Towards Real-Time CFD Simulation of In-Flight Icing via Reduced-Order Modeling

Zhao Zhan

Department of Mechanical Engineering

McGill University

Montreal, Québec

August, 2016

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Doctor of Philosophy

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Professor Wagdi G. Habashi, for giving me the opportunity to work on this interesting and challenging research topic at McGill University's CFD Laboratory, and for mentoring and supporting me throughout the course of my research. I would also like to thank my co-supervisor, Professor Marco Fossati, who helped me to advance my research via valuable advice and in-depth discussions.

I would like to express my gratitude to our system administrator, Jesse Stacey, for solving all software/hardware related issues in a timely manner. I also express my gratitude to the staff of Newmerical Technologies International (now ANSYS-Montreal) for providing suggestions in using their software package FENSAP-ICE. In particular, thanks to Jian Chen for his help in mesh generation for the complex regional jet geometry. Additionally, I appreciate the helpful discussions with Dr. Isik Ozcer.

I also wish to acknowledge the Natural Sciences and Engineering Research Council of Canada for funding through an Industrial Research Chair. Moreover, I am grateful to Compute Canada and Calcul Québec for the use of their supercomputer resources, and for the excellent technical support they have provided.

Additionally, I am thankful to my colleagues in CFD Lab for creating a positive and interactive research environment. A special thanks goes to David Bilodeau for assisting in reviewing the thesis.

Lastly, I give my heartfelt gratitude to the most important people in my life, my parents and my husband. Their love, encouragement and support are the greatest motivation for my pursuit of excellence.

TABLE OF CONTENTS

NOMENCLATURE

Abbreviations

AoA	Angle of Attack
BFGS	Broyden, Fletcher, Goldfarb and Shanno method
CFD	Computational fluid dynamics
CM	Continuous maximum
CPU	Central processing unit
CVT	Centroidal Voronoi tessellation
DoE	Design of experiments
EFD	Experimental fluid dynamics
FAA	Federal Aviation Administration
FFD	Flight fluid dynamics
GA	Genetic algorithm
GPU	Graphics processing unit
IAS	Indicated air speed
IM	Intermittent maximum
LHS	Latin hypercube sampling
LOOCV	Leave-one-out cross-validation
LR	Logistic regression
LWC	Liquid water content
MPI	Message Passing Interface
MVD	Median volumetric diameter
NASA	National Aeronautics and Space Administration

N-S	Navier-Stokes
NTSB	National Transportation Safety Board
PA	Pressure altitude
PDE	Partial differential equations
POD	Proper orthogonal decomposition
QN	Quasi-Newton
RBF	Radial basis function
RJ	Regional jet
ROB	Reduced-order basis
ROM	Reduced-order modeling
TAS	True air speed
TRL	Technology Readiness Levels

Symbols

N_D	Dimension of design space (parameter space)
N_S	Number of samples (snapshots)
N_P	Number of data points contained in a snapshot

Design of experiments

$\Omega \subseteq \mathbb{R}^{N_D}$	Open set
$\mathbf{z} \in \Omega$	Generator
$\hat{V} \in \Omega$	Voronoi region
$\rho(\cdot)$	Density function
$\mathbf{z}^* \in \Omega$	Mass centroid of Voronoi region $\hat{V}$
$W \in \mathbb{R}^{N_D}$	Discrete set of points

$\mathcal{F}(\cdot)$	Energy/cost function
μ	Mean of Gaussian distribution
σ^2	Variance of Gaussian distribution
λ	Rate parameter of exponential distribution

Proper orthogonal decomposition

$\boldsymbol{U} \in \mathbb{R}^{N_P}$	Observation/snapshot
$\overline{\boldsymbol{U}} \in \mathbb{R}^{N_P}$	Arithmetic mean
$\widetilde{\boldsymbol{U}} \in \mathbb{R}^{N_P}$	Snapshot deviation from mean
$\boldsymbol{\varphi} \in \mathbb{R}^{N_P}$	POD basis
$\mathbf{R} \in \mathbb{R}^{N_S \times N_S}$	Correlation matrix between snapshots
$\boldsymbol{\beta} \in \mathbb{R}^{N_S}$	Eigenvector of correlation matrix **R**
λ	Eigenvalue of correlation matrix **R**
E	Energy content
M	Number of modes from truncation
α	Mode coefficient
$\mathbf{x}^\delta \in \mathbb{R}^{N_D}$	Untried input condition
α^δ	Mode coefficient at untried input condition

Kriging

$\mathbf{x} \in \mathbb{R}^{N_D}$	Input parameter defining observation condition
y	Observed function value
$\boldsymbol{y} \in \mathbb{R}^{N_S}$	Vector of observed function values
Y	Random variable
$\boldsymbol{Y} \in \mathbb{R}^{N_S}$	Vector of random variables

$\mathbf{R} \in \mathbb{R}^{N_S \times N_S}$	Correlation matrix between random variables
σ^2	Covariance of random variable Y
μ	Mean of random variable Y
$\hat{\sigma}^2$	Estimated covariance
$\hat{\mu}$	Estimated mean
$\boldsymbol{\theta} \in \mathbb{R}^{N_D}$	Weight parameter
$L(\cdot)$	Concentrated log-likelihood function
s^2	Mean square error of Kriging predictor

Machine learning

$J(\cdot)$	Objective function / distortion measure
r	Binary indicator
$\boldsymbol{\mu} \in \mathbb{R}^{N_P}$	Cluster center
S	Training set
t	Label of the class
$\phi(\cdot)$	Basis function
N_F	Number of features
G	Polynomial degree
$\mathcal{C}$	Class
p	Posterior probability
y	Predicted posterior probability
$\sigma(\cdot)$	Logistic sigmoid function
$\boldsymbol{w} \in \mathbb{R}^{N_F}$	Model parameter
$J(\cdot)$	Penalized cost function

λ	Regularization coefficient

Unconstrained minimization

$f(\cdot)$	Objective function
$\boldsymbol{\theta}_0 \in \mathbb{R}^{N_D}$	Initial guess
$\boldsymbol{p} \in \mathbb{R}^{N_D}$	Search direction
$\nabla^2 f$	Hessian matrix
$\boldsymbol{B} \in \mathbb{R}^{N_D \times N_D}$	Approximation to Hessian matrix
α	Step length
$\phi(\cdot)$	Step length function

Others

C_L	Lift coefficient
C_D	Drag coefficient
C_M	Pitching moment coefficient
ε_{mass}	Error of mass of ice
ε_{C_L}	Error of $C_L - \alpha$ curve
$\lVert\cdot\rVert$	Euclidean norm
$\lvert\cdot\rvert$	Absolute value
L_∞	Infinity norm

Units

ft	Foot
kt	Knot (1 knot = 1 nautical mile per hour)

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Computational fluid dynamics (CFD) is playing a rapidly growing role in the aero-icing certification process. However, three-dimensional (3D) viscous turbulent CFD-icing simulations of complete aircraft are prohibitive, in particular parametric studies where the cost of repeated calculations could become overwhelming. A reduced-order modeling (ROM) approach is proposed to explore the complete icing envelopes, with comparable accuracy to CFD.

The ROM approach uses a limited, but strategically selected, number of snapshots (solutions obtained at various operating conditions) of the complex icing problem, and extracts a basis of vectors (or modes) that represent its most fundamental physical features. Proper orthogonal decomposition (POD) is adopted to extract modes from the snapshots. A linear combination of the POD modes can be subsequently used to obtain solutions for conditions different than the snapshots, with coefficients obtained via multi-dimensional interpolation. In regard to the continuous and intermittent maximum icing envelopes, a local POD approach, using machine learning algorithms, is developed. It clusters similar snapshots and delimits ice-type regions within the envelope, such that distinct physical features can be extracted separately. An error-driven iterative sampling method combining a greedy approach and a centroidal Voronoi tessellation sampling technique is developed to position additional snapshots in the regions of high nonlinearity, finding a good balance between accuracy and the total number of snapshots.

The proposed ROM framework and iterative sampling methodology is first assessed on an airfoil. It is demonstrated that the continuous and intermittent maximum icing envelopes can be explored via ROM in terms of both shape/mass of ice and the associated lift coefficient curves. The methodology is then applied, for the first time to the best knowledge of the author, to a

detailed aero-icing study of a regional jet, in terms of: 1) the “complete” exploration of the continuous maximum icing conditions for the shape/mass of ice, and 2) its aerodynamic degradation (changes in lift, drag and pitching moment) due to ice contamination during holding around an airport, and the consequent effect during descent and aborted landing. The results strongly support the drive to incorporate more CFD information into in-flight icing certification procedures as well as pilot training programs, leading to increased aviation safety.

RÉSUMÉ

La mécanique des fluides numérique (CFD) joue un rôle de plus en plus important lors du processus de certification du givrage en vol. Toutefois, la prédiction du cumul de glace sur un aéronef entier est trop coûteuse, particulièrement lors d'analyses paramétriques pour lesquelles le coût de calculs répétitifs peut devenir prohibitif. Un modèle d'ordre réduit (MOR; ROM en anglais) est proposé pour balayer à coût modeste la plage entière de l'enveloppe de certification, tout en offrant une précision comparable à la CFD.

La méthode MOR utilise un nombre limité mais stratégiquement sélectionné de solutions, dites prises ou snapshots, obtenues à différentes conditions opérationnelles du problème et en extrait une base de valeurs propres (ou modes) représentant les caractéristiques physiques fondamentales du système. Une décomposition orthogonale propre (POD) est adoptée pour extraire les modes de ces prises. Une combinaison linéaire des modes est par la suite utilisée pour obtenir des solutions à des conditions autres que celles des prises, par interpolation multidimensionnelle. En ce qui concerne les domaines de givrage maximaux continus et intermittents, une décomposition orthogonale propre locale, utilisant des algorithmes d'apprentissage automatique, est développée. Cette méthodologie permet la partition de prises similaires et la délimitation de différentes régions selon le type de glace, de telle sorte que leurs caractéristiques physiques distinctes peuvent être traitées séparément. Une méthode d'échantillonnage itératif combinant un algorithme glouton (dit greedy en anglais), et une technique de partition d'échantillonnage de type centroïde de Voronoï, sont développées pour finement positionner des prises additionnelles dans les régions de forte non-linéarité, tout en maintenant un juste équilibre entre la précision et le nombre total de prises.

L’approche ROM et la méthodologie d'échantillonnage itératif sont d’abord évaluées pour un profil aérodynamique, et il est démontré que les enveloppes de givrage maximum-continu et intermittent peuvent être explorées via ROM pour prédire la forme et la masse de glace, ainsi que la portance. La méthode est par la suite appliquée, pour la première fois au meilleur des connaissances de l'auteur, à l’étude détaillée de l’aéro-givrage d'un jet régional, en termes de: 1) l'exploration "complète" des conditions de givrage maximales continues pour la forme/masse de glace et 2) la dégradation aérodynamique (changement de portance, de traînée et de moment de tangage) en raison de l’accumulation de glace lors de survol prolongé d’un aéroport, de descente et d'atterrissage avorté. Le niveau de précision atteint démontre la pertinence d’incorporer plus d'informations CFD dans les programmes de certification pour le givrage en vol, ainsi que lors de la formation de pilotes, avec pour but d’améliorer la sécurité aérienne.

CHAPTER 1 INTRODUCTION

1.1 Motivation and objective

In-flight icing poses substantial risks to aviation safety [1]. As a consequence of accretion of ice on wings and/or other critical surfaces, lift decreases, drag increases, and the center of gravity shifts. These adverse effects will degrade the aerodynamic performance and controllability of the airplane, resulting in incidents and accidents.

From 1996 to 2008, the National Transportation Safety Board (NTSB) had issued 82 icing-related recommendations to the Federal Aviation Administration (FAA), based on its aviation accident investigations [2]. For instance, following the 1997 fatal crash of Comair Airlines Flight 3272 near Monroe, Michigan, the NTSB called for FAA to:

> *"(A-98-92) Conduct research to identify realistic ice accumulations and determine the effects and dangers of such ice accumulations. The information developed through such research should be incorporated into aircraft certification requirements and pilot training programs."*

More recently, following the 2009 crash of Empire Airlines Flight 8284 in Lubbock, Texas, the NTSB made recommendations that put more emphasis on pilot training [3]:

> *"(A-11-46) Define and codify minimum simulator model fidelity requirements for aerodynamic degradations resulting from airframe ice accumulation. These requirements should be consistent with performance degradations that the National Transportation Safety Board and other agencies have extracted during the investigations of icing accidents and incidents."*

Among NTSB's icing-related recommendations to the FAA, the following are recognized as being key factors and challenges in reducing risks posed to aviation safety by icing, namely:

- rigorous aircraft in-flight icing certification;
- realistic and thorough training for pilots to recognize and deal with degraded flight characteristics due to airframe icing.

1.1.1 Icing certification

Icing effects on aircraft are assessed through an extensive procedure, intended to ensure safe operation for conditions specified in the icing envelopes described in the Appendix C of the Code of Federal Regulations - Title 14 (Aeronautics and Space) - Part 25 (Airworthiness Standards: Transport Category Airplanes) [4]. Two types of icing envelopes have to be considered in the certification process: continuous maximum (CM) representing stratus-type clouds and intermittent maximum (IM) representing cumulus-type clouds (Figure 1-1). These icing envelopes specify atmospheric icing conditions in terms of pressure altitude, temperature,

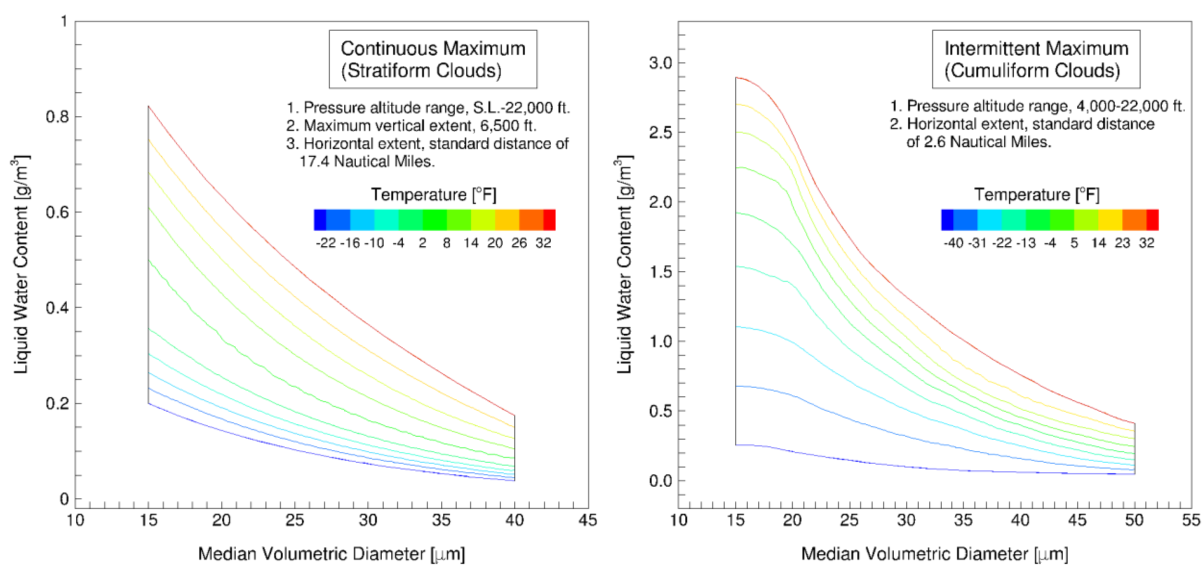


Figure 1-1: Appendix C envelopes: continuous maximum (left); intermittent maximum (right).

liquid water content (LWC) and water droplet median volumetric diameter (MVD). Most recently, new appendices have been created to take into account icing conditions containing freezing drizzle and freezing rain. These are Appendix O (Supercooled large drop icing conditions) added to Part 25, and Appendix D (Mixed phase and ice crystal icing conditions) added to Part 33 (Airworthiness Standards: Aircraft Engines) [5].

The traditional icing certification process includes numerical (computational fluid dynamics, CFD) and wind/icing tunnel simulations (experimental fluid dynamics, EFD), flight behind an icing tanker, and, ultimately, flight into natural icing conditions (flight fluid dynamics, FFD). With the exponential increase in computer power and the accompanying sophistication of numerical technologies, CFD has been playing a rapidly growing, shortly to become primary, role in the aero-icing certification process. Given the large number of icing conditions to be tested, the use of wind/icing tunnels is slowly waning, as the full geometry (whole airplane, with engines or propellers running) cannot be tested, nor the altitude conditions replicated. Natural flight tests are the ultimate step to be completed before obtaining certification [6]. However, not all icing conditions can be easily found in nature within a limited number of flights. Similarly, 3D viscous turbulent CFD-icing simulations of complete or even partial aircraft geometries are considered expensive and even prohibitive for parametric studies where the cost of repeated calculations could become overwhelming.

To alleviate the computational burden, a reduced-order modeling (ROM) approach based on proper orthogonal decomposition (POD) and multidimensional interpolation is developed within this thesis, dramatically reducing the computational complexity of covering the entire icing envelopes with comparable accuracy to full-fledged CFD. Meanwhile, the proposed method should also be able to incorporate seamlessly EFD and FFD data when and if available.

1.1.2 Icing effects in flight simulators

The current implementation of icing effects in commercial flight training simulators is rather primitive [7]. The reason for this is because flight simulation models are based on flight test data supplied by manufacturers. Since only limited aircraft performance data in icing conditions is available, and data from accidents can be accounted for only a posteriori, icing effects in flight simulators are represented by a combination of increased weight, a displacement of the center of gravity and engine vibration. This rather primitive simulation may give pilots false impressions that are far from the realities of an actual icing encounter [3]. It goes without saying that if icing effects were to be represented with more fidelity in flight simulators, pilots could be better trained to recognize and recover, within the few seconds available to them, from the degraded handling qualities of an iced aircraft [8]. This ought to have a considerable beneficial impact on aviation safety.

In order to introduce icing effects into flight training simulators, National Aeronautics and Space Administration (NASA) Glenn's Icing Branch launched a program in 1998 [7]. The airplane chosen for this activity was a De Havilland DHC-6 Twin Otter, which has been historically employed by NASA as an icing research aircraft. Using the combination of wind tunnel and flight test data, an Ice Contamination Effects Flight Training Device was developed [9]. Their research demonstrated that icing effects could be modeled accurately in flight training devices to show how they adversely affect airworthiness. However, as the aircraft type and icing condition changes, this wind tunnel experiment and flight test based method could become prohibitively expensive in practical applications.

In this thesis, a CFD-based icing effects flight simulator is proposed, which would be aircraft-specific. CFD-icing tools are capable of providing more detailed, reliable and repeatable information about the ice accretion and the associated "deltas" in aerodynamic degradation, for lift, drag and moments. However, generating a large amount of CFD data corresponding to a wide range of pilot inputs, e.g., pitch angle, roll angle, indicated air speed, flight altitude, etc., requires a tremendous amount of computational resources, making it extremely difficult. Reduced-order modeling is therefore proposed, which is capable of providing solutions of as great accuracy and as much detail as the full 3D Navier-Stokes (N-S) equations. These ROM solutions are available at a computational cost several orders of magnitude smaller than the N-S solutions, and ultimately, may provide real-time representations of in-flight icing for pilot training.

1.1.3 Thesis objective

The objective of this thesis is to develop an optimized ROM framework for the parametric analysis of in-flight icing problems, with specific applications in icing certification and icing effects flight simulators. While considerable developments in ROM have already been made at the McGill CFD Laboratory [10-12], research work still needs to be done to make this approach usable at the level required by aircraft manufacturers and engineers for these types of applications. Namely, a method characterized by a relatively high grade (>5) in the Technology Readiness Levels (TRL) [13].

For icing certification application, the focus is on the exploration of all icing conditions enclosed by the icing envelopes. Based on a limited, but strategically selected, number of CFD solutions (defined at different icing conditions), the ROM approach is capable to predict the outputs of interest (e.g. shape of ice, pressure distribution, shear stress distribution) at "untried" input conditions. However, distinct physical features exhibited in the icing envelope pose substantial challenges for ROM to make accurate predictions. Therefore, the method of local reduced-order modeling has been developed during this thesis, which deals with distinct physical features separately, leading to more accurate ROM performances over the entire icing envelopes.

For flight simulators, data is needed in real-time, making the use of anything but lookup tables, impossible. Therefore, the goal is to advance the ROM framework to achieve real-time performance (around or below 1/15th second), such that it can be incorporated into a flight simulator to generate aerodynamic data "on-the-fly". In case of near-real-time performance, ROM can be used to fill up the lookup table in an accurate and efficient way. Nevertheless, realistic in-flight icing training can be realized by introducing reliable performance degradation data obtained from CFD and ROM, which will be beneficial for the flight simulation industry.

In summary, this thesis aims to:

1. Develop an automatic approach to address the problem of optimal identification of snapshots;
2. Develop local reduced-order modeling technique, to deal with distinct physical features exhibited in the icing envelopes;
3. Advance the ROM methodologies/algorithms towards real-time or near-real-time performance;
4. Assess the proposed methodologies using representative 2D and 3D geometries.

1.2 Literature review

1.2.1 Reduced-order modeling

Many engineering problems can be modeled by partial differential equations (PDEs), such as the N-S equations and Euler equations in fluid mechanics. Solving these PDEs typically requires discretization of the physical domain, e.g., finite element method, finite volume method and finite difference method, which result in very high dimensional (in terms of degrees of freedom) models. In disciplines requiring repeated model evaluations over a large range of system inputs, such as design optimization, optimal control and parameter variation studies, reduced-order modeling (ROM), or model order reduction, has become a very active research topic to alleviate the computational burden.

Reduced-order models are defined as opposed to high dimensional (or high fidelity) models. Unlike the low-fidelity methods based on reduced dimensions (2D or quasi-3D approximations) and/or reduced physics (empirical correlations, inviscid flow, incompressible flow), in the present context, the ROM approach makes low order approximation to the high fidelity models, at a much lower computational cost (usually several orders of magnitude smaller), while maintaining the spatial dimensions and physics of the problem. A common feature shared by reduced-order methods is that they use a number of snapshots (solutions of the high fidelity model at different input conditions) of the complex system to extract a basis of vectors (or modes), which span a low-order subspace where the solution of the system can be represented as a linear combination of the basis vectors. For time-dependent systems, snapshots are taken at discrete time instants; while for parametric systems, the reduced-order basis (ROB) can be selected to cover solution variety over variation of parameters. Different approaches can be used to define the basis, e.g. Lagrangian reduced-basis [14], centroidal Voronoi tessellations based approaches [15], and proper orthogonal decomposition (POD) [16, 17]. In this thesis, POD is adopted, given that it not only provides the optimal linear representation of the dominant features/physics, but also allows for the truncation of the linear combination of modes for a given level of accuracy. POD-based ROM is widely applied in many engineering fields such as in aeroelasticity [18-22], optimal flow control [14, 23], optimal design [24], inverse design [25, 26], mesh adaptation [27], parametric studies [28, 29] and aero-icing analysis [12, 30-34].

Once the basis vectors are determined, the scalar coefficients in the linear combination of modes can be obtained by adopting one of the two following approaches.

The projection-based ROM, which can be generally categorized as an “intrusive” approach, projects the governing PDEs onto the subspace spanned by the ROB to yield a smaller set of nonlinear equation (ODEs), and then solves these ODEs for the coefficients of the linear combination of POD modes [35-39]. A major drawback of the projection-based ROM is that it requires direct operation on the governing equations (hence the code) of the problem, which may lack numerical stability. More importantly, for applications with mixed-type of snapshots, such as in icing certification, when the numerical snapshots (CFD data) are mixed with ones obtained from flight (FFD) and/or experimental tests (EFD), this projection-based ROM is no more applicable.

The interpolation-based ROM, which is categorized as a “non-intrusive” approach, uses response surface methods to get the coefficients. For each POD mode, a response surface is formed by projecting the high fidelity solutions (snapshots) onto this specific mode. Based on the assumption that the response surface formed by the projection coefficients is smooth, for any “untried” input condition, the unknown mode coefficient can be obtained via multi-dimensional interpolation. This method does not involve any intrusive modifications to the governing equations or code, making it more robust. It can also work with any combination of snapshots coming from CFD, EFD and/or FFD. From a computational point of view, defining a response surface for each coefficient of the linear expansion is much more effective than solving a system of ODEs as required by the projection-based approach. Therefore, with this approach, obtaining real-time computing is more straightforward. The response surface for the mode coefficients can be obtained by polynomial interpolation [23, 25], Akima or Kriging interpolations [12, 30, 31], radial basis functions (RBFs) [24, 29], or by Smolyak sparse grid interpolation method [40]. In this thesis, interpolation-based ROM is adopted, with more discussion given in the following section.

1.2.2 Interpolation-based ROM

The interpolation-based ROM was first introduced by Ly and Tran in [23], for the modeling and control of steady-state Rayleigh-Bénard convection problems. In that setting, the input parameter is the Rayleigh number, and the snapshots are CFD solutions containing the temperature distributions. The boundary control problem was formulated as finding the optimum Rayleigh number to minimize a cost functional. It was shown that POD can be used to model the natural convection and the approach is very efficient and is suited for process control.

In the work of Bui-Thanh et al. [25], POD with cubic spline interpolation was applied to parametric variation study of steady subsonic flow about an airfoil with varying angle of attack (AoA) and Mach number. The same methodology was used by Yapalparvi et al. [41] to predict the unsteady flow fields and trajectories of tumbling plates. In this application, time is considered as one of the two input parameters. Mifsud et al. evaluated spline interpolation methods and RBFs in [29], where parametric studies were performed for two-parameter inviscid steady flow about a flare stabilized projectile, and a two-parameter supersonic turbulent flow around a fin-stabilized projectile with drooping nose control. In the work of Audouze et al. [24], POD with RBFs were applied to 1D and 2D multi-parameter, nonlinear, steady state convection-reaction-diffusion problems. Their work also employs a greedy residual search in the parameter space to incrementally update the reduced-order model. More recently, Xiao et al. [40] introduced Smolyak sparse grid method for calculating the POD coefficients, and applied this non-intrusive ROM to two unsteady flow problems: 2D flow past a cylinder and flow within a gyre. In another work of Xiao et al. [22], POD with RBFs was proposed for fluid-structure interaction applications.

In the field of aero-icing studies, ROM was first introduced by Nakakita et al. in [30]. In that work, POD with Akima interpolation was applied to the prediction of ice shapes over an airfoil under a one-variable (freestream temperature) problem, as well as ice shapes over a DLR-F6 geometry with two parameters (freestream temperature and AoA). Later on, POD with Kriging was introduced by Lappo and Habashi [31], and applied to the prediction of ice shapes accreted on an airfoil with four input parameters, namely AoA, LWC, free stream temperature and MVD. In the work of Jung et al. [32], POD with piecewise linear interpolation was adopted in a two-parameter test case (AoA and MVD) to predict the droplet collection efficiency and ice shapes for an airfoil.

It is worth noting that in all the above-mentioned interpolation-based ROM methods, the accuracy of ROM is evaluated in a rather "spotty" approach. Namely, by selecting some unsampled locations in the parameter space (or time instants for time-dependent problems), the associated CFD solutions are computed as references, then the accuracy of ROM is evaluated by comparing the ROM solution with the corresponding CFD solution [22, 23, 25, 29-32, 40, 41], or using the residual returned by the CFD solver, taking the ROM solution as a candidate guess solution [24]. Errors obtained in this type of approach only provide spotty information regarding the accuracy of ROM in the parameter space (or time span), since for the majority of the unsampled design space, the accuracy of ROM remains unknown.

A more systematic way to evaluate ROM accuracy in the entire design space was proposed by Fossati and Habashi in [12], where the method of leave-one-out cross-validation (LOOCV) was adopted. This approach gives a good estimation of the performance of ROM over the entire design space. Based on this estimation, an error driven iterative sampling method is proposed in this thesis, which is capable of adaptively enriching the set of snapshots and improving the accuracy of ROM in a systematic way. Details about LOOCV and iterative sampling will be given in Section 2.4.

1.2.3 Local reduced-order modeling

Given a set of snapshots, global or local POD methods can be used to identify the basis vectors. The global POD approach uses all the available snapshots to generate the basis. This method is straightforward, but for problems with locally distinct physical characteristics, such as glaze and rime ice formations in the case of aero-icing, or subsonic, supersonic and hypersonic regimes in the case of aerodynamic studies, reduced-order solutions obtained via global POD may be affected by very different snapshots features. Local POD, on the other hand, deals with distinct physical characteristics separately. The local approach calls for the subdivision of the solution (and parameter) spaces into subregions, each ideally comprising snapshot characterized by similar or sufficiently close physical features. In the recent literature, *k*-means clustering has often been used for grouping similar snapshots into clusters [42-44].

To build local reduced-order models, three issues have to be solved. Firstly, similar snapshots have to be grouped into clusters such that POD can be applied locally to each cluster, i.e. only to that selected set of snapshots, to generate a set of local ROBs. The desired clustering can be achieved by using an unsupervised learning algorithm known as *k*-means clustering, also adopted in [42-44].

The second problem is identifying the most suitable cluster for the new solution of an untried condition to be represented as a linear combination of the POD basis of only that cluster. This is a "classification problem": given a set of known input parameters and the corresponding clusters obtained from the k-means algorithm, a new input condition has to be assigned to one of the available K discrete classes. This can be addressed by using a supervised learning algorithm. In [44], a nearest neighbor classifier is employed, while logistic regression is adopted in the present thesis.

The third problem is to define the boundaries of the subregions in the parameter space such that each cluster is ideally enclosed. Neighboring clusters must also be contiguous, leaving no void regions in the parameter space. In [42, 43], problems arising at the boundary of each region are solved by adding neighboring snapshots to each cluster to obtain a set of overlapping clusters and ensuring that there are no "gaps" between clusters. The overlap is performed in the solution space, while in the present work the boundary has been defined in the input parameter space. This issue can also be addressed by logistic regression and other classification methods [44]. Each input is assigned to one of the K discrete classes, hence the input space is divided into decision regions whose boundaries are identified as decision boundaries.

1.2.4 Sampling of the design space

A crucial aspect in the success of a ROM approach is to optimally define the snapshots that best extract the physical features of the solution. Defining a suitably rich set of snapshots in the parameter space (also referred to as design space) is a problem of design of experiments (DoE) [45, 46]. Classical approaches have been adopted in the ROM literature, e.g. uniform/grid sampling [23, 25, 47], Latin hypercube sampling (LHS) [24, 29] and LPτ [31]. These sampling methods are straightforward, but do not allow adaptive enrichment of the snapshots set, i.e. introducing new samples in desired locations of the design space without changing the previous distribution, which is a crucial aspect of improving ROM model accuracy.

Recently, a greedy sampling algorithm was introduced to adaptively place new sample points at the location in the parameter space where the maximum error occurs [48-54]. These techniques are developed in the community of projection-based model order reduction, rely on residual-based error bounds or error indicators to assess the accuracy of ROM solutions, and the location of new samples can be determined by a direct search over a set of predefined candidate samples [48, 49, 51], adaptive parameter domain partition [50], or by an optimization algorithm [52-54].

In case formal error indicators are not available or difficult to define, the degree of accuracy of reduced-order solutions can be estimated via leave-one-out cross-validation (LOOCV): an approach compatible with mixed-type snapshots. Sampling techniques based on centroidal Voronoi tessellations (CVT) [55] could be employed in conjunction with LOOCV to identify the location of the snapshots in the parameter space according to a prescribed density function. As the density function for CVT can be based on either errors in the parameter space or a priori knowledge of the physics of the problem, it has the capacity to provide additional sample points judiciously placed in regions of high nonlinearity. In this thesis, an error driven iterative sampling methodology combining LOOCV error and CVT sampling technique is developed [33, 34], with more details given in Section 2.4.

1.3 Thesis contributions

1.3.1 Algorithmic advances

This thesis made fundamental contributions in the area of non-intrusive ROM, leading to faster and more accurate ROM performances.

1. Developed an automatic error-driven iterative sampling methodology to improve the accuracy of ROM in a systematic approach;

2. Developed local POD method based on machine learning algorithms, to efficiently identify and cluster the snapshots in both solution space and parameter space;
3. Advanced Kriging interpolation by adopting gradient-based optimization method, combining with the currently available global optimizer, forming a more cost efficient hybrid optimization method;
4. Parallelized the ROM code using Message Passing Interface (MPI) to achieve considerable speed-up;
5. Optimized the framework of the proposed methodologies to higher grade of TRL.

