



Master thesis

Assessment of hyper-reduction techniques in the context of CFD-based intrusive reduced order modeling

Cedric Fallet
Matrikelnummer: 4808435

First Examiner: Prof. Dr. Stefan Görtz
Second Examiner: Prof. Dr. Ulrich Römer
Scientific Supervisor: Dr. Andrea Garbo

start of thesis: 14. September 2022
end of thesis: 13. March 2023

DLR e. V. Institut für Aerodynamik und Strömungstechnik
Lilienthalplatz 7, 38108 Braunschweig

Ihr Gesprächspartner Prof. Dr. S. Görtz

Telefon 0531
Telefax 0531 295-3357
E-Mail stefan.goertz@dlr.de

12. September 2022

Description of Master Thesis

Assessment of hyper-reduction techniques in the context of CFD-based intrusive reduced order modeling

Student Name: Cedric Fallet
Enrollment number: 4808435
Duration: 14.09.2022 – 13.03.2023
First Examiner: Prof. Dr. Stefan Görtz
Second Examiner: Prof. Dr. Ulrich Römer
Scientific Supervisor: Dr. Andrea Garbo

Reduced order model techniques (ROMs) have been proposed and extensively studied by the aerospace engineering community with the goal of replacing expensive computational fluid dynamics (CFD) analyses by faster approximate models without causing a perceivable degradation in prediction accuracy. Of particular interest are intrusive ROM techniques that rely on a CFD solver to keep a high-fidelity representation of the physics within their formulation. At the DLR Institute of Aerodynamics and Flow Technology, a parametric, intrusive ROM based on the DLR CFD solver TAU has been successfully implemented and extensively applied to several steady and unsteady application scenarios. However, the TAU solver has some intrinsic limitations that prevent an effective implementation of state-of-the-art ROM techniques, in particular hyper-reduction algorithms. Because of limited literature about meaningful application of hyper-reduction techniques, it is challenging to quantify the real benefit (if it exists) in terms of computational time reduction of such methods.

The main goal of this Master Thesis is to create an effective interface between the available ROM algorithm implemented on DLR's SMARTy library and the new DLR CFD solver CODA that would allow to assess the benefit of hyper-reduction techniques for high-dimensional aerospace problems. The student will first familiarize himself with the ROM formulation and implementation to successfully interface the existing ROM code with the CFD solver CODA in memory based on the FlowSimulator simulation environment. Then, the interface will be extended to enable the use of state-of-the-art hyper-reduction techniques, including the discrete empirical interpolation method (DEIM) and an advanced greedy algorithm. The resulting framework will be finally tested on industrial-relevant application problems (2D and possibly 3D) in order to assess the effectiveness of the ROM techniques.

Aufