
7.5.1	Lorenz model with adaptive localization in time	132
7.5.2	QG model with adaptive localization in time	133
7.5.3	Lorenz model with adaptive localization in time and space	137
7.5.4	QG model with adaptive localization in time and space	137
7.5.5	On the choice of the learning model	141
7.5.6	Discussion of numerical results	143
7.6	Concluding Remarks and Future Work	143
8	A Learning Based Approach for Uncertainty Analysis in Numerical Weather Prediction Models	146
8.1	Introduction	146
8.2	Model errors	148
8.3	Approximating model errors using machine learning	151
8.3.1	Problem one: estimating in advance aspects of interest of the model error	152
8.3.2	Problem two: identifying the physical packages that contribute most to the forecast uncertainty	153
8.3.3	Machine learning algorithms	155
8.4	Numerical experiments	156
8.4.1	The WRF model	157

8.4.2	Experiments for problem one: predicting pointwise precipitation forecast errors over a small geographic region	160
8.4.3	Experiments for problem one: predicting the norm of precipitation forecast error over the entire domain	164
8.4.4	Experiments for problem two: identify the physical processes that contribute most to the forecast uncertainty	166
8.5	Conclusions	167
9	Conclusions and Future Research Directions	169
	Bibliography	172

List of Figures

2.1	Histograms of the isomer system (2.17) results at the final time $T=10$. . .	18
2.2	Isomer system (2.17) solutions provided by the traditional tau-leap (2.8) and by accelerated tau-leap (2.13) methods at the final time $T=10$ (units). A small time step of $\tau = 0.01$ (units) is used. The number of samples for both methods is 10,000.	19
2.3	Histograms of isomer system (2.17) solutions obtained with SSA, traditional tau-leap (2.8), and symmetric accelerated tau-leap (2.16) methods at the final time $T=10$. The number of samples is 10,000 for all methods. .	20
2.4	Plot of errors vs. CPU times for different algorithms. The points on the graphs correspond to the following five time steps: $\tau = 0.01$, $\tau = 0.05$, $\tau = 0.075$, $\tau = 0.1$, and $\tau = 0.15$ (units). The final time $T=10$ (units). Each point is obtained from 10,000 samples/runs.	21
2.5	Histograms of Schlogl system (2.18) results at final time $T=4$ (units). . . .	22
2.6	Histograms of Schlogl system (2.18) solutions with $\tau = 0.0001$ (units), final time $T=4$ (units), and 10,000 samples.	22
3.1	Histograms of Schlogl system (2.18) solutions with $\tau = 0.4$ (units), final time $T=4$ (units), and 10,000 samples.	36
3.2	Histograms of isomer system (2.17) solutions with $\tau = 0.05$ (units), final time $T=1$ (units), and 10,000 samples.	38
3.3	Histograms of Lotka-Volterra system (3.24) solutions with $\tau = 0.01$ (units), final time $T=1$ (units), and 10,000 samples.	39

4.1	The evolution of a Lorenz variable in time for model solution, real solution and corrected model solution. The corrected model solution of the Lorenz system is obtained using the predicted global model error. The 12 th and 30 th variables in the model are perturbed, meaning that they are affected by the structural uncertainty.	50
4.2	The structure of the SSM coefficient matrices for the global error model. The structural uncertainty affects the 12 th and 30 th variables in the Lorenz model (7.9).	52
4.3	The structure of the SSM coefficient matrices for the global error model. The structural uncertainty affects the 8 th , 16 th , 25 th and 33 rd variables in the Lorenz model (7.9).	53
4.4	The evolution of a Lorenz variable in time for model solution, real solution and corrected model solution. The corrected model solution is obtained using the predicted local error. The structural uncertainty affects the 22 nd variable in the model, and the localization radius is one.	54
4.5	The local error SSM coefficients for the Lorenz system (7.9). The structural uncertainty affects the 22 nd variable in the model, and the localization radius is one.	55
4.6	The evolution of a Lorenz variable in time for model solution, real solution and corrected model solution. The corrected model solution is obtained through applying the local model discrepancy. The structural uncertainty affects the 22 nd variable in the model, and the localization radius is five.	56
4.7	The local error SSM coefficients for the Lorenz system (7.9). The structural uncertainty affects the 22 nd variable in the model, and the localization radius is five.	57
4.8	Histogram of minimum root mean square discrepancies between the corrected solution and the exact trajectory using the local error modeling approach.	58
4.9	Model forecasts and corrected model solutions for the stratospheric chemical system with missing reactions. A global model of structural uncertainty is used.	60
4.10	The structure of the matrix coefficients of the global-error SSM for the stratospheric chemical system.	61

5.1	Seventh order polynomial used as initial conditions for 1D Burgers model.	76
5.2	Computational efficiency of the tensorial POD 1D Burgers model. CPU time is given in seconds.	76
5.3	Isocontours of the reduced model errors for different POD basis dimensions and parameters μ . The reduced-order model uses a basis constructed from the full order simulation with parameter value $\mu_p = 0.8$	78
5.15	Average error of the POD dimension prediction on a randomly selected test data with desired accuracy of $\bar{\epsilon} = 10^{-3}$. The average of the absolute values of the error in prediction are 1.31, 0.21 and 1.38 for singular value based method and MP-LROM models constructed using ANN and GP. . .	87
5.4	Histogram of errors in prediction using ANN MP-LROM.	89
5.5	Histogram of errors in prediction using GP MP-LROM.	90
5.6	The average of errors in predictions using five different trained models. The top labels shows the corresponding μ_p and K_{POD} for each parameter μ .	91
5.7	Isocontours for the E_{fold} using GP method.	91
5.8	Isocontours for the E_{fold} using ANN method.	91
5.9	Isocontours for the VAR_{fold} using GP method.	92
5.10	Isocontours for the VAR_{fold} using ANN method.	92
5.11	The error in predictions of all methods using $K_{POD} = 10$ and $\mu_p = 1$. For GP error models the overall average of errors in predictions is 0.0131, 0.0487, 0.0095 for MFC, ROMES and MP-LROM respectively. For ANN error models the overall average of errors in predictions is 0.0056, 0.0240, 0.0029 for MFC, ROMES and MP-LROM respectively. The top x-axis shows the corresponding logarithms of the residuals norms used as inputs for the ROMES method.	93
5.12	Isocontours of the reduced model errors for different POD basis dimensions and viscosity parameters μ_p	93
5.13	Histogram of errors in prediction of the reduced basis dimension using ANN MP-LROM for different sample sizes	94
5.14	Histogram of errors in prediction of the reduced basis dimension using GP ML-ROM for different sample sizes	94

6.1	The description of the interpolation of four subspaces in a tangent space to a Grassmann manifold. The origin point of the interpolation is $S_{\mu_{pM}}$ and $m = 3$	103
6.2	The average range of parameter μ obtained with MP-LROM models for $K_{POD} = 9$ and $\mu_p = 0.7$. The desired accuracy is $\bar{\epsilon} = 10^{-2}$. The numbers represent the left and the right edges of the predicted vs the true feasible intervals.	110
6.3	A description of the most important stages of the parameter domain decomposition algorithm. The arrows describe the internal steps for each stage initiated in the order depicted by the arrows' indices.	111
6.4	The diffusion parametric domain decomposition defining the local feasible intervals and their corresponding errors. Associated with one feasible interval there is a centered parameter μ_p high-fidelity trajectory that guides the construction of a reduced basis and operators such that the subsequent reduced-order model solutions along this interval are accurate to within the threshold depicted by the Y-axis labels.	113
6.5	Strategies comparison for generation of accurate ROMs for a new viscosity parameter $\mu^* = 0.35$ and $t_f = 0.01$. In panel (a) K_{POD} is set to 14, whereas in panel (b) ν is set to 1.	116
6.6	Strategies comparison for generation of accurate ROMs for a new viscosity parameter $\mu^* = 0.35$ and $t_f = 1$. In panel (a) K_{POD} is set to 14, whereas in panel (b) ν is set to 1.	117
7.1	Data assimilation with the Lorenz-96 model (7.9). EnKF is applied with a Gaussian covariance localization function. EnKF results with adaptive covariance localization are shown for different choices of the weighting factors w_1, w_2 . The localization is adaptive in time, and is compared to results with fixed localization radius. The training phase consists of 80 assimilation cycles, followed by the testing phase with 20 assimilation cycles. The ML algorithm used here is RF. <i>The overall performance of the adaptive localization is better than that of the hand-tuned radius.</i>	133
7.2	Data assimilation with the Lorenz-96 model (7.9). Shown is the evolution of the localization radius in time over all 100 assimilation cycles. The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$	134

7.3	The 35 most important features of the Lorenz-96 model (7.9)	134
7.4	Data assimilation with the QG model (7.10). EnKF is applied with a GC covariance localization function. The (<i>log</i>) RMSE is shown on the vertical axis, and the time (assimilation cycles) is shown on the horizontal axis. EnKF results with adaptive covariance localization are shown for different choices of the weighting factors w_1, w_2 . The localization is adaptive in time, and is compared to results with fixed localization radius. The training phase consists of 80 assimilation cycles, followed by the testing phase with 20 assimilation cycles. The ML algorithm used here is RF. <i>The analysis results with adaptive localization outperform those obtained with the hand-tuned radius.</i>	135
7.5	Data assimilation with the QG model (7.10). Shown is the evolution of the localization radius in time over all 100 assimilation cycles. The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$	136
7.6	The 35 most important features of the QG model (7.10)	136
7.7	Data assimilation with the Lorenz-96 model (7.9). EnKF is applied with a Gaussian covariance localization function. EnKF results with adaptive covariance localization are shown for different choices of the weighting factors w_1, w_2 . The localization is adaptive in time and space, and is compared to results with fixed localization radius. The training phase consists of 800 assimilation cycles, followed by the testing phase with 200 assimilation cycles. The ML algorithm used here is RF.	138
7.8	Statistical variability in localization radii for different state variables of the Lorenz-96 model (7.9). The total 1000 assimilation cycles include both training and testing phases. The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$	138
7.9	Evolution of localization radii in both time and space for the Lorenz-96 model (7.9). The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$	139

7.10	Data assimilation with the QG model (7.10). EnKF is applied with a GC covariance localization function. The (<i>log</i>) RMSE is shown on the vertical axis, and the time (assimilation cycles) is shown on the horizontal axis. EnKF results with adaptive covariance localization are shown for different choices of the weighting factors w_1, w_2 . The localization is adaptive in time, and is compared to results with a fixed localization radius. The training phase consists of 80 assimilation cycles, and testing phase follows with 20 assimilation cycles. The time and space adaptive radii results are not as good as those obtained with the fixed, hand-tuned radius. The ML algorithm used here is RF.	140
7.11	Statistical variability in the localization radius for each state variable of QG model (7.10). The total assimilation cycles are 100 including both training and testing phases. The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$	140
7.12	Variability of localization radius in both time and space for QG model (7.10). The pool of radii is $\{3, 4\}$ The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$	141
7.13	Data assimilation with the Lorenz-96 model (7.9), and QG model (7.10). Average RMSE resulting from EnKF over test phase, of the learning algorithm, is obtained by using both RF and GB. Results with adaptive covariance localization are shown for different choices of the weighting factors w_1, w_2 . The dynamical model and adaptivity type, e.g. time vs space-time, are shown under each panel. The space-time adaptivity in QG model uses limited pool of radii with fixed weight combination as discussed in Figure 7.10.	142
8.1	Initial conditions, the analysis and the WRF forecast for the simulation time 12PM on 5/1/2017. Shown in the plots are the accumulated precipitation in millimeter unit.	159
8.2	Shown in the plots are contours of observable discrepancies which are the differences in the accumulated precipitation results of WRF forecast against the analysis data $\Delta_{t=12PM}$ on 5/1/2017 for two different physics combinations. The observation operator extracts the precipitation solution from the WRF state vector.	160

8.3	WRF prediction and NCEP analysis at 6PM on 5/1/2017. Zoom-in panels show the predictions over Virginia.	163
8.4	Discrepancy between WRF forecasts and the NCEP analysis over Virginia at 6PM on 5/1/2017. The forecast correction clearly improves the model results.	164
8.5	Frequency of change in the physics with respect to change in the input data from $\Delta_{t=12PM}^{\text{test}}$ to $\Delta_{t=12PM}^{\text{test}}/2$. Each data set contains 50 data points, and we report here the number of changes of each package.	168

List of Tables

4.1	RMSE between corrected model solution and real data	51
5.1	Average and variance of error in predictions of MP-LROM models (5.19) constructed via ANN and GP using errors $\varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ in training data for different sample sizes.	79
5.2	Average and variance of error in predictions of MP-LROM models (5.19) constructed via ANN and GP using logarithms of errors $\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ in training data for different sample sizes.	80
5.3	MP-LROM statistical results over five-fold cross-validation.	80
5.4	Total computational cost for building the ANN models for MP-LROM, ROMES and MFC methods. This is the cost of generating one multivariate MP-LROM model and 120 univariate ROMES and MFC models.	82
5.5	Average error of all 120 E_{fold} s for three methods.	83
5.6	Average variance of all 120 VAR_{fold} s for three methods	83
5.7	Total computational cost for building the artificial neural network (ANN) models for MP-LROM, ROMES and MFC methods. This is the cost of generating one multivariate MP-LROM model, one univariate ROMES model and one univariate MFC models.	84
5.8	Average and variance of errors in prediction of reduced basis dimension using MP-LROM models for different sample sizes	86
5.9	POD basis dimension discrepancies between the MP-LROM predictions and true values over five-fold cross-validation. The errors variance is also computed.	86

7.1	CPU-time of the training time of the two ML algorithms, RF and GB for both time adaptivity and space-time adaptivity approaches. Data assimilation experiment carried out using Lorenz model (7.9).	142
8.1	The minimum, average, and maximum RMSE between the predicted $\hat{\Delta}_{t=6PM}$ and the true $\Delta_{t=6}$ over 252 physics combinations.	162
8.2	The minimum and average of $\Delta_{t=6PM}$ for the original WRF forecast vs the improved forecast	163
8.3	Difference between predicted discrepancy norm $\ \hat{\Delta}_{t=6PM}\ _2$ and the reference discrepancy norm $\ \Delta_{t=6PM}\ _2$. The <i>RMS E</i> is taken over all test cases.	165
8.4	The RMSE between estimated discrepancy using predicted physical combinations $\hat{\Delta}_{t=12PM}^{\text{test}}$ and the reference discrepancy $\Delta_{t=12PM}^{\text{test}}$	167

Chapter 1

Introduction

1.1 Scientific context

Scientific computing involves very large-scale simulations of the real world to describe, predict, and understand the behavioral patterns of complex physical phenomena such as the weather, ocean dynamics, oil reservoirs, earthquakes, and volcanoes. Every simulation requires two fundamental ingredients, firstly a mathematical theory that describes the physical system under investigation, and secondly, a computer model that implements this theory and relates a set of input parameters to the observed data. For example, many physical phenomena are described mathematically by partial differential equations, and, after applying suitable discretization schemes, are simulated on a computer. PDE-based models such as fluid flow simulations seek to approximate the behavior of a real system based on the physical equations that govern the evolution of a natural system. The model approximation of reality, however, is imperfect because it is subject to uncertainties coming from different sources: uncertainty in model parameter values, uncertainty in input data and uncertainty in the structure of the model itself. For example, numerical simulations using the computer model of the system inherently introduce truncation and round-off errors. Moreover, our knowledge of the physical phenomenon are not complete and some times the equations describing the physical phenomenon are not fully representing all the dynamics of the nature. Capturing, representing and formulating the source of uncertainty in the models and later reducing the uncertainty is very important in design optimization, code verification and validation (V&V) and policy making. In particular, when a simulation should be validated, it is crucial to quantify the expected uncertainty in the outputs. Usually, an ensemble of experiments with controlled parameters are required to estimate

the uncertainty for an experiment.

The process of uncertainty quantification includes identification, characterization, propagation, analysis and finally reduction of the uncertainties. Uncertainty quantification embraces various mathematical and statistical approaches such as direct evaluation for linearly parametric models, sampling methods such as Monte Carlo, quasi-Monte Carlo techniques [helton2003latin, lemieux2009monte], generalized polynomial chaos expansions (GPCE), perturbation methods [1, 2, 3], spectral representation [4, 5, 6], collocation techniques, Bayesian methods, Galerkin modeling, parameter estimation, and forward propagation of uncertainty with differential equations. Uncertainty reduction is also the process of improving the model forecast by reducing the overall uncertainties in the model. Hence, more accurate model forecasts require capturing and reducing the uncertainty in the input data as well as in the models.

While uncertainty propagation techniques can measure the impact of uncertain parameters on some quantities of interest, they often become infeasible due to the large number of model realizations required. Similar difficulties are encountered when solving Bayesian inference problems since sampling from the posterior distribution is required. Aiming to tackle these challenges, the current research explores and proposes several novel approaches that try to improve the model performance by error modeling, identifying the source of uncertainty, analyzing the uncertainty and finally improving the model predictions by reducing the uncertainty. We consider different applications and models to study various aspect of error modeling and uncertainty analysis in computational models from stochastic simulation of chemical reactions, physical systems described in Partial Differential Equations (PDE), Reduced Order Models (ROM), Data Assimilation (DA) to Numerical Weather Prediction (NWP) models in order to study several forms of uncertainty analysis and reduction.

1.2 Research objectives

The overall goal of this research is to develop approaches for uncertainty analysis and error modeling in large-scale simulations. We construct probabilistic and statistical approaches, as well as learning based algorithms, to quantify and reduce the impact of uncertainties and errors on the solution of computational science problems. Specifically, this research considers four different problems, detailed below.

Stochastic simulations. Biological systems are frequently modeled as chemical interaction networks. In the context of large simulation of chemical kinetics, the stochastic simulation approach provides very comprehensive description of biological systems evolution and behavior. The complete solution to these problems is the Chemical Master Equation (CME) and the Stochastic Simulation Algorithm (SSA) [7] that follows a Monte Carlo approach to sample from the CME. However, the SSA is very slow and quite impractical for very large scale biological systems. The explicit tau-leap method [8] provides pretty good approximation for the SSA algorithm but the uncertainty of the approximation increases as we increase the time step for the sake of speed in very large simulation models. The tau-leap method suffers from both stability and accuracy when increasing the time step. We propose a approaches to reduce the error and improve the performance of tau-leap methods in stochastic simulation of chemical reactions.

Modeling structural uncertainty in physics based models. The second class of models are large simulation models that try to approximate the true physical phenomenon. These models are usually presented by a set of PDEs that describe the physics governing the behavior of the natural phenomenon. But there is a discrepancy between the output of model and reality. This uncertainty can have different sources, such as uncertainty in measurements, input data, model parameters, discretization error, initial and boundary conditions error. Finally, the uncertainty might be due to lack of knowledge about all the physics of the phenomenon and consequently not including it in the model [9, 10]. This form of uncertainty which represents the misfit as a missing dynamics is called structural uncertainty. We seek to understand structural uncertainty by studying the observable errors, i.e., the discrepancies between the model solutions and measurements of the physical system [11]. Furthermore, this information is used to improve model output.

Building reduced order models with controlled errors and uncertainty. The need for efficient computational complexity and speed motivated building the surrogate models such as response surfaces, low resolution, and reduced-order models [12]. Reduced-order models are usually constructed to approximate the high-fidelity models by ignoring some of the physical aspects or decreasing the spatial resolution [13, 14, 15, 16, 17]. Traditional construction of ROM models in a parametric setting is usually achieved by generating a global basis for the whole parametric space [18, 19]. However, this strategy requires generating very large dimensional bases which leads to a modest decrease of the computational load and consequently slow reduced-order models. Local approaches in contrast, have been designed for parametric or time domains which generate local bases for both the state variables and non-linear terms [20, 21, 22, 23]. When designing a local reduced-order

model, a basis with specific dimension and certain parametric value should be picked. The parameter used in the ROM can have different value than the parameter of basis. As we take part from the basis parameter value and decrease the basis dimension (reducing the computational complexity), the accuracy decreases and the uncertainty increases. In this study, we predict the uncertainty in local parametric reduced-order models, and construct them with specific basis dimensions and parametric values.

Reducing probabilistic errors in data assimilation. Data assimilation is a key factor of many forecasting problems. Specifically in meteorology, DA is used to obtain analysis of the state of the atmosphere that can be used as initial condition in weather forecast problem. The most common approach for solving DA problems is the Kalman Filter (KF) family of method [24, 25]. The ensemble Kalman Filter (EnKF) [26, 27, 28] is one of the common approaches that can handle the non-linear systems efficiently. EnKF works by operating on an ensemble of the system state which are statistical representative of the system state. However, the size of the ensemble should be sufficient enough to be statistically representative of the system state. If the sample size is too small it fails to reflect the true state well which is called under-sampling error [28]. Under-sampling error leads to the issues of malfunctioning the DA algorithm [29, 30, 31, 32] We propose an adaptive localization approach to reduce the uncertainty coming from sampling error in ensemble based DA.

Understanding uncertainty due to the interaction of physical processes in numerical weather prediction. Numerical weather prediction (NWP) models include multiple physical processes that are essential to accurately representing the atmospheric dynamics. Accurately predicting the precipitation is one of the big challenges in weather predictions. Interaction of different physical processes causes uncertainty in model prediction. There is no physical combination that can work best for all times, all locations and different phenomenon. We seek to analyze the uncertainty due to interaction of physical processes and improve model predictions by reducing the uncertainty in model output.

1.3 Research accomplishments

We next summarize the main contributions of this dissertation.

1.3.1 Approximate exponential algorithms to solve the chemical master equation

We develop new simulation algorithms for stochastic chemical kinetics that exploit the linearity of the chemical master equation and its matrix exponential exact solution. These algorithms make use of various approximations of the matrix exponential to evolve probability densities in time. A sampling of the approximate solutions of the chemical master equation is used to derive accelerated stochastic simulation algorithms. Numerical experiments compare the new methods with the established stochastic simulation algorithm and the tau-leaping method.

1.3.2 Metropolis-Hastings sampling approaches to solve stochastic chemical kinetics

This work proposes the use of Metropolis-Hastings algorithm for the stochastic simulation of chemical reactions. The proposed method uses SSA distribution which is a standard method for solving well-stirred chemically reacting systems as a desired distribution. A new numerical solvers based on exponential form of exact and approximate solutions of CME (Chemical Master Equation) is employed for obtaining target and proposal distributions in Metropolis-Hastings algorithm to accelerate the accuracy of the tau-leap method. Samples generated by this technique have the same distribution as SSA and the histogram of samples show it's convergence to SSA.

1.3.3 Structural uncertainty analysis in physical models

We seek to understand structural uncertainty by studying the observable errors, i.e., the discrepancies between the model solutions and measurements of the physical system. The dynamics of these errors is modeled using a state-space approach, which enables to identify the source of uncertainty and to recognize the missing dynamics inside model. Two different strategies to model the error dynamics are developed: the global approach and the local approach. The global approach considers the whole state space of the model to analyze the uncertainty, while in the local approach the source of uncertainty is local dependencies between model state variables. The proposed methodology is applied to two test problems, Lorenz-96 and a stratospheric chemistry model.

1.3.4 Error prediction of local parametric reduced order models

We introduce multivariate input-output models to predict the errors and bases dimensions of local parametric Proper Orthogonal Decomposition reduced-order models. We refer to these mappings as the Multivariate Predictions of Local Reduced-Order Model characteristics (MP-LROM) models. We employ Gaussian Processes and Artificial Neural Networks to construct approximations of these multivariate mappings. Numerical results with a viscous Burgers model illustrate the performance and potential of the machine learning based regression MP-LROM models to approximate the characteristics of parametric local reduced-order models.

Next we show two applications of MP-LROM. First, the error model is used in conjunction with a greedy sampling algorithm to generate decompositions of one dimensional parametric domains with overlapping regions, such that the associated local reduced-order models meet the prescribed accuracy requirements. The parameter domain decomposition creates a database of the available local bases, local reduced-order, and high-fidelity models, and identifies the most accurate solutions for an arbitrary parametric configuration. Next, this database is used to enhance the accuracy of the reduced-order models using different interpolation and concatenation techniques.

1.3.5 Adaptive localization in data assimilation

We develop an adaptive covariance localization mechanism for the EnKF family of methods following a machine learning approach. Two different methodologies are introduced, adaptive localization in time, and adaptive localization in space and time. In time adaptivity, the localization radius changes adaptively based on characteristics of consecutive assimilation cycles. In space-time adaptivity, the localization radius changes both in time and for each state of the model independently based on the state value or correlation between state values. The adaptive localization model, can be trained off-line using historical data and the well-trained model can be used to estimate the proper values of localization radius for future assimilation cycles. Numerical experiments are carried out to test the proposed methods, using the Lorenz-96 model, and a Quasi-Geostrophic (QG) model showing better results than the traditional way of localization.

1.3.6 Uncertainty analysis in numerical weather prediction models

We study the uncertainty in numerical weather prediction (NWP) models due to the interaction of physical processes. The discrepancy between NWP output and reality gives valuable information about the physics and dynamics in the model. This study tries to quantify and further reduce the uncertainty in NWP models using information hidden in discrepancy between NWP prediction and reality. The future model error is predicted using machine learning and further used to improve the model prediction. Furthermore, we analyzed what physical processes cause more error in model prediction that needs to be changed or improved. The numerical experiments are carried out with the Weather Research Forecast (WRF) model and the proposed approach is used to improve the model prediction results.

1.4 Dissertation layout

The remaining parts of this dissertation, detailing the research accomplishments described in the previous section, are organized as follows:

- Chapter 2 develops several numerical algorithms that approximately solve the chemical master equation.
- Chapter 3 develops a statistical approach that uses the exact and approximate exponential solutions of the chemical master equation to increase the accuracy of explicit tau-leaping solver.
- Chapter 4 analyzes the structural uncertainty in physical models and introduces a statistical methodology to reduce this form of uncertainty.
- Chapter 5 develops a machine learning based approach called MP-LROM to predict the errors of reduced-order models.
- Chapter 6 describes the application of MP-LROM model to guide the construction of parametric reduced order surrogates, and to increase their efficiency and accuracy.
- Chapter 7 introduces an adaptive localization approach based on machine learning technique for data assimilation applications .
- Chapter 8 proposes machine learning based approaches to reduce the uncertainty in a weather forecasting model due to the interaction of different physical packages.

- Chapter 9 draws the conclusions and points to future research directions.

Chapter 2

Approximate Exponential Algorithms to Solve the Chemical Master Equation

2.1 Introduction

In many biological systems the small number of participating molecules make the chemical reactions inherently stochastic. The system state is described by probability densities of the numbers of molecules of different species. The evolution of probabilities in time is described by the chemical master equation (CME) [7]. Gillespie proposed the Stochastic Simulation Algorithm (SSA), a Monte Carlo approach that samples from CME [7]. SSA became the standard method for solving well-stirred chemically reacting systems. However, SSA simulates one reaction and is inefficient for most realistic problems. This motivated the quest for approximate sampling techniques to enhance the efficiency.

The first approximate acceleration technique is the tau-leaping method [8] which is able to simulate multiple chemical reactions appearing in a pre-selected time step of length τ . The tau-leap method is accurate if τ is small enough to satisfy the leap condition, meaning that propensity functions remain nearly constant in a time step. The number of firing reactions in a time step is approximated by a Poisson random variable [33]. Explicit tau-leaping method is numerically unstable for stiff systems [34]. Stiffness systems have well-separated “fast” and “slow” time scales present, and the “fast modes” are stable. The implicit tau-leap method [35] overcomes the stability issue but it has a damping effect on the computed variances. More accurate variations of the implicit tau-leap method have been proposed to alleviate the damping [36, 8, 37, 38, 39, 40]. Simulation efficiency has

been increased via parallelization [41].

Direct solutions of the CME are computationally important specially in order to estimate moments of the distributions of the chemical species [42]. Various approaches to solve the CME are discussed in [43].

Sandu re-drives the SSA algorithm from CME in [44]. The analysis reveals the hidden approximations made by the SSA and the tau-leap methods. The approach explains the explicit tau-leap method as an exact sampling procedure from an approximate solution of the CME. The numerical solver is obtained by first discretizing the CME in time, advancing the probability density by one time step, and then sampling the new probability density from the approximation.

This chapter extends the study [44] and proposes new approximations to the CME solution based on various approximations of matrix exponentials. Different approximation techniques of the matrix exponential lead to various algorithms that have their own strengths and weaknesses with regard to accuracy, stability and computational complexity. Numerical experiments are performed with two different chemical systems to assess the accuracy and stability of each of the algorithms.

The chapter is organized as follows. Section 2.2 reviews the stochastic simulation of chemical kinetics. Section 2.3 developed the new approximation methods. Numerical experiments to illustrate the proposed schemes are carried out in Section 2.4. Conclusions are drawn in Section 2.5.

2.2 Simulation of stochastic chemical kinetics

Consider a chemical system in a constant volume container. The system is well-stirred and in thermal equilibrium at some constant temperature. There are N different chemical species S^1, \dots, S^N . Let $X^i(t)$ denote the number of molecules of species S_i at time t . The state vector $x(t) = [X^1(t), \dots, X^N(t)]$ defines the numbers of molecules of each species present at time t . The chemical network consists of M reaction channels R_1, \dots, R_M . Each individual reaction destroys a number of molecules of reactant species, and produces a number of molecules of the products. Let ν_j^i be the change in the number of S^i molecules caused by a single reaction R_j . The state change vector $\nu_j = [\nu_j^1, \dots, \nu_j^N]$ describes the change in the entire state following R_j .

A propensity function $a_j(x)$ is associated with each reaction channel R_j . The probability that one R_j reaction will occur in the next infinitesimal time interval $[t, t+dt)$ is $a_j(x(t)) \cdot dt$.

The purpose of a stochastic chemical simulation is to trace the time evolution of the system state $x(t)$ given that at the initial time \bar{t} the system is in the initial state $x(\bar{t})$.

2.2.1 Chemical Master Equation

The Chemical Master Equation (CME) [7] has complete information about time evolution of probability of system's state

$$\frac{\partial \mathcal{P}(x, t)}{\partial t} = \sum_{r=1}^M a_r(x - v_r) \mathcal{P}(x - v_r, t) - a_0(x) \mathcal{P}(x, t). \quad (2.1)$$

Let Q^i be the total possible number of molecules of species S^i . The total number of all possible states of the system is:

$$Q = \prod_{i=1}^N (Q^i + 1).$$

We denote by $I(x)$ the state-space index of state $x = [X^1, \dots, X^N]$

$$I(x) = (Q^{N-1} + 1) \dots (Q^1 + 1) \cdot X^N + \dots \\ + (Q^2 + 1) (Q^1 + 1) \cdot X^3 + (Q^1 + 1) \cdot X^2 + X^1 + 1$$

One firing of reaction R_r changes the state from x to $\bar{x} = x - v_r$. The corresponding change in state space index is:

$$I(x) - I(x - v_r) = d_r, \\ d_r = (Q^{N-1} + 1) \dots (Q^1 + 1) \cdot v_r^N + \dots \\ + (Q^2 + 1) (Q^1 + 1) \cdot v_r^3 + (Q^1 + 1) \cdot v_r^2 + v_r^1.$$

The discrete solutions of the CME (2.1) are vectors in the discrete state space, $\mathcal{P}(t) \in \mathbb{R}^Q$. Consider the diagonal matrix $A_0 \in \mathbb{R}^{Q \times Q}$ and the Toeplitz matrices $A_1, \dots, A_M \in \mathbb{R}^{Q \times Q}$ [44]

$$(A_0)_{i,j} = \begin{cases} -a_0(x_j) & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases}, \quad (A_r)_{i,j} = \begin{cases} a_r(x_j) & \text{if } i - j = d_r, \\ 0 & \text{if } i - j \neq d_r, \end{cases}$$

as well as their sum $A \in \mathbb{R}^{Q \times Q}$ with entries

$$A = A_0 + A_1 + \dots + A_M, \quad A_{i,j} = \begin{cases} -a_0(x_j) & \text{if } i = j, \\ a_r(x_j) & \text{if } i - j = d_r, r = 1, \dots, M, \\ 0 & \text{otherwise,} \end{cases} \quad (2.2)$$

where x_j denotes the unique state with state space index $j = \mathcal{I}(x_j)$. In fact matrix A is a square ($Q \times Q$) matrix which contains all the propensity values for each possible value of all species or let's say all possible states of reaction system. All possible states for a reaction system consists of N species where each specie has at most Q^i $i = 1, 2, \dots, N$ value.

The CME (2.1) is a linear ODE on the discrete state space

$$\mathcal{P}' = A \cdot \mathcal{P}, \quad \mathcal{P}(\bar{t}) = \delta_{\mathcal{I}(\bar{x})}, \quad t \geq \bar{t}, \quad (2.3)$$

where the system is initially in the known state $x(0) = \bar{x}$ and therefore the initial probability distribution vector $\mathcal{P}(0) \in \mathbb{R}^Q$ is equal to one at $\mathcal{I}(\bar{x})$ and is zero everywhere else. The exact solution of the linear ODE (2.3) is follows:

$$\mathcal{P}(\bar{t} + T) = \exp(T A) \cdot \mathcal{P}(\bar{t}) = \exp\left(T \sum_{r=0}^M A_r\right) \cdot \mathcal{P}(\bar{t}). \quad (2.4)$$

2.2.2 Approximation to Chemical Master Equation

Although the CME (2.1) fully describes the evolution of probabilities it is difficult to solve in practice due to large state space. Sandu [44] considers the following approximation of the CME:

$$\frac{\partial \mathcal{P}(x, t)}{\partial t} = \sum_{r=1}^M a_r(\bar{x}) \mathcal{P}(x - v_r, t) - a_0(\bar{x}) \mathcal{P}(x, t) \quad (2.5)$$

where the arguments of all propensity functions have been changed from x or $x - v_j$ to \bar{x} . In order to obtain an exponential solution to (2.5) in probability space we consider the diagonal matrix $\bar{A}_0 \in \mathbb{R}^{Q \times Q}$ and the Toeplitz matrices $\bar{A}_1, \dots, \bar{A}_M \in \mathbb{R}^{Q \times Q}$ [44]. \bar{A}_r matrices are square ($Q \times Q$) matrices are built upon the current state of system in reaction system which is against A_r matrices that contain all possible states of reaction system.

$$(\bar{A}_0)_{i,j} = \begin{cases} -a_0(\bar{x}) & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases}, \quad (\bar{A}_r)_{i,j} = \begin{cases} a_r(\bar{x}) & \text{if } i - j = d_r, \\ 0 & \text{if } i - j \neq d_r, \end{cases} \quad (2.6)$$

together with their sum $\bar{A} = \bar{A}_0 + \dots + \bar{A}_M$. The approximate CME (2.5) can be written as the linear ODE

$$\mathcal{P}' = \bar{A} \cdot \mathcal{P}, \quad \mathcal{P}(\bar{t}) = \delta_{\mathcal{I}(\bar{x})}, \quad t \geq \bar{t},$$

and has an exact solution

$$\mathcal{P}(\bar{t} + T) = \exp(T \bar{A}) \cdot \mathcal{P}(\bar{t}) = \exp\left(T \sum_{r=0}^M \bar{A}_r\right) \cdot \mathcal{P}(\bar{t}). \quad (2.7)$$

2.2.3 Tau-leaping method

In tau-leap method the number of times a reaction fires is a random variable from a Poisson distribution with parameter $a_r(\bar{x})\tau$. Since each reaction fires independently, the probability that each reaction R_r fires exactly k_r times, $r = 1, 2, \dots, M$, is the product of M Poisson probabilities.

$$\mathcal{P}(K_1 = k_1, \dots, K_M = k_M) = \prod_{r=1}^M e^{-a_r(\bar{x})\tau} \cdot \frac{(a_r(\bar{x})\tau)^{k_r}}{K_r!} = e^{-a_0(\bar{x})\tau} \cdot \prod_{r=1}^M \frac{(a_r(\bar{x})\tau)^{k_r}}{K_r!}$$

The state vector after these reactions changes as follows:

$$X(\bar{t} + \tau) = \bar{x} + \sum_{r=1}^M K_r \nu_r. \quad (2.8)$$

The probability to go from state \bar{x} at \bar{t} to state x at $\bar{t} + \tau$, $\mathcal{P}(X(\bar{t} + \tau) = x)$, is the sum of all possible firing reactions which is:

$$\mathcal{P}(X, \bar{t} + \tau) = e^{-a_0(\bar{x})T} \cdot \sum_{k \in \mathcal{K}(x-\xi)} \prod_{r=1}^M \frac{(a_r(\bar{x}T))^{k_r}}{K_r!}$$

Equation (2.7) can be approximated by product of each matrix exponential:

$$\mathcal{P}(\bar{t} + T) = \exp(T\bar{A}_0) \cdot \exp(T\bar{A}_1) \cdots \exp(T\bar{A}_r) \cdot \mathcal{P}(\bar{t}). \quad (2.9)$$

It has been shown in [44] that the probability given by the tau-leaping method is exactly the probability evolved by the approximate solution (2.9).

2.3 Approximations to the exponential solution

2.3.1 Strang splitting

In order to improve the approximation of the matrix exponential in (2.9) we consider the symmetric Strang splitting [45]. For $T = n\tau$ Strang splitting applied to an interval of length τ leads to the approximation

$$\mathcal{P}(\bar{t} + i\tau) = e^{\tau/2\bar{A}_r} \cdots e^{\tau/2\bar{A}_1} e^{\tau\bar{A}_0} \cdot e^{\tau/2\bar{A}_1} \cdots e^{\tau/2\bar{A}_r} \cdot \mathcal{P}(\bar{t} + (i-1)\tau) \quad (2.10)$$

where the matrices \bar{A}_r are defined in (2.6).

2.3.2 Column based splitting

In column based splitting the matrix A (2.2) is decomposed in a sum of columns

$$A = \sum_{j=1}^Q A_j, \quad A_j = c_j e_j^T.$$

Each matrix A_j has the same j -th column as the matrix A , and is zero everywhere else. Here c_j is the j -th column of matrix A and e_j is the canonical vector which is zero everywhere except the j -th component. The exponential of τA_j is:

$$e^{\tau A_j} = \sum_{k \geq 0} \frac{\tau^k (A_j)^k}{k!}. \quad (2.11)$$

Since $e_j^T c_j$ is equal to the j -th diagonal entry of matrix A :

$$e_j^T c_j = -a_0(x_j)$$

the matrix power A_j^k reads

$$A_j^k = c_j e_j^T c_j e_j^T \cdots c_j e_j^T = (-a_0(x_j))^{k-1} c_j e_j^T = (-a_0(x_j))^{k-1} A_j.$$

Consequently the matrix exponential (2.11) becomes

$$e^{\tau A_j} = I + \sum_{k \geq 1} \frac{(-\tau a_0(x_j))^{k-1}}{k!} (\tau A_j) = I + S_j \tau A_j, \quad S_j = \sum_{k \geq 1} \frac{(-\tau a_0(x_j))^{k-1}}{k!}.$$

We have

$$e^{\tau A} = e^{\tau \sum_{j=1}^Q A_j} \approx \prod_{j=1}^Q e^{\tau A_j} \approx \prod_{j=1}^Q (I + S_j \tau A_j)$$

and the approximation to the CME solution reads

$$\mathcal{P}(\bar{i} + i\tau) \approx \prod_{j=1}^Q (I + S_j \tau A_j) \cdot P(\bar{i} + (i-1)\tau).$$

2.3.3 Accelerated tau-leaping

In this approximation method we build the matrices

$$(B_r)_{i,j} = \begin{cases} -a_r(x_j) & \text{if } i = j, \\ a_r(x_j) & \text{if } i - j = d_r, \\ 0 & \text{otherwise} \end{cases}$$

where $a_r(x)$ are the propensity functions. The matrix A in (2.2) can be written as

$$A = \sum_{r=1}^M B_r.$$

The solution of the linear CME (2.4) can be approximated by

$$\mathcal{P}(\bar{t} + \tau) = e^{\tau A} \cdot \mathcal{P}(\bar{t}) \approx e^{\tau B_1} e^{\tau B_2} \dots e^{\tau B_M} \cdot P(\bar{t}). \quad (2.12)$$

Note that the evolution of state probability by $e^{\tau B_j} \cdot P(\bar{t})$ describes the change in probability when only reaction j fires in the time interval τ . The corresponding evolution of the number of molecules that samples the evolved probability is

$$x(\bar{t} + \tau) = x(\bar{t}) + V_j K(a_j(x(\bar{t}))\tau).$$

where $K(a_j(x(\bar{t}))\tau)$ is a random number drawn from a Poisson distribution with parameter $a_j(x(\bar{t}))\tau$, and V_j is the j -th column of stoichiometry matrix.

The approximate solution (2.12) accounts for the change in probability due to a sequential firing of reactions $M, M - 1, \dots, 1$. Sampling from the resulting probability density can be done by changing the system state sequentially consistent with the firing of each reaction. This leads to algorithm (2.13). The accelerated tau-leap method (2.13) provides accurate and stable results especially for large time steps since the change in state is accomplished through a sequential firing of reactions. From the complexity point of view, the number of propensity function calculations and Poisson random numbers generated are the same as for traditional tau-leap. We note that in traditional tau-leap a vector of Poisson random variables is generated at once, which is possibly more efficient than generating the random numbers one at a time.

$$\begin{aligned} \hat{X}_M &= x(\bar{t}) \\ \text{for } i &= M, M - 1, \dots, 1 \\ \hat{X}_{i-1} &= \hat{X}_i + V_i K(a_i(\hat{X}_i)\tau) \\ x(\bar{t} + \tau) &= \hat{X}_0. \end{aligned} \quad (2.13)$$

Moreover, (2.12) can also be written as:

$$\begin{aligned} \mathcal{P}(\bar{t} + \tau) &\approx e^{\tau B_1} e^{\tau B_2} \dots e^{\tau B_M} \cdot P(\bar{t}) \\ &\approx \left(e^{\tau B_1} e^{\tau B_2} \dots e^{\tau B_{\frac{M}{2}-1}} \right) \cdot \\ &\quad \left(e^{\tau B_{\frac{M}{2}}} e^{\tau B_{\frac{M}{2}+1}} \dots e^{\tau B_M} \cdot P(\bar{t}) \right). \end{aligned} \quad (2.14)$$

Then, (2.13) can be written as:

$$\begin{aligned} \hat{X}_M &= x(\bar{t}) \\ \text{for } i &= M, M-1, \dots, \frac{M}{2} \\ \hat{X}_{i-1} &= \hat{X}_i + V_i K(a(\hat{X}_i) \tau) \\ \text{for } i &= \frac{M}{2} - 1, \dots, 1 \\ \hat{X}_{i-1} &= \hat{X}_i + V_i K(a(\hat{X}_{\frac{M}{2}-1}) \tau) \\ x(\bar{t} + \tau) &= \hat{X}_0. \end{aligned} \quad (2.15)$$

2.3.4 Symmetric accelerated tau-leaping

A more accurate version of accelerated tau-leaping can be constructed by using symmetric Strang splitting (2.10) to approximate the matrix exponential in (2.12). Following the procedure used to derive (2.13) leads to the following sampling algorithm:

$$\begin{aligned} \hat{X}_M &= x(\bar{t}) \\ \text{for } i &= M, M-1, \dots, 1 \\ \hat{X}_{i-1} &= \hat{X}_i + V_i K(a_i(\hat{X}_i) \tau/2) \\ \text{for } i &= 1, 2, \dots, M \\ \hat{X}_i &= \hat{X}_i + V_{i-1} K(a_i(\hat{X}_{i-1}) \tau/2) \\ x(\bar{t} + \tau) &= \hat{X}_M. \end{aligned} \quad (2.16)$$

The symmetric accelerated tau-leap algorithm (2.16) is twice as expensive as the accelerated tau-leap (2.15) for the same value of τ since it computes the propensity functions and generates Poisson random variables two times per step.

2.4 Numerical experiments

The above approximation techniques are used to solve two test systems, reversible isomer and the Schlogl reactions [34]. The experimental results are presented in following sections.

2.4.1 Isomer reaction

The reversible isomer reaction system is [34]



The stoichiometry matrix and the propensity functions are:

$$V = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \begin{aligned} a_1(x) &= c_1 x_1, \\ a_2(x) &= c_2 x_2. \end{aligned}$$

The reaction rate values are $c_1 = 10$, $c_2 = 10$ (units), the time interval is $[0, T]$ with $T = 10$ (time units), initial conditions are $x_1(0) = 40$, $x_2(0) = 40$ molecules, and maximum values of species are $Q^1 = 80$ and $Q^2 = 80$ molecules.

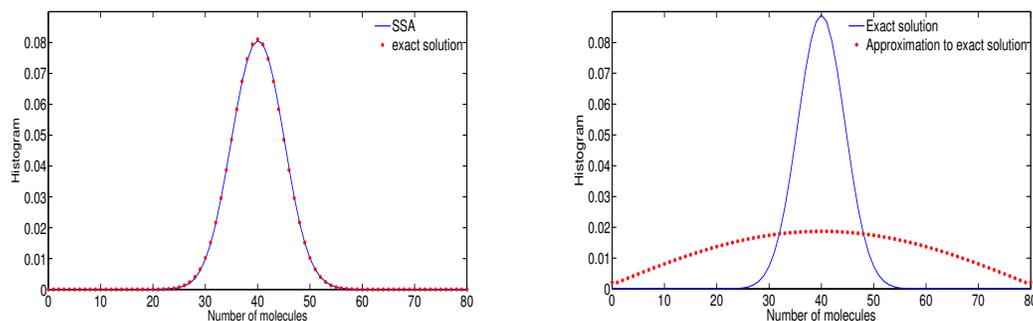
The exact exponential solution of CME obtained from (2.4) is a joint probability distribution vector for the two species at final time. Figure 2.1(a) shows that the histogram of 10,000 SSA solutions is very close to the exact exponential solution. The approximate solution using the sum of exponentials (2.7) is illustrated in Figure 2.1(b). This approximation is not very accurate since it uses only the current state of the system. Other approximation methods based on the product of exponentials (2.9), Strang splitting, (2.10) and column based splitting are not very accurate, and consequently we choose not to report their results. The reason of poor approximation of product of exponentials and Strang splitting methods is due to the error propagation that occurs during successive matrix vector multiplications.

The results reported in Figure 2.2 indicate that for small time steps τ the accelerated tau-leap (2.13) solution is very close to the results provided by traditional explicit tau-leap. Symmetric accelerated tau-leap method (2.16) yields even better results, as shown in Figure 2.3. For small time steps the traditional and symmetric accelerated methods give similar results, however, for large time steps, the results of the symmetric accelerated method is considerably more stable.

To assess accuracy we measure the numerical error as the the difference between the PDF of each algorithm and the PDF of SSA.

$$error = \frac{\sum_{i=0}^{Q(1)} |SSA_{molecules}(i) - NewAlgorithm_{molecules}(i)|}{Q(1)}$$

Figure 2.4 plots the errors given by different algorithms versus computational time. Each point on the curves corresponds to a different time step τ ; specifically, 10,000 simulations



(a) 10,000 SSA runs versus the exact solution (2.4) (b) Exact solution (2.4) versus the approximation to exact solution using sum of exponentials (2.7)

Figure 2.1: Histograms of the isomer system (2.17) results at the final time $T=10$.

with that value of τ are carried out and the samples are used to obtain a histogram that approximates the PDF. For larger time steps τ (smaller CPU times) the errors of symmetric accelerated tau-leap and accelerated tau-leap are smaller than the error of traditional tau-leap. The two new algorithms are more effective for lower accuracy computations. For small time steps τ (target CPU times) the traditional tau-leap is the most efficient. The CPU time for the symmetric accelerated tau-leap the largest among the three methods for a given time step since it requires computing propensity functions twice per step.

Absolute stability and stiffness

To assess the stability of the new algorithms we apply them to solve the isomer reaction [34]. It turns out that for this test problem the mean and variance of the solution obtained by accelerated tau-leap and by symmetric accelerated tau-leap are the same as for traditional tau-leap, since the only change in these algorithms is the sequential firing of reactions. The procedure outlined in [34] for obtaining the absolute stability is to fix τ and let the number of steps n tend to infinity. Having $X^* = X^{\text{theoretical}}(\infty) = X_1^{\text{theoretical}}(\infty)$ and $\lambda = c_1 + c_2$ in isomer reaction, the following asymptotic values of the mean and variance are obtained [34]

$$E[X(\infty)] = E(X^*), \quad \text{Var}[X(\infty)] = \frac{2}{2 - \lambda\tau} \text{Var}(X^*).$$

The mean value given by tau-leap method converges to the theoretical mean value, while the variance does not. The same conclusions hold for the accelerated tau-leap and symmetric accelerated tau-leap methods.

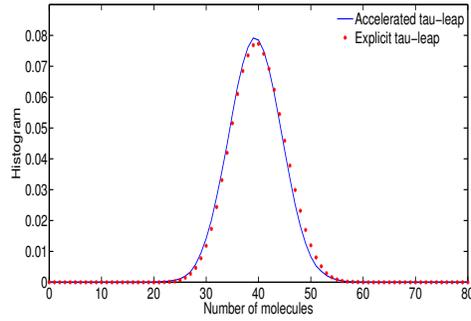


Figure 2.2: Isomer system (2.17) solutions provided by the traditional tau-leap (2.8) and by accelerated tau-leap (2.13) methods at the final time $T=10$ (units). A small time step of $\tau = 0.01$ (units) is used. The number of samples for both methods is 10,000.

2.4.2 Schlogl reaction

We next consider the Schlogl reaction system [34]



whose solution has a bi-stable distribution. Let N_1, N_2 be the numbers of molecules of species B_1 and B_2 , respectively. The reaction stoichiometry matrix and the propensity functions are:

$$\begin{aligned} V &= \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix} \\ a_1(x) &= \frac{c_1}{2} N_1 x(x-1), \\ a_2(x) &= \frac{c_2}{6} N_1 x(x-1)(x-2), \\ a_3(x) &= c_3 N_2, \\ a_4(x) &= c_4 x. \end{aligned}$$

The following parameter values (each in appropriate units) are used:

$$\begin{aligned} c_1 &= 3 \times 10^{-7}, & c_2 &= 10^{-4}, & c_3 &= 10^{-3}, \\ c_4 &= 3.5, & N_1 &= 1 \times 10^5, & N_2 &= 2 \times 10^5. \end{aligned}$$

with the final time $T = 4$ (units), the initial condition $x(0) = 250$ molecules, and the maximum values of species $Q^1 = 900$ molecules.

Figure 2.5(a) illustrates the result of exact exponential solution (2.4) versus SSA. Figure 2.5(b) reports the sum of exponentials (2.7) result which is not a very good approximation.

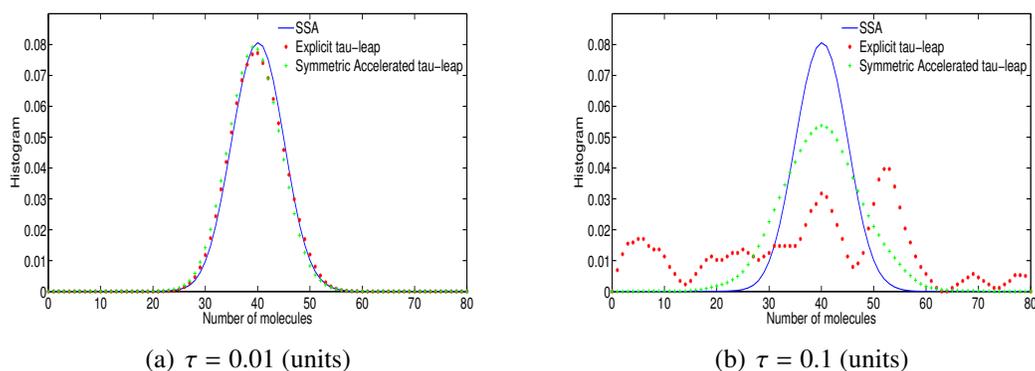


Figure 2.3: Histograms of isomer system (2.17) solutions obtained with SSA, traditional tau-leap (2.8), and symmetric accelerated tau-leap (2.16) methods at the final time $T=10$. The number of samples is 10,000 for all methods.

The product of exponentials (2.9) and Strang splitting (2.10) results are not reported here since they provide inaccurate approximations.

Figures 2.6(a) and 2.6(b) present the results obtained with the accelerated tau-leap and the symmetric tau-leap, respectively. For small time step the results are very accurate. For large step sizes the results become less accurate but continue to be more stable than tau-leap. Systems having a large number of reactions may be more affected by the low accuracy, and improvements such as the ones described in equations (2.14) and (2.15) may prove helpful.

2.5 Conclusions

This study proposes new numerical solvers for stochastic simulations of chemical kinetics. The proposed approach exploits the linearity of the CME and the exponential form of its exact solution. The matrix exponential appearing in the CME solution is approximated as a product of simpler matrix exponentials. This leads to an approximate (“numerical”) solution of the probability density evolved to a future time. The solution algorithms sample exactly this approximate probability density and provide extensions of the traditional tau-leap approach.

Different approximations of the matrix exponential lead to different numerical algorithms: Strang splitting, column splitting, accelerated tau-leap, and symmetric accelerated tau-

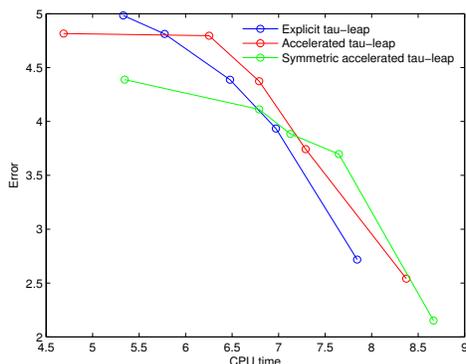
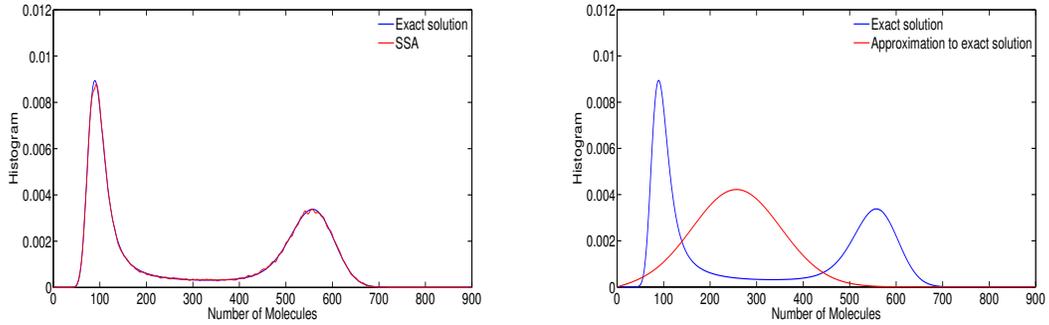


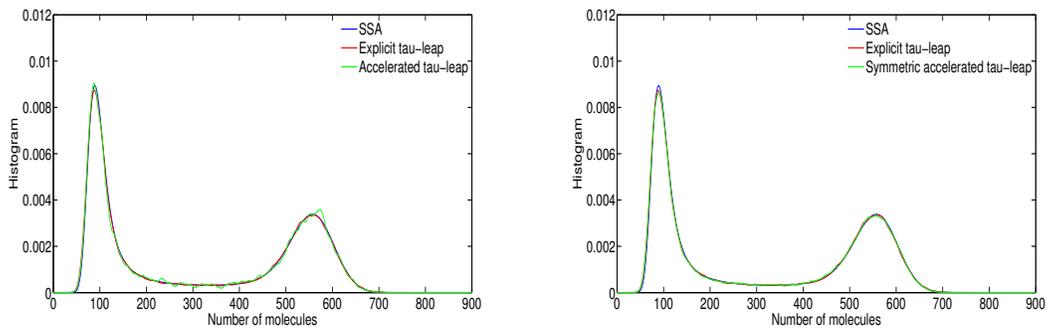
Figure 2.4: Plot of errors vs. CPU times for different algorithms. The points on the graphs correspond to the following five time steps: $\tau = 0.01$, $\tau = 0.05$, $\tau = 0.075$, $\tau = 0.1$, and $\tau = 0.15$ (units). The final time $T=10$ (units). Each point is obtained from 10,000 samples/runs.

leap. Numerical results illustrate that the new approximation methods are more stable than explicit tau-leap especially for large time steps, but are less accurate for some reaction systems. Despite this fact the class novel numerical solvers proposed herein is of interest since it is based on a totally different approach than the one used to derive classical schemes. Specifically, one first discretizes the chemical master equation, then draws samples from the resulting probability density. Current work by the authors focuses on improving the accuracy of the new family of approximation techniques.



(a) 10,000 SSA runs versus the exact solution (2.4) (b) Exact solution (2.4) versus the approximation to exact solution using sum of exponentials (2.7)

Figure 2.5: Histograms of Schlogl system (2.18) results at final time $T=4$ (units).



(a) Traditional tau-leap (2.8) and accelerated tau-leap (2.13) (b) Traditional tau-leap (2.8) and symmetric accelerated tau-leap (2.16)

Figure 2.6: Histograms of Schlogl system (2.18) solutions with $\tau = 0.0001$ (units), final time $T=4$ (units), and 10,000 samples.

Appendix A. Examples

We exemplify the process of building matrix \mathbf{A} (2.2) for the Schlogl and isomer reactions.

Isomer reaction Here for simplicity, we exemplify the implementation of the system for the maximum values of species $Q^1 = 2$ and $Q^2 = 2$. According to (2.2.1), $Q = (Q^1 + 1) \times (Q^2 + 1) = 3^2$.

The vector d according to (2.2.1) is $[2, -2]$. The state matrix which contains all possible states has dimension $81^2 \times 2$ matrix:

$$\mathbf{x} = \begin{bmatrix} 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \end{bmatrix}^T \in \mathbb{R}^{3^2 \times 2}.$$

The matrix $\mathbf{A} \in \mathbb{R}^{Q \cdot Q \times Q \cdot Q}$ As an example for a maximum number of species $Q^1 = 2, Q^2 = 2$ the matrix \mathbf{A} is:

$$\mathbf{A} = \begin{bmatrix} -a_0(x_{1,:}) & 0 & a_2(x_{3,:}) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -a_0(x_{2,:}) & 0 & \ddots & 0 & 0 & 0 & 0 & 0 \\ a_1(x_{1,:}) & 0 & -a_0(x_{3,:}) & 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & a_1(x_{2,:}) & 0 & \ddots & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & a_1(x_{3,:}) & 0 & \ddots & 0 & a_2(x_{7,:}) & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & \ddots & 0 & a_2(x_{8,:}) & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 & -a_0(x_{7,:}) & 0 & a_2(x_{9,:}) \\ 0 & 0 & 0 & 0 & 0 & \ddots & 0 & -a_0(x_{8,:}) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_1(x_{7,:}) & 0 & -a_0(x_{9,:}) \end{bmatrix} \in \mathbb{R}^{9 \times 9}.$$

Schlogl reaction Here for simplicity, we exemplify the implementation of the system for the maximum value of the number of molecules $Q^1 = 5$. According to (2.2.1) the dimensions of \mathbf{A} are: $(Q^1 + 1 \times Q^1 + 1) = 6 \times 6$. The vector d (2.2.1) for this system $[1, -1, 1, -1]$. All possible states for this system are contained in the state vector

$$\mathbf{x} = [0, 1, 2, \dots, 5]^T \in \mathbb{R}^{1 \times 6}.$$

As an example matrix \mathbf{A} for maximum number of molecules $Q = 5$ is the following tridiagonal matrix:

$$\mathbf{A} = \begin{bmatrix}
 -a_0(\mathbf{x}_1) & a_2(\mathbf{x}_2) + a_4(\mathbf{x}_2) & 0 & 0 & 0 & 0 \\
 a_1(\mathbf{x}_1) + a_3(\mathbf{x}_1) & -a_0(\mathbf{x}_2) & \ddots & 0 & 0 & 0 \\
 0 & a_1(\mathbf{x}_2) + a_3(\mathbf{x}_2) & \ddots & \ddots & 0 & 0 \\
 0 & 0 & \ddots & \ddots & a_2(\mathbf{x}_5) + a_4(\mathbf{x}_5) & 0 \\
 0 & 0 & 0 & \ddots & -a_0(\mathbf{x}_5) & a_2(\mathbf{x}_6) + a_4(\mathbf{x}_6) \\
 0 & 0 & 0 & 0 & a_1(\mathbf{x}_5) + a_3(\mathbf{x}_5) & -a_0(\mathbf{x}_6)
 \end{bmatrix} \in \mathbb{R}^{6 \times 6}.$$

Chapter 3

Solving Stochastic Chemical Kinetics by Metropolis-Hastings Sampling

3.1 Introduction

In biological systems, chemical reactions are modeled stochastically. The system's state (the number of molecules of each individual species) is described by probability densities describing the quantity of molecules of different species at a given time. The evolution of probabilities through time is described by the chemical master equation (CME) [7].

The Stochastic Simulation Algorithm (SSA) first introduced by Gillespie [7], is a Monte Carlo approach to sample from the CME. The accuracy of different approaches in simulating stochastic chemical reactions is compared to histogram of samples obtained by SSA. However, SSA has a number of drawbacks such as it simulates one reaction at a time and therefore it is inefficient for most realistic problems. Alternative approaches have been developed trying to enhance the efficiency of SSA but most of them suffer from accuracy issues. The explicit tau-leaping method [8] is able to simulate multiple chemical reactions in a pre-selected time step of length τ by using Poisson random variables [33]. However, explicit tau-leaping method is numerically unstable for stiff systems [34]. Different implicit tau-leap approaches have been proposed to alleviate the stability issue [35, 36, 8, 37]. Sandu [44] considers an exact exponential solution to the CME, leading to a solution vector that coincides with the probability of SSA. Several approximation methods to the exact exponential solution as well as approximation to the explicit tau-leap are given in [46].

The availability of exact and approximate probability solutions motivates the use of Markov chain Metropolis algorithm to enhance the accuracy of explicit tau-leap method when using large time steps. The proposed method relies on explicit tau-leaping to generate candidate samples. The proposed probability density corresponds to that of tau-leaping [44], and the target probability density is provided by the CME. During the Markov process the candidate samples are evaluated based on approximations of target and proposal probability and are either accepted or rejected. The proposed technique requires the computation of a matrix exponential during the Markov process. The dimension of matrix grows with increasing number of species in a reaction system. In order to manage the computational expense of matrix exponentiation efficient approaches based on Krylov [47] and rational approximations [48, 49] are employed. Further computational savings are obtained by exponentiating only a sub-matrix that encapsulates the essential information about the transition of the system from the current to the proposed state.

The chapter is organized as follows. Section 3.2 reviews Monte Carlo approaches, and Section 3.3 discusses the application of Metropolis-Hastings algorithm to sample from the probability distribution generated by CME. Computationally efficient methods to accelerate exponentiating the matrix are discussed in Section 3.4. Numerical experiments carried out in Section 3.5 illustrate the accuracy of the proposed schemes. Conclusions are drawn in Section 3.6.

3.2 Markov Chain Monte Carlo methods

Markov Chain Monte Carlo (MCMC) methods are a class of algorithms to generate samples from desired probability distributions. A Markov chain is a discrete time stochastic process, i.e., a sequence of random variables (states) x_0, x_1, \dots where the probability of the next state of system depends only on the current state of the system and not on previous ones [50].

3.2.1 Metropolis methods

The Metropolis method is an MCMC process to obtain a sequence of random samples from a desired probability distribution $\pi(x)$, $x \in X \subset \mathbb{R}^n$, which is usually complex. A Markov chain with state space X and equilibrium distribution $\pi(x)$ is constructed and long runs of the chain are performed [51]. The original MCMC algorithm was given by Metropolis et al. [52] and was later modified by Hastings [53], with a focus on statistical problems.

A random walk is performed around the current state of the system x_{t-1} . A proposal distribution $g(x^*|x_{t-1})$ is used to suggest a candidate x^* for the next sample given the previous sample value x_{t-1} . The proposal distribution should be symmetric $g(x_{t-1}|x^*) = g(x^*|x_{t-1})$. The algorithm works best if the proposal density matches the shape of the target distribution, i.e. $g(x_{t-1}|x^*) \approx \pi(x)$. Proposals x^* are accepted or rejected in a manner that leads system toward the region of higher target probability $\pi(x)$ [54]. Specifically, one computes the target density ratio

$$\alpha = \frac{\pi(x^*)}{\pi(x_{t-1})} \quad (3.1)$$

and draws a random variable $\zeta \sim \text{uniform}(0, 1)$. The proposal is accepted or rejected as follows:

$$x_t := \begin{cases} x^* & \text{if } \zeta < \min(1, \alpha) \quad (\text{proposal accepted}), \\ x_{t-1} & \text{otherwise} \quad (\text{proposal rejected}). \end{cases}$$

3.2.2 Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm allows more freedom in the choice of the proposal distribution by relaxing the symmetry constraint [55]. The acceptance ratio (3.1) is changed to

$$\alpha = \alpha_1 \cdot \alpha_2. \quad (3.2)$$

Here α_1 is the ratio between the target probabilities of the proposal sample x^* and of the previous sample x_{t-1} . This can be evaluated by a function f which is an approximation of π

$$\alpha_1 = \frac{\pi(x^*)}{\pi(x_{t-1})} \approx \frac{f(x^*)}{f(x_{t-1})} \quad (3.3)$$

α_2 is the ratio of the proposal densities of x^* conditioned by x_{t-1} , and of x_{t-1} conditioned by x^* . This ratio is equal to one if the proposal distribution is symmetric.

$$\alpha_2 = \frac{g(x_{t-1}|x^*)}{g(x^*|x_{t-1})}. \quad (3.4)$$

Convergence of the Markov chain is guaranteed if the properties of detailed balance and ergodicity conditions are fulfilled [56]. Detailed balance requires that the probability of moving from x_{t-1} is the same as moving from x^* .

$$\pi(x_{t-1}) g(x_{t-1}|x^*) = \pi(x^*) g(x^*|x_{t-1})$$

Ergodicity requires that a chain starting from any state x_1 will return to x_1 if it runs long enough. In practice, it is not possible to establish with full certainty that a chain has converged [56].

3.3 Metropolis-Hastings for stochastic simulation

Here we discuss the application of the Metropolis-Hastings algorithm to generate samples from the CME distribution. SSA is currently the standard model for solving well-stirred chemically reacting systems; however, SSA does one reaction at a time that making it slow for real problems. On the other hand, alternative techniques such as explicit and implicit tau-leap methods are faster than SSA but suffer from low accuracy at larger time steps.

In the proposed approach, explicit tau-leap is employed to generate candidate samples. The samples are evaluated based on the acceptance ratio of Metropolis-Hastings algorithm. At the end of algorithm, the samples generated by this technique have the same distribution as given by CME, and the histogram of samples converges to the histogram of SSA solutions.

3.3.1 Target distribution

The target (exact) distribution $\mathcal{P}(x, t)$ of the state of the chemical system is given by the solution of the CME [7]

$$\frac{\partial \mathcal{P}(x, t)}{\partial t} = \sum_{r=1}^M a_r(x - v_r) \mathcal{P}(x - v_r, t) - a_0(x) \mathcal{P}(x, t). \quad (3.5)$$

Let Q^i is the total possible number of molecules of species S^i , $i = 1, \dots, N$. The total number of all possible states of the system is

$$Q = \prod_{i=1}^N (Q^i + 1). \quad (3.6)$$

CME is a linear ODE on the discrete state space of states \mathbb{R}^Q

$$\mathcal{P}' = A \cdot \mathcal{P}, \quad \mathcal{P}(\bar{t}) = \delta_{I(\bar{x})}, \quad t \geq \bar{t}. \quad (3.7)$$

and has an exact solution:

$$\mathcal{P}(\bar{t} + \tau) = \exp(T A) \cdot \mathcal{P}(\bar{t}) = \exp\left(\tau \sum_{r=0}^M A_r\right) \cdot \mathcal{P}(\bar{t}). \quad (3.8)$$

As explained in [44, 46], the diagonal matrix $A_0 \in \mathbb{R}^{Q \times Q}$ and the Toeplitz matrices $A_1, \dots, A_M \in \mathbb{R}^{Q \times Q}$ are:

$$(A_0)_{i,j} = \begin{cases} -a_0(x_j) & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases}, \quad (A_r)_{i,j} = \begin{cases} a_r(x_j) & \text{if } i - j = d_r, \\ 0 & \text{if } i - j \neq d_r, \end{cases} \quad (3.9)$$

and their sum $A \in \mathbb{R}^{Q \times Q}$ is

$$A = A_0 + \dots + A_M, \quad A_{i,j} = \begin{cases} -a_0(x_j) & \text{if } i = j, \\ a_r(x_j) & \text{if } i - j = d_r, r = 1, \dots, M, \\ 0 & \text{otherwise.} \end{cases} \quad (3.10)$$

Here x_j denotes the unique state with state space index $j = \mathcal{I}(x_j)$, where $\mathcal{I}(x)$ is the state-space index of state $x = [X^1, \dots, X^N]$:

$$\begin{aligned} \mathcal{I}(x) = & (Q^{N-1} + 1) \dots (Q^1 + 1) \cdot X^N + \dots \\ & + (Q^2 + 1) (Q^1 + 1) \cdot X^3 + (Q^1 + 1) \cdot X^2 + X^1 + 1. \end{aligned} \quad (3.11)$$

One firing of reaction R_r changes the state from x to $\bar{x} = x - \nu_r$. The corresponding change in state space index is:

$$\begin{aligned} \mathcal{I}(x) - \mathcal{I}(x - \nu_r) &= d_r, \\ d_r &= (Q^{N-1} + 1) \dots (Q^1 + 1) \cdot \nu_r^N + \dots \\ &+ (Q^2 + 1) (Q^1 + 1) \cdot \nu_r^3 + (Q^1 + 1) \cdot \nu_r^2 + \nu_r^1. \end{aligned}$$

At the current time \bar{t} the system is in the known state $x(\bar{t}) = \bar{x}$ and consequently the current distribution $\mathcal{P}(\bar{t}) = \delta_{\mathcal{I}(\bar{x})}$ is equal to one at $\mathcal{I}(\bar{x})$ and is zero everywhere else. The target distribution in our method is the exact solution (2.2)

$$\pi = \exp\left(\tau \sum_{r=0}^M A_r\right) \cdot \delta_{\mathcal{I}(\bar{x})}. \quad (3.12)$$

3.3.2 Proposal distribution

In our algorithm the explicit tau-leap method is employed to generate the candidate samples. Sandu [44] shows that the probability distribution generated by the tau-leap method is the solution of a linear approximation of the CME

$$\mathcal{P}(\bar{t} + \tau) = \exp(\tau \bar{A}) \cdot \mathcal{P}(\bar{t}) = \exp\left(\tau \sum_{r=0}^M \bar{A}_r\right) \cdot \mathcal{P}(\bar{t}) \quad (3.13)$$

where the diagonal matrix $\bar{A}_0 \in \mathbb{R}^{Q \times Q}$ and the Toeplitz matrices $\bar{A}_1, \dots, \bar{A}_M \in \mathbb{R}^{Q \times Q}$ are:

$$(\bar{A}_0)_{i,j} = \begin{cases} -a_0(\bar{x}) & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases}, \quad (\bar{A}_r)_{i,j} = \begin{cases} a_r(\bar{x}) & \text{if } i - j = d_r, \\ 0 & \text{if } i - j \neq d_r, \end{cases} \quad (3.14)$$

where the arguments of all propensity functions are the current state \bar{x} [46]. Therefore the proposal distribution used in our method is:

$$g = \exp\left(\tau \sum_{r=0}^M \bar{A}_r\right) \cdot \delta_{\mathcal{I}(\bar{x})}. \quad (3.15)$$

3.3.3 Markov process

The Markov process starts with the values of species at the current time. The candidate sample is generated by the tau-leap method. Both the candidate sample and the current sample are evaluated based on the acceptance ratio (3.2). The target density ratio (3.3) is

$$\alpha_1 = \frac{\pi(x^*)}{\pi(x_{t-1})} = \frac{\delta_{I(x^*)}^T \cdot \exp\left(\tau \sum_{r=0}^M A_r\right) \cdot \delta_{I(\bar{x})}}{\delta_{I(x_{t-1})}^T \cdot \exp\left(\tau \sum_{r=0}^M A_r\right) \cdot \delta_{I(\bar{x})}}. \quad (3.16)$$

For the tau-leap method x^* is generated independent of x_{t-1} and vice versa. Hence the proposal density ratio (3.4) is

$$\alpha_2 = \frac{g(x_{t-1})}{g(x^*)} = \frac{\delta_{I(x_{t-1})}^T \cdot \exp\left(\tau \sum_{r=0}^M \bar{A}_r\right) \cdot \delta_{I(\bar{x})}}{\delta_{I(x^*)}^T \cdot \exp\left(\tau \sum_{r=0}^M \bar{A}_r\right) \cdot \delta_{I(\bar{x})}}. \quad (3.17)$$

From (3.16) and (3.17) the acceptance ratio α is:

$$\alpha = \frac{\delta_{I(x^*)}^T \cdot \exp\left(\tau \sum_{r=0}^M A_r\right) \cdot \delta_{I(\bar{x})}}{\delta_{I(x_{t-1})}^T \cdot \exp\left(\tau \sum_{r=0}^M A_r\right) \cdot \delta_{I(\bar{x})}} \cdot \frac{\delta_{I(x_{t-1})}^T \cdot \exp\left(\tau \sum_{r=0}^M \bar{A}_r\right) \cdot \delta_{I(\bar{x})}}{\delta_{I(x^*)}^T \cdot \exp\left(\tau \sum_{r=0}^M \bar{A}_r\right) \cdot \delta_{I(\bar{x})}}. \quad (3.18)$$

In the acceptance/rejection test, samples which have a higher density ratio will be selected as the next state and samples which have a lower density ratio will be rejected. The samples generated by this approach have approximately the same density as CME (SSA) even when using a large time step in the proposal (explicit tau-leap). The only drawback of this method is the cost of performing matrix exponential. In the following section we discuss several ways to reduce this computational cost.

3.4 Matrix exponential

The computation of a large matrix exponential is a problem of general interest, and a multitude of approaches suited to different situations are available. The most straightforward, and naive approach is a direct application of the definition of the matrix exponential

$$\exp(\mathbf{A}) = \sum_{k=0}^{\infty} \frac{\mathbf{A}^k}{k!}. \quad (3.19)$$

While this approach is guaranteed to converge if sufficiently, possibly very many, terms are used, there are substantial numerical stability problems in the case where either the norm or the dimension of \mathbf{A} is very large [47, 48].

3.4.1 Rational approximation methods for full matrix exponential

Several rational approximation methods have been developed to overcome the stability and speed of convergence problems posed by the direct method. These are based on standard function approximation methods, in the case of the Pade approximation [48], or on the approximation of complex contour integrals, in the case of CRAM [49]. These methods are usually paired with the “scaling and squaring” process of Higham [57] to further increase the stability of the computation.

Pade approximation

The Pade approximation for $\exp(\mathbf{A})$ is computed using the (p, q) -degree rational function:

$$\begin{aligned} P_{pq}(\mathbf{A}) &= [D_{pq}(\mathbf{A})]^{-1} N_{pq}(\mathbf{A}), \\ N_{pq}(\mathbf{A}) &= \sum_{j=0}^p \frac{(p+q-j)! p!}{(p+q)! j! (p-j)!} \mathbf{A}^j, \\ D_{pq}(\mathbf{A}) &= \sum_{j=0}^q \frac{(p+q-j)! q!}{(p+q)! j! (q-j)!} (-\mathbf{A})^j, \end{aligned}$$

which is obtained by solving the algebraic equation:

$$\sum_{k=0}^{\infty} \frac{\mathbf{A}^k}{k!} - \frac{N_{pq}(\mathbf{A})}{D_{pq}(\mathbf{A})} = O(\mathbf{A}^{p+q+1})$$

in which $P_{pq}(x)$ must match the Taylor series expansion up to order $p+q$ [47]. MATLAB’s `expm` function makes use of thirteenth order Pade approximation with scaling and squaring [48].

Rational approximations of integral contours

The integral contour approach constructs parabolic and hyperbolic contour integrals on left complex plane and uses quadrature points θ_k from the contour and quadrature weights α_k [49] to approximate full matrix exponentiation:

$$r(z) = \sum_{k=1}^N \frac{\alpha_k}{z - \theta_k}.$$

In the case where the spectrum of \mathbf{A} is confined to a region near the negative real axis of the complex plane methods based on the rational approximation of integral contours are likely to converge faster than the Pade approximation. In this work we use the Matlab scripts provided by [49] for both rational approximation methods and the coefficients θ and α .

Chebyshev Rational Approximation Method (CRAM)

Let $\pi_{k,k}$ denote the set of rational functions

$$r_{k,k}(x) = \frac{p_k(x)}{q_k(x)}$$

where p_k and q_k are the polynomials of order k computed such as to optimize the following error

$$(p_k, q_k) = \arg \inf_{r_{k,k} \in \pi_{k,k}} \left\{ \sup_{x \in \mathbb{R}_-} |r_{k,k}(x) - e^x| \right\}.$$

The primary difficulty in making use of the CRAM method is the procurement of suitable coefficients of the polynomials p_k and q_k . A method for obtaining these coefficients is given in [58], and the values for $k = 14$ and $k = 16$ are provided in [49].

3.4.2 Krylov based approximation for matrix vector products

For our purposes we do not seek the entire solution of $\exp(\mathbf{A})$, in fact we would like only a single element of the result. Krylov based approximations get us one step closer to this ideal. Where the rational approximation methods seek to approximate the entirety of equation (3.19), Krylov based methods seek only an approximation to the matrix-vector product $\exp(\mathbf{A})b$.

This is done by first computing the m -dimensional Krylov subspace

$$\mathcal{K}_m = \text{span} \{b, \mathbf{A}b, \dots, \mathbf{A}^{m-1}b\}$$

using the Arnoldi iteration to compute the $n \times m$ orthonormal basis matrix \mathbf{V}_m and the $m \times m$ upper Hessenberg matrix \mathbf{H}_m with $m \ll n$ such that

$$\text{span}(\mathbf{V}_m) = \mathcal{K}_m, \quad \mathbf{H}_m = \mathbf{V}_m^T \mathbf{A} \mathbf{V}_m.$$

The approximation is constructed as

$$\exp(\mathbf{A})b = \mathbf{V}_m \mathbf{V}_m^T \exp(\mathbf{A}) \mathbf{V}_m \mathbf{V}_m^T b = \|b\| \mathbf{V}_m \exp(\mathbf{H}_m) e_1 \quad (3.20)$$

where e_1 is the first canonical basis vector. The small matrix exponential term in (3.20) can be computed using one of the rational approximation methods with scaling and squaring extremely cheaply. The EXPOKIT software of Sidje [47] makes use of these techniques, with some extra consideration for Markovian cases, where the approximation of $w(t) = \exp(t\mathbf{A})v$ is subject to the constraint that the resulting vector is a probability vector with components in the range of $[0, 1]$ and the sum of these components is approximately one.

3.4.3 Faster approximation techniques for a single element

Since we seek only a single element of the matrix exponential $(\exp(\mathbf{A}))_{i,j}$ we propose two techniques to speed up this computation.

A single element Krylov approach

Using equation (3.20) with $b = e_j$ leads to

$$(\exp(\mathbf{A}))_{i,j} = e_i^T \exp(\mathbf{A}) e_j = (e_i^T \mathbf{V}_m) (\exp(\mathbf{H}_m) e_1). \quad (3.21)$$

The exponential matrix entry is computed for the cost of an m -dimensional Pade approximation and an m -dimensional dot product since $(e_i^T \mathbf{V}_m)$ can be computed for “free” by simply reading off the i th row of \mathbf{V}_m , and similarly $(\exp(\mathbf{H}_m) e_1)$ is just the first column of $\exp(\mathbf{H}_m)$. This approach avoids the construction of any additional n -dimensional vectors or their products.

Exponentiation of a selected sub-matrix

Computing the exponential of a large matrix is expensive. When the number of species in a reaction system is high, the dimensions of the matrix (3.9) for target probability as well as dimensions of matrix (3.14) for proposal probability grow quickly. For the case of n species where each has a maximum Q molecules the dimension of matrix will be $(Q + 1)^n \times (Q + 1)^n$.

In order to reduce costs we propose to exponentiate a sub-matrix of the full matrix. The selected rows and columns contain indices of both the current state of system at t_n and candidate state at $t_n + \tau$. The motivation comes from the fact that states which are far from the current and the proposed ones do not impact significantly the acceptance/rejection test of Metropolis-Hastings algorithm and can be disregarded. Numerical experiments indicate that the error in an element $(\exp(\mathbf{A}))_{i,j}$ computed using a properly sized sub-matrix instead of full matrix is small.

In order to obtain the proper size of a sub-matrix for each reaction system, we use specific information from the reaction system such as propensity functions, time step and maximum number of molecules in the system. Recall the general tau-leap formula [8].

$$x(\bar{t} + \tau) = x(\bar{t}) + \sum_{j=1}^M V_j K(a_j(x(\bar{t}))\tau).$$

where $K(a_j(x(\bar{t}))\tau)$ is a random number drawn from a Poisson distribution with parameter $a_j(x(\bar{t}))\tau$ and V_j is the j -th column of stoichiometry matrix. The expected value of the jump in the number of molecules is

$$E[x(\bar{t} + \tau) - x(\bar{t})] = \sum_{j=1}^M V_j a_j(x(t_0)) \tau. \quad (3.22)$$

Motivated by (3.22) we consider the following initial estimate of the size of the sub-matrix:

$$S \propto \frac{\|V\|}{N} \sum_{j=1}^M a_j(x(t_0)) \tau \propto \bar{a}(x(t_0)) \tau, \quad (3.23)$$

where $\bar{a}(x(t_0))$ is the average of propensity functions for all reactions calculated at the initial values of number of molecules.

We seek to select a range of state indices that covers the current and proposed states. The sub-matrices are built by selecting only the rows and columns in this range from (3.9) and (3.14). If the range of indices is small then the exponential computations are fast. However, if this range does not cover the representative states (both the current sample and the proposed sample), the probability ratio of the proposed sample can be far from the target probability, and the proposed sample is likely to be rejected. Choosing the size of the sub-matrix for maximum efficiency has to balance the cost of obtaining a sample (smaller is better for the cost of exponentiation) with the likelihood of accepting samples (larger is better for accuracy of approximation).

3.5 Numerical experiments

This section discusses the application of the Metropolis-Hastings algorithm to generate samples from the SSA distribution for three test systems: Schlogl [34], reversible isomer [34], and Lotka Volterra reactions [7].

3.5.1 Schlogl reaction

We first consider the Schlogl reaction system from [34] explained in 2.4.2: with final time $T = 4$, initial conditions $x(0) = 250$ molecules, and maximum values of species $Q^1 = 900$ molecules. We consider a time step $\tau = 0.4$ for which the explicit tau-leap solution has a relatively large error compared to SSA.

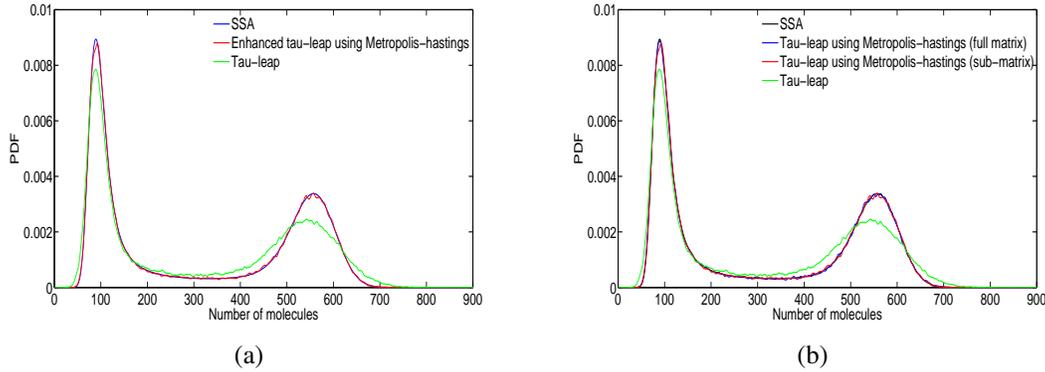


Figure 3.1: Histograms of Schlogl system (2.18) solutions with $\tau = 0.4$ (units), final time $T=4$ (units), and 10,000 samples.

The initial guess for the size of sub-matrix given by (3.23) is 250×250 and works well for the model. To accept 1,000 samples the MCMC process rejects about $\sim 1,200$ samples when using full matrix (whose size is 901×901). While the number of rejected using sub-matrix is approximately 1,300. Decreasing the size of sub-matrix leads to a larger number of rejected samples. For example using a sub-matrix of size 100×100 results in approximately 2,500 rejected samples, so this matrix size is too small. Another metric to assess whether the sub-matrix size is appropriate is the size of the residual obtained by exponentiating the full matrix and the sub-matrix. In this simulation the residual is below 10^{-8} for a sub-matrix size of 250×250 . We have observed empirically that when the residual is larger than 10^{-2} the sample is likely to be rejected. The moderate number of rejected samples using the sub-matrix and the small residual indicate that the 250×250 size yields a good approximation for large matrix exponentiation.

Figure 3.1(a) illustrates the histogram of Schlogl reaction results obtained by SSA, explicit tau-leap and Metropolis-Hastings using full matrix size. Figure 3.1(b) shows that the results obtained with a sub-matrix of size 250×250 have no visible reduction in accuracy. Since all the eigenvalues of the matrix lie very closely to each other we employ the order ten of rational approximation of integral contours discussed in Section 3.4.1 which is faster than other techniques for exponentiating both the full matrix and the sub-matrix. The CPU time for obtaining one sample using the sub-matrix is 0.32 sec., about half the CPU time required when using the full matrix (0.76 sec). For comparison the CPU times for obtaining one sample are 0.15 sec. using SSA and 0.02 sec. using tau-leaping.

3.5.2 Isomer reaction

The reversible isomer reaction system from [34] is explained in 2.4.1 The reaction rate values are $c_1 = 10$, $c_2 = 10$ (units), the time interval is $[0, T]$ with $T = 10$ (time units), the initial conditions are $x_1(0) = 40$, $x_2(0) = 40$ molecules, and the maximum values of the species are $Q^1 = 80$ and $Q^2 = 80$ molecules.

The estimate give by equation (3.23) is 20 and since this reaction system has two species the initial guess for the size of the sub-matrix is $20^2 \times 20^2$. In order to be more conservative a sub-matrix of size 500×500 is selected. In order to accept 1,000 samples the Markov process rejects approximately 5,000 samples when using the full matrix (of size $6,561 \times 6,561$), and about 8,000 samples when using the sub-matrix. Decreasing the size of sub-matrix leads to many more rejected samples. Our empirical observations show again that when the residual is larger than 10^{-2} the sample is likely to be rejected. We conclude that the current sub-matrix provides a good approximation for large matrix exponentiation.

Figure 3.2(a) shows the histogram of the isomer reaction solutions obtained by SSA, explicit tau-leap and by Metropolis-Hastings using the full size matrix (3.9) and (3.14). Figure 3.2(b) shows the results using the sub-matrix of size 500×500 . There is no visible reduction in accuracy. Since all the eigenvalues of the matrix lie very closely to each other we employ the order ten of rational approximation of integral contours discussed in Section 3.4.1 which is faster than other techniques for exponentiating both the full matrix and the sub-matrix. The CPU times for obtaining one sample are 20.37 sec. using the sub-matrix and 38.70 sec. using the full matrix. For comparison obtaining one sample using SSA takes 0.15 sec. and using tau-leaping takes 0.05 sec.

3.5.3 Lotka Volterra reaction

The last test case is Lotka Volterra reaction system [7]:



The reaction stoichiometry matrix and the propensity functions are:

$$V = \begin{bmatrix} 1 & -1 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}, \quad \begin{aligned} a_1(x) &= c_1 x_1 Y, & a_2(x) &= c_2 x_2 x_1, \\ a_3(x) &= c_3 x_2, & a_4(x) &= c_4 x_1. \end{aligned}$$

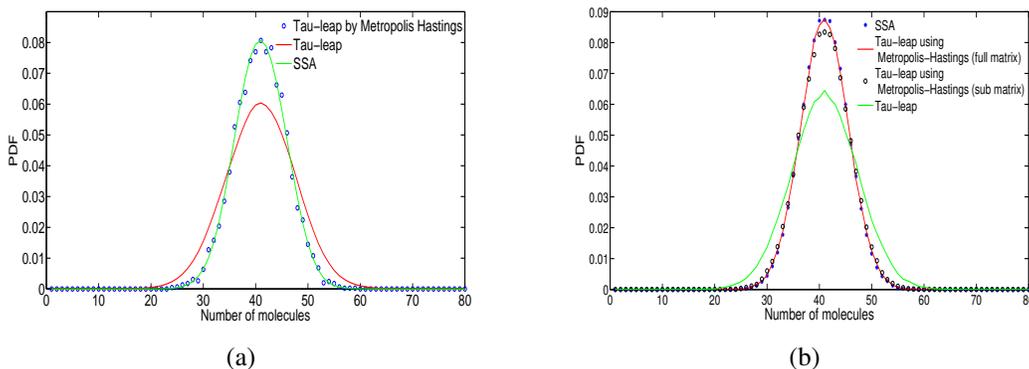


Figure 3.2: Histograms of isomer system (2.17) solutions with $\tau = 0.05$ (units), final time $T=1$ (units), and 10,000 samples.

The following parameter values are used (in appropriate units):

$$c_1 = 0.0002, \quad c_2 = 0.01, \quad c_3 = 10, \quad c_4 = 10, \quad Y = 10^{-5},$$

the final time is $T = 1$, the initial conditions are $x_1(0) = 1000$, $x_2(0) = 1000$ molecules, and the maximum values of species are $Q^1 = 2000$ and $Q^2 = 2000$ molecules. The resulting full matrix has dimension $2001^2 \times 2001^2$ and exponentiation is not feasible without the sub-matrix approximation.

The value predicted by equation (3.23) is 125, and since this reaction system has two species the initial guess for the size of sub-matrix is $15,625 \times 15,625$. This size does not work well for this system with very large number of molecules and almost never covers both current and candidate states. We increase the size of sub-matrix to $500,000 \times 500,000$, a value obtained by trial and error. Figure 3.3 illustrates the histogram of Lotka-Volterra solutions obtained by SSA, tau-leap method, and Metropolis-Hastings using a sub-matrix of size discussed above. The Metropolis-Hastings sampling is very accurate.

The CPU time of matrix exponentiation using the contour integral method discussed in Section 3.4.1 is four times faster than using the Krylov method discussed in Section 3.4.2. However, the Krylov method gives more accurate and stable results for large matrices [47]. Using 30 vectors in Krylov method, gives smaller number of rejected samples during the Markov process than the number of rejected samples using contour integral method. The CPU time for obtaining one sample using Metropolis-Hastings is a few hours, in comparison to 1.51 sec. using SSA and 0.21 sec. using tau-leap.

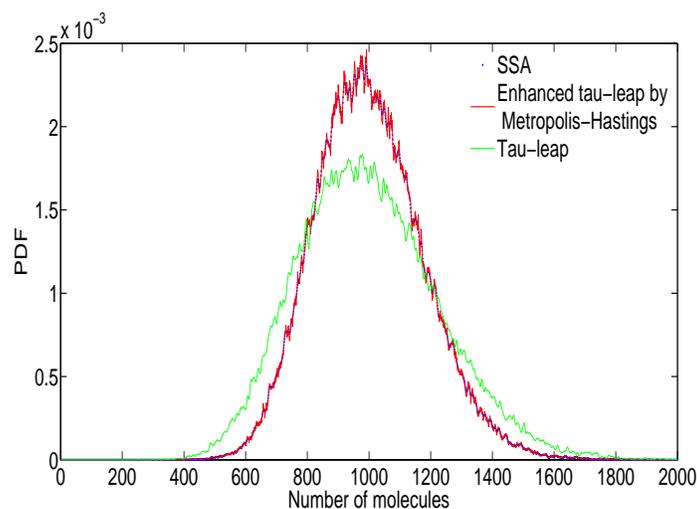


Figure 3.3: Histograms of Lotka-Volterra system (3.24) solutions with $\tau = 0.01$ (units), final time $T=1$ (units), and 10,000 samples.

3.6 Conclusions

This study applies the Metropolis-Hastings algorithm to stochastic simulation of chemical kinetics. The proposed approach makes use of the CME and the exponential form of its exact solution as the target probability in the Markov process. The approximation of the explicit tau-leap method is then employed for the proposal probability. The samples generated by constructing the Markov process have the same distribution as SSA even the proposals are obtained using explicit tau-leap with a large time step. Computing matrix exponentials can become a computational bottleneck for large systems. Efficient approximation methods are needed. A practical approach consists of selecting a sub-matrix and exponentiating it using fast approaches like Expokit and rational approximation to significantly reduce the cost of the algorithm. Current work of the authors focuses on developing faster approximation techniques that compute a single element of the matrix exponential. While the practical performance of the Metropolis-Hastings stochastic simulation algorithm is not yet competitive with existing methods, the approach is novel and opens the door for many future developments.

Chapter 4

A State-Space Approach to Analyze Structural Uncertainty in Physical Models

4.1 Introduction

Studies of the natural world rely on both experimental and theoretical tools to understand and characterize the physical phenomena of interest. First-principles computer simulation models encapsulate our knowledge about the physical processes that govern the evolution of a natural system. Metrology, as “the science of measurement, embracing both experimental and theoretical determinations at any level of uncertainty in any field of science and technology”, is intimately connected to computer simulations. Metrology (specifically, the data obtained from measurements of physical quantities) is routinely used to inform computer simulation models. For example, many of the physical parameters used in the physics-based models have values given by direct measurements. Vice-versa, computer simulation models are being increasingly used by the metrological community to compare and to interpret measurements, to design experiments, and as a surrogate for destructive testing. Virtual Metrology seeks to predict metrology variables (either measurable or non measurable) using computer models that describe the underlying physical processes, and found applications in circuit manufacturing [59]. Computer simulation methods have been used to improve accuracy in optical metrology and nano-optics design [60]. The use of computer simulation models based on partial differential equations in metrology is discussed in [61]. Computer simulation methods have been shown to be

an effective technique to investigate probabilistic regularities and to develop mathematical and applied statistic tools used in problems of statistical quality control [62]. Computer models are also used to draw valid inferences from the analysis of metrological data, e.g., to find criteria for rejection/acceptance of hypotheses [63].

First-principles models encapsulate our knowledge about the physical processes that govern the evolution of a natural system. The solutions of such models approximate the evolution of the physical system under consideration. However, models are imperfect since they are subject to multiple uncertainties associated with the input data, external forcings, parameters, and the structure of the model. The process of uncertainty quantification includes the identification, characterization, propagation, analysis, and finally the reduction of these uncertainties [64, 65]. Modern metrology is concerned with the characterization of uncertainties in computer simulation model results.

Type-A evaluation of measurement uncertainty[66] in computer model simulation results is typically due to uncertainty in model inputs, for example, from parameters that are coming from uncertain measurements, or from random forcings that drive the model. Type-B evaluation of measurement uncertainty [66] in the model results is more subtle, and is typically associated with both the numerical errors and with the inaccurate representation of the real-world physical processes in the model. There is a long tradition in simulation models to estimate the upper bounds of numerical errors, and to control them (bring them within the desired accuracy level) by adapting the resolution of the numerical solvers. However, the uncertainty associated with missing or poorly represented physical processes is much less understood. There is considerable need to quantify this form of type-B evaluation of measurement uncertainty in the model results, and to find upper limits on its magnitude. This work seeks to develop a general approach to understanding and quantifying model structural uncertainty.

Structural uncertainty, also named model-form uncertainty, is due to insufficient knowledge of the true physics and therefore to an incomplete representation of reality by the model [9, 10]. While uncertainties due to data and parameter values have been studied extensively, comparatively little work has been devoted to date to studying structural uncertainties. Two approaches have been employed to resolve the structural uncertainty: model averaging and discrepancy modeling (or model calibration). Model averaging considers weighted averages of the outputs of different models, with the weights obtained from a metric of model adequacy, e.g., likelihood with respect to the data. The limitations of this approach are that it requires the use of multiple models, and that the models may not be independent. Discrepancy modeling assesses the model based on the discrepancy between model outputs and the real data [11].

In [67] a Gaussian Process is used to model the discrepancy between the model outputs and reality, and then a Markov Chain Monte Carlo (MCMC) is employed to sample the posterior distribution of the discrepancy. In [65] a Bayesian approach is taken to quantify different aspects of model uncertainty such as uncertainty in parameters, parametric variability, residual variability, observation error, code uncertainty, and model inadequacy. The model inadequacy is defined as the difference between the true mean value of the physical quantity of interest and the model output for the true values of the inputs. The posterior distribution over the parameters is sampled by MCMC to obtain the calibrated values of parameters. In [11] a distinction is drawn between external discrepancies that correspond to the entire model inadequacies, and internal discrepancies corresponding to individual sub-functions that compose the model. The Integrated Bayesian Uncertainty Estimator (IBUNE) discussed in [68] considers uncertainties coming from parameters, inputs, and model structure. IBUNE combines Bayesian model averaging (BMA) to reduce the structural uncertainties of individual sub-models with the Shuffled Complex Evolution Metropolis (SCEM) algorithm for probabilistic parameter estimation to the input and parameter uncertainties.

This study seeks to identify the missing dynamics in models that leads to the discrepancy between the model outputs and the measurements of the physical world. Our approach is to represent the dynamics of the structural uncertainty using linear state-space models, and to identify the parameters of these models from the known model states and model-observations discrepancies. The predicted structural uncertainties can be used to correct the model solutions, and therefore to improve the model. Previous work that has applied the theory of linear dynamical systems to structural uncertainty includes the use of an extended Kalman filter for system identification of seismic model errors [69], and the use of an ensemble Kalman filter to manage the structural uncertainties in reservoir models [70].

The remainder of this chapter is organized as follows, Section 4.2 describes the proposed methodology for analysis of structural uncertainty. Numerical results are presented in Section 4.3. Section 4.4 summarizes the findings of this work and draws the conclusions.

4.2 Structural uncertainty in computer simulation models

We now discuss the Type-B evaluation of measurement uncertainty in computer simulation models due to insufficient knowledge of the true physics and therefore to an incomplete

representation of reality by the model [9, 10]. This is named structural uncertainty or model-form uncertainty.

Consider a physical process with internal state \mathbf{v}_t at the discrete time $t \in \mathbb{N}$, evolving according to the dynamic equation:

$$\mathbf{v}_t = \mathcal{P}(\mathbf{v}_{t-1}) + \eta_t, \quad t = 1, \dots, T. \quad (4.1)$$

The exact physics \mathcal{P} and the physical state \mathbf{v}_t are not known exactly. We take measurements of a finite number of observables $\mathbf{y}_t \in \mathbb{R}^m$ of the physical process:

$$\mathbf{y}_t = h(\mathbf{v}_t) + \boldsymbol{\varepsilon}_t, \quad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, \mathbf{R}_t), \quad t = 1, \dots, T, \quad (4.2)$$

where $\boldsymbol{\varepsilon}_t$ is the observation error, assumed to be normally distributed.

A computer model \mathcal{M} evolves the model state $\mathbf{x}_t \in \mathbb{R}^n$ from one discrete time (t) to the next ($t + 1$):

$$\mathbf{x}_t = \mathcal{M}(\mathbf{x}_{t-1}, \hat{\theta}), \quad t = 1, \dots, T. \quad (4.3a)$$

Without loss of generality we assume that the parameters of the model $\theta \in \mathbb{R}^\ell$ take the best-fitting values $\hat{\theta}$ and we don't have any parameter uncertainty. The observation operator \mathcal{H} maps the model state onto the observation space; the model-predicted values $\mathbf{z}_t \in \mathbb{R}^m$ of the observations (8.3) are:

$$\mathbf{z}_t = \mathcal{H}(\mathbf{x}_t), \quad t = 1, \dots, T. \quad (4.3b)$$

The computer model \mathcal{M} seeks to approximate the true physical process; specifically, the model state approximates the physical state,

$$\mathbf{x}_t \approx \xi(\mathbf{v}_t), \quad t = 1, \dots, T, \quad (4.4)$$

where the operator ξ can represent, for example, the sampling of a continuous temperature field onto a finite dimensional computational grid. The model dynamics (8.1a) approximates the dynamics of the physical system (8.1b). If we initialize the model at time t with an idealized value (a projection of the real state), the model prediction at time $t + 1$ will differ from the reality:

$$\xi(\mathbf{v}_t) = \mathcal{M}(\xi(\mathbf{v}_{t-1}), \hat{\theta}) + \boldsymbol{\delta}_t(\mathbf{v}_{t-1}), \quad t = 1, \dots, T. \quad (4.5)$$

The discrepancy $\boldsymbol{\delta}_t$ between model prediction and the reality, projected onto the model space, is the structural uncertainty of concern here. In the discussion that follows we will make the following simplification. We assume that the physical system is finite dimensional, $\mathbf{v}_t \in \mathbb{R}^n$, and that the model state lives in the same space as reality, i.e., $\mathbf{x}_t \approx \mathbf{v}_t$

and $\xi(\cdot) \equiv id$ is the identity operator in (8.1c), and $\mathcal{H}(\cdot) \equiv h(\cdot)$ in (8.3) and (8.4). This assumption means that the discretization errors are very small, and that the main source of error is the missing representation of some physical processes. With these assumptions the evolution equations for the physical system (8.1b) and the physical observations equation (8.3) become, respectively:

$$\mathbf{v}_t = \mathcal{P}(\mathbf{v}_{t-1}) + \eta_t \quad (4.6a)$$

$$= \mathcal{M}(\mathbf{v}_{t-1}, \hat{\theta}) + \delta_t(\mathbf{v}_{t-1}) + \eta_t, \quad t = 0, \dots, T-1,$$

$$\mathbf{y}_t = h(\mathbf{v}_t) + \varepsilon_t. \quad (4.6b)$$

We seek to understand the structure of the model-form error (8.2) by comparing the observations (8.6b) of the real system against the model predicted values of these observables (8.4), i.e., by considering the discrepancies in the observable quantities:

$$\Delta_t = \mathbf{z}_t - \mathbf{y}_t \in \mathbb{R}^m, \quad t = 1, \dots, T. \quad (4.7)$$

Specifically, our goal is to model the state error δ_t in order to gain understanding of the missing dynamics, i.e., of the physical processes that are not captured by the model \mathcal{M} . Moreover, good estimates of the discrepancy δ_t allow to improve model predictions by applying the correction (8.6a) to model results:

$$\mathbf{v}_t \approx \mathbf{x}_t + \delta_t. \quad (4.8)$$

We consider two techniques to quantify the structure of the uncertainty in the models. The first is the global structural uncertainty approach, which models the discrepancy between the model state and physical state over the entire state space. The second is the local uncertainty approach, which only takes account the variables in the vicinity of the source of uncertainty and not all variables in the system. We now give a detailed description of these approaches.

4.2.1 Global models for structural uncertainty

Rooted in control engineering, state-space models (SSMs) are dynamical systems that describe the probabilistic dependence between the latent variables and the observed measurements [71]. A state-space vector evolves forward in time under the control of external inputs. An output equation captures the relationship between the system state, the input, and the output. A linear time-invariant (LTI) discrete-time SSM has the form:

$$\zeta_t = \mathbf{A} \cdot \zeta_{t-1} + \mathbf{B} \cdot \mathbf{u}_{t-1} + \mathbf{K} \cdot \mathbf{v}_{t-1}, \quad t = 1, \dots, T; \quad \zeta_0 = \text{given}, \quad (4.9a)$$

$$\gamma_t = \mathbf{C} \cdot \zeta_t + \mathbf{D} \cdot \mathbf{u}_t + \varepsilon_t, \quad t = 1, \dots, T, \quad (4.9b)$$

where (4.9a) is the state equation, and (4.9b) is the output equation. Here $\zeta_t \in \mathbb{R}^{n_{\text{ssm}}}$ is the state vector, $\mathbf{u}_t \in \mathbb{R}^{p_{\text{ssm}}}$ is the input or control variable, $\mathbf{y}_t \in \mathbb{R}^{m_{\text{ssm}}}$ is output of the system (observable quantities), $\mathbf{v}_t \in \mathbb{R}^{q_{\text{ssm}}}$ is the disturbance of the model, and $\boldsymbol{\varepsilon}_t \in \mathbb{R}^{m_{\text{ssm}}}$ is the observation error. The evolution of the system is captured by the dynamics or state matrix $\mathbf{A} \in \mathbb{R}^{n_{\text{ssm}} \times n_{\text{ssm}}}$, the input matrix $\mathbf{B} \in \mathbb{R}^{n_{\text{ssm}} \times p_{\text{ssm}}}$, the output or sensor matrix $\mathbf{C} \in \mathbb{R}^{m_{\text{ssm}} \times n_{\text{ssm}}}$, the feed-through or feed forward matrix $\mathbf{D} \in \mathbb{R}^{m_{\text{ssm}} \times p_{\text{ssm}}}$, and the disturbance matrix $\mathbf{K} \in \mathbb{R}^{n_{\text{ssm}} \times q_{\text{ssm}}}$.

In order to understand the structure of uncertainty in this work we employ SSMs (4.9) to describe the evolving discrepancy between the model and the true physics, as follows:

- The state vector ζ_t consists of hidden states that represent the dynamics of the structural model error, and are driven by the error dynamics of the model.
- The hidden error ζ_t depends on the model solution at every time step and on the discrepancy between the model outputs and the physical observations. The inputs of our SSM are therefore the model predictions (8.1a) :

$$\hat{\mathbf{x}}_t = \mathcal{M}(\mathbf{x}_{t-1}, \hat{\theta}), \quad t = 1, \dots, T. \quad (4.10)$$

- The outputs of the SSM are the observed discrepancies (8.5) between the model forecast (4.10) and reality:

$$\hat{\Delta}_t = \hat{\mathbf{z}}_t - \mathbf{y}_t = h(\hat{\mathbf{x}}_t) - \mathbf{y}_t, \quad t = 1, \dots, T. \quad (4.11)$$

- The outputs of the SSM predict the total stat-space discrepancy $\hat{\Delta}_t$.

The SSM that models the global error dynamics is:

$$\zeta_t = \mathbf{A} \cdot \zeta_{t-1} + \mathbf{B} \cdot \hat{\mathbf{x}}_t + \mathbf{K} \cdot \boldsymbol{\eta}_t, \quad \zeta_0 = \text{given}, \quad (4.12a)$$

$$\hat{\Delta}_t = \mathbf{C} \cdot \zeta_t + \mathbf{D} \cdot \hat{\mathbf{x}}_t + \boldsymbol{\varepsilon}_t. \quad (4.12b)$$

From the sequence of known inputs $\hat{\mathbf{x}}_t$ and the known outputs $\hat{\Delta}_t$, $t = 1, \dots, T$ one can perform a system identification and fit the matrices \mathbf{A} , \mathbf{B} , \mathbf{K} , \mathbf{C} , \mathbf{D} . The matrix entries contain information about the structure and source of uncertainty in models; this aspect will be discussed in detail in the numerical experiments Section 4.3.

It is important to note that in the global approach, the SSM (4.12) approximates the dynamics of the model error over the entire model space. Since the dimension of the hidden error vector $\zeta_t \in \mathbb{R}^{n_{\text{ssm}}}$ can be much smaller than the dimension of the model state-space

error vector $\delta_t \in \mathbb{R}^n$, $n_{\text{ssm}} < n$, the hidden error vector predicted by (4.12) needs to be lifted up to the model space:

$$\delta_t = \mathbf{F} \cdot \zeta_t + \mathbf{G} \cdot \hat{\mathbf{x}}_t. \quad (4.13)$$

In order to define the projection matrices $\mathbf{F} \in \mathbb{R}^{n \times n_{\text{ssm}}}$ and $\mathbf{G} \in \mathbb{R}^{n \times n}$ and one needs to use additional information about the structure of the model and the observation operator. Some possible definitions are discussed below.

- The global error dynamics model (4.12) can be defined on the entire state-space, i.e., $n_{\text{ssm}} = n$. In this case the hidden error approximates the entire model error vector $\zeta_t \approx \delta_t$, and therefore $\mathbf{F} = \mathbf{I}$ and $\mathbf{G} = \mathbf{0}$ in (4.13).
- The hidden error dynamics vector (4.12) can represent the projection of the full model error onto a subspace spanned by the orthonormal basis $\Phi \in \mathbb{R}^{n \times n_{\text{ssm}}}$ with $\Phi^T \cdot \Phi = \mathbf{I} \in \mathbb{R}^{n_{\text{ssm}} \times n_{\text{ssm}}}$. In this case we have:

$$\zeta_t \approx \Phi^T \cdot \delta_t, \quad \delta_t \approx \Phi \Phi^T \cdot \zeta_t \quad \Rightarrow \quad \mathbf{F} = \Phi \Phi^T \quad \text{and} \quad \mathbf{G} = \mathbf{0}.$$

The reduced order basis $\Phi \in \mathbb{R}^{n \times n_{\text{ssm}}}$ can be obtained from snapshots of a more complex model trajectories by proper orthogonal decomposition.

- The lifting operation (4.13) can be constructed using the observation operator. Specifically, assuming a small model error we have:

$$\hat{\Delta}_t = h(\mathbf{x}_t) - h(\mathbf{v}_t) \approx \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}_t) (\mathbf{x}_t - \mathbf{v}_t) = -\mathbf{H}_t \delta_t, \quad \text{where} \quad \mathbf{H}_t = \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}_t). \quad (4.14)$$

Let \mathbf{H}_t^+ be the pseudo-inverse of \mathbf{H}_t . From (4.14) and (4.12b) we have that:

$$\mathbf{x}_t - \mathbf{v}_t = -\delta_t \approx \mathbf{H}_t^+ \hat{\Delta}_t \approx \mathbf{H}_t^+ \mathbf{C} \zeta_t + \mathbf{H}_t^+ \mathbf{D} \hat{\mathbf{x}}_t \quad \Rightarrow \quad \mathbf{F} = \mathbf{H}_t^+ \mathbf{C} \quad \text{and} \quad \mathbf{G} = \mathbf{H}_t^+ \mathbf{D}. \quad (4.15)$$

Before the next time step the model forecast (4.10) is corrected with the predicted discrepancy (4.13) in order to obtain a better approximation of the true physical state:

$$\mathbf{x}_t = \hat{\mathbf{x}}_t + \delta_t. \quad (4.16)$$

4.2.2 Local models for structural uncertainty

In order to capture the global error dynamics the SSM (4.12) needs to be large. Finding the matrix coefficients from the inputs and outputs of this model is highly challenging. Moreover, many models have a dynamics driven by local dependencies among variables. The model structural errors due to local inaccuracies in the representation of physics have local impacts.