1.3.2 Engineering contributions

The thesis made significant contributions in the engineering field, particularly in the area of icing certification and flight simulators.

1. A major engineering contribution is the use of ROM to explore the CM and IM icing envelopes, for 2D and 3D geometries. It was demonstrated that ROM can be used to completely cover the gaps in icing certification, rather than the current spotty approach.
2. The aero-icing analysis for a regional jet demonstrated the possibility of incorporating CFD data into flight simulators. In such case, in-flight icing training will no longer be simply based on limited manufacturers and accident data, but can benefit from an aircraft-specific EFD-FFD-CFD database.

1.4 Thesis outline

Chapter 2 describes the mathematical models adopted/advanced in this thesis. Firstly, a brief summary of interpolation-based ROM using POD and Kriging is presented; then, the gradient-based optimization method adopted for Kriging model fitting is introduced; after that, local POD using machine learning algorithms is explained in detail, followed by the introduction of an error driven iterative sampling method and leave-one-out error estimation; a very brief introduction of the CFD-icing tools used in this thesis is also given; and finally, computational cost analysis of the proposed CFD-ROM framework is presented, as well as its parallelization.

Chapter 3 demonstrates the proposed methodologies for the exploration of Appendix C icing envelopes. A 2D geometry (GLC305 airfoil) was selected for the assessment, and the output of interest is the shape of ice accumulated under different icing conditions and the associated lift coefficient curve ($C_L - \alpha$ curve). For the CM, both global and local POD was used and the effectiveness of the two compared. For the IM, local POD was exclusively employed.

Chapter 4 presents the assessment of the proposed methodologies on a regional jet. The first part describes the exploration of Appendix C CM icing envelope for a 25-minute holding pattern. The second part presents aerodynamic degradations resulting from airframe ice accumulation, with detailed parametric analysis over varied flight conditions. Flow details in terms of pressure distribution and shear stress distribution, as well as integrated quantities like lift, drag and pitching moment coefficients (C_L, C_D, C_M) are investigated, for the consequent effect during descent and aborted landing.

Finally, conclusions and future works are given in Chapter 5.

CHAPTER 2 METHODOLOGIES

2.1 Reduced-order modeling via POD and Kriging

The non-intrusive ROM method based on POD and multi-dimensional interpolation developed at the McGill CFD Laboratory [10-12] is adopted in the present work, with only a brief summary given for the purpose of self-containment.

2.1.1 Proper orthogonal decomposition

The POD, also known as Karhunen-Loève expansion, is a procedure for extracting basis functions or modes from a set of snapshots obtained experimentally or from numerical simulations [16]. For an ensemble of N_S observations $\{\boldsymbol{U}_1, \dots, \boldsymbol{U}_{N_S}\}$, where $\boldsymbol{U}_i = \boldsymbol{U}(\mathbf{x}_i) \in \mathbb{R}^{N_P}$, and $\mathbf{x}_i \in \mathbb{R}^{N_D}$ specifies input parameters defining observation conditions. N_P denotes the number of points from which the observation is defined, i.e., in the case of CFD simulation, N_P equals to the number of grid points; in experiment measurements, N_P equals to the number of probes arranged. For CFD-based icing analysis, $\boldsymbol{U}$ may be ice shape represented by the Cartesian coordinates of the N_P surface nodes, while $\mathbf{x}$ defines icing condition in terms of MVD, LWC, etc. For aerodynamic analysis, $\boldsymbol{U}$ can be the solution field of interest, e.g. pressure and shear stress, while $\mathbf{x}$ defines flight conditions like AoA, speed, flight altitude, etc. In the present implementation, the arithmetic mean

$$\overline{\boldsymbol{U}} = \frac{1}{N_S} \sum_{i=1}^{N_S} \boldsymbol{U}_i, \tag{2.1}$$

is subtracted from the snapshots, leaving a modified snapshots set $\mathbf{A} = \{\widetilde{\boldsymbol{U}}_1, \dots, \widetilde{\boldsymbol{U}}_{N_S}\} \in \mathbb{R}^{N_P \mathrm{x} N_S}$, which represents observation deviations from the mean values. POD yields a set of basis vectors

$\boldsymbol{\varphi}_j \in \mathbb{R}^{N_P}, j = 1, \ldots, N_S$ that best represents the dominant physical behavior featured within the snapshots. The POD basis is optimal in the sense that the projection of $\widetilde{\boldsymbol{U}}$ onto $\boldsymbol{\varphi}$ is maximized, according to the following maximization problem [16]:

$$\max_{\boldsymbol{\phi}} \frac{\langle (\widetilde{\boldsymbol{U}}, \boldsymbol{\phi})^2 \rangle}{(\boldsymbol{\phi}, \boldsymbol{\phi})} = \frac{\langle (\widetilde{\boldsymbol{U}}, \boldsymbol{\varphi})^2 \rangle}{(\boldsymbol{\varphi}, \boldsymbol{\varphi})}, \tag{2.2}$$

where $(\cdot,\cdot)$ indicates an inner product, and $\langle \cdot \rangle$ denotes an averaging operation. It can be shown that the POD basis vectors are the eigenvectors of the kernel $\mathbf{K} = \mathbf{A}\mathbf{A}^T$, where $\mathbf{K}$ is a $N_P \times N_P$ matrix. If $\boldsymbol{U}_i$ is generated by CFD simulation, N_P, i.e. the number of grid points, can easily reach the order of $10^6 - 10^8$, making it very costly to extract the eigenvectors.

By using the method of "snapshots" proposed by Sirovich [17], the desired eigenvectors $\boldsymbol{\varphi}$ can be computed as a combination of the snapshots $\boldsymbol{U}_i$, i.e.

$$\boldsymbol{\varphi}_j = \sum_{i=1}^{N_S} \beta_i^j \boldsymbol{U}_i, \tag{2.3}$$

which in turn reduces the maximization problem (2.2) to an eigenvalue-eigenvector problem in the form

$$\mathbf{R}\boldsymbol{\beta} = \boldsymbol{\Lambda}\boldsymbol{\beta}, \tag{2.4}$$

where

$$\mathbf{R} = \frac{1}{N_S} \mathbf{A}^T \mathbf{A}, \tag{2.5}$$

is a correlation matrix between snapshots. Therefore, instead of solving an eigenvalue problem for a $N_P \times N_P$ matrix $\mathbf{K}$, a smaller problem for the matrix $\mathbf{R}$ of dimension $N_S \times N_S$ has to be solved, where N_S is the number of snapshots. The eigenvalues of the covariance matrix are arranged in descending order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{N_S} \geq 0$, which have the interpretation of giving the "energy" of the system projected on the $\boldsymbol{\varphi}_j$ mode. Therefore, the fraction of the total energy associated to each mode $\boldsymbol{\varphi}_j$ can be computed as

$$E_j = \lambda_j \Big/ \sum_{i=1}^{Ns} \lambda_i. \tag{2.6}$$

An "untried" solution of the system can be approximated via a linear combination of the modes

$$\boldsymbol{U}(\mathbf{x}^\delta) = \overline{\boldsymbol{U}} + \sum_{j=1}^{M \le N_S} \alpha_j^\delta \boldsymbol{\varphi}_j, \tag{2.7}$$

where M indicates the truncation of the expansion at the desired level of energy content.

For each mode $\boldsymbol{\varphi}_j$, the projection coefficient at each snapshot location $\mathbf{x}_i$ can be expressed as

$$\alpha_j^i = \boldsymbol{U}_i \cdot \boldsymbol{\varphi}_j. \tag{2.8}$$

The $\alpha_j^i, i = 1, \dots N_S$ form a multi-dimensional response surface for each mode $\boldsymbol{\varphi}_j$, having as input the parameters $\mathbf{x}_i$ of the analysis and as outputs the α_j^i coefficients. Then for any untried input parameter $\mathbf{x}^\delta$, the mode coefficient α_j^δ can be obtained from interpolation.

2.1.2 Multi-dimensional interpolation: Kriging

Kriging was initially developed in the geostatistics community, to evaluate gold deposit at an unexplored location, from information observed at nearby locations [56]. In recent decades, following Sack's landmark work [45], Kriging has gained popularity in the engineering community, especially in design and optimization, to approximate deterministic computer models [57-60]. In these applications, Kriging is used to build surrogate models (also referred to as meta-models or response surfaces), which serve as an internal model used during optimization processes.

Suppose that we have a function sampled at N_S points $\mathbf{x}_i \in \mathbb{R}^{N_D}, i = 1, \ldots, N_S$, where N_D is the dimension of the design space, and function values at these points are denoted by $y_i = y(\mathbf{x}_i)$. Note that y is the α coefficients in the context of ROM. Kriging models the function as a realization of Gaussian stochastic process [57]. The strength of Kriging is that it provides a measure of the possible error in the predictor. As an interpolation method, the response surface passes exactly through the sampled points, therefore at these sampled points the error is 0. But in between, Kriging gives some standard error, which measures the uncertainly of the predictor at the unknown points. This uncertainty is modeled by a random variable $Y(\mathbf{x})$, which is normally distributed with mean μ and covariance σ^2. Consider two points $\mathbf{x}_i$ and $\mathbf{x}_j$, assuming the function being modeled is continuous and smooth, then the correlations between the random variables $Y(\mathbf{x}_i)$ and $Y(\mathbf{x}_j)$ is given by

$$Corr\left[Y(\mathbf{x}_i), Y(\mathbf{x}_j)\right] = \exp\left(-\text{distance}(\mathbf{x}_i, \mathbf{x}_j)\right), \tag{2.9}$$

which indicates that the closer the two points $\mathbf{x}_i$ and $\mathbf{x}_j$ are, the higher the correlation between them. The distance between the two points $\mathbf{x}_i$ and $\mathbf{x}_j$ is not Euclidean distance, but weighted by a parameter $\boldsymbol{\theta} \in \mathbb{R}^{N_D}$, which measures the relative importance of the N_D design variables

$$\text{distance}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{N_D} \theta_l \left|\mathrm{x}_{il} - \mathrm{x}_{jl}\right|^{p_l}, \tag{2.10}$$

where the parameters θ_l and p_l satisfy $\theta_l \geq 0$ and $0 < p_l \leq 2$. The $\boldsymbol{\theta}$ parameter serves the purpose to adjust the relative importance of each design variable, hence improves the accuracy of Kriging over other interpolation methods. Usually $p_l = 2$ is used as it gives smooth correlation with continuous gradient [58].

The uncertainty of the function's values at the N_S points can be expressed as

$$\mathbf{Y} = \begin{pmatrix} Y(\mathbf{x}_1) \\ \vdots \\ Y(\mathbf{x}_{N_S}) \end{pmatrix}. \tag{2.11}$$

This random vector has mean equal to $\boldsymbol{\mu}$, which is a $N_S \times 1$ vector, and the covariance matrix equals to

$$Cov(\mathbf{Y}) = \sigma^2 \mathbf{R}, \tag{2.12}$$

where $\mathbf{R}$ is a $N_S \times N_S$ matrix with $R_{i,j}$ given by Equation (2.9).

The values of θ_l $(l = 1, \dots, N_D)$ can be estimated by maximizing the concentrated log-likelihood function [57]

$$L(\boldsymbol{\theta}) = -\frac{N_S}{2}\log(\hat{\sigma}^2) - \frac{1}{2}\log(|\mathbf{R}|), \tag{2.13}$$

where

$$\hat{\mu} = \frac{\mathbf{1}'\boldsymbol{R}^{-1}\boldsymbol{y}}{\mathbf{1}'\boldsymbol{R}^{-1}\mathbf{1}}, \tag{2.14}$$

$$\hat{\sigma}^2 = \frac{(\boldsymbol{y} - \mathbf{1}\hat{\mu})'\boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu})}{N_S}, \tag{2.15}$$

and

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{N_S} \end{pmatrix}, \tag{2.16}$$

represents the observed function values.

Maximizing the likelihood function by selecting proper parameters means that the data observed will be most likely to be generated by the model. A hybrid optimization method combining genetic algorithm and gradient-based local search method is proposed in this thesis, with a brief introduction given in the following section. The gradient of the log-likelihood function can be expressed as

$$\frac{\partial L}{\partial \theta_l} = \frac{1}{2} tr\left\{\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial \theta_l}\right\} - \frac{1}{2\hat{\sigma}^2}(\boldsymbol{y} - \mathbf{1}\hat{\mu})'\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial \theta_l}\mathbf{R}^{-1}(-\mathbf{1}\hat{\mu}). \tag{2.17}$$

After obtained the parameter $\boldsymbol{\theta}$, a new function value y^δ at an untried location $\mathbf{x}^\delta$ can be determined by the Kriging predictor:

$$\hat{y}(\mathbf{x}^\delta) = \hat{\mu} + \mathbf{r}'\mathbf{R}^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu}), \tag{2.18}$$

where

$$\mathbf{r} = \begin{pmatrix} Corr[Y(\mathbf{x}^\delta), Y(\mathbf{x}_i)] \\ \vdots \\ Corr[Y(\mathbf{x}^\delta), Y(\mathbf{x}_{N_S})] \end{pmatrix}. \tag{2.19}$$

The mean square error of the predictor derived using the standard stochastic-process is

$$s^2(\mathbf{x}^*) = \hat{\sigma}^2\left[1 - \mathbf{r}'\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{1}'\mathbf{R}^{-1}\mathbf{1})^2}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}}\right], \tag{2.20}$$

and the value of $s^2(\mathbf{x}^*)$ is zero at any sampled point.

Remarks:

1. The mean term in Kriging predictor (2.18) is a constant, therefore this type of Kriging is called ordinary Kriging. Variants that use polynomial approximation or Bayesian statistics procedures for the mean term are also available, such as universal Kriging and Blind (or Bayesian) Kriging [61].
2. A desirable feature of ordinary Kriging is that with the same set of points $\{\mathbf{x}_i\}_{i=1}^{N_S}$ and the corresponding outputs $\{y_i\}_{i=1}^{N_S}$, the model parameter $\boldsymbol{\theta} \in \mathbb{R}^{N_D}$ is fixed. In other words, for any new function value y^δ at any untried $\mathbf{x}^\delta$, we can use the same $\boldsymbol{\theta}$ trained from the N_S samples, without doing the maximization of log-likelihood function repeatedly. This is especially beneficial in the context of ROM, since the most time consuming part of Kriging is the optimization process, and it can now be done offline,

making the online computations more efficient. Further discussions of the online and offline costs are presented in Section 2.6.1.

2.2 Gradient-based optimization method

Kriging interpolation requires the maximization of the likelihood function (2.13), for which the analytical formula is known, but the shape of the function might become highly nonlinear and present multiple optima, as the problem and/or the snapshots change. In the previous work of Lappo [11], genetic algorithm (GA) was adopted for the maximization of the likelihood function of Kriging. The GA was selected due to its robustness in identifying the global optimum (as opposed to local optimum), even with complex and highly nonlinear objective functions. A drawback of this approach is that it takes a relatively long time to locate the exact local optimum. Therefore, a gradient-based local search method is implemented and combined with the GA approach to speed up the search for a global optimum.

In the context of maximization of the Kriging log-likelihood function, $\boldsymbol{\theta} \in \mathbb{R}^{N_D}$ is the vector of variables, $L(\boldsymbol{\theta})$ is the objective function to be maximized. Note that although θ_l has to satisfy $\theta_l \geq 0, l = 1, \dots, N_D$, it is a trivial constrain that can be easily resolved in the code. Therefore the problem of searching for the best $\boldsymbol{\theta} \in \mathbb{R}^{N_D}$ that maximizes the log-likelihood function $L(\boldsymbol{\theta})$ can be recast into the following form of unconstrained minimization:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{N_D}} f(\boldsymbol{\theta}), \tag{2.21}$$

where $f(\cdot) = -L(\cdot)$.

Optimization algorithms start from an initial guess $\boldsymbol{\theta}_0$ and generate a series of points $\boldsymbol{\theta}_k, k = 1, \dots, \infty$ until certain stopping criterion is met. Generally, there are two strategies for searching: line search and trust region. The line search method selects a direction $\boldsymbol{p}_k$ at each iteration k and searches along this direction for lower values of objective function $f(\boldsymbol{\theta})$, and the step length α is selected using a proper algorithm. Trust region method builds a model function to approximate the objective function f near $\boldsymbol{\theta}_k$, within a certain region, and then determine search direction $\boldsymbol{p}_k$ by solving a sub-problem in the trust region [62]. In this thesis, line search method is selected. As a classical optimization method, detailed description and algorithm can be found in [62, 63], with only a brief summary given here for the purpose of self-containment.

2.2.1 Selection of search direction

Regarding search directions, four typical methods are listed in Table 2-1. Although Newton's method has a quadratic rate of convergence, it can sometimes be expensive and error-prone, due to the explicit computation of the Hessian matrix. A quasi-Newton (QN) method was therefore selected, as it does not require the computation of the Hessian and yet still attains a

Table 2-1: Summary of search directions

Search direction	Definitions	Pros and cons	Rate of convergence
Steepest descent	$\boldsymbol{p}_k = -\nabla f_k$	- Does not require Hessian - Can be very slow for difficult problems	Linear
Conjugate gradient	$\boldsymbol{p}_k^N = -\nabla f(\boldsymbol{\theta}_k) + \beta_k \boldsymbol{p}_{k-1}$	- Does not require storage of matrices - Not as fast as Newton or quasi-Newton	Linear
Newton	$\boldsymbol{p}_k^N = -(\nabla^2 f_k)^{-1} \nabla f_k$	- Fast convergence - Requires Hessian, may be expensive and error prone	Quadratic
Quasi-Newton	$\boldsymbol{p}_k^N = -\boldsymbol{B}_k^{-1} \nabla f_k$	- Fast convergence - Does not require Hessian	Super-linear

Note: ∇f_k is the gradient of f at point $\boldsymbol{\theta}_k$, $\nabla^2 f_k$ is the Hessian matrix, β is a scalar to ensure $\boldsymbol{p}_k$ and $\boldsymbol{p}_{k-1}$ are conjugate.

super-linear rate of convergence. In place of the Hessian $\nabla^2 f_k$, QN uses an approximation $\boldsymbol{B}_k$, which is updated after each iteration step, by incorporating information obtained from the current step. The most effective QN update algorithm is the BFGS method, named for its developers Broyden, Fletcher, Goldfarb and Shanno

$$\boldsymbol{B}_{k+1} = \boldsymbol{B}_k - \frac{\boldsymbol{B}_k \boldsymbol{s}_k \boldsymbol{s}_k^T \boldsymbol{B}_k}{\boldsymbol{s}_k^T \boldsymbol{B}_k \boldsymbol{s}_k} + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{y}_k^T \boldsymbol{y}_k}, \tag{2.22}$$

where $\boldsymbol{s}_k = \boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k$, $\boldsymbol{y}_k = \nabla f_{k+1} - \nabla f_k$.

2.2.2 Determine step length

In line search methods, in each iteration, the algorithm selects a search direction $\boldsymbol{p}_k$, and decides how far to go in this direction,

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha_k \boldsymbol{p}_k. \tag{2.23}$$

The step length α_k has to meet the Wolfe conditions to: 1) ensure sufficient decrease of the objective function, and 2) eliminate unacceptably small steps to make the algorithm efficient.

The ideal step length would be the global minimizer of function $\phi(\cdot)$, defined by

$$\phi(\alpha) = f(\boldsymbol{\theta}_k + \alpha \boldsymbol{p}_k), \qquad \alpha > 0. \tag{2.24}$$

As it is too expensive to solve for the exact global minimizer of ϕ, in practice an inexact line search is performed to identify a suitable step length, which produces sufficient decrease of objective function value f at minimum cost. To find a minimum of the one-dimensional function (2.24), the step length is determined using an iterative backtrack procedure [62]. Namely, starting from an initial step length $\alpha_0 = 1$, if $\boldsymbol{\theta}_k + \boldsymbol{p}_k$ is not acceptable, reduce α until an acceptable $\boldsymbol{\theta}_k + \alpha_k \boldsymbol{p}_k$ is found.

The sufficient decrease condition can be written in the form of ϕ,

$$\phi(\alpha) \leq \phi(0) + c_1 \alpha \phi'(0). \tag{2.25}$$

In practice, $c_1$is quite small, e.g. 10^{-4}. Suppose initial step length $\alpha_0 = 1$, if

$$\phi(\alpha_0) \leq \phi(0) + c_1 \alpha_0 \phi'(0), \tag{2.26}$$

then this step length satisfies the sufficient condition, and the search can be terminated. Otherwise, the acceptable step length lies in the interval $[0, \alpha_0]$. A quadratic approximation $\phi_q(\alpha)$ to ϕ can be constructed as

$$\phi_q(\alpha) = \left(\frac{\phi(\alpha_0) - \phi(0) - \alpha_0 \phi'(0)}{\alpha_0^2}\right)\alpha^2 + \phi'(0)\alpha + \phi(0), \tag{2.27}$$

which interpolates three known quantities: $\phi(0)$, $\phi'(0)$ and $\phi(\alpha_0)$. Note that $\phi_q(0) = \phi(0)$, ${\phi_q}'(0) = \phi'(0)$, $\phi_q(\alpha_0) = \phi(\alpha_0)$. The new value α_1 can be obtained by setting $\phi_q'(\alpha) = 0$, which gives

$$\alpha_1 = -\frac{\phi'(0)\alpha_0^2}{2[\phi(\alpha_0) - \phi(0) - \alpha_0 \phi'(0)]}. \tag{2.28}$$

If this α_1 satisfies the sufficient decrease condition, terminate the search. Otherwise, construct a cubic function that interpolates the four known quantities: $\phi(0)$, $\phi'(0)$, $\phi(\alpha_0)$ and $\phi(\alpha_1)$, which is

$$\phi_c(\alpha) = a\alpha^3 + b\alpha^2 + \phi'(0)\alpha + \phi(0). \tag{2.29}$$

where

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\alpha_0^2 \alpha_1^2 (\alpha_1 - \alpha_0)} \begin{bmatrix} \alpha_0^2 & -\alpha_1^2 \\ -\alpha_0^3 & \alpha_1^3 \end{bmatrix} \begin{bmatrix} \phi(\alpha_1) - \phi(0) - \phi'(0)\alpha_1 \\ \phi(\alpha_0) - \phi(0) - \phi'(0)\alpha_0 \end{bmatrix}. \tag{2.30}$$

By differentiating $\phi_c(\alpha)$, one can obtain the minimizer α_2 of ϕ_c which lies in the interval $[0, \alpha_1]$ given by

$$\alpha_2 = \frac{-b + \sqrt{b^2 - 3a\phi'(0)}}{3a}. \tag{2.31}$$

Once the suitable step length is determined, a new search point can be found using Equation (2.23), and the algorithm proceeds to the next search iteration, until a certain termination criterion is met.

2.3 Local POD using machine learning algorithms

2.3.1 The role of machine learning

The field of machine learning deals with the construction of algorithms that can learn patterns from a collection of data and make accurate predictions of untried conditions. Machine learning is based on the principles of statistics and probability, and overlaps extensively with other fields like pattern recognition, data mining, and artificial intelligence. Machine learning algorithms can be generally categorized as supervised and unsupervised learning [64, 65]. In supervised learning, one has a set of training data that comprises input variables and the corresponding output. A learner/model is trained on the basis of existing data to predict the output from the input values. The method is called "supervised" because the learning process is guided by the available outputs. For unsupervised learning, on the other hand, the training data consists of input data only, and the goal is to discover underlying structures within the data. Without a known output to guide the prediction, this process is called "unsupervised learning".

In the context of the proposed local POD, the target of grouping the snapshots into clusters can be achieved via an unsupervised learning algorithm known as *k*-means clustering, also adopted in [42-44]. In this context the snapshots are the inputs and the desired output is the label that identifies a snapshot with a cluster. The identification of the proper cluster for an untried solution and the definition of the boundaries of each cluster in the parameter space are problems that fall into the category of supervised learning, more specifically classification. Given the operating conditions of the snapshots and the class labels obtained from *k*-means clustering, a training set of data can be defined. Using this data set, one can train a prediction model, or learner, which will predict the cluster (class labels) to which the untried solution is expected to belong, based on the predicted posterior probability. The boundaries of the subregions are called decision boundaries defined by a hyperplane that has equal probability for two neighboring clusters. For self-containment, the ideas behind *k*-means clustering and logistic regression are briefly outlined below.

2.3.2 *K*-means clustering

K-means algorithm is the most used clustering method, also referred to as Lloyd's algorithm [66]. In a clustering problem, one is given a data set $\{\boldsymbol{U}_1, \dots, \boldsymbol{U}_{N_S}\}$, where $\boldsymbol{U}_i \in \mathbb{R}^{N_P}, i = 1, \dots, N_S$ and the goal is to partition the data set into K subsets, such that those solutions that are within each cluster share similar features. The dissimilarity/distance between two samples $\boldsymbol{U}_i$ and $\boldsymbol{U}_j$ can be computed by the Euclidean distance squared

$$d(\boldsymbol{U}_i, \boldsymbol{U}_j) = \left\|\boldsymbol{U}_i - \boldsymbol{U}_j\right\|^2. \tag{2.32}$$

Supposing that the number of desired clusters K is given, then the problem of clustering is equivalent to minimizing an objective function, also called a distortion measure, given by [64]

$$J = \sum_{i=1}^{N_S} \sum_{k=1}^{K} r_{ik} \|\boldsymbol{U}_i - \boldsymbol{\mu}_k\|^2, \tag{2.33}$$

where $r_{ik} \in \{0,1\}$ is a binary indicator defined as

$$r_{ik} = \begin{cases} 1 & \text{if } k = \arg\min_j \left\|\boldsymbol{U}_i - \boldsymbol{\mu}_j\right\|^2 \\ 0 & \text{otherwise,} \end{cases} \tag{2.34}$$

and $\boldsymbol{\mu}_k$ is cluster center given by

$$\boldsymbol{\mu}_k = \frac{\Sigma_i r_{ik} \boldsymbol{U}_i}{\Sigma_i r_{ik}}. \tag{2.35}$$

The objective function J can be minimized by a two-step iterative procedure:

1. Assign the samples to the nearest cluster center;
2. Update the cluster center to the mean of samples assigned to it.

Steps 1 and 2 are iterated until the assignments no longer change. Given that the procedure may converge to a local minimum, it is advised [65] to initialize the cluster center $\boldsymbol{\mu}_k$ multiple times by randomly choosing K samples in the training set, and select the one that yields the lowest objective function value.

2.3.3 Logistic regression

Logistic regression (LR) is a type of generalized linear method adopted for classification problems [64, 65]. To assign an input vector $\mathbf{x}_i \in \mathbb{R}^{N_D}$ to one among K available classes $\mathcal{C}_k, k = 1, \dots, K$, the learner/model is trained using a training set $S = \{(\mathbf{x}_1, t_1), \dots, (\mathbf{x}_{N_S}, t_{N_S})\}$, where $t \in \mathbb{R}$ is the label of the class. Since the classes are disjoint sets, the input space can always be divided into decision regions, with boundaries identified as decision boundaries. Although LR is a linear model, it can account for nonlinear features by using nonlinear basis functions $\boldsymbol{\phi}(\mathbf{x})$ (e.g. polynomial and trigonometric expansion, radial basis expansion, to name a few) such that nonlinear decision boundaries can be obtained in the original input space $\mathbf{x}$, while being linear in the expanded feature space $\boldsymbol{\phi} \in \mathbb{R}^{N_F}$ [64, 65].

An example of polynomial expansion is illustrated as follows: let $\{\phi_j(\mathbf{x}) | j = 1, \dots, N_F\}$ be a basis of the set of all polynomials in $\mathbf{x}$ of degree G, since $\mathbf{x} = (x_1, x_2, \dots, x_{N_D}) \in \mathbb{R}^{N_D}$, each $\phi_j(\mathbf{x})$ is a term like $x_1^{g_1} x_2^{g_2} \cdots x_{N_D}^{g_{N_D}}$ where $g_1 + g_2 + \cdots + g_{N_D} \leq G$. In the case where $N_D = 2$, degree $G = 1$, we will have $\phi_1(\mathbf{x}) = 1$, $\phi_2(\mathbf{x}) = x_1$, $\phi_3(\mathbf{x}) = x_2$ and $N_F = 3$. Namely, $\boldsymbol{\phi}(\mathbf{x}) = \sum_{j=1}^{N_F} \phi_k(\mathbf{x}) = 1 + x_1 + x_2$. While for case with $N_D = 2$, degree $G = 2$, we will have $\phi_1(\mathbf{x}) = 1$,

$\phi_2(\mathbf{x}) = x_1^2$, $\phi_3(\mathbf{x}) = x_1 x_2$, $\phi_4(\mathbf{x}) = x_2^2$, and $N_F = 4$, then $\boldsymbol{\phi}(\mathbf{x}) = \sum_{p=1}^{N_F} \phi_j(\mathbf{x}) = 1 + x_1^2 + x_1 x_2 + x_2^2$, so on and so forth.

Consider the case of a two-class (i.e. $K = 2$) classification, where $t \in \{0,1\}$, such that $t = 1$ represents class $\mathcal{C}_1$ and $t = 0$ represents class $\mathcal{C}_2$. The posterior probability of class $\mathcal{C}_1$ can be expressed as a logistic sigmoid applied to a linear function of the feature vector $\boldsymbol{\phi}$, that is

$$p(\mathcal{C}_1|\boldsymbol{\phi}) = y(\boldsymbol{\phi}) = \sigma(\boldsymbol{w}^T\boldsymbol{\phi}), \tag{2.36}$$

with $p(\mathcal{C}_2|\boldsymbol{\phi}) = 1 - p(\mathcal{C}_1|\boldsymbol{\phi})$, this model is known as logistic regression [64]. Here $\sigma(\cdot)$ is the logistic sigmoid function defined by

$$\sigma(a) = \frac{1}{1 + \exp(-a)}, \tag{2.37}$$

that maps the real axis into a finite interval $[0, 1]$, representing posterior probability ranging from zero to one. The decision boundary is the hyperplane defined by

$$\{\boldsymbol{\phi} \mid \boldsymbol{w}^T\boldsymbol{\phi} = 0\}, \tag{2.38}$$

which has equal probability for classes $\mathcal{C}_1$ and $\mathcal{C}_2$. From Equation (2.36) we have

$$y(\boldsymbol{\phi}) = \sigma(\boldsymbol{\phi}, \boldsymbol{w}) = \sigma\left(\sum_{j=0}^{N_F} w_j \phi_j(\mathbf{x})\right) = \sigma\big(\boldsymbol{w}^T\boldsymbol{\phi}(\mathbf{x})\big), \tag{2.39}$$

where $\boldsymbol{\phi} = \left(\phi_1, \dots, \phi_{N_F}\right)^T$ is the feature vector and $\boldsymbol{w} = \left(w_1, \dots, w_{N_F}\right)^T$ is the model parameter which can be determined by minimizing a penalized cost function defined by

$$J(\boldsymbol{w}) = -\sum_{n=1}^{N_S}\{t_n \ln(y_n) + (1 - t_n)\ln(1 - y_n)\} + \frac{\lambda}{2}\sum_{j=2}^{N_F} w_j^2, \tag{2.40}$$

where

$$y_n = p(\mathcal{C}_1|\boldsymbol{\phi}) = \sigma(\boldsymbol{w}^T\boldsymbol{\phi}_n). \tag{2.41}$$

This cost function consists of two parts: the negative logarithm of the likelihood $p(\boldsymbol{t}|\boldsymbol{w}) = \prod_{n=1}^{N_S} y_n{}^{t_n}(1-y_n)^{1-t_n}$ modeling the conditional Bernoulli distribution of t_n given $\boldsymbol{\phi}_n$, and a regularization term with coefficient λ to balance model fitting. If a model perfectly fits all data points in the training set, it will lack generality when predicting unseen data, thus λ is introduced to prevent this over-fitting phenomenon. On the other hand, large value of λ will heavily penalize the $\boldsymbol{w}$ parameters, thus cause under-fitting. To this effect, the proper regularization coefficient λ has to be determined by trial and error.

As the resulting optimization problem from Equation (2.40) is convex [65], a gradient-based algorithm can be used to search for the best model parameter $\boldsymbol{w}$. The quasi-Newton line search method with BFGS updating is selected for the optimization.

For multi-class classification, $t \in \{1, \dots, K\}$. The problem can be converted into binary classification using a technique called one-versus-the-rest or one-versus-all [64, 65], which trains K binary classifiers $f_k: \mathbb{R}^{N_D} \mapsto \{0,1\}$ on the training set S, each distinguishes one class k from all the others. An unseen sample will be assigned to the class whose classifier gives highest probability

$$\forall \mathbf{x} \in \mathbb{R}^{N_D}, \qquad p(\mathbf{x}) = \underset{k\in\{1,\dots,K\}}{\operatorname{argmax}} f_k(\mathbf{x}). \tag{2.42}$$

2.4 Error driven iterative sampling method

Local reduced-order models can meet the challenge of addressing highly nonlinear problems characterized by distinct physical regimes. However, identifying the best number and location of snapshots in the parameter space is a non-trivial task. The POD-based ROM can be seen as a spectral method with empirical basis functions. Therefore, the quality of the reduced-order solutions is critically dependent on the information provided by the POD basis, which is in turn determined by the snapshots. Since it may not be known a priori how many snapshots to compute and/or which operating conditions are the most important to consider, an iterative sampling strategy based on a "greedy" application of centroidal Voronoi tessellation (CVT) is proposed, where an initial ensemble of snapshots is iteratively optimized according to a prescribed error level. The concept behind this iterative sampling approach is that if the inaccuracy of the reduced model is caused by insufficient information contained in the snapshot set, then the accuracy may be improved by enriching the set of basis considering new snapshots positioned in areas of high error. The CVT based iterative sampling strategy is analogous in principle to greedy approaches [48-54]. Namely, they all place points where the error is high. However, the way the error is computed is very different. Classical greedy approaches rely on residual-based error bounds and error indicators, while in this thesis the leave-one-out cross-validation (LOOCV) error has been selected to guide the proposed iterative sampling process.

The iterative sampling strategy improves an initial distribution of snapshots in the parameter space in three steps during each iteration:

1) Error-driven sampling of the parameter space;
2) Computation of the snapshots via CFD;
3) ROM versus CFD error evaluation via LOOCV.

The flowchart of error driven iterative sampling is show in Figure 2-1.

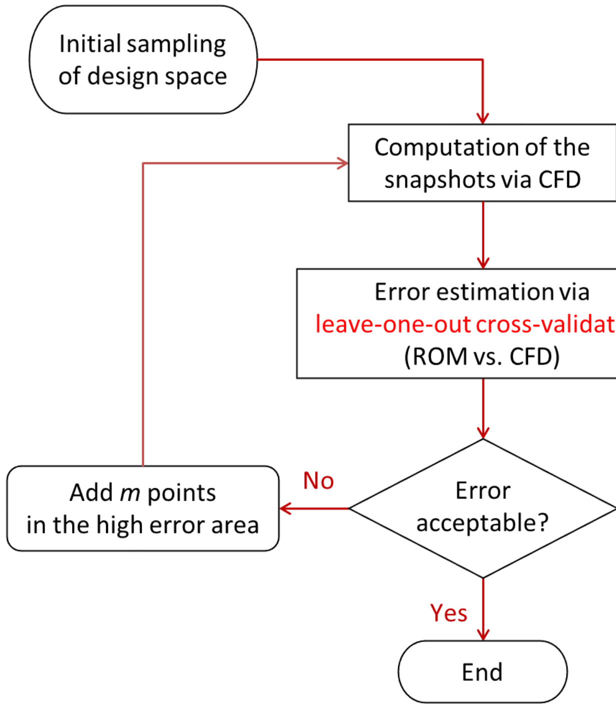


Figure 2-1: Flowchart of error driven iterative sampling.

2.4.1 Centroidal Voronoi tessellation

Defining a suitably rich set of snapshots in the parameter space (also referred to as design space) is a problem of design of experiments. In view of iterative sampling, it becomes necessary for the algorithm to be able to add more samples in desired locations of the design space without changing the previous distribution. Moreover, the method should be able to define samples that are either uniformly distributed in the design space or aggregated according to a specified density function. Given all this, CVT is adopted to perform the sampling, since it has superior capabilities of uniform and dispersed sampling with respect to other classical methods, and most importantly, it allows for aggregating of samples in the areas of higher density.

Given an open set $\Omega \subseteq \mathbb{R}^{N_D}$ and a set of points $\mathbf{z}_i \in \Omega, i = 1, \dots, s,$ the Voronoi region $\hat{V}_i$ corresponding to the point $\mathbf{z}_i$ is defined as

$$\hat{V}_i = \{\mathbf{x} \in \Omega \mid \|\mathbf{x} - \mathbf{z}_i\| < \|\mathbf{x} - \mathbf{z}_j\| \text{ for } j = 1, \dots, s, j \neq i\}, \tag{2.43}$$

where $\|\cdot\|$ denotes the Euclidean norm. The points $\{\mathbf{z}_i\}_{i=1}^{s}$ are called generators of these Voronoi region, and the set $\{\hat{V}_i\}_{i=1}^{s}$ is called a Voronoi tessellation of Ω. Given a density function $\rho(\mathbf{x})$ defined in $\hat{V}_i$, the mass centroid $\mathbf{z}_i^*$ of the Voronoi region $\hat{V}_i$ is

$$\mathbf{z}_i^* = \frac{\int_{V_i} \mathbf{x}\rho(\mathbf{x})d\mathbf{x}}{\int_{V_i} \rho(\mathbf{x})d\mathbf{x}}. \tag{2.44}$$

If the generators $\mathbf{z}_i$ for the Voronoi regions $\hat{V}_i$ are at the same time the mass centroids of these regions, namely

$$\mathbf{z}_i = \mathbf{z}_i^*, i = 1, \dots, s, \tag{2.45}$$

then such a tessellation is called a centroidal Voronoi tessellation [55].

CVT-based sampling is identical to finding s generators $\mathbf{z}_i$ that tessellate the design space into s Voronoi regions V_i for $\mathbf{z}_i$ and $\mathbf{z}_i = \mathbf{z}_i^*$. In this context, discrete CVT is adopted. Instead of a region Ω, the design space is represented by a discrete set of points $W = \{\mathbf{x}_i\}_{i=1}^{n} \in \mathbb{R}^{N_D}$. Voronoi sets corresponding to generators $\{\mathbf{z}_i\}_{i=1}^{s} \in \mathbb{R}^{N_D}$ are now defined by

$$\begin{aligned} \hat{V}_\mathrm{i} = \{\mathbf{x} \in W \mid \|\mathbf{x} - \mathbf{z}_i\| \leq \|\mathbf{x} - \mathbf{z}_j\| \text{ for } j = 1, \dots, s, \\ j \neq i, \text{where equality holds only for } i < j\}. \end{aligned} \tag{2.46}$$

And the mass centroid $\mathbf{z}^*$ of a Voronoi set $V \subset W$ is given by

$$\sum_{\mathbf{x} \in V} \rho(\mathbf{x})\|\mathbf{x} - \mathbf{z}^*\|^2 = \inf_{\mathbf{z} \in V^*} \sum_{\mathbf{x} \in V} \rho(\mathbf{x})\|\mathbf{x} - \mathbf{z}\|^2, \tag{2.47}$$

where V^* can be taken to be V or a larger set like $\mathbb{R}^{N_D}$ [55].

In [55] it is stated that discrete CVT is closely related to *k*-means clusters, for which Voronoi regions and centroids are referred to as clusters and cluster centers, respectively. Therefore Lloyd's method is adopted for discrete CVT, which performs a two-step iterative process between constructing Voronoi tessellation and replacing generators with the mass centroids. The energy (which is also referred to as cost or distortion error) is given by

$$\mathcal{F}\big((\mathbf{z}_i, V_i), i = 1, \dots, s\big) = \sum_{i=1}^{s} \sum_{\mathbf{x} \in V_i} \rho(\mathbf{x}) \|\mathbf{x} - \mathbf{z}_i\|^2 d\mathbf{x}. \tag{2.48}$$

Depending on how the density function is defined, the CVT is capable to perform three types of sampling:

- Uniform sampling

$$\rho(\mathbf{x}) \equiv 1. \tag{2.49}$$

- Biased sampling

  $\rho(\mathbf{x})$ can be defined as standard probability density function, e.g. Gaussian distribution,

$$\rho(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}, \tag{2.50}$$

  where the mean and the variance are denoted by μ and σ^2. Or exponential distribution,

$$\rho(x \mid \lambda) = e^{\lambda x}, \tag{2.51}$$

  where λ is the rate parameter.

- Error driven sampling

  $\rho(\mathbf{x})$ is defined on the error obtained from the LOOCV procedure of ROM, described in the next paragraph.

2.4.2 Leave-one-out cross-validation

The error of the ROM prediction can be obtained by comparing the reduced solution with a reference solution (in the case of this thesis a high fidelity CFD solution) via a leave-one-out cross-validation approach [64, 65]. Given a set of N_s snapshots, one of the snapshots is excluded from the set and adopted as a reference solution. Next, the ROM solution for the condition of the selected reference solution is defined on the basis of the other $N_s - 1$ snapshots, and finally compared with the reference one. In order to have an indication over the entire parameter space of the analysis, this approach is repeated N_s times, excluding at each time a different snapshot. A flowchart illustrating this process is given in Figure 2-2.

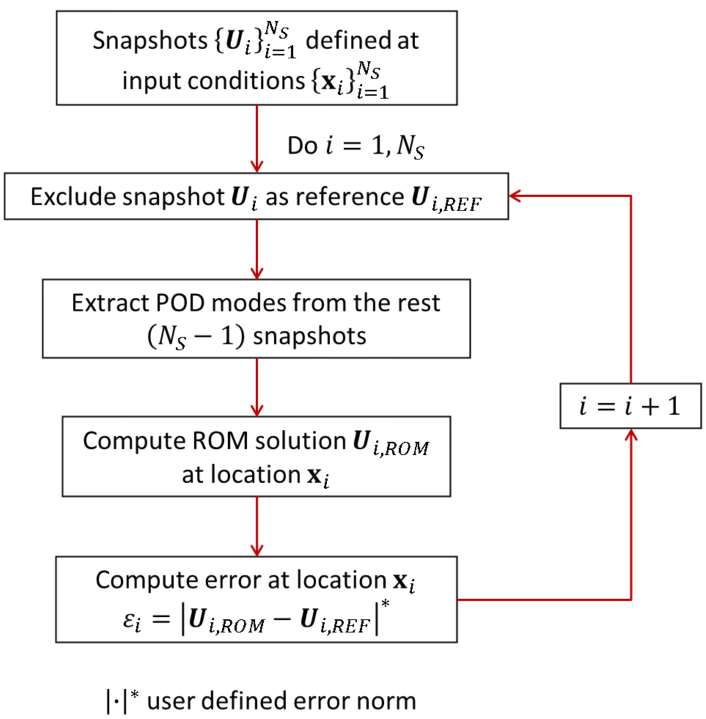


Figure 2-2: Flowchart of LOOCV.

LOOCV assesses the ROM error when removing one snapshot. It is expected that adding the excluded snapshot will help improve the accuracy, since a richer set of snapshots will be used for the final analysis. Whenever the maximum error is revealed exactly at the location of a taken out snapshot, which means that it is a crucial snapshot and that the area is characterized by high nonlinearity, CVT introduces new points (snapshots) in the neighborhood of that location but not exactly at that location since two generators cannot coincide in the CVT context. Moreover, LOOCV is a general approach that works with any type of snapshot, making it an optimal choice for non-intrusive ROM. Through the iteration process, the number of snapshots will increase, and eventually be rich enough to provide a reliable error estimation.

2.5 CFD-icing tools

2.5.1 The FENSAP-ICE package

All CFD-aero and CFD-icing snapshots adopted in this paper are obtained using FENSAP-ICE simulation system [67], which includes modules for N-S flow prediction, water droplet impingement computation by an Eulerian method, the prediction of the 3D ice accretion/water runback, and conjugate heat transfer for anti-icing and de-icing, all based on partial differential equations for viscous turbulent flows [68-71]. Validation of FENSAP-ICE is beyond the scope of this thesis and has been extensively covered in other articles [68-71]. It should be pointed out that although ROM inherits the quality of the underlying CFD, the proposed framework for model reduction is independent of the accuracy of the CFD solver.

2.5.2 Configuration for icing calculation

For the simulation of ice accretion for a certain period of time, several approaches are available such as one-shot, multi-shot or unsteady icing calculations, as illustrated in Figure 2-3

[72]. In multi-shot approach, each module (airflow, impingement, accretion) are solved independently, with selected variables passed between them. At the end of each iteration, a displaced grid accommodating the accreted shape of ice is generated. This grid is used by the flow solver for the next iteration.

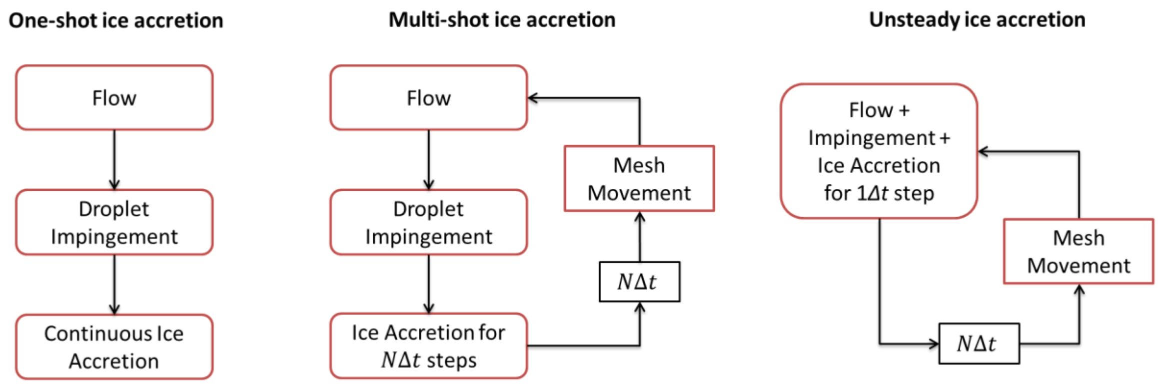


Figure 2-3: One-shot, multi-shot and unsteady ice accretion configuration.

2.6 Computational cost and parallelization

All methodologies proposed in this thesis are implemented using FORTRAN 90. Parallel computation is realized via Message Passing Interface (MPI). The computational cost can be split into offline and online costs, as described below.

2.6.1 Computational cost

2.6.1.1 Offline cost

The collection/computation of snapshots, the adaptive sampling and training of local reduced-order models can be considered as the offline cost. For LOOCV, the offline cost is equivalent to building N_S reduced-order models, which includes POD basis extraction on the remaining $N_S - 1$ snapshots and multi-dimensional interpolation for the modes coefficients. The

LOOCV cost depends on the complexity/size of the problem, i.e. the number of snapshots N_S, the dimension of snapshots N_P, the nonlinearity of the problem (high nonlinearity requires more modes to be retained in the expansion to ensure the desired level of energy content), and the interpolation method. When the LOOCV error is reduced to an acceptable level via iterative sampling, the set of snapshots is fixed. Then the extracted POD bases and their associated projection coefficients can be stored in a database for later use. Moreover, as mentioned in Section 2.1.2, the Kriging parameter $\boldsymbol{\theta}$ is fixed once the snapshots are fixed. Therefore the $\boldsymbol{\theta}$ parameter can also be computed and stored in the same database, such that the time-consuming optimization process is avoided during the online stage.

2.6.1.2 Online cost

To compute a target solution, the online cost consists only in the identification of the corresponding local ROB and interpolation of coefficients for the modes. Depending on the size of the problem, i.e. the number of modes M and dimension of mode N_P, the cost of the online phase is usually of the order of seconds, or a fraction of a second, as opposed to the hours required by high-fidelity CFD.

2.6.2 Parallel computations

Computation of snapshots defined from DoE is well suited for parallel computing, since the computations are completely independent of each other. Note that the snapshots may contain multiple solution fields. For example, in a flow solution, fields like density, pressure, velocity (x, y, z components), temperature and shear stress (x, y, z components) are typically computed. In this case, the ROM operations are performed field by field. Moreover, it is possible for the user to specify arbitrary fields that are of interest to be retained in the ROM solution, making the analysis more efficient.

For small-scale problems, like the 2D analysis presented in Chapter 3, a single CPU is capable of achieving near-real-time performance. However, for large-scale problems, like the 3D analysis presented in Chapter 4, where the snapshots are N-S solutions solved on a mesh of approximately 10 million nodes, the execution speed of ROM must still be accelerated via parallel computation. MPI is implemented in two parts of the code: POD and calculation of modes coefficients. For POD, the snapshots field variables are stored in a matrix of size $N_P \times N_S$ (for 3D N-S solutions, the matrix size is typically in the order of $10^7 \times 10^2$). This matrix is split into N_{CPU} blocks, and each processor operates on its own portion of data to obtain the POD modes (Equation 2.3 and 2.5). For the interpolation of modes coefficients, M coefficients (tasks) are distributed onto N_{CPU} processors. Eventually, a linear combination of the modes is computed and the solution is written. For LOOCV, the building of each one of the N_S reduced-order models is already parallelized using MPI. Moreover, given the independence of the operations involved, the computation of the N_S LOOCV can be "embarrassingly" parallelized to further reduce the total offline cost.

CHAPTER 3 TWO-DIMENSIONAL ANALYSIS

The proposed local reduced-order modeling coupled with the iterative sampling methodology is applied to the problem of in-flight icing certification. A two-dimensional airfoil has been considered as the reference geometry for which to assess the effects of ice accretion over the continuous maximum (CM) and intermittent maximum (IM) icing envelopes. The geometry selected is a GLC-305, a business jet type of airfoil with a chord length of 36 inches. Airspeeds and altitudes typical of a business jet aircraft were used, e.g. pressure altitude of 10,000 feet with a true air speed (TAS) of 175 knots [73]. For the present study, accretion time was computed based on airspeed and the standard reference cloud extent. The purpose is to demonstrate that Appendix C can be explored via ROM in terms of both ice characteristics and aerodynamic degradation. Namely, one can obtain the accreted ice shape and the mass of ice at all conditions inside the icing envelope, based on a set of pre-computed ice shapes taken as snapshots. Similarly, aerodynamic performance in terms of $C_L - \alpha$ curves can also be obtained everywhere inside the icing envelope on the basis of a set of pre-computed $C_L - \alpha$ curves taken as snapshots.

Numerical solutions are taken as snapshots, as well as references to examine the quality of reduced-order solutions. The calculation of the snapshots was performed on a Compute Canada supercomputing cluster. A hybrid mesh was employed with 204,612 nodes, 57,989 quadrilateral elements and 87,495 triangular elements (Figure 3-1). For the analysis of the ice shape, each snapshot $\boldsymbol{U}_i$ consists of the x and y coordinates of the 1049 nodes defining the surface mesh. For the aerodynamic analysis, each snapshot $\boldsymbol{U}_i$ consists of lift coefficient curves for a range of angles of attack from $-4°$ to $15°$, at a Mach number of 0.21 and a Reynolds number of 10.5×10^6.

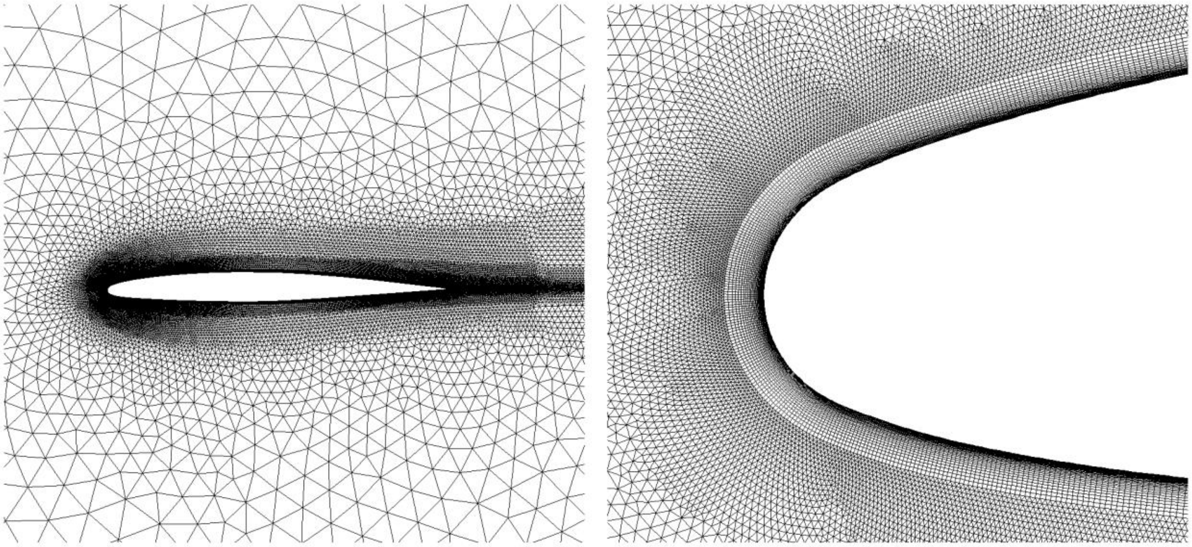


Figure 3-1: Hybrid mesh around the GLC305 airfoil.

The accuracy of the ROM predictions versus CFD results is evaluated via the LOOCV. Two types of errors corresponding to the two types of snapshots are defined. For the shape of ice, an integral quantity i.e. the mass of ice is adopted

$$\varepsilon_{mass}^{i} = \left| mass_{ice,ROM}^{i} - mass_{ice,CFD}^{i} \right|, i = 1, \ldots, N_S, \tag{4.1}$$

where the unit is gram per unit span. Mass of ice is computed by subtracting the area of clean airfoil from the area of ice accumulated airfoil, then multiplying ice density (assumed constant at 917 kg/m^3). For the lift coefficient curve, the error for each snapshot $\boldsymbol{U}_i$ is defined as the infinity norm of the vector of the errors ε_j^i at each point j of the solution

$$\begin{gathered} \varepsilon_j^i = \left| \frac{U_{j,ROM}^i - U_{j,CFD}^i}{U_{j,CFD}^i} \right|, i = 1, \ldots, N_S, j = 1, \ldots, N_P \\ \varepsilon_{C_L}^i = \max\left\{ \varepsilon_1^i, \cdots, \varepsilon_{N_P}^i \right\}, i = 1, \cdots, N_S. \end{gathered} \tag{4.2}$$

In both icing envelopes, it is possible to identify three typical types of icing snapshots: no or trace ice (clean or nearly clean airfoil), glaze/mixed ice and rime ice. The motivation for this behavior can be explained as follows: at certain airspeed, for a narrow region near the 32 °F boundary, the stagnation temperature is higher than 32 °F, such that no ice will be accumulated on the airfoil. Moving from the 32 °F boundary towards the -22 °F or -40 °F boundaries of the CM and IM, the temperature continues to decrease, creating icing conditions. The type of ice varies from glaze to mixed for relatively high temperatures, and eventually rime ice for lower ones. These different types of solutions will pollute each other in the context of ROM, an expected result of using global POD for highly nonlinear problems. For aerodynamic performance, the pollution is more evident: the clean airfoil has a much higher $C_{L,max}$ than the ice-contaminated airfoil, and their proximity in the icing envelope (the narrow transition region from no-ice area to ice-contaminated area) makes it very difficult for global ROM to make accurate predictions. Local ROM is therefore introduced to handle these distinct solutions separately by subdividing the icing envelope into three subregions. In particular, given that all snapshots in the no-ice subregion are identical, no ROM computations are required for untried conditions classified into this region, provided the ice-free zone can be properly identified by a decision boundary. Detailed analysis will be given in the remainder of this section.

3.1 Continuous maximum

3.1.1 Initial sampling and the snapshots

The initial sampling consists of 56 points (Figure 3-2), each representing a different icing condition in terms of MVD, LWC and, implicitly, external temperature. This initial sampling is not uniform but clustered towards the high temperature and high LWC region, because experience suggests that near freezing temperature and large amount of LWC is more likely to produce glaze ice, which has irregular horns that are more difficult for ROM to predict, comparing with streamline shaped rime ice. Therefore, more samples are placed in this region.

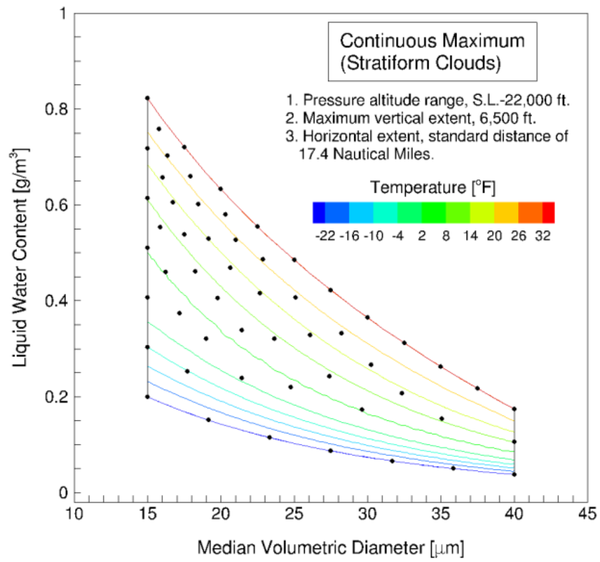


Figure 3-2: Initial sampling of the CM.

The exposure time considered for the CM is 4 minutes, and the ice shape snapshots are obtained via a 4-shot CFD-icing calculation. Figure 3-3 to Figure 3-5 illustrate three typical ice shapes and the associated $C_L - \alpha$ curves, as well as the Mach number contours for three icing conditions selected from the original set of snapshots. These figures demonstrate ice shape comparisons and $C_L - \alpha$ curve comparisons of ROM solution versus CFD solution, during LOOCV at this specific snapshot location, i.e. the ROM solution is obtained based on the remaining 55 snapshots. The solution shown in Figure 3-3 is a horn-like shape glaze ice, determined by relatively high temperature and large liquid water content; the solution shown in Figure 3-4 is a streamlined shape rime ice, formed under very low temperature conditions; and the solution shown in Figure 3-5 is the clean airfoil without any ice formation, in above freezing total temperature conditions. It is worth noting that for glaze ice with irregular horns, as shown in Figure 3-3, although ROM did not capture precisely the shape of ice horns, the aerodynamic performance prediction in terms of $C_L - \alpha$ curve could still be very accurate.

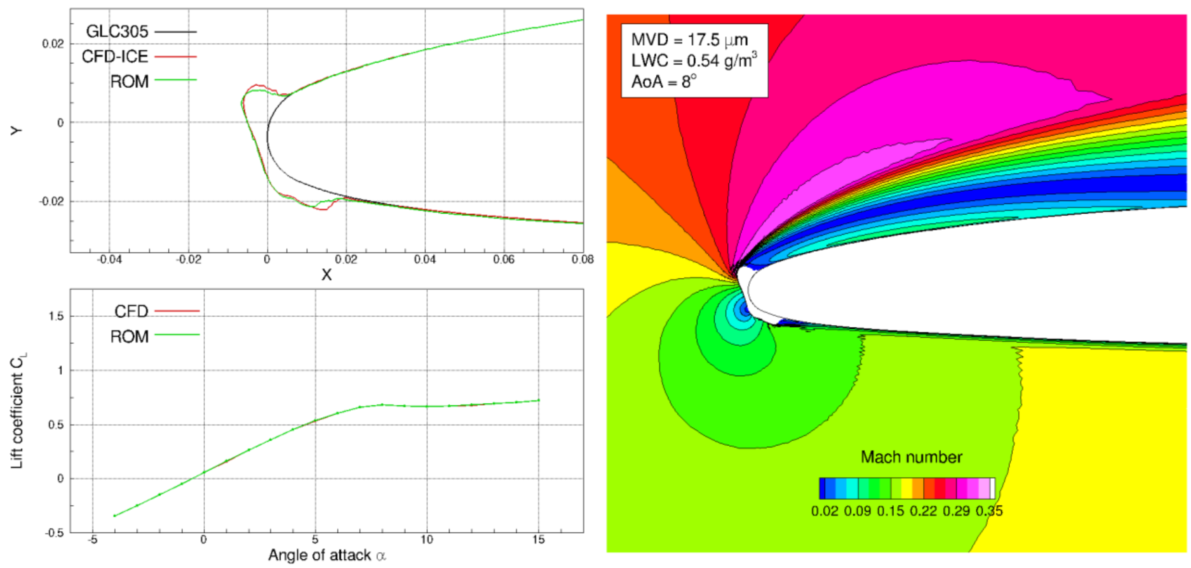


Figure 3-3: Glaze ice and the associated C_L - α curve (left); Mach number contours (right).

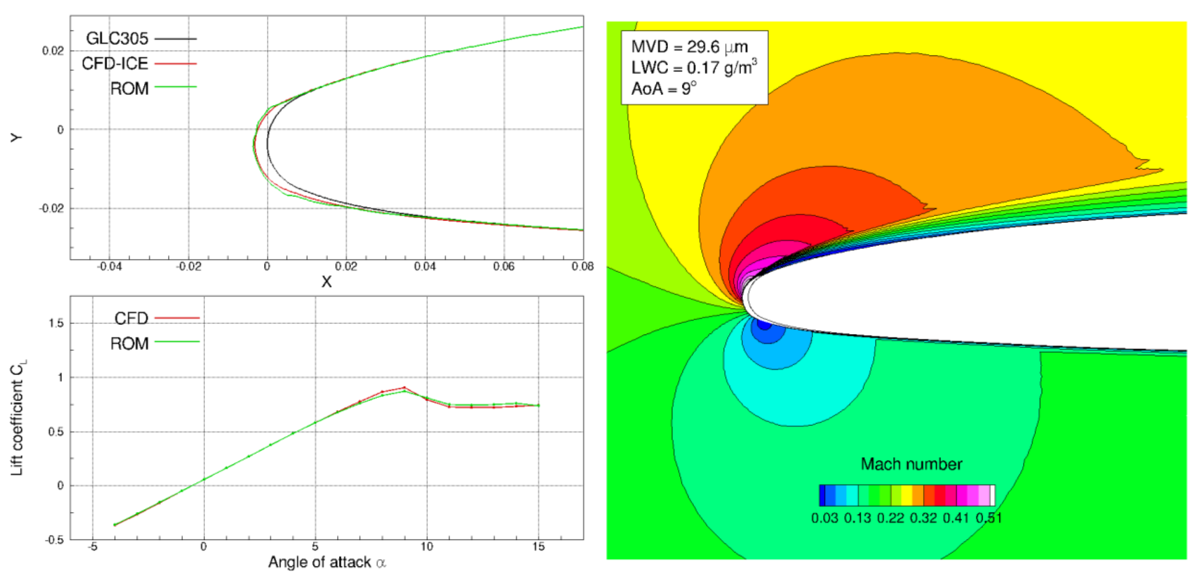


Figure 3-4: Rime ice and the associated C_L - α curve (left); Mach number contours (right).

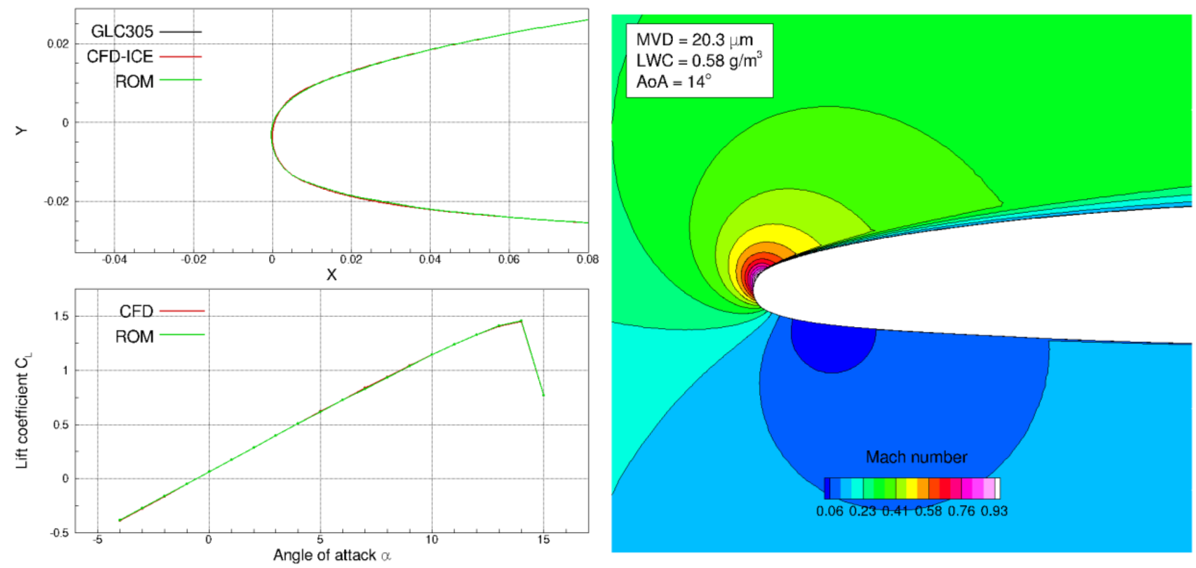


Figure 3-5: No-ice and the associated C_L - α curve (left); Mach number contours (right).