In order to address these points we consider local error models that seek to capture the structural uncertainty associated with only subsets of variables of the model. The assumption is that structural errors are associated with certain parts of the model, and that the effects of one type of structural error is observed locally in the evolution of “nearby” model variables and quantities of interest. By nearby we mean variables in physical proximity (e.g., grid points located at a short distance from each other) or variables that are tightly coupled to one another. Some prior knowledge about the source of structural errors may be required in order to define the local subset of variables affected by it. The interaction between local and remote variables is assumed to be sufficiently weak such that we can capture the evolution of errors using only local models: the errors depend on local variable states as inputs, and provide the discrepancies between the model forecasts and true physics only on for those observable quantities that measure local variables.

The large SSM (4.12) modeling the global error dynamics is split into a set of L small-dimensional “local” SSMs:

$$\zeta_t^\ell = \mathbf{A}^\ell \cdot \zeta_{t-1}^\ell + \mathbf{B}^\ell \cdot \widehat{\mathbf{x}}_t^\ell + \mathbf{K}^\ell \cdot \eta_t^\ell, \quad \zeta_0^\ell = \text{given}, \quad (4.17a)$$

$$\widehat{\Delta}_t^\ell = \mathbf{C}^\ell \cdot \zeta_t^\ell + \mathbf{D}^\ell \cdot \widehat{\mathbf{x}}_t^\ell + \boldsymbol{\varepsilon}_t^\ell, \quad (4.17b)$$

$$\boldsymbol{\delta}_t^\ell = \mathbf{F}^\ell \cdot \zeta_t^\ell + \mathbf{G}^\ell \cdot \widehat{\mathbf{x}}_t^\ell, \quad (4.17c)$$

$$\mathbf{x}_t^\ell := \widehat{\mathbf{x}}_t^\ell + \boldsymbol{\delta}_t^\ell, \quad \ell = 1, \dots, L. \quad (4.17d)$$

Here \mathbf{x}^ℓ consists of a local subset of model variables (e.g., associated with a certain geographic region), ζ_t^ℓ are the hidden states associated with the local model errors, and $\widehat{\Delta}_t^\ell$ are the observations associated with that particular geographic region. In the local approach the large model (4.12) is replaced by a set of small error models (4.17), one for each local set of variables that are affected by structural errors.

In absence of any prior knowledge regarding the location of potential sources of uncertainty one can iterate over subsets of variables of the model, assume the uncertainty is associated with those variables, and construct the local error SSM (4.17) using only the variables in the locality of the source of uncertainty. The state is corrected according to (4.17d). The difference between the observables associated with the corrected solution

(4.17d) and the real data allows to rank the different possible sources of error, and to select the ones that lead to the smallest differences. Algorithm 1 summarizes this approach.

Algorithm 1 Identifying the subsets of variables most affected by model uncertainty using local structural uncertainty modeling.

```

UncertaintyVars=Subset of variables in the model concern of having uncertainty.
For variable= UncertaintyVars[0]+L: UncertaintyVars[end]-L
     $\hat{\mathbf{x}}^\ell = \mathbf{x} \left( \text{variable} - \frac{L}{2} : \text{variable} + \frac{L}{2} \right)$ 
    Obtain  $\mathbf{x}^\ell$  using (4.17)
    rank(variable) =RMSE  $(y, \mathbf{x}^\ell)$ 
End For
[min,index]=minimum(rank)
return  $\mathbf{x}^\ell = \mathbf{x}(\text{index} - \frac{L}{2} : \text{index} + \frac{L}{2})$ 

```

4.3 Numerical experiments

In order to study the structural uncertainty and to illustrate the proposed modeling of structural errors we employ two test systems. The first one is the Lorenz-96 model with 40 variables [72], and the second model is the stratospheric chemistry model [73]. For our experiments we use the SSM library in Matlab [74] to obtain the state-space equations of the error models, i.e., the \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} matrices. We use the identity observation operator, i.e., we observe the errors in all model states. From (4.15) with $\mathbf{H}_t^+ = \mathbf{I}$ we have $\delta_t \approx \hat{\Delta}_t$.

4.3.1 Lorenz-96 model

The one dimensional Lorenz model is given by [72]:

$$\frac{dX_k}{dt} = -X_{k-2}X_{k-1} + X_{k-1}X_{k+1} - X_k + F, \quad k = 1, 2, \dots, K, \quad (4.18)$$

with $K = 40$ variables, periodic boundary conditions, and a forcing term $F = 8$. The variables of Lorenz model are periodic meaning that, values of X_{k-K} and X_{k+K} is equal to X_k .

A more complex Lorenz model includes two distinct time scales. It couples each variable X_k of the slow model (7.9) with a fast oscillating system described by the variables

$\{Y_{j,k}\}_{k=1,\dots,K,j=1,\dots,J}$. The two-scale Lorenz model (4.19) adds five fast variables ($J = 5$) to each slow variable in the basic Lorenz (7.9):

The equations are:

$$\begin{aligned} \frac{dX_k}{dt} &= X_{k-1}(X_{k+1} - X_{k-2}) - X_k - (hc/b) \sum_{j=1}^J Y_{j,k} \\ \frac{dY_{j,k}}{dt} &= -cbY_{j+1,k}(Y_{j+2,k} - Y_{j-1,k}) - cY_{j,k} + (hc/b)X_k, \\ k &= 1, 2, \dots, K, \quad j = 2, \dots, J. \end{aligned} \quad (4.19)$$

System (4.19) has 200 variables. We select the coupling coefficient value $h = 0.2$. The parameters $c = 1$ and $b = 10$ are chosen such that the convective (fast) scales oscillate ten times faster than the large (slow) scales.

We consider (4.19) to be the real physical system, and (7.9) to be the computer model approximation. The mapping of variables is $\mathbf{v}_t = \{X_k, Y_{j,k}\}_{1 \leq k \leq K, 1 \leq j \leq J}$, $\mathbf{x}_t = \{X_k\}_{1 \leq k \leq K}$, and $\xi(\mathbf{v}_t) = \{X_k\}_{1 \leq k \leq K}$. For simplicity, we use the identity observation operator $\mathcal{H}(\mathbf{x}_t) = \mathbf{x}_t$. Using the discrepancy between the model solution and the true physics we employ the global and the local modeling approaches to analyze the structure of uncertainty in the model.

Global structural uncertainty modeling results

For the global approach at each time step we observe the discrepancy between the model solution and the real data for all model variables (8.5). The model (7.9) is integrated forward in time starting from real data to obtain the model forecast $\hat{\mathbf{x}}_t$ (4.10). The SSM states ζ_t are predicted from past values, the observed discrepancies, and the model forecast. Before the next step is taken the model solution is corrected using the predicted global error according to (4.12a). This procedure is summarized by Algorithm 2.

In the first experiment we have perturbed the 12th and 30th variables in the Lorenz system, and kept all other slow variables without the additional forcing from the fast scales not captured by the model. This means that the structural uncertainty affects only two variables in the model. The system was integrated forward in time for 1000 time steps, and the number of error hidden states was empirically chosen equal to eight. Figure 4.1 shows the corrected model solution with reduced uncertainty. The SSM coefficient matrices are shown in Figure 4.2 when perturbing only the 12th and 30th variables in the Lorenz system. The output matrix \mathbf{C} identifies well the source of uncertainty: the larger entries correspond

Algorithm 2 Reducing the uncertainty in the Lorenz model using the global approach.

$\Delta_t = \widehat{\mathbf{x}}_t - \mathbf{v}_t, \quad t \in \{\text{training times}\}.$
 $[\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \zeta_0] = \text{SSM}(\{\Delta_t, \widehat{\mathbf{x}}_t\}_{t \in \{\text{training times}\}}).$
 $\mathbf{x}_0 = y_1$
 For $t = 1 : T$
 $\widehat{\mathbf{x}}_t = \mathcal{M}(\mathbf{x}_{t-1}, \hat{\theta})$
 $\zeta_t = \mathbf{A} \zeta_{t-1} + \mathbf{B} \widehat{\mathbf{x}}_t$
 $\Delta_t = \mathbf{C} \zeta_t + \mathbf{D} \widehat{\mathbf{x}}_t$
 $\mathbf{x}_t = \widehat{\mathbf{x}}_t + \Delta_t$
 End For

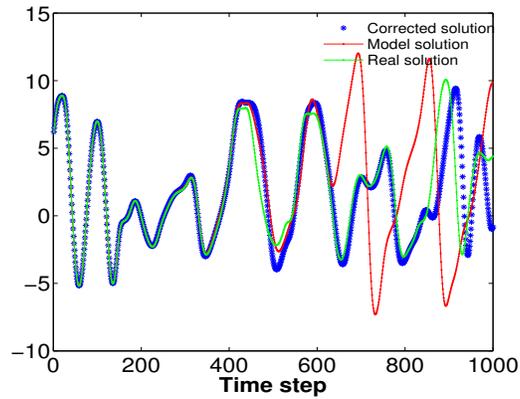


Figure 4.1: The evolution of a Lorenz variable in time for model solution, real solution and corrected model solution. The corrected model solution of the Lorenz system is obtained using the predicted global model error. The 12th and 30th variables in the model are perturbed, meaning that they are affected by the structural uncertainty.

to the 12th and 30th variables. The state matrix **A** has the largest entries near the diagonal, indicating that global error correlations are spatially localized. The input matrix **B** shows no special structure that reveals the uncertainty location. The feed forward matrix **D** also captures the perturbed states.

In the second experiment the structural uncertainty affects the 8th, 16th, 25th, and 33rd variables in the Lorenz system (7.9). Figure 4.3 shows the SSM coefficient matrices. The sensor matrix **C** again identifies well the perturbed variables, with entry peaks corresponding to the perturbed variables. The state matrix **A** has the largest entries near the diagonal, indicating that global error correlations are spatially localized. The input matrix **B** does not have any special structure. The feed forward matrix **D** also captures the perturbed states.

Furthermore, the length of the the time series used in SSM for tuning the parameters, affects the prediction power. The more data used, the parameters of SSM gets better tuned and the hidden dynamics of the error will be identified better. Table (1.a) illustrates the predictability power of the SSM using different length of time series. The variables we use when modeling with SSM, also impacts the accuracy to great extent. Table (1.b) shows the predictability power of the SSM using different number of variables.

(a) Effect of length of time-series on SSM performance		(b) Effect of using different variable numbers on SSM performance	
Length of time-series	RMSE	Number of variables	RMSE
1000	2.2449	40	2.2449
800	3.0075	20	3.5834
400	4.1406	10	5.2142

Table 4.1: RMSE between corrected model solution and real data

Local structural uncertainty modeling results

In the local approach all the assumptions made for the global approach also hold. The difference is that we construct the SSM representing the uncertainty in variable i using only a subset of variables in the physical vicinity of i , and do not include all variables in the system. Therefore, for local error modeling one needs to determine the size of the local neighborhood where errors are correlated with the error in variable i . In case of the Lorenz model (7.9) the physical distance between variables equal the difference between their indices, modulo 20 (due to the periodic boundary conditions). Therefore one needs

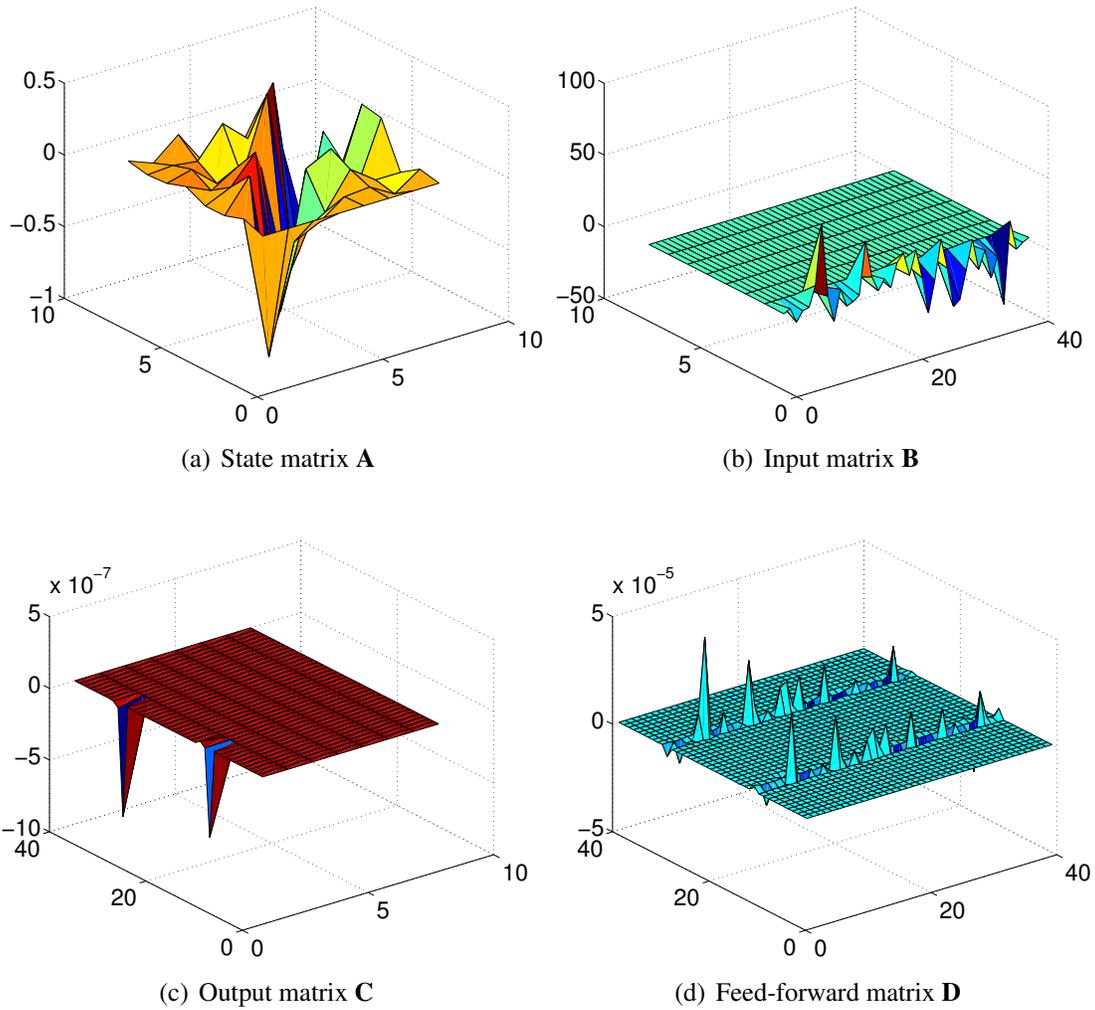


Figure 4.2: The structure of the SSM coefficient matrices for the global error model. The structural uncertainty affects the 12th and 30th variables in the Lorenz model (7.9).

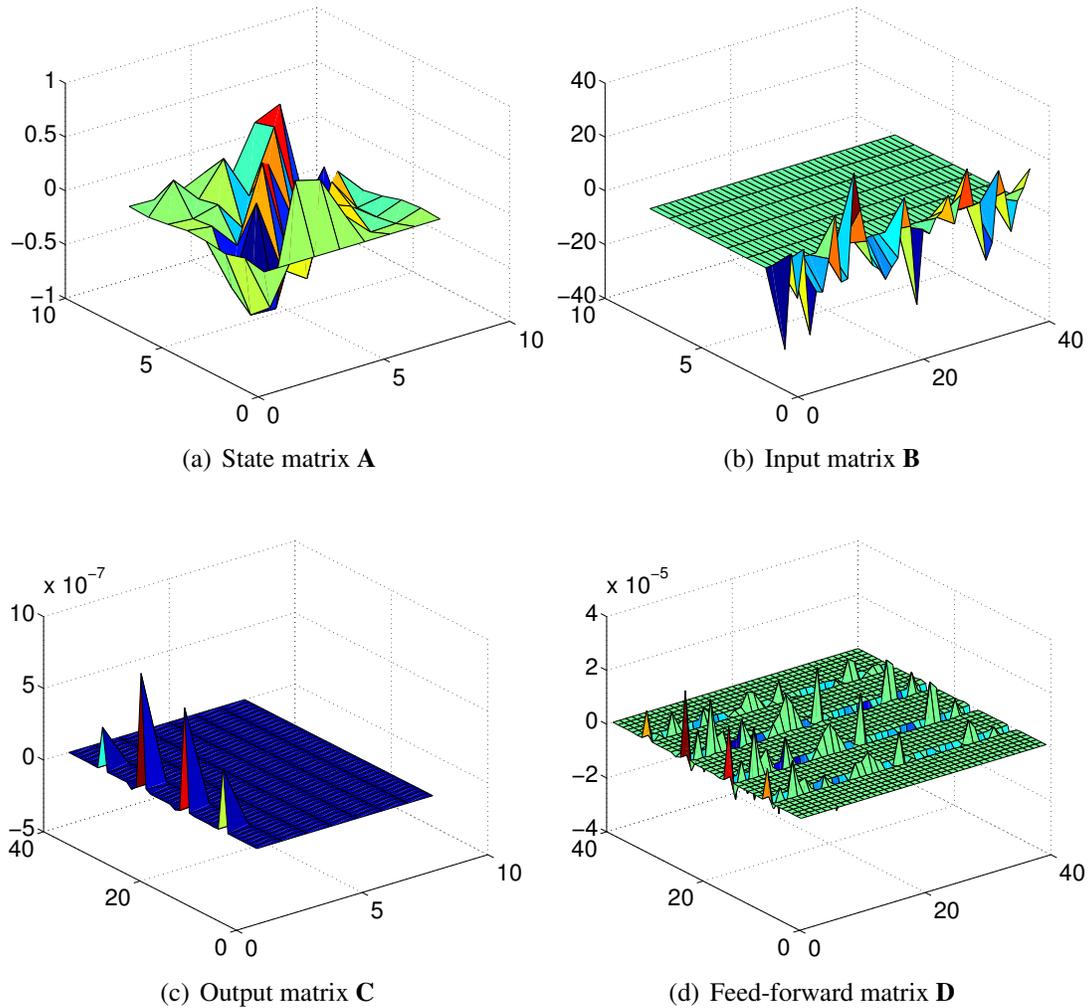


Figure 4.3: The structure of the SSM coefficient matrices for the global error model. The structural uncertainty affects the 8th, 16th, 25th and 33rd variables in the Lorenz model (7.9).

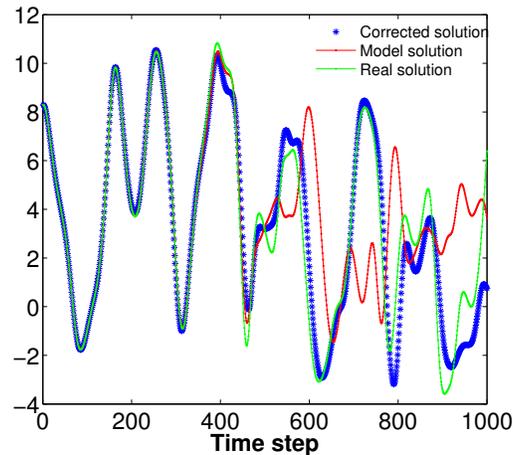


Figure 4.4: The evolution of a Lorenz variable in time for model solution, real solution and corrected model solution. The corrected model solution is obtained using the predicted local error. The structural uncertainty affects the 22nd variable in the model, and the localization radius is one.

to determine the decorrelation distance L such that the local error model for variable i uses information from variables indexed $i - L$ to $i + L$.

For the first set of experiments we set the localization radius to $L = 1$. The 22nd variable is perturbed, and we use the model predictions $\hat{\mathbf{x}}$ of variables 21, 22, 23 to build the local error SSM. The corrected model solution, obtained according to (4.17), is shown in Figure 4.4. Figure 4.5 shows the matrix coefficients for the local error SSM. Since we only included three variables in the SSM the dimension of the output matrix \mathbf{C} is 3×3 ; the largest entry corresponds to the perturbed variable. The state matrix \mathbf{A} is nearly the identity matrix, the input matrix \mathbf{B} does not have any special structure, and the feed-forward matrix \mathbf{D} is zero. However, the corrected solution is closer to the “truth”, as shown in Figure 4.4.

In the next experiment we set the localization radius to $L = 5$. Figure 4.6 shows the corrected solution using a local error SSM model with 11 variables. The corrected solution is much closer to the truth than the one obtained for $L = 1$, see also Figure 4.4. This implies that an accurate corrected model solution depends on a good choice of the localization radius; in this experiment including more variables is beneficial. Figure 4.7 shows the SSM matrix coefficients for this experiment. The sensor matrix \mathbf{C} has a peak in the middle variable corresponding to the perturbed variable. The structure of the rest of the matrices are very similar to the experiment where the localization radius is three.

However, in practice, the real source of uncertainty is unknown and we might not have

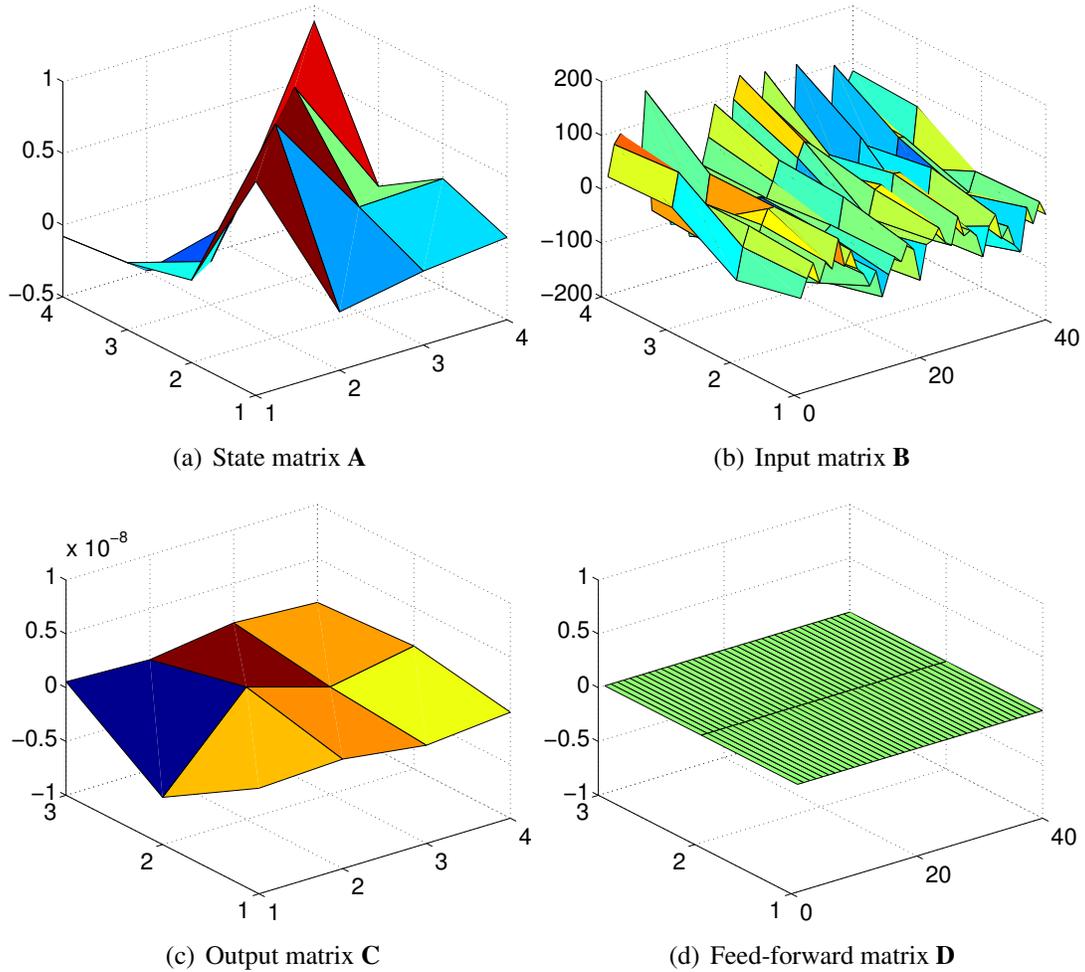


Figure 4.5: The local error SSM coefficients for the Lorenz system (7.9). The structural uncertainty affects the 22nd variable in the model, and the localization radius is one.

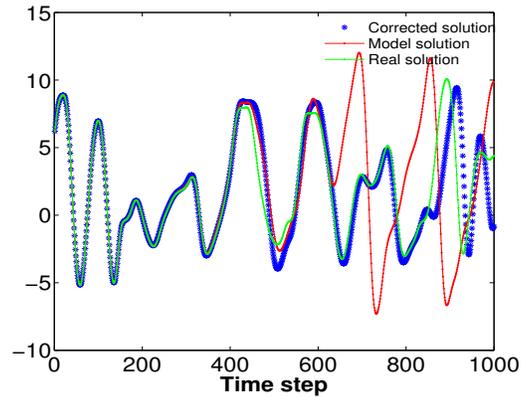


Figure 4.6: The evolution of a Lorenz variable in time for model solution, real solution and corrected model solution. The corrected model solution is obtained through applying the local model discrepancy. The structural uncertainty affects the 22nd variable in the model, and the localization radius is five.

sufficient knowledge about the states of the model that are affected by large structural uncertainties. One can find the most uncertain variables by iterating over subsets of model states, as summarized in Algorithm 1. At each iteration we assume that the source of uncertainty affects one particular subset of variables. A local error SSM is constructed, the corrected solution is computed, and the root mean square of the difference between the corrected solution and the real trajectory is found. The subsets of variables corresponding to the minimum root mean square differences are considered the most likely to be affected by structural uncertainty, since local corrections applied to those variables results in most improvements of the overall model predictions.

In the third experiment we perturb one variable in the Lorenz system. We iterate over all variables in the Lorenz system, and for each we apply the local approach with a given localization radius. The differences between the corrected solution and the real trajectory are obtained. We consider the root mean square errors between the vectors of solutions at each time moment along the trajectory. The histogram of minimum of the root mean square differences will show for which variable corrections result in the minimum error which helps identifying the potential source of uncertainty in the system. Figure 4.8 shows the histogram of root mean square differences between corrected solution and real data after perturbing the 30th variable. We iterate over all variables in the Lorenz system, and for each we apply the local approach with a localization radius of three. The frequency of the minimum of the discrepancy between model solution and real data for the 30th variable is higher than for other variables. In another experiment we perturb the 14th variable in

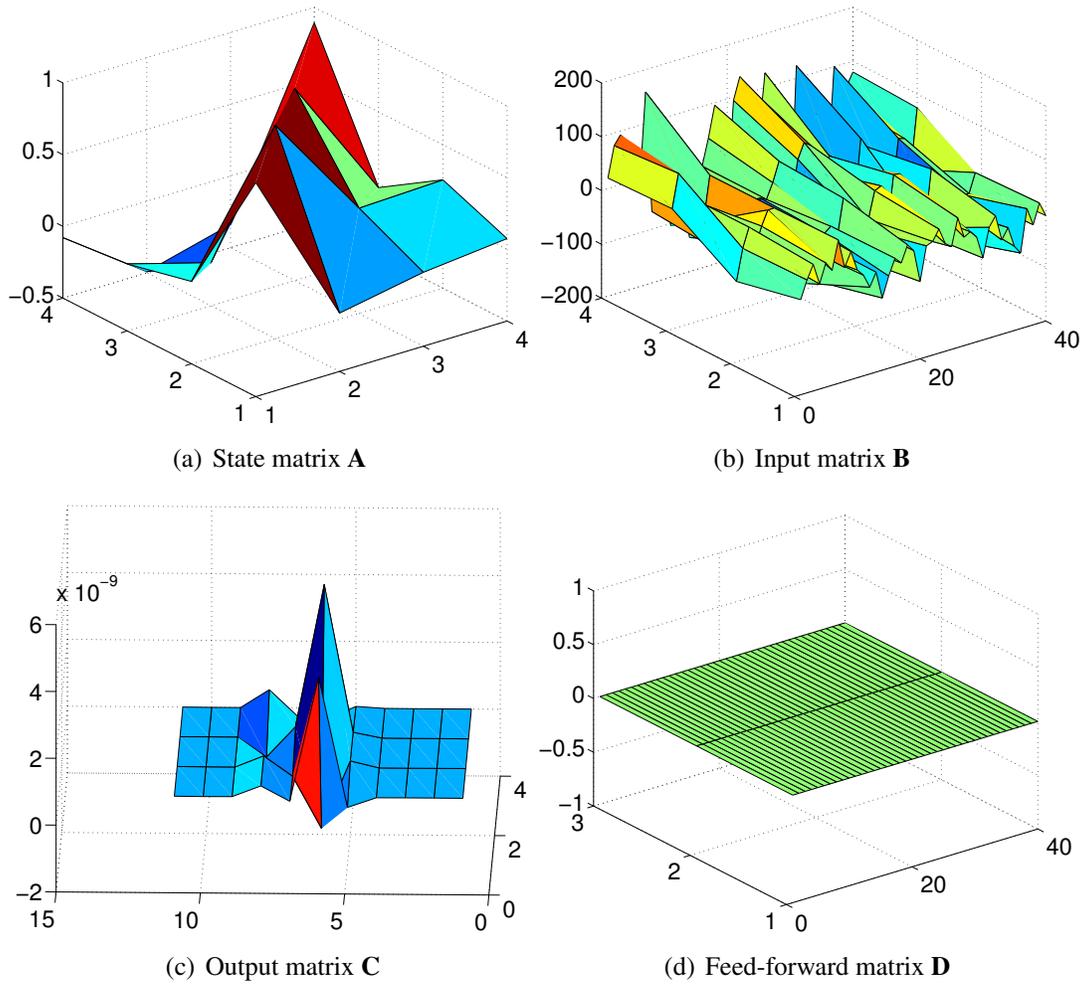


Figure 4.7: The local error SSM coefficients for the Lorenz system (7.9). The structural uncertainty affects the 22nd variable in the model, and the localization radius is five.

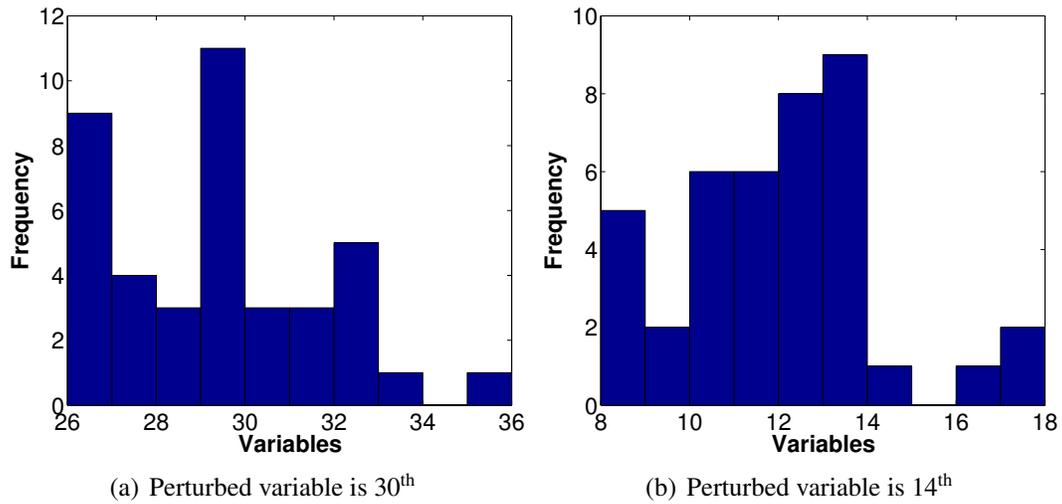
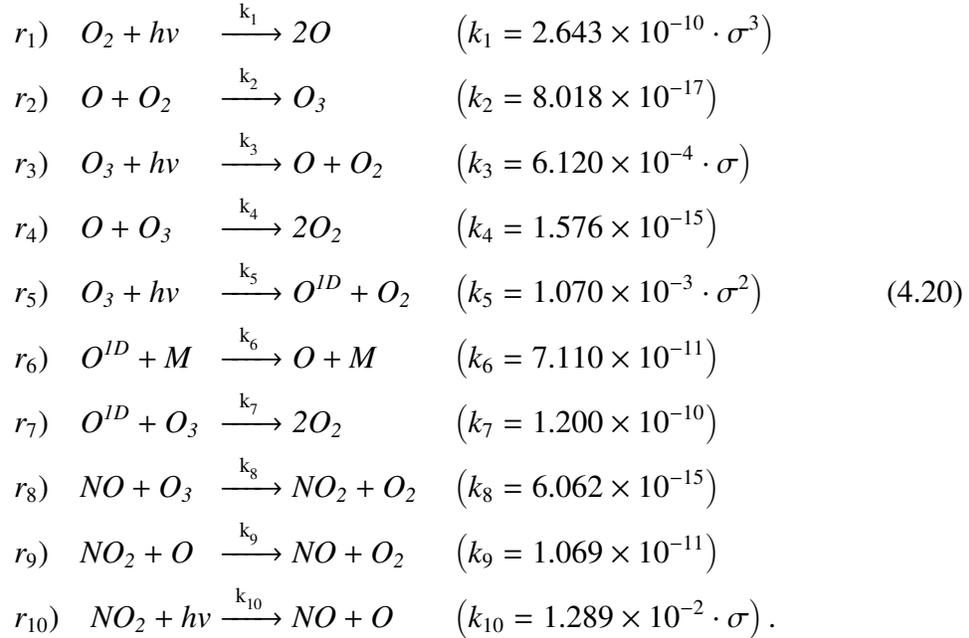


Figure 4.8: Histogram of minimum root mean square discrepancies between the corrected solution and the exact trajectory using the local error modeling approach.

the Lorenz system with the localization radius of five. Figure 4.8 shows the histogram of discrepancies perturbing the 14th variable; again, the frequency of the minimum of the root mean square differences between model solution and real data for the 14th variable is higher than for other variables, which indicates that the source of uncertainty is well identified.

4.3.2 Stratospheric chemistry model

A basic stratospheric chemistry mechanism [73] is given by the following set of reactions:



Here $M = 8.120E + 16$ molecules/cm³ is the atmospheric number density. The rate coefficients are scaled for time expressed in seconds, and $\sigma(t) \in [0, 1]$ represents the normalized sunlight intensity during a diurnal cycle [73].

In our experiments we consider the full set of equations (4.20) as describing the true physical system (8.6a). Our model (8.1a) is the set of reactions $r_1 \dots r_7$; not captured by the model are those reactions r_8, r_9, r_{10} that involve nitrous oxides (NO, NO_2). The three missing reactions correspond to missing dynamics in the model, i.e., physical phenomena that exist in reality but are not captured by the computer model. Our goal is to understand what is the missing dynamics in the chemical system that leads to systematic differences between the model and reality. In order to tackle this we construct a global error model (4.12) that estimates the global discrepancy between the system of full reactions and the system with three missing reactions. The assumptions made for the Lorenz model also hold here, i.e. we don't have observation error and disturbance, and we use the identity observation operator, therefore according to (4.15) we have $\delta_t \approx \hat{\Delta}_t$.

Figure 4.9 shows the solutions of the reference system, as well as the original and the corrected solutions of the model with missing physics. Correcting the solutions of the

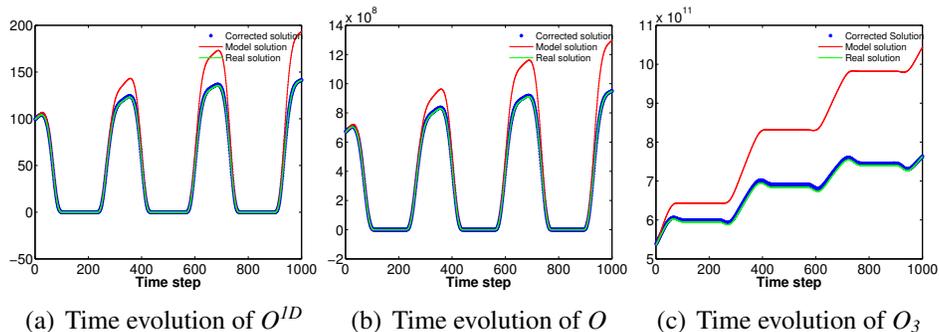


Figure 4.9: Model forecasts and corrected model solutions for the stratospheric chemical system with missing reactions. A global model of structural uncertainty is used.

model with missing dynamics allows to recover very well the trajectories of the physical (reference) model. Figure 4.10 shows the coefficient matrices of the global SSM for the stratospheric model. There is no particular structure associated with these matrices. While the linear SSM dynamics can mimic the evolution of structural errors in the nonlinear chemical system (4.20), it cannot explain the real source of discrepancies as coming from three missing reactions.

Having the corrected solution, one is interested to learn about the number of missing chemical reactions as well as their corresponding reaction rates; the missing chemical reactions represent the missing dynamics in the model. Assuming we know what the missing reactions are, the corresponding reaction rates can be obtained through an optimization process that adjusts their values such as to minimize the model-observation discrepancies. In this experiment we carry out the optimization using a genetic algorithm (GA). The reason for using the GA approach rather than a traditional gradient-based approach is that computing the Jacobian of this system is very expensive. GAs are a class of evolutionary algorithms that is used both for constrained and unconstrained optimization problems. A population of candidate solutions is generated randomly in the domain of search space and the fitness of each individual solution is evaluated using the value of the objective function in the optimization problem. In an iterative process the population of candidate solutions then evolves toward the best solution in the search space. At each iteration the properties of the candidate solutions are changed randomly by cross-over and mutation (operations inspired by biological evolution) to ensure the dispersion of the possible solutions throughout the search space. The best fitted solutions will be selected to form the next generation of feasible solutions. The algorithm is stopped when the maximum number of iterations is reached, or the highest fitness value was obtained, or successive iterations do not improve

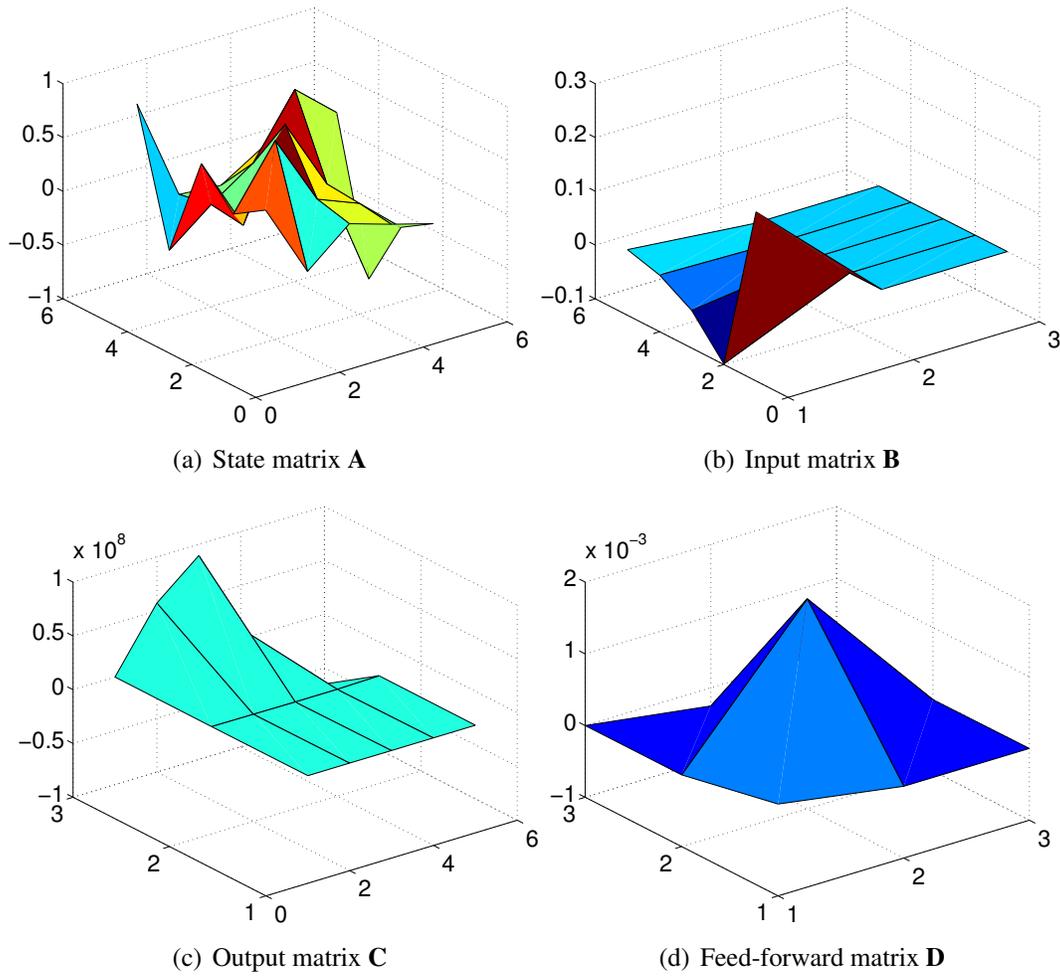


Figure 4.10: The structure of the matrix coefficients of the global-error SSM for the stratospheric chemical system.

the result any further [75].

For this experiment we use Matlab's GA toolbox with a population size of 30, 50 iterations, cross over rate of 0.85, and a mutation rate of 0.05. The fitness function is the root mean square difference between the true physics (solution of the stratospheric model with all reactions) and the solution of the model (stratospheric model with three reactions having uncertain reaction coefficients). With tight upper bounds and lower bounds for the reaction rates of the missing reactions, the optimization finds the reaction rate coefficients for r_8, r_9, r_{10} to have the values 7.781×10^{-15} , 1.937×10^{-11} , and 3.4621×10^{-2} , respectively. The absolute differences between the approximated values and the true values of the rate coefficients are small, 1.719×10^{-15} , 8.687×10^{-12} , and 0.021, respectively. This experiment illustrates the fact that any information we have about the source of structural uncertainty (e.g., three missing chemical reactions) can and should be used to obtain a more accurate, physically based description of the associated model errors.

4.4 Conclusions

Physics-based computer models of the real world are imperfect since not all physical processes that drive reality are fully captured by the model.

Type-B evaluation of measurement uncertainty [66] in the model results is due to both the numerical errors and the inaccurate representation of the real-world physical processes in the model. While numerical errors are relatively well studied, there is considerable need to quantify the structural model uncertainty, i.e., the model inadequacy due to missing aspects of the dynamics of the true physical system.

This work studies model structural uncertainty based on the information provided by the discrepancy between the model solution and the true state of the physical system, as measured by the available observations. The proposed approach is to approximate the dynamics of structural errors using linear state-space models. The parameter matrices of these models are obtained from fitting their response to match the mapping of model states as inputs to the observed model-reality differences as outputs.

Two different strategies to model the error dynamics are discussed: the global approach and the local approach. The global approach seeks to build one state-space model that approximates the entire structural uncertainty over the model's entire state-space. The local approach uses the ansatz that structural errors are only correlated locally, and that the correlation decreases with increasing distance between model components. Low-dimensional

local state-space models are constructed to approximate the dynamics of the local errors, and they use only information from subsets of variables and data.

Numerical experiments are carried out with two test problems, the Lorenz-96 system and a stratospheric chemistry model. These experiments reveal that the state-space discrepancy models provide intuition about the variables affected most by the missing dynamics. Global error models can identify the sources of uncertainty inside the physics-based models. Local error models capture the evolution of uncertainty over subsets of variables and are considerably less expensive to construct. When there is insufficient knowledge about the source of structural uncertainty the local approach allows to locate the subsets of variables that are most affected by it. The structural errors estimated with both global and local approaches can be used to correct the model solution and obtain improved estimates of the true physics. However, even if the state-space models can reproduce well the dynamics of the error, they may do this using an internal dynamics that is unrelated to the model dynamics or the true physics. In particular, reproducing the error dynamics does not necessarily explain the missing physics, as it was illustrated in the experiments with the stratospheric chemistry system. Consequently, in order to fully understand structural errors it is important to incorporate in the analysis all available information about the nature of the missing physical processes.

Chapter 5

Multivariate Predictions of Local Reduced-Order-Model errors and Dimensions

5.1 Introduction

Many physical phenomena are described mathematically by partial differential equations (PDEs), and, after applying suitable discretization schemes, are simulated on a computer. PDE-based models frequently require calibration and parameter tuning in order to provide realistic simulation results. Recent developments in the field of uncertainty quantification [4, 12, 76, 77] provide the necessary tools for validation of such models even in the context of variability and lack of knowledge of the input parameters. Techniques to propagate uncertainties through models include direct evaluation for linearly parametric models, sampling methods such as Monte Carlo [78], Latin hypercube [79] and quasi-Monte Carlo techniques [80], perturbation methods [1, 2, 3] and spectral representation [4, 5, 6]. While stochastic Galerkin methods [4] are intrusive in nature, Monte Carlo sampling methods [78] and stochastic collocations [5] do not require the modification of existing codes and hence they are non-intrusive. While uncertainty propagation techniques can measure the impact of uncertain parameters on some quantities of interest, they often become infeasible due to the large number of model realizations required. Similar difficulties are encountered when solving Bayesian inference problems since sampling from the posterior distribution is required.

The need for computational efficiency motivated the development of surrogate models such as response surfaces, low resolution, and reduced-order models. Data fitting or response surface models [12] are data-driven models. The underlying physics remain unknown and only the input-output behavior of the model is considered. Data fitting can use techniques such as regression, interpolation, radial basis function, Gaussian Processes, Artificial Neural Networks and other supervised machine-learning methods. The latter techniques can automatically detect patterns in data, and one can use them to predict future data under uncertainty in a probabilistic framework [81]. While easy to implement due to the non-intrusive nature, the prediction abilities may suffer since the governing physics are not specifically accounted for.

Low-fidelity models attempt to reduce the computational burden of the high-fidelity models by neglecting some of the physical aspects (e.g., replacing Navier-Stokes and Large Eddy Simulations with inviscid Euler's equations and Reynolds-Averaged Navier-Stokes [13, 14, 15], or decreasing the spatial resolution [16, 17]). The additional approximations, however, may considerably degrade the physical solution with only a modest decrease of the computational load.

Reduced basis [82, 83, 84, 85, 86] and Proper Orthogonal Decomposition [87, 88, 89, 90, 91] are two of the popular reduced-order modeling (ROM) strategies available in the literature. Data analysis is conducted to extract basis functions from experimental data or detailed simulations of high-dimensional systems (method of snapshots [92, 93, 94]), for subsequent use in Galerkin projections that yield low dimensional dynamical models. While these type of models are physics-based and therefore require intrusive implementations, they are usually more robust than data fitting and low-fidelity models. However, since surrogate model robustness depends heavily on the problem, it must be carefully analyzed especially for large-scale nonlinear dynamical systems.

Construction of ROM models in a parametric setting can be achieved by generating a global basis [18, 19], but this strategy generates large dimensional bases that may lead to slow reduced-order models. Local approaches have been designed for parametric or time domains generating local bases for both the state variables [20, 21] and non-linear terms [22, 23]. A recent survey of state-of-the-art methods in projection-based parametric model reduction is available in [95].

In this study, we propose multivariate data fitting models to predict the local parametric Proper Orthogonal Decomposition reduced-order models errors and bases dimensions. We refer to them as MP-LROM models. Let us consider a local parametric reduced-order model of dimension K_{POD} constructed using a high-fidelity solution associated with the parameter configuration μ_p .

Our first MP-LROM model consists in the mapping $\{\mu, \mu_p, K_{POD}\} \mapsto \log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$, where $\varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ is the error of the local reduced-order model solution with respect to the high-fidelity solution for a viscosity parameter configuration μ . Our proposed approach is inspired from the multi-fidelity correction (MFC) [96] and reduced order model error surrogates method (ROMES) [97]. MFC [96, 98, 13, 99] has been developed for low-fidelity models in the context of optimization. The MFC model simulates the input-output relation $\mu \mapsto \varepsilon_{\mu}^{HF}$, where ε_{μ}^{HF} is the low-fidelity model error depending on a global reduced basis with a constant reduced-order model dimension. The ROMES method [97] introduced the concept of error indicators for global reduced-order models and generalized the MFC framework by approximating the mapping $\rho(\mu) \mapsto \log \varepsilon_{\mu}^{HF}$. The error indicators $\rho(\mu)$ include rigorous error bounds and reduced-order residual norms. No variation of the reduced basis dimension was taken into account. By estimating the log of the reduced-order model error instead of the error itself, the input-output map exhibits a lower variance as shown by our numerical experiments as well as those in [97].

The second proposed MP-LROM model addresses the issue of a-priori selection of the reduced basis dimension for a prescribed accuracy of the reduced solution. The standard approach is to analyze the spectrum of the snapshots matrix, and use the largest singular value removed from the expansion to estimate the accuracy level [100]. To also take into account the error due to the full-order-model equations projection in the reduced space, here we propose the mapping $\{\mu_p, \log \varepsilon_{\mu_p, \mu_p, K_{POD}}^{HF}\} \mapsto K_{POD}$ to predict the dimension of a local parametric reduced-order model given a prescribed error threshold.

To approximate the mappings $\{\mu, \mu_p, K_{POD}\} \mapsto \log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ and $\{\mu_p, \log \varepsilon_{\mu_p, \mu_p, K_{POD}}^{HF}\} \mapsto K_{POD}$, we propose regression models constructed using Gaussian Processes (GP) [101, 102] and Artificial Neural Networks (ANN). In the case of one dimensional Burgers model, the resulted MP-LROM error models are accurate and their predictions are compared against those obtained by the MFC and ROMES models. The predicted dimensions of local reduced-order models using our proposed MP-LROM models are more accurate than those derived using the standard method based on the spectrum of snapshots matrix.

The remainder of the chapter is organized as follows. Section 5.3 reviews the reduced-order modeling parametric framework. The MP-LROM models and the regression machine learning methods used in this study to approximate the MP-LROM mappings are described in details in Section 5.3. Section 5.4 describes the viscous 1D-Burgers model and compares the performances of the MP-LROM and state of the art models. Conclusions are drawn in Section 5.5.

5.2 Parametric reduced-order modeling

Proper Orthogonal Decomposition has been successfully applied in numerous applications such as compressible flow [103] and computational fluid dynamics [104, 105, 106], to mention a few. It can be thought of as a Galerkin approximation in the state variable built from functions corresponding to the solution of the physical system at specified time instances. A system reduction strategy for Galerkin models of fluid flows based on a partition in slow, dominant, and fast modes, has been proposed in [107]. Closure models and stabilization strategies for POD of turbulent flows have been investigated in [108, 109].

In this study, we consider discrete inner products (Euclidean dot product), though continuous products may be employed as well. Generally, an unsteady problem can be written in semi-discrete form as an initial value problem; i.e., as a system of nonlinear ordinary differential equations

$$\frac{d\mathbf{x}(\mu, t)}{dt} = \mathbf{F}(\mathbf{x}, t, \mu), \quad \mathbf{x}(\mu, 0) = \mathbf{x}_0 \in \mathbb{R}^{N_{\text{state}}}, \quad \mu \in \mathcal{P}. \quad (5.1)$$

The input-parameter μ typically characterizes the physical properties of the flow. By \mathcal{P} we denote the input-parameter space. For a given parameter configuration μ_p we select an ensemble of N_t time instances of the flow $\mathbf{x}(\mu_p, t_1), \dots, \mathbf{x}(\mu_p, t_{N_t}) \in \mathbb{R}^{N_{\text{state}}}$, where N_{state} is the total number of discrete model variables, and $N_t \in \mathbb{N}^*$. The POD method chooses an orthonormal basis $U_{\mu_p} = [\mathbf{u}_1^{\mu_p} \ \dots \ \mathbf{u}_{K_{\text{POD}}}^{\mu_p}] \in \mathbb{R}^{N_{\text{state}} \times K_{\text{POD}}}$, such that the mean square error between $\mathbf{x}(\mu_p, t_i)$ and the POD expansion $\mathbf{x}_{\mu_p}^{\text{POD}}(t_i) = U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu, t_i)$, $\tilde{\mathbf{x}}_{\mu_p}(\mu, t_i) = U_{\mu_p}^T \mathbf{x}(\mu_p, t_i) \in \mathbb{R}^{K_{\text{POD}}}$, is minimized on average. The POD space dimension $K_{\text{POD}} \ll N_{\text{state}}$ is appropriately chosen to capture the dynamics of the flow. Algorithm 3 describes the reduced-order basis construction procedure [110]. The choice of the model order is usually done based on the desired precision of the reduced-order model. In practice selecting $\gamma = 0.99$ usually leads to reduced-order models solutions satisfying the required precision. Goal oriented approaches such as selecting the basis dimension to enhance the accuracy of quantities of interests depending on reduced-order model solutions could also be designed.

Next, a Galerkin projection of the full model state (5.1) onto the space $\mathcal{X}^{K_{\text{POD}}}$ spanned by the POD basis elements is used to obtain the reduced-order model

$$\frac{d\tilde{\mathbf{x}}_{\mu_p}(\mu, t)}{dt} = U_{\mu_p}^T \mathbf{F}\left(U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu, t), t, \mu\right), \quad \tilde{\mathbf{x}}_{\mu_p}(\mu, 0) = U_{\mu_p}^T \mathbf{x}_0. \quad (5.2)$$

The notation $\tilde{\mathbf{x}}_{\mu_p}(\mu, t)$ expresses the solution dependence on the varying parameter μ and also on μ_p the configuration whose associated high-fidelity trajectory was employed to generate the POD basis. While being accurate for $\mu = \mu_p$, the reduced model (5.2) may

Algorithm 3 POD basis construction

-
- 1: Compute the singular value decomposition for the snapshots matrix $[\mathbf{x}(\mu_p, t_1) \cdots \mathbf{x}(\mu_p, t_{N_t})] = \bar{U}_{\mu_p} \Sigma_{\mu_p} \bar{V}_{\mu_p}^T$, with the singular vectors matrix $\bar{U}_{\mu_p} = [\mathbf{u}_1^{\mu_p} \cdots \mathbf{u}_{N_t}^{\mu_p}]$.
 - 2: Using the singular-values $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{N_t} \geq 0$ stored in the diagonal matrix Σ_{μ_p} , define $I(m) = \sum_{i=1}^m \lambda_i^2 / (\sum_{i=1}^{N_t} \lambda_i^2)$.
 - 3: Choose K_{POD} , the dimension of the POD basis, such that $K_{POD} = \arg \min_m \{I(m) : I(m) \geq \gamma\}$ where $0 \leq \gamma \leq 1$ is the percentage of total information captured by the reduced space $\mathcal{X}^{K_{POD}} = \text{range}(U_{\mu_p})$. It is common to select $\gamma = 0.99$. The basis U_{μ_p} consists of the first K_{POD} columns of \bar{U}_{μ_p} .
-

lose accuracy when moving away from the initial setting. Several strategies have been proposed to derive a basis that spans the entire parameter space. These include the reduced basis method combined with the use of error estimates [85, 111, 19], global POD [112, 113], Krylov-based sampling methods [114, 115], and greedy techniques [116, 117]. The fundamental assumption used by these approaches is that a smooth low-dimensional global manifold characterizes the model solutions over the entire parameter domain. The purpose of our study is to estimate the solution error and dimension of the reduced-order model (5.2) that can be subsequently used to generate a global basis for the parameter space.

5.3 Multivariate prediction of local reduced-order models characteristics (MP-LROM)

We propose multivariate input-output models

$$\phi : \mathbf{z} \mapsto y, \quad (5.3)$$

$\mathbf{z} \in \mathbb{R}^r$, to predict characteristics $y \in \mathbb{R}$ of local parametric reduced-order models (5.2).

5.3.1 Error Model

Inspired from the MFC and ROMES methodologies we introduce an input-output model to predict the level of error $\varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$, where

$$\varepsilon_{\mu, \mu_p, K_{POD}}^{HF} = \|\mathbf{x}(\mu, t_1) - U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu, t_1) \quad \mathbf{x}(\mu, t_2) - U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu, t_2) \quad \cdots \quad \mathbf{x}(\mu, t_{N_t}) - U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu, t_{N_t})\|_F. \quad (5.4)$$

Here $\|\cdot\|_F$ denotes the Frobenius norm, and K_{POD} is the dimension of the reduced-order model. In contrast with ROMES and MFC models that predict the error of global reduced-order models with fixed dimensions, using univariate functions, here we propose a multivariate model

$$\phi_{MP-LROM}^e : \{\mu, \mu_p, K_{POD}\} \mapsto \log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF} \quad (5.5)$$

to predict the error of local parametric reduced-order models (5.2) of various dimensions. Since the dimension of basis usually influences the level of error we include it among the input variables. To design models with reduced variances we look to approximate the logarithm of the error as suggested in [97].