Figure 3-6 (left) illustrates the eigenvalue convergence plot for the shape of ice, represented by x and y coordinates, respectively. The Y-axis is the normalized energy content associated to

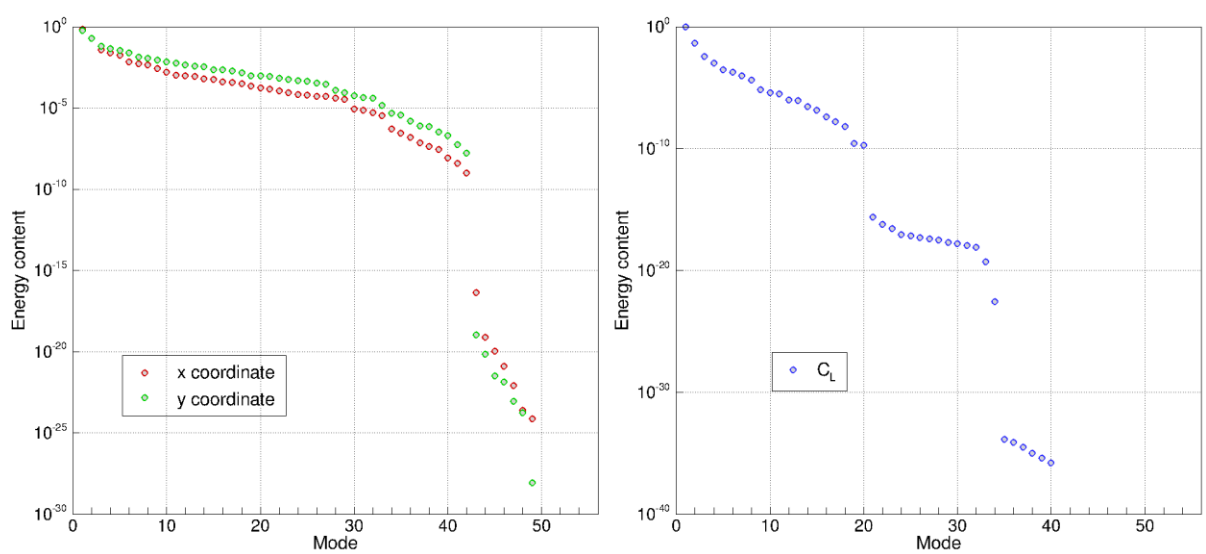


Figure 3-6: Eigenvalues vs. modes convergence for shape of ice (left) and C_L - α curve (right).

each POD mode, i.e. $\lambda_j / \sum_{i=1}^{Ns} \lambda_i$. The POD expansion can be truncated at a user-defined energy content, given that adding more modes will not significantly change the predicted target solution. In this analysis, an energy content of 99.9999% is selected, corresponding to 34 modes for x-coordinate and 38 modes for y-coordinate used in ice shape prediction. Figure 3-6 (right) illustrates the eigenvalue convergence plot for $C_L - \alpha$ curve, and 14 modes were used for the same energy content.

3.1.2 Iterative sampling based on global ROM

Figure 3-7 shows the LOOCV error distribution associated with the initial set of snapshots (the LOOCV errors at each snapshot location are interpolated to obtain error estimation everywhere in the parameter space). The highest error in terms of mass of ice is around 70 grams per unit span (which accounts for 32% of $mass_{ice,CFD,max}$), and the L_∞ error of $C_L - \alpha$ curve is around 47%. These errors were used to define the density function $\rho(\mathbf{x})$ in Equation (2.47) for the subsequent iteration of CVT sampling. The method of global POD was adopted in the first six iterations, namely all snapshots were considered together. Figure 3-8 illustrates the error distribution after six iterations consisting of 96 snapshots. Despite the application of the iterative sampling approach, the errors were not reduced but, instead, increased: the highest error of mass of ice has been increased to around 89 grams per unit span, and the L_∞ error of $C_L - \alpha$ curve is around 67%. This can be explained by the fact that using global POD for highly nonlinear problems, as more snapshots are added, more physical features are introduced into the snapshots set, and different types of solutions pollute each other. Moreover, it is worth noting that in Figure 3-8, for both types of snapshots, the highest errors lie in a narrow strip, which is the transition region between no-ice and glaze ice conditions. Such phenomenon clearly indicates that distinct physical features in this region require special treatment, and hence local POD is adopted.

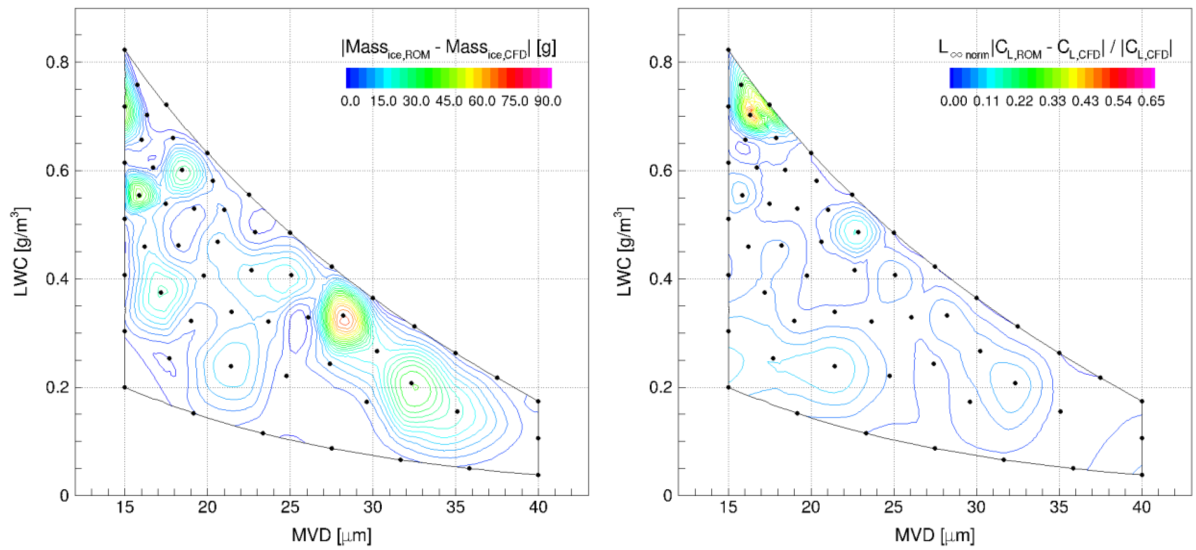


Figure 3-7: Global ROM LOOCV errors after the first sampling iteration: mass of ice (left); C_L (right).

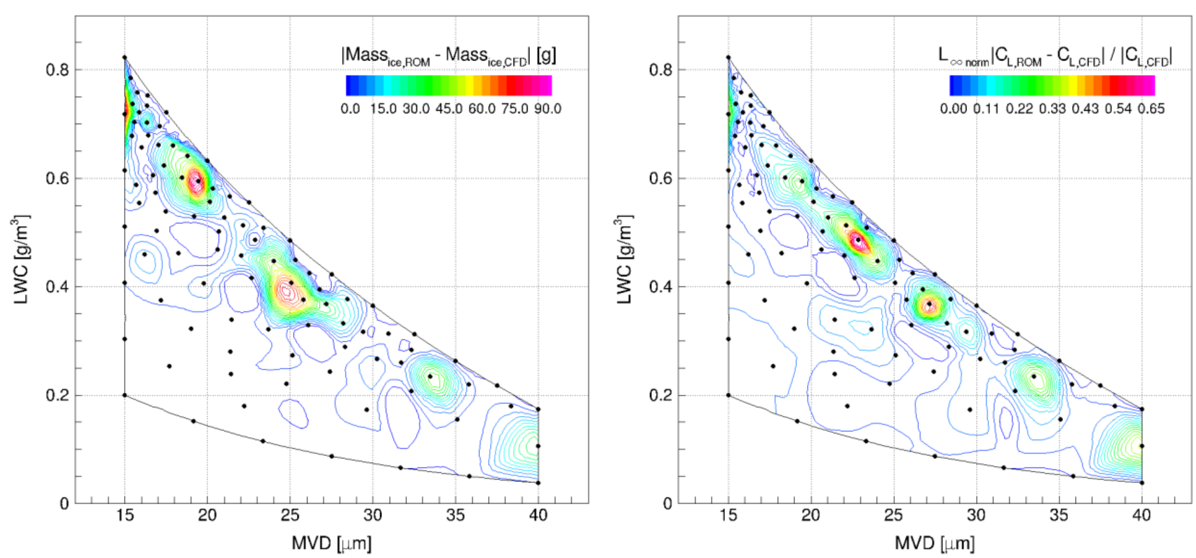


Figure 3-8: Global ROM LOOCV errors of the sixth sampling iteration: mass of ice (left); C_L (right).

3.1.3 Local ROM and error driven sampling

In order to define local ROB, the 96 snapshots are grouped into three subsets, as three types of icing solutions are expected. The clustering analysis was done based on $C_L - \alpha$ curves because their pollution had more prominent effects compared to those of ice shapes. The decision boundaries between clusters were determined via logistic regression, using a degree-3 polynomial feature mapping with a regularization factor of 1×10^{-2}, as shown in Figure 3-9 (left). Before doing the leave-one-out error evaluation on each cluster, the snapshots on the decision boundaries need to be computed such that each cluster is enclosed. As shown in Figure 3-9 (left), the initial set of boundary lines are marked by green dash lines. Between cluster 2 and 3, eight points were defined and the corresponding snapshots (ice shapes and $C_L - \alpha$ curves) were shared between these neighboring sub-regions, leaving no uncharted areas in the parameter space. Between cluster 1 and 2, nine points were defined on the decision boundary, however only required by cluster 2, since cluster 1 comprises ice-free solutions and no more computation needs to be done there. To find the exact location of iced vs. no-ice boundary, k-means clustering was applied on the 96 snapshots plus 9 boundary snapshots between clusters 1 and 2, and the new boundary identified by logistic regression is marked as red dash line in Figure 3-9 (left), where evident changes in terms of line location can be observed in the high MVD region. Another nine boundary points (red circles), as well as the snapshots, were defined and a third boundary marked as blue dash line is obtained by repeating the same procedure on 96 plus 18 boundary snapshots, and this time the change is very limited. After a few more iterations, it was shown that the decision boundary eventually converged to a specific position in the icing envelope that identifies precisely the no-ice zone. Note that the 8 boundary snapshots between clusters 2 and 3, once determined, were excluded from the following clustering analysis in the search for iced vs. no-ice boundary, therefore, the boundary line between clusters 2 and 3 remains unchanged.

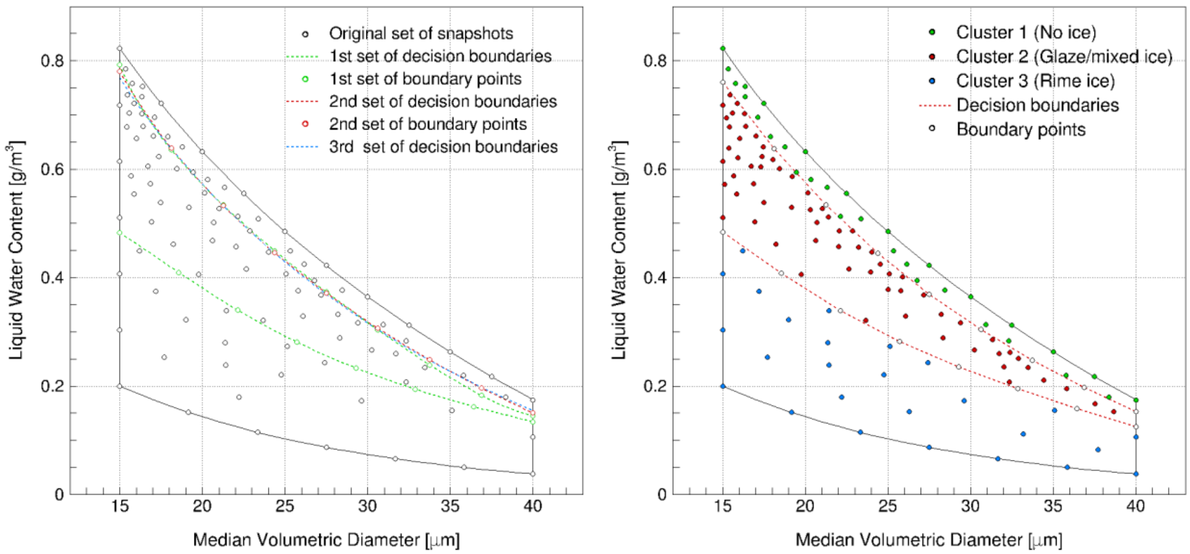


Figure 3-9: Decision boundaries identified by supervised learning (left); snapshots and clustering after 12 iterations (right).

Once the boundaries were established, error driven sampling was continued on each cluster. At the end of the iterative sampling, 140 snapshots were obtained, partitioned into 3 clusters for a final state of 3 local reduced order bases (Figure 3-9 right). It is also worth noting that the clustering based on the aerodynamic solution implicitly identifies the different physical regimes in the parameter space, i.e. in the case of icing simulation, it is capable of identifying a no-ice region (cluster 1), a glaze-mixed ice region (cluster 2) and a rime ice region (cluster 3) in the icing envelope.

Figure 3-10 (left) shows the error distribution for mass of ice for the last iteration. The maximum error has been reduced to 47 grams per unit span (which accounts for 21% of $mass_{ice,CFD,max}$), mainly in the transition region where the tiniest amount of ice starts to accumulate, while for the majority of the CM icing envelope, the error is less than 10 grams per unit span. Figure 3-10 (right) illustrates the overall mass of ice accumulation throughout the CM icing envelope. This result is obtained from 3,000 ROM solutions, which are uniformly distributed in the parameter space. Each target condition is sorted into a corresponding cluster by the classifier trained via logistic regression. Then, the specific ROB from that cluster is used to build the reduced solution. As shown in the graph, the accumulated mass of ice can reach a maximum of 220 grams per unit span, mainly in the region where total temperature is close to the freezing point and large amounts of liquid water are conducive to the formation of larger amount of ice. This mass of ice distribution could be helpful for the design of an ice protection system to manage optimum energy requirements for different icing conditions. In this test case, the online cost of computing each ROM solution on a desktop computer takes 0.13 second using a single CPU, while the computational time of each CFD-icing solution is 9 hours using 16 CPUs on a supercomputer (offline cost). Detailed information about the computational cost is summarized in Table 3-1.

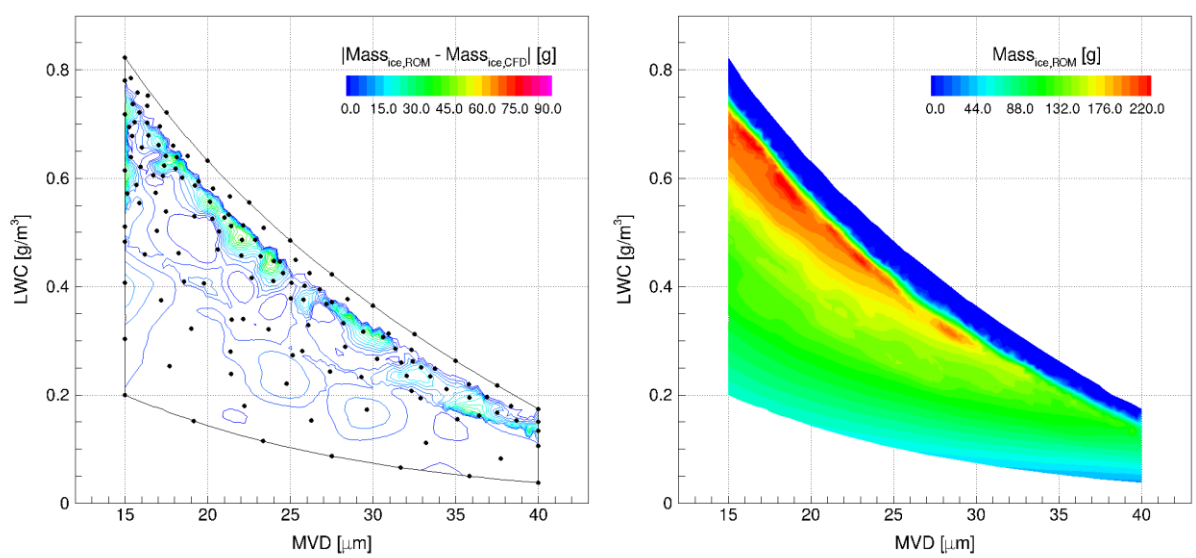


Figure 3-10: Local ROM LOOCV error after 12 iterations (left); mass of ice variation obtained from 3,000 ROM solutions (right).

Figure 3-11 (left) demonstrates the error distribution in terms of $C_L - \alpha$ curves for the last iteration. The maximum error has been reduced to 13.3%, while for the majority of the icing envelope the error is less than 3%. This level of accuracy is considered to be a good compromise given the cost necessary to get the snapshots. Figure 3-11 (right) illustrates the penalty in maximum lift coefficient due to ice accretion. This result is again obtained from 3,000 ROM solutions based on 140 pre-computed CFD snapshots. The decrease in $C_{L,max}$ can reach 50% for a considerably large region of the icing envelope, which is detrimental to flight safety in an icing scenario. In this case, the online cost for each ROM solution takes 0.083 second on a single CPU, while the computational time of each CFD solution is 12 hours on 32 CPUs (Table 3-1). The near real-time performance obtained in this case supports the possibility of incorporating local ROM technology into flight simulators to enable realistic in-flight icing scenarios during pilot training.

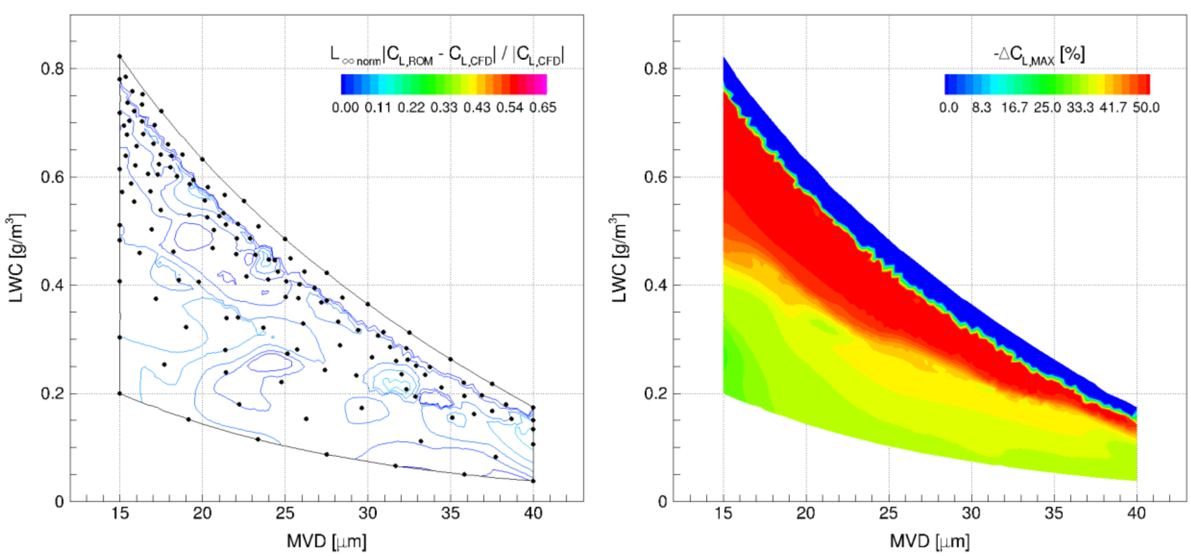


Figure 3-11: Local ROM LOOCV error after 12 iterations (left); aerodynamic degradation in terms of loss of $C_{L,max}$ obtained from 3,000 ROM solutions (right).

Table 3-1: Summary of computational cost (CM exploration)

	Process		Computational cost
Offline	CFD-icing simulations	Each snapshot	9 h on 16 CPUs* (ice shape) 12 h on 32 CPUs* ($C_L - \alpha$ curve)
	Iterative sampling	Each LOOCV; each CVT sampling	1.5 ~ 3 h on a single CPU** (depending on the number of snapshots) 0.5 ~ 1 min on a single CPU**
	Machine learning	*K*-means clustering, logistic regression (defining decision boundaries)	3 ~ 5 seconds on a single CPU**
	Build database	Extract POD modes from the final set of snapshots, compute Kriging model parameter	14 min on a single CPU** (ice shape) 2.5 min on a single CPU** ($C_L - \alpha$ curve)
Online	Solve target	Logistic regression (classification), linear combination of POD modes	0.13 s on a single CPU** (ice shape) 0.083 s on a single CPU** ($C_L - \alpha$ curve)

* Intel Xeon E5462 quad-core, 2.8 GHz (supercomputer Colosse)

** AMD Phenom II X6 1075T Processor, 800 MHz (desktop computer)

3.1.4 Comparison of local vs. global ROM

The advantages of local ROM versus global ROM for highly nonlinear problems are clearly shown in Figure 3-12. The upper graph demonstrates the mass of ice error reduction history throughout 12 iterations. Within the first 6 iterations that global ROM was adopted, the maximum error did not decrease due to pollution of distinctive physical characteristics of the solutions. In the following 6 iterations, local ROM was adopted and the maximum error of mass of ice was reduced by 50%. The effectiveness of local ROM is much more evident in the case of $C_L - \alpha$ curves. As can be seen, the maximum error has been reduced to 13% by local ROM, as opposed to 67% for global ROM. In this comparison local and global ROMs are not using the same number of snapshots, however it has been demonstrated that global ROM failed to yield reasonable accuracy by adding more snapshots, and it is for this reason that local ROM was considered.

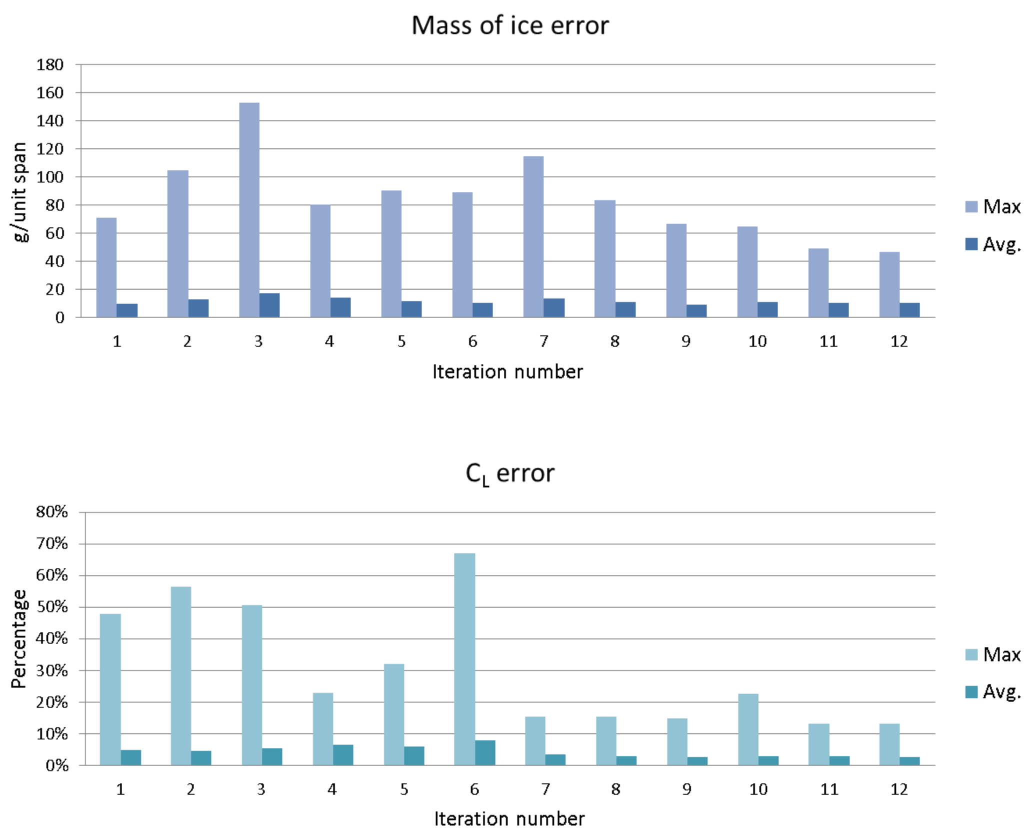


Figure 3-12: Error reduction history throughout iterative samplings: mass of ice (upper); C_L (lower).

3.2 Intermittent maximum

The effectiveness of local reduced-order modeling demonstrated in the test case of CM supported the exploration of the IM envelope with local ROM only. The exposure time considered is 54 seconds, based on the standard horizontal distance specified in Appendix C (2.6 nautical miles) and the selected TAS (175 knots). The ice shape snapshots are obtained via a 3-

shot CFD-icing simulation, with uniform shot timings. One should note that IM (thunderstorms) conditions are difficult to find and dangerous to test in nature. Very often, simulation means are accepted in evaluating these conditions.

3.2.1 Determination of local ROM

The initial sampling consists of 56 points (Figure 3-13). Again, it is not uniform but biased towards the high temperature and high LWC region, for the same reason as explained in the CM exploration section. To implement local ROM from the very beginning, *k*-means clustering was conducted in the solution space, grouping 56 snapshots into 3 subsets (Figure 3-14). Decision boundaries (green dashed lines) were obtained via logistic regression using a degree-4 polynomial feature mapping with regularization factor 1×10^{-4}. Nineteen (19) points were defined on the decision boundaries (green hollow circles) and the corresponding snapshots were computed. However, due to the relatively sparse sampling of the first iteration, by running *k*-means clustering and LR again on all available snapshots (initial 56, plus 19 boundary points), a

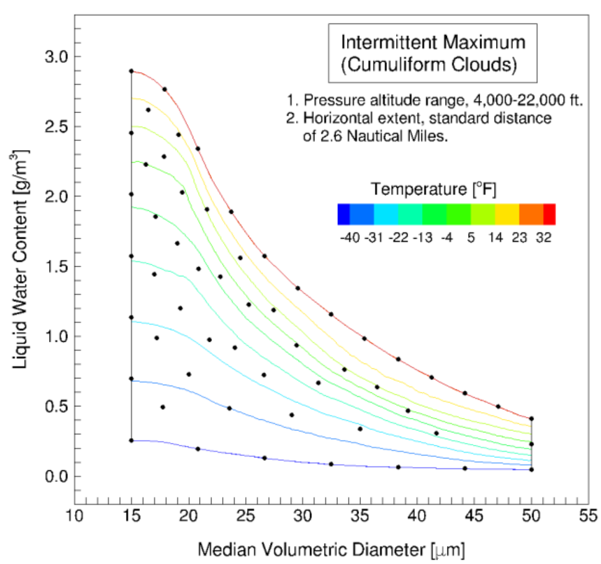


Figure 3-13: Initial sampling of the IM.

second set of decision boundaries were obtained (red dashed lines) that are quite different from the first set. To determine local reduced order models that better identify different physical regions in the parameter space, an additional 19 points (red hollow circles) were defined on the new decision boundaries and the corresponding snapshots were computed. To assess the change in the decision boundaries, *k*-means clustering and LR were implemented on all 94 snapshots (75 plus 19 new), and the third set of decision boundaries, marked by blue dashed lines, were identified very close to the red dashed ones representing the second set of decision boundaries. Although precise location of boundaries can be obtained by continuing this iterative approach, repeated computation could be expensive. As a trade-off between computational cost and the precise subdivision of parameter space, the second set of decision boundaries were adopted to define subregions for the construction of local ROM, as the maximum difference in LWC between the second and third sets of decision boundaries is 6.06×10^{-2} g/m^3, and the averaged difference is 1.74×10^{-2} g/m^3, leading to an insignificant influence on the final results.

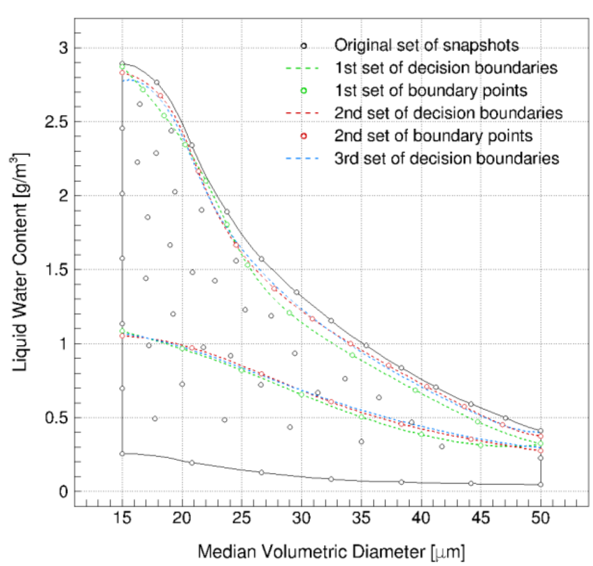


Figure 3-14: Decision boundaries defined by supervised learning.

3.2.2 Error driven iterative sampling

Figure 3-15 shows the error distribution associated with the second set of decision boundaries, obtained from local ROM. As can be seen, the highest error of mass of ice is around 58 grams per unit span (which accounts for 29% $mass_{ice,CFD,max}$), and the L_{∞} error of $C_L - \alpha$ curve is around 18%. In total, eleven iterations were conducted resulting in 119 snapshots, as shown in Figure 3-16.

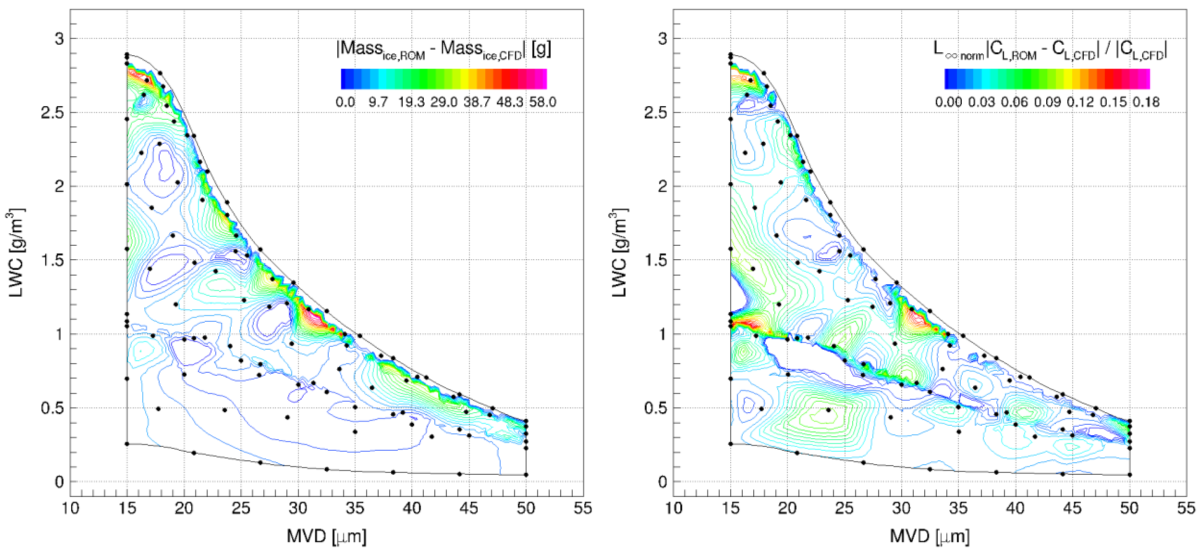


Figure 3-15: Local ROM LOOCV errors on 94 snapshots: mass of ice (left); C_L (right).

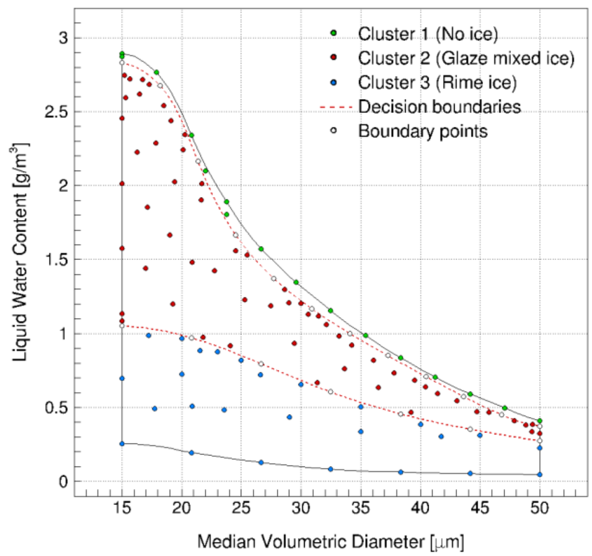


Figure 3-16: Snapshots and clustering after 11 iterations.

Figure 3-17 (left) reports the error distribution in the mass of ice for the last iteration (contours obtained from interpolation of the LOOCV errors at each snapshot location). The maximum error has been reduced to 44 grams per unit span, while for the majority of the IM icing envelope, the error is less than 10 grams per unit span. Figure 3-17 (right) illustrates the overall mass of ice accumulation throughout the IM. This result is obtained by using 3,000 ROM solutions based on the 119 pre-computed CFD snapshots. The accumulated mass of ice can reach 200 grams per unit span, mainly in the region where total temperature is close to freezing and large amounts of liquid water are conducive to the formation of larger amount of ice. In this test case, each ROM solution takes 0.30 second on a single CPU, while the computational time of each CFD-icing solution is 8 hours on 16 CPUs as illustrated in Table 3-2.

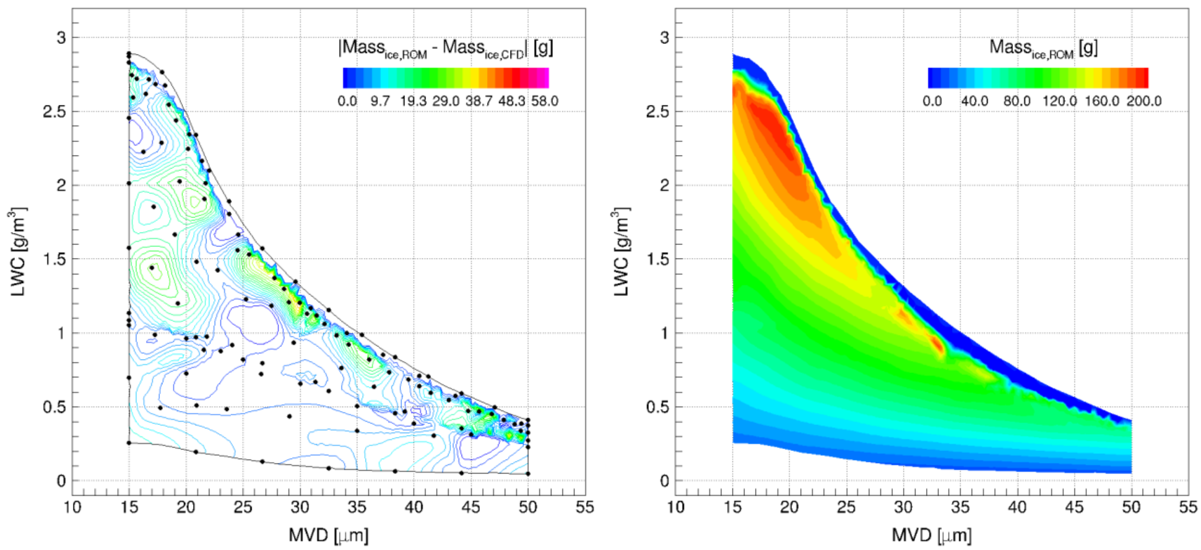


Figure 3-17: Local ROM LOOCV error after 11 iterations (left); mass of ice variation obtained from 3,000 ROM solutions (right).

Figure 3-18 (left) demonstrates the error distribution in terms of $C_L - \alpha$ curves for the eleventh iteration with 119 snapshots, with a maximum error less than 12%. Figure 3-18 (right) illustrates the change in maximum lift coefficient due to ice accretion. This result is again obtained by using 3,000 ROM solutions based on the 119 pre-computed CFD snapshots. The loss of $C_{L,max}$ can be as high as 50% for a considerably large region in the icing envelope, resulting in adverse impact on aerodynamic performance. In this case, each ROM solution takes 0.30 second on a single CPU, while the computational time of each CFD solution is 12 hours on 32 CPUs. Details can be found in Table 3-2.

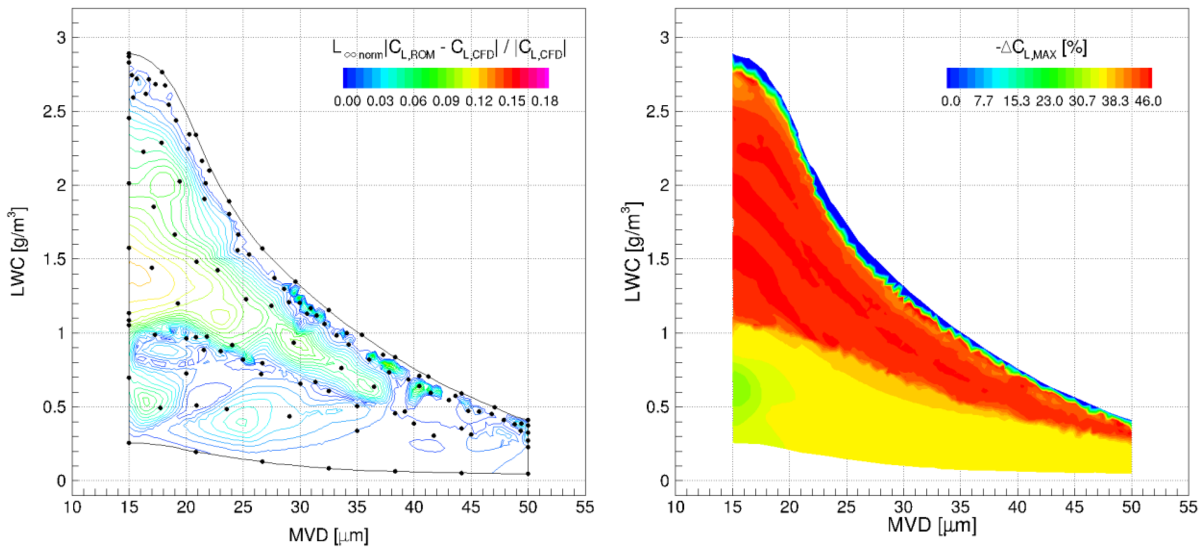


Figure 3-18: Local ROM LOOCV error after 11 iterations (left); aerodynamic degradation in terms of loss of $C_{L,max}$ obtained from 3,000 ROM solutions (right).

Table 3-2: Summary of computational cost (IM exploration)

	Process		Computational cost
Offline	CFD-icing simulations	Each snapshot	8 h on 16 CPUs* (ice shape) 12 h on 32 CPUs* ($C_L - \alpha$ curve)
	Iterative sampling	Each LOOCV; each CVT sampling	1 ~ 2 h on a single CPU** (depending on the number of snapshots) 0.5 ~ 1 min on a single CPU**
	Machine learning	*K*-means clustering, logistic regression (defining decision boundaries)	3 ~ 5 seconds on a single CPU**
	Build database	Extract POD modes from the final set of snapshots, compute Kriging model parameter	6 min on a single CPU** (ice shape) 1 min on a single CPU** ($C_L - \alpha$ curve)
Online	Solve target	Logistic regression (classification), linear combination of POD modes	0.3 s on a single CPU** (ice shape) 0.3 s on a single CPU** ($C_L - \alpha$ curve)

* Intel Xeon E5462 quad-core, 2.8 GHz (supercomputer Colosse)

** AMD Phenom II X6 1075T Processor, 800 MHz (desktop computer)

The error reduction history throughout 11 iterations for both mass of ice and lift coefficient curve is shown in Figure 3-19. Local ROM was adopted through all eleven iterations. The experience obtained through exploration of the IM suggests that the initial sampling should not be too sparse so as to avoid large movement of decision boundaries, which requires computation of additional boundary snapshots. Although there are positions to which the decision boundaries will eventually converge, a stopping criterion should be specified to avoid excessive computational cost, while still maintaining reasonable accuracy.

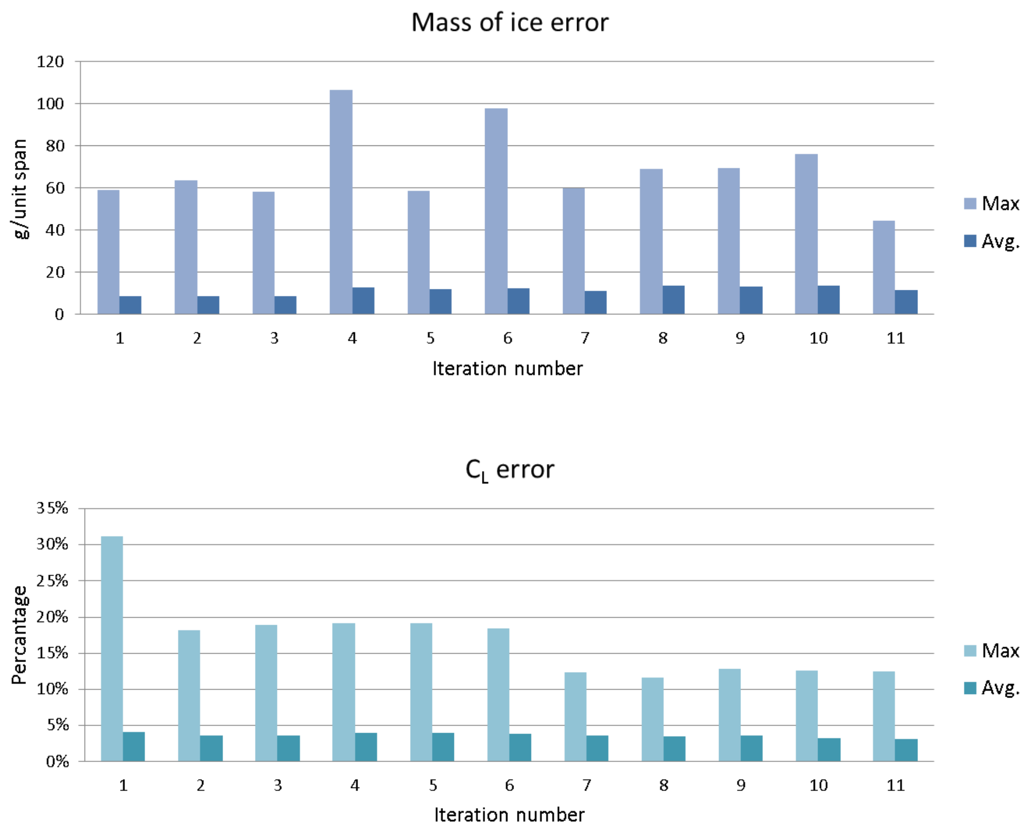


Figure 3-19: Error reduction history throughout iterative samplings: mass of ice (upper); C_L (lower).

3.3 Discussion

There are two open questions about the proposed methodologies worth further investigation. The first open question regarding iterative sampling involves the number of new snapshots that should be introduced within each iteration. The density function in CVT sampling is defined by the errors obtained from LOOCV, but the number of new snapshots to be added is rather empirical. While it would be more straightforward to add one snapshot at a time, in the highest error region, efficiency is diminished since the computation of several snapshots can be run simultaneously on a supercomputer. Another open issue is the optimal number of clusters. In this paper, three clusters were proposed since there are three distinct classes of ice formations. For other problems, the optimal number of clusters may not be easily determined a priori, but should be chosen to properly represent the physics of the problem while remaining mathematically tractable.

CHAPTER 4 THREE-DIMENSIONAL ANALYSIS

This chapter extends the assessment of the ROM framework to the complete exploration of the FAA continuous maximum (CM) icing envelope for a regional jet (RJ). Machine learning algorithms are used to address the clustering of snapshots and delimit ice-type regions within the envelope. For aerodynamic degradations resulting from airframe ice accumulation, detailed parametric analysis is performed on varied flight conditions. The greedy algorithm and CVT are combined in an iterative framework to position snapshots in the regions of high nonlinearity, determining a judicious balance between accuracy and the number of snapshots.

4.1 Exploration of the continuous maximum icing envelope

The proposed local ROM, coupled with the iterative sampling methodology, are applied to the problem of in-flight icing certification, to estimate ice build-up on unprotected aircraft surfaces during a holding pattern in a 17.4 nautical miles region of the CM icing conditions. A RJ with a 2.93 meter mean aerodynamic chord has been considered in this work (Figure 4-1). Airspeeds and altitudes have been selected to represent a typical holding condition, e.g. pressure altitude of 5,000 meters with a true air speed of 268 knots at an AoA of 3.7°. The exposure time considered is 25 minutes, a preliminary exploration for the maximum 45-minute hold certification requirement. The purpose is to demonstrate that ice shapes and mass within the CM icing envelope can be completely explored via ROM, not only for sections of the wing but for the entire aircraft.

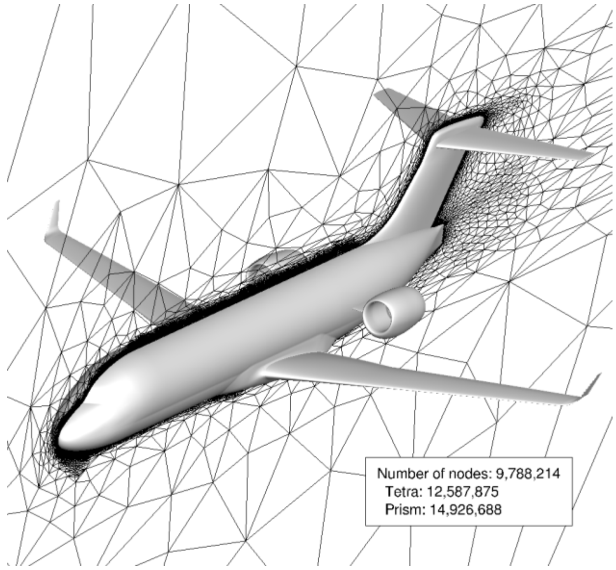


Figure 4-1: Geometry and mesh of a generic RJ.

Given the complexity of the RJ geometry, mesh adaptation was used to improve mesh/solution quality and convergence of the flow solver. A hybrid mesh of 9,788,214 nodes, 14,926,688 prism elements and 12,587,875 tetrahedral elements was used to provide the necessary CFD-icing snapshots. Ice accretion was done in one-shot for 25 minutes, with the availability of multi-shot ice accretion when and if more precision is needed.

For the analysis of the ice shapes, the snapshots $\boldsymbol{U}_i$, are the Cartesian coordinates of the 202,260 nodes defining the surface mesh. As a result of ice accretion, the surface mesh is displaced from the original clean surface. The accuracy of the ROM predictions versus CFD results is evaluated via the LOOCV. For each snapshot $\boldsymbol{U}_i$, the vector of the errors ε_j^i at each node j of the surface mesh is defined as the difference of ice thickness δ_j^i at that mesh point, namely

$$\varepsilon_j^i = \left|\delta_{j,ROM}^i - \delta_{j,CFD}^i\right|, i = 1, \dots, N_S, j = 1, \dots, N_P. \tag{4.1}$$

Then for any location i in the parameter space, the LOOCV error is expressed as the mean square root of the error vector

$$\varepsilon_{ice}^{i} = \sqrt{\frac{\sum_{j=1}^{N_P}\left(\varepsilon_j^i\right)^2}{N_{P,iced}}}, i = 1, \dots, N_S, j = 1, \dots, N_P, \tag{4.2}$$

where $N_{P,iced}$ is the number of surface nodes displaced due to ice accretion. These errors were used to define the density function in CVT for the subsequent sampling iteration, which identifies new samples/snapshots to be added in the high error region.

In Chapter 3 it is shown that three typical types of ice snapshots exist within the CM: no ice (clean geometry) or trace ice (ice becomes perceptible), glaze/mixed ice, and rime ice. These different types of solutions could numerically pollute each other in the context of global POD, therefore local POD is employed to handle these distinct solutions by subdividing the icing envelope into three sub-regions. In this work, local ROM is validated on a 3D geometry; detailed analysis will be given in the remainder of this section.

4.1.1 Initial sampling and the snapshots

The initial sampling consists of 36 points (Figure 4-2), each representing a different icing condition in terms of MVD, LWC and, implicitly, external temperature. Figure 4-3 to Figure 4-5 illustrate examples of typical snapshots, representing glaze ice, rime ice and trace ice, respectively. These three figures demonstrate ice thickness contours obtained for the specific icing condition, as well as ice shape comparisons of ROM solution versus CFD solution at different sections of the airplane, during LOOCV at this specific snapshot location, i.e. the ROM solution is obtained based on the remaining 35 snapshots.

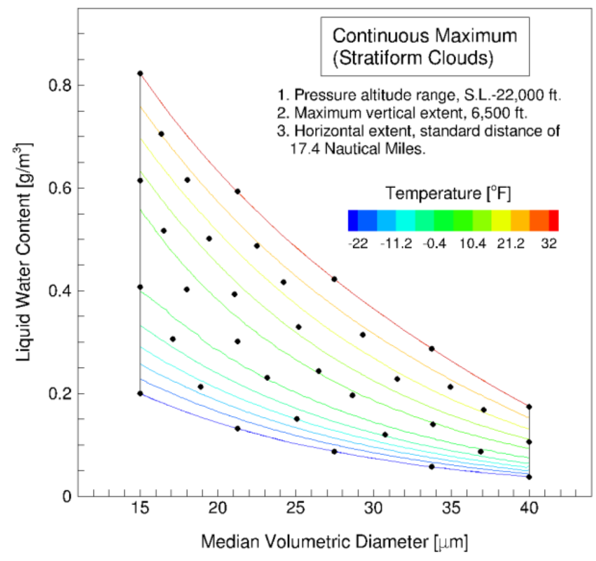


Figure 4-2: Continuous maximum icing envelope, with initial sampling.

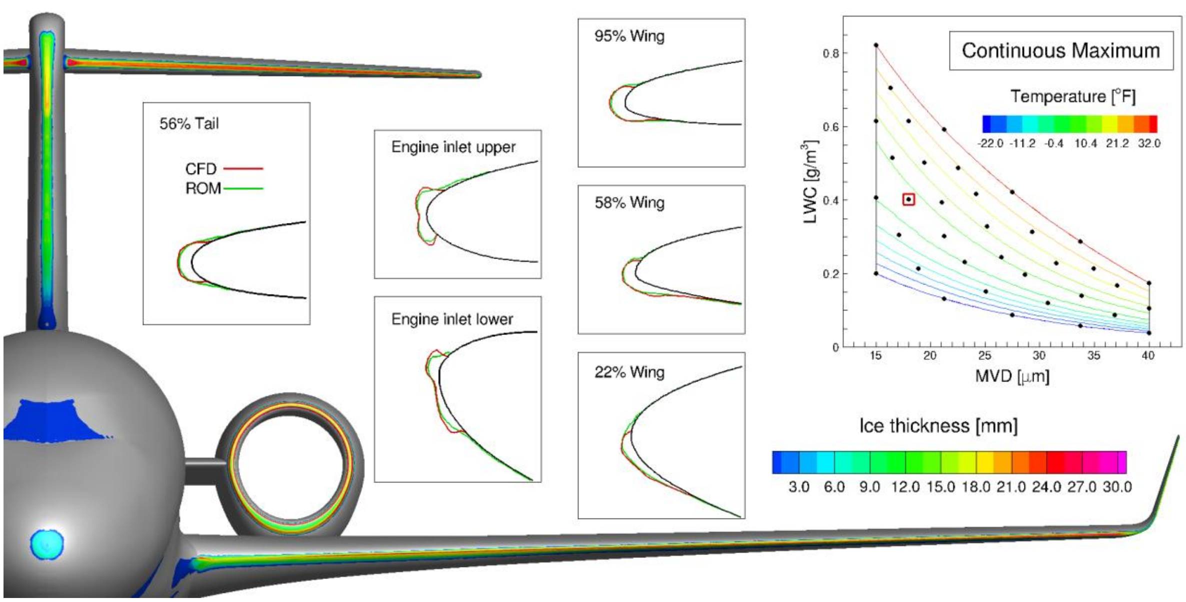


Figure 4-3: Contours of ice thickness, along with ice shape comparisons between ROM and CFD, glaze ice.

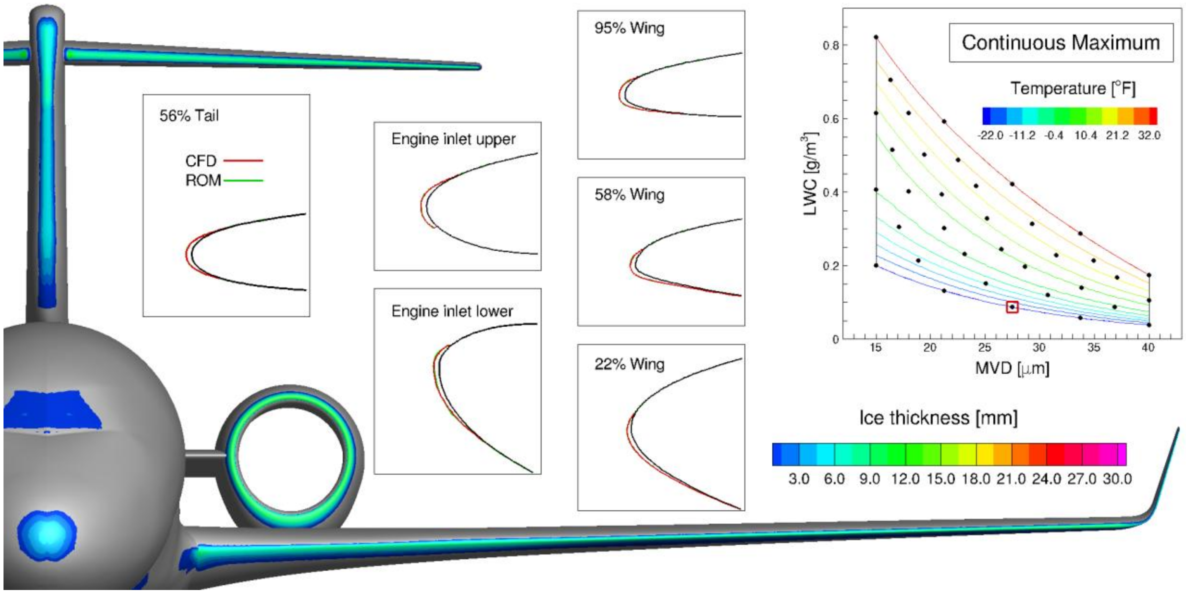


Figure 4-4: Contours of ice thickness, along with ice shape comparisons between ROM and CFD, rime ice.

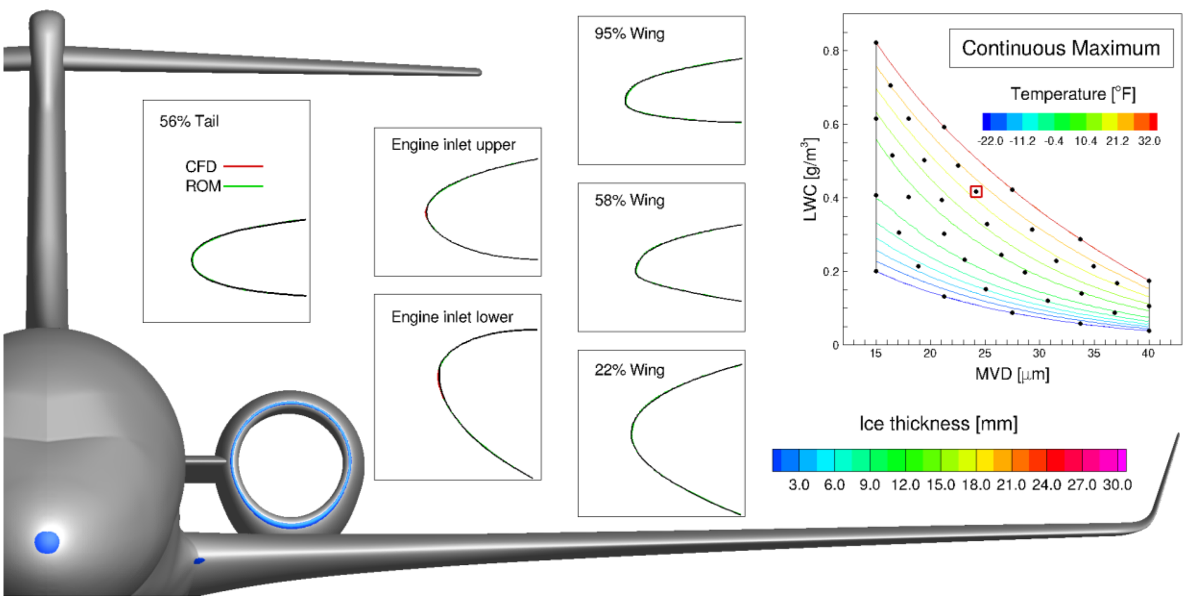


Figure 4-5: Contours of ice thickness, along with ice shape comparisons between ROM and CFD, trace or no ice.

Figure 4-6 illustrates the eigenvalue convergence plot for shape of ice, represented by x, y and z coordinates, respectively. The Y-axis is the normalized energy content associated to each POD mode, i.e. $\lambda_j / \sum_{i=1}^{Ns} \lambda_i$. The POD expansion can be truncated at user defined energy content, given that adding more modes will not significantly change the predicted target solution. In this analysis, an energy content of 99.9999% is selected, corresponding to 26 modes for x coordinate, 28 modes for y coordinate, and 25 modes for z coordinate used in ice shape prediction.

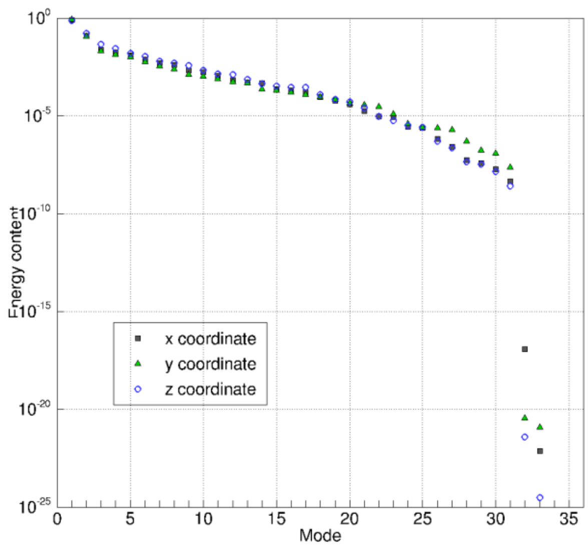


Figure 4-6: Eigenvalues vs. modes convergence for shape of ice.

4.1.2 Error driven sampling and local ROM

The experience obtained from the 2D analysis (Chapter 3) suggests that an adequate number of snapshots should be collected before the subdivision of the parameter space. Since no a priori knowledge exists as to where the most critical region (glaze/mixed ice) is located within the envelope, it is preferable to start from a smaller number of uniformly distributed snapshots, and iteratively enrich them according to a suitable error indicator.

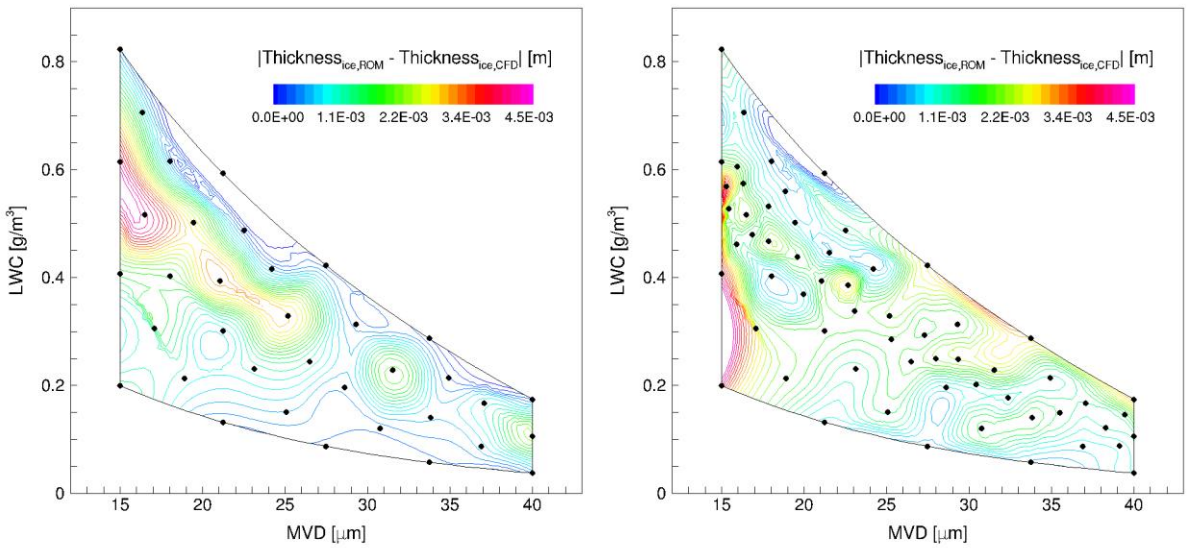


Figure 4-7: Global ROM LOOCV errors: the first sampling iteration with 36 snapshots (left); the fourth sampling iteration, with 60 snapshots (right).

Figure 4-7 (left) shows the LOOCV error distribution associated with the initial set of snapshots (the LOOCV errors at each snapshot location are interpolated to obtain error estimation everywhere in the parameter space). The highest error in terms of ice thickness is 4.4 mm (which accounts for 15% $\delta_{ice,CFD,max}$). Figure 4-7 (right) illustrates the LOOCV error after the fourth iteration, with 60 snapshots. As more snapshots are added, the error has not been further reduced, but increased. This is a clear signal that global POD is not adequate for this type of analysis, since snapshots with distinct features start to pollute each other, therefore these different features need to be treated separately, namely using local POD.

Figure 4-8 (left) shows the snapshots and clustering after 6 sampling iterations, with 80 snapshots. The clustering analysis was done based on ice thickness over the aircraft. The 80 snapshots are grouped into three subsets: no-ice or trace ice (cluster 1), glaze/mixed ice (cluster 2) and rime ice (cluster 3). The decision boundaries between clusters are determined via logistic regression, using a degree-4 polynomial feature mapping with a regularization factor 1×10^{-4}, as shown in Figure 4-8 (left). Before doing the leave-one-out error evaluation for each cluster, the snapshots on the decision boundaries need to be computed such that each cluster is totally enclosed. Therefore, ten points were defined on the decision boundary between clusters 1 and 2, and six points were defined on the decision boundary between clusters 2 and 3. These corresponding snapshots were shared between the neighboring subregions, leaving no untagged areas in the parameter space. The LOOCV results for the 3 clusters are shown in Figure 4-8 (right). Comparing with global POD (Figure 4-7), the accuracy has been greatly improved.

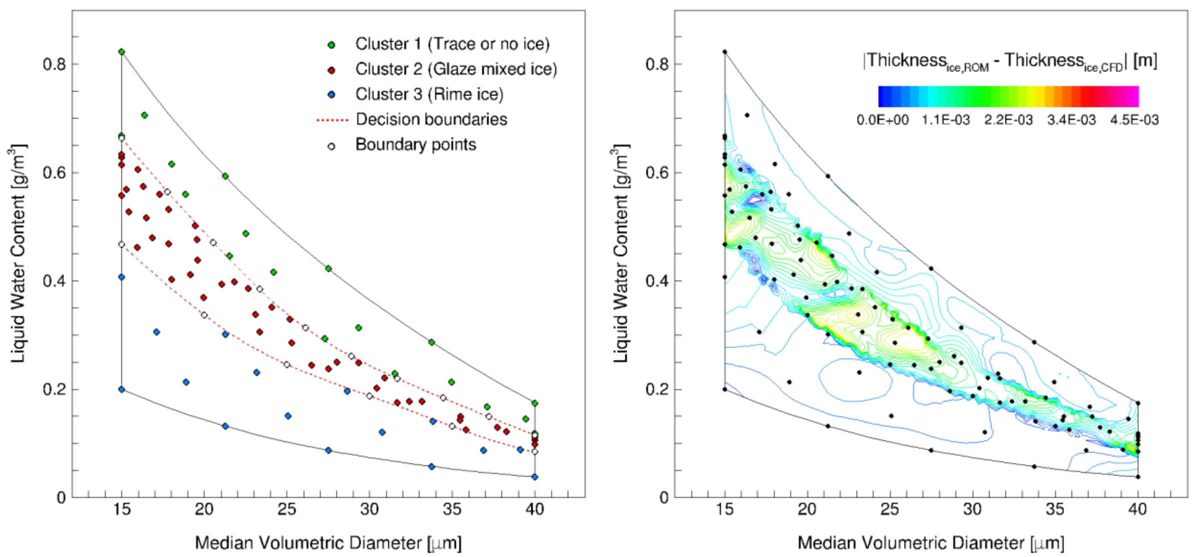


Figure 4-8: Snapshots and clustering after 6 iterations, with 80 snapshots (left); local ROM LOOCV error after 6 iterations (right).