For high-dimensional parametric spaces, ROMES method handles well the curse of dimensionality with their proposing univariate models. In combination with active subspace method [118], we can reduce the number of input variables in case the amount of variability in the parametric space is mild. This will increase our error model feasibility even for high-dimensional parametric space.

5.3.2 Dimension of the reduced basis

The basis dimension represents one of the most important characteristic of a reduced-order model. The reduced manifold dimension directly affects both the on-line computational complexity of the reduced-order model and its accuracy [119, 120, 121]. By increasing the dimension of the basis, the projection error usually decreases and the accuracy of the reduced-order model is enhanced. However this is not necessarily valid as seen in [122, Section 5]. Nevertheless the spectrum of the snapshots matrix offers guidance regarding the choice of the reduced basis dimension when some prescribed reduced-order model error is desired. However the accuracy depends also on the ‘in-plane’ error, which is due to the fact that the full-order-model equations are projected on the reduced subspace [123, 124].

We seek to predict the dimension of the local parametric reduced-order model (5.2) by

accounting for both the orthogonal projection error onto the subspace, which is computable by the sum of squares of singular values, and the ‘in-plane’ error. As such we propose to model the mapping

$$\phi_{MP-LROM}^d : \{\mu_p, \log \varepsilon_{\mu_p, \mu_p, K_{POD}}^{HF}\} \mapsto K_{POD}. \quad (5.6)$$

Once such model is available, given a positive threshold $\bar{\varepsilon}$ and a parametric configuration μ_p , we will be able to predict the dimension K_{POD} of the basis U_{μ_p} , such that the reduced-order model error satisfies

$$\|\mathbf{x}(\mu_p, t_1) - U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu_p, t_1) \quad \mathbf{x}(\mu_p, t_2) - U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu_p, t_2) \quad \cdots \quad \mathbf{x}(\mu_p, t_{N_t}) - U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu_p, t_{N_t})\|_F \approx \bar{\varepsilon}. \quad (5.7)$$

5.3.3 Supervised Machine Learning Techniques

In order to estimate the level of reduced-order model solution error $\varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ (5.4) and the reduced basis dimension K_{POD} , we will use regression machine learning methods to approximate the maps $\phi_{MP-LROM}^e$ and $\phi_{MP-LROM}^d$ described in (5.5) and (5.6).

Artificial Neural Networks and Gaussian Processes are used to build a probabilistic model $\phi : \mathbf{z} \mapsto \hat{y}$, where ϕ is a transformation function that learns through the input features \mathbf{z} to estimate the deterministic output y [81]. As such, these probabilistic models are approximations of the mappings introduced in (5.3). The input features \mathbf{z} can be either categorical or ordinal. The real-valued random variable \hat{y} is expected to have a low variance and reduced bias. The features of \mathbf{z} should be descriptive of the underlying problem at hand [125]. The accuracy and stability of estimations are assessed using the K -fold cross-validation technique. The samples are split into K subsets (“folds”), where typically $3 \leq K \leq 10$. The model is trained on $K - 1$ sets and tested on the K -th set in a round-robin fashion [81]. Each fold induces a specific error quantified as the average of the absolute values of the differences between the predicted and the K -th set values

$$E_{\text{fold}} = \frac{\sum_{i=1}^N |\hat{y}^i - y^i|}{N}, \quad \text{VAR}_{\text{fold}} = \frac{\sum_{i=1}^N (\hat{y}^i - E_{\text{fold}})^2}{N - 1}, \quad \text{fold} = 1, 2, \dots, K, \quad (5.8a)$$

where N is the number of test samples in the fold. The error is then averaged over all folds:

$$E = \frac{\sum_{\text{fold}=1}^K E_{\text{fold}}}{K}, \quad \text{VAR} = \frac{\sum_{\text{fold}=1}^K (E_{\text{fold}} - E)^2}{K - 1}. \quad (5.8b)$$

The variance of the prediction results (5.8a) accounts for the sensitivity of the model to the particular choice of data set. It quantifies the stability of the model in response to

the new training samples. A smaller variance indicates more stable predictions, however, this sometimes translates into a larger bias of the model. Models with small variance and high bias make strong assumptions about the data and tend to underfit the truth, while models with high variance and low bias tend to overfit the truth [126]. The trade-off between bias and variance in learning algorithms is usually controlled via techniques such as regularization or bagging and boosting [125].

In what follows we briefly review the Gaussian Process and Artificial Neural Networks techniques.

Gaussian process kernel method

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [127]. A Gaussian process is fully described by its mean and covariance functions

$$\phi(\mathbf{z}) \sim \text{gp}(m(\mathbf{z}), \mathbf{K}), \quad (5.9)$$

where $m(\mathbf{z}) = \mathbb{E}[\phi(\mathbf{z})]$, and \mathbf{K} is the covariance matrix with entries $K_{i,j} = \mathbb{E}[(\phi(\mathbf{z}^i) - m(\mathbf{z}^i))(\phi(\mathbf{z}^j) - m(\mathbf{z}^j))]$ [127].

In this work we employ the commonly used squared-exponential-covariance Gaussian kernel with

$$k : \mathbb{R}^r \times \mathbb{R}^r \rightarrow \mathbb{R}, \quad k(\mathbf{z}^i, \mathbf{z}^j) = \sigma_\phi^2 \exp\left(-\frac{\|\mathbf{z}^i - \mathbf{z}^j\|}{2\hbar^2}\right) + \sigma_n^2 \delta_{i,j}, \quad (5.10)$$

and $K_{ij} = k(\mathbf{z}^i, \mathbf{z}^j)$ [127], where \mathbf{z}^i and \mathbf{z}^j are the pairs of data points in training or test samples, δ is the Kronecker delta symbol and $\|\cdot\|$ is some appropriate norm. The model (5.10) has three hyper-parameters. The length-scale \hbar governs the correlation among data points. The signal variance $\sigma_\phi^2 \in \mathbb{R}$ and the noise variance $\sigma_n^2 \in \mathbb{R}$ govern the precision of variance and noise, respectively.

Consider a set of training data points $\mathbf{Z} = [\mathbf{z}^1 \ \mathbf{z}^2 \ \dots \ \mathbf{z}^n] \in \mathbb{R}^{r \times n}$ and the corresponding noisy observations $\mathbf{y} = [y^1 \ y^2 \ \dots \ y^n] \in \mathbb{R}^{1 \times n}$,

$$y^i = \phi(\mathbf{z}^i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_n^2), \quad i = 1, \dots, n. \quad (5.11)$$

Consider also the set of test points $\mathbf{Z}^* = [\mathbf{z}^{*1} \ \mathbf{z}^{*2} \ \dots \ \mathbf{z}^{*m}] \in \mathbb{R}^{r \times m}$ and the predictions

$$\hat{\mathbf{y}} = [\hat{y}^1 \ \hat{y}^2 \ \dots \ \hat{y}^m] \in \mathbb{R}^{1 \times m},$$

$$\hat{y}^i = \phi(\mathbf{z}^{*i}), \quad i = 1, \dots, m. \quad (5.12)$$

For a Gaussian prior the joint distribution of training outputs \mathbf{y} and test outputs $\hat{\mathbf{y}}$ is

$$\begin{bmatrix} \mathbf{y}^T \\ \hat{\mathbf{y}}^T \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mathbf{m}(\mathbf{Z})^T \\ \mathbf{m}(\mathbf{Z}^*)^T \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}^* \\ \mathbf{K}^{*T} & \mathbf{K}^{**} \end{bmatrix} \right), \quad (5.13)$$

where

$$\mathbf{m}(\mathbf{Z}) = [m(\mathbf{z}^1) \ m(\mathbf{z}^2) \ \dots \ m(\mathbf{z}^n)] \in \mathbb{R}^{1 \times n}, \quad \mathbf{m}(\mathbf{Z}^*) = [m(\mathbf{z}^{*1}) \ m(\mathbf{z}^{*2}) \ \dots \ m(\mathbf{z}^{*m})] \in \mathbb{R}^{1 \times m},$$

$$\mathbf{K}^* = (K_{ij}^*)_{i=1, \dots, n; j=1, \dots, m} = k(\mathbf{z}^i, \mathbf{z}^{j*}) \text{ and } \mathbf{K}^{**} = (K_{ij}^{**})_{i=1, \dots, m; j=1, \dots, m} = k(\mathbf{z}^{i*}, \mathbf{z}^{j*}).$$

The predictive distribution represents the posterior after observing the data [125] and is given by

$$p(\hat{\mathbf{y}}|\mathbf{Z}, \mathbf{y}, \mathbf{Z}^*) \sim \mathcal{N} \left(\mathbf{K}^{*T} \mathbf{K}^{-1} \mathbf{y}, \mathbf{K}^{**} - \mathbf{K}^{*T} \mathbf{K}^{-1} \mathbf{K}^* \right), \quad (5.14)$$

where superscript T denotes the transpose operation.

The prediction of Gaussian process will depend on the choice of the mean and covariance functions, and on their hyper parameters \hbar , σ_ϕ^2 and σ_n^2 which can be inferred from the data

$$\boldsymbol{\theta}^* = [\hbar, \sigma_\phi^2, \sigma_n^2] = \arg \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}),$$

by minimizing the marginal negative log-likelihood function

$$L(\boldsymbol{\theta}) = -\log p(\mathbf{y}|\mathbf{Z}, \boldsymbol{\theta}) = \frac{1}{2} \log \det(\mathbf{K}) + \frac{1}{2} (\mathbf{y} - \mathbf{m}(\mathbf{Z})) \mathbf{K}^{-1} (\mathbf{y} - \mathbf{m}(\mathbf{Z}))^T + \frac{n}{2} \log(2\pi).$$

Artificial Neural Networks

The study of Artificial Neural Networks begins in the 1910s in order to imitate human brain's biological structure. Pioneering work was carried out by Rosenblatt, who proposed a three-layered network structure, the perceptron [128]. ANN detect the pattern of data by discovering the input-output relationships. Applications include the approximation of functions, regression analysis, time series prediction, pattern recognition, and speech synthesis and recognition [129, 130].

ANN consist of neurons and connections between the neurons (weights). Neurons are organized in layers, where at least three layers of neurons (an input layer, a hidden layer, and an output layer) are required for construction of a neural network.

In ANN, having multiple layers leads to a more generalized model. However, by increasing the size of the network, a larger number of weights is required increasing the chance of over-fitting. All learning algorithms are prone to under-fitting and over-fitting. Under-fitting occurs when the learning model performs poorly on the training data, because it can not capture the relationship between the input and output data. On the other hand, over-fitting occurs when the learning model performs well on the training data but it doesn't perform well on the new data. This is mostly because the learning model has memorized the training examples but has not learned to generalize for the new data and hence it is not able to generalize sufficiently. Consequently the model performs poorly on a different data set. Usually if enough samples are present in the data set then there is no concern for over-fitting. K-fold cross-validation, adding regularization parameter and drop out [131] are also very common techniques to prevent over-fitting.

The input layer distributes input signals $\mathbf{z} = [z_1 \ z_2 \ \cdots \ z_r]$ to the first hidden layer. For a neural network with L hidden layers and m^ℓ neurons in each hidden layer, let $\hat{\mathbf{y}}^\ell = [\hat{y}_1^\ell \ \hat{y}_2^\ell \ \cdots \ \hat{y}_{m^\ell}^\ell]$ be the vector of outputs from layer ℓ , $\mathbf{b}^\ell = [b_1^\ell \ b_2^\ell \ \cdots \ b_{m^\ell}^\ell]$ the biases at layer ℓ , and $\mathbf{w}_j^\ell = [w_{j_1}^\ell \ w_{j_2}^\ell \ \cdots \ w_{j_{m^\ell}}^\ell]$ the weights connecting the neuron j to the input of that layer (output of previous layer). The vectors $\hat{\mathbf{y}}^\ell$ and \mathbf{w}_j^ℓ share the same dimension which varies along the layers depending on the number of input features, neurons and outputs. Then the feed-forward operation is

$$\begin{aligned} x_j^{\ell+1} &= \mathbf{w}_j^{\ell+1T} \hat{\mathbf{y}}^\ell + b_j^{\ell+1}, \quad \hat{\mathbf{y}}^0 = \mathbf{z}, \quad j = 1, \dots, m^\ell. \\ \hat{y}_j^{\ell+1} &= \varphi(\mathbf{x}^{\ell+1}), \quad \ell = 0, 1, \dots, L-1. \end{aligned}$$

All products of previous layer output with current layer neuron weights will be summed and the bias value of each neuron will be added to obtain the vector $\mathbf{x}^\ell = [x_1^\ell \ x_2^\ell \ \cdots \ x_{m^\ell}^\ell]$. Then the final output of each layer will be obtained by passing the vector \mathbf{x}^ℓ through the transfer function φ , which is a differentiable function and can be log-sigmoid, hyperbolic tangent sigmoid, or linear transfer function.

The training process of ANN adjusts the weights and the biases in order to reproduce the desired outputs when fed the given inputs. The training process via the back propagation algorithm [132] uses a gradient descent method to modify weights and thresholds such that the error between the desired output and the output signal of the network is minimized [133]. In supervised learning the network is provided with samples from which it discovers the relations of inputs and outputs. The output of the network is compared with the desired

output, and the error is back-propagated through the network and the weights will be adjusted. This process is repeated during several iterations, until the network output is close to the desired output [134].

5.4 Numerical experiments

We illustrate the application of the proposed MP-LROM models to predict the error and dimension of the local parametric reduced-order models for a one-dimensional Burgers model. The 1D-Burgers model proposed herein is characterized by the viscosity coefficient. To assess the performance of the MP-LROM models constructed using Gaussian Process and Artificial Neural Networks, we employ various cross-validation tests. The dimensions of the training and testing data sets are chosen empirically. For Artificial Neural Networks models the number of hidden layers and neurons in each hidden layer vary for each type of problem under study. The squared-exponential-covariance kernel (5.10) is used for Gaussian Process models.

As a guidance, we provide a short summary describing the performed numerical experiments. Subsection 5.4.1 gives details of the Burgers model including the discretized versions of the high-fidelity and reduced-order models. Section 5.4.2 covers the MP-LROM error model construction, first explaining the generation of the data set. Then the importance of using large enough data set as well as normalizing the data set is explained. Tables 5.1 and 5.2 and Figures 5.4, 5.5 and 5.6 show the averages and variances of errors in prediction of Gaussian Process and Neural Network MP-LROM models for different sample sizes and scalings. Next, the five-fold cross-validation is used to assess the MP-LROM error models and the results are shown in Table 5.3. The approximated MP-LROM error models are then compared against the ROMES and multi-fidelity correction models, as illustrated in the Tables 5.4, 5.5, 5.6 and 5.7 and Figures 5.7, 5.8, 5.9, 5.10, and 5.11.

Section 5.4.2 covers the MP-LROM model for predicting the basis dimension of reduced-order models. Different data sets are used to generate Gaussian Process and Neural Network MP-LROM models and the prediction errors and their variances are shown in Table 5.8 and Figures 5.13 and 5.14. The five-fold cross-validation is used to assess the proposed models and the results are presented in Table 5.9. Finally the MP-LROM models are verified against the standard approach based on the spectrum of snapshots matrix as seen in Figure 5.15.

5.4.1 One-dimensional Burgers' equation

Burgers' equation is an important partial differential equation from fluid mechanics [135]. The evolution of the velocity u of a fluid evolves according to

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \mu \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, L], \quad t \in (0, t_f], \quad (5.15)$$

with $t_f = 1$ and $L = 1$. Here μ is the viscosity coefficient.

The model has homogeneous Dirichlet boundary conditions $u(0, t) = u(L, t) = 0, t \in (0, t_f]$. For the initial conditions, we used a seventh order polynomial constructed using the least-square method and the data set $\{(0, 0); (0.2, 1); (0.4, 0.5); (0.6, 1); (0.8, 0.2); (0.9, 0.1); (0.95, 0.05); (1, 0)\}$. We employed the polyfit function in Matlab and the polynomial is shown in Figure 5.1.

The discretization uses a spatial mesh of N_s equidistant points on $[0, L]$, with $\Delta x = L/(N_s - 1)$. A uniform temporal mesh with N_t points covers the interval $[0, t_f]$, with $\Delta t = t_f/(N_t - 1)$. The discrete velocity vector is $\mathbf{u}(t_j) \approx [u(x_i, t_j)]_{i=1,2,\dots,N_{\text{state}}} \in \mathbb{R}^{N_{\text{state}}}$, $j = 1, 2, \dots, N_t$, where $N_{\text{state}} = N_s - 2$ (the known boundaries are removed). The semi-discrete version of the model (5.15) is

$$\mathbf{u}' = -\mathbf{u} \odot A_x \mathbf{u} + \mu A_{xx} \mathbf{u}, \quad (5.16)$$

where \mathbf{u}' is the time derivative of \mathbf{u} , and $A_x, A_{xx} \in \mathbb{R}^{N_{\text{state}} \times N_{\text{state}}}$ are the central difference first-order and second-order space derivative operators, respectively, which take into account the boundary conditions, too. The model is implemented in Matlab and the backward Euler method is employed for time discretization. The nonlinear algebraic systems are solved using the Newton-Raphson method and the allowed number of Newton iterations per each time step is set to 50. The solution is considered to have converged when the Euclidean norm of the residual is less than 10^{-10} .

The viscosity parameter space \mathcal{P} is set to the interval $[0.01, 1]$. Smaller values of μ correspond to sharper gradients in the solution, and lead to dynamics more difficult to accurately approximate using reduced-order models.

The reduced-order models are constructed using the POD method whereas the quadratic nonlinearities are computed via tensorial POD [110] for efficiency. A floating point operations analysis of tensorial POD, POD and POD/DEIM for p^{th} order polynomial nonlinearities is available in [110]. The computational efficiency of the tensorial POD 1D Burgers model can be noticed in Figure 5.2. Both on-line and off-line computational costs

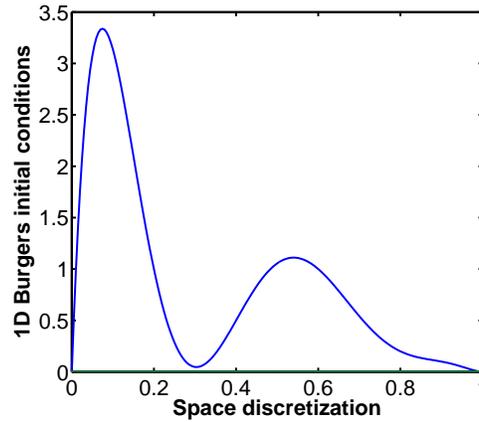


Figure 5.1: Seventh order polynomial used as initial conditions for 1D Burgers model.

are shown. Here we selected $\mu = \mu_p = 0.7$, $N_t = 301$, POD dimension $K_{POD} = 9$, and we let the number of space points N_s to vary. For $N_s = 201$ and 701, the tensorial POD model is $5.17\times$ and $61.12\times$ times faster than the high-fidelity version. The rest of our numerical experiments uses $N_s = 201$ and $N_t = 301$.

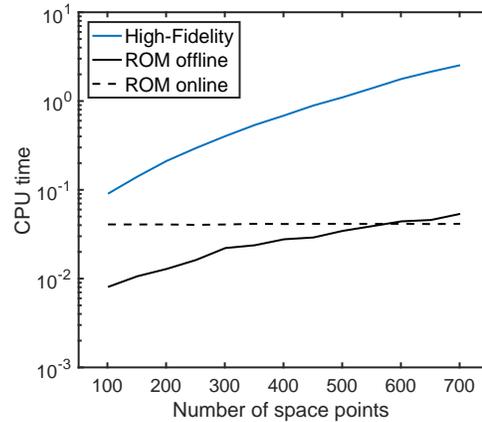


Figure 5.2: Computational efficiency of the tensorial POD 1D Burgers model. CPU time is given in seconds.

5.4.2 Multivariate prediction of local reduced-order models characteristics (MP-LROM) using regression machine learning methods

Error estimation of local ROM solutions

Here, we will use GP and ANN to approximate the MP-LROM error model introduced in (5.5). The approximated models have the following form

$$\phi_{MP-LROM}^e : \{\mu, \mu_p, K_{POD}\} \mapsto \widehat{\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}}, \quad (5.17)$$

where the input features include a viscosity parameter value μ , a parameter value μ_p associated with the full model run that generated the basis U_{μ_p} , and the dimension of the reduced manifold K_{POD} . The target is the estimated logarithm of error of the reduced-order model solution at μ using the basis U_{μ_p} and the corresponding reduced operators computed using the Frobenius norm

$$\begin{aligned} \log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF} = \\ \log \left(\left\| \mathbf{x}(\mu, t_1) - U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu, t_1) \quad \mathbf{x}(\mu, t_2) - U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu, t_2) \quad \cdots \quad \mathbf{x}(\mu, t_{N_t}) - U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu, t_{N_t}) \right\|_F \right). \end{aligned} \quad (5.18)$$

The probabilistic models described generically in equation (5.17) are just approximations of the MP-LROM model (5.5) and have errors. For our experiments, the data set includes 10 and 100 equally distributed values of μ_p and μ over the entire parameter region; i.e., $\mu_p \in \{0.1, 0.2, \dots, 1\}$ and $\mu \in \{0.01, \dots, 1\}$, 12 reduced basis dimensions K_{POD} spanning the interval $\{4, 5, \dots, 14, 15\}$ and the reduced-order model logarithm of errors $\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$. This leads to a data set of 12000 samples. Despite using a large data set of 12000 samples, training and evaluating the MP-LROM model require only 100 high-fidelity models runs. Out of these 100 high-fidelity runs, only 10 are used to construct the reduced-order models (bases plus reduced operators for all possible $K_{POD} \in \{4, 5, \dots, 14, 15\}$) corresponding to the 10 values of μ_p . The total number of generated reduced-order models is 120 corresponding to all 10 μ_p and 12 K_{POD} values. These 120 reduced-order models are then run for all 100 parameters μ leading to 12000 reduced-order models simulations corresponding to 12000 samples in the data set. Evaluating the true errors of the 12000 reduced-order models solutions requires only knowing the high-fidelity model solutions for the 100 parameters μ in the database. Since all 10 parameters μ_p belong to the set of

all 100 parameters μ , only 100 high-fidelity runs are needed to construct the entire data set of 12000 samples.

Figure 5.3 shows isocontours of the error $\varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ and $\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ of the reduced-order model solution for various viscosity parameter values μ and POD basis dimensions. The design of the reduced-order models relies on the high-fidelity trajectory for $\mu_p = 0.8$. The target values $\varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ vary over a wide range (from 300 to 10^{-6}) motivating the choice of implementing models that target $\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ to decrease the variance of the predicted results.

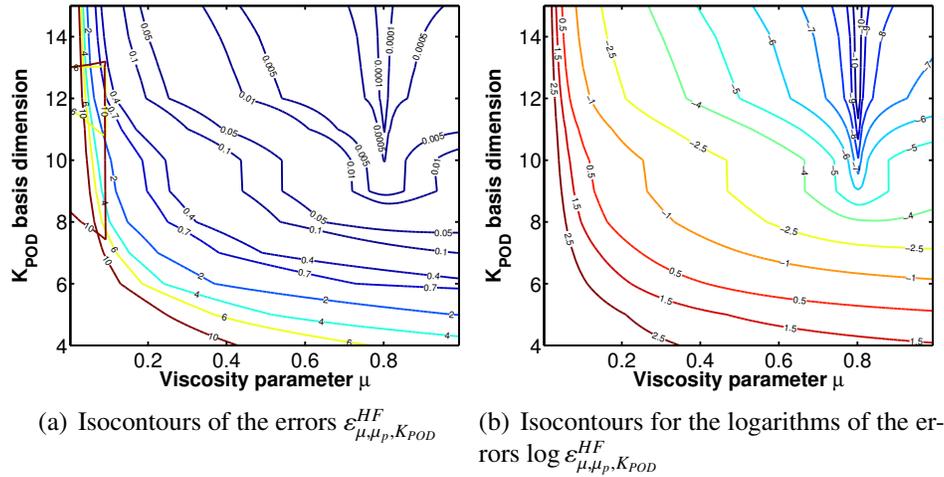


Figure 5.3: Isocontours of the reduced model errors for different POD basis dimensions and parameters μ . The reduced-order model uses a basis constructed from the full order simulation with parameter value $\mu_p = 0.8$.

A more detailed analysis, comparing models

$$\phi_{MP-LROM}^e : \{\mu, \mu_p, K_{POD}\} \mapsto \hat{\varepsilon}_{\mu, \mu_p, K_{POD}}^{HF} \quad (5.19)$$

that target no scaled data and model (5.17) is given in the following.

The approximated MP-LROM models for estimating the local parametric reduced-order model errors are constructed using a Gaussian Process with a squared-exponential covariance kernel (5.10) and a neural network with six hidden layers and hyperbolic tangent sigmoid activation function in each layer. Tables 5.1 and 5.2 show the averages and variances of errors in prediction of MP-LROM models for different sample sizes. Every subset of samples is selected randomly from a shuffled original data set. The misfit is computed

using the same formulas presented in (5.8a) to evaluate the prediction errors. Table 5.1 shows the prediction errors of (5.19) computed via equation (5.8a) with $y = \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ and $\hat{y} = \hat{\varepsilon}_{\mu, \mu_p, K_{POD}}^{HF}$; i.e., no data scaling; the predictions have a large variance and a low accuracy. The testing sets used by Gaussian Process and ANN MP-LROM models for each number of samples are the same explaining the similar behaviour in the case of 5000 samples. The data set contains ROM errors ranging 6 orders of magnitude. The randomly selected training data set contains samples which are very different from the randomly selected test data explaining the increase in the average and variance of errors. Scaling the data and targeting $\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ results using (5.17), reduce the variance of the predictions, and increase the accuracy, as shown in Table 5.2. The same formula (5.8a) with $y = \log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ and $\hat{y} = \widehat{\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}}$ was applied. We notice that, for increasing sample sizes less or equal than 700 and for scaled data, the variances of GP and ANN predictions are not necessarily decreasing. This behavior changes and the variances of both regression models decrease for increasing sample sizes larger than 700 as seen in Table 5.2. The performance of the ANN and GP is highly dependent on the number of samples in the data set. As the number of data points grows, the accuracy increases and the variance decreases. The results show that GP outperforms ANN for small numbers of samples ≤ 1000 whereas, for larger data sets, ANN is more accurate than GP.

Sample size	GP MP-LROM		ANN MP-LROM	
	E_{fold}	VAR_{fold}	E_{fold}	VAR_{fold}
100	13.4519	5.2372	12.5189	25.0337
400	6.8003	31.0974	6.9210	26.1814
700	5.6273	14.3949	7.2325	19.9312
1000	3.7148	13.8102	5.6067	14.6488
3000	0.5468	0.0030	1.2858	1.2705
5000	6.0563	22.7761	3.8819	23.9059

Table 5.1: Average and variance of error in predictions of MP-LROM models (5.19) constructed via ANN and GP using errors $\varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ in training data for different sample sizes.

Sample size	GP MP-LROM		ANN MP-LROM	
	E_{fold}	VAR_{fold}	E_{fold}	VAR_{fold}
100	0.5319	0.0118	1.2177	0.1834
400	0.3906	0.0007	0.8988	0.2593
700	0.3322	0.0018	0.7320	0.5602
1000	0.2693	0.0002	0.5866	0.4084
3000	0.1558	0.5535×10^{-4}	0.01202	0.2744×10^{-4}
5000	0.0775	0.4085×10^{-5}	0.0075	0.3812×10^{-5}

Table 5.2: Average and variance of error in predictions of MP-LROM models (5.19) constructed via ANN and GP using logarithms of errors $\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ in training data for different sample sizes.

Figures 5.4 and 5.5 show the corresponding histogram of the errors in prediction of MP-LROM models (5.17) and (5.19) using 100 and 1000 training samples for ANN and GP methods, respectively. The histograms shown in Figure 5.5 can assess the validity of GP assumptions (5.9), (5.11), (5.13). The difference between the true and estimated values should behave as samples from the distribution $\mathcal{N}(0, \sigma_n^2)$ [97]. In our case they are hardly normally distributed and this indicates that the data sets are not from Gaussian distributions.

Scaling the data and targeting $\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ errors clearly improve the performance of the MP-LROM models. Consequently for the rest of the manuscript we will only use model (5.17).

To assess the quality of the MP-LROM models, we also implemented a five-fold cross-validation test over the entire dataset. The results computed using formula (5.8b) are shown in Table 5.3. ANN outperforms GP and estimates the errors more accurately. It also has less variance than the Gaussian Process which indicates it has more stable predictions.

	E	VAR
ANN MP-LROM	0.004004	2.16×10^{-6}
GP MP-LROM	0.092352	1.32×10^{-5}

Table 5.3: MP-LROM statistical results over five-fold cross-validation.

Figure 5.6 illustrates the average of errors in prediction of five different errors models computed using ANN and GP regression methods. The predictions were made with a

random fixed test set excluded from the entire data set and contains various values of μ , K_{POD} and μ_p shown in the x -axes of Figure 5.6. The error models were constructed using a training set formed by 80% randomly selected data of the entire data set (not including the test set).

Building different GP and ANN MP-LROM error models, each trained on different part of the data set and then testing them with the same fixed test set, reduces the bias in prediction. Again, ANN outperforms GP having more accurate errors estimates.

We also compared the MP-LROM models with those obtained by implementing ROMES method [97] and MFC technique [96]. The ROMES method constructs univariate models

$$\phi_{ROMES} : \log \rho(\mu) \mapsto \log \varepsilon_{\mu}^{HF}, \quad (5.20)$$

where the input $\rho(\mu)$ consists of error indicators. Examples of indicators include residual norms, dual-weighted residuals and other error bounds. MFC implements input-output models

$$\phi_{MFC} : \mu \mapsto \log \varepsilon_{\mu}^{HF}, \quad (5.21)$$

where the input of error models is the viscosity parameter μ . Both ROMES and MFC methods use a global reduced-order model with a fixed dimension in contrast to our method that employs local reduced-order models with various dimensions. ROMES and MFC models are univariate whereas the MP-LROM models are multivariate.

To accommodate our data set to the requirements of the ROMES and MFC methods, we separated the data set into 120 subsets. Each of these subsets has 100 samples corresponding to a single μ_p and K_{POD} and 100 values of parameter $\mu \in \{0.01, 0.02 \dots, 1\}$. For each subset we constructed ANN and GP models to approximate the input-output models defined in (5.20) and (5.21) using the same training set. In the case of ROMES method we employed the logarithms of residuals norms as inputs. We first computed the corresponding reduced-order solution and then the associated logarithm of residual norm by using the projected reduced order solution into the high-fidelity model for parameter μ . The output of both ROMES and MFC models approximates the logarithm of the Frobenius norm of the reduced-order-model errors.

The required numbers of high-fidelity models runs, constructed reduced-order models, reduced-order models runs, and computed residuals are shown in Table 5.4 together with their associated computational times. We also present the computational cost of computing the ANN weights for each of the methods. The numbers of high fidelity models runs required by each of the methods are the same.

	MP-LROM	ROMES	MFC
No of samples	12000	12000	12000
No of high fidelity models runs	100	100	100
CPU time	9s	9s	9s
No of ROM models built	120	120	120
CPU time	0.6s	0.6s	0.6s
No of ROM models run	12000	12000	12000
CPU time	10.8s	10.8s	10.8s
No of computed residuals	0	12000	0
CPU time	0	12s	0
CPU time for building ANN	2279.95s	427.29s	421.23s
Total CPU time	2300.35s	459.69s	441.63

Table 5.4: Total computational cost for building the ANN models for MP-LROM, ROMES and MFC methods. This is the cost of generating one multivariate MP-LROM model and 120 univariate ROMES and MFC models.

The overall computational cost is higher for constructing the MP-LROM model due to the cost of generating the ANN weights. Since constructing the ANN weights for each of the ROMES and MFC models can be done in parallel, the CPU time for generating 120 models is the same as for generating one model.

Figures 5.7-5.10 shows the isocontours of the E_{fold} and VAR_{fold} computed using (5.8a) for different K_{POD} and μ_p using ROMES, MFC, and MP-LROM models constructed using GP and ANN methods. In total there are 12×10 configurations corresponding to different K_{POD} and μ_p and as many ROMES and MFC models. The MP-LROM models are global in nature and the training set is the whole original data set. The testing set is the same for all the compared models and differs from the training sets. We can see that MP-LROM models are more accurate than those obtained via ROMES and MFC models. Including more samples associated with various POD basis sizes and μ_p is benefic. We also trained and tested all the models using five-fold cross-validation. The average error and variance of all 120 E_{fold} s and VAR_{fold} s are compared against those obtained using MP-LROM error models and are summarized in tables 5.5 and 5.6. This shows that the MFC models outperform the ROMES ones, for our experiment, and the MP-LROM models are the most accurate. The MP-LROM models perform better since they employ more features and samples than the other models which help the error models tune the weights better. We also notice the efficiency of the MFC models from accuracy point of view considering that they use very few samples. In the case of large parametric domains the MP-LROM error

models may require a very large data set with a lot of features. By using only subsets of the whole data set near the vicinity of the parameters of interest and applying the active subset method [118] can help prevent the potential curse of dimensionality problem that MP-LROM might suffer.

	ROMES	MFC	MP-LROM
ANN	0.3844	0.0605	8.8468×10^{-4}
GP	0.2289	0.0865	0.0362

Table 5.5: Average error of all 120 E_{folds} for three methods.

	ROMES	MFC	MP-LROM
ANN	0.0541	0.0213	4.9808×10^{-7}
GP	0.0051	0.0049	5.4818×10^{-4}

Table 5.6: Average variance of all 120 $\text{VAR}_{\text{folds}}$ for three methods

Finally we compare and show the average of the errors in prediction of five different errors models designed using ROMES, MFC, and MP-LROM methods for one of the subsets corresponding to $K_{POD} = 10$ and $\mu_p = 1$. The testing set is randomly selected from the samples and is not included in the training sets. The training set for both ROMES and MFC models are the same. In order to prevent the bias in prediction, each time the error models are trained on randomly selected 80% of the training sets and tested with the fixed test set. We repeated this five times and the average of error in prediction is obtained. Figure 5.11 shows the average of error in prediction for all models implemented using GP and ANN methods.

The required numbers of high-fidelity models runs, constructed reduced-order models, reduced-order models runs, and computed residuals are shown in Table 5.7 together with their associated computational times. We also present the computational cost of computing the ANN weights for each of the methods. The numbers of high fidelity models runs required by each of the methods are the same.

	MP-LROM	ROMES	MFC
No of samples	12000	100	100
No of high fidelity models runs	100	100	100
CPU time	9s	9s	9s
No of constructed ROM models	120	1	1
CPU time	0.6s	0.005s	0.005s
No of ROM models runs	12000	100	100
CPU time	10.8s	0.09s	0.09s
No of computed residuals	0	100	0
CPU time	0	0.01s	0
CPU time for building ANN	2279.95s	427.29s	421.23s
Total CPU time	2300.35s	436.485s	430.325s

Table 5.7: Total computational cost for building the artificial neural network (ANN) models for MP-LROM, ROMES and MFC methods. This is the cost of generating one multivariate MP-LROM model, one univariate ROMES model and one univariate MFC models.

The overall computational cost is higher for constructing the MP-LROM model mainly due to the cost of generating the ANN weights. This is not due to a higher number of high-fidelity models runs. The data set for MP-LROM model also requires 12000 reduced-order models runs in comparison to only 100 reduced-order models runs needed by ROMES or MFC. However, MP-LROM model is capable to predict the logarithm of errors for 120 reduced-order models whereas ROMES and MFC for only one.

Since we are including more features in our mappings, we achieve more accurate predictions compared to other existing methods such as ROMES and MFC. However, there is always a trade-off between the computational complexity and the accuracy. For more accurate results, one can generate a bigger dataset with more samples taken from the parameter domain of the underlying model. This elevates the computational complexity since the probabilistic mappings are more costly to construct in the training phase. Techniques such as principal component analysis and active subspace can alleviate the curse of dimensionality for big data sets by selecting the most effective features and ignoring the less-effective ones.

Selecting the dimension of reduced-order model

Here we construct MP-LROM models to predict the reduced basis dimension that account for a-priori specified accuracy levels in the reduced-order model solution. The models are constructed using GP and ANN methods and have the following form

$$\phi_{MP-LROM}^d : \{\mu_p, \log \varepsilon_{\mu_p, \mu_p, K_{POD}}^{HF}\} \mapsto \widehat{K_{POD}}. \quad (5.22)$$

The input features of this model consist of the viscosity parameter $\mu_p \in [0.01, 1]$ and the log of the Frobenius norm of the error between the high-fidelity and reduced-order models (5.18). The searched output $\widehat{K_{POD}}$ is the estimation of the dimension of the reduced manifold K_{POD} . The data set contains equally distributed values of μ_p over the entire parametric domain $\mu_p \in \{0.01, 0.0113, 0.0126, \dots, 0.9956\}$, reduced basis dimensions K_{POD} spanning the set $\{4, 5, \dots, 14, 15\}$ and the logarithm of the reduced-order model error $\log \varepsilon_{\mu_p, \mu_p, K_{POD}}^{HF}$. We use GP and ANN methods to construct two MP-LROM models to predict the dimension of local reduced-order models given a prescribed accuracy level.

During the training phase, the MP-LROM models will learn the dimensions of reduced-order basis K_{POD} associated with the parameter μ_p and the corresponding error $\log \varepsilon_{\mu_p, \mu_p, K_{POD}}^{HF}$. Later they will be able to estimate the proper dimension of reduced basis by providing it the specific viscosity parameter μ_p and the desired precision $\log \bar{\varepsilon}$. The computational cost is low once the models are constructed. The output indicates the dimension of the reduced manifold for which the ROM solution satisfies the corresponding error threshold. Thus we do not need to compute the entire spectrum of the snapshots matrix in advance which for large spatial discretization meshes translates into important computational costs reduction. Figure 5.12 illustrates the contours of the log of reduced-order model errors over all the values of the viscosity parameter $\mu_p \in \{0.01, 0.0113, 0.0126 \dots 1\}$ and various POD dimensions $K_{POD} = \{4, 5, \dots, 14, 15\}$.

A neural network with 5 hidden layers and hyperbolic tangent sigmoid activation function in each layer is used while for the Gaussian Process we have used the squared-exponential-covariance kernel (5.10). For both MP-LROM models, the results were rounded such as to generate natural numbers. Table 5.8 shows the average and variance of error in GP and ANN predictions using different sample sizes. ANN outperforms GP and as the number of data points grows, the accuracy increases and the variance decreases. The results are obtained using a conventional validation with 80% of the sample size dedicated for training data and the other 20% for the test data. The employed formula is described in equation (5.8a).

sample size	MP-LROM GP		MP-LROM ANN	
	E_{fold}	VAR_{fold}	E_{fold}	VAR_{fold}
100	0.2801	0.0901	0.1580	0.02204
1000	0.1489	0.0408	0.0121	0.0015
3000	0.1013	0.0194	0.0273	0.0009
5000	0.0884	0.0174	0.0080	0.0002

Table 5.8: Average and variance of errors in prediction of reduced basis dimension using MP-LROM models for different sample sizes

Figures 5.13 and 5.14 show the prediction errors using 100 and 1000 training samples for both MP-LROM models constructed via ANN and GP models. The histograms shown in Figure 5.14, as stated before, can assess the validity of GP assumptions. Once the number of samples is increased, the data set distribution shape is closer to the Gaussian profile $\mathcal{N}(0, \sigma_n^2)$ than in the case of the data set distribution shown in Figure 5.5 used for generation of MP-LROM models for the prediction of local reduced-order model errors.

To assess the accuracy of the MP-LROM models, the data set is randomly partitioned into five equal size sub-samples, and five-fold cross-validation test is implemented. The five results from the folds are averaged and they are presented in Table 5.9. The ANN model correctly estimated the dimension of the reduced manifold in 87% cases. GP correctly estimates the POD dimension 53% of the times. The variance results shows that the GP model has more stable predictions indicating a higher bias in the data.

Dimension Discrepancy	zero	one	two	three	four	> four	VAR
ANN MP-LROM	87%	11%	2%	0	0	0	2.779×10^{-3}
GP MP-LROM	53%	23%	15%	5%	3%	1%	4.575×10^{-4}

Table 5.9: POD basis dimension discrepancies between the MP-LROM predictions and true values over five-fold cross-validation. The errors variance is also computed.

In Figure 5.15, we compare the output of the MP-LROM models against the singular values based estimation on a set of randomly selected test data. The estimation derived from the singular values is the standard method for selecting the reduced manifold dimension when a prescribed level of accuracy of the reduced solution is desired. Here the desired accuracy $\bar{\epsilon}$ is set to 10^{-3} . The mismatches between the predicted and true dimensions are

depicted in Figure 5.15. The predicted values are the averages over five different MP-LROM models constructed using ANN and GP methods. The models were trained on random 80% split of data set and tested on the fixed selected 20% test data. We notice that the snapshots matrix spectrum underestimates the true dimension of the manifold as expected since the ‘in-plane’ errors are not accounted. The ANN predictions were extremely accurate for most of the samples while the GP usually overestimated the reduced manifold dimensions.

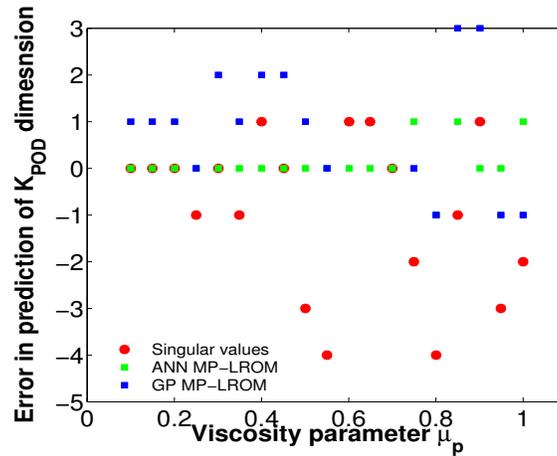


Figure 5.15: Average error of the POD dimension prediction on a randomly selected test data with desired accuracy of $\bar{\epsilon} = 10^{-3}$. The average of the absolute values of the error in prediction are 1.31, 0.21 and 1.38 for singular value based method and MP-LROM models constructed using ANN and GP.

5.5 Conclusions

In this study, we introduced new multivariate input-output models (MP-LROM) to predict the errors and dimensions of local parametric reduced-order models. Approximation of these mappings were built using Gaussian Process and Artificial Neural Networks. Initially, we compared our MP-LROM error models against those constructed with multi-fidelity correction technique (MFC) and reduced order model error surrogates method (ROMES). Since global bases are used by MFC and ROMES methods, we implemented

corresponding local error models using only small subsets of the data utilized to generate our MP-LROM models. In contrast, the MP-LROM models are global and rely on a global database. Moreover, our MP-LROM models differ from the ROMES [97] and MFC models [96], having more additional features such as reduced subspace dimension and are specially projected for accurate predictions of local parametric reduced-order models errors. As such, the MP-LROM models require significantly more and different data than MFC models. The numerical experiments revealed that our MP-LROM models are more accurate than the models constructed with MFC and ROMES methods for estimating the errors of local parametric reduced-order 1D-Burgers models with a single parameter. In the case of large parametric domains, the MP-LROM error models could be affected by the curse of dimensionality due to the large number of input features. In the future we plan to use only subsets of the global data set near the vicinity of the parameters of interest and combine our technique with the active subspace method [118] to prevent the potential curse of dimensionality that the MP-LROM models might suffer.

Next we addressed the problem of selecting the dimension of a local reduced-order model when its solution must satisfy a desired level of accuracy. The approximated MP-LROM models based on Artificial Neural Networks better estimated the ROM basis dimension in comparison with the results obtained by truncating the spectrum of the snapshots matrix.

In the future we seek to decrease the computational complexity of the MP-LROM error models. Currently the training data required by the machine learning regression MP-LROM models rely on many high-fidelity simulations. By employing error bounds, residual norms [97] and a-posteriori error estimation results [136, 117], this dependency could be much decreased. ROMES and MFC models will also benefit since they require the same amount of high-fidelity models runs as MP-LROM models. On-going work focuses on applications of MP-LROM error model. We are currently developing several algorithms and techniques that employ MP-LROM error model as a key component to generate decomposition maps of the parametric space associated with accurate local reduced-order models.

In addition, we plan to construct machine learning MP-LROM models to estimate the errors in quantities of interest computed with reduced-order models. The predictions of such error models can then be used to speed up the current trust-region reduced-order framework [137, 138] by eliminating the need of high-fidelity simulations for the quality evaluation of the updated controls.

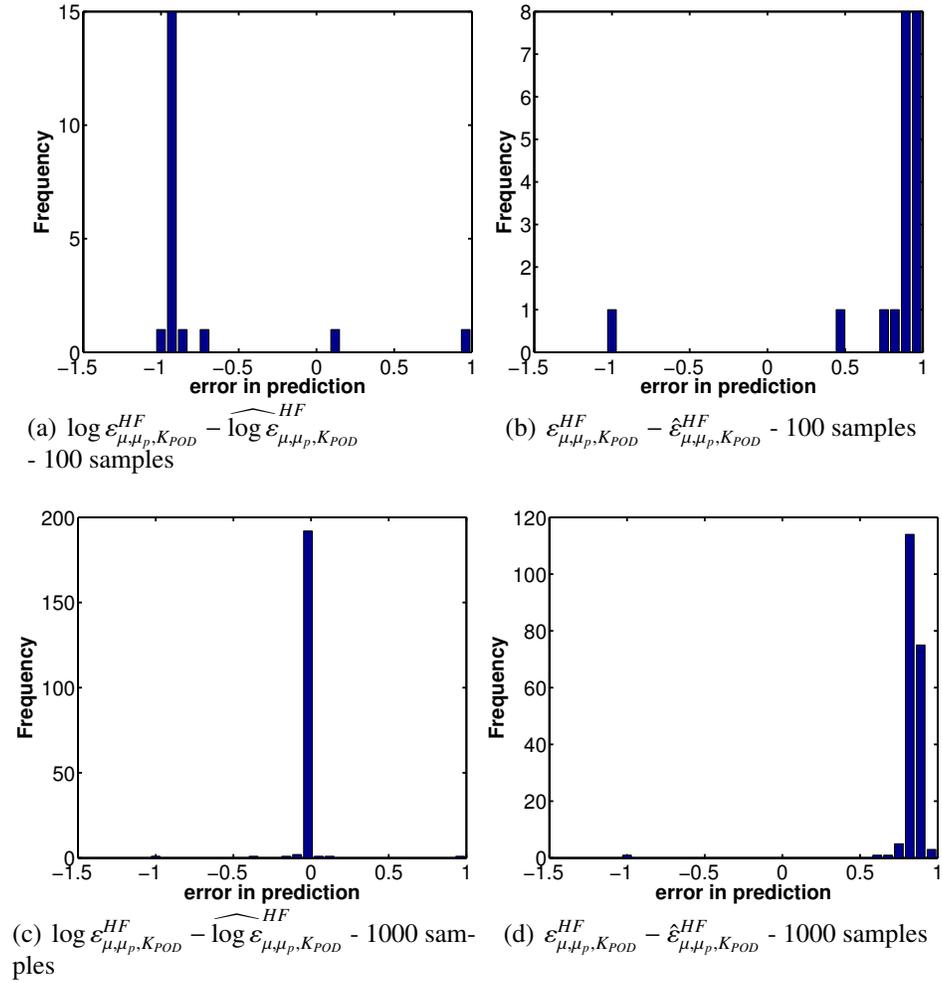


Figure 5.4: Histogram of errors in prediction using ANN MP-LROM.

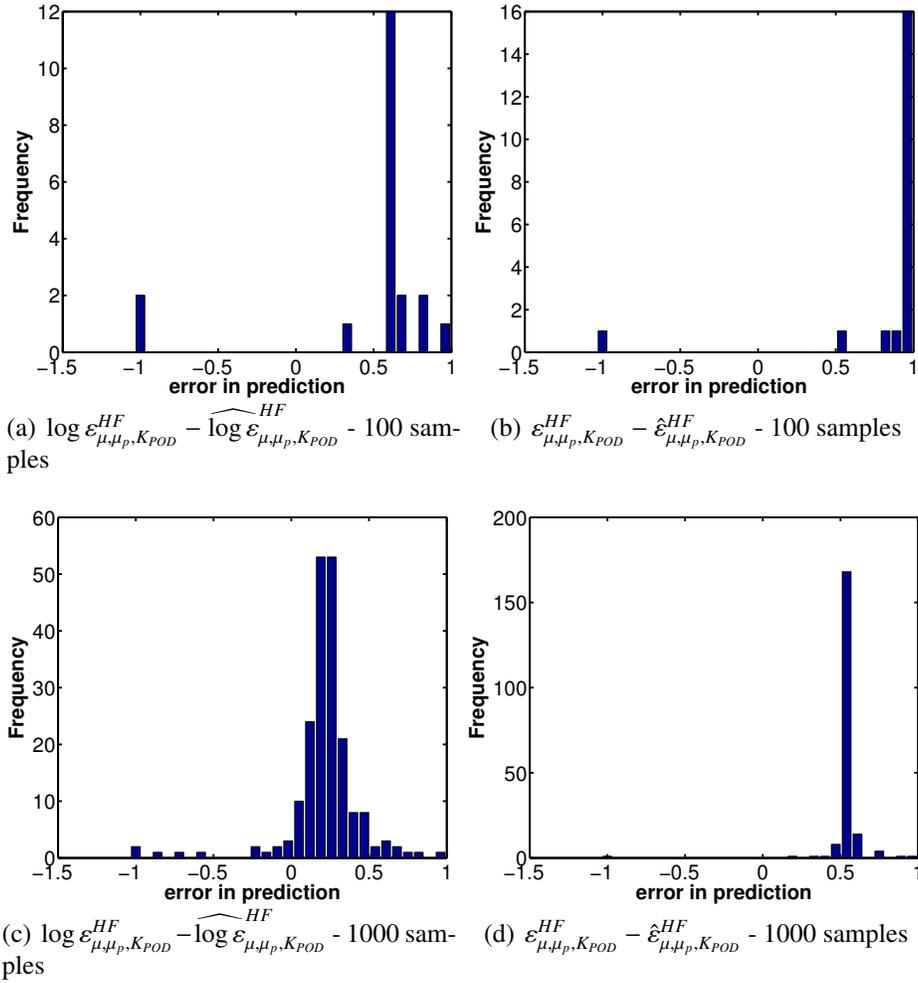


Figure 5.5: Histogram of errors in prediction using GP MP-LROM.

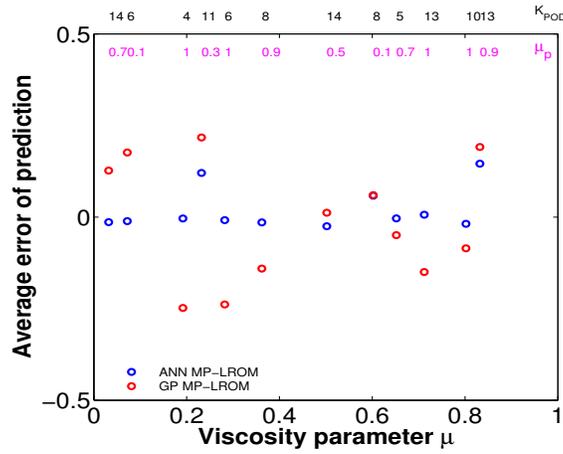


Figure 5.6: The average of errors in predictions using five different trained models. The top labels shows the corresponding μ_p and K_{POP} for each parameter μ .

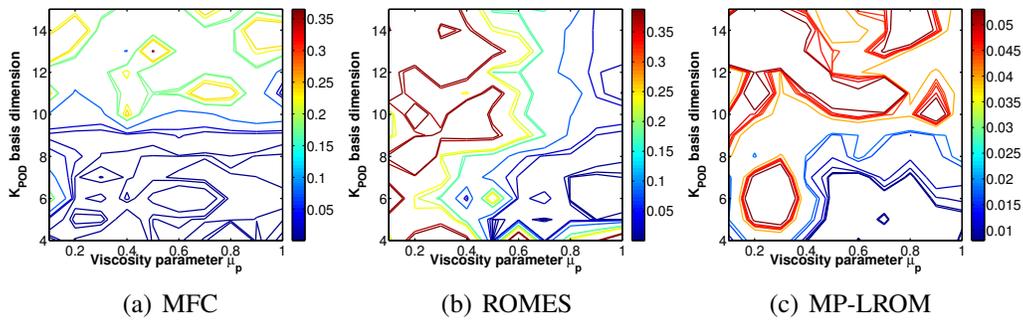


Figure 5.7: Isocontours for the E_{fold} using GP method.

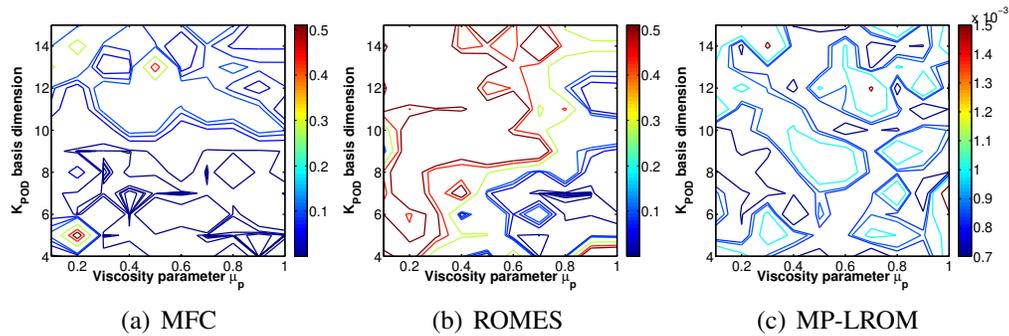
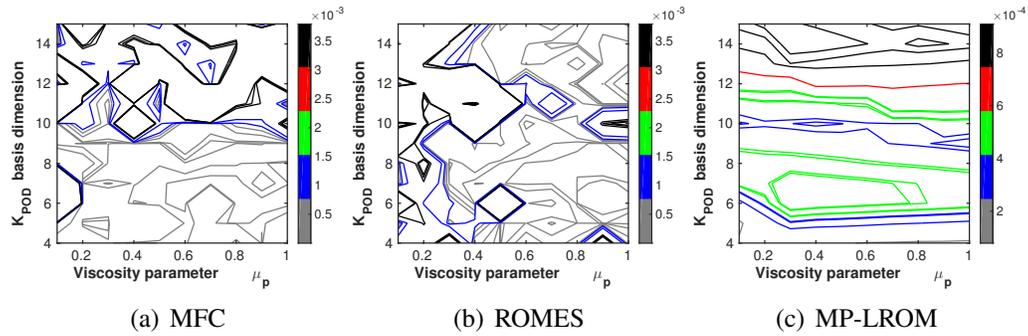
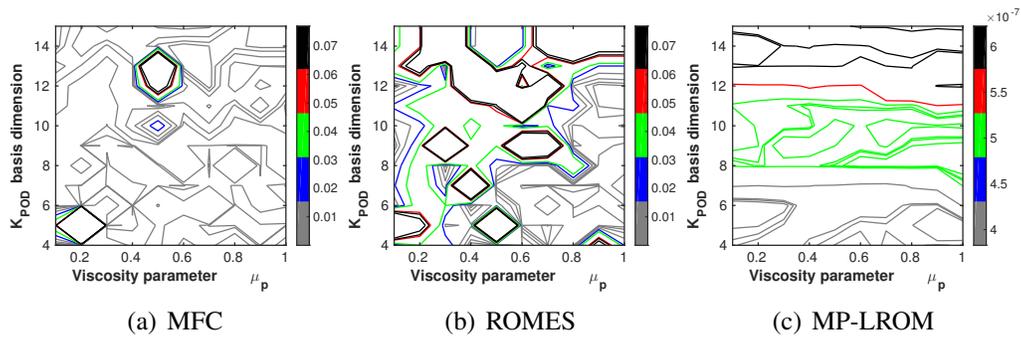


Figure 5.8: Isocontours for the E_{fold} using ANN method.