Once the boundaries are established, error driven sampling is continued on each cluster. At the end of the iterative sampling, 103 snapshots are obtained, partitioned into 3 clusters, for a final state of three local ROB (Figure 4-9 left). Figure 4-9 (right) shows the ice thickness error distribution for the last iteration. The maximum error has been reduced to 2.7 mm (which accounts for 9% $\delta_{ice,CFD,max}$), a 40% deduction of the maximum error obtained from the initial sampling. For the majority of the icing envelope, the error is less than 1.2 mm.

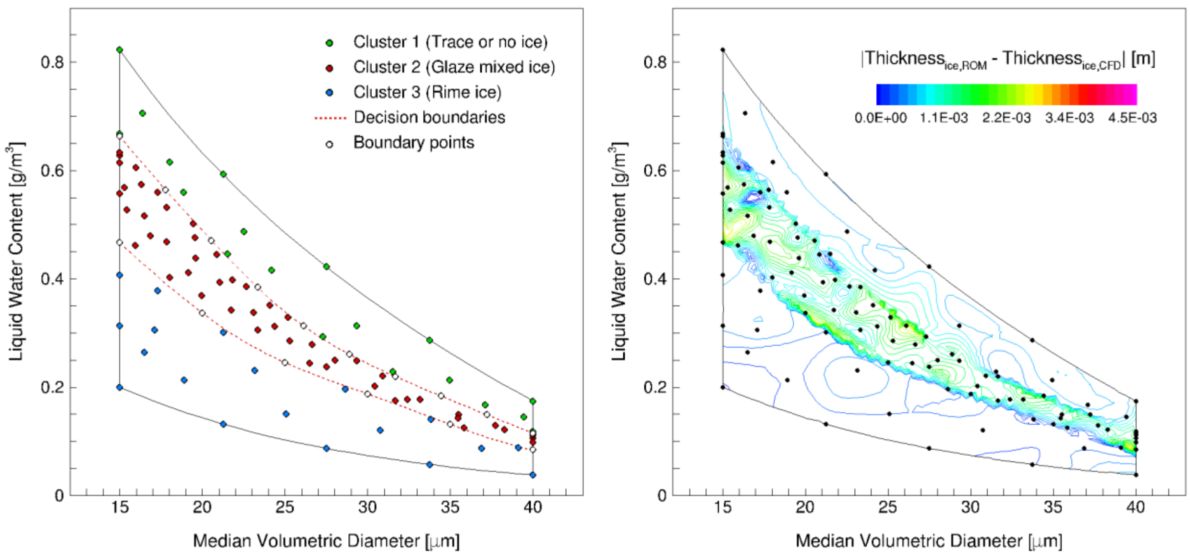


Figure 4-9: Snapshots and clustering for the last sampling iteration, with 103 snapshots (left); local ROM LOOCV errors for the last sampling iteration (right).

4.1.3 Complete exploration of the CM

To explore the CM, 1,000 icing conditions are uniformly sampled in the parameter space, with each target condition sorted into a corresponding cluster by the classifier trained via LR (Figure 4-10 left). Then, the specific ROB from that cluster is used to build the reduced solution. With the displaced surface mesh, the volume of accreted ice can be determined by calculating the volume of the space enclosed by the iced surface and the clean geometry. TetGen [74] is adopted to fill this space with a tetrahedral mesh, from which the total volume can be directly calculated. The mass of the ice can then be determined by multiplying the volume with density of ice (assumed constant at 917 kg/m^3). Figure 4-10 (right) illustrates the overall mass of ice accumulation throughout the CM icing envelope.

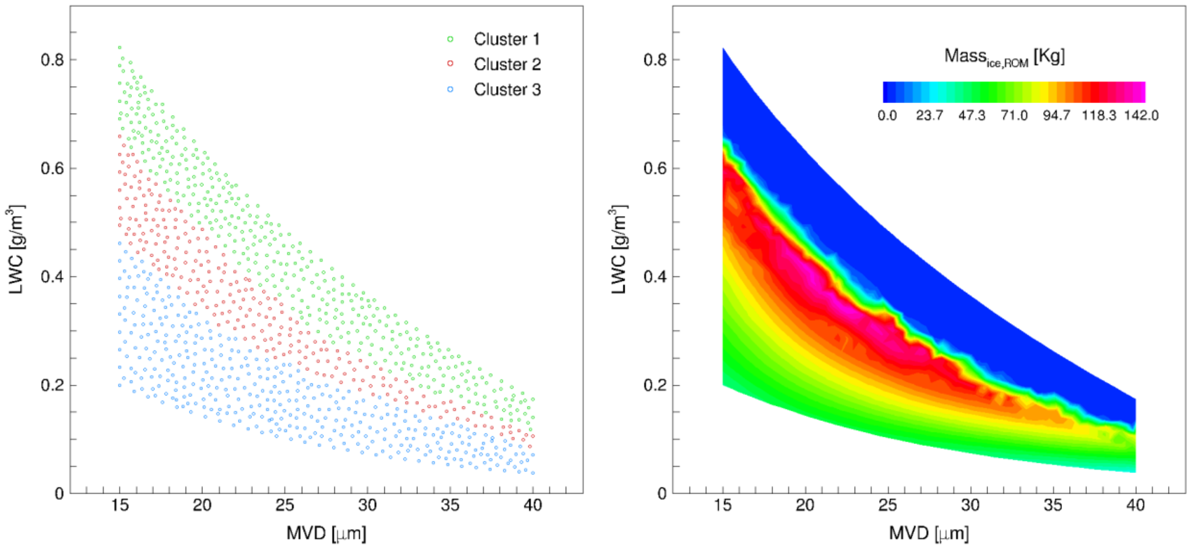


Figure 4-10: Classification of 1,000 target conditions (left); mass of ice variation obtained from 1,000 ROM solutions (right).

As shown in the graph (Figure 4-10 right), the accumulated mass of ice can reach a maximum of 142 kilograms, mainly in the region where total temperature is close to the freezing point and large amounts of liquid water are conducive to the formation of a larger amount of ice. This mass of ice distribution could be helpful for rapidly and accurately determine the critical mass in the design of ice protection systems and thus manage optimum energy requirements for all icing conditions. As shown in Table 4-1, in the current test case for a complete aircraft, the computational time of each CFD-icing solution is 16 to 32 hours using 128 CPUs, and is the offline cost. The online cost of computing each ROM solution takes 1.4 seconds using 6 CPUs.

Table 4-1: Computational cost for CM exploration for the RJ

	Process		Computational cost
Offline	CFD-icing simulations	Each snapshot	16 ~ 32 h on 128 CPUs[*] (depending on convergence rate)
	Iterative sampling	Each LOOCV;	1 ~ 2 h on 6 CPUs[**] (depending on the number of snapshots)
		each CVT sampling	1 min on a single CPU[**]
	Machine learning	*K*-means clustering, logistic regression (defining decision boundaries)	10 ~ 15 seconds on a single CPU[**]
	Build database	Extract POD modes from the final set of snapshots, compute Kriging model parameter	22 min on 6 CPUs[**]
Online	Solve target	Logistic regression (classification), linear combination of POD modes	1.4 s on 6 CPUs[**]

[*] Intel Xeon E5-2670 eight-core (supercomputer Guillimin)

[**] AMD Phenom II X6 1075T Processor, 800 MHz (desktop computer)

4.2 Aerodynamic degradation of an ice-contaminated regional jet

After holding in CM icing conditions for a certain time, ice build-up will adversely affect aerodynamic performance. It is important to reassess performance during subsequent aerodynamic maneuvers such as descent or aborted landing, for which pilots need to rapidly pull up and climb to a newly assigned altitude, afterwards, either prepare for re-landing, or divert to an alternate airport.

The purpose of this section is to demonstrate that, based on a set of pre-computed flow solutions under various flight conditions taken as snapshots, flow details in terms of pressure distribution and shear stress distribution (which are the major sources of lift and drag) for any untried flight condition can be obtained via ROM. In this work, the focus is on the analysis of steady longitudinal flight performance of a RJ. The resultant data can be incorporated into flight simulators, enabling icing scenario training for the pilots to better understand the hazards of in-flight icing, particularly in holding, descent or an aborted landing.

The ice shape considered is obtained from a 45-minute exposure under a CM icing condition (MVD = 21.26 μm, LWC = 0.30 g/m^3), at typical holding flight conditions (5,000 meters pressure altitude, 268 knots TAS and 3.7° AoA), with clean aircraft configuration (flaps/slats retracted, no deflection of control surfaces) (Figure 4-11). A hybrid mesh with 9,788,214 nodes, 14,926,688 prism elements and 12,587,875 tetrahedral elements is considered. For the aerodynamic analysis in question, the snapshots $\boldsymbol{U}_i$ are the flow variables of interest: pressure at the 9,788,214 nodes of the volume mesh and shear stress (x, y, z components) at the 202,260 surface nodes. Figure 4-12 gives an example of the flow solution over this ice-contaminated RJ, under flight condition IAS = 230 kt, AoA = 9° and PA = 3,000 ft. As a reference, flow over clean geometry, under the same flight condition as the iced one, is demonstrated in Figure 4-13. One can observe that due to ice accretion, the low-pressure region on the upper surface of the wing has been diminished, resulting in reduced lift. Meanwhile, the maximum shear stress increased

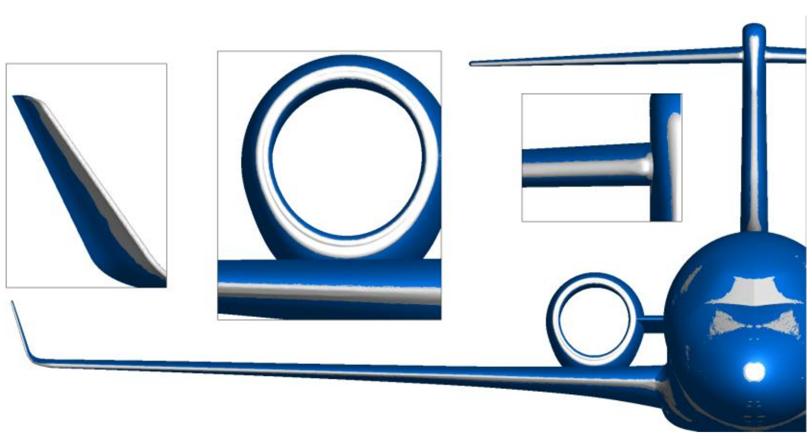


Figure 4-11: Ice shape considered for aerodynamic analysis.

significantly, leading to increased drag. Both effects will adversely affect performance, as will be shown in the remainder of this chapter.

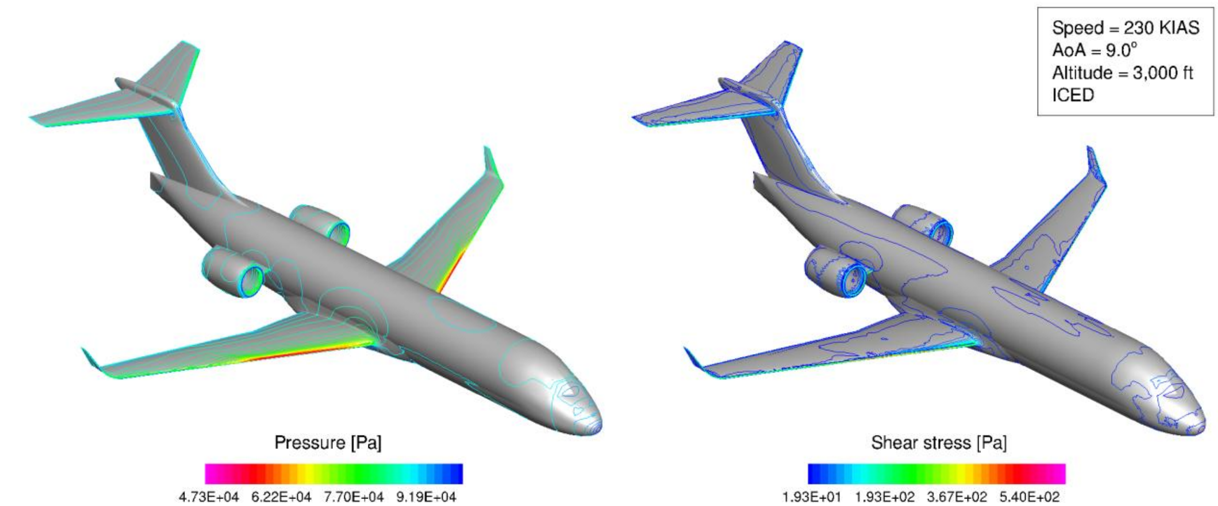


Figure 4-12: Flow over iced RJ.

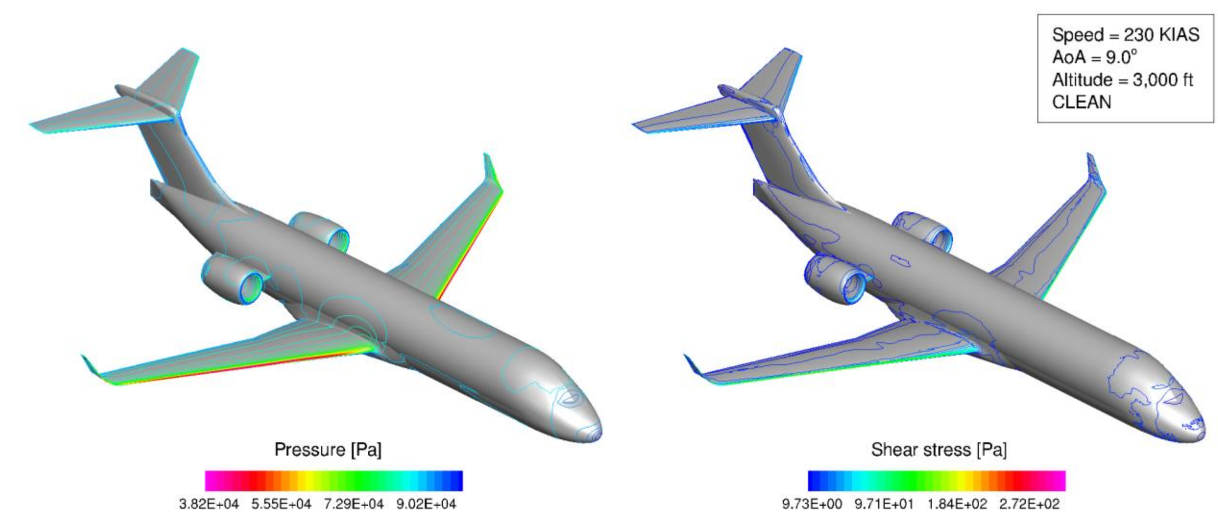


Figure 4-13: Flow over clean RJ.

4.2.1 Parameters of the analysis and iterative sampling

For the steady flight analysis and flight performance of a RJ in descent, climb and level flights, three parameters are selected: indicated air speed (IAS), AoA and pressure altitude (PA). These parameters are typical inputs made by pilots for longitudinal operation of the aircraft (AoA is equivalent to pitch angle minus flight path angle). Besides these three parameters, atmospheric parameters such as static pressure, static temperature and density are also required for setting up the numerical simulations. The International Standard Atmosphere model [75] is adopted to determine how atmospheric parameters change over the range of flight altitudes. The ranges for the parameters covered in the present analysis are shown in Table 4-2.

Table 4-2: Flight conditions parameters, with their corresponding ranges

	IAS [kt]	AoA [°]	PA [ft]	P_{static} [Pa]	T_{static} [°K]	Density [kg/m^3]	Mach
Min	190	0	3000	54900	256.47	0.75	0.30
Max	270	9	16000	90808	282.21	1.12	0.55

The initial sampling consists of 84 points (Figure 4-14), with 64 inside the design space and 20 on the boundaries. The error level is estimated on the basis of a leave-one-out approach: for any location i in the parameter space where a snapshot $\boldsymbol{U}_i$ is available, the error can be computed as the normalized L^2-norm of the vector of the errors ε_j^i at each node j of the mesh

$$\varepsilon_j^i = \frac{\left\| U_{j,ROM}^i - U_{j,CFD}^i \right\|_2}{\left\| U_{j,CFD}^i \right\|_2}, i = 1, \cdots, N_S, j = 1, \cdots, N_P. \tag{4.3}$$

For the pressure field, $U_j^i = p_j^i$ is the pressure value at node j of snapshot i; while for the shear stress field, $U_j^i = \tau_j^i = \sqrt{\left(\tau_{j,x}^i\right)^2 + \left(\tau_{j,y}^i\right)^2 + \left(\tau_{j,z}^i\right)^2}$ is the magnitude of shear stress at node j of snapshot i. Then the volume weighted overall error at each location i in the parameter space can be expressed as

$$\varepsilon^i = \sqrt{\frac{\sum_{j=1}^{N_P}\left(\varepsilon_j^i \upsilon_j\right)}{\sum_{j=1}^{N_P}\upsilon_j}}, i = 1, \cdots, N_S, j = 1, \cdots, N_P, \tag{4.4}$$

where υ_j is the cell volume of each node of the mesh.

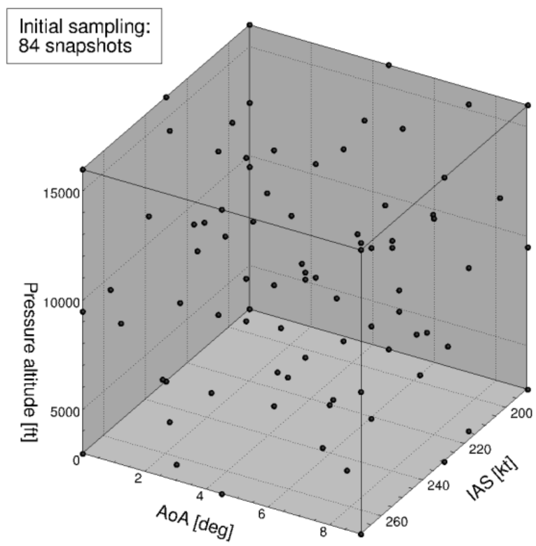


Figure 4-14: Initial sampling of 84 snapshots.

Figure 4-15 illustrates the eigenvalue convergence plot for the solution fields of pressure and shear stress τ (in x, y and z coordinates). The Y-axis is the normalized energy content associated to each POD mode, i.e. $\lambda_j/\sum_{i=1}^{Ns}\lambda_i$. In this analysis, an energy content of 99.999% is selected, corresponding to 44 modes for the pressure field, 67 modes for τ_x, 67 modes for τ_y, and 56 modes for τ_z used in flow solution prediction.

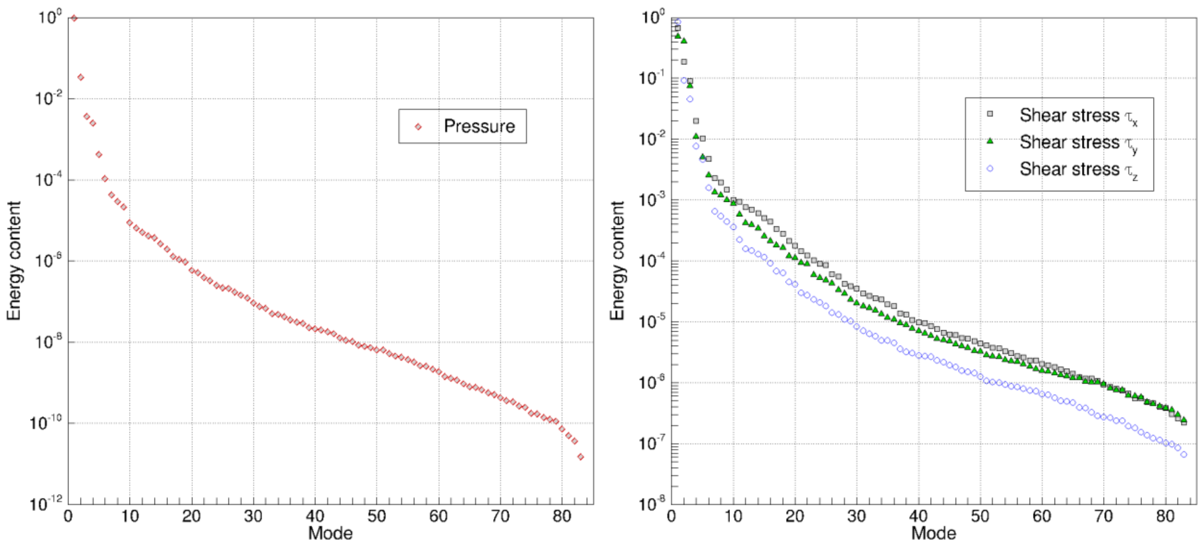


Figure 4-15: Eigenvalues vs. modes convergence: pressure (left); shear stress (right).

Figure 4-16 shows the LOOCV volume weighted L^2-norm error of the pressure and shear stress fields over the ice-contaminated aircraft. The LOOCV errors at each snapshot location are interpolated to obtain error estimation everywhere in the parameter space. As can be seen, the highest error of pressure and shear stress are 1.5% and 8.6%, respectively. Given that the pressure field accuracy is fairly good, the error driven sampling is focused on shear stress only. After seven sampling iterations, 129 snapshots were defined (Figure 4-17), with LOOCV the error reduced to 1.3% and 6.0%, for pressure and shear stress, respectively. For the present demonstration, a maximum error of 6.0% is considered satisfactory and no further snapshots are added to the set of CFD solutions. One must keep in mind that in natural icing testing an accuracy of 6% would be considered extraordinary.

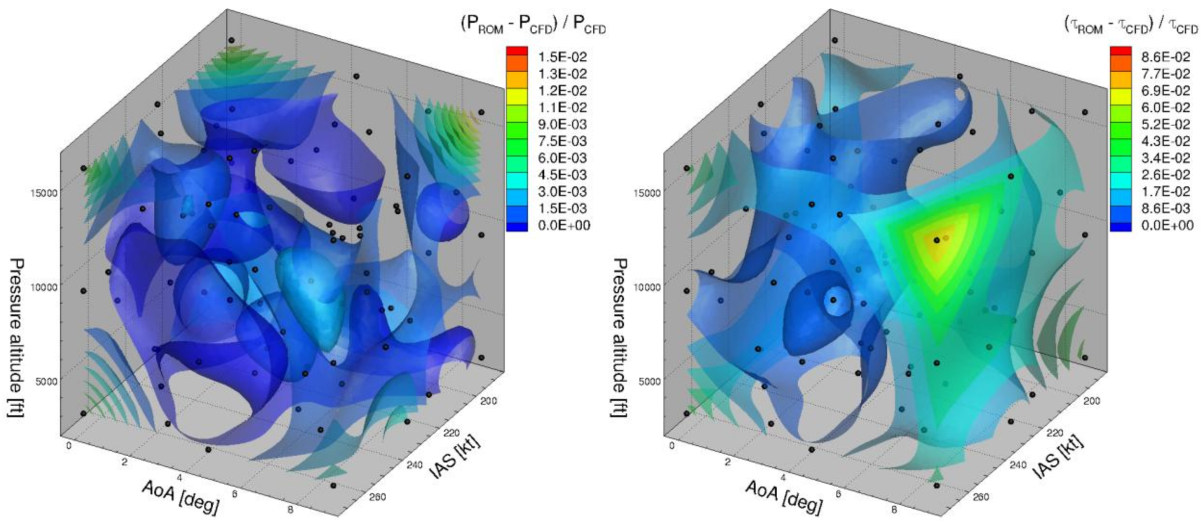


Figure 4-16: LOOCV error of the initial sampling, pressure field (left) and shear stress field (right) over iced RJ.

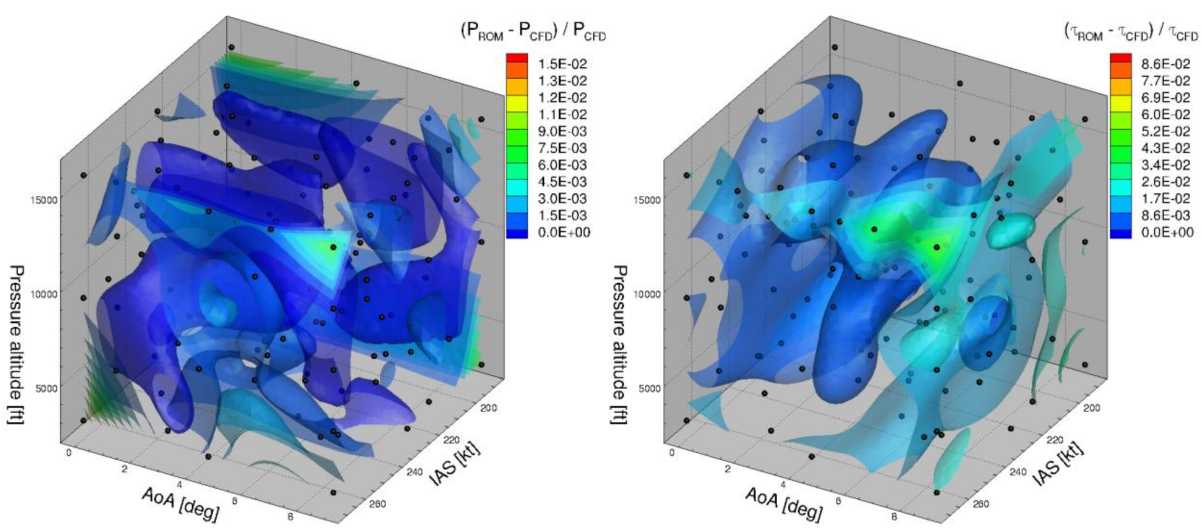


Figure 4-17: LOOCV error of the final sampling, pressure field (left) and shear stress field (right) over iced RJ.

In order to assess the aerodynamic performance degradation, flow solutions over the clean geometry are needed and 129 such snapshots under the same flight conditions as the ice-contaminated aircraft have been computed. The LOOCV of clean solutions are shown in Figure 4-18. The highest error of pressure and shear stress are 1.0% and 11.9%, respectively. Although higher than the ice-contaminated case, this level of accuracy can be considered acceptable.

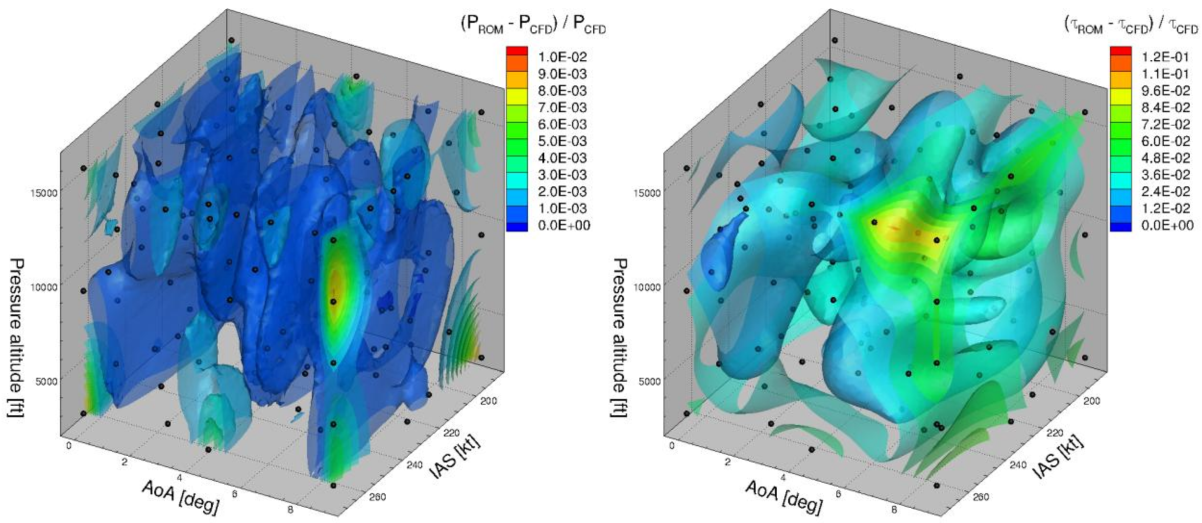


Figure 4-18: LOOCV error of the final sampling, pressure (left) and shear stress (right) of clean RJ.

4.2.2 Aerodynamic analysis for an aborted descent

Based on these two sets of snapshots (129 CFD solutions each, clean and iced) and the estimated error, one is reasonably confident to make predictions for other flight conditions within the parameter space. A hypothetical flight path representing an aborted descent is simulated using 21 target points, as shown in Figure 4-19. The parameter values of points on the flight path are selected to be as close to a realistic flight operation as possible. Among these 21 flight conditions, targets 1 to 9 represent the process when the airplane is allowed to exit its holding pattern and descend to 3,000 feet. Targets 9 to 13 represent the abortion of the descent, where the AoA increases from 0.9° to 7.0°, and the airplane starts to regain altitude. Targets 13 to 21 illustrate the climb stage, for which the aircraft gets back to the holding altitude and resumes level flight AoA, after which it may need to prepare for a second landing, or fly to an alternate airport. The input parameters values of targets 1, 9, 13 and 21 are listed in Table 4-3.

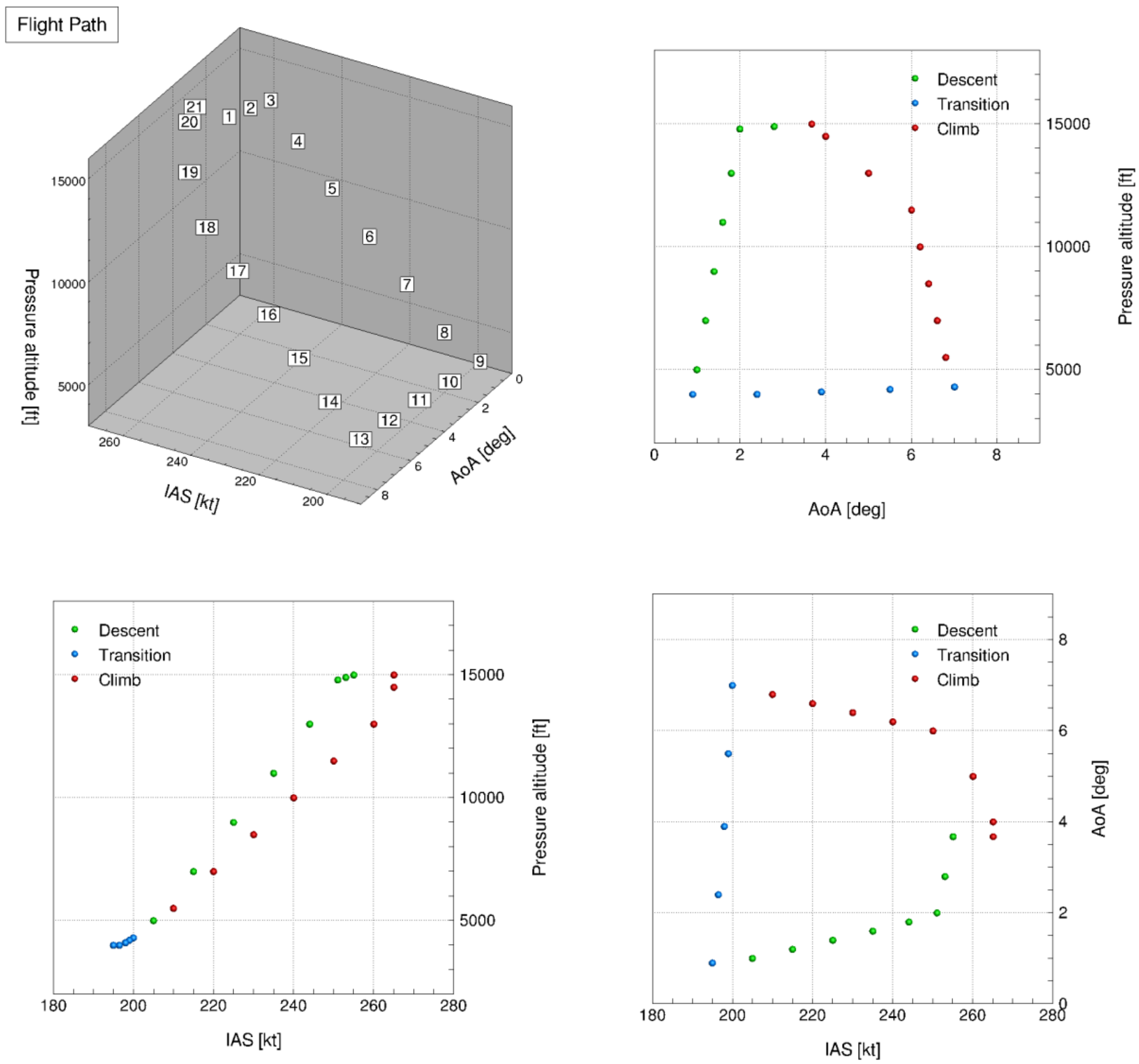


Figure 4-19: Flight path simulating an aborted descent.

Table 4-3: Flight conditions parameters on RJ flight path

	IAS [kt]	AoA [°]	PA [ft]
Target 1	255	3.7	15,000
Target 9	195	0.9	4,000
Target 13	200	7.0	4,300
Target 21	265	3.7	15,000

Based on the two sets of 129 snapshots each, 21 ROM solutions for the ice-contaminated airplane and another 21 ROM solutions for the clean airplane are obtained. As a validation, CFD solutions for flight conditions 1, 9, 13 and 21, on both iced and clean geometries, are computed to check the accuracy of ROM solutions. The volume weighted L^2-norm error of field variable in terms of pressure and shear stress are summarized in Table 4-4. As one can see, all errors are well bounded by the LOOCV error estimators; therefore the leave-one-out approach provides a reliable error estimation.

Table 4-4: Field variables' error of targets

	Iced		**Clean**	
	$\frac{\lvert P_{ROM}-P_{CFD}\rvert}{P_{CFD}}$	$\frac{\lvert \tau_{ROM}-\tau_{CFD}\rvert}{\tau_{CFD}}$	$\frac{\lvert P_{ROM}-P_{CFD}\rvert}{P_{CFD}}$	$\frac{\lvert \tau_{ROM}-\tau_{CFD}\rvert}{\tau_{CFD}}$
Target 1	6.4527E-04	6.1656E-03	6.1826E-04	9.3841E-03
Target 9	7.1029E-05	6.8925E-03	5.6252E-04	2.6860E-02
Target 13	2.2571E-04	4.4945E-03	2.3879E-04	7.7051E-03
Target 21	1.5195E-04	2.0054E-02	3.8306E-04	1.4939E-02

Figure 4-20 gives a detailed comparison of ROM vs. CFD results in terms of field quantities (pressure and shear stress) over the ice-contaminated airplane, under target flight condition 13, which is located at the bottom of the descent, where the airplane just re-establishes a climbing altitude (IAS = 200 kt, AoA = 7.0°, PA = 4,300 ft). Three cross-sections over the wing (21.6%, 50.1% and 95.2% of wing span) and one cross-section on the tail (56.3% tail span) are selected, denoted by part a), b), c) and d), respectively. As illustrated, ROM solution agrees very well with the CFD solution. Figure 4-21 shows comparison of ROM vs. CFD for pressure and shear stress, at the same target flight condition 13, but over the clean airplane. Again, the ROM solution agrees well with the CFD solution for most regions of the wing and tail. Figure 4-22 and Figure 4-23 demonstrate flow field comparison under flight condition 21, which is at the top of the climb path (IAS = 265 kt, AoA = 3.7°, PA = 15,000 ft), for ice-contaminated and clean geometries, respectively. The offline and online costs are summarized in Table 4-5. In this test case, the computational time of each CFD flow solution is 24 hours on 128 CPUs (offline cost). With these CFD solutions taken as snapshots, building a reduced model database (extracting POD modes, solving Kriging model parameters) takes 36.8 minutes using 16 CPUs (offline cost).

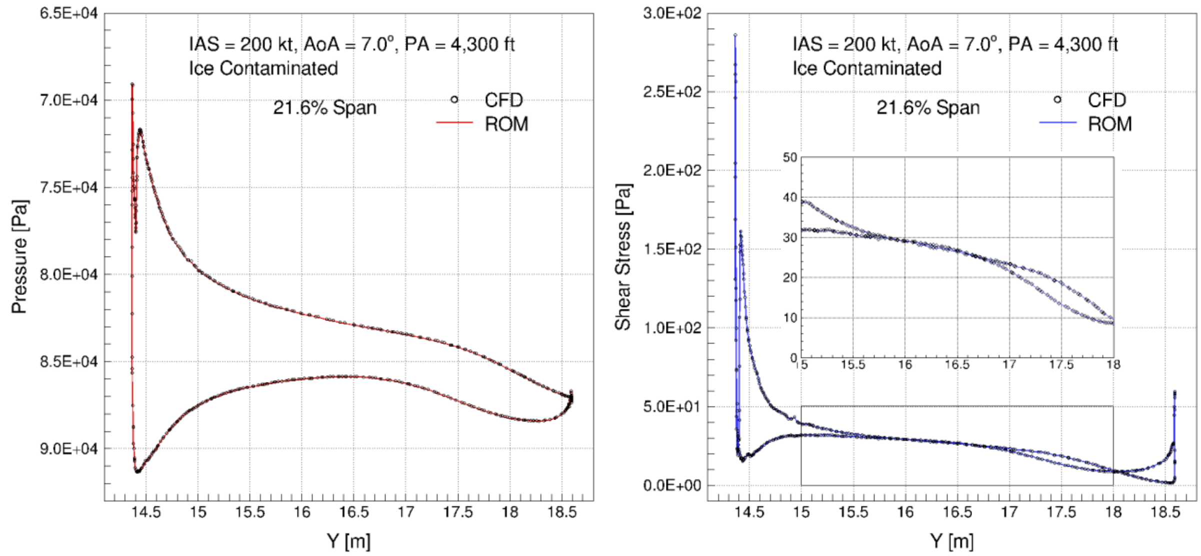
Table 4-5: Computational cost of aerodynamic analysis for the RJ

<table>
<tr><th></th><th colspan="2">Process</th><th>Computational cost</th></tr>
<tr><td rowspan="3">Offline</td><td>CFD simulations</td><td>Each snapshot</td><td>24 h on 128 CPUs*</td></tr>
<tr><td>Iterative sampling</td><td>LOOCV;
each CVT sampling</td><td>6 ~ 27 h on 16 CPUs*
(depending on the number of snapshots)
1 min on a single CPU**</td></tr>
<tr><td>Build database</td><td>Extract POD modes from the final set of snapshots, compute Kriging model parameter</td><td>36.8 min on 16 CPUs*</td></tr>
<tr><td>Online</td><td>Solve target</td><td>Logistic regression (classification), linear combination of POD modes</td><td>28 s on 16 CPUs*</td></tr>
</table>

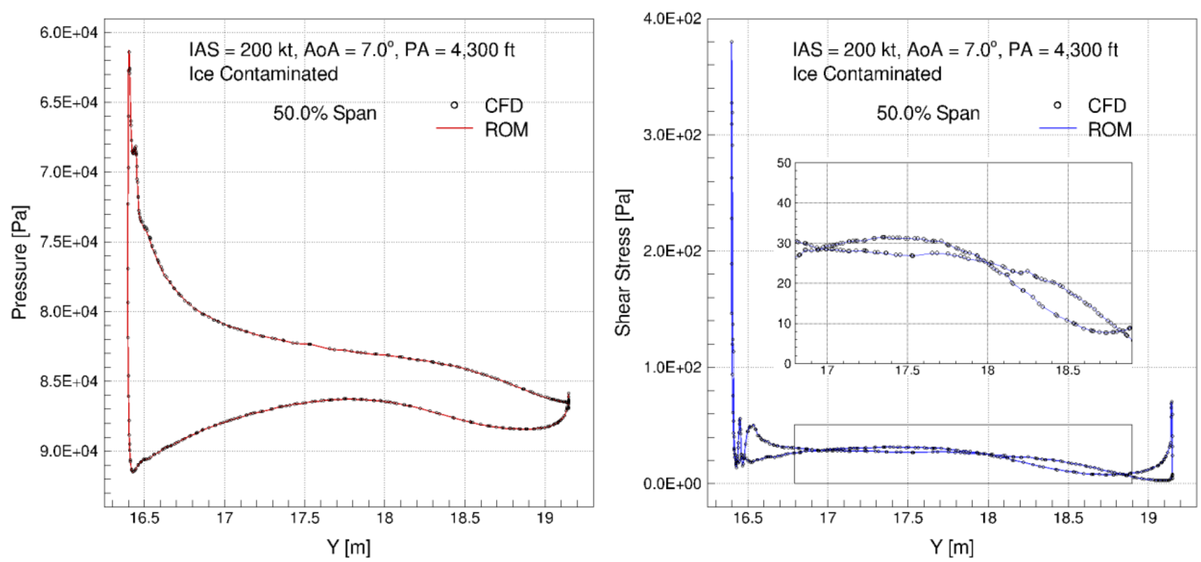
* Intel Xeon E5-2670 eight-core (supercomputer Guillimin)

** AMD Phenom II X6 1075T Processor, 800 MHz (desktop computer)

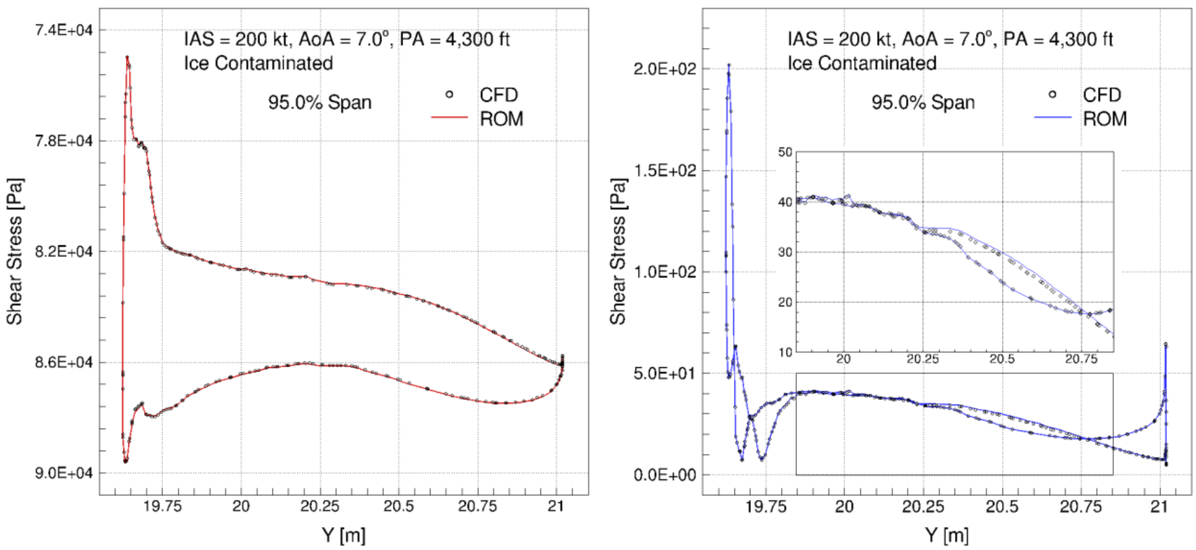
Once this database is stored, solving a ROM solution containing pressure and shear stress fields (over the entire computational domain) only takes 28 seconds using 16 CPUs (online cost).



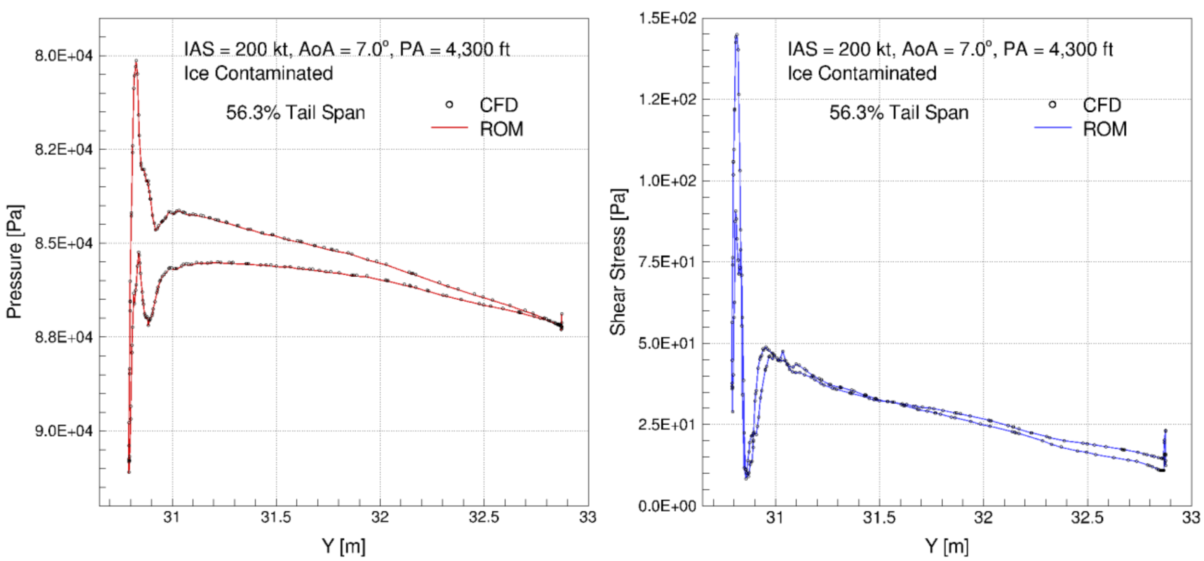
a)



b)

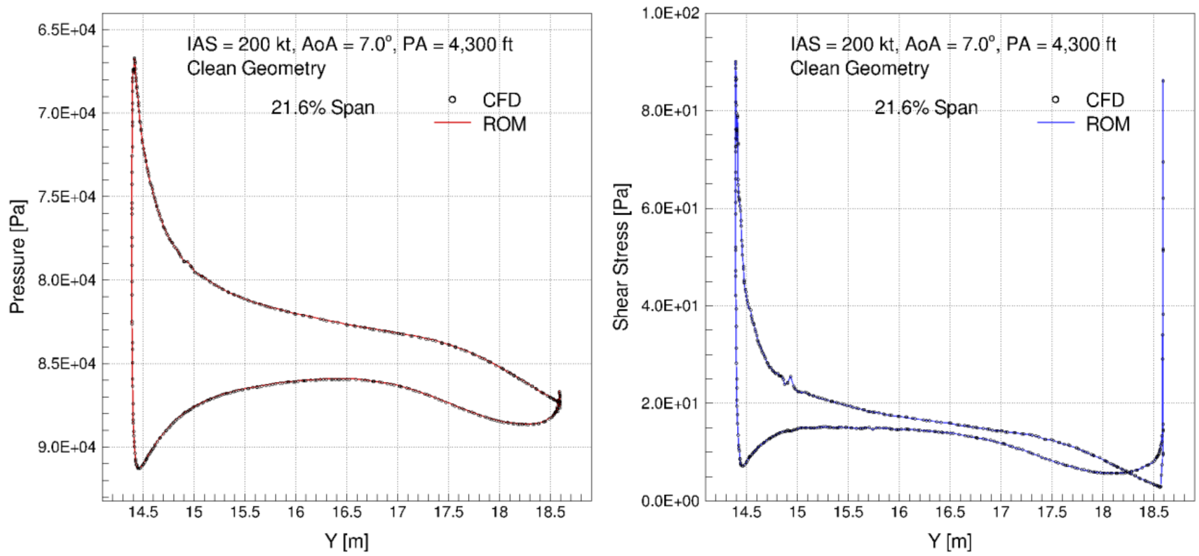


c)

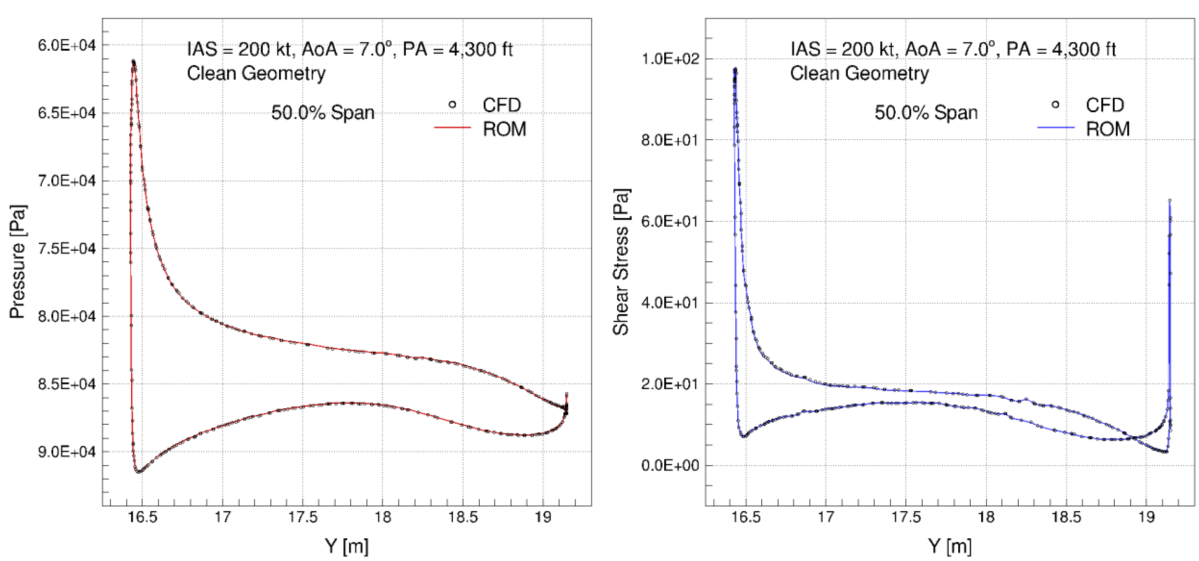


d)

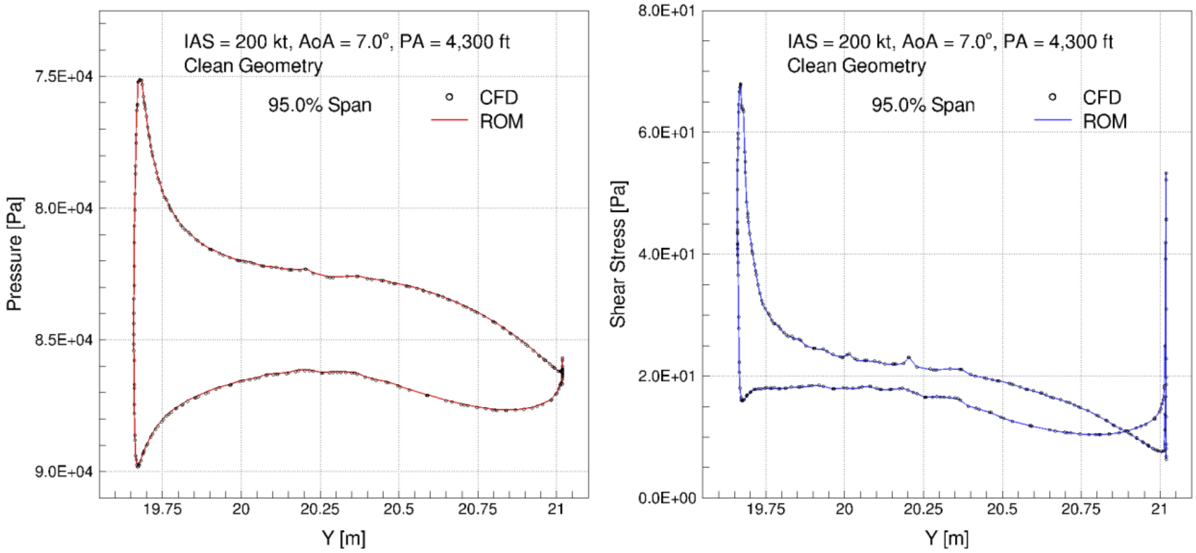
Figure 4-20: ROM vs. CFD comparison of pressure and shear stress, over iced geometry, target point 13.


IAS = 200 kt, AoA = 7.0°, PA = 4,300 ft
Clean Geometry
21.6% Span
CFD
ROM
Pressure [Pa]
Shear Stress [Pa]
Y [m]

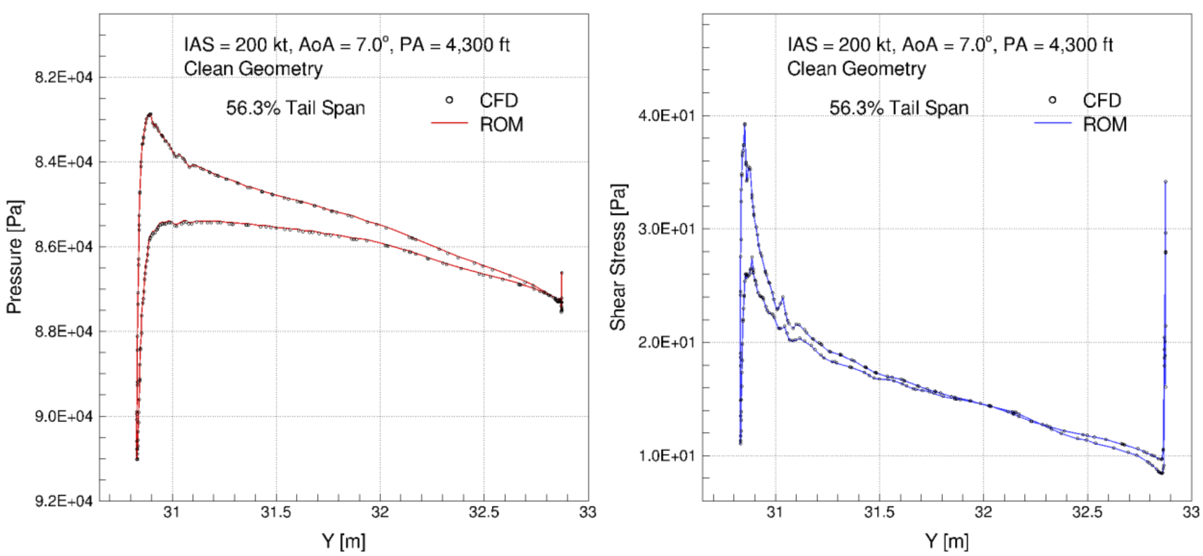
a)


IAS = 200 kt, AoA = 7.0°, PA = 4,300 ft
Clean Geometry
50.0% Span
CFD
ROM
Pressure [Pa]
Shear Stress [Pa]
Y [m]

b)

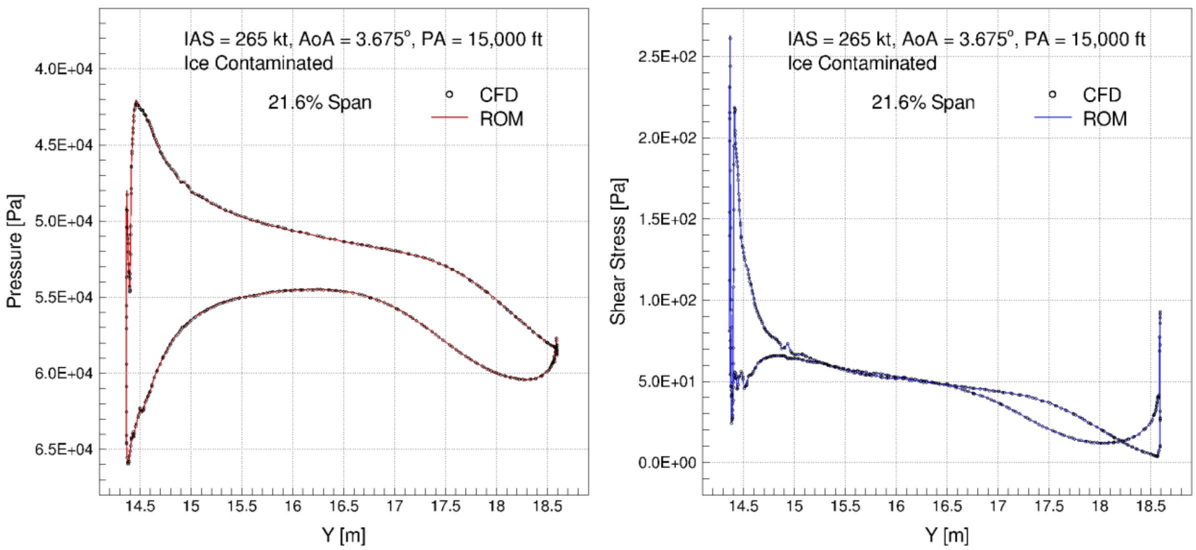


c)

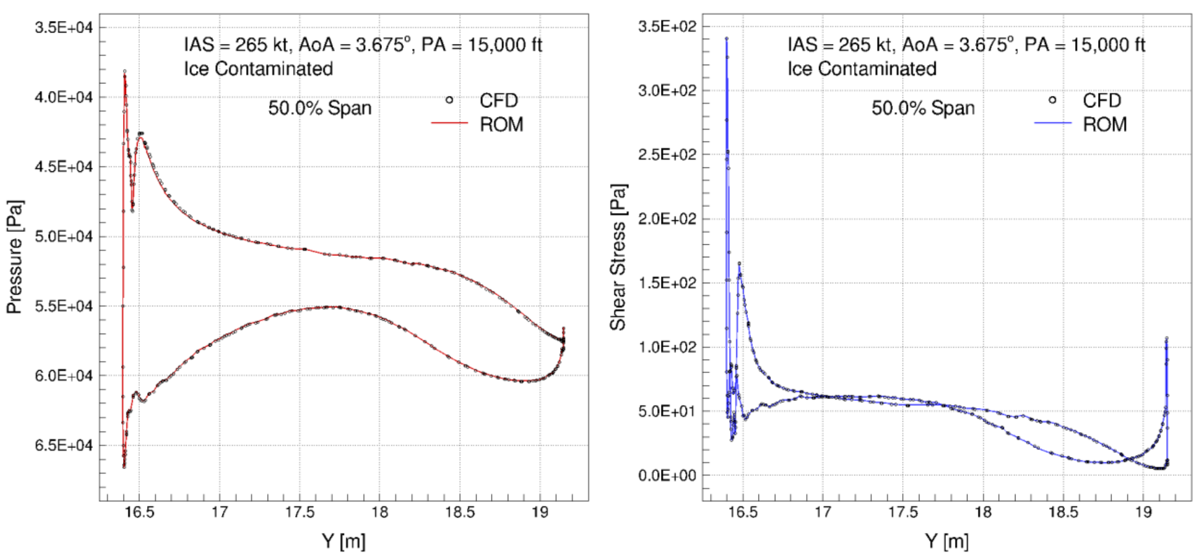


d)

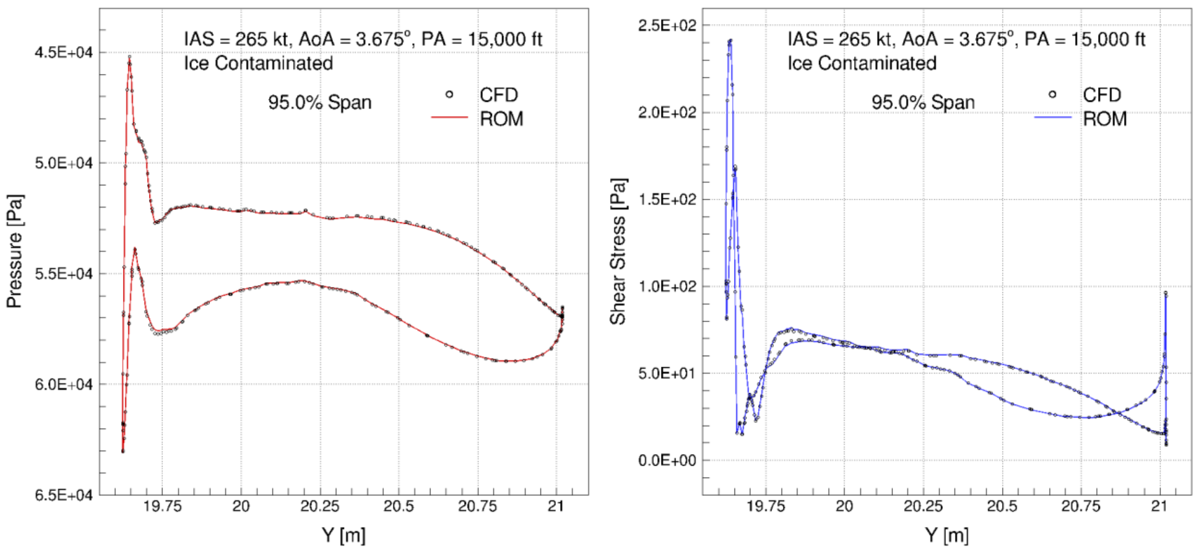
Figure 4-21: ROM vs. CFD comparison of pressure and shear stress, over clean geometry, target point 13.