Figure 5.9: Isocontours for the VAR_{fold} using GP method.Figure 5.10: Isocontours for the VAR_{fold} using ANN method.

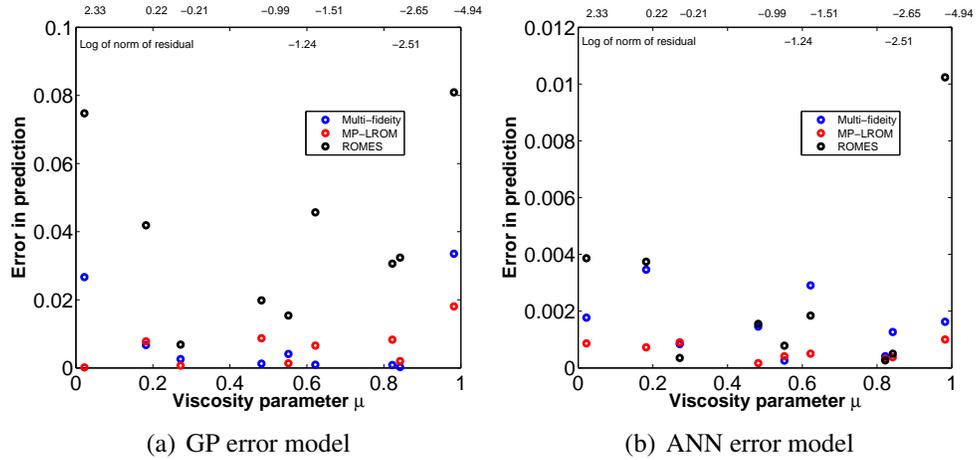


Figure 5.11: The error in predictions of all methods using $K_{POD} = 10$ and $\mu_p = 1$. For GP error models the overall average of errors in predictions is 0.0131, 0.0487, 0.0095 for MFC, ROMES and MP-LROM respectively. For ANN error models the overall average of errors in predictions is 0.0056, 0.0240, 0.0029 for MFC, ROMES and MP-LROM respectively. The top x-axis shows the corresponding logarithms of the residuals norms used as inputs for the ROMES method.

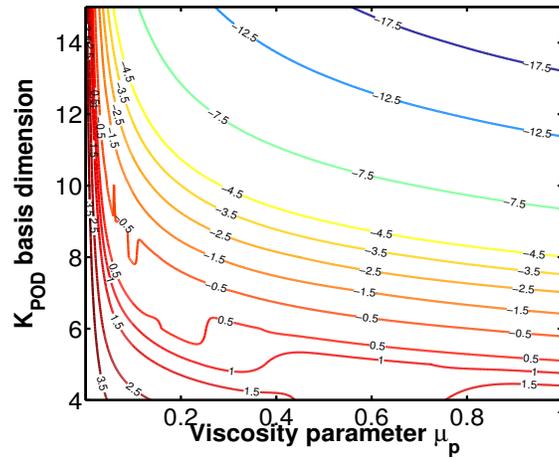


Figure 5.12: Isocontours of the reduced model errors for different POD basis dimensions and viscosity parameters μ_p .

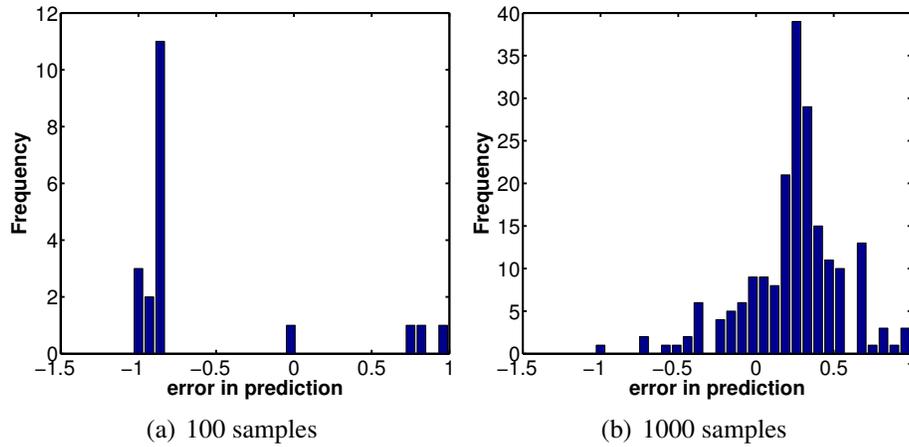


Figure 5.13: Histogram of errors in prediction of the reduced basis dimension using ANN MP-LROM for different sample sizes

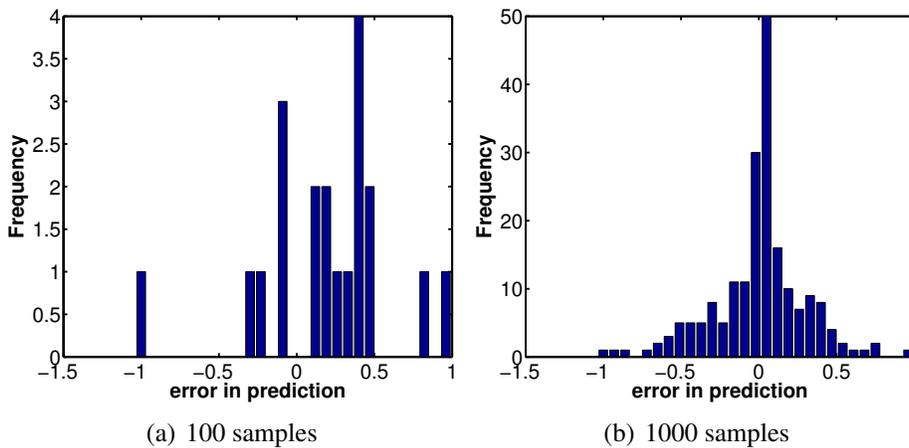


Figure 5.14: Histogram of errors in prediction of the reduced basis dimension using GP ML-ROM for different sample sizes

Chapter 6

Parametric Domain Decomposition for Accurate Reduced Order Models: Applications of MP-LROM Methodology

6.1 Introduction

Many physical phenomena in science and engineering are investigated today using large-scale computer simulation models. The ever-increasing complexity of these high-fidelity models poses considerable challenges related to computational time, memory requirements, and communication overhead in a parallel environment. A popular approach to alleviate these challenges is to construct inexpensive surrogate (approximate) models that capture the most important dynamical characteristics of the underlying physical models, but reduce the computational complexity by orders of magnitude. Examples of surrogates include response surfaces, low resolution models, and reduced-order models.

Reduced-order modeling uses snapshots of high-fidelity model solutions at different times to extract a low-dimensional subspace that captures most of the high-fidelity solution energy. The reduced-order surrogate is obtained by projecting the dynamics of the high-fidelity model onto the low-dimensional subspace. This is usually achieved by orthogonal or oblique projections coined as Galerkin or Petrov–Galerkin methods where the solution is searched as a linear combination of the basis vectors. Since the Galerkin method

is actually an elliptic approach, applying it to hyperbolic models must be done with careful consideration [139]. The reduced dimension leads to a considerable reduction in computational complexity at the expense of a decreased solution accuracy. A reduced-order model approximates well the high-fidelity solution at a given point of operation (e.g., for the model parameter values for which the snapshots have been taken), but becomes less accurate away from that point (e.g., when the high-fidelity model is run with different parameter values).

To be useful, reduced-order models must accommodate changes over the entire parametric space without losing their accuracy, simplicity and robustness. Reduced-order model (ROM) accuracy and robustness can be achieved by constructing a global basis [18, 19], but this strategy generates large dimensional bases that may lead to slow reduced-order models. Moreover, for fluid flows, the Galerkin expansion with global modes presumes synchronized flow dynamics. Whereas this assumption is true for internal flows, it is not suited for transient shear flows with uni-directional 'hyperbolic' convection of vortices [140]. Changes in the operational settings may lead to deformation of leading flow structures [141] especially if the model is characterized by bifurcations and multiple attractors. Approaches such as limiting the operational setting, extending the mode sets [142] and offline/online set adaptation address the issue of mode deformation.

In localization approaches, the reduced-order models are built offline and one is chosen depending on the current state of the system. Local approaches have been designed for parametric [143, 22] or state spaces generating local bases for both the state variables [20, 21] and non-linear terms [22, 23]. Dictionary approaches [144, 145] pre-compute offline many basis vectors and then adaptively select a small subset during the online stage. Error bounds for reduced-order approximations of parametrized parabolic partial differential equations are available in [84].

In this study, we employ machine learning regression models to guide the construction of parametric space decompositions for solving parametric partial differential equations using accurate local reduced-order models. Whereas the current methodologies are defined in the sense of Voronoi tessellation and rely on K-means algorithms, our approach delimitates subregions of the parametric space by applying an Artificial Neural Networks model to estimate the errors of reduced-order models following a parametric domain sampling algorithm.

Machine learning methodologies have been applied to predict and model the approximation errors of low-fidelity and surrogate models [146, 147, 148]. The multi-fidelity correction (MFC) approach [96, 98, 13, 99] has been developed to approximate the low-fidelity models errors in the context of optimization. The reduced order model error surrogates

method (ROMES) [97] seeks to estimate full errors from indicators such as error bounds and reduced-order residual norms. Both ROMES and MFC models predict the error of global reduced-order models with fixed dimension using univariate functions.

In contrast, the authors' multivariate predictions of local reduced-order-model method (MP-LROM) [149] proposes a multivariate model to compute the error of local reduced-order surrogates. A MP-LROM model based on Artificial Neural Network and a sampling algorithm are applied here to construct decompositions of the parametric domain for solving parametric partial differential equations using local reduced-order models that are accurate within an admissible prescribed threshold. The proposed strategy relies on a greedy algorithm that samples the vicinity of each parameter value used to generate a local reduced-order model and finds an open ball such that for all the parameters in the ball the error of the local reduced-order model is less than the desired threshold. The essential ingredient is the MP-LROM error model which approximates the error of reduced-order model. Then a greedy technique is used to sample the parametric domain and generates a feasible region where a specific local reduced-order model provides accurate solutions within a prescribed tolerance. The union of these feasible regions forms a decomposition of the parametric domain. Different thresholds lead to different domain decompositions. The current methodology is designed for one dimensional parametric spaces and it is applied to the viscous 1D-Burgers model. A decomposition for the viscosity domain is generated for various error thresholds. Once the decomposition is constructed there is no need to run the high-fidelity model again, since for each parameter value μ there exists a parameter μ_p , and the associated reduced-order model (basis and reduced operators), whose solution error is accurately estimated a-priori. The dimension K_{POD} of the local basis is usually small since it depends only on one high-fidelity model trajectory.

The decomposition leads to a database of available bases, local reduced-order models and high fidelity trajectory. This database can be used to generate more accurate reduced-order models for an arbitrary parametric configuration. Three different approaches are compared here; i.e., bases interpolation, bases concatenation, and high-fidelity model solutions combination. For the first method, we perform a Lagrangian interpolation of the bases in the matrix space [150], or linearly interpolate their projections onto some local coordinate systems [150, 151]. The second method follows the idea of the spanning ROM introduced in [152], where a projection basis is created by concatenating some of the available bases for an arbitrary parameter. The third method interpolates the associated high-fidelity solutions and then extracts the singular vectors to generate a new basis and local reduced-order model.

The remainder of the chapter is organized as follows. Section 6.2 introduces the new methodology for constructing decompositions of the parametric domain using MP-LROM

error predictions. Then the potential of combining the existing information for generating more accurate reduced-order model is discussed. Section 6.3 presents the applications of the proposed methodologies to a viscous 1D-Burgers system. Conclusions are drawn in Section 6.4.

6.2 MP-LROM error model and its application to parametric domain decomposition

The MP-LROM models [149] are multivariate input-output maps $\phi : \mathbf{z} \mapsto y$ that predict different characteristics of local parametric reduced-order models, such as the error with respect to the high-fidelity model solution or basis dimension as explained in chapter 5. In what follows we explain the application of MP-LROM error model to better design of ROM.

6.2.1 Designing the decomposition of the parametric domain

Motivated by the need of fast and accurate simulations along the entire parametric space, we propose an alternative to the global parametric approach where only a single basis and a single reduced order model is constructed. Our alternative relies on a series of local reduced bases and reduced order models whose solutions meet prescribed admissible error thresholds beyond the parametric configurations employed for bases' construction. A parametric region, where a local reduced order model constructed using a single high-fidelity model trajectory is accurate to within a prescribed threshold, is called a feasible region. We delimitate such a region by employing the MP-LROM error model and sampling the neighborhood of the parametric configuration used to construct the local reduced order model. Our solution consists in designing a reunion of feasible regions completely covering the entire parametric space. This decomposition of the parametric domain was obtained as the solution of the following problem.

Problem 1 (Accurate local parametric ROMs). *For an arbitrary parameter configuration $\mu \in \mathcal{P}$ construct a reduced-order model (5.2) that provides an accurate and efficient approximation of the high-fidelity solution (5.1)*

$$\|\mathbf{x}(\mu, t_1) - U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu, t_1) \quad \mathbf{x}(\mu, t_2) - U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu, t_2) \quad \cdots \quad \mathbf{x}(\mu, t_{N_t}) - U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu, t_{N_t})\|_F \leq \bar{\epsilon}, \quad (6.1)$$

for some prescribed admissible error level $\bar{\varepsilon} > 0$. The snapshots used to generate the basis U_{μ_p} and reduced operators can be obtained with any parametric configuration $\mu_p \in \mathcal{P}$.

Our methodology proposes to select a finite countable subset $\mathcal{I} = \{\mu_{p_j}, j = 1, \dots, M\} \subset \mathcal{P}$ and for each μ_{p_j} , a reduced order basis $U_{\mu_{p_j}}$ along with the reduced operators are constructed for $j = 1, \dots, M$. We denote by \mathcal{U} the set of bases $U_{\mu_{p_j}}, j = 1, \dots, M$. If for each parameter configuration μ_{p_j} there exists an open ball $\mathcal{B}(\mu_{p_j}, r_j) \in \mathcal{P}$ such that, for all parameters $\mu \in \mathcal{B}(\mu_{p_j}, r_j)$, the reduced order solution $\tilde{\mathbf{x}}_{\mu_p}(\mu, t)$ satisfies (6.1) for $\mu_p = \mu_{p_j}$ and the parametric domain \mathcal{P} is a subset of the union of all these open balls, we obtain the sought decomposition of the parametric domain.

Next we derive a condition that guarantees the actual reduced-order model error

$$\varepsilon_{\mu, \mu_{p_j}, K_{POD}}^{HF} = \|\mathbf{x}(\mu, t_1) - U_{\mu_{p_j}} \tilde{\mathbf{x}}_{\mu_{p_j}}(\mu, t_1) \quad \mathbf{x}(\mu, t_2) - U_{\mu_{p_j}} \tilde{\mathbf{x}}_{\mu_{p_j}}(\mu, t_2) \quad \cdots \quad \mathbf{x}(\mu, t_{N_t}) - U_{\mu_{p_j}} \tilde{\mathbf{x}}_{\mu_{p_j}}(\mu, t_{N_t})\|_F, \quad (6.2)$$

depending on parameter configuration μ , parameter configuration μ_{p_j} and basis dimension K_{POD} , satisfies the prescribed admissible threshold $\varepsilon_{\mu, \mu_{p_j}, K_{POD}}^{HF} \leq \bar{\varepsilon}$, for any arbitrary parameter configuration μ inside of an open ball.

Theorem 6.2.1. *If $\lim_{\mu \rightarrow \mu_{p_j}} \|\mathbf{x}(\mu, t_1) - U_{\mu_{p_j}} \tilde{\mathbf{x}}_{\mu_{p_j}}(\mu, t_1) \quad \mathbf{x}(\mu, t_2) - U_{\mu_{p_j}} \tilde{\mathbf{x}}_{\mu_{p_j}}(\mu, t_2) \quad \cdots \quad \mathbf{x}(\mu, t_{N_t}) - U_{\mu_{p_j}} \tilde{\mathbf{x}}_{\mu_{p_j}}(\mu, t_{N_t})\|_F = \bar{\varepsilon}^*$, and $\bar{\varepsilon}^* \leq \frac{\bar{\varepsilon}}{2}$, then there exists $r_j > 0$ such that $\varepsilon_{\mu, \mu_{p_j}, K_{POD}}^{HF} \leq \bar{\varepsilon}$ is satisfied for all parameters μ inside the ball $\mathcal{B}(\mu_{p_j}, r_j)$.*

Proof. From $\lim_{\mu \rightarrow \mu_{p_j}} \|\mathbf{x}(\mu, t_1) - U_{\mu_{p_j}} \tilde{\mathbf{x}}_{\mu_{p_j}}(\mu, t_1) \quad \mathbf{x}(\mu, t_2) - U_{\mu_{p_j}} \tilde{\mathbf{x}}_{\mu_{p_j}}(\mu, t_2) \quad \cdots \quad \mathbf{x}(\mu, t_{N_t}) - U_{\mu_{p_j}} \tilde{\mathbf{x}}_{\mu_{p_j}}(\mu, t_{N_t})\|_F = \bar{\varepsilon}^*$, using the limit definition, we have that for all $\varepsilon > 0$, there exists another real number $\delta > 0$ such that

$$|\|\mathbf{x}(\mu, t_1) - U_{\mu_{p_j}} \tilde{\mathbf{x}}_{\mu_{p_j}}(\mu, t_1) \quad \mathbf{x}(\mu, t_2) - U_{\mu_{p_j}} \tilde{\mathbf{x}}_{\mu_{p_j}}(\mu, t_2) \quad \cdots \quad \mathbf{x}(\mu, t_{N_t}) - U_{\mu_{p_j}} \tilde{\mathbf{x}}_{\mu_{p_j}}(\mu, t_{N_t})\|_F - \bar{\varepsilon}^*| < \varepsilon,$$

for all μ satisfying $d(\mu, \mu_{p_j}) < \delta$. By taking $\varepsilon = \bar{\varepsilon}^*$ and $\delta = r_j$ we obtain that

$$\varepsilon_{\mu, \mu_{p_j}, K_{POD}}^{HF} < 2\bar{\varepsilon}^* \leq \bar{\varepsilon}, \quad \forall \mu \in \mathcal{B}(\mu_{p_j}, r_j),$$

which completes the proof. \square

The theoretical result allows to compute the reduced-order model error at $\mu = \mu_{p_j}$ at certain parametric configurations in the proximity of μ_p . Moreover, one may be able to make statements about the degree of smoothness of the solution in the parametric space, therefore placing a lower limit on $\bar{\varepsilon}$. A particular case is when the reduced solution error is

monotonically decreasing with smaller distances $d(\mu, \mu_{p_j})$. A small radius $r_j > 0$ then can be simply obtained by sampling and computing the residuals of the high-fidelity model using the projected reduced-order model solution.

One can also test for linear behavior of the high-fidelity solution $\mathbf{x}(\mu, t)$ in a small neighborhood of μ_{p_j} . Another possible screening test consists in checking the derivatives $\frac{\partial \mathbf{x}}{\partial \mu}(\mu, t)$ at equally distributed parameter values across the neighborhood of μ_{p_j} . If the derivatives are small, then the high-fidelity solutions do not vary much inside the open ball and they can be well approximated in the reduced manifold spanned by $U_{\mu_{p_j}}$.

The decomposition construction process ends as soon as the entire parameter domain \mathcal{P} is covered with a union of overlapping balls $\mathcal{B}(\mu_{p_j}, r_j)$, $j = 1, \dots, M$, corresponding to different reduced order bases and local models

$$\mathcal{P} \subset \bigcup_{j=1}^M \mathcal{B}(\mu_{p_j}, r_j), \quad (6.3)$$

such that for each $j = 1, 2, \dots, M$ and $\forall \mu \in \mathcal{B}(\mu_{p_j}, r_j) \cap \mathcal{P}$, the error of the reduced-order model solution (5.2) satisfies $\varepsilon_{\mu, \mu_{p_j}, K_{POD}}^{HF} \leq \bar{\varepsilon}$. The number of balls M is finite only if the space of all high-fidelity solution over the entire parametric domain can be approximated with a finite number of low-dimensional linear subspaces. This extends the concept of a single global low-dimensional manifold [112, 113]. The cardinality of \mathcal{I} depends on the high-fidelity solution variability along the parameter space. In theory, less variability should lead to smaller values of M . For our reduced-order models to satisfy the accuracy threshold for the entire parametric domain, we have to select thresholds $\bar{\varepsilon}$ not smaller than the Kolmogorov K_{POD} -width [153] of the so-called solution manifold $\{\mathbf{x}(\mu, t), \mu \in \mathcal{P}, t \in \{t_i\}_{i=1}^{N_t}\}$.

This approach is inspired from the construction of unsteady local reduced-order models where the time domain is split in multiple regions [20, 23]. In this way the reduced basis dimension is kept small allowing for fast on-line simulations. The cardinality of \mathcal{I} is inversely proportional with the prescribed level of accuracy $\bar{\varepsilon}$. As the desired error threshold $\bar{\varepsilon}$ decreases, the decomposition changes since usually the radii r_j are expected to become smaller, and more balls are required to cover the parametric domain; i.e., M is increased.

The construction of the parametric domain decomposition (6.3) using the local parametric reduced-order models requires the following ingredients

1. The ability to probe the vicinity of $\mu_{p_j} \in \mathcal{P}$ and to efficiently estimate the level of error $\varepsilon_{\mu, \mu_{p_j}, K_{POD}}^{HF}$ (6.2).

2. The ability to find $r_j > 0$ such that $\varepsilon_{\mu, \mu_{p_j}, K_{POD}}^{HF} \leq \bar{\varepsilon}$, for all $\mu \in \mathcal{B}(\mu_{p_j}, r_j) \cap \mathcal{P}$.
3. The ability to identify the location of a new $\mu_{p_\ell}^*$ (for the construction of a new local reduced-order model) given the locations of the previous local parameters μ_{p_j} , $j = 1, \dots, \ell - 1$, so that

$$\begin{aligned} \mathcal{B}(\mu_{p_\ell}^*, r_\ell) &\not\subset \left(\bigcup_{i=1}^{\ell-1} \mathcal{B}(\mu_{p_i}, r_i) \right), \\ \mathcal{B}(\mu_{p_\ell}^*, r_\ell) \cap \left(\bigcup_{i=1}^{\ell-1} \mathcal{B}(\mu_{p_i}, r_i) \right) &\neq \emptyset. \end{aligned} \tag{6.4}$$

The second condition in (6.4) assures that the decomposition will have no coverage gap; i.e., equation (6.3) is satisfied.

In practice, an approximated MP-LROM error model is used to sample the vicinity of μ_{p_j} and predicts the error for each sample parameter value μ . Based on these error predictions, we construct the ball $\mathcal{B}(\mu_{p_j}, r_j)$, or perhaps a larger set called a μ_{p_j} -feasible region, where the local reduced-order model is accurate to within the prescribed threshold $\bar{\varepsilon}$ according to the MP-LROM model. Since the approximated MP-LROM model has errors in its predictions, the precision is guaranteed only if the sum of the true reduced-order model error and approximated MP-LROM model error is smaller than $\bar{\varepsilon}$. For a one dimensional parametric domain, a greedy algorithm to be described in Subsection 6.3.1 is applied to identify the location of a new parametric configuration $\mu_{p_\ell}^*$ (for the construction of a new basis) depending on the locations of the previous μ_{p_i} , $i = 1, \dots, \ell - 1$. We seek to impose (6.4), so the entire parametric domain \mathcal{P} satisfies (6.3) after the decomposition construction is finished. Again this is not necessarily guaranteed since we employ an approximated MP-LROM error model for this task.

6.2.2 Combining available information for accurate local ROMs at arbitrary parametric configurations

We next solve another practical problem: given a collection of local reduced bases and operators and high fidelity trajectories computed at various locations in the parameter space, construct a hierarchy of the available bases and models producing the most accurate local parametric reduced-order solutions for an arbitrary viscosity parameter μ^* . For the parametric domain situated at the intersection of different feasible regions introduced in Section 6.2.1, we may improve the reduced solution accuracy by assigning a new reduced-

order model based on the already existing local bases or high-fidelity simulations. Moreover, if a hierarchy of local reduced bases, local reduced-order, and high-fidelity models is available for a parametric configuration μ^* , we can employ the top ranked bases and models to generate a new reduced-order model whose accuracy may be increased. This can be achieved by interpolation or concatenation of the underlying reduced bases or interpolation of the available high-fidelity solutions.

The POD method produces an orthogonal basis that approximately spans the state solution space of the model for a specific parameter configuration. Moving away from the initial parametric configuration may require the construction of new bases and reduced operators since the initial reduced-order model may not be accurate anymore. However, if states depend continuously on parameters, the POD basis constructed for one parameter configuration may approximately span the solution space at different parametric settings in a local vicinity.

Several methods to combine the available information to generate more accurate reduced-order models for arbitrary parameter configurations μ^* have been proposed in the literature. One is the interpolation of the available reduced-order bases $U_{\mu_{p_j}}$, $j = 1, \dots, M$. The parametric dependence of the bases has been modeled with various linear and nonlinear spatially-dependent interpolants.

Here we discuss different strategies that involve Lagrange interpolation of bases in the matrix space and in the tangent space of the Grassmann manifold. In addition we propose to concatenate the available reduced bases followed by an orthogonalization process, and to interpolate the solutions of the high fidelity model as means to derive the reduced-order basis for a parameter configuration μ^* .

Basis interpolation

Lagrange interpolation of bases Assuming the reduced manifold $\mathbf{U} : \mathcal{P} \rightarrow \mathbb{R}^{N_{\text{state}} \times K_{\text{POD}}}$ poses a continuous and linear dependency with respect to the parametric space, and if M discrete bases $U_{\mu_{p_j}} = \mathbf{U}(\mu_{p_j})$ have been already constructed for various parametric configurations μ_{p_j} , $j = 1, 2, \dots, M$, then a basis corresponding to the new configuration μ^* can be obtained using Lagrange's interpolation formula

$$U_{\mu^*} = \sum_{j=1}^M U_{\mu_{p_j}} L_j(\mu^*), \quad L_j(\mu^*) = \prod_{i \neq j} \frac{\mu^* - \mu_{p_i}}{\mu_{p_j} - \mu_{p_i}}. \quad (6.5)$$

It is worth mentioning that the resulting interpolated basis vectors are not orthogonal. One drawback of this approach is the lack of linear variation in the angles between pairs of

reduced subspaces [150] spanned by the reduced bases $U_{\mu_{p_j}}$. Differential geometry results can be employed to alleviate these deficiencies.

Grassmann manifold In the study proposed by Amsallem and Farhat [151], basis (matrix) interpolation was performed in the tangent space of the Grassmann manifold G at a carefully selected point S representing a subspace spanned by one of the available reduced bases. It has been shown that Grassmann manifold can be endowed with a differentiable structure [154, 155]; i.e., at each point S of the manifold a tangent space exists. The mapping from the manifold to the tangent space is called the logarithmic mapping, while the backward projection is referred to as exponential mapping [156]. According to [151], a new subspace S_{μ^*} , and its subsequent basis U_{μ^*} , associated with a new parameter μ^* , can be obtained by interpolating the reduced subspaces projections into the tangent space of the Grassmann manifold and then projecting back using the exponential mapping. The reduced subspaces $\{S_{\mu_{p_i}}\}_{i=1}^M$ are spanned by the already computed reduced bases $U_{\mu_{p_i}}$, $i = 1, \dots, M$. The steps required by this methodology [151] are described in the Algorithm 4. A graphical description of the method is provided in Figure 6.1.

According to [151], the subspace angle interpolation [150, 157] is identical to the interpolation in a tangent space to the Grassmann manifold of two reduced-order bases. Consequently the latter methodology can be viewed as a generalization of the former approach.

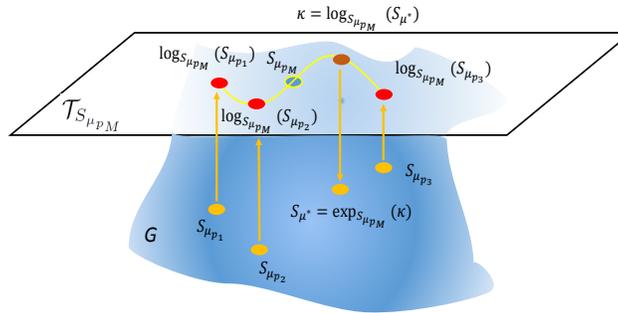


Figure 6.1: The description of the interpolation of four subspaces in a tangent space to a Grassmann manifold. The origin point of the interpolation is $S_{\mu_{p_M}}$ and $m = 3$.

Basis concatenation

Basis concatenation idea was introduced in [152] and emerged from the notion of a global basis [112, 113]. In the global strategy, the existing high-fidelity snapshots corresponding

Algorithm 4 Interpolation in a tangent space to a Grassmann manifold [151]

- 1: Select a point $S_{\mu_{p_M}}$ of the manifold to represent the origin point for the interpolation spanned by the basis $U_{\mu_{p_M}}$.
- 2: The tangent space $\mathcal{T}_{S_{\mu_{p_M}}}$ and the subspaces $\{S_{\mu_{p_i}}\}_{i=1}^m$ are considered, with $m \leq M - 1$. Each point $S_{\mu_{p_i}}$ sufficiently close to $S_{\mu_{p_M}}$ is mapped to a point of $\mathcal{T}_{S_{\mu_{p_M}}}$, using the logarithm map $\log_{S_{\mu_{p_M}}}$ [156]. The bases spanning the tangent space points $\log_{S_{\mu_{p_M}}}(S_{\mu_{p_i}})$ are computed by

$$(I - U_{\mu_{p_M}} U_{\mu_{p_M}}^T) U_{\mu_{p_i}} (U_{\mu_{p_M}} U_{\mu_{p_i}})^{-1} = R_i \Lambda_i Q_i^T \text{ (SVD factorization),}$$

$$\Gamma_{\mu_{p_i}} = R_i \tan^{-1}(\Lambda_i) Q_i^T.$$

- 3: Each entry of the matrix Γ_{μ^*} associated with the target parameter μ^* is computed by interpolating the corresponding entries of the matrices $\Gamma_{\mu_{p_i}} \in \mathbb{R}^{N_{state} \times K_{POD}}$ associated with the parameter points μ_{p_i} , $i = 1, \dots, m$. A univariate or multivariate Lagrange interpolation may be chosen similar with the one introduced in (6.5).
- 4: The matrix Γ_{μ^*} representing a point in the tangent space $\mathcal{T}_{S_{\mu_{p_M}}}$ is mapped to a subspace S_{μ^*} on the Grassmann manifold spanned by a matrix U_{μ^*} using the exponential map [156]

$$\Gamma_{\mu^*} = R^* \Lambda^* Q^{*T} \text{ (SVD factorization),}$$

$$U_{\mu^*} = U_{\mu_{p_M}} Q^* \cos(\Lambda^*) + R^* \sin(\Lambda^*).$$

to various parameter configurations are collected in a single snapshot matrix and then a matrix factorization is performed to extract the most energetic singular vectors. This global basis is then used to build reduced-order models for parameter values not included in the initial snapshots set.

Assuming $\mathbf{X}_{\mu_{p_1}}, \mathbf{X}_{\mu_{p_2}}, \dots, \mathbf{X}_{\mu_{p_M}} \in \mathbb{R}^{N_{\text{state}} \times N_t}$ are the snapshots corresponding to M high-fidelity model trajectories, the following error estimate holds [100, Proposition 2.16]:

$$\bar{\mathbf{X}} = [\mathbf{X}_{\mu_{p_1}} \cdots \mathbf{X}_{\mu_{p_M}}] = \bar{U} \Lambda \bar{V}^T \text{ (SVD factorization),} \quad (6.6a)$$

$$\|\bar{\mathbf{X}} - \bar{U}^d \underline{\mathbf{X}}\|_F^2 = \sum_{i=K_{POD}+1}^{N_t} \lambda_i^2 = \mathcal{O}(\lambda_{K_{POD}+1}^2), \quad \bar{U}_{ij}^d = \bar{U}_{ij}, \quad i = 1, 2, \dots, N_{\text{state}}, \quad j = 1, 2, \dots, K_{POD}, \quad (6.6b)$$

where λ_i is the i^{th} singular value of $\bar{\mathbf{X}}$, and $\underline{\mathbf{X}} = [\bar{U}^d]^T \bar{\mathbf{X}} \in \mathbb{R}^{K_{POD} \times (MN_t)}$. In practice, usually $N_t < N_{\text{state}}$, so the snapshot matrices are rank deficient. In this case and when the reduced-order bases $U_{\mu_{p_1}}, \dots, U_{\mu_{p_M}}$, corresponding to the trajectories $\mu_{p_1}, \dots, \mu_{p_M}$, are available, we can construct a basis \hat{U} by simply concatenating columns of $U_{\mu_{p_1}}, \dots, U_{\mu_{p_M}}$ such that the accuracy level in (6.6) is preserved.

Proposition 6.2.2. Consider the following SVD of snapshots matrices $\mathbf{X}_{\mu_{p_1}}, \dots, \mathbf{X}_{\mu_{p_M}} \in \mathbb{R}^{N_{\text{state}} \times N_t}$

$$\mathbf{X}_{\mu_{p_j}} = U_{\mu_{p_j}} \Lambda_j V_{\mu_{p_j}}^T, \quad j = 1, 2, \dots, M, \quad (6.7)$$

with $N_t < N_{\text{state}}$ and $\text{rank}(\mathbf{X}_{\mu_{p_i}}) < N_t$, $i = 1, 2, \dots, M$. Then, there exist positive integers

$K_{POD}^1, \dots, K_{POD}^M$, and

$\hat{\mathbf{X}} \in \mathbb{R}^{(\sum_{i=1}^M K_{POD}^i) \times (MN_t)}$, such that $\bar{\mathbf{X}}$ defined in (6.6) satisfies

$$\|\bar{\mathbf{X}} - \hat{U} \hat{\mathbf{X}}\|_F^2 \leq \mathcal{O}(\lambda_{K_{POD}+1}^2), \quad (6.8)$$

where $\lambda_{K_{POD}+1}$ is the $(K_{POD}+1)$ singular value of snapshots matrix $\bar{\mathbf{X}}$, and $\hat{U} = [U_{\mu_{p_1}}^d \cdots U_{\mu_{p_M}}^d] \in \mathbb{R}^{N_{\text{state}} \times (\sum_{i=1}^M K_{POD}^i)}$, $[U_{\mu_{p_l}}^d]_{ij} = [U_{\mu_{p_l}}]_{ij}$, $i = 1, 2, \dots, N_{\text{state}}$, $j = 1, 2, \dots, K_{POD}^l$, $l = 1, 2, \dots, M$.

Proof. Since $\mathbf{X}_{\mu_{p_1}}, \dots, \mathbf{X}_{\mu_{p_M}} \in \mathbb{R}^{N_{\text{state}} \times N_t}$ are rank deficient matrices, there exist at least M positive integers $K_{POD}^1, \dots, K_{POD}^M$, such that the singular values associated with $\mathbf{X}_{\mu_{p_1}}, \dots, \mathbf{X}_{\mu_{p_M}}$ satisfy

$$[\lambda_{K_{POD}^j+1}^j]^2, \dots, [\lambda_{K_{POD}^j+1}^M}]^2 \leq [\lambda_{K_{POD}+1}]^2, \quad \forall K_{POD} = 0, \dots, N_t - 1. \quad (6.9)$$

From [100, Proposition 2.16] and (6.7) we have the following estimates

$$\|\mathbf{X}_{\mu_{p_j}} - U_{\mu_{p_j}}^d \underline{\mathbf{X}}_{\mu_{p_j}}\|_F^2 = \sum_{i=K_{POD}^j+1}^{N_t} [\lambda_i^j]^2 = \mathcal{O}([\lambda_{K_{POD}^j+1}^j]^2) \leq \mathcal{O}([\lambda_{K_{POD}+1}]^2), \quad (6.10)$$

where λ_i^j is the i^{th} singular value of $\mathbf{X}_{\mu_{p_j}}$ and $\underline{\mathbf{X}}_{\mu_{p_j}} = [U_{\mu_{p_j}}^d]^T \mathbf{X}_{\mu_{p_j}} \in \mathbb{R}^{K_{POD}^j \times N_t}$, for $j = 1, \dots, M$.

By denoting

$$\hat{\mathbf{X}} = \begin{bmatrix} \mathbf{X}_{\mu_{p_1}} \mathbf{0}_1 & \cdots & \cdots & \mathbf{0}_1 \\ \mathbf{0}_2 & \mathbf{X}_{\mu_{p_2}} \mathbf{0}_2 & \cdots & \mathbf{0}_2 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_M & \cdots & \cdots & \mathbf{0}_M & \mathbf{X}_{\mu_{p_M}} \end{bmatrix},$$

where the null matrix $\mathbf{0}_j$ belongs to $\mathbb{R}^{K_{POD}^j \times N_t}$, $j = 1, \dots, M$, we have

$$\begin{aligned} \|\bar{\mathbf{X}} - \hat{U} \hat{\mathbf{X}}\|_F^2 &= \|[\mathbf{X}_{\mu_{p_1}} \cdots \mathbf{X}_{\mu_{p_M}}] - [U_{\mu_{p_1}}^d \underline{\mathbf{X}}_{\mu_{p_1}} \cdots U_{\mu_{p_M}}^d \underline{\mathbf{X}}_{\mu_{p_M}}]\|_F^2 \leq \\ &\left(\sum_{i=1}^M \|\mathbf{X}_{\mu_{p_i}} - U_{\mu_{p_i}}^d \underline{\mathbf{X}}_{\mu_{p_i}}\|_F \right)^2 = \sum_{i=1}^M \sum_{j=1}^M \|\mathbf{X}_{\mu_{p_i}} - U_{\mu_{p_i}}^d \underline{\mathbf{X}}_{\mu_{p_i}}\|_F \cdot \\ &\|\mathbf{X}_{\mu_{p_j}} - U_{\mu_{p_j}}^d \underline{\mathbf{X}}_{\mu_{p_j}}\|_F \leq \mathcal{O}([\lambda_{K_{POD}+1}]^2). \end{aligned}$$

□

By assuming that $\text{rank}(\mathbf{X}_{\mu_{p_i}}) < N_t$, for all $j = 1, \dots, M$, imposes N_t as an upper limit for all the positive integers K_{POD}^j , $j = 1, \dots, M$. This assumption is not unrealistic since typically the snapshots matrix stores correlated data and therefore contains linearly dependent columns. For linearly independent matrices, the precision is controlled by the spectra of snapshots matrices $\mathbf{X}_{\mu_{p_i}}$, $i = 1, \dots, M$, but there is no guarantee that the expansion based on the concatenated basis \hat{U} can provide similar accuracy precision as the expansion based on \bar{U} (6.6) for all $K_{POD} = 1, 2, \dots, N_t$.

In practice usually $\sum_{i=1}^M K_{POD}^i$ is larger than K_{POD} , thus more bases functions are required to form \hat{U} to achieve a similar level of precision as in (6.6) where \bar{U} is built using a global singular value decomposition. According to [158], the faster approach to compute the left singular vectors and singular values only is to apply a QR factorization followed by a R-bidiagonalization [159]. The R-SVD decomposition of a matrix of dimension $N_{\text{state}} \times N_t$ has a computational complexity of $\mathcal{O}(4N_{\text{state}}^2 N_t + 13N_t^3)$. As such, the decomposition of matrix $\bar{\mathbf{X}}$ requires $\mathcal{O}(4MN_{\text{state}}^2 N_t + 13M^3 N_t^3)$ operations, whereas all combined singular value decompositions of matrices $\mathbf{X}_{\mu_{p_i}}$, $i = 1, \dots, M$, have a computational complexity of $\mathcal{O}(4MN_{\text{state}}^2 N_t + 13MN_t^3)$. This estimation suggests that the matrix factorization of $\bar{\mathbf{X}}$ is more computationally demanding. However, the first term $4MN_{\text{state}}^2 N_t$ often dominates as $N_t \ll N_{\text{state}}$ in practice, leading to almost similar computational times. The off-line stage of the concatenation method may also include the application of a Gram-Schmidt-type algorithm to orthogonalize the overall set of vectors in \hat{U} .

While the Lagrange interpolation of bases mixes the different energetic singular vectors in an order dictated by the singular values magnitude, this strategy concatenates the dominant singular vectors for each case and preserves their structure.

Interpolation of high-fidelity model solutions

The method discussed herein assumes that the model solution depends continuously on the parameters. Thus it is natural to consider constructing the basis for a parameter configuration μ^* by interpolating the existing high-fidelity model solutions associated with various parameter settings, and then performing the SVD of the interpolated results. For example, the Lagrange solution interpolant is given by

$$\mathbf{X}_{\mu^*} = \sum_{j=1}^M \mathbf{X}_{\mu_{p_j}} L_j(\mu^*), \quad (6.11)$$

where $\mathbf{X}_{\mu_{p_j}} \in \mathbb{R}^{N_{\text{state}} \times N_t}$ is the model solution corresponding to parameter μ_{p_j} and the interpolation polynomials are defined in (6.5).

A new basis is constructed from the interpolated model solution (6.11) for the new parametric configuration μ^* . From computational point of view the complexity of the off-line stage of the solution interpolation method (6.11) is smaller than in the case of the bases concatenation and interpolation approaches. Only one singular value decomposition is required in contrast with the multiple factorizations needed in the latter two strategies where the involved matrices have the same size $N_{\text{state}} \times N_t$. Having only N_t snapshots the size of the outcome basis should be smaller than in the case of basis concatenation approach.

6.3 Numerical experiments

We illustrate the application of the MP-LROM error model to design decompositions of the parametric domain for a one-dimensional Burgers model. Moreover, the database of accurate local bases, local reduced-order and high-fidelity models are used to enhance the accuracy of the reduced-order model for a different parametric configuration μ^* . For each of the applications introduced, we present in detail the proposed solution approaches and the corresponding numerical results. The 1D-Burgers model proposed herein is characterized by one scalar viscosity parameter which was described in 5.4.1.

6.3.1 Designing the decomposition of the parametric domain

In the sequel, we present some of the details associated with the construction of ANN model approximating the MP-LROM error model (5.5). A full description is available in [149]. The whole data set contains 12000 samples including 10 and 100 equally distributed values of μ_p and μ over the entire parameter region; i.e., $\mu_p \in \{0.1, 0.2, \dots, 1\}$ and $\mu \in \{0.01, \dots, 1\}$, 12 reduced basis dimensions K_{POD} spanning the interval $\{4, 5, \dots, 14, 15\}$ and the reduced-order model logarithm of errors $\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$. The trained models were assessed by the common five-fold cross-validation technique using the Burger's model. The reported accuracy was 0.004004 [149, table 3].

The whole data set contains 12000 samples. It's worth to note that, only one high-fidelity model simulation is required for computing the reduced solutions errors for the parametric configuration μ using reduced-order models of various K_{POD} and various bases and reduced operators constructed at μ_p . As such, only 100 high-fidelity simulations were needed to construct the entire data set due also to the fact that the set of parameters μ_p is a subset of the selected parameters μ . High-fidelity simulations are used to accurately calculate the errors associated with the existing reduced-order models for parametric configurations μ .

Next we seek to build a decomposition of the viscosity domain $[0.01, 1]$ for the 1D-Burgers model using the ANN MP-LROM error model.

As discussed in Section 6.2, we take the following steps. First we identify “ μ_p -feasible” intervals $[d_\ell, d_r]$ in the parameter space such that local reduced-order model depending only on the high-fidelity trajectory at μ_p is accurate to within the prescribed threshold for any $\mu \in [d_\ell, d_r]$. Second, a greedy algorithm generates the decomposition

$$[0.01, 1] \subset \bigcup_{i=1}^M [d_\ell^i, d_r^i], \quad (6.12)$$

by covering the parameter space with a union of μ_{p_i} feasible intervals, where each μ_{p_i} -feasible interval is characterized by an error threshold $\bar{\varepsilon}_i$ (which can vary from one interval to another). This relaxation is suggested since for intervals associated with small parameters μ_{p_i} , it is difficult to achieve small reduced-order models errors similar to those obtained for larger parametric configurations.

For the existing reduced basis methods a global reduced-order model depending on multiple high-fidelity trajectories is usually constructed. In contrast, our approach uses the ANN MP-LROM error model to decompose the parameter space into smaller regions where the local reduced-order model solutions are accurate to within some tolerance levels. Since

the local bases required for the construction of the local reduced-order models depend on only a single full simulation, the dimension of the POD subspace is small, leading to lower on-line computational complexity.

Construction of a μ_p -feasible interval

The ANN MP-LROM error model can accurately estimate $\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ (5.4) associated with reduced-order models. Thus we can employ it to establish a range of viscosity parameters around μ_p such that the reduced-order solutions depending on U_{μ_p} satisfy some desired accuracy level. More precisely, starting from parameter μ_p , a fixed POD basis dimension and a tolerance error $\log \bar{\varepsilon}$, we are searching for an interval $[d_l, d_r]$ such that the estimated prediction $\widehat{\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}}$ of the true error $\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}$ (5.4) meets the requirement

$$\widehat{\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}} < \log \bar{\varepsilon}, \forall \mu \in [d_l, d_r]. \quad (6.13)$$

Our proposed strategy makes use of a simply incremental approach by sampling the vicinity of μ_p to account for the estimated errors $\widehat{\log \varepsilon_{\mu, \mu_p, K_{POD}}^{HF}}$ forecasted by the ANN MP-LROM error model. A grid of new parameters μ is built around μ_p and the error models predict the errors outward of μ_p . Once the error models outputs are larger than the prescribed error $\log \bar{\varepsilon}$, the previous μ satisfying the constraint (6.13) is set as d_l , for $\mu < \mu_p$ or d_r for $\mu > \mu_p$.

Figure 6.2 illustrates the range of parameters predicted by the ANN MP-LROM model against the true feasible interval and the results show good agreement. For this experiment we set $\mu_p = 0.7$, dimension of POD subspace $K_{POD} = 9$ and $\bar{\varepsilon} = 10^{-2}$. Values of $\mu = \mu_p \pm 0.001 \cdot i$, $i = 1, 2, \dots$ are passed to the ANN MP-LROM model. The average range of parameters obtained over five different configurations with ANN is $[0.650, 0.780]$. We trained the models with 80% random split of the data set and test it over the fixed test set of Figure 6.2. For this design, the true range of parameters is $[0.650, 0.785]$ underlying the predicting potential of MP-LROM model built using ANN technique.

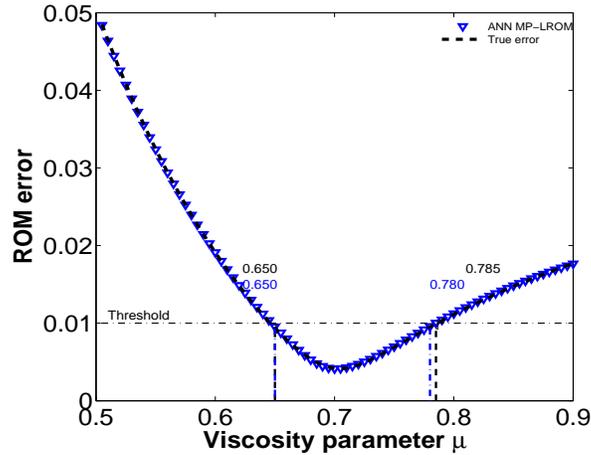


Figure 6.2: The average range of parameter μ obtained with MP-LROM models for $K_{POD} = 9$ and $\mu_p = 0.7$. The desired accuracy is $\bar{\varepsilon} = 10^{-2}$. The numbers represent the left and the right edges of the predicted vs the true feasible intervals.

The decomposition of the parametric domain as a union of μ_p -feasible intervals

A union of different μ_{p_k} -feasible intervals can be designed to cover a general entire 1D-parametric domain $[A, B]$. Once such construction is available, it will allow for reduced-order simulations with a-priori error quantification for any value of viscosity parameter $\mu \in [A, B]$.

A greedy strategy based on the ANN MP-LROM error model is described in Algorithm 5 and its output is a collection of feasible intervals $\cup_{k=1}^n [d_l^k, d_r^k] \supset [A, B]$. After each iteration k of the algorithm, a μ_{p_k} -feasible interval $[d_l^k, d_r^k]$ is constructed. Each interval is associated with some accuracy threshold $\bar{\varepsilon}_k$. For small viscous parametric values we found out that designing μ_{p_k} -feasible intervals associated with higher precision levels (very small thresholds $\bar{\varepsilon}_k$) is impossible since the dynamics of parametric 1D-Burgers model solutions change dramatically with smaller viscosity parameters. In consequence we decided to let $\bar{\varepsilon}_k$ vary along the parametric domain to accommodate the solution physical behaviour. Thus a small threshold $\bar{\varepsilon}_0$ will be initially set and as we will advance charting the parameter domain $[A, B]$ from right to left, the threshold $\bar{\varepsilon}_k$ will be increased.

The algorithm starts by selecting the first centered parameter μ_{p_0} responsible for basis generation. It can be set to $\mu_{p_0} = B$ but may take any value in the proximity of B , $\mu_{p_0} \leq B$. This choice depends on the variability of parametric solutions in this domain region and

by selecting μ_{p_0} to differ from the right edge of the domain, the number n of the feasible intervals should decrease.

The next step is to set the threshold $\bar{\varepsilon}_0$ along with the maximum permitted size of the initial feasible interval to be constructed. This is set to $2 \cdot r_0$, thus r_0 can be referred as the interval radius. Along with the radius, the parameter Δr will decide the maximum number of ANN MP-LROM model calls employed for the construction of the first μ_{p_0} -feasible interval. While the radius is allowed to vary during the algorithm iterations, Δr is kept constant. Finally the dimension of POD basis has to be selected together with three parameters β_1 , β_2 and β_3 responsible for changes in the threshold and radius and selecting a new parameter location μ_{p_k} encountered during the procedure.

Next the algorithm starts the construction of the μ_{p_0} feasible interval. The process is described in the top part of Figure 6.3(a). Then we are sampling the vicinity of μ_{p_0} for equally distributed parameters μ and compute the ANN MP-LROM model predictions. The sampling process and the comparison between the predicted errors $\widehat{\log \varepsilon_{\mu, \mu_{p_0}, K_{POD}}^{HF}}$ and $\log \bar{\varepsilon}_0$ are depicted in Figure 6.3(a). A green vertical segment indicates that the estimated error satisfies the threshold; i.e., $\widehat{\log \varepsilon_{\mu, \mu_{p_0}, K_{POD}}^{HF}} < \log \bar{\varepsilon}_0$, whereas the red segment indicates the opposite. The left limit of the μ_{p_0} -feasible interval is obtained when either $\mu > \mu_{p_0} - r_0$ or $\widehat{\log \varepsilon_{\mu, \mu_{p_0}, K_{POD}}^{HF}} > \log \bar{\varepsilon}_0$. The left limit d_l^0 , denoted with a green dashed line in Figure 6.3(a), is set equal to the last parameter μ such that $\widehat{\log \varepsilon_{\mu, \mu_{p_0}, K_{POD}}^{HF}} \leq \log \bar{\varepsilon}_0$.

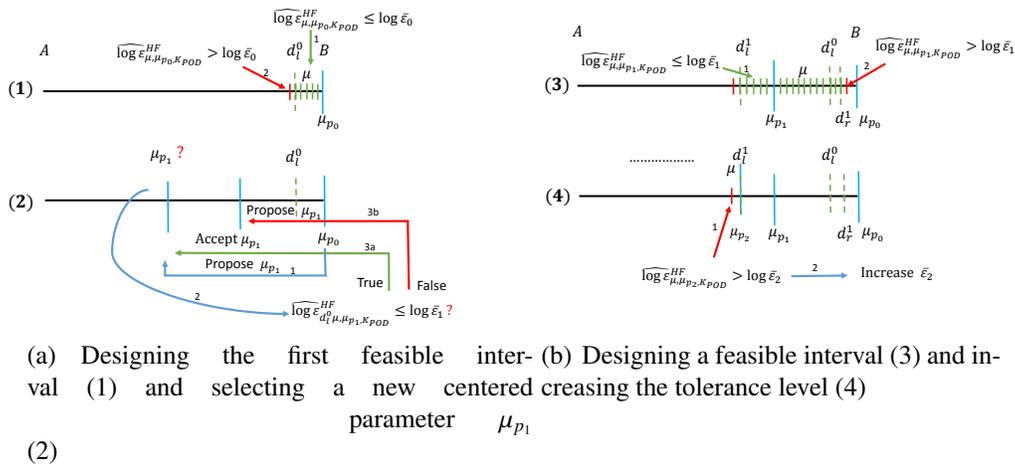


Figure 6.3: A description of the most important stages of the parameter domain decomposition algorithm. The arrows describe the internal steps for each stage initiated in the order depicted by the arrows' indices.

The next step searches for a centered parameter $\mu_{p_{k+1}}$ and this process is described at the bottom of the Figure 6.3(a) for $k = 0$. The centered parameter $\mu_{p_{k+1}}$ is first proposed based on an empirical formula described in line 25 of Algorithm 5. This formula depends on the current centered parameter μ_{p_k} , the number of tested parameters μ during the construction of μ_{p_k} -feasible interval, parameters Δr and β_3 . Next, the algorithm checks if the following constraint is satisfied

$$[d_l^{k+1}, d_r^{k+1}] \cap \left(\bigcup_{i=1}^k [d_l^i, d_r^i] \right) \neq \emptyset, \quad (6.14)$$

without taking into account the error in the ANN MP-LROM model prediction. This is achieved by comparing the error model prediction $\widehat{\log \varepsilon_{d_l^k, \mu_{p_{k+1}}, K_{POD}}^{HF}}$ and threshold $\log \bar{\varepsilon}_{k+1}$ (see instruction 27 and bottom of Figure 6.3(a) for $k = 0$). If the predicted error is smaller than the current threshold, assuming a monotonically increasing error with larger distances $d(\mu, \mu_{p_{k+1}})$, the reduced-order model solutions should satisfy the accuracy threshold for all $\mu \in [\mu_{p_{k+1}}, d_l^k]$. In consequence the equation (6.14) will be satisfied for the current $\mu_{p_{k+1}}$, if we set $r_{k+1} = \mu_{p_{k+1}} - d_l^k$ (see instruction 30). In the case the error estimate is larger than the present threshold, the centered parameter $\mu_{p_{k+1}}$ is updated to the middle point between old $\mu_{p_{k+1}}$ and d_l^k (see also the bottom of Figure 6.3(a)). For the situation where the monotonic property of the error does not hold in practice, a simply safety net is used at instruction 12.