IAS = 265 kt, AoA = 3.675°, PA = 15,000 ft
Ice Contaminated
21.6% Span
CFD
ROM
Pressure [Pa]
Y [m]
Shear Stress [Pa]

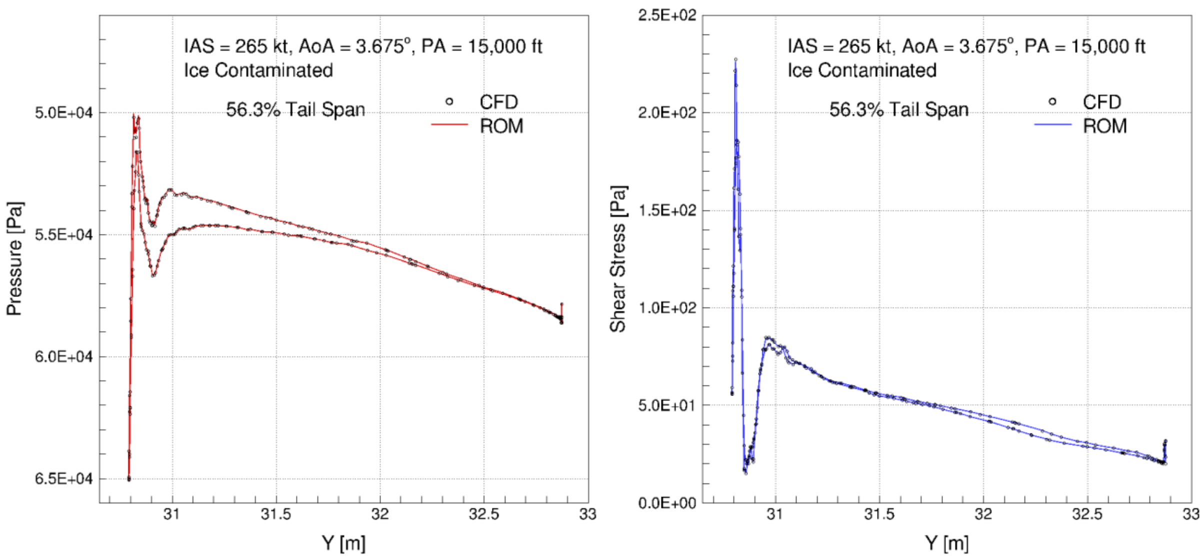
a)


IAS = 265 kt, AoA = 3.675°, PA = 15,000 ft
Ice Contaminated
50.0% Span
CFD
ROM
Pressure [Pa]
Y [m]
Shear Stress [Pa]

b)

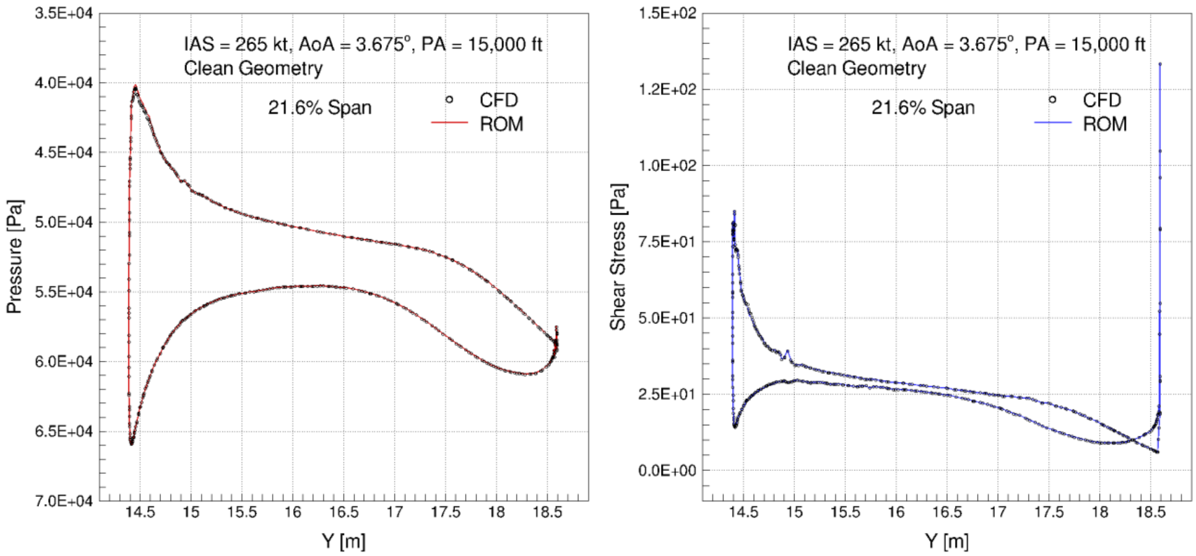


c)

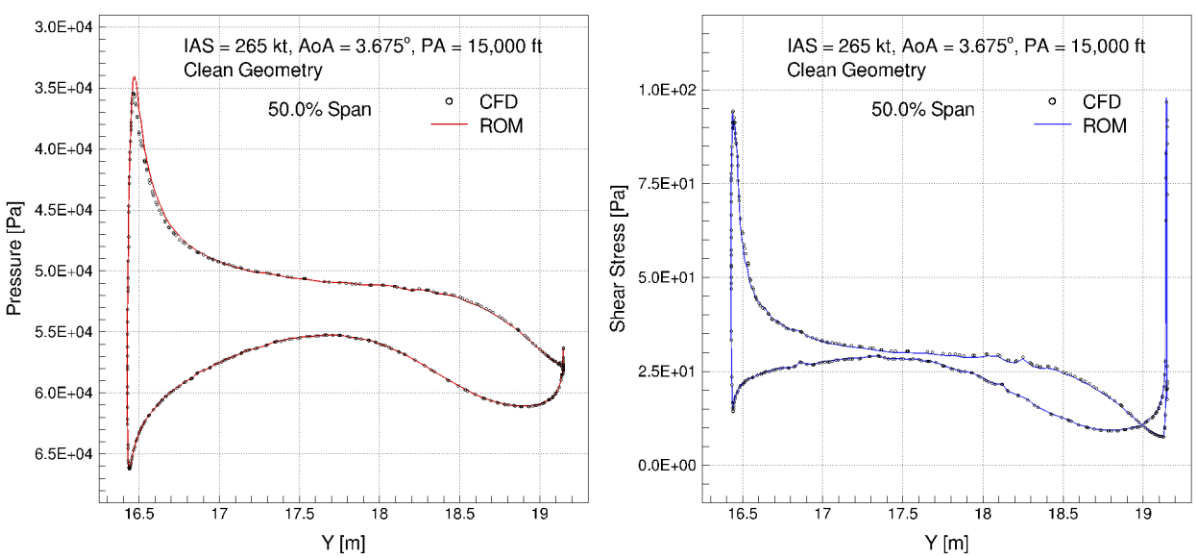


d)

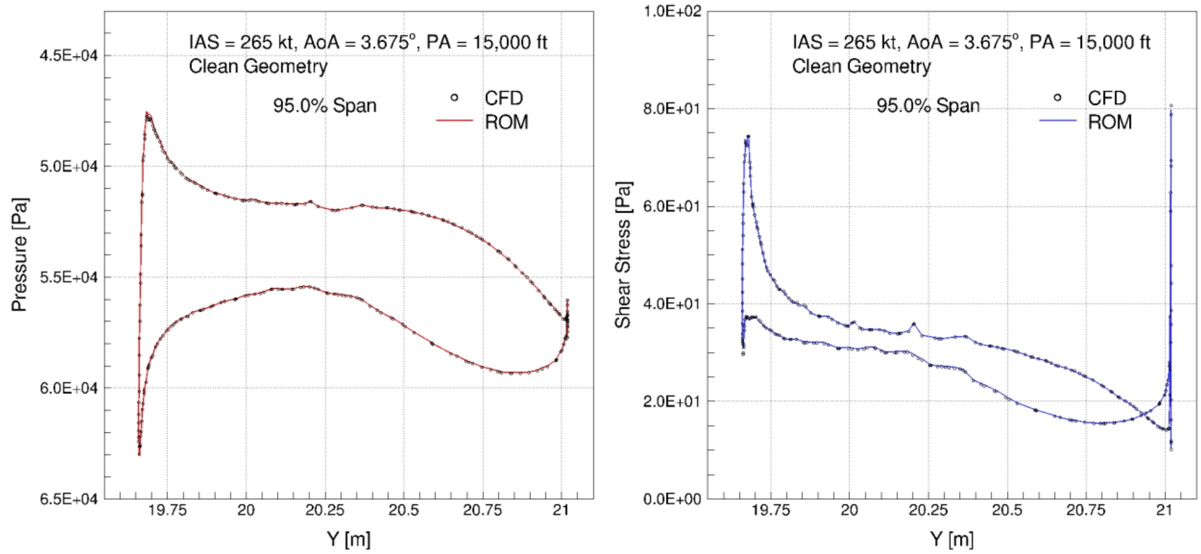
Figure 4-22: ROM vs. CFD comparison of pressure and shear stress, over iced geometry, target point 21.


IAS = 265 kt, AoA = 3.675°, PA = 15,000 ft
Clean Geometry
21.6% Span
CFD
ROM
Pressure [Pa]
Y [m]
IAS = 265 kt, AoA = 3.675°, PA = 15,000 ft
Clean Geometry
21.6% Span
CFD
ROM
Shear Stress [Pa]
Y [m]

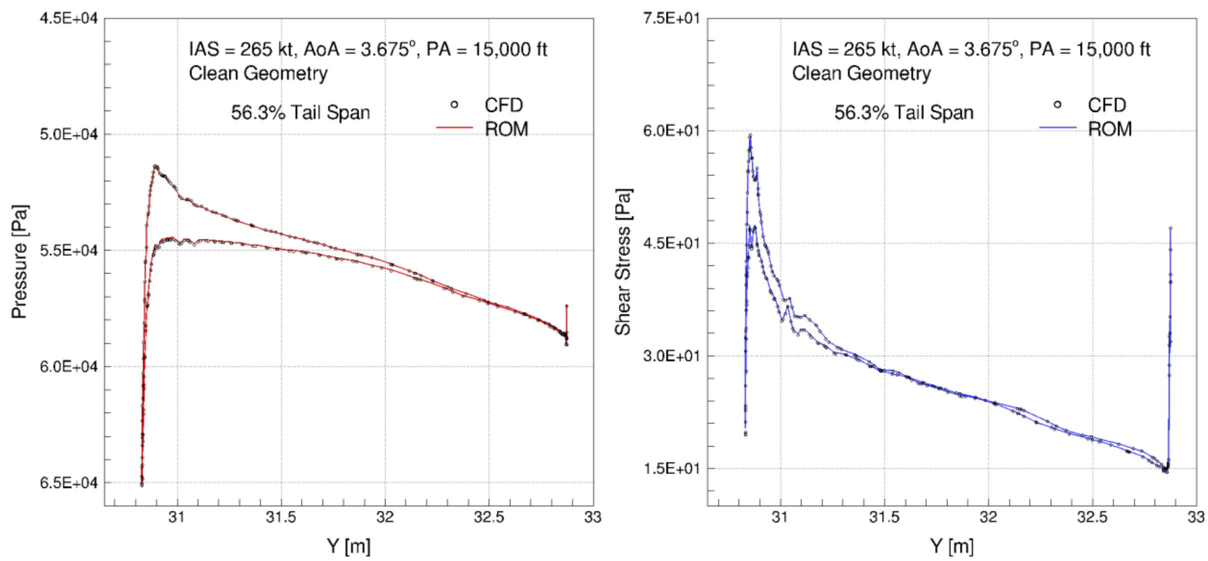
a)


IAS = 265 kt, AoA = 3.675°, PA = 15,000 ft
Clean Geometry
50.0% Span
CFD
ROM
Pressure [Pa]
Y [m]
IAS = 265 kt, AoA = 3.675°, PA = 15,000 ft
Clean Geometry
50.0% Span
CFD
ROM
Shear Stress [Pa]
Y [m]

b)



c)



d)

Figure 4-23: ROM vs. CFD comparison of pressure and shear stress, over clean geometry, target point 21.

4.2.3 Aerodynamic degradation: icing encounters flight simulator

Figure 4-20 to Figure 4-23 demonstrated the accuracy of ROM in terms of field variables. Since ROM solutions contain information on all grid points, it is possible to integrate the field variables (pressure and shear stress) to obtain integrated qualities such as lift coefficient (C_L), drag coefficient (C_D) and pitching moment coefficient (C_M). Although at some sections of the wing ROM and CFD do not perfectly match, after integration, C_L, C_D and C_M have a very small error, as shown in Table 4-6. Note that $\Delta C_L = |C_{L,CFD} - C_{L,ROM}|$, $\Delta C_D = |C_{D,CFD} - C_{D,ROM}|$ and $\Delta C_M = |C_{M,CFD} - C_{M,ROM}|$.

Table 4-6: Integrated variables' error at various target points

	Iced			Clean		
	$\frac{\Delta C_L}{C_{L,CFD}}$	$\frac{\Delta C_D}{C_{D,CFD}}$	$\frac{\Delta C_M}{C_{M,CFD}}$	$\frac{\Delta C_L}{C_{L,CFD}}$	$\frac{\Delta C_D}{C_{D,CFD}}$	$\frac{\Delta C_M}{C_{M,CFD}}$
Target 1	2.78E-03	2.19E-03	3.02E-03	4.76E-03	9.85E-03	4.55E-03
Target 9	1.67E-02	1.66E-02	2.11E-02	9.03E-03	4.42E-02	1.48E-02
Target 13	1.27E-03	6.74E-04	1.06E-03	1.06E-03	3.51E-03	1.22E-03
Target 21	9.58E-03	9.16E-03	1.08E-02	4.33E-05	7.64E-03	2.61E-04

It is of interest to see how performance is penalized during this aborted descent due to ice accretion. Moreover, what is important in a flight simulator are not the exact values but the differences in lift, drag and moments, which can be very accurate with the present method. Figure 4-24 illustrates aerodynamic degradation in terms of lift, drag and pitching moment for the ice-contaminated airplane, while Figure 4-25 gives the performance penalty in percentage. As can be seen, the increase of drag due to ice accretion is significant, reaching as much as 61%, compared to the clean aircraft. Meanwhile, lift may decrease by up to 17%, and the change of pitching moment is as high as 20%. The reason for relatively low penalty on lift is that the ice shape is rime (MVD = 21.26 μm, LWC = 0.30 g/m^3, Temperature = -13.4 °C). Such aerodynamic degradation values can be incorporated into flight simulators, with additional CFD-based visual aids/clues, making pilot training correspond to real-time and real-life by greatly improving simulator fidelity. Moreover, although the present analysis covers pre-stall AoAs, it can be easily extended to the stall and post-stall range of AoAs to enable stall recognition and recovery training.

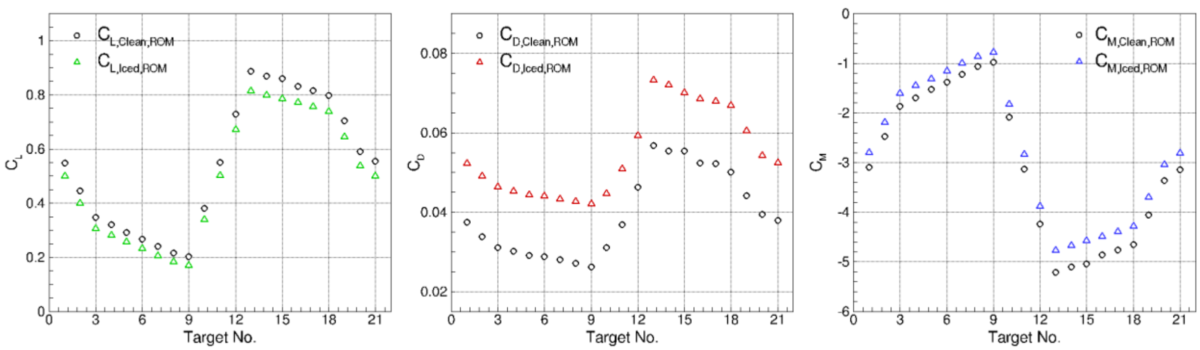


Figure 4-24: Aerodynamic degradations in C_L, C_D and C_M.

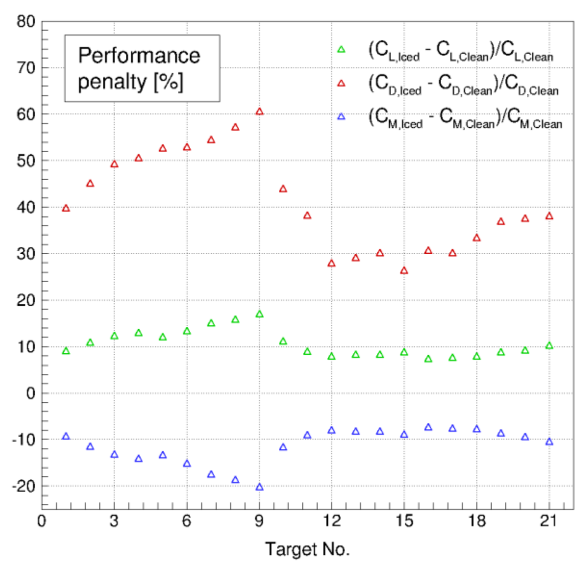


Figure 4-25: Aerodynamic penalty for ice-contaminated RJ.

CHAPTER 5 CONCLUSIONS

The thesis presents a rigorous and self-contained ROM framework based on proper orthogonal decomposition, multi-dimensional interpolation and machine learning algorithms, along with an error driven iterative sampling approach to adaptively define an optimal set of snapshots.

The proposed methodologies were applied to an aero-icing study for an airfoil, where icing effects in terms of the mass of ice and maximum lift coefficient were investigated. The application of the proposed approach in the context of in-flight icing demonstrated that Appendix C can be, it is believed for the first time, completely explored. Then, the methodologies were applied, again for the first time, to a detailed aero-icing study of a regional jet, in terms of: 1) the "complete" exploration of the FAA CM icing conditions for the shape/mass of ice, and 2) its aerodynamic degradation due to ice contamination during holding, and the consequent effect during descent and aborted landing.

Future research work can be done in investigating the use of graphics processing units (GPUs) to further speed up the execution of the ROM code to achieve real-time performance. Although in this thesis, only CFD data is used as the snapshots for ROM, it should be pointed out that the proposed methodology is comprehensive in permitting the combination of CFD, EFD and FFD. In such case, certification will not only be more complete, and dangerous areas fully identified, but in addition certification campaigns could be shortened to one season instead of several. The proposed methodology can also be used to make the flight simulator a proactive tool in preventing accidents rather than just their reoccurrence. In addition, while the icing certification envelope is the same for all aircraft, its effect on each class of aircraft is far from being the same. Thus, the current methodology can be used to assess the significance of an icing encounter before launching the aircraft into it, and also to give confidence to the pilots as to the anticipated behavior of the aircraft. Thus, in summary, the technology enables designers to avoid any “blind spots” and provides data for conditions that could not be located in nature, or that are too dangerous or impossible to test in real life. All these will lead to beneficial and sustained impact on aviation safety through CFD simulations.

REFERENCES

[1] Dillingham, G. L., "Aviation Safety: Improved Planning Could Help FAA Address Challenges Related to Winter Weather Operations," GAO-10-678, United States Government Accountability Office, July 2010.

[2] Dillingham, G. L., "Aviation Safety: Icing and Winter Weather-Related Recommendations That NTSB Has Issued Since 1996 (GAO-10-679SP), an E-supplement to (GAO-10-678)," GAO-10-679SP, United States Government Accountability Office, 2010.

[3] "Aircraft Accident Report: Crash During Approach to Landing, Empire Airlines Flight 8284, Avions de Transport Régional Aérospatiale Alenia ATR 42-320, N902FX, Lubbock, Texas, January 27, 2009," NTSB/AAR-11/02, National Transportation Safety Board, Washington, D.C. 20594, 2011.

[4] Code of Federal Regulations, Title 14 - Aeronautics and Space, Part 25 - Airworthiness Standards: Transport Category Airplanes, Federal Aviation Administration, Washington, D.C. 20591, revised and re-issued annually.

[5] "Airplane and Engine Certification Requirements in Supercooled Large Drop, Mixed Phase, and Ice Crystal Icing Conditions," Department of Transportation, Federal Aviation Administration, Notice of Proposed Rulemaking, FAA-2010-0636, Federal Register, Vol. 75, No. 124, 2010, pp. 37311-37339.

[6] Bernstein, B. C., Campo, W., Algodoal, L., Bottino, F., Lilie, L., and Henriques, A., "The Embraer-170 and -190 Natural Icing Flight Campaigns: Keys to Success," AIAA Paper AIAA-2006-264, Jan. 2006.

[7] Barnhart, B. P., and Ratvasky, T. P., "Effective Training for Flight in Icing Conditions," I/ITSEC 2006 Paper No. 2973, NASA/TM-2007-214693, Mar. 2007.

[8] "Flight Simulation Training Device Qualification Standards for Extended Envelope and Adverse Weather Event Training Tasks," Department of Transportation, Federal Aviation Administration, Notice of Proposed Rulemaking, FAA-2014-0391, Federal Register, Vol. 79, No. 132, 2014, pp. 39462-39753.

[9] Ratvasky, T. P., Ranaudo, R. J., Barnhart, B. P., Dickes, E. G., and Gingras, D. R., "Development and Utility of a Piloted Flight Simulator for Icing Effects Training," AIAA Paper AIAA-2003-0022, NASA /TM-2003-212116, Apr. 2003.

[10] Nakakita, K., "Toward Real-Time Aero-Icing Simulation Using Reduced Order Models," Master's thesis, McGill University, Montreal, QC, 2007.

[11] Lappo, V., "Real-Time Aero-Icing Simulations Using Reduced Order Models," Master’s thesis, McGill University, Montreal, QC, 2010.

[12] Fossati, M., and Habashi, W. G. "Multiparameter Analysis of Aero-Icing Problems Using Proper Orthogonal Decomposition and Multidimensional Interpolation," *AIAA Journal*, Vol. 51, No. 4, 2013, pp. 946-960.

[13] Mankins, J. C., "Technology Readiness Levels," A White Paper, Advanced Concepts Office, Office of Space Access and Technology, NASA, Apr. 1995.

[14] Ito, K., and Ravindran, S. S. "Reduced Basis Method for Optimal Control of Unsteady Viscous Flows," *International Journal of Computational Fluid Dynamics*, Vol. 15, No. 2, 2001, pp. 97-113.

[15] Burkardt, J., Gunzburger, M., and Lee, H.-C. "POD and CVT-based Reduced-Order Modeling of Navier-Stokes Flows," *Computer Methods in Applied Mechanics and Engineering,* Vol. 196, No. 1, 2006, pp. 337-355.

[16] Holmes, P., Lumley, J. L., and Berkooz, G., *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, Cambridge, U.K., 1996, Chap. 3.

[17] Sirovich, L. "Turbulence and the Dynamics of Coherent Structures. Part II: Symmetries and Transformations," *Quarterly of Applied Mathematics,* Vol. 45, 1987, pp. 573-582.

[18] Lucia, D. J., Beran, P. S., and Silva, W. A. "Reduced-Order Modeling: New Approaches for Computational Physics," *Progress in Aerospace Sciences*, Vol. 40, No. 1, 2004, pp. 51-117.

[19] Dowell, E. H., and Hall, K. C. "Modeling of Fluid-Structure Interaction," *Annual Review of Fluid Mechanics,* Vol. 33, No. 1, 2001, pp. 445-490.

[20] Amsallem, D., Cortial, J., and Farhat, C. "Toward Real-Time Computational-Fluid-Dynamics-Based Aeroelastic Computations Using a Database of Reduced-Order Information," *AIAA Journal,* Vol. 48, No. 9, 2010, pp. 2029-2037.

[21] Hall, K. C., Thomas, J. P., and Dowell, E. H. "Proper Orthogonal Decomposition Technique for Transonic Unsteady Aerodynamic Flows," *AIAA journal,* Vol. 38, No. 10, 2000, pp. 1853-1862.

[22] Xiao, D., Yang, P., Fang, F., Xiang, J., Pain, C. C., and Navon, I. M. "Non-Intrusive Reduced Order Modelling of Fluid-Structure Interactions," *Computer Methods in Applied Mechanics and Engineering,* Vol. 303, May 2016, pp. 35-54.

[23] Ly, H. V., and Tran, H. T. "Modeling and Control of Physical Processes using Proper Orthogonal Decomposition," *Mathematical and Computer Modelling,* Vol. 33, No. 1, 2001, pp. 223-236.

[24] Audouze, C., De Vuyst, F., and Nair, P. B. "Reduced-Order Modeling of Parameterized PDEs using Time-Space-Parameter Principal Component Analysis," *International Journal for Numerical Methods in Engineering,* Vol. 80, No. 8, 2009, pp. 1025-1057.

[25] Bui-Thanh, T., Damodaran, M., and Willcox, K., "Proper Orthogonal Decomposition Extensions for Parametric Applications in Compressible Aerodynamics," AIAA Paper AIAA-2003-4213, June 2003.

[26] LeGresley, P. A., and Alonso, J. J., "Airfoil Design Optimization using Reduced Order Models based on Proper Orthogonal Decomposition," AIAA Paper AIAA-2000-2545, June 2000.

[27] Fossati, M., Najafiyazdi, M., and Habashi, W. G., "Mesh Adaptation for Unsteady Problems via Reduced Order Modeling," AIAA Paper AIAA-2011-3692, June 2011.

[28] Epureanu, B. I. "A Parametric Analysis of Reduced Order Models of Viscous Flows in Turbomachinery," *Journal of fluids and structures,* Vol. 17, No. 7, 2003, pp. 971-982.

[29] Mifsud, M. J., Shaw, S. T., and MacManus, D. G. "A High-Fidelity Low-Cost Aerodynamic Model Using Proper Orthogonal Decomposition," *International Journal for Numerical Methods in Fluids,* Vol. 63, No. 4, 2010, pp. 468-494.

[30] Nakakita, K., Nadarajah, S., and Habashi, W. "Toward Real-Time Aero-Icing Simulation of Complete Aircraft via FENSAP-ICE," *Journal of Aircraft,* Vol. 47, No. 1, 2010, pp. 96-109.

[31] Lappo, V., and Habashi, W. G., "Reduced Order POD/Kriging Modeling for Real-Time 3D CFD," Proceedings of 11th Pan-American Congress of Applied Mechanics (PACAM XI), ABCM, Rio de Janeiro, RJ, Brazil, Jan. 2010.

[32] Jung, S. K., Shin, S., Myong, R. S., and Cho, T. H. "An Efficient CFD-based Method for Aircraft Icing Simulation using a Reduced Order Model," *Journal of Mechanical Science and Technology,* Vol. 25, No. 3, 2011, pp. 703-711.

[33] Zhan, Z., Habashi, W. G., and Fossati, M. "Local Reduced-Order Modeling and Iterative Sampling for Parametric Analyses of Aero-Icing Problems," *AIAA Journal,* Vol. 53, No. 8, 2015, pp. 2174-2185.

[34] Zhan, Z., Habashi, W. G., and Fossati, M. "Real-Time Regional Jet Comprehensive Aero-Icing Analysis via Reduced-Order Modeling," *AIAA Journal*, 2016, accessed August 6, 2016. doi: http://arc.aiaa.org/doi/abs/10.2514/1.J055013

[35] Lucia, D. J., and Beran, P. S. "Projection Methods for Reduced Order Models of Compressible Flows," *Journal of Computational Physics,* Vol. 188, No. 1, 2003, pp. 252-280.

[36] Iollo, A., "Remarks on the Approximation of the Euler Equations by a Low Order Model," No. 3329, Institut National de Recherche en Informatique et en Automatique, Dec. 1997.

[37] Bourguet, R., Braza, M., and Dervieux, A. "Reduced-Order Modeling for Unsteady Transonic Flows around an Airfoil," *Physics of Fluids*, Vol. 19, No. 11, 2007, pp. 111701-1-111701-4.

[38] Fang, F., Pain, C. C., Navon, I. M., Elsheikh, A. H., Du, J., and Xiao, D. "Non-Linear Petrov-Galerkin Methods for Reduced Order Hyperbolic Equations and Discontinuous

Finite Element Methods," *Journal of Computational Physics,* Vol. 234, Feb. 2013, pp. 540-559.

[39] Berkooz, G., Holmes, P., and Lumley, J. L. "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows," *Annual Review of Fluid Mechanics*, Vol. 25, No. 1, 1993, pp. 539-575.

[40] Xiao, D., Fang, F., Buchan, A. G., Pain, C. C., Navon, I. M., and Muggeridge, A. "Non-Intrusive Reduced Order Modelling of the Navier-Stokes Equations," *Computer Methods in Applied Mechanics and Engineering,* Vol. 293, Aug. 2015, pp. 522-541.

[41] Yapalparvi, R., Chandar, D. D. J., and Damodaran, M., "Prediction of Unsteady Flow Fields and Trajectories of Tumbling Plates Using Reduced Order Modeling," AIAA Paper AIAA-2009-329, Jan. 2009.

[42] Amsallem, D., Zahr, M. J., and Farhat, C. "Nonlinear Model Order Reduction Based on Local Reduced-Order Bases," *International Journal for Numerical Methods in Engineering,* Vol. 92, No. 10, 2012, pp. 891-916.

[43] Washabaugh, K., Amsallem, D., Zahr, M., and Farhat, C., "Nonlinear Model Reduction for CFD Problems Using Local Reduced Order Bases," AIAA Paper AIAA-2012-2686, June 2012.

[44] Peherstorfer, B., Butnaru, D., Willcox, K., and Bungartz, H. J. "Localized Discrete Empirical Interpolation Method," *SIAM Journal on Scientific Computing,* Vol. 36, No. 1, 2014, pp. A168-A192.

[45] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. "Design and Analysis of Computer Experiments," *Statistical Science,* Vol. 4, No. 4, 1989, pp. 409-423.

[46] Statnikov, R. B., and Matusov, J. B., *Multicriteria Optimization and Engineering*, Chapman and Hall, New York, 1995, pp. 225-261.

[47] Epureanu, B. I., Dowell, E. H., and Hall, K. C. "Reduced-Order Models of Unsteady Transonic Viscous Flows in Turbomachinery," *Journal of Fluids and Structures,* Vol. 14, No. 8, 2000, pp. 1215-1234.

[48] Grepl, M. A., and Patera, A. T. "A Posteriori Error Bounds for Reduced-Basis Approximations of Parametrized Parabolic Partial Differential Equations," *ESAIM: Mathematical Modelling and Numerical Analysis,* Vol. 39, No. 1, 2005, pp. 157-181.

[49] Veroy, K., and Patera, A. T. "Certified Real-Time Solution of the Parametrized Steady Incompressible Navier-Stokes Equations: Rigorous Reduced-Basis a Posteriori Error Bounds," *International Journal for Numerical Methods in Fluids,* Vol. 47, No. 8, 2005, pp. 773-788.

[50] Haasdonk, B., Dihlmann, M., and Ohlberger, M. "A Training Set and Multiple Bases Generation Approach for Parameterized Model Reduction Based on Adaptive Grids in Parameter Space," *Mathematical and Computer Modelling of Dynamical Systems,* Vol. 17, No. 4, 2011, pp. 423-442.

[51] Haasdonk, B., and Ohlberger, M. "Reduced Basis Method for Finite Volume Approximations of Parametrized Linear Evolution Equations," *ESAIM: Mathematical Modelling and Numerical Analysis,* Vol. 42, No. 2, 2008, pp. 277-302.

[52] Bui-Thanh, T., Willcox, K., and Ghattas, O. "Parametric Reduced-Order Models for Probabilistic Analysis of Unsteady Aerodynamic Applications," *AIAA Journal,* Vol. 46, No. 10, 2008, pp. 2520-2529.

[53] Bui-Thanh, T., Willcox, K., and Ghattas, O. "Model Reduction for Large-Scale Systems with High-Dimensional Parametric Input Space," *SIAM Journal on Scientific Computing*, Vol. 30, No. 6, 2008, pp. 3270-3288.

[54] Paul-Dubois-Taine, A., and Amsallem, D. "An Adaptive and Efficient Greedy Procedure for the Optimal Training of Parametric Reduced-Order Models," *International Journal for Numerical Methods in Engineering*, Vol. 102, No. 5, 2015, pp. 1262-1292.

[55] Du, Q., Faber, V., and Gunzburger, M. "Centroidal Voronoi Tessellations: Applications and Algorithms," *SIAM Review*, Vol. 41, No. 4, 1999, pp. 637-676.

[56] Matheron, G. "Principles of geostatistics," *Economic geology*, Vol. 58, No. 8, 1963, pp. 1246-1266.

[57] Jones, D. R. "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, Vol. 21, No. 4, 2001, pp. 345-383.

[58] Forrester, A. I. J., and Keane, A. J. "Recent Advances in Surrogate-based Optimization," *Progress in Aerospace Sciences*, Vol. 45, No. 1, 2009, pp. 50-79.

[59] Jeong, S., Murayama, M., and Yamamoto, K. "Efficient Optimization Design Method using Kriging Model," *Journal of aircraft*, Vol. 42, No. 2, 2005, pp. 413-420.

[60] Martin, J. D., and Simpson, T. W. "Use of Kriging Models to Approximate Deterministic Computer Models," *AIAA Journal*, Vol. 43, No. 4, 2005, pp. 853-863.

[61] Joseph, V. R., Hung, Y., and Sudjianto, A. "Blind Kriging: a New Method for Developing Metamodels," *Journal of Mechanical Design*, Vol. 130, No. 3, 2008, pp. 031102-031102-8.

[62] Nocedal, J., and Wright, S. J., *Numerical Optimization*, 2nd ed., Springer Science & Business Media, New York, 2006, Chaps. 2, 3, 6.

[63] Dennis Jr, J. E., and Schnabel, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996, Chaps. 7, 9.

[64] Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, New York, 2006, Chaps. 4, 9.

[65] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning*, Springer, New York, 2009, Chaps. 2, 4, 5, 14.

[66] Lloyd, S. P. "Least-Squares Quantization in PCM," *IEEE Transactions on Information Theory,* Vol. 28, No. 2, 1982, pp. 129-137.

[67] Habashi, W. G. "Recent Advances in CFD for In-Flight Icing Simulations," *Japan Society of Fluid Mechanics,* Vol. 28, No. 2, 2009, pp. 99-118.

[68] Beaugendre, H., Morency, F., and Habashi, W. G. "FENSAP-ICE's Three-dimensional In-flight Ice Accretion Module: ICE3D," *Journal of Aircraft,* Vol. 40, No. 2, 2003, pp. 239-247.

[69] Bourgault, Y., Boutanios, Z., and Habashi, W. G. "Three-Dimensional Eulerian Approach to Droplet Impingement Simulation using FENSAP-ICE, Part 1: Model, Algorithm, and Validation," *Journal of Aircraft,* Vol. 37, No. 1, 2000, pp. 95-103.

[70] Baruzzi, G. S., Habashi, W. G., Guevremont, J. G., and Hafez, M. M. "A Second Order Finite Element Method for the Solution of the Transonic Euler and Navier-Stokes Equations," *International Journal for Numerical Methods in Fluids,* Vol. 20, No. 8-9, 1995, pp. 671-693.

[71] Croce, G., Beaugendre, H., and Habashi, W. G., "CHT3D-FENSAP-ICE Conjugate Heat Transfer Computations with Droplet Impingement and Runback Effects," AIAA Paper AIAA-2002-0386, Jan. 2002.

[72] "User Manual. FENSAP-ICE: FENSAP, DROP3D, ICE3D, C3D/CHT3D; OPTIGRID; FENSAP-FS," Version 2013 Release 1.0, Newmerical Technologies International, Montreal, QC, Canada H3A 2M7, 2013.

[73] Addy, H. E., Broeren, A. P., Zoeckler, J. G., and Lee, S., "A Wind Tunnel Study of Icing Effects on a Business Jet Airfoil," NASA/TM-2003-212124, AIAA Paper AIAA-2003-0727, Jan. 2003.

[74] Si, H., "TetGen: A Quality Tetrahedral Mesh Generator and 3D Delaunay Triangulator," User's Manual, Version 1.5, Weierstrass Institute for Applied Analysis and Stochastic, Berlin, Germany, 2013.

[75] "U.S. Standard Atmosphere, 1976," U.S. Government Printing Office, Washington, D.C., Oct. 1976, pp. 11-20.