The instructions between lines 5 and 21 generate the μ_{p_k} -feasible interval, for the case when the current centered parameter $\mu_{p_k} \neq d_l^{k-1}$ (see top part of Figure 6.3(b) for $k = 1$). Here by *int* we refer to the integer part of a real number. We used the Matlab command `floor` for the implementation. For situation when $\mu_{p_k} = d_l^{k-1}$ (see bottom of Figure 6.3(b) for $k = 2$), the threshold has to be increased (by setting $\bar{\varepsilon}_k = \beta_1 \bar{\varepsilon}_k$ at line 23), since the reduced-order model solutions can not satisfy the desired precision according to the predicted errors. In consequence, β_1 has to be selected larger than 1. The need for relaxing the threshold suggests that the greedy search is currently operating into a parametric region where only a slight change in the parameter μ away from μ_{p_k} leads to predicted ROM errors larger than the current threshold. Relaxing the threshold and decreasing the radius size (select $\beta_2 < 1$ in line 23 of Algorithm 3) can be used as a strategy to identify a feasible region for the current centered parameter μ_{p_k} . Similarly, relaxing the threshold and expanding the search ($\beta_2 > 1$) could also represent a viable strategy. However, expanding the search in a parametric regime with large changes in the model dynamics, even if the threshold was relaxed, may lead to useless evaluations of the expressions in lines 7 and 18 of the Algorithm 3. Thus β_2 should be selected smaller than 1. Once the feasible region is obtained, the radius r_k is reset to the initial value r_0 (see line 25 of the Algorithm 3). By selecting $\beta_3 > 1$, the computational complexity of Algorithm 3 is decreased since the first

proposal of the new centered parameter $\mu_{p_{k+1}}$ will always be smaller than the left limit d_l^k of the current feasible interval. The entire algorithm stops when $\mu_{p_{k+1}} \leq A$.

For our experiments we set $A = 0.01$, $B = 1$, $\bar{\epsilon}_0 = 10^{-2}$, $\Delta r = 5 \times 10^{-3}$, $r_0 = 0.5$, $K_{POD} = 9$, $\beta_1 = 1.2$, $\beta_2 = 0.9$ and $\beta_3 = 1.4$. We initiate the algorithm by setting $\mu_{p_0} = 0.87$, and the first feasible interval $[0.7700, 1]$ is obtained. Next the algorithm selects $\mu_1 = 0.73$ with the associated range of $[0.6700, 0.8250]$ using the same initial threshold level. As we cover the parametric domain from right to left; i.e., selecting smaller and smaller parameters μ_{p_k} , the algorithm enlarges the current threshold $\bar{\epsilon}_k$, otherwise the error model predictions would not satisfy the initial precision. We continue this process until we get the threshold 6.25 with $\mu_{32} = 0.021$ and the corresponding feasible interval $[0.01, 0.039]$. The generated decomposition is depicted in Figure 6.4 where the associated threshold varies with the parameter change.

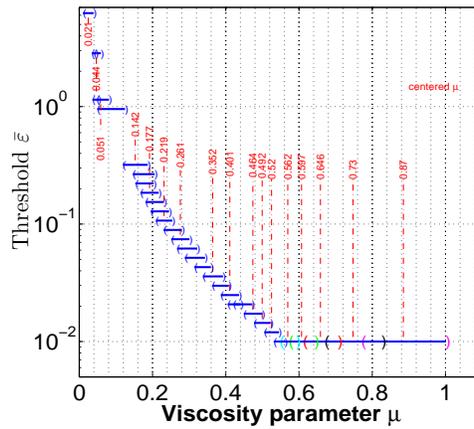


Figure 6.4: The diffusion parametric domain decomposition defining the local feasible intervals and their corresponding errors. Associated with one feasible interval there is a centered parameter μ_p high-fidelity trajectory that guides the construction of a reduced basis and operators such that the subsequent reduced-order model solutions along this interval are accurate to within the threshold depicted by the Y-axis labels.

6.3.2 Combining available information for accurate local ROMs at arbitrary parametric configurations

We now use the available database of local bases and reduced-order models resulted from the parameter decomposition to construct a new POD basis for a different parameter con-

Algorithm 5 Generation of 1D-parametric domain decomposition for reduced-order models usage. Extension to multi-dimensional parametric space is subject to future research.

- 1: Select μ_{p_0} as the right edge of the parameter interval, i.e. $\mu_{p_0} = B$.
 - 2: Set error threshold $\hat{\epsilon}_0$, step size Δr for selection of sampling parameters μ , the maximum search radius r_0 , dimension of POD basis K_{POD} and β_1 , β_2 and β_3 .
 - 3: Set $k = 0$.
 - 4: WHILE $\mu_{p_k} \geq A$ Do
 - 5: FOR $i = 1$ to $\text{int}(\frac{r_k}{\Delta r}) + 1$
 - 6: Set $\mu = \mu_{p_k} + i\Delta r$
 - 7: IF $(\phi(\mu, \mu_{p_k}, K_{POD}) > \log \bar{\epsilon}_k \text{ OR } \mu > B)$ THEN
 - 8: Set $d_r^k = \mu_{p_k} + (i - 1)\Delta r$. EXIT.
 - 9: END IF
 - 10: END FOR
 - 11: IF $k > 0$ THEN
 - 12: IF $d_r^k < d_l^{k-1}$ THEN
 - 13: $\mu_{p_k} = \frac{\mu_{p_k} + d_l^{k-1}}{2}$. GOTO 5.
 - 14: END IF
 - 15: END IF
 - 16: FOR $j = 1$ to $\text{int}(\frac{r_k}{\Delta r}) + 1$
 - 17: Set $\mu = \mu_{p_k} - j\Delta r$
 - 18: IF $(\phi(\mu, \mu_{p_k}, K_{POD}) > \log \bar{\epsilon}_k \text{ OR } \mu < A)$ THEN
 - 19: Set $d_l^k = \mu_{p_k} - (j - 1)\Delta r$. EXIT.
 - 20: END IF
 - 21: END FOR
 - 22: IF $(i = 1)$.OR. $(j = 1)$ THEN
 - 23: Set $\bar{\epsilon}_k = \beta_1 \cdot \bar{\epsilon}_k$; $r_k = \beta_2 \cdot r_k$; GOTO 5.
 - 24: ELSE
 - 25: $\mu_{p_{k+1}} = \mu_{p_k} - \beta_3(j - 1)\Delta r$; $\bar{\epsilon}_{k+1} = \bar{\epsilon}_k$; $r_{k+1} = r_0$.
 - 26: END IF
 - 27: WHILE $\phi(d_l^k, \mu_{p_{k+1}}, K_{POD}) > \log \bar{\epsilon}_{k+1}$ DO
 - 28: $\mu_{p_{k+1}} = \frac{\mu_{p_{k+1}} + d_l^k}{2}$.
 - 29: END WHILE
 - 30: Set $r_{k+1} = \mu_{p_{k+1}} - d_l^k$.
 - 31: $k = k + 1$.
 - 32: END WHILE
-

figuration μ^* leading to more accurate reduced-order model solution. The numerical experiments described here focus on the construction of a POD basis for $\mu^* = 0.35$ by combining the basis data or high fidelity results data available from existing simulations with $\mu_{p_1} = 0.3$ and $\mu_{p_2} = 0.4$.

The performances of the discussed methods (bases concatenation, Lagrange interpolation of bases in the matrix space and in the tangent space of the Grassmann manifold, Lagrange interpolation of high-fidelity solutions) are shown in the case of three main experiments: variation in the final time t_f , in the non-linear advection coefficient ν and POD basis dimension. The bases concatenation method followed by orthogonalization is referred as Gram-Schmidt whereas the Lagrange interpolation of bases and high-fidelity solutions are simply described as "Lagrange bases" and "Lagrange sol" in the legends of Figures 6.5 and 6.6. All the methods employ reduced bases having the same dimension. Only even basis dimensions are utilized and the concatenation method combines half of the modes of each involved bases. This strategy was selected based on the spectra of the snapshots matrices associated with $\mu_{p_1} = 0.3$ and $\mu_{p_2} = 0.4$. More precisely, we selected the first $\frac{K_{POD}}{2}$ singular vectors from each of the bases since for all our experiments $\lambda_{\frac{K_{POD}}{2}}^{\mu_{p_1}} > \lambda_{\frac{K_{POD}}{2}+1}^{\mu_{p_2}}$ and $\lambda_{\frac{K_{POD}}{2}}^{\mu_{p_2}} > \lambda_{\frac{K_{POD}}{2}+1}^{\mu_{p_1}}$, where $\lambda_{\frac{K_{POD}}{2}+1}^{\mu_{p_1}}$ denotes the singular value corresponding to the first singular vector associated with trajectory μ_{p_1} not taken into account in the POD expansion. Of course this choice is not optimal. The optimal solution can be obtained by solving a combinatorial optimization problem, where the searched space has the size of $2K_{POD}$ chose K_{POD} .

The first two experiments scale the time and space and modify the linear and nonlinear characteristics of the model. For example, in the case of a tiny small final time and advection coefficient, the diffusion linear part represents the main dynamical engine of the model thus it behaves linearly. The results are compared against reduced-order models constructed using $U_{\mu_{p_1}}$ and $U_{\mu_{p_2}}$, respectively.

Figure 6.5 illustrates the Frobenius norm error between the high fidelity and reduced-order model solutions for the final time $t_f = 0.01$. Panel (a) presents the accuracy results as a function of the advection coefficient ν . Interpolating the high-fidelity solutions leads to the most accurate reduced-order model. For large advection coefficients all of the methods suffer accuracy losses. Among the potential explanations we include the constant dimension of the POD basis and the linear dependence on the viscosity parameter assumed by all of the methods in various forms. Figure 5.3 shows that the basis dimension must be increased as the advection coefficient decreases to maintain constant error.

Since the Grassmann manifold approach is a generalization of the subspace angle inter-

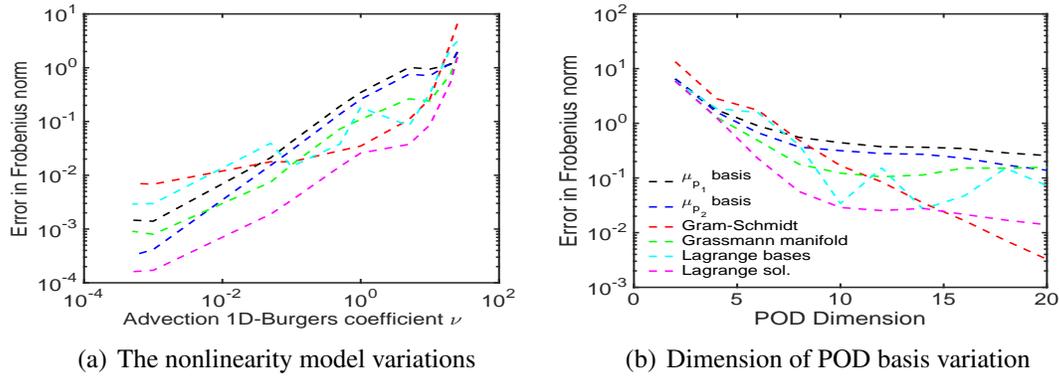


Figure 6.5: Strategies comparison for generation of accurate ROMs for a new viscosity parameter $\mu^* = 0.35$ and $t_f = 0.01$. In panel (a) K_{POD} is set to 14, whereas in panel (b) ν is set to 1.

polation method we decide to show only the results corresponding to the former method. While Lagrangian interpolation of the bases is performed in both matrix space and tangent space of the Grassmann manifold (shown in cyan and green), the later approach performs better in this scenario. The concatenation of bases using Gram-Schmidt algorithm was successful only for larger advection coefficients (red curve in Figure 6.5(a)), for a POD dimension set to 14.

Increasing the dimension of the basis enhances the so called Gram-Schmidt reduced-order model solution accuracy for $\nu = 1$ (see Figure 6.5(b)). For this case Lagrange interpolation in the matrix space shows better performances in comparison with the output of the Grassmann manifold approach.

Next we increase the nonlinearity characteristics of the model by setting the final time to $t_f = 1$ and Figure 6.6 illustrates the Frobenius norm errors as a function of the advection coefficient ν and POD dimension. The errors produced by the reduced-order model derived via Grassmann manifold method are similar with the ones obtained by the surrogate model relying on a POD basis computed via the Lagrange interpolation of the high-fidelity model solutions.

The Lagrange interpolation of bases in the matrix space is not successful as seen in both panels of Figure 6.6. Increasing the POD dimension to 20, the Gram-Schmidt approach enhances the accuracy of the solution (see Figure 6.6(b)), for $\nu = 1$.

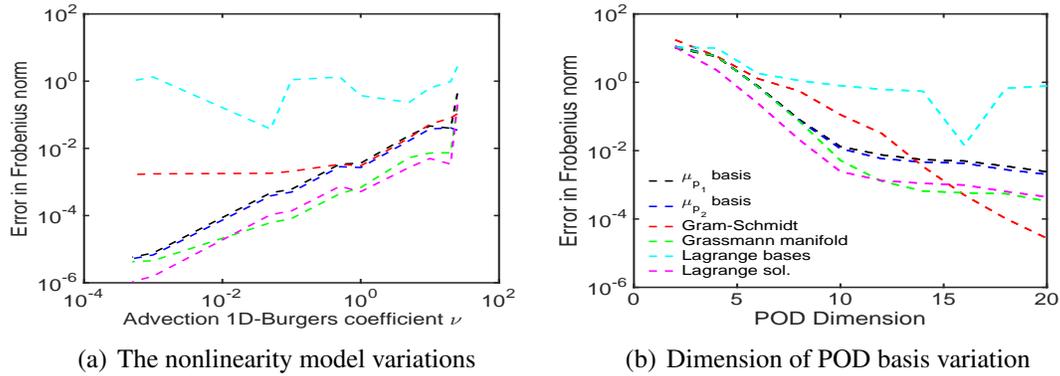


Figure 6.6: Strategies comparison for generation of accurate ROMs for a new viscosity parameter $\mu^* = 0.35$ and $t_f = 1$. In panel (a) K_{POD} is set to 14, whereas in panel (b) ν is set to 1.

6.4 Conclusions

This study develops a new and efficient methodology for constructing reduced-order surrogates for parametric high-fidelity models. We demonstrate the value of MP-LROM error model to generate accurate description of the local reduced-order models errors along the entire parametric domain.

The MP-LROM error prediction and parametric domain sampling are utilized to generate decompositions of the space of parameters into overlapping sub-regions. For each region an associated local reduced-order basis and operators are constructed using snapshots from a single representative high-fidelity trajectory. The procedure guarantees that, for all values of the parameters in a region, the high-fidelity model is approximated by the same local reduced-order surrogate within a feasible prescribed accuracy level. Under specific conditions, we derived a theoretical lower limit for the accuracy level of a local parametric reduced-order model associated with a parametric sub-region. The novel methodology was applied to a 1D-Burgers model. A decomposition of the viscosity parameter interval into sub-intervals with associated errors thresholds was performed.

Next we employed the hierarchy of local bases, local reduced-order and high-fidelity models producing the most accurate solutions for an arbitrary parameter configuration. Based on this hierarchy, three already existing methods involving bases interpolation and concatenation and high-fidelity model solutions interpolation were applied to enhance the quality of the associated reduced-order model solutions. Several experiments were per-

formed by scaling the time and space and modifying the nonlinear characteristics of the 1D-Burgers model. In most cases, interpolating the already existing high-fidelity trajectories generated the most accurate reduced-order models for a new viscous parameter revealing that the solution behavior over the parametric region under study can be linearly approximated. Lagrange interpolation of bases in the tangent space of the Grassmann manifold and concatenation of bases for larger reduced subspaces also showed good performances.

This work illustrates the rich potential of machine learning to impact the field of computational science and engineering. The new methodology ensures that a small number of reduced-order models can approximate well the high-fidelity model solutions for all parametric configurations. This greatly decreases overall computational costs since reduced-order models are expensive to construct. In the present work the parametric domain decomposition algorithm is applied to a one-dimensional parameter space.

Chapter 7

A Machine Learning Approach to Adaptive Covariance Localization

7.1 Introduction

Predicting the behavior of complex dynamical systems by computer simulation is crucial in numerous fields such as oceanography, glaciology, seismology, nuclear fusion, medicine, and atmospheric sciences, including weather forecasting and meteorology. Data assimilation (DA) is the set of methodologies that combine multiple sources of information about a physical system, with the goal of producing an accurate description of the state of that system. These sources of information include computer model outputs, observable measurements, and probabilistic representations of errors or noise. DA generates a best representation of the state of the system, called the analysis, together with the associated uncertainty. In numerical weather prediction the analysis state that can be used to initialize subsequent computer model runs that produce weather forecasts.

DA algorithms generally fall into one of two main categories, variational and statistical approaches. The variational approach for DA solves an optimization problem to generate an analysis state that minimizes the mismatch between model-based output and collected measurements, based on the level of uncertainty associated with each [160, 161, 162]. Algorithms in the statistical approach apply Bayes' theorem to describe the system state using a probability distribution conditioned by all available sources of information. A typical starting point for most of the algorithms in this approach is the Kalman filter (KF) [24, 25], which assumes that the underlying sources of errors are normally distributed, with known

means and covariances. Applying KF in large-scale settings is not feasible due to the intrinsic requirement of dealing with huge covariance matrices.

The EnKF [26, 27, 28] follows a Monte-Carlo approach to propagate covariance information, which makes it a practical approach for large-scale settings. Various flavors of EnKF have been developed [26, 27, 163, 164, 165, 166, 167], and have been successfully used for sequential DA in oceanographic and atmospheric applications.

EnKF carries out two main computational stages in every assimilation cycle, while operating on an ensemble of the system states to represent probability distributions. The “forecast” (prediction) stage involves a forward propagation of the system state from a previous time instance to generate an ensemble of forecast states at the current time. The “analysis” stage uses the covariance of the system states to assimilate observations and to generate an analysis ensemble, i.e., a Monte-Carlo representation of the posterior distribution of the model state conditioned by measurements.

In typical atmospheric applications the model state space has dimension $\sim 10^9 - 10^{12}$, and a huge ensemble is required to accurately approximate the corresponding covariance matrices. However, computational resources limit the number of ensemble members to 30 – 100, leading to “under-sampling” [28] and its consequences: filter divergence, inbreeding, and long-range spurious correlations [30, 29, 31, 32]. A lot of effort has been dedicated to solving the issue of under-sampling. Inbreeding and the filter divergence are alleviated by some form of inflation [168]. Long range spurious correlations are removed by covariance localization [169]. Covariance localization is implemented by multiplying the regression coefficient in the Kalman gain with a decaying distance-dependent function such as a Gaussian [31] or the Gaspari-Cohn fifth order piecewise polynomial [170].

Different localization techniques have been recently considered for different observation types, different type of state variables, or for an observation and a state variable that are separated in time. For horizontal localization, only a single GC function is used, and is tuned by finding the best value of its parameter (half the distance at which the GC function goes to zero). Tuning the localization for big atmospheric problems is very expensive. Similar challenges appear in the vertical localization [29].

Previous efforts for building adaptive algorithms for covariance localization includes the work of Anderson [171] based on a hierarchical ensemble filter. This approach adapts the localization radius by minimizing the root-mean-square difference of regression coefficients obtained by a group of ensemble filters. Adaptive localization by formulating the Kalman update step as a differential equation in terms of its ensemble members was proposed in [172], and computing localization as a power of smoothed ensemble sample correlation was discussed in [173, 174]. The approach proposed in [31] formulates

localization as a function of ensemble size and correlation between the observation and state variable. Correlation factors are obtained and applied as traditional localization for each pair of observation and state variable during assimilation. An Observing System Simulation Experiment (OSSE) algorithm is developed in [30, 175]. OSSE computes the localization radius of influence from a set of observation-state pairs by minimizing the root-mean-square (RMS) of the posterior ensemble mean compared to true model state. A probabilistic approach proposed in [176] defines the optimal radius of influence as the one that minimizes the distance between the Kalman gain using the localized sampling covariance and the Kalman gain using the true covariance. Further, the authors generalized this method for the case when the true covariance is unknown but it can be estimated probabilistically based on the ensemble sampling covariance. The relation between the localization length for domain localization and observation localization is investigated in [177]. This study concluded that the optimal localization length is linearly dependent on an effective local observation dimension given by the sum of the observation weights. In [175] two techniques for estimating the localization function are compared. The first approach is the Global Group Filter (GGF) which minimizes the RMS difference between the estimated regression coefficients using a hierarchical ensemble filter. The second approach is the Empirical Localization Function (ELF) that minimizes the RMSE difference between the true values of the state variables and the posterior ensemble mean. The ELF has smaller errors than the hand-tuned filter, while the GGF has larger errors than the hand-tuned counterpart.

In this study we propose to adapt covariance localization parameters using machine learning algorithms. Two approaches are proposed and discussed. In the *localization-in-time* method the radius of influence is held constant in space, but it changes adaptively from one assimilation cycle to the next. In the *localization-in-space-and-time* method the localization radius is space-dependent, and is also adapted for each assimilation time instance. The learning process is conducted off-line based on historical records such as reanalysis data, and the trained model is subsequently used to predict the proper values of localization radii in future assimilation windows.

The paper is organized as follows. Section 7.2 reviews the EnKF algorithm, the under-sampling issue and typical solutions, and relevant machine learning models. Section 7.3 presents the new adaptive localization algorithms in detail, and discusses their computational complexity and implementation details. Section 7.4 discusses the setup of numerical experiments and the two test problems, the Lorenz and the quasi-Geostrophic (QG) models. Numerical results with the adaptive localization methodology are reported in Section 7.5. Conclusions and future directions are highlighted in Section 7.6.

7.2 Background

This section reviews the mathematical formulation of EnKF, and associated challenges such as under-sampling, filter divergence, and development of long-range spurious correlations. We discuss traditional covariance localization, a practical and successful ad-hoc solution to the problem of long-range spurious correlations, that requires an empirical tuning of the localization parameter, e.g., the radius of influence. The last subsection reviews the basic elements of a machine learning algorithm, with special attention being paid to ensemble based learning algorithms in machine learning.

7.2.1 Ensemble Kalman filters

EnKF proceeds in a prediction-correction fashion and carries out two main steps in every assimilation cycle: *forecast* and *analysis*. Assume an analysis ensemble $\{\mathbf{x}_{k-1}^a(e) \mid e = 1, \dots, N_{\text{ens}}\}$ is available at a time instance t_{k-1} . In the forecast step, an ensemble of forecasts $\{\mathbf{x}_k^f(e) \mid e = 1, \dots, N_{\text{ens}}\}$ is generated by running the numerical model forward to the next time instance t_k where observations are available:

$$\mathbf{x}_k^f(e) = \mathcal{M}_{t_{k-1} \rightarrow t_k}(\mathbf{x}_{k-1}^a(e)) + \eta_k(e), \quad e = 1, \dots, N_{\text{ens}}, \quad (7.1a)$$

where \mathcal{M} is a discretization of the model dynamics. To simulate the fact that the model is an imperfect representation of reality, random model error realizations $\eta_k(e)$ are added. Typical assumption is that the model error is a random variable distributed according to a Gaussian distribution $\mathcal{N}(0, \mathbf{Q}_k)$. In this paper we follow a perfect-model approach for simplicity, i.e., we set $\mathbf{Q}_k = \mathbf{0} \forall k$.

The generated forecast ensemble provides estimates of the ensemble mean $\bar{\mathbf{x}}_k^f$ and the flow-dependent background error covariance matrix \mathbf{B}_k at time instance t_k :

$$\bar{\mathbf{x}}_k^f = \frac{1}{N_{\text{ens}}} \sum_{e=1}^{N_{\text{ens}}} \mathbf{x}_k^f(e), \quad (7.1b)$$

$$\mathbf{X}'_k = [\mathbf{x}_k^f(e) - \bar{\mathbf{x}}_k^f]_{e=1, \dots, N_{\text{ens}}}, \quad (7.1c)$$

$$\mathbf{B}_k = \frac{1}{N_{\text{ens}} - 1} \mathbf{X}'_k \mathbf{X}'_k{}^T, \quad \mathbf{X}'_k = [\mathbf{x}_k^f(e) - \bar{\mathbf{x}}_k^f]_{e=1, \dots, N_{\text{ens}}}. \quad (7.1d)$$

In the analysis step each member of the forecast is analyzed separately using the Kalman

filter formulas [26, 27]:

$$\mathbf{x}_k^a(e) = \mathbf{x}_k^f(e) + \mathbf{K}_k \left([\mathbf{y}_k + \zeta_k(e)] - \mathcal{H}_k(\mathbf{x}_k^f(e)) \right), \quad (7.1e)$$

$$\mathbf{K}_k = \mathbf{B}_k \mathbf{H}_k^T (\mathbf{H}_k \mathbf{B}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1}, \quad (7.1f)$$

where $\mathbf{H}_k = \mathcal{H}'_k(\bar{\mathbf{x}}_k^f)$ is the linearized observation operator, e.g. the Jacobian, at time instance t_k . The stochastic (“perturbed”) version of the EnKF [26] adds different realizations of the observation noise $\zeta_k \in \mathcal{N}(0, \mathbf{R}_k)$ to each individual observation in the assimilation procedure. The same Kalman gain matrix \mathbf{K}_k is used to assimilate observation(s) to each member of the forecast ensemble.

Deterministic (“square root”) versions of EnKF [178] avoid adding random noise to observations, and thus avoid additional sampling errors. They also avoid the explicit construction of the full covariance matrices and work by updating only a matrix of state deviations from the mean. A detailed discussion of EnKF and variants can be found in [179, 180].

7.2.2 Inbreeding, filter divergence, and spurious correlations

EnKF is subject to sampling errors due to under-sampling whenever the number of ensembles is too small to be statistically representative of the large-dimensional model state. In practical settings, under-sampling leads to filter divergence, inbreeding, and the development of long-range spurious correlations [28].

Inbreeding and filter divergence In inbreeding the background error is under-estimated, which causes the filter to put more emphasis on the background state and less emphasis on the observations. This means that the forecast state is influenced adequately by the observational data, and the filter fails to adjust an incorrectly estimated forecast estate. Inbreeding and the filter divergence can be resolved using covariance inflation [168]; this is not further considered in this work. However, the machine learning approach proposed here for covariance localization can be extended to inflation.

Long-range spurious correlations The small number of ensemble members may result in a poor estimation of the true correlation between state components, or between state variables and observations. In particular, spurious correlations might develop between variables that are located at large physical distances, when the true correlation between

these variables is negligible. As a result, state variables are artificially affected by observations that are physically remote [167, 169]. This generally results in degradation of the quality of the analysis, and eventually leads to filter divergence.

7.2.3 Covariance localization

Covariance localization seeks to filter out the long range spurious correlations and enhance the estimate of forecast error covariance [169, 181, 182]. Standard covariance localization is typically carried out by applying a Schur (Hadamard) product [183, 184] between a correlation matrix ρ with distance-decreasing entries and the ensemble estimated covariance matrix, resulting in the localized Kalman gain:

$$\mathbf{K}_k = (\rho \circ \mathbf{B}_k) \mathbf{H}_k^T (\mathbf{H}_k (\rho \circ \mathbf{B}_k) \mathbf{H}_k^T + \mathbf{R}_k)^{-1}, \quad (7.2)$$

Localization can be applied to $\mathbf{H}_k \mathbf{B}_k$, and optionally to the \mathbf{B}_k projected into the observations space $\mathbf{H}_k \mathbf{B}_k \mathbf{H}_k^T$ [32]. Since the correlation matrix is a covariance matrix, the Schur product of the correlation function and the forecast background error covariance matrix is also a covariance matrix. Covariance localization has the virtue of increasing the rank of the flow-dependent background error covariance matrix $\rho \circ \mathbf{B}_k$, and therefore increasing the effective sample size.

A popular choice of the correlation function ρ is defined by the Gaspari-Cohn (GC) fifth order piecewise polynomial [170] function that is non-zero only for a small local region and zero every other places [32]:

$$\rho(z) = \begin{cases} -\frac{1}{4} (\|z\|/c)^5 + \frac{1}{2} (\|z\|/c)^4 + \frac{5}{8} (\|z\|/c)^3 - \frac{5}{3} (\|z\|/c)^2 + 1, & 0 \leq \|z\| \leq c, \\ \frac{1}{12} (\|z\|/c)^5 - \frac{1}{2} (\|z\|/c)^4 + \frac{5}{8} (\|z\|/c)^3 + \frac{5}{3} (\|z\|/c)^2 & c \leq \|z\| \leq 2c, \\ -5 (\|z\|/c) + 4 - \frac{2}{3} (c/\|z\|), & \\ 0, & 2c \leq \|z\| \end{cases} \quad (7.3)$$

The correlation length scale is $c = \sqrt{\frac{10}{3}} \ell$, [185] where ℓ is a characteristic physical distance. The correlation decreases smoothly from $\rho(0) = 1$ to zero at a distance more than twice the correlation length. Depending on the implementation, z can be either the distance between an observation and grid point or the distance between grid points in the physical space.

7.2.4 Machine learning

Machine learning (ML) has found numerous applications in data science, data mining, and data analytics. However, the immense potential of applying ML to help solve computational science problems remains largely untapped to date. Recently, data-driven approaches to predict and model the approximation errors of low-fidelity and surrogate models have shown promising results [149, 186, 187]. The multivariate predictions of local reduced-order-model method (MP-LROM) [149, 186] proposes a multivariate model to compute the error of local reduced-order surrogates. In [188] a new filter in DA framework is developed which is called Cluster Hybrid Monte Carlo (HMC) sampling filter (*Cℓ*HMC) for non-Gaussian data assimilation which relaxes the Gaussian assumption in the original HMC sampling filter [189], by employing a clustering step in the forecast-phase.

One of the most fundamental, and well-understood, family of ML algorithms is *regression analysis*. Generally speaking, a multivariate regression model [190] approximates the relationship between a set of dependent variables, and a set of independent variables. There is a plethora of ML algorithms for regression analysis, however we limit ourselves to *ensemble* ML algorithms.

Ensemble methods [191, 192] have proven successful in enhancing the performance and results of ML algorithms. The ensemble methods are based on the idea that *a group of weak learners can combine to form a strong learner*. Common types of ML ensembles include the Bootstrap aggregation – bagging for short – [193, 191], and Boosting [194]. In bagging, the training set is used to train an ensemble of ML models, and all trained models are equally important, i.e. the decisions made by all models are given the same weight. Each of the models is trained using a subset randomly drawn from the training dataset. A widely successful algorithm in this family of methods, is Random Forests (RF) [195]. In the boosting approach, on the other hand, the decisions made by the learners are weighted based on the performance of each model. A widely common algorithm in this approach is Gradient Boosting (GB) [196].

Random forests RFs [195] work by constructing an ensemble of decision trees, such that each tree builds a classification or regression model in the form of a tree structure. Instead of using the whole set of features available for the learning algorithm at once, each subtree uses a subset of features. The ensemble of trees is constructed using a variant of the bagging technique, thus yielding a small variance of the learning algorithm [191]. Furthermore, to ensure robustness of the ensemble-based learner, each subtree is assigned

a subset of features selected randomly in a way that minimizes the correlation between individual learners. Amongst the most special-purpose popular versions of RFs are Iterative Dichotomiser 3 (ID3) [197] and its successor (C4.5) [198], and conditional inference trees [199].

Consider a dataset D and a set of features F to be used by the RF. For each tree in the forest, a bootstrap sample $D^i \subset D$ is randomly selected. Instead of examining all possible feature-splits, a subset of the features $f \subset F$ with $|f| \ll |F|$ is randomly selected [200]. Each node then splits on the best feature in the subset f rather than F . This approach has the advantage that the RFs can be efficiently constructed in parallel, and that the correlation between trees in the ensemble is reduced. Random sampling and bootstrapping can be efficiently applied to RFs to generate a parallel, robust, and very fast learner for high-dimensional data and features.

Gradient boosting GB [196] proceeds, similar to other boosting methods, by incrementally building the prediction model as an ensemble of weak predictors. Specifically, GB algorithm build a sequence of simple regression (or classification) trees where each tree is constructed over the prediction residual of the preceding trees [201, 202]. This procedure gives a chance to each subtree to correct its predecessors, and consequently build an accurate ensemble-based model. GB is widely viewed as an optimization algorithm on a suitable cost function. An example is a squared error function, e.g. the sum of square of the differences between prediction and true data for each training sample. A gradient descent approach is followed, to minimize the cost function, by adding trees to the ensemble model. Increasing the number of trees leads to a more complex model, and tends in general to overfit the learning dataset [203, 204]. Traditional wisdom suggests adding subtrees to the model until no further significant improvement is achieved. However the number of nodes, and levels, in each subtree should be constrained to avoid overfitting.

7.3 Machine Learning Approach for Adaptive Localization

This section develops two machine learning approaches for adaptive covariance localization. In the first approach the localization radius changes in time, meaning that the same localization radius is used at all spatial points, but at each assimilation cycle the localization radius differs. In the second approach the localization radius changes both in time and space, and is different for each assimilation cycle and for each state variable. In both ap-

proaches the localization radius is adjusted depending on the model behavior and overall state of the system. Here we study what features of the solution affect the most the optimal value of localization radius, such that using that localization radius the difference between analysis and the true state gets minimized. RF or GB, or other suitable regression, model is used to construct the learning model that takes the impactful set of features as input and outputs the localization radius. We now describe in detail the features and the objective function of the proposed learning model.

7.3.1 Features and decision criteria

ML algorithms learn a function that maps input variables F , the features of the underlying problem, onto output target variables. The input to the learning model is a set of features F which describes the underlying problem. During the learning phase the algorithm finds the proper function using a known data set. This function is used to predict target outputs given new instances of input variables. In this work the target variables are the localization radii at each assimilation cycle. We consider atmospheric models that have numerous variables and parameters, and select the feature set F that capture the characteristics of the important behavioral patterns of the dynamical system at each assimilation cycle. Specifically, the idea is to focus on the set features that best reflect the changes in analysis state with respect to changes in the localization radii.

Selection of the feature set We now consider the selection of important features of the model results and data sets to be fed into the ML algorithms. Relying on the Gaussianity assumption of the prior distribution, natural features to consider are the first and second order moments of the prior distribution of the model state at each assimilation cycle. However, the large dimensionality of practical models can make it prohibitive to include the entire ensemble average vector (forecast state \mathbf{x}^f) as a feature for ML. One idea to reduce the size of the model state information is to select only model states with negligible correlations among them, e.g., states that are physically located at distances larger than the radius of influence. Another useful strategy to reduce model features is to select descriptive summaries such as the minimum and the maximum magnitude of state components in the ensemble.

The correlations between different variables of the model are descriptive of the behavior of the system at each assimilation cycle, and therefore are desirable features. Of course, it is impractical to include the entire state error covariance matrix among the features. Following the same reasoning as for state variables, we suggest including blocks of the correlation

matrix for variables located nearby in physical space, i.e., for subsets of variables that are highly correlated.

Decision criteria Under the Gaussianity assumption the quality of the DA solution is given by the quality of its first two statistical moments. Each of these aspects is discussed below.

A first important metric for the quality of ensemble-based DA algorithms is how well does the analysis ensemble mean (analysis state) represent the true state of the system. To quantify the accuracy of the ensemble mean we use the root mean-squared error (RMSE), defined as follows:

$$RMS E_k = \frac{1}{\sqrt{N_{\text{state}}}} \left\| \mathbf{x}_k - \mathbf{x}^{\text{true}}(t_k) \right\|_2, \quad (7.4)$$

where \mathbf{x}^{true} is the true system state, and $\|\cdot\|_2$ is the Euclidian norm. Since the true state is not known in practical applications we also consider the deviation of the state from collected measurements as a useful indication of filter performance. The observation-state $RMS E$ is defined as follows:

$$RMS E_k^{\text{x|y}} = \frac{1}{\sqrt{N_{\text{obs}}}} \left\| \mathcal{H}(\mathbf{x}_k) - \mathbf{y}_k \right\|_2. \quad (7.5)$$

Replacing \mathbf{x} in (8.12) and (7.5) with the forecast state \mathbf{x}^f or with the analysis state \mathbf{x}^a provides the formulas for the forecast or the analysis error magnitudes, respectively. The quality of the DA results measured by either (8.12) in case of perfect problem settings, or by (7.5) in case of real applications.

In this work we use the observation-analysis error metric (7.5), denoted by $RMS E^{\text{x|y}}$, as the first decision criterion.

A second important aspect that defines a good analysis ensemble is its spread around the true state. The spread can be visually inspected via the Talagrand diagram (rank histogram) [205, 206]. A quality analysis ensemble leads to a rank histogram that is close to a uniform distribution. Conversely, U-shaped and Bell-shaped rank histograms correspond to under-dispersion and over-dispersion of the ensemble, respectively. Ensemble based methods, especially with small ensemble sizes, are generally expected to yield U-shaped rank histograms, unless they are well-designed and well-tuned. The calculation of the rank statistics in model space requires the ordering the true state entries with respect to the generated ensemble members, which is not feasible in practical applications. A practical rank histogram can alternatively be constructed by ordering the entries of the observation vector with respect to the ensemble members entries projected into the observation space [205, 206].

In this work we use the uniformity of the analysis rank histogram, in observation space, as the second decision criterion.

We now discuss practical ways to quantify the level of uniformity of rank histograms. The level of uniformity of forecast rank histograms is used as a learning feature, and that of the analysis histogram as a decision criterion.

A reasonable approach is to quantify the level of uniformity by the similarity between a distribution fitted to the rank histogram and a uniform distribution. A practical measure of similarity between two probability distributions P and Q is the Kullback-Leibler (KL) divergence [207]:

$$D_{KL}(P||Q) = \mathbb{E}_P [\log(P) - \log(Q)] . \quad (7.6)$$

We first fit a beta distribution $Beta(\alpha, \beta)$ to the rank histogram (where the histogram domain $[0, N_{\text{ens}}]$ is mapped to $[0, 1]$ by a linear transformation). Considering that $Beta(1, 1)$ is a uniform distribution over the interval $[0, 1]$, we use the following measure of uniformity of the rank histogram:

$$D_{KL}(Beta(\alpha, \beta)||Beta(1, 1)) . \quad (7.7)$$

7.3.2 ML-based adaptive localization algorithm

We have identified two useful, but complementary, decision criteria, one that measures the quality of ensemble mean, and the second one that measures the quality of the ensemble spread. For the practical implementation we combine them into a single criterion, as follows:

$$C_r = w_1 RMS E^{\text{analy}} + w_2 D_{KL}(Beta(\alpha, \beta)||Beta(1.0, 1.0)) , \quad (7.8)$$

where the weighting parameters realize an appropriate scaling of the two metrics. The weights w_1, w_2 can be predefined, or can be learned from the data them as part of the ML procedure.

The best set of localization radii are those that that minimize the combined objective function (7.8).

Algorithm 6 summarizes the proposed adaptive localization methodology.

7.3.3 Adaptive localization in time

The first proposed learning algorithm uses the same (scalar) localization radius for all variables of the model. The value of this radius changes adaptively from one assimilation

cycle to the next. Specifically, at the current cycle we perform the assimilation using all localization radii from the pool and for each case compute the cost function C_r (7.8). After trying all possible radii from the pool, the radius associated with the minimum cost function is selected as winner. The analysis of the current assimilation cycle is the one computed using the winner radius.

At each assimilation cycle we collect a sample consisting of the features described in 7.3.1 as inputs, and the winner localization radius r as output (target variable) of the learning model. During the training phase, at each assimilation cycle, the ML algorithm learns the best localization radius corresponding to the system state and behavior. During the test phase, the learning model uses the current system information to estimate the proper value of the localization radius, without trying all possible values of localization radii. Algorithm 6 summarizes the adaptive localization procedure.

7.3.4 Adaptive localization in time and space

The second proposed learning algorithm is to adapt the localization radii in both time and space. A different localization radius is used for each of the state variables, and these radii change at each assimilation cycle. Here the localization radius is a vector \mathbf{r} containing a scalar localization parameter for each state variable of the system. The pool of radii in this methodology contains multiple possible vectors. The pool can be large since it can include permutations of all possible individual scalar radii values. Similar to previous learning algorithm, at each time point we perform the assimilation with one of the vectors from the pool of radii, and select the one corresponding to the minimum cost function as the winner.

At each assimilation cycle we collect a sample consisting of the model features as inputs and the winner vector of localization radii as output of the learning model. In the training phase, the model learns the relation between system state and localization radii and during the test phase it estimates the proper value of localization radius for each state individually. The number of target variables the learning model could be as large as the number of state variables. This situation can be improved by imposing that the same scalar radii are used for multiple components, e.g., for entire areas of the physical model.

7.3.5 Computational considerations

During the training phase, the learning phase of the proposed algorithm needs to try all possible radii from the pool, and re-do the assimilation with that localization radius. This is computationally demanding, but the model can be trained off-line using historical data. The testing phase the learning model predicts a good value of the localization radius, which is then used in the assimilation; no additional costs are incurred except for the (relatively inexpensive) prediction made by the trained model.

7.4 Setup of the Numerical Experiments

7.4.1 Forward operator

In order to study the performance of the proposed adaptive localization algorithm we employ two test models, namely the Lorenz-96 model [72], and the QG-1.5 model [208].

Lorenz-96 model The Lorenz-96 model is given by [72]:

$$\frac{dX_k}{dt} = -X_{k-2}X_{k-1} + X_{k-1}X_{k+1} - X_k + F, \quad k = 1, 2, \dots, K, \quad (7.9)$$

with $K = 40$ variables, periodic boundary conditions, and a forcing term $F = 8$. A vector of equidistant component values ranging from $[-2, 2]$ was integrated forward in time for 1000 steps, each of size 0.005 [units], and the final state was taken as the reference initial condition for the experiments. The background uncertainty is set to 8% of average magnitude of the reference solution. All state vector components are observed, i.e., $\mathcal{H} = \mathbf{I} \in \mathbb{R}^{K \times K}$ with \mathbf{I} the identity operator. To avoid filter collapse, the analysis ensemble is inflated at the end of each assimilation cycle, with the inflation factor set to $\delta = 1.09$.

Quasi-geostrophic (QG-1.5) model The QG-1.5 model described by Sakov and Oke [208] is a numerical approximation of the following equations:

$$\begin{aligned} q_t &= \psi_x - \varepsilon J(\psi, q) - A\Delta^3\psi + 2\pi \sin(2\pi y), \\ q &= \Delta\psi - F\psi, \\ J(\psi, q) &\equiv \psi_x q_x - \psi_y q_y, \\ \Delta &:= \partial^2/\partial x^2 + \partial^2/\partial y^2, \end{aligned} \quad (7.10)$$

where ψ is surface elevation or the stream function and q is the potential vorticity. We use the following values of the model coefficients (7.10) from [208],: $F = 1600$, $\varepsilon = 10^{-5}$, and $A = 2 \times 10^{-12}$. Boundary conditions used are $\psi = \Delta\psi = \Delta^2\psi = 0$. The domain of the model is a 1×1 [space units] square, with $0 \leq x \leq 1$, $0 \leq y \leq 1$, and is discretized by a grid of size 129×129 (including boundaries). A standard linear operator to observe 300 components of ψ is used [188]. The observation error variance is 4.0 [units squared] and synthetic observations are obtained by adding white noise to measurements of the sea height level (SSH) extracted from a model run with lower viscosity [188]. Here, the inflation factor is set to $\delta = 1.06$, and the localization function is GC (7.3) with the empirically-tuned optimal radius $\ell = 3$.

Assimilation filter All experiments are implemented in Python using the DATeS framework [209]. The performance of the proposed methodology is compared against the deterministic implementation of EnKF (DEnKF) with parameters empirically tuned as reported in [208].

Machine learning model Several ML regressors to model and predict the localization radii, for ensemble data assimilation algorithms, have been explored and tested. However, for brevity, we use R and GB as the main learning tools in the numerical experiments discussed below. We use *Scikit-learn*, the machine learning library in Python [210], to construct the ML models used in this work.

7.5 Numerical Results

7.5.1 Lorenz model with adaptive localization in time

The adaptive localization in time approach uses the same localization radius for all variables, and adapts the localization radius value at each assimilation cycle. This experiment has 100 assimilation cycles, where the first 80% are dedicated to the training phase and the last 20% to the testing phase. The pool of radii for this experiment covers all possible values for the Lorenz model: $r \in [1, 40]$.

We compare the performance of the adaptive localization algorithms against the best hand-tuned fixed localization radius value of 4 which is obtained through testing all possible localization radii ($[1, 40]$). Figure 7.1 shows the logarithm of RMSE between analysis

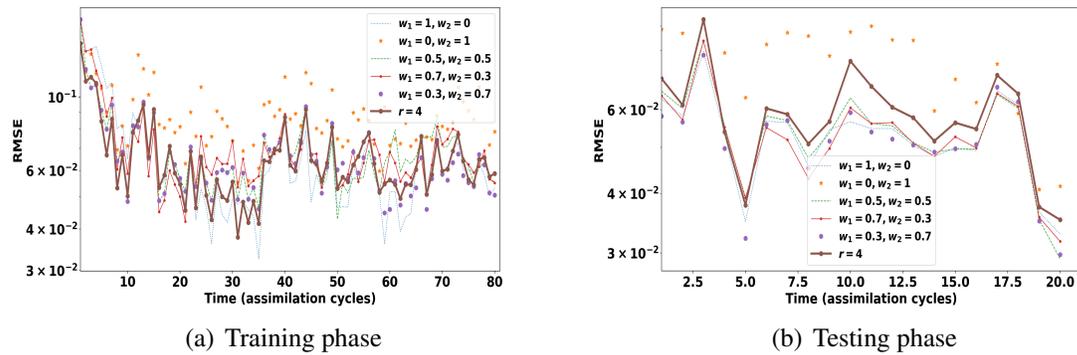


Figure 7.1: Data assimilation with the Lorenz-96 model (7.9). EnKF is applied with a Gaussian covariance localization function. EnKF results with adaptive covariance localization are shown for different choices of the weighting factors w_1, w_2 . The localization is adaptive in time, and is compared to results with fixed localization radius. The training phase consists of 80 assimilation cycles, followed by the testing phase with 20 assimilation cycles. The ML algorithm used here is RF. *The overall performance of the adaptive localization is better than that of the hand-tuned radius.*

ensemble and the true (reference) state. The performance of adaptive localization methodology is evaluated for different weights w_1, w_2 . The results indicate that increasing the weight of the KL distance measure increases the performance. *For the best choices of the weights the overall performance of the adaptive localization is slightly better than that of the fixed, hand-tuned radius.*

Figure 7.2 shows the variability in the tuned localization radius over time for both training and test phase. The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$ for this experiment. The adaptive algorithm changes the radius considerably over the simulation.

Using the RF methodology we were able to recognize and select the most important features affecting the target variable prediction. Figure 7.3 shows the 35 most important features of the Lorenz model which we included in our experiments.

7.5.2 QG model with adaptive localization in time

We use 100 assimilation cycles of the QG model, with 80% dedicated to the training phase, and 20% to the test phase. The pool of radii for this experiment is $r \in [1, 10]$. EnKF is

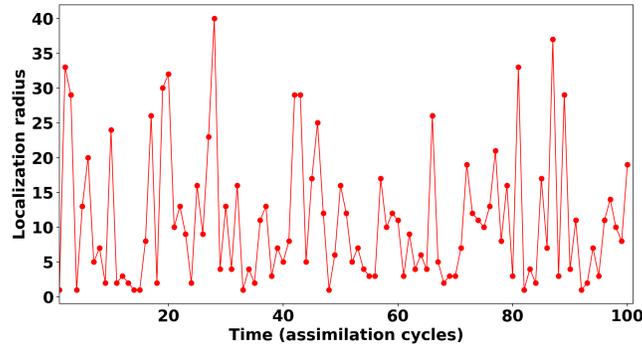


Figure 7.2: Data assimilation with the Lorenz-96 model (7.9). Shown is the evolution of the localization radius in time over all 100 assimilation cycles. The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$.

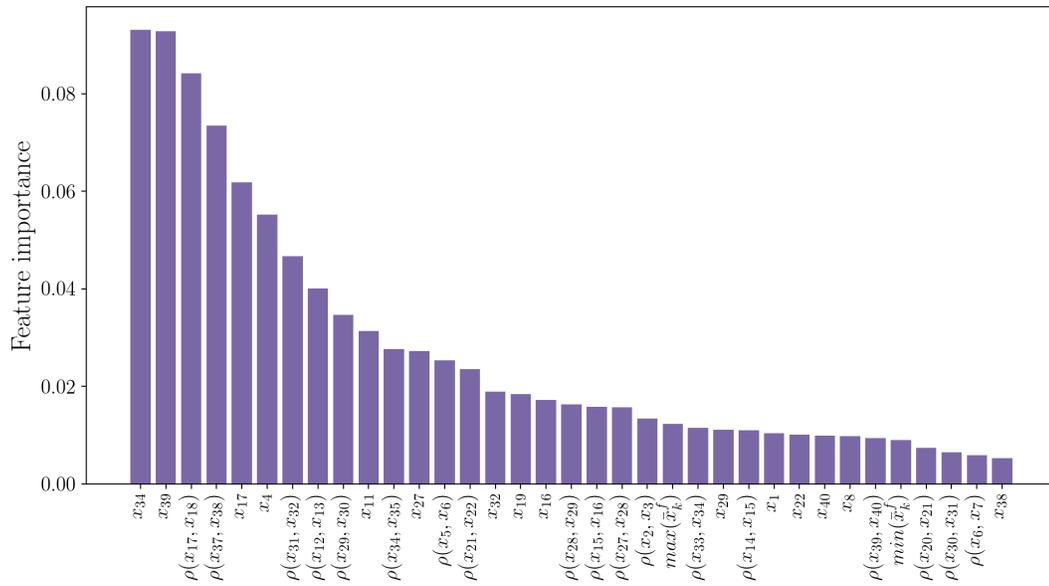


Figure 7.3: The 35 most important features of the Lorenz-96 model (7.9)

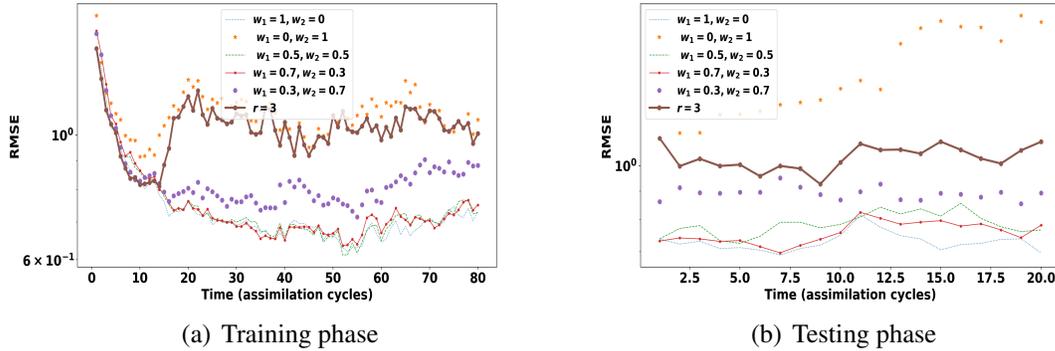


Figure 7.4: Data assimilation with the QG model (7.10). EnKF is applied with a GC covariance localization function. The (\log) RMSE is shown on the vertical axis, and the time (assimilation cycles) is shown on the horizontal axis. EnKF results with adaptive covariance localization are shown for different choices of the weighting factors w_1 , w_2 . The localization is adaptive in time, and is compared to results with fixed localization radius. The training phase consists of 80 assimilation cycles, followed by the testing phase with 20 assimilation cycles. The ML algorithm used here is RF. *The analysis results with adaptive localization outperform those obtained with the hand-tuned radius.*

used with 25 ensemble members, inflation factor $\delta = 1.06$, and GC localization function. An empirically optimal localization radius with these configurations was found by hand-tuning to be $r = 3$. We use it as a comparison benchmark for the performance of the adaptive localization.

Figure 7.4 shows the logarithm of RMSE between analysis obtained at each assimilation cycle and the true analysis. The performance of adaptive localization with different weights w_1 , w_2 is evaluated against the fixed localization with radius $r = 3$. With higher weights for the KL distance measure, the performance of adaptive localization also increases. *The analysis results with adaptive localization outperform those obtained with the hand-tuned radius.*

Figure 7.5 shows the variability of the localization radius in time for the weights $w_1 = 0.7$ and $w_2 = 0.3$. The adaptive algorithm changes the radius considerably over the course of the simulation.

Figure 7.6 shows the 35 most important features of the QG model which we included in our experiments.

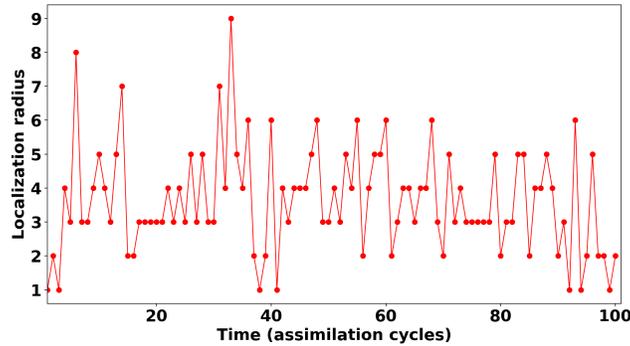


Figure 7.5: Data assimilation with the QG model (7.10). Shown is the evolution of the localization radius in time over all 100 assimilation cycles. The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$

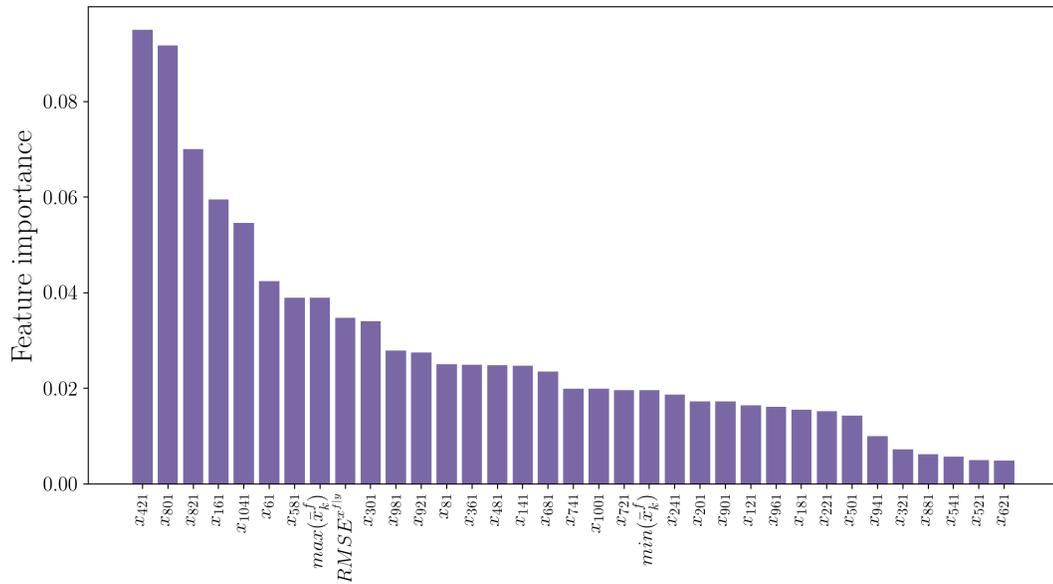


Figure 7.6: The 35 most important features of the QG model (7.10)

7.5.3 Lorenz model with adaptive localization in time and space

Here the localization radius changes at each assimilation cycle and it is also changing for each individual state variable of the system. The pool of radii for this experiment consists of random vectors of size 40 where each component of the vector can have value in the range of all possible radii for the Lorenz model i.e [1, 40]. Each component of the vectors in the pool can have different permutations of values in the range of [1, 40]. The total number of all possible vectors is huge, and testing all in the training phase is infeasible. One way to limit the number of trials is to test randomly selected vectors of radii in the pool. For this experiments we set the number of trials to 30 and at each trial we randomly pick a vector of radii from the pool. The localization radius of each state variable is the corresponding component in the vector, the cost function of using each of the trials is obtained at each assimilation cycle. The number of target variables to estimate at each assimilation cycle in the test phase is 40 and hence we need more samples for the training phase. The number of assimilation cycles for this experiment is 1000, from which 80% dedicated to the training phase, and 20% to the testing phase. The EnKF uses 25 ensemble members, the inflation factor of 1.09 and the localization function is Gaussian.

Figure 7.7 shows the logarithm of RMSE between analysis and the reference state. The performance of adaptive localization with different weights w_1 , w_2 is evaluated against the fixed localization radius $r = 4$. In the testing phase the results with the adaptive radii are slightly better than those with the optimal fixed radius.

Figure 7.8 shows the statistical variability in localization radii for the Lorenz model over time with the weights $w_1 = 0.7$ and $w_2 = 0.3$. Figure 7.8(a) shows the average of localization radius variability in time for each state variable of the Lorenz model and Figure 7.8(b) shows the standard deviation of localization radius change in time for each state variable. The average and standard deviations are taken over the state variables; we see that the adaptive values chosen by the algorithm can vary considerably. This variability can be further seen in Figure 7.9, which shows the evolution of localization radii in both time and space for the Lorenz model. The first 100 cycles of training phase and the last 100 cycles of the testing phase are selected. The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$.

7.5.4 QG model with adaptive localization in time and space

The pool of localization radii for this experiment consists of random vectors of size 1085, where each component of the vector can have values in the range of proper radii for the

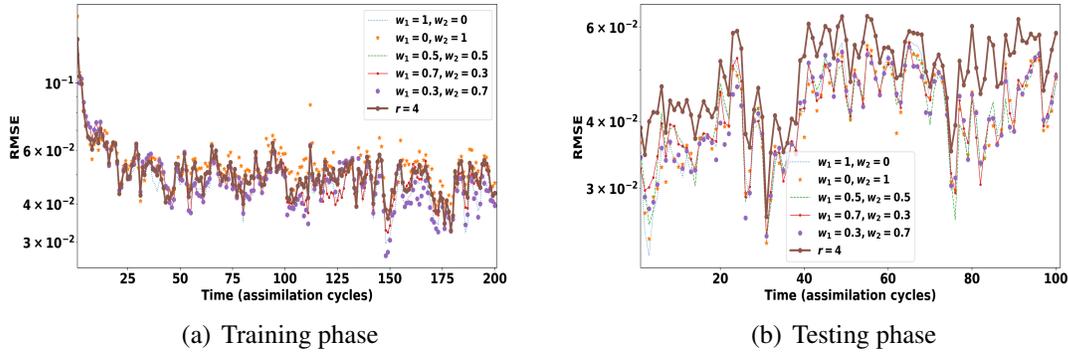


Figure 7.7: Data assimilation with the Lorenz-96 model (7.9). EnKF is applied with a Gaussian covariance localization function. EnKF results with adaptive covariance localization are shown for different choices of the weighting factors w_1, w_2 . The localization is adaptive in time and space, and is compared to results with fixed localization radius. The training phase consists of 800 assimilation cycles, followed by the testing phase with 200 assimilation cycles. The ML algorithm used here is RF.

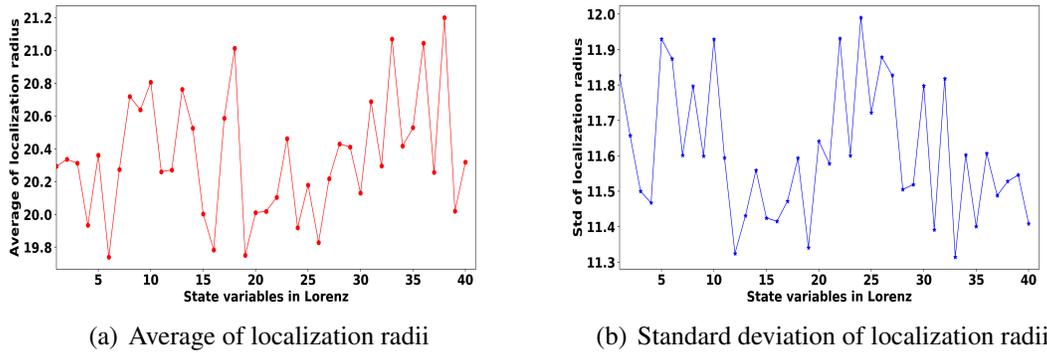


Figure 7.8: Statistical variability in localization radii for different state variables of the Lorenz-96 model (7.9). The total 1000 assimilation cycles include both training and testing phases. The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$.

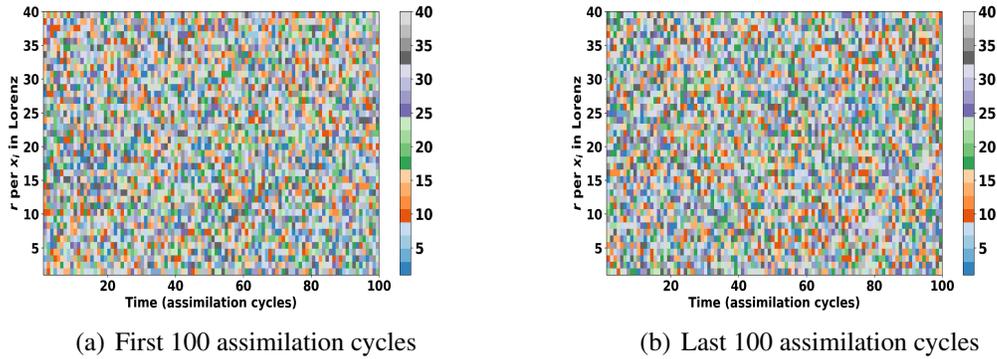


Figure 7.9: Evolution of localization radii in both time and space for the Lorenz-96 model (7.9). The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$.

QG model. One practical restriction is that the localization radius used for neighboring grid points should not be too different. We noticed that having too much variability in the choice of localization radii for grid points located nearby in space may lead to physical imbalances and filter divergence. One remedy is to narrow down the range of possible radii to a limited range. Here for example, we restricted the localization radius possible values to $[2, 3]$, $[3, 4]$, or $[4, 5]$. EnKF uses 25 ensemble members, an inflation factor of 1.06, and the GC localization function.

Figure 7.10 shows the RMSE of the analysis error at each assimilation cycle. The time and space adaptive radii results are not as good as those obtained with the fixed, hand-tuned radius. This is likely due to the very limited range of radii that the algorithm was allowed to test in each experiment.

Figure 7.8 shows the statistical variability in localization radii for the QG model, with the adaptive criterion weights $w_1 = 0.7$ and $w_2 = 0.3$. The variability is computed across all state vector components. The limited range of values from which the radius selection is made leads to a small variability of the radii. The changes made by the adaptive algorithm are shown in Figure 7.12 for the first 10 cycles of training phase and for the last 10 cycles of test phase. The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$. We notice that the radii chosen for different state variables seem to be uncorrelated in space or time.

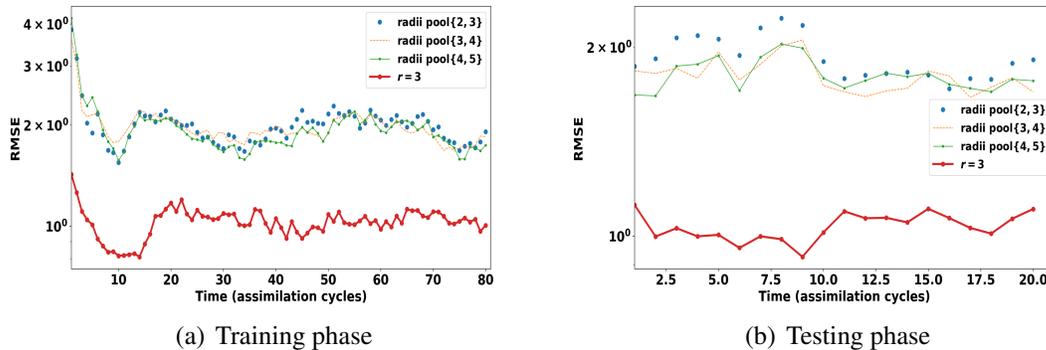


Figure 7.10: Data assimilation with the QG model (7.10). EnKF is applied with a GC covariance localization function. The (\log) RMSE is shown on the vertical axis, and the time (assimilation cycles) is shown on the horizontal axis. EnKF results with adaptive covariance localization are shown for different choices of the weighting factors w_1 , w_2 . The localization is adaptive in time, and is compared to results with a fixed localization radius. The training phase consists of 80 assimilation cycles, and testing phase follows with 20 assimilation cycles. The time and space adaptive radii results are not as good as those obtained with the fixed, hand-tuned radius. The ML algorithm used here is RF.

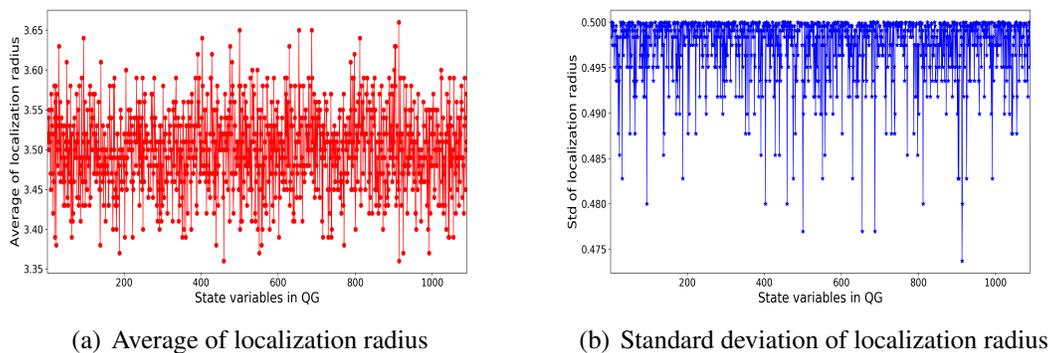


Figure 7.11: Statistical variability in the localization radius for each state variable of QG model (7.10). The total assimilation cycles are 100 including both training and testing phases. The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$.

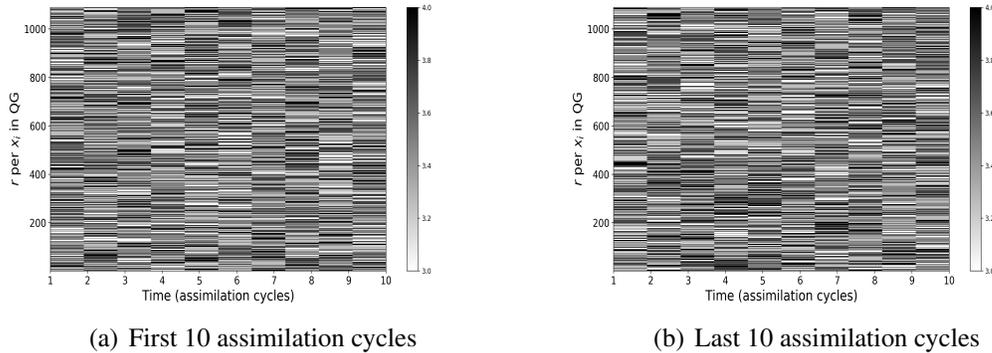


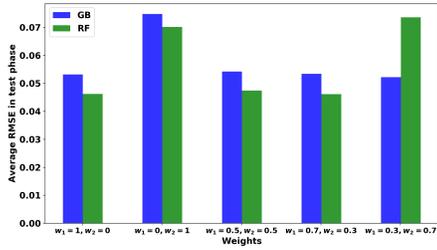
Figure 7.12: Variability of localization radius in both time and space for QG model (7.10). The pool of radii is $\{3, 4\}$. The weights of the adaptive localization criterion are $w_1 = 0.7$ and $w_2 = 0.3$.

7.5.5 On the choice of the learning model

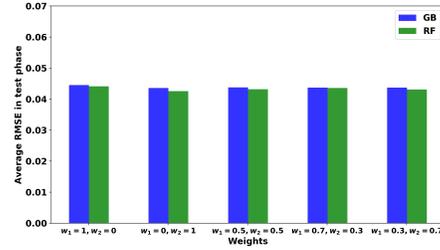
The work in this paper is not aimed to cover or compare all suitable ML algorithms in the context of adaptive covariance localization. In the numerical experiments presented above, we chose the RF as the main learning model, however the method proposed is not limited this choice, and can be easily extended to incorporate other suitable regression model. For example RF could be replaced with GB, however the computational cost of training the regressor, and the performance of the DA algorithm must be both accounted for.

DA performance To compare the performance of the DA filter with localization radii predicted by RF against GB, we study the RMSE obtained by incorporating each of these two learning models. Figure 7.13 shows the average RMSE over the test phase resulting by replacing RF with GB. Here, the RMSE results for both cases, i.e. time-only and space-time adaptivity, resulting by incorporating RF tend to be slightly lower that resulting when GB is used.

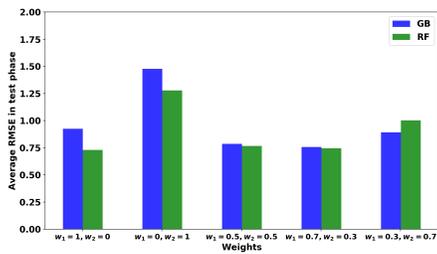
Computational time Table 7.1 shows the CPU-time spent in fitting the training dataset or training the learning model with both RF and GB. Here we show the running times based on the numerical experiments carried out using Lorenz-96 model. Learning RF model is less time consuming than QG, especially in the case of space-time adaptivity. This is mainly because RF, by construction, supports multi-target regression, while GB



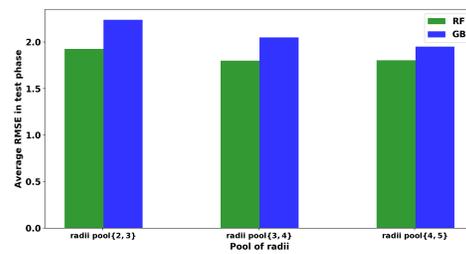
(a) Lorenz-96; time adaptivity



(b) Lorenz-96; space-time adaptivity



(c) QG; time adaptivity



(d) QG; space-time adaptivity

Figure 7.13: Data assimilation with the Lorenz-96 model (7.9), and QG model (7.10). Average RMSE resulting from EnKF over test phase, of the learning algorithm, is obtained by using both RF and GB. Results with adaptive covariance localization are shown for different choices of the weighting factors w_1 , w_2 . The dynamical model and adaptivity type, e.g. time vs space-time, are shown under each panel. The space-time adaptivity in QG model uses limited pool of radii with fixed weight combination as discussed in Figure 7.10.

Table 7.1: CPU-time of the training time of the two ML algorithms, RF and GB for both time adaptivity and space-time adaptivity approaches. Data assimilation experiment carried out using Lorenz model (7.9).

CPU time (seconds)		Adaptivity type	
		time	space-time
ML model	GB	0.0467	16.3485
	RF	0.0308	0.7508

does not. A simple extension of GB is used for space-time adaptivity, by fitting a regressor to each of the outputs.

From both Figure 7.13, and Table 7.1, we can empirically conclude that RF yields a combination of better performance and lower computational time, than GB.

7.5.6 Discussion of numerical results

Several points can be concluded from the experimental results. Firstly, one needs to consider different decision criterion weights for different problems. Here the best weights are not the same for both models. For the Lorenz model, a combination with a larger weight for KL distance has a positive affect. A more balanced set of weights works better for the QG model. Secondly, adaptivity leads to a considerable variability of the localization radii in both time and space. As the feature importance plots show, the values of state variables have a significant bearing on radius predictions. Moreover, the importance of all state variables is not the same, and some variables in the model have a higher impact on the prediction of localization radii. Finally, the training of the localization algorithms in both time and space with the current methodology is computationally expensive. Future research will focus on making the methodology truly practical for very large models.

7.6 Concluding Remarks and Future Work

This study proposes an adaptive covariance localization approach for the EnKF family of data assimilation methods. Two methodologies are presented and discussed, namely adaptivity in time and adaptivity in space and time. The adaptive localization approach is based on RF and GB machine learning regression techniques. The learning model can be trained off-line using historical records, e.g., reanalysis data. Once it is successfully trained, the regression model is used to estimate the values of localization radii in future assimilation cycles. Numerical results carried out using two standard models suggest that the proposed automatic approach performs at least as good as the traditional EnKF with empirically hand-tuned localization parameters.

In order to extend the use of machine learning techniques to support data assimilation, an important question that will be addressed in future research concerns the optimal choice of features in large-scale numerical models. Specifically, one has to select sufficient aspects of the model state to carry the information needed to train a machine learning algorithm. In the same time, the size of the features vector needs to be relatively small, even when the

model state is extremely large. Next, the computational expense of the training phase is due to the fact that the analysis needs to be repeated with multiple localization radii. Future work will seek to considerably reduce the computational effort by intelligently narrowing the pool of possible radii to test, and by devising assimilation algorithms that reuse the bulk of the calculations when computing multiple analyses with multiple localization radii.

Algorithm 6 Adaptive localization algorithm

```

1: Input:  $Radii\_Pool = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots, \mathbf{r}_n]$ 
2:  $dataset = []$ 
3: For  $k \in$  assimilation times
4:   If  $k \in$  training phase
5:      $Cost = \infty$ 
6:     Obtain the feature set  $\mathbf{F}$ , as described in 7.3.1
7:     For  $i = 1$  to  $n$ 
8:        $r = Radii\_Pool[i]$ 
9:       Obtain  $\mathbf{x}_k^f, \mathbf{x}_k^a$ 
10:      Evaluate  $C_r$  (7.8)
11:      If ( $C_r < Cost$ ) Then
12:         $Winner\_Radius = r$ 
13:         $Cost = C_r$ 
14:      End If
15:       $dataset[k, :] = [\mathbf{F}, Winner\_Radius]$ 
16:    End For loop
17:    Use the analysis ensemble obtained using  $Winner\_Radius$  in the next assimilation
    cycle
18:  End If
19:  Train the learning model with  $dataset$ 
20:  If  $k \in$  test phase
21:     $\mathbf{r} =$  Learning model predicts the localization radius
22:    Perform the assimilation, i.e. calculate that analysis, using a covariance matrix
    localized with  $\mathbf{r}$ 
23:  End If
24: End For loop

```

Chapter 8

A Learning Based Approach for Uncertainty Analysis in Numerical Weather Prediction Models

8.1 Introduction

Computer simulation models of the physical world, such as numerical weather prediction (NWP) models, are imperfect and can only approximate the complex evolution of physical reality. Some of the errors are due to the uncertainty in the initial and boundary conditions, forcings, and model parameter values. Other errors, called structural model errors, are due to our incomplete knowledge about the true physical processes, and manifest themselves as missing dynamics in the model [211]. Examples of structural errors include the misrepresentation of sea-ice in the spring and fall, errors affecting the stratosphere above polar regions in winter [212], as well as errors due to the interactions among (approximately-represented) physical processes.

Data assimilation improves model forecasts by fusing information from both model outputs and observations of the physical world in a coherent statistical estimation framework [213, 214, 215, 212]. While traditional data assimilation reduces the uncertainty in the model state and model parameter values, no methodologies to reduce the structural model uncertainty are available to date.

In this study we consider the Weather Research and Forecasting (WRF) model [216], a mesoscale atmospheric modeling system. The WRF model includes multiple physical

processes and parametrization schemes, and choosing different model options can lead to significant variability in the model predictions [217, 218].

Among different atmospheric phenomena, the prediction of precipitation is extremely challenging and is obtained by solving the atmospheric dynamic and thermodynamic equations [218]. Model forecasts of precipitation are very sensitive to physics options such as the micro-physics, cumulus, long wave, and short wave radiation [219, 218, 220]. Other physics settings that can affect the WRF precipitation predictions include surface physics, planetary boundary layer (PBL), land-surface (LS) parameterizations, and lateral boundary condition. Selecting the right physical process representations and parameterizations is a challenge. In practice the values of physical parameters are empirically determined such as to minimize the difference between the measurements and model predictions [216, 220].

Considerable effort has been dedicated to determining the best physical configurations of the weather forecast models such as to improve their predictions of precipitation. No single choice of physical parameters works perfectly for all times, geographical locations, or meteorological conditions [221, 222]. Lowrey and Yang [220] investigated the errors in precipitation predictions caused by different parameters including micro-physics and cumulus physics, the buffer zone, the initialization interval, the domain size and the initial and boundary conditions. Jankov et al. [223] examined different combinations of cumulus convection schemes, micro-physical options, and boundary conditions. They concluded that no configuration was the clear winner at all times, and the variability of precipitation predictions was more sensitive to the choice of the cumulus options rather than micro-physical schemes. Another study conducted by Nasrollahi [218] showed that the best model ability to predict hurricanes was achieved using a particular cumulus parameterization scheme combined with a particular micro-physics scheme. Therefore, the interactions of different physical parameterizations have a considerable impact on model errors, and can be considered as one of the main sources of uncertainty that affect the forecast accuracy.

This paper demonstrates the potential of machine learning techniques to help solve two important problems related to the structural/physical uncertainty in numerical weather prediction models. The first problem addressed herein is the estimation of systematic model errors in output quantities of interest at future times, and the use of this information to improve the model forecasts. The second problem considered is the identification of those specific physical processes that contribute most to the forecast uncertainty in the quantity of interest under specified meteorological conditions.

The application of machine learning techniques to problems in environmental science has grown considerably in recent years. In [224] a kernel based regression method is devel-

oped as a forecasting approach with performance close to Ensemble Kalman Filter (EnKF) and less computational resources. Krasnopol et al. [225] employ an Artificial Neural Network technique for developing an ensemble stochastic convection parameterization for climate models. Attia et al. [188] develop a new filtering algorithm called Cluster Hybrid Monte Carlo sampling filter (CLHMC) non-Gaussian data assimilation which relaxes the Gaussian assumptions by employing a clustering step. Moosavi et al. [226] use regression machine learning techniques for adaptive localization in ensemble based data assimilation.

This study focuses on the uncertainty in forecasts of cumulative precipitation caused by imperfect representations of physics and their interaction in the WRF model. The total accumulated precipitation includes all phases of convective and non-convective precipitation. Specifically, we seek to use the discrepancies between WRF forecasts and measured precipitation levels in the past in order to estimate in advance the WRF prediction uncertainty. The model-observation differences contain valuable information about the error dynamics and the missing physics of the model. We use this information to construct two probabilistic functions. The first one maps the discrepancy data and the physical parameters onto the expected forecast errors. The second maps the forecast error levels onto the set of physical parameters that are consistent with them. Both maps are constructed using supervised machine learning techniques, specifically, using Artificial Neural Networks and Random Forests [81]. The two probabilistic maps are used to address the problems posed above, namely the estimation of model errors in output quantities of interest at future times, and the identification of physical processes that contribute most to the forecast uncertainty.

The remainder of this study is organized as follows. Section 8.2 covers the definition of the model errors. Section 8.3 describes the proposed approach of error modeling using machine learning. Section 8.4 reports numerical experiments with the WRF model that illustrate the capability of the new approach to answer two important questions regarding model errors. Conclusions are drawn in Section 8.5.

8.2 Model errors

First-principles computer models capture our knowledge about the physical laws that govern the evolution of a real physical system. The model evolves an initial state at the initial time to states at future times. All models are imperfect, e.g., atmospheric model uncertainties are associated with sub-grid modeling, boundary conditions, and forcings. All these modeling uncertainties are aggregated into a component that is generically called *model*

error [227, 228, 229]. In the past decade there has been a considerable scientific effort to incorporate model errors and estimate their impact on the best estimate in both variational and statistical approaches [213, 230, 231, 232, 233, 212, 234].

In what follows, we describe our mathematical formulation of the model error associated with NWP models. A similar formulation has been used in [211] where the model structural uncertainty is studied based on the information provided by the discrepancy between the model solution and the true state of the physical system, as measured by the available observations.

Consider the following NWP computer model \mathcal{M} , that describes the time-evolution of the state of the atmosphere:

$$\mathbf{x}_t = \mathcal{M}(\mathbf{x}_{t-1}, \Theta), \quad t = 1, \dots, T. \quad (8.1a)$$

The state vector $\mathbf{x}_t \in \mathbb{R}^n$ contains the dynamic variables of the atmosphere such as temperature, pressure, precipitation, tracer concentrations, at all spatial locations covered by the model, and at t . All the physical parameters of the model are lumped into $\Theta \in \mathbb{R}^\ell$.

Formally, the true state of the atmosphere can be described by a physical process \mathcal{P} with internal states v_t , which are unknown. The atmosphere, as an abstract physical process, evolves in time as follows:

$$v_t = \mathcal{P}(v_{t-1}), \quad t = 1, \dots, T. \quad (8.1b)$$

The model state seeks to approximate the physical state:

$$\mathbf{x}_t \approx \psi(v_t), \quad t = 1, \dots, T, \quad (8.1c)$$

where the operator ψ maps the physical space onto the model space, e.g., by sampling the continuous meteorological fields onto a finite dimensional computational grid [211].

Assume that the model state at $t - 1$ has the ideal value obtained from the true state via (8.1c). The model prediction at t will differ from the reality:

$$\psi(v_t) = \mathcal{M}(\psi(v_{t-1}), \Theta) + \delta_t(v_t), \quad t = 1, \dots, T, \quad (8.2)$$

where the discrepancy $\delta_t \in \mathbb{R}^n$ between the model prediction and reality is the *structural model error*. This vector lives in the model space.

Although the global physical state v_t is unknown, we obtain information about it by measuring of a finite number of observables $\mathbf{y}_t \in \mathbb{R}^m$, as follows:

$$\mathbf{y}_t = h(v_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \mathbf{R}_t), \quad t = 1, \dots, T, \quad (8.3)$$

Here h is the observation operator that maps the true state of atmosphere to the observation space, and the observation error ϵ_t is assumed to be normally distributed.

In order to relate the model state to observations we also consider the observation operator \mathcal{H} that maps the model state onto the observation space; the model-predicted values $\mathbf{o}_t \in \mathbb{R}^m$ of the observations (8.3) are:

$$\mathbf{o}_t = \mathcal{H}(\mathbf{x}_t), \quad t = 1, \dots, T. \quad (8.4)$$

We note that the measurements \mathbf{y}_t and the predictions \mathbf{o}_t live in the same space and therefore can be directly compared. The difference between the observations (8.6b) of the real system and the model predicted values of these observables (8.4) represent the model error in observation space:

$$\Delta_t = \mathbf{o}_t - \mathbf{y}_t \in \mathbb{R}^m, \quad t = 1, \dots, T. \quad (8.5)$$

For clarity, in what follows we make the following simplifying assumptions [211]:

- the physical system is finite dimensional $v_t \in \mathbb{R}^n$,
- the model state lives in the same space as reality, i.e., $\mathbf{x}_t \approx v_t$ and $\psi(\cdot) \equiv id$ is the identity operator in (8.1c), and
- $\mathcal{H}(\cdot) \equiv h(\cdot)$ in (8.3) and (8.4).

These assumptions imply that the discretization errors are very small, and that the main source of error are the parameterized physical processes represented by Θ and the interaction among these processes. Uncertainties from other sources, such as boundary conditions, are assumed to be negligible.

With these assumptions, the evolution equations for the physical system (8.1b) and the physical observations equation (8.3) become, respectively:

$$v_t = \mathcal{M}(v_{t-1}, \Theta) + \delta_t(v_t), \quad t = 1, \dots, T, \quad (8.6a)$$

$$\mathbf{y}_t = h(v_t) + \epsilon_t. \quad (8.6b)$$

The model errors δ_t (8.2) are not fully known at any time t , as having the exact errors is akin to having a perfect model. However, the discrepancies between the modeled and measured observable quantities (8.5) at past times have been computed and are available at the current time t .

Our goal is to use the errors in observable quantities at past times, Δ_τ for $\tau = t - 1, t - 2, \dots$, in order to estimate the model error δ_τ at future times $\tau = t, t + 1, \dots$. This is achieved by unravelling the hidden information in the past Δ_τ values. Good estimates of the discrepancy δ_t , when available, could improve model predictions by applying the correction (8.6a) to model results:

$$\mathbf{v}_t \approx \mathbf{x}_t + \delta_t. \quad (8.7)$$

Our proposed error modeling approach constructs input-output mappings to estimate given aspects of model errors δ_t . The inputs to these mappings are the physical parameters Θ of the model. The outputs to these mappings are different aspects of the error in a quantity of interest, such as the model errors over a specific geographical location, or the error norm of model error integrated over the entire domain.

Specifically, the aspect of interest (quantity of interest) in this study is the error in precipitation levels forecasted by the model. The parameters Θ describe the set of physical processes that are essential to be included in the WRF model in order to produce accurate precipitation forecasts. The WRF model is modular and different combinations of the physical packages can be selected, each corresponding to a different value of Θ .

We use the error mappings learned from past model runs to estimate the model errors δ_t . We also consider estimating what combination of physical processes Θ leads to lower model errors, or reversely, what interactions of which physics cause larger errors in the prediction of the quantity of interest.

8.3 Approximating model errors using machine learning

We propose a multivariate input-output learning model to predict the model errors δ , defined in (8.2), stemming from the uncertainty in parameters Θ . To this end, we define a probabilistic function ϕ that maps every set of input features $F \in \mathbb{R}^r$ to output target variables $\Lambda \in \mathbb{R}^o$:

$$\phi(F) \approx \Lambda, \quad (8.8)$$

and approximate the function ϕ using machine learning.

Different particular definitions of ϕ in (8.8) will be used to address two different problems related to model errors, as follows:

1. The first problem is to estimate the systematic model error in certain quantities of interest at future times, and to use this information in order to improve the WRF

forecast. To achieve this one quantifies the model error aspects that correspond to running WRF with different physical configurations (different parameters Θ).

2. The second problem is to identify the specific physical processes that contribute most to the forecast uncertainty in the quantity of interest under specified meteorological conditions. To achieve this one finds the model configurations (physical parameters Θ) that lead to forecast errors smaller than a given threshold under specified meteorological conditions.

In what follows we explain in detail the function ϕ specification, the input features, and the target variables for each of these problems.

8.3.1 Problem one: estimating in advance aspects of interest of the model error

Forecasts produced by NWP models are contaminated by model errors. These model errors are highly correlated in time; hence historical information about the model errors can be used as an input to the learning model to gain insight about model errors that affect the forecast. We are interested in the uncertainty caused due to the interaction between the various components in the physics based model; these interactions are lumped into the parameter Θ that is supplied as an input to the learning model. The learning model aims to predict the error of NWP model of next forecast window using the historical values of model error and the physical parameters used in the model. We define the following mapping:

$$\phi^{\text{error}}(\Theta, \Delta_{\tau}, \mathbf{o}_{\tau}, \mathbf{o}_t) \approx \Delta_t \quad \tau < t. \quad (8.9)$$

We use a machine learning algorithm to approximate the function ϕ^{error} . The learning model is trained using a dataset that consists of the following inputs:

- WRF physical packages that affect the physical quantity of interest (Θ),
- historical WRF forecasts (\mathbf{o}_{τ} for $\tau \leq t - 1$),
- historical model discrepancies (Δ_{τ} for $\tau \leq t - 1$),
- WRF forecast at the current time (\mathbf{o}_t),
- the available model discrepancy at the current time (Δ_t) since we have access to the observations from reality y_t at the current time step.

In supervised learning process, the learning model identifies the effect of physical packages, the historical WRF forecast, the historical model discrepancy, and the WRF forecast at the current time on the available model discrepancy at the current time. After the model get trained on the historical data, it yields an approximation to the mapping ϕ^{error} . We denote this approximate mapping by $\hat{\phi}^{\text{error}}$.

During the test phase the approximate mapping $\hat{\phi}^{\text{error}}$ is used to estimate the model discrepancy $\hat{\Delta}_{t+1}$ in advance. We emphasize that the model prediction (WRF forecast) at the time of interest $t + 1$ (\mathbf{o}_{t+1}) is available, where as the model discrepancy $\hat{\Delta}_{t+1}$ is an unknown quantity. In fact the run time of WRF is much smaller than the time interval between t and $t + 1$, or in other way, the time interval is large enough to run the WRF model and obtain the forecast for next time window, estimate the model errors for next time window and finally improve the model forecast by combining the model forecast and model errors.

At the test time we predict the future model error as follows:

$$\hat{\Delta}_{t+1} \approx \hat{\phi}^{\text{error}}(\Theta, \Delta_{\tau}, \mathbf{o}_{\tau}, \mathbf{o}_{t+1}), \quad \tau < t + 1.$$

As explained in [211], the predicted error $\hat{\Delta}_{t+1}$ in the observation space can be used to estimate the error δ_{t+1} in the model space. In order to achieve this one needs to use additional information about the structure of the model and the observation operator. For example, if the error $\hat{\Delta}_{t+1}$ represents the projection of the full model error onto the observation space, we have:

$$\Delta_{t+1} \approx \mathbf{H}_t \cdot \delta_{t+1}, \quad \hat{\delta}_{t+1} \approx \mathbf{H}_t (\mathbf{H}_t^T \mathbf{H}_t)^{-1} \mathbf{H}_t^T \cdot \hat{\Delta}_{t+1}, \quad (8.10a)$$

where we use the linearized observation operator at the current time, $\mathbf{H}_t = h'(\mathbf{x}_t)$. A more complex approach is to use a Kalman update formula:

$$\hat{\delta}_{t+1} \approx \text{cov}(\mathbf{x}_t, \mathbf{o}_t) (\text{cov}(\mathbf{o}_t, \mathbf{o}_t) + \mathbf{R}_t)^{-1} \hat{\Delta}_{t+1}, \quad (8.10b)$$

where \mathbf{R}_t is the covariance of observation errors. The Kalman update approach requires estimates of the covariance matrices between model variables; such covariances are already available in an ensemble based data assimilation system. Once we estimate the future model error δ_{t+1} , we can improve the NWP output using equation (8.7).

8.3.2 Problem two: identifying the physical packages that contribute most to the forecast uncertainty

Typical NWP models incorporate an array of different physical packages to represent multiple physical phenomena that act simultaneously. Each physical package contains several

alternative configurations (e.g., parameterizations or numerical solvers) that affect the accuracy of the forecasts produced by the NWP model. A particular scheme in a certain physical package best captures the reality under some specific conditions (e.g., time of the year, representation of sea-ice, etc.). The primary focus of this study is the accuracy of precipitation forecasts, therefore we seek to learn the impacts of all the physical packages that affect precipitation. To this end, we define the following mapping:

$$\phi^{\text{physics}}(\Delta_t) \approx \Theta, \quad (8.11)$$

that estimates the configuration Θ of the physical packages such that the WRF run generates a forecast with an error consistent with the prescribed level Δ_t (where Δ_t defined in equation (8.5) is the forecast error in observation space at time t .)

We train the model to learn the effect of the physical schemes on the mismatch between WRF forecasts and reality. The input data required for the training process is obtained by running the model with various physical package configurations Θ_i^{train} , and comparing the model forecast against the observations at all past times τ to obtain the corresponding errors $\Delta_{\tau,i}^{\text{train}}$ for $\tau \leq t$ and $i \in \{\text{training data set}\}$. The output data is the corresponding physical combinations Θ that leads to the input error threshold.

In order to estimate the combinations of physical process configuration that contribute most to the uncertainty in predicting precipitation we take the following approach. The dataset consisting of the observable discrepancies during the current time window Δ_t is split into a training part and a testing part. In the test phase we use the approximated function $\hat{\phi}^{\text{physics}}$ to estimate the physical process settings $\hat{\Theta}_j^1$ that are consistent with the observable errors $\Delta_{t,j}^{(1)}$. Here we select $\Delta_{t,j}^{(1)} = \Delta_{t,j}^{\text{test}}$ for each $j \in \{\text{test data set}\}$. Note that in this case, since we know what physics has been used for the current results, one can take $\hat{\Theta}_j^{(1)}$ to be the real parameter values $\Theta_j^{(1)}$ used to generate the test data. However, in general, one selects $\Delta_{t,j}^{(1)}$ in an application-specific way and the corresponding parameters need to be estimated.

Next, we reduce the desired forecast error level to $\Delta_{t,j}^{(2)} = \Delta_{t,j}^{(1)}/2$, and use the approximated function $\hat{\phi}^{\text{physics}}$ to estimate the physical process setting $\hat{\Theta}_j^{(2)}$ that corresponds to this more accurate forecast. To identify the package setting that has the largest impact on the observable error we monitor the variability in the predicted parameters $\hat{\Theta}_j^{(2)} - \hat{\Theta}_j^{(1)}$. Specifically, the number of times the setting of a physical process in $\hat{\Theta}_j^{(2)}$ is different from its setting in $\hat{\Theta}_j^{(1)}$ is an indicator of the variability in model prediction when that package is changed. A higher variability in predicted physical packages implies a larger contribution towards the model errors - as estimated by the ML model.

8.3.3 Machine learning algorithms

In order to approximate the functions (8.9) and (8.11) discussed earlier we use regression machine learning methods. Choosing a right learning algorithm to use is challenging as it largely depends on the problem and the data available [188, 235, 149, 226]. Here, we use Random Forests (RF) and Artificial Neural Networks (ANN) as our learning algorithms [81]. Both RF and ANN algorithms can handle non-linearity in regression and classification. Given that the physical phenomena governing precipitation are highly nonlinear, and atmospheric dynamics is chaotic, we believe that RF and ANN approaches are well suited to capture the associated features. We briefly review these techniques next.

Random forests

A random forest [195] is an ensemble based method that constructs multiple decision trees. The principle idea behind ensemble methods is that a group of weak learners can come together to form a strong learner [193, 195]. The decision tree is built top-down from observations of target variables. The observation dataset is partitioned, smaller subsets are represented in branches, and decisions about the target variables are represented in the leaves.

There are many specific decision-tree algorithms available, including ID3 (Iterative Dichotomiser 3) [197], C4.5 (successor of ID3) [198], CART (Classification And Regression Tree), CHAID (CHi-squared Automatic Interaction Detector), and conditional inference trees [199]. If the dataset has multiple attributes, one can decide which attribute to place at the root or at different levels of the tree by considering different criteria such as information gain or the gini index [236].

Trees can be non-robust, with small changes in the tree leading to large changes in regression results. Moreover, trees tend to over-fit the data [237]. The random forest algorithm uses the bagging technique for building an ensemble of decision trees which are accurate and powerful at handling large, high dimensional datasets. Moreover, the bagging technique greatly reduces the variance [191]. For each tree in the forest, a bootstrap sample [193, 191] is selected from the dataset and instead of examining all possible feature-splits, some subset of the features is selected [200]. The node then splits on the best feature in the subset. By using a random sample of features the correlation between trees in the ensemble decreases, and the learning for each tree is much faster by restricting the features considered for each node.

Artificial neural networks

ANN is a computational model inspired by human brain's biological structure. ANN consist of neurons and connections between the neurons (weights) which are organized in layers. At least three layers of neurons (an input layer, a hidden layer, and an output layer) are required for construction of a neural network, where the input layer distributes the input signals to the first hidden layer. The feed-forward operation in a network passes information to neurons in a subsequent hidden layer. The neurons combine this information, and the output of each layer is obtained by passing the combined information through a differentiable transfer function that can be log-sigmoid, hyperbolic tangent sigmoid, or linear transfer function.

In supervised learning the network is provided with samples from which it discovers the relations of inputs and outputs. The learning problem consists of finding the optimal parameters of network such that the error between the desired output and the output signal of the network is minimized. The network first is initialized with randomly chosen weights and then the error is back-propagated through the network using a gradient descent method. The gradient of the error function is computed and used to modify weights and biases such that the error between the desired output and the output signal of the network is minimized [133, 132]. This process is repeated iteratively until the network output is close to the desired output [134].

8.4 Numerical experiments

We apply the proposed learning models to the Weather Research and Forecasting model [216] in order to:

- predict the bias in precipitation forecast caused by structural model errors,
- predict the statistics associated with the precipitation errors, and
- identify the specific physics packages that contribute most to precipitation forecast errors for given meteorological conditions.

8.4.1 The WRF model

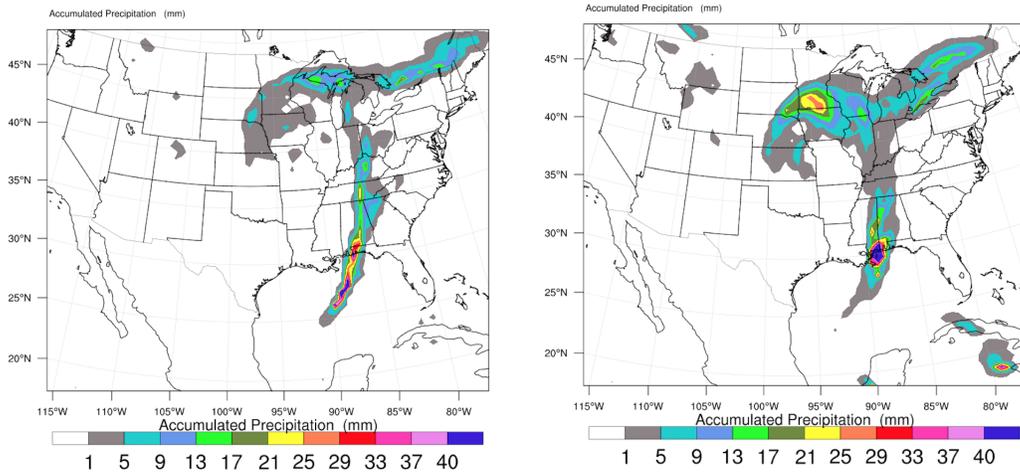
In this study we use the non-hydrostatic WRF model version 3.3. The simulation domain, shown in Fig. 8.1, covers the continental United States and has dimensions of 60×73 horizontal grid points in the west-east and south-north directions respectively, with a horizontal grid spacing of 60km [238]. The grid has 60 vertical levels to cover the troposphere and lower part of the stratosphere between the surface to approximately 20km . In all simulations, the 6-hourly analysis from the National Centers for Environmental Prediction (NCEP) are used as the initial and boundary conditions of the model [239]. The stage IV estimates are available at an hourly temporal resolution over continental United States. For experimental purposes, we use the stage IV NCEP analysis as a proxy for the true state of the atmosphere. The simulation window begins at 6AM UTC (Universal Time Coordinated) on May 1st 2017, and the simulation time is a six hour window time the same day. The “true” states of the atmosphere are available using the NCEP analysis data hourly. All the numerical experiments use the NCEP analysis data to run WRF model on May 1st 2017.

The model configuration parameters Θ represent various combinations of micro-physics schemes, cumulus parameterizations, short wave, and long wave radiation schemes. Specifically, each process is represented by the schema values of each physical parameter it uses, as detailed in WRF model physics options and references [240]. The micro-physics option provides atmospheric heat and moisture tendencies in atmosphere which also accounts for the vertical flux of precipitation and the sedimentation process. The cumulus parameterization is used to vertically redistribute heat and moisture independent of latent heating due to precipitation. The long wave radiation considers clear-sky and cloud upward and downward radiation fluxes and the short wave radiation considers clear-sky and cloudy solar fluxes.

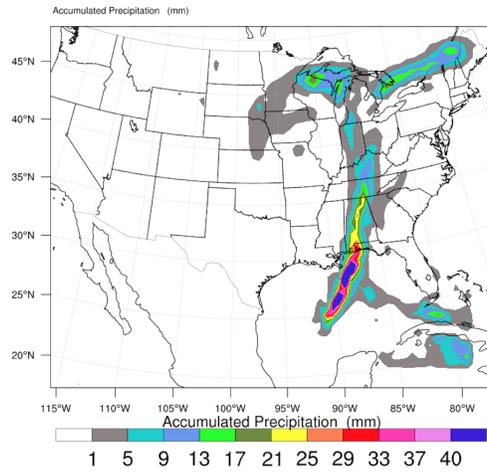
A total number of 252 combinations of the four physical modules are used in the simulations. The micro-physics schemes include: Kessler [241], Lin [242], WSM3 Hong [243], WSM5 Hong [243], Eta (Ferrier), WSM6 [244], Goddard [245], Thompson [246], Morrison [247]. The cumulus physics schemes applied are: Kain-Fritsch [248], Betts-Miller-Janjic [249], Grell Freitas[250]. The long wave radiation physics include: RRTM [251], Cam [252]. Short wave radiation physics include: Dudhia [253], Goddard [254], Cam [252].

For each of the 252 different physics combinations, the effect of each physics combination on precipitation is investigated. The NCEP analysis grid points are 428×614 , while the WRF computational model have 60×73 grid points. For obtaining the discrepancy between the WRF forecast and NCEP analysis we linearly interpolate the analysis

to transfer the physical variables onto the model grid. Figure 8.1(a) and 8.1(b) shows the NCEP analysis at 6AM and 12PM on 5/1/2017 which are used as initial condition and “true” (verification) state, respectively. The WRF forecast corresponding to the physics micro-physics: Kessler, cu-physics: Kain-Fritsch, ra-lw-physics: Cam , ra-sw-physics: Dudhia is illustrated in Figure 8.1(c). Figure 8.2 shows contours of discrepancies at 12PM ($\Delta_{t=12PM}$) discussed in equation (8.5) for two different physical combinations, which illustrates the effect that changing the physical schemes has on the forecast.

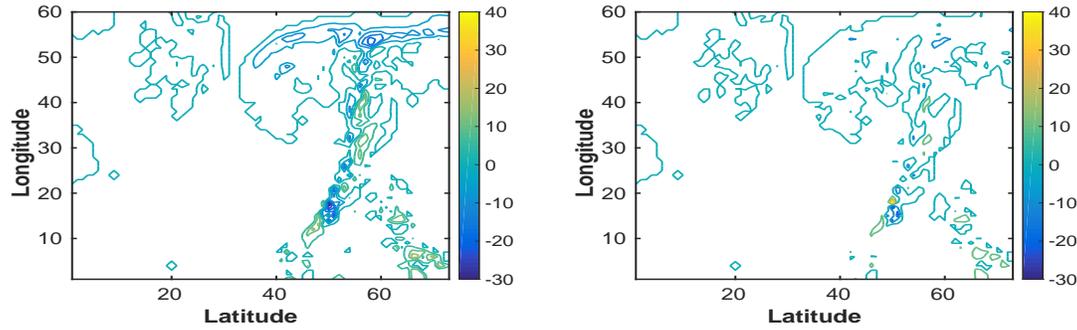


(a) NCEP analysis at 6AM provides initial conditions
 (b) NCEP analysis at 12PM provides a proxy for the true state of the atmosphere



(c) WRF forecast at 12PM corresponding to the physics micro-physics: Kessler, cumulus physics: Kain-Fritsch, long wave radiation physics: Cam, short wave radiation physics: Dudhia

Figure 8.1: Initial conditions, the analysis and the WRF forecast for the simulation time 12PM on 5/1/2017. Shown in the plots are the accumulated precipitation in millimeter unit.



(a) Micro-physics scheme: Kessler, cumulus physics: Kain-Fritsch, short wave radiation: Cam, long wave radiation: Dudhia
 (b) micro-physics scheme: Lin, cumulus physics: Kain-Fritsch, short wave radiation: RRTM Mlawer, long wave radiation: Cam

Figure 8.2: Shown in the plots are contours of observable discrepancies which are the differences in the accumulated precipitation results of WRF forecast against the analysis data $\Delta_{t=12PM}$ on 5/1/2017 for two different physics combinations. The observation operator extracts the precipitation solution from the WRF state vector.

8.4.2 Experiments for problem one: predicting pointwise precipitation forecast errors over a small geographic region

We demonstrate our learning algorithms to forecast precipitation in the state of Virginia on May 1st 2017 at 6PM. Our goal is to use the learning algorithms to correct the bias created due to model errors and hence improve the forecast for precipitation. As described in section 8.3.1, we learn the function ϕ^{error} of equation (8.9) using the training data from the previous forecast window (6AM to 12PM):

$$\phi^{\text{error}}(\Theta, \Delta_{\tau}, \mathbf{o}_{\tau}, \mathbf{o}_{t=12PM}) \approx \Delta_{t=12PM}, \quad 7AM \leq \tau < 12PM.$$

We use two learning algorithms to approximate the function ϕ^{error} , namely, the RF and ANN using Scikit-learn, machine learning library in Python [210]. The RF with ten trees and CART learning tree algorithm is used. The ANN with six hidden layers and hyperbolic tangent sigmoid activation function in each layer and linear activation function at last layer is employed. The number of layers and number of neurons in each layer are tuned empirically. For training purposes, we use the NCEP analysis of the May 1st 2017 at 6AM as initial conditions for the WRF model. The forecast window is 6 hours and the WRF model forecast final simulation time is 12PM. The input features are:

- The physics combinations (Θ).
- The hourly WRF forecasts projected onto observation space o_τ , $AM \leq \tau \leq 12PM$. The WRF state (\mathbf{x}_t) includes all model variables such as temperature, pressure, precipitation, etc. The observation operator extracts the precipitation portion of the WRF state vector, $\mathbf{o}_t \equiv \mathbf{x}_t^{\text{precipitation}}$. Accordingly, Δ_t is the discrepancy between WRF precipitation forecast \mathbf{o}_t and the observed precipitation \mathbf{y}_t .
- The observed discrepancies at past times (Δ_τ , $7AM \leq \tau < 12PM$).

The output variable is the discrepancy between the NCEP analysis and the WRF forecast at 12PM, i.e., the observable discrepancies for the current forecast window ($\Delta_{t=12PM}$). In fact, for each of the 252 different physical configurations, the WRF model forecast as well as the difference between the WRF forecast and the analysis are provided as input-output combinations for learning the function ϕ^{error} . The number of grid points over the state of Virginia is 14×12 . Therefore for each physical combination we have 168 grid points, and the total number of samples in the training data set is $252 \times 168 = 42,336$ with 15 features.

Both ANN and RF are trained with the above input-output combinations described above and during the training phase, the learning model learns the effect of interaction between different physical configurations on the WRF forecast and model error and obtains the approximation to the function ϕ^{error} which we denote by $\hat{\phi}^{\text{error}}$. The goal is to have more accurate forecast in the future time windows. We don't have the analysis data of future time windows but we can run WRF for future time windows and also predict the future model error using the approximated function $\hat{\phi}^{\text{error}}$. Once we obtain the predicted model error we can use that information in order to raise the accuracy of WRF forecast. In the testing phase we use the function $\hat{\phi}^{\text{error}}$ to predict the future forecast error $\hat{\Delta}_{t=6PM}$ given the combination of physical parameters as well as the WRF forecast at time 6PM as input features.

$$\hat{\Delta}_{t=6PM} \approx \hat{\phi}^{\text{error}}(\Theta, \Delta_\tau, \mathbf{o}_\tau, \mathbf{o}_{t=6PM}), \quad 1PM \leq \tau < 6PM.$$

To quantify the accuracy of the predicted error we calculate the Root Mean Squared Error (RMSE) between the true and predicted discrepancies at 6PM:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{\Delta}_{t=6PM}^i - \Delta_{t=6PM}^i)^2}, \quad (8.12)$$

where $n = 168$ is the number of grid points over Virginia, $\hat{\Delta}_{t=6PM}^i$ is the predicted discrepancy in the i^{th} grid point, and $\Delta_{t=6PM}^i$ is the i^{th} actual discrepancy in the i^{th} grid point.

The actual discrepancy is obtained as the difference between the NCEP analysis and WRF forecast at time $t = 6\text{PM}$. This error metric is computed for each of the 252 different configurations of the physics. The minimum, maximum and average RMSE over the 252 runs is reported in Table 8.1.

	minimum(RMSE)	average(RMSE)	maximum(RMSE)
ANN	1.264×10^{-3}	1.343×10^{-3}	5.212×10^{-3}
RF	1.841×10^{-3}	1.931×10^{-3}	7.9×10^{-3}

Table 8.1: The minimum, average, and maximum RMSE between the predicted $\hat{\Delta}_{t=6\text{PM}}$ and the true $\Delta_{t=6}$ over 252 physics combinations.

The predicted discrepancy in the observation space $\hat{\Delta}_{t=6\text{PM}}$ can be used to approximate the discrepancy in the model space $\hat{\delta}_{t=6\text{PM}}$ using equation (8.10). Here all the grid points are observed and therefore the error in the model space equal to the error in the observation space. Next, the estimate forecast error can be used to correct the forecast bias caused by model errors using (8.7), and hence to improve the forecast at 6PM: $\hat{\mathbf{x}}_{t=6\text{PM}} = \mathbf{x}_{t=6\text{PM}} + \hat{\delta}_{t=6\text{PM}}$. Figure 8.3(a) shows the WRF forecast for 6PM for the state of Virginia using the following physics packages (the physics options are given in parentheses):

- Micro-physics (Kessler),
- Cumulus-physics (Kain),
- Short-wave radiation physics (Dudhia),
- Long-wave radiation physics (Janjic).

Figure 8.3(b) shows the NCEP analysis at time 6PM, which is our proxy for the true state of the atmosphere. The discrepancy between the NCEP analysis and the raw WRF forecast is shown in the Figure 8.4(a). Using the model error prediction we can improve the WRF result by adding the predicted bias to the WRF forecast. The discrepancy between the corrected WRF forecast and the NCEP analysis is shown in the Figure 8.4(b). The results show a considerable reduction of model errors as compared to the uncorrected forecast of Figure 8.4(a). Table 8.2 shows the minimum and average of original model error vs the improved model errors.

	minimum($\Delta_{t=6PM}$)	average($\Delta_{t=6PM}$)
Original forecast	6.751×10^{-2}	5.025×10^{-1}
Improved forecast	2.134×10^{-4}	6.352×10^{-2}

Table 8.2: The minimum and average of $\Delta_{t=6PM}$ for the original WRF forecast vs the improved forecast

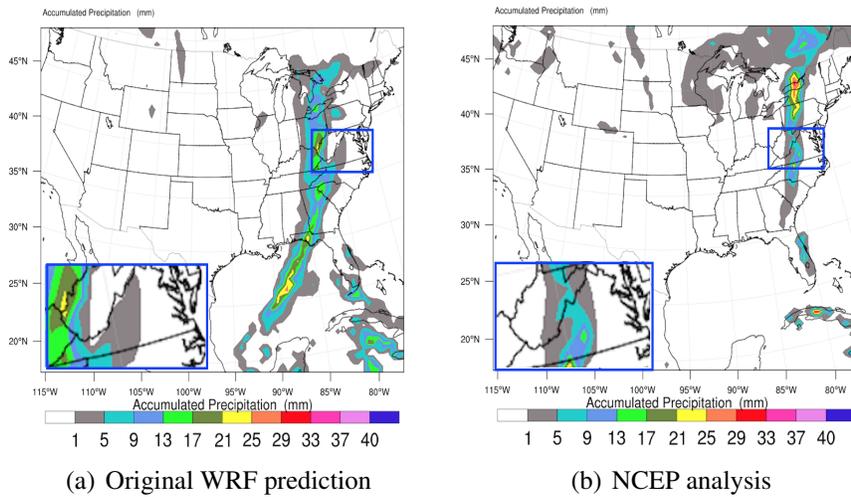


Figure 8.3: WRF prediction and NCEP analysis at 6PM on 5/1/2017. Zoom-in panels show the predictions over Virginia.

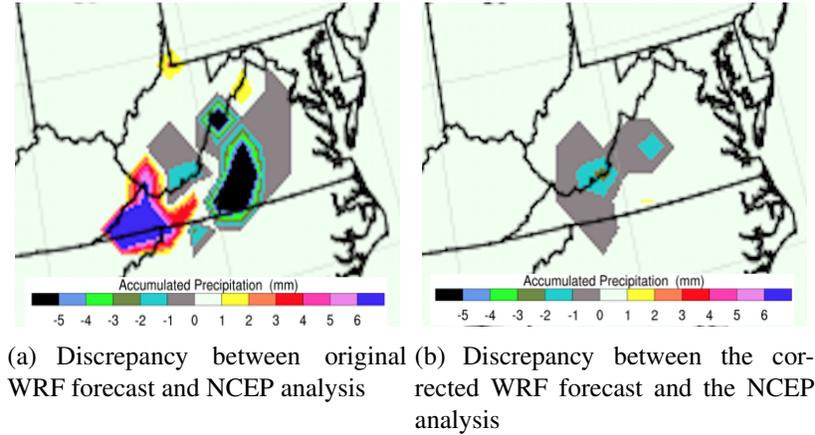


Figure 8.4: Discrepancy between WRF forecasts and the NCEP analysis over Virginia at 6PM on 5/1/2017. The forecast correction clearly improves the model results.

8.4.3 Experiments for problem one: predicting the norm of precipitation forecast error over the entire domain

We now seek to estimate the two-norm of precipitation model error over the entire continental U.S., which gives a global metric for the accuracy of the WRF forecast, and helps provide insight about the physics configurations that result in more accurate forecasts. To this end the following mapping is constructed:

$$\phi^{\text{error}}(\Theta, \|\mathbf{o}_\tau\|_2, \|\Delta_\tau\|_2, \|\mathbf{o}_{t=12\text{PM}}\|_2, \bar{\mathbf{o}}_{t=12\text{PM}}) \approx \|\Delta_{t=12\text{PM}}\|_2, \quad 7\text{AM} \leq \tau < 12\text{PM}.$$

To build the training dataset, we run WRF with each of the 252 different physical configurations. The forecast window is 6 hours and the WRF model forecast final simulation time is at 12PM. The hourly WRF forecast and discrepancy between the analysis and WRF forecast is used as training features.

The input features are:

- different physics schemes (Θ),
- the norms of the WRF model predictions at previous time windows, as well as at the current time ($\|\mathbf{o}_{t=12\text{PM}}\|_2, \|\mathbf{o}_\tau\|_2, 7\text{AM} \leq \tau < 12\text{PM}$), and

- the norms of past observed discrepancies ($\|\Delta_\tau\|_2$, $7\text{AM} \leq \tau < 12\text{PM}$).

The output variable is the norm of the discrepancy between WRF precipitation prediction and the NCEP precipitation analysis for the current time window ($\|\Delta_{t=12\text{PM}}\|_2$).

We use two different learning algorithms, namely, RF with ten trees in the forest and ANN with four hidden layers, the hyperbolic tangent sigmoid activation function in each layer and linear activation function at last layer. The number of layers and neurons at each layer is tuned empirically. The total number of samples in the training set is 252 with 15 of features. During the training phase the model learns the effect of interaction of different physical configurations on model error and obtains the approximated function $\hat{\phi}^{\text{error}}$.

In the test phase we feed the approximated function the model information from 1PM to the endpoint of the next forecast window 6PM to predict the norm of the model error $\|\hat{\Delta}_{t=6\text{PM}}\|_2$.

$$\hat{\phi}^{\text{error}}(\Theta, \|\mathbf{o}_\tau\|_2, \|\Delta_\tau\|_2, \|\mathbf{o}_{t=6\text{PM}}\|_2, \bar{\mathbf{o}}_{t=6\text{PM}}) \approx \|\Delta_{t=6\text{PM}}\|_2, \quad 1\text{PM} \leq \tau < 6\text{PM}.$$

Validation of the learned error mapping Table 8.3 shows the RMSE between the actual and predicted norms of discrepancies for ANN and RF. The RMSE is taken over the 252 runs with different physics combinations. Both learning models are doing well, with the ANN giving slightly better results than the RF.

	$RMS E(\ \hat{\Delta}_{t=6\text{PM}}\ _2, \ \Delta_{t=6\text{PM}}\ _2)$
ANN	2.6109×10^{-3}
RF	2.9188×10^{-3}

Table 8.3: Difference between predicted discrepancy norm $\|\hat{\Delta}_{t=6\text{PM}}\|_2$ and the reference discrepancy norm $\|\Delta_{t=6\text{PM}}\|_2$. The *RMS E* is taken over all test cases.

Analysis of the best combination of physical packages Based on our prediction of the norm of model error, the best physics combination that leads to lowest norm of precipitation error over the entire continental U.S. for the given meteorological conditions is:

- the BMJ cumulus parameterization, combined with
- the WSM5 micro-physics,
- Cam long wave, and

- Dudhia short wave radiation physics.

According to the true model errors, the best physics combination leading to the lowest norm of model error is achieved using the BMJ cumulus parameterization, combined with the WSM5 micro-physics, Cam long wave, and Cam short wave radiation physics.

8.4.4 Experiments for problem two: identify the physical processes that contribute most to the forecast uncertainty

The interaction of different physical processes greatly affects precipitation forecast, and we are interested in identifying the major sources of model errors in WRF. To this end we construct the physics mapping (8.11) using the norm and the statistical characteristics of the model-data discrepancy (over the entire U.S.) as input features:

$$\phi^{\text{physics}}(\bar{\Delta}_{t=12\text{PM}}, \|\Delta_{t=12\text{PM}}\|_2) \approx \Theta.$$

Statistical characteristics include the mean, minimum, maximum, and variance of the field across all grid points over the continental U.S. Note that this is slightly different than (8.11) where the inputs are the raw values of these discrepancies for each grid point. The output variable is the combination of physical processes Θ that leads to model errors consistent with the input pattern $\bar{\Delta}_{t=12\text{PM}}$ and $\|\Delta_{t=12\text{PM}}\|_2$.

To build the dataset, the WRF model is simulated for each of the 252 different physical configurations, and the mismatches between the WRF forecasts and the NCEP analysis at the end of the current forecast window are obtained. Similar to the previous experiment, the initial conditions used in the WRF model is the NCEP analysis for the May 1st 2017 at 6AM. The forecast window is 6 hours and the WRF model forecast is obtained for time 12PM. The discrepancy between the NCEP analysis at 12PM and WRF forecast at 12PM forms the observable discrepancy for the current forecast window $\Delta_{t=12\text{PM}}$. For each of the 252 different physical configurations, this process is repeated and statistical characteristics of the WRF forecast model error $\bar{\Delta}_{t=12\text{PM}}$, and the norm of model error $\|\Delta_{t=12\text{PM}}\|_2$ are used as feature values of the function ϕ^{physics} .

Validation of the learned physics mapping From all the collected data points, 80% (202 samples) are used for training the learning model, and the remaining 20% (50 samples) are used for testing purposes.

The RF has default ten trees in the forest and ANN has four hidden layers and hyperbolic tangent sigmoid activation function in each layer with linear activation function at last

layer. The number of layers and neurons at each layer is tuned empirically. The learning model uses the training dataset to learn the approximate mapping $\hat{\phi}^{\text{physics}}$. This function is applied to each of the 50 test samples $\Delta_{t=12\text{PM}}^{\text{test}}$ to obtain the predicted physical combinations $\hat{\Theta}_1$. In order to evaluate these predictions, we run the WRF model again with the $\hat{\Theta}_1$ physical setting and obtain the new forecast $\hat{\Theta}_{t=12\text{PM}}$, and the corresponding observable discrepancy $\hat{\Delta}_{t=12\text{PM}}^{\text{test}}$. The RMSE between the norm of actual observable discrepancies and the norm of predicted discrepancies are shown in Table 8.4. The small values of the difference demonstrates the performance of the learning algorithm.

	$RMS E(\ \hat{\Delta}_{t=12\text{PM}}^{\text{test}}\ _2, \ \Delta_{t=12\text{PM}}^{\text{test}}\ _2)$
ANN	4.1376×10^{-3}
RF	5.8214×10^{-3}

Table 8.4: The RMSE between estimated discrepancy using predicted physical combinations $\hat{\Delta}_{t=12\text{PM}}^{\text{test}}$ and the reference discrepancy $\Delta_{t=12\text{PM}}^{\text{test}}$.

Analysis of variability in physical settings We repeat the test phase for each of the 50 test samples with the scaled values of observable discrepancies $\Delta_{t=12\text{PM}}^{\text{test}}/2$ as inputs, and obtain the predicted physical combinations $\hat{\Theta}_2$. Large variability in the predicted physical settings $\hat{\Theta}$ indicate that the respective physical packages variability have a strong influence on the WRF forecast error. We count the number of times the predicted physics $\hat{\Theta}_2$ is different from $\hat{\Theta}_1$ when the input data spans the entire test data set.

The results shown in Figure 8.5 indicate that micro-physics and cumulus physics are not too sensitive to the change of input data, whereas short-wave and long-wave radiation physics are quite sensitive to changes in the input data. Therefore our learning model indicates that having an accurate short-wave and long-wave radiation physics package will aid in greatly reducing the uncertainty in precipitation forecasts due to missing/incorrect physics.

8.5 Conclusions

This study proposes a novel use of machine learning techniques to understand, predict, and reduce the uncertainty in the WRF model precipitation forecasts due to the interaction of several physical processes included in the model.

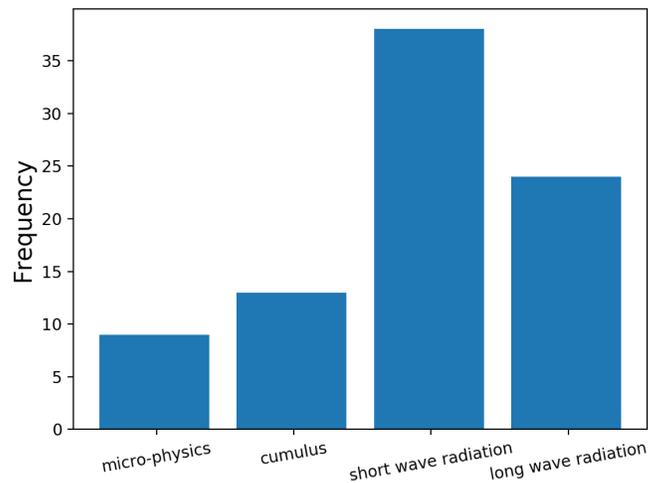


Figure 8.5: Frequency of change in the physics with respect to change in the input data from $\Delta_{t=12\text{PM}}^{\text{test}}$ to $\Delta_{t=12\text{PM}}^{\text{test}}/2$. Each data set contains 50 data points, and we report here the number of changes of each package.

We construct probabilistic approaches to learn the relationships between the configuration of the physical processes used in the simulation and the observed model forecast errors. These relationships are then used to solve two important problems related to model errors, as follows: estimating the systematic model error in a quantity of interest at future times, and identifying the physical processes that contribute most to the forecast uncertainty in a given quantity of interest under specified conditions.

Numerical experiments are carried out with the WRF model using the NCEP analyses as a proxy for the real state of the atmosphere. Ensembles of model runs with different parameter configurations are used to generate the training data. Random forests and Artificial neural network models are used to learn the relationships between physical processes and forecast errors. The experiments validate the new approach, and illustrates how it is able to estimate model errors, indicate best model configurations, and pinpoint to those physical packages that influence most the WRF prediction accuracy.

While the numerical experiments are done with WRF, and are focused on forecasting precipitation, the methodology developed herein is general and can be applied to the study of errors in other models, for other quantities of interest, and for learning additional relationships between model physics and model errors.

Chapter 9

Conclusions and Future Research Directions

Large scale simulation models are essential to understand real world phenomenon. A fine-tuned simulation model can be used to forecast the behavior of the real system. However, there still exists a discrepancy between the model forecast and the real observations of reality. This disparity is arising due to our assumptions of a perfect model, perfect model inputs, and parameters. The uncertainty analysis research plays an important role to detect the source of uncertainty, quantify and analyze the uncertainty in model forecasts. Further, the uncertainty analysis information can be merged with model forecast for uncertainty reduction and model forecast amelioration.

This dissertation develops novel techniques and methodologies for uncertainty analysis. We investigate different types of uncertainty in computational models, including the uncertainty due to lack of knowledge about the true physics of the natural phenomenon, uncertainty due to interaction of physical processes in model, uncertainty due to sampling in ensemble based approaches, and uncertainty due to model order reduction, etc. We explore different origins of uncertainty in models, probing uncertainty quantification and reduction techniques in the context of several types of models, including stochastic simulation of chemical reactions, physical models, reduced-order models, data assimilation, and numerical weather prediction models.

For each of the models, we obtain the discrepancy between the model prediction and the real phenomenon observation. The discrepancy in time have valuable information about the root of uncertainty and guide us to develop and apply different methodologies such as data fitting approaches, regression machine learning algorithms, and state space

models, in order to model the error, quantify the uncertainty, and finally improve the model predictions using our uncertainty analysis information.

In stochastic simulation of chemical reactions we derive new numerical algorithms that approximate the exact exponential solution of chemical master equation. These approximation techniques are further used following a statistical approach which reduce the uncertainty of tau-leap method when using large time steps.

The structural uncertainty in physics-based computer models due to our incomplete knowledge about the true physical processes is studied. This uncertainty manifests itself as missing dynamics in the model. The dynamics of these errors is modeled using a state-space approach, which enables to identify the source of uncertainty and to recognize the missing dynamics inside model. Furthermore, the model solution is improved by taking into account the error predicted by the state-space approach.

In the context of reduced-order models, we introduce multivariate input-output models to predict the errors and bases dimensions of local parametric reduced-order models. We refer to these mappings as multivariate predictions of local reduced-order model characteristics (MP-LROM) models. We approximate these multivariate mappings using machine learning regression techniques. Furthermore, we illustrate the application of MP-LROM error models for more efficient design of reduce-order models. The application develops an algorithm for local parametric domain decomposition of reduce-order models which is used following interpolation and concatenation techniques to reduce the uncertainty and raise the accuracy of the reduced-order models. In the future we seek to increase the efficiency of MP-LROM error models by declining the need for high-fidelity simulations of the training data. Using error bounds, residual norms and a-posteriori error estimation decreases the demand for high-fidelity simulations.

In data assimilation, we introduce adaptive localization for the EnKF family of methods. The EnKF family suffers from sampling error when the sample size is not enough to represent well the true state of the system. We propose a machine learning based approach for adaptive localization which reduces the uncertainty originating from sampling error. The first one is the adaptive localization in time, where the localization radius changes at each assimilation cycle. The second one is the adaptive localization in space and time where the localization radius changes both at each time step as well as each system state. The future research in this area focuses on study and resolving the existing challenges including: constraint optimization for doing the localization for each state independently without model physics violation, choosing the better localization radii from the possible localization radii and feature selection method for very big scale models.

In numerical weather prediction models, we study uncertainty quantification and reduc-

tion in WRF model forecast due to interaction of physical processes that affect the accuracy of precipitation prediction. We tried to quantify and further reduce the uncertainty of model prediction using past discrepancy data between model forecast and reality. A machine learning approach uses the past discrepancy data to predict the future model error which then is used to enhance the original model forecast. Furthermore, we analyzed what physical processes cause more error in model prediction that needs to be changed or improved. The future research focuses on using the past observable discrepancy data as time series and apply deep learning algorithms Recurrent Neural Networks (RNN) or Long Short Term Memory (LSTM) networks in order to predict the future model error more accurately.

Bibliography

- [1] Dan Cacuci. *Sensitivity and Uncertainty Analysis: Theory*. Chapman and Hall/CRC Press, 2003.
- [2] Dan G. Cacuci. Second-order adjoint sensitivity analysis methodology (2nd-asam) for computing exactly and efficiently first- and second-order sensitivities in large-scale linear systems: I. computational methodology. *Journal of Computational Physics*, 284(0):687 – 699, 2015.
- [3] Dan G Cacuci. Second-order adjoint sensitivity analysis methodology (2nd-asam) for computing exactly and efficiently first-and second-order sensitivities in large-scale linear systems: Ii. illustrative application to a paradigm particle diffusion problem. *Journal of Computational Physics*, 284:700–717, 2015.
- [4] OP Le Maître and OM Knio. Spectral methods for uncertainty quantification. scientific computation, 2010.
- [5] MS Eldred and John Burkardt. Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. *AIAA paper*, 976(2009):1–20, 2009.
- [6] A.K. Alekseev, I.M. Navon, and M.E. Zelentsov. The estimation of functional uncertainty using polynomial chaos and adjoint equations. *International Journal for Numerical Methods in Fluids*, 67(3):328–341, 2011.
- [7] Gillespie, D.T. Exact stochastic simulation of coupled chemical reactions. *Journal of Chemical Physics*, 81(25):2340–2361, 1977.
- [8] Gillespie, D.T. Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics*, 115(4):1716–1733, 2001.

- [9] Jens Christian Refsgaard, Jeroen P Van der Sluijs, James Brown, and Peter Van der Keur. A framework for dealing with uncertainty due to model structure error. *Advances in Water Resources*, 29(11):1586–1597, 2006.
- [10] Andrew H Briggs, Milton C Weinstein, Elisabeth AL Fenwick, Jonathan Karnon, Mark J Sculpher, and A David Paltiel. Model parameter estimation and uncertainty analysis a report of the ISPOR-SMDM modeling good research practices task force working group–6. *Medical Decision Making*, 32(5):722–732, 2012.
- [11] Mark Strong and Jeremy E Oakley. When is a model good enough? deriving the expected value of model improvement via specifying internal model discrepancies. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):106–125, 2014.
- [12] Ralph C Smith. *Uncertainty Quantification: Theory, Implementation, and Applications*, volume 12. SIAM, 2013.
- [13] Shawn E Gano, John E Renaud, and Brian Sanders. Hybrid variable fidelity optimization by using a kriging-based scaling function. *Aiaa Journal*, 43(11):2422–2433, 2005.
- [14] Pierre Sagaut. *Large eddy simulation for incompressible flows: an introduction*. Springer Science & Business Media, 2006.
- [15] David C Wilcox et al. *Turbulence modeling for CFD*, volume 2. DCW industries La Canada, CA, 1998.
- [16] P. Courtier, J.N. Thepaut, and A. Hollingsworth. A strategy for operational implementation of 4D-VAR, using an incremental approach. *Quart. J. Roy. Meteor. Soc.*, 120:1367–1387, 1994.
- [17] Yannick Trémolet. Incremental 4d-var convergence study. *Tellus A*, 59(5):706–718, 2007.
- [18] Michael Hinze and Stefan Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. In *Dimension Reduction of Large-Scale Systems*, pages 261–306. Springer, 2005.
- [19] Christophe Prud’homme, Dimitrios V Rovas, Karen Veroy, Luc Machiels, Yvon Maday, Anthony T Patera, and Gabriel Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *Journal of Fluids Engineering*, 124(1):70–80, 2002.

- [20] M.L. Rapún and J.M. Vega. Reduced order models based on local POD plus Galerkin projection. *Journal of Computational Physics*, 229(8):3046–3063, 2010.
- [21] M Dihlmann, M Drohmann, and B Haasdonk. Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning. *Proc. of ADMOS*, 2011, 2011.
- [22] Jens L Eftang and Benjamin Stamm. Parameter multi-domain hpempirical interpolation. *International Journal for Numerical Methods in Engineering*, 90(4):412–428, 2012.
- [23] Benjamin Peherstorfer, Daniel Butnaru, Karen Willcox, and Hans-Joachim Bungartz. Localized discrete empirical interpolation method. *SIAM Journal on Scientific Computing*, 36(1):A168–A192, 2014.
- [24] Rudolph E Kalman and Richard S Bucy. New results in linear filtering and prediction theory. *Journal of basic engineering*, 83(1):95–108, 1961.
- [25] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.
- [26] Gerrit Burgers, Peter J van Leeuwen, and Geir Evensen. Analysis scheme in the Ensemble Kalman Filter. *Monthly Weather Review*, 126:1719–1724, 1998.
- [27] Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics . *Journal of Geophysical Research*, 99(C5):10143–10162, 1994.
- [28] Peter L Houtekamer and Herschel L Mitchell. Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, 126(3):796–811, 1998.
- [29] Jeffrey L Anderson. Exploring the need for localization in ensemble data assimilation using a hierarchical ensemble filter. *Physica D: Nonlinear Phenomena*, 230(1):99–111, 2007.
- [30] Jeffrey Anderson and Lili Lei. Empirical localization of observation impact in Ensemble Kalman Filters. *Monthly Weather Review*, 141(11):4140–4153, 2013.
- [31] Jeffrey L Anderson. Localization and sampling error correction in Ensemble Kalman Filter data assimilation. *Monthly Weather Review*, 140(7):2359–2371, 2012.

- [32] R Petrie. Localization in the Ensemble Kalman Filter. *MSc Atmosphere, Ocean and Climate University of Reading*, 2008.
- [33] T. Kurtz. The relationship between stochastic and deterministic models for chemical reactions. *Journal of Chemical Physics*, 57(7):2976–2978, 1972.
- [34] Y. Cao, R. Petzold, M. Rathinam, D. Gillespie. The numerical stability of leaping methods for stochastic simulation of chemically reacting systems. *Journal of Chemical Physics*, 121(24):12169–12178, 2004.
- [35] M. Rathinam, L. Petzold, Y. Cao, D. Gillespie. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *Journal of Chemical Physics*, 119(24):784–12, 2003.
- [36] Gillespie, D., Petzold, L. Improved leap-size selection for accelerated stochastic simulation. *Journal of Chemical Physics*, 119(16):8229–8234, 2003.
- [37] A. Sandu T.H. Ahn. Implicit simulation methods for stochastic chemical kinetics. <http://arxiv.org/abs/1303.3614>, 2013.
- [38] Yang Cao, Daniel T Gillespie, and Linda R Petzold. The slow-scale stochastic simulation algorithm. *The Journal of chemical physics*, 122(1):014116, 2005.
- [39] Y. Cao, H. Li, L. Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *Journal of Chemical Physics*, 9(121):4059–4067, 2004.
- [40] M. Rathinam, L. Petzold, Y. Cao, D. Gillespie. Consistency and stability of tau leaping schemes for chemical reaction systems. *SIAM J. of Multiscale Modeling and Simulation*, 3(4):867–895, 2005.
- [41] T.H. Ahn, A. Sandu, L. Watson, C. Shaffer, Y. Cao, W. Baumann. Parallel load balancing strategies for ensembles of stochastic biochemical simulations. *Tech. rep., Virginia Tech.*, 2012.
- [42] S. MacNamara, K. Burrage, R.B. Sidje. Multiscale modeling of chemical kinetics via the master equation. *SIAM J. of Multiscale Modeling and Simulation*, 4(6):1146–1168, 2008.
- [43] S. Engblom. Numerical methods for the chemical master equation. *Ph.D. thesis, Uppsala University, Department of Information Technology*, 2006.

- [44] A. Sandu. A new look at chemical master equation. *Numerical Algorithms*, 65(3):485–498, 2013.
- [45] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 3(5):506–517, 1968.
- [46] Azam Mooasvi and Adrian Sandu. Approximate exponential algorithms to solve the chemical master equation. *Mathematical Modelling and Analysis*, 20(3):382–395, 2015.
- [47] R.B. Sidje. Expokit: A software package for computing matrix exponentials. *ACM Trans. Software.*, 24(1):130–156, 1998.
- [48] C. Moler and C.V. Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1), 2003.
- [49] M. Pusa. Rational approximations to the matrix exponential in burn up calculations. *Nuclear science and engineering.*, 169(2):155–167, 2010.
- [50] A. Altaieb and D. Chauveau. Bayesian analysis of the logit model and comparison of two Metropolis-Hastings strategies. *Comput. Stat. Data Anal.*, 39(2):137–152, 2002.
- [51] G. O. Roberts and A. F. M. Smith. Simple conditions for the convergence of the gibbs sampler and Metropolis-Hastings algorithms. *Stochastic Processes and their Applications*, 49(2):207–216, 1994.
- [52] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.
- [53] W.K. Hastings. Monte Carlo sampling methods using Markov Chains and their applications. *Biometrika*, 57, 1970.
- [54] D.B. Hitchcock. A history of the Metropolis-Hastings algorithm. *The American Statistician.*, 75(4):254–257, 2003.
- [55] S. Chib and E. Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician.*, 49(4):327–335, 1995.

- [56] C.A. Kastner, A. Braumann, L.W.P. Man, S. Mosbach, G.P.E. Brownbridge, J. Akroyd, M. Kraft, and C. Himawan. Bayesian parameter estimation for a jet-milling model using Metropolis Hastings and wanglandau sampling. *Chemical Engineering Science.*, 99(4):244–257, 2013.
- [57] Nicholas J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM review.*, 51(4):747–764, 2009.
- [58] AJ Carpenter, A Ruttan, and RS Varga. Extended numerical computations on the 1/9 conjecture in rational approximation theory. In *Rational Approximation and Interpolation*, pages 383–411. Springer, 1984.
- [59] A. Ferreira, A. Roussy, and L. Conde. Virtual metrology models for predicting physical measurement in semiconductor manufacturing. In *Conference: Advanced Semiconductor Manufacturing Conference*, 2009.
- [60] Sven Burger. Simulation method improves accuracy for optical metrology and nanooptics design. *SPIE Newsroom. Optical Design and Engineering.*, 2009.
- [61] T.J. Esward and L. Wright. Efficient updating of pde models for metrology. *Measurement*, 79:267 – 275, 2016.
- [62] B. Y. Lemeshko. Computer simulation as an effective technique in metrology and statistical product quality control. In *2004 7th International Conference on Actual Problems of Electronic Instrument Engineering Proceedings, 2004. APEIE 2004.*, pages 259–259, Sept 2004.
- [63] Giampaolo E D’Errico. The problem of false discoveries from a metrological point of view. *Journal of Physics: Conference Series*, 459(1):012041, 2013.
- [64] Martin Pilch, Timothy G Trucano, and Jon C Helton. Ideas underlying quantification of margins and uncertainties (QMU): a white paper. *Unlimited Release SAND2006-5001, Sandia National Laboratory, Albuquerque, New Mexico*, 87185:2, 2006.
- [65] Marc C Kennedy and Anthony O’Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.
- [66] GUM: guide to the expression of uncertainty in measurement, 2017.

- [67] Dave Higdon, Marc Kennedy, James C Cavendish, John A Cafeo, and Robert D Ryne. Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466, 2004.
- [68] Newsha K Ajami, Qingyun Duan, and Soroosh Sorooshian. An integrated hydrologic Bayesian multimodel combination framework: Confronting input, parameter, and model structural uncertainty in hydrologic prediction. *Water Resources Research*, 43(1), 2007.
- [69] Masaru Hoshiya and Etsuro Saito. Structural identification by extended Kalman filter. *Journal of Engineering Mechanics*, 110(12):1757–1770, 1984.
- [70] Alexandra Seiler, Jan C Rivenæs, Sigurd Ivar Aanonsen, and Geir Evensen. Structural uncertainty modelling and updating by production data integration. In *SPE/EAGE Reservoir Characterization & Simulation Conference*, 2009.
- [71] Derek Rowell. State-space representation of LTI systems. URL: <http://web.mit.edu/2.14/www/Handouts/StateSpace.pdf>, 2002.
- [72] Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, number 1, 1996.
- [73] Adrian Sandu. Time-stepping methods that favor positivity for atmospheric chemistry modeling. *IMA VOLUMES IN MATHEMATICS AND ITS APPLICATIONS*, 130:21–38, 2002.
- [74] Peter Van Overschee and BL De Moor. *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.
- [75] Mitchell Melanie. An introduction to genetic algorithms. *Cambridge, Massachusetts London, England, Fifth printing*, 3:62–75, 1999.
- [76] Mircea Grigoriu. *Stochastic systems: uncertainty quantification and propagation*. Springer Science & Business Media, 2012.
- [77] Dan G Cacuci, Mihaela Ionescu-Bujor, and Ionel Michael Navon. *Sensitivity and uncertainty analysis, volume II: applications to large-scale systems*, volume 2. Chapman and Hall/CRC Press, 2005.
- [78] Alexander Shapiro. Monte Carlo sampling methods. *Handbooks in Operations Research and Management Science*, 10:353–425, 2003.

- [79] Jon C Helton and Freddie Joe Davis. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety*, 81(1):23–69, 2003.
- [80] Christiane Lemieux. *Monte Carlo and quasi-Monte Carlo sampling*. Springer Science & Business Media, New York, NY, 2009.
- [81] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [82] TA Porsching. Estimation of the error in the reduced basis method solution of nonlinear equations. *Mathematics of Computation*, 45(172):487–496, 1985.
- [83] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.
- [84] M.A. Grepl and A.T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(01):157–181, 2005.
- [85] G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.
- [86] M. Dihlmann and B. Haasdonk. Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems. *Submitted to the Journal of Computational Optimization and Applications*, 2013.
- [87] K. Karhunen. Zur spektraltheorie stochastischer prozesse. *Annales Academiae Scientiarum Fennicae*, 37, 1946.
- [88] M.M. Loève. *Probability Theory*. Van Nostrand, Princeton, NJ, 1955.
- [89] H. Hotelling. Analysis of a complex of statistical variables with principal components. *Journal of Educational Psychology*, 24:417–441, 1933.
- [90] E.N. Lorenz. Empirical Orthogonal Functions and Statistical Weather Prediction. Technical report, Massachusetts Institute of Technology, Dept. of Meteorology, 1956.
- [91] John Leask Lumley. The structure of inhomogeneous turbulent flows. *Atmospheric turbulence and radio wave propagation*, pages 166–178, 1967.

- [92] L. Sirovich. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quarterly of Applied Mathematics*, 45(3):561–571, 1987.
- [93] L. Sirovich. Turbulence and the dynamics of coherent structures. II. Symmetries and transformations. *Quarterly of Applied Mathematics*, 45(3):573–582, 1987.
- [94] L. Sirovich. Turbulence and the dynamics of coherent structures. III. Dynamics and scaling. *Quarterly of Applied Mathematics*, 45(3):583–590, 1987.
- [95] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.
- [96] Natalia M Alexandrov, Robert Michael Lewis, Clyde R Gumbert, Lawrence L Green, and Perry A Newman. Approximation and model management in aerodynamic optimization with variable-fidelity models. *Journal of Aircraft*, 38(6):1093–1101, 2001.
- [97] Martin Drohmann and Kevin Carlberg. The romes method for statistical modeling of reduced-order-model error. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):116–145, 2015.
- [98] MS Eldred, AA Giunta, SS Collis, NA Alexandrov, and RM Lewis. Second-order corrections for surrogate-based optimization with model hierarchies. In *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, Aug*, pages 2013–2014, 2004.
- [99] Deng Huang, TT Allen, WI Notz, and RA Miller. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32(5):369–382, 2006.
- [100] Stefan Volkwein. Proper orthogonal decomposition: applications in optimization and control. *CEA-EDFINRIA Numerical Analysis Summer School*, 2007.
- [101] Marek Słowski. Bayesian neural networks and Gaussian processes in identification of concrete properties. *Computer Assisted Mechanics and Engineering Sciences*, 18(4):291–302, 2011.
- [102] M Lilley. *Gaussian processes as neural networks*. PhD thesis, Honours thesis, Victoria University of Wellington, 2004. Available from <http://www.mcs.vuw.ac.nz/people/Marcus-Frean>, 2004.

- [103] Clarence W Rowley, Tim Colonius, and Richard M Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1):115–129, 2004.
- [104] K. Kunisch and S. Volkwein. Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics. *SIAM Journal on Numerical Analysis*, 40(2):492–515, 2002.
- [105] C.W. Rowley. Model Reduction for Fluids, using Balanced Proper Orthogonal Decomposition. *International Journal of Bifurcation and Chaos (IJBC)*, 15(3):997–1013, 2005.
- [106] K. Willcox and J. Peraire. Balanced model reduction via the Proper Orthogonal Decomposition. *AIAA Journal*, pages 2323–2330, 2002.
- [107] B.R. Noack, M. Schlegel, M. Morzynski, and G. Tadmor. System reduction strategy for galerkin models of fluid flows. *International Journal for Numerical Methods in Fluids*, 63(2):231–248, 2010.
- [108] O. San and T. Iliescu. Proper orthogonal decomposition closure models for fluid flows: Burgers equation. Technical Report arXiv:1308.3276 [physics.flu-dyn], Cornell University, August 2013.
- [109] David Wells, Zhu Wang, Xuping Xie, and Traian Iliescu. Regularized reduced order models. *arXiv preprint arXiv:1506.07555*, 2015.
- [110] R. Stefanescu, A. Sandu, and I.M. Navon. Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations. *International Journal for Numerical Methods in Fluids*, 76(8):497–521, 2014.
- [111] Alfio Quarteroni, Gianluigi Rozza, and Andrea Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1(1):1–49, 2011.
- [112] JA Taylor and MN Glauser. Towards practical flow sensing and control via pod and lse based low-dimensional tools. *Journal of Fluids Engineering*, 126(3):337–345, 2004.
- [113] Ryan Schmit and Mark Glauser. Improvements in low dimensional tools for flow-structure interaction problems: using global pod. In *APS Division of Fluid Dynamics Meeting Abstracts*, volume 1, 2003.

- [114] Luca Daniel, Ong Chin Siong, Low Sok Chay, Kwok Hong Lee, and Jacob White. A multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 23(5):678–693, 2004.
- [115] DS Weile, E Michielssen, Eric Grimme, and K Gallivan. A method for generating rational interpolant reduced order models of two-parameter linear systems. *Applied Mathematics Letters*, 12(5):93–102, 1999.
- [116] Bernard Haasdonk and Mario Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 42(2):277–302, 2008.
- [117] Ngoc-Cuong Nguyen, Gianluigi Rozza, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous burgers’ equation. *Calcolo*, 46(3):157–185, 2009.
- [118] Paul G Constantine, Eric Dow, and Qiqi Wang. Active subspace methods in theory and practice: applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4):A1500–A1524, 2014.
- [119] Karl Kunisch and Stefan Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische Mathematik*, 90(1):117–148, 2001.
- [120] M. Hinze and S. Volkwein. Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition. *Computational Optimization and Applications*, 39:319–345, 2008.
- [121] Marco Fahl and Ekkehard W Sachs. Reduced order modelling approaches to PDE-constrained optimization based on proper orthogonal decomposition. In *Large-scale PDE-constrained optimization*, pages 268–280. Springer, 2003.
- [122] Clarence W Rowley, Tim Colonius, and Richard M Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1):115–129, 2004.
- [123] M. Rathinam and L. Petzold. A new look at proper orthogonal decomposition. *SIAM J. Numer. Anal.*, 41(5):1893–1925 (electronic), 2003.

- [124] Chris Homescu, Linda R Petzold, and Radu Serban. Error estimation for reduced-order models of dynamical systems. *SIAM Journal on Numerical Analysis*, 43(4):1693–1714, 2005.
- [125] C.M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.
- [126] Andrew Ng. Stanford CS229 , Machine Learning. Technical report, Stanford, 2008.
- [127] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptive computation and machine learning series. University Press Group Limited, 2006.
- [128] M. Hagan, H. Demuth, M. Beale, and O. De Jesus. *Neural Network Design (2nd Edition)*. Martin Hagan, 2014.
- [129] J.S.R. Jang, C.T. Sun, and E. Mizutani. *Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. MATLAB curriculum series. Prentice Hall, 1997.
- [130] Ramin Ayanzadeh, Azam S Zavar Mousavi, and Saeid Setayeshi. Fossil fuel consumption prediction using emotional learning in amygdala. In *Electrical Engineering (ICEE), 2011 19th Iranian Conference on*, pages 1–6. IEEE, 2011.
- [131] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [132] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.
- [133] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3):183–192, 1989.
- [134] S.S. Haykin. *Neural Networks and Learning Machines*. Number v. 10 in Neural networks and learning machines. Prentice Hall, 2009.
- [135] Johannes Martinus Burgers. A mathematical model illustrating the theory of turbulence. *Adv. in Appl. Mech.*, 1:171–199, 1948.
- [136] Martin Gubisch, Ira Neitzel, and Stefan Volkwein. A-posteriori error estimation of discrete POD models for PDE-constrained optimal control. Technical report, Konstanzer Schriften in Mathematik, 03 2016.

- [137] E. Arian, M. Fahl, and E.W. Sachs. Trust-region proper orthogonal decomposition for flow control. *ICASE: Technical Report 2000-25*, 2000.
- [138] M. Bergmann and L. Cordier. Optimal control of the cylinder wake in the laminar regime by trust-region methods and pod reduced-order models. *Journal of Computational Physics*, 227(16):7813–7840, 2008.
- [139] B.R. Noack, M. Schlegel, M. Morzynski M., and G. Tadmor. Galerkin Method for Nonlinear Dynamics. In B.R. Noack, M. Morzynski, and G. Tadmor, editors, *Reduced-Order Modelling for Flow Control*, volume 528, pages 111–149. Springer, 2011.
- [140] Bernd R Noack, Paul Papas, and Peter A Monkewitz. The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. *Journal of Fluid Mechanics*, 523:339–365, 2005.
- [141] RD Prabhu, S Scott Collis, and Yong Chang. The influence of control on proper orthogonal decomposition of wall-bounded turbulent flows. *Physics of fluids*, 13(2):520–537, 2001.
- [142] Bernd R Noack, Konstantin Afanasiev, MAREK MORZYŃSKI, Gilead Tadmor, and Frank Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.
- [143] J. Burkardt, M. Gunzburger, and H.C. Lee. POD and CVT-based reduced-order modeling of Navier–Stokes flows. *Computer Methods in Applied Mechanics and Engineering*, 196(1-3):337–355, 2006.
- [144] S. Kaulmann and B. Haasdonk. Online greedy reduced basis construction using dictionaries. In I. Troch and F. Breitenecker, editors, *Proceedings of 7th Vienna International Conference on Mathematical Modelling*, pages 112–117. 2012.
- [145] Yvon Maday and Benjamin Stamm. Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces. *SIAM Journal on Scientific Computing*, 35(6):A2417–A2441, 2013.
- [146] Brendan Tracey, Karthik Duraisamy, and Juan Alonso. Application of supervised learning to quantify uncertainties in turbulence and combustion modeling. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 259, 2013.

- [147] Julia Ling and J Templeton. Evaluation of machine learning algorithms for prediction of regions of high reynolds averaged navier stokes uncertainty. *Physics of Fluids*, 27(8):085103, 2015.
- [148] Brendan D Tracey. *Machine Learning for Model Uncertainties in Turbulence Models and Monte Carlo Integral Approximation*. PhD thesis, STANFORD UNIVERSITY, 2015.
- [149] Azam Moosavi, Razvan Stefanescu, and Adrian Sandu. Multivariate predictions of local reduced-order-model errors and dimensions. *International Journal for Numerical Methods in Engineering*, pages n/a–n/a, 2017. nme.5624.
- [150] Thuan Lieu and Michel Lesoinne. Parameter adaptation of reduced order models for three-dimensional flutter analysis. *AIAA Paper*, 888:2004, 2004.
- [151] D. Amsallem and C. Farhat. An Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity. *AIAA Journal*, 46(7):1803–1813, 2008.
- [152] Gary Weickum, MS Eldred, and Kurt Maute. Multi-point extended reduced order modeling for design optimization and uncertainty analysis. In *Proceedings of the 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (2nd AIAA Multidisciplinary Design Optimization Specialist Conference)*, Newport, RI, pages 2006–2145, 2006.
- [153] Toni Lassila, Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Generalized reduced basis methods and n-width estimates for the approximation of the solution manifold of parametric PDEs. In *Analysis and Numerics of Partial Differential Equations*, pages 307–329. Springer, 2013.
- [154] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematica*, 80(2):199–220, 2004.
- [155] Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. MATRIX ANAL. APPL*, 20(2):303–353, 1998.
- [156] Evgeni Begelfor and Michael Werman. Affine invariance revisited. In *Proceedings of the 2006 IEEE Computer Society Conference of Computer Vision and Pattern Recognition*, volume 2, pages 2087–2094, 2006.

- [157] Thuan Lieu, Charbel Farhat, and Michel Lesoinne. POD-based aeroelastic analysis of a complete F-16 configuration: ROM adaptation and demonstration. *AIAA Paper*, 2295:2005, 2005.
- [158] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU Press, Baltimore, fourth edition, 2013.
- [159] Tony F Chan. An improved algorithm for computing the Singular Value Decomposition. *ACM Transactions on Mathematical Software (TOMS)*, 8(1):72–83, 1982.
- [160] Ahmed Attia, Vishwas Rao, and Adrian Sandu. A Hybrid Monte Carlo sampling smoother for four dimensional data assimilation. *International Journal for Numerical Methods in Fluids*, 2016. fld.4259.
- [161] Ahmed Attia, Razvan Stefanescu, and Adrian Sandu. The reduced-order Hybrid Monte Carlo sampling smoother. *International Journal for Numerical Methods in Fluids*, 2016. fld.4255.
- [162] Eugenia Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge University Press, 2002.
- [163] Geir Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53, 2003.
- [164] Pieter L Houtekamer and Herschel L Mitchell. Data assimilation using an ensemble Kalman filter technique . *Monthly Weather Review*, 126:796–811, 1998.
- [165] Edward Ott, Brian R Hunt, Istvan Szunyogh, Aleksey V Zimin, Eric J Kostelich, Matteo Corazza, Eugenia Kalnay, DJ Patil, and James A Yorke. A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A*, 56(5):415–428, 2004.
- [166] Craig H Bishop, Brian J Etherton, and Sharanya J Majumdar. Adaptive sampling with the ensemble transform Kalman filter. part I: Theoretical aspects. *Monthly weather review*, 129(3):420–436, 2001.
- [167] Jeffrey L Anderson. An ensemble adjustment Kalman filter for data assimilation. *Monthly weather review*, 129(12):2884–2903, 2001.
- [168] Jeffrey L Anderson and Stephen L Anderson. A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127(12):2741–2758, 1999.

- [169] Thomas M Hamill, Jeffrey S Whitaker, and Chris Snyder. Distance-dependent filtering of background error covariance estimates in an Ensemble Kalman Filter. *Monthly Weather Review*, 129(11):2776–2790, 2001.
- [170] Gregory Gaspari and Stephen E Cohn. Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, 125:723–757, 1999.
- [171] Jeffrey L Anderson. An adaptive covariance inflation error correction algorithm for ensemble filters. *Tellus A*, 59(2):210–224, 2007.
- [172] Kay Bergemann and Sebastian Reich. A mollified Ensemble Kalman Filter. *Quarterly Journal of the Royal Meteorological Society*, 136(651):1636–1643, 2010.
- [173] Craig H Bishop and Daniel Hodyss. Flow-adaptive moderation of spurious ensemble correlations and its use in ensemble-based data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 133(629):2029–2044, 2007.
- [174] Craig H Bishop and Daniel Hodyss. Ensemble covariances adaptively localized with eco-rap. part 2: a strategy for the atmosphere. *Tellus A*, 61(1):97–111, 2009.
- [175] Lili Lei and Jeffrey L Anderson. Comparisons of empirical localization techniques for serial Ensemble Kalman Filter in a simple atmospheric general circulation model. *Monthly Weather Review*, 142(2):739–754, 2014.
- [176] Yicun Zhen and Fuqing Zhang. A probabilistic approach to adaptive covariance localization for serial ensemble square root filters. *Monthly Weather Review*, 142(12):4499–4518, 2014.
- [177] Paul Kirchgessner, Lars Nerger, and Angelika Bunse-Gerstner. On the choice of an optimal localization radius in Ensemble Kalman Filter methods. *Monthly Weather Review*, 142(6):2165–2175, 2014.
- [178] M.K. Tippett, J.J. Anderson, and C.H. Bishop. Ensemble square root filters. *Monthly Weather Review*, 131:1485–1490, 2003.
- [179] Geir Evensen. *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media, 2009.
- [180] Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data assimilation: methods, algorithms, and applications*. SIAM, 2016.

- [181] Peter L Houtekamer and Herschel L Mitchell. A sequential Ensemble Kalman Filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1):123–137, 2001.
- [182] Jeffrey S Whitaker and Thomas M Hamill. Ensemble data assimilation without perturbed observations. *Monthly Weather Review*, 130(7):1913–1924, 2002.
- [183] Elizabeth Million. The hadamard product. *Course Notes*, 3:6, 2007.
- [184] Jssai Schur. Bemerkungen zur theorie der beschränkten bilinearformen mit unendlich vielen veränderlichen. *Journal für die reine und Angewandte Mathematik*, 140:1–28, 1911.
- [185] Andrew C Lorenc. The potential of the Ensemble Kalman Filter for nwp—a comparison with 4d-var. *Quarterly Journal of the Royal Meteorological Society*, 129(595):3183–3203, 2003.
- [186] Razvan Stefanescu, Azam Moosavi, and Adrian Sandu. Parametric domain decomposition for accurate reduced order models: Applications of mp-lrom methodology. *Journal of Computational and Applied Mathematics*, 2017.
- [187] Jin-Long Wu, Jian-Xun Wang, Heng Xiao, and Julia Ling. Physics-informed machine learning for predictive turbulence modeling: A priori assessment of prediction confidence. *arXiv preprint arXiv:1607.04563*, 2016.
- [188] Ahmed Attia, Azam Moosavi, and Adrian Sandu. Cluster sampling filters for non-Gaussian data assimilation. *arXiv preprint arXiv:1607.03592*, 2016.
- [189] Ahmed Attia and Adrian Sandu. A hybrid Monte Carlo sampling filter for non-Gaussian data assimilation. *AIMS Geosciences*, 1(geosci-01-00041):41–78, 2015.
- [190] David A Freedman. *Statistical models: theory and practice*. cambridge university press, 2009.
- [191] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.
- [192] David W Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *J. Artif. Intell. Res.(JAIR)*, 11:169–198, 1999.
- [193] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.
- [194] Peter Bühlmann and Torsten Hothorn. Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, pages 477–505, 2007.

- [195] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [196] Leo Breiman et al. Arcing classifier (with discussion and a rejoinder by the author). *The annals of statistics*, 26(3):801–849, 1998.
- [197] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.
- [198] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.
- [199] Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. Conditional variable importance for random forests. *BMC bioinformatics*, 9(1):307, 2008.
- [200] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.
- [201] Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.
- [202] Llew Mason, Jonathan Baxter, Peter L Bartlett, and Marcus R Frean. Boosting algorithms as gradient descent. In *Advances in neural information processing systems*, pages 512–518, 2000.
- [203] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.
- [204] Cheng Li. A gentle introduction to gradient boosting. URL: http://www.ccs.neu.edu/home/vip/teach/MLcourse/4_boosting/slides/gradient_boosting.pdf, 2014.
- [205] Jeffrey L Anderson. A method for producing and evaluating probabilistic forecasts from ensemble model integrations. *Journal of Climate*, 9(7):1518–1530, 1996.
- [206] Guillem Candille and Olivier Talagrand. Evaluation of probabilistic prediction systems for a scalar variable. *Quarterly Journal of the Royal Meteorological Society*, 131(609):2131–2150, 2005.
- [207] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
- [208] Pavel Sakov and Peter R Oke. A deterministic formulation of the ensemble Kalman filter: an alternative to ensemble square root filters. *Tellus A*, 60(2):361–371, 2008.

- [209] Ahmed Attia and Adrian Sandu. Dates: A highly-extensible data assimilation testing suite. *arXiv preprint arXiv:1704.05594*, 2017.
- [210] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [211] Azam Moosavi and Adrian Sandu. A state-space approach to analyze structural uncertainty in physical models. *Metrologia*, 2017.
- [212] Yannick Trémolet. Model-error estimation in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 133(626):1267–1280, 2007.
- [213] S Akella and I.M. Navon. Different approaches to model error formulation in 4D-Var: a study with high-resolution advection schemes. *Tellus A*, 61(1):112–128, 2009.
- [214] François-Xavier Le Dimet and Olivier Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A: Dynamic Meteorology and Oceanography*, 38(2):97–110, 1986.
- [215] I Michael Navon, Xiaolei Zou, J Derber, and J Sela. Variational data assimilation with an adiabatic version of the nmc spectral model. *Monthly weather review*, 120(7):1433–1446, 1992.
- [216] Weather research forecast model. <https://www.mmm.ucar.edu/weather-research-and-forecasting-model>.
- [217] Robert G Fovell. Impact of microphysics on hurricane track and intensity forecasts. In *Preprints, 7th WRF Users Workshop, NCAR*, 2006.
- [218] Nasrin Nasrollahi, Amir AghaKouchak, Jialun Li, Xiaogang Gao, Kuolin Hsu, and Soroosh Sorooshian. Assessing the impacts of different WRF precipitation physics in hurricane simulations. *Weather and Forecasting*, 27(4):1003–1016, 2012.
- [219] Robert G Fovell. Influence of cloud-radiative feedback on tropical cyclone motion. In *29th Conference on Hurricanes and Tropical Meteorology*, 2010.
- [220] Marla R Knebl Lowrey and Zong-Liang Yang. Assessing the capability of a regional-scale weather model to simulate extreme precipitation patterns and flooding in central Texas. *Weather and Forecasting*, 23(6):1102–1126, 2008.

- [221] WA Gallus Jr. Eta simulations of three extreme rainfall events: Impact of resolution and choice of convective scheme. *Wea. Forecasting*, 14:405–426, 1999.
- [222] Wei Wang and Nelson L Seaman. A comparison study of convective parameterization schemes in a mesoscale model. *Monthly Weather Review*, 125(2):252–278, 1997.
- [223] Isidora Jankov, Paul J Schultz, Christopher J Anderson, and Steven E Koch. The impact of different physical parameterizations and their interactions on cold season QPF in the American River basin. *Journal of Hydrometeorology*, 8(5):1141–1151, 2007.
- [224] Robin C Gilbert, Michael B Richman, Theodore B Trafalis, and Lance M Leslie. Machine learning methods for data assimilation. *Computational Intelligence in Architecturing Complex Engineering Systems*, pages 105–112, 2010.
- [225] VM Krasnopolsky, Michael Fox-Rabinovitz, Alexei Belochitski, Philip J Rasch, Peter Blossey, and Yefim Kogan. *Development of neural network convection parameterizations for climate and NWP models using Cloud Resolving Model simulations*. US Department of Commerce, National Oceanic and Atmospheric Administration, National Weather Service, National Centers for Environmental Prediction, 2011.
- [226] Azam Moosavi, Ahmed Attia, and Adrian Sandu. A machine learning approach to adaptive covariance localization. *arXiv preprint arXiv:1801.00548*, 2018.
- [227] J. Glimm, S. Hou, Y.H. Lee, D.H. Sharp, and K. Ye. Sources of uncertainty and error in the simulation of flow in porous media. *Computational & Applied Mathematics*, 23:109–120, 2004.
- [228] D. Orrell, L. Smith, J. Barkmeijer, and T.N. Palmer. Model error in weather forecasting. *Nonlinear Processes in Geophysics*, 8:357–371, 2001.
- [229] T.N. Palmer, G.J. Shutts, R. Hagedorn, F.J. Doblas-Reyes, T. Jung, and M. Leutbecher. Representing model uncertainty in weather and climate prediction. *Annu. Rev. Earth Planet. Sci.*, 33:163–93, 2005.
- [230] Carla Cardinali, Nedjeljka vZagar, Gabor Radnoti, and Roberto Buizza. Representing model error in ensemble data assimilation. *Nonlinear Processes in Geophysics*, 21(5):971–985, 2014.
- [231] James A Hansen. Accounting for model error in ensemble-based state estimation and forecasting. *Monthly Weather Review*, 130(10):2373–2391, 2002.

- [232] Vishwas Rao and Adrian Sandu. A posteriori error estimates for the solution of variational inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):737–761, 2015.
- [233] Yannick Tr'emolet. Accounting for an imperfect model in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 132(621):2483–2504, 2006.
- [234] Dusanka Zupanski and Milija Zupanski. Model error estimation employing an ensemble data assimilation approach. *Monthly Weather Review*, 134(5):1337–1354, 2006.
- [235] Elham Asgari and Kaveh Bastani. The utility of Hierarchical Dirichlet Process for relationship detection of latent constructs. In *Academy of Management Proceedings*, 2017.
- [236] Lidia Ceriani and Paolo Verme. The origins of the gini index: extracts from *variabilita e mutabilita* (1912) by corrado gini. *The Journal of Economic Inequality*, 10(3):421–443, 2012.
- [237] Mark R Segal. Machine learning benchmarks and random forest regression. *Center for Bioinformatics & Molecular Biostatistics*, 2004.
- [238] Jiali Wang and Veerabhadra R Kotamarthi. Downscaling with a nested regional climate model in near-surface fields over the contiguous united states. *Journal of Geophysical Research: Atmospheres*, 119(14):8778–8797, 2014.
- [239] National Oceanic and Atmospheric Administration (NOAA). <https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forecast-system-gfs>.
- [240] WRF Model Physics Options and References. http://www2.mmm.ucar.edu/wrf/users/phys_references.html.
- [241] E Kessler. On the continuity and distribution of water substance in atmospheric circulations. *Atmospheric research*, 38(1-4):109–145, 1995.
- [242] Yuh-Lang Lin, Richard D Farley, and Harold D Orville. Bulk parameterization of the snow field in a cloud model. *Journal of Climate and Applied Meteorology*, 22(6):1065–1092, 1983.
- [243] Song-You Hong, Jimy Dudhia, and Shu-Hua Chen. A revised approach to ice microphysical processes for the bulk parameterization of clouds and precipitation. *Monthly Weather Review*, 132(1):103–120, 2004.

- [244] Song-You Hong and Jeong-Ock Jade Lim. The WRF single-moment 6-class microphysics scheme (wsm6). *J. Korean Meteor. Soc*, 42(2):129–151, 2006.
- [245] Wei-Kuo Tao, Joanne Simpson, and Michael McCumber. An ice-water saturation adjustment. *Monthly Weather Review*, 117(1):231–235, 1989.
- [246] Gregory Thompson, Paul R Field, Roy M Rasmussen, and William D Hall. Explicit forecasts of winter precipitation using an improved bulk microphysics scheme. part ii: Implementation of a new snow parameterization. *Monthly Weather Review*, 136(12):5095–5115, 2008.
- [247] Hugh Morrison, Gregory Thompson, and V Tatarskii. Impact of cloud microphysics on the development of trailing stratiform precipitation in a simulated squall line: Comparison of one-and two-moment schemes. *Monthly Weather Review*, 137(3):991–1007, 2009.
- [248] John S Kain. The Kain–Fritsch convective parameterization: an update. *Journal of Applied Meteorology*, 43(1):170–181, 2004.
- [249] Zavisva Janjic. The step-mountain eta coordinate model: Further developments of the convection, viscous sublayer, and turbulence closure schemes. *Monthly Weather Review*, 122(5):927–945, 1994.
- [250] Georg A Grell and Saulo R Freitas. A scale and aerosol aware stochastic convective parameterization for weather and air quality modeling. *Atmospheric Chemistry & Physics Discussions*, 13(9), 2013.
- [251] Eli J Mlawer, Steven J Taubman, Patrick D Brown, Michael J Iacono, and Shepard A Clough. Radiative transfer for inhomogeneous atmospheres: Rrtm, a validated correlated-k model for the longwave. *Journal of Geophysical Research: Atmospheres*, 102(D14):16663–16682, 1997.
- [252] Andrew J Conley, Rolando Garcia, Doug Kinnison, Jean-Francois Lamarque, Dan Marsh, Mike Mills, Anne K Smith, Simone Tilmes, Francis Vitt, Hugh Morrison, et al. Description of the near community atmosphere model (cam 5.0). *NCAR technical note*, 2012.
- [253] Jimy Dudhia. Numerical study of convection observed during the winter monsoon experiment using a mesoscale two-dimensional model. *Journal of the Atmospheric Sciences*, 46(20):3077–3107, 1989.

- [254] Ming-Dah Chou and Max J Suarez. A solar radiation parameterization (clrad-sw) for atmospheric studies. *NASA Tech. Memo*, 10460:48, 1999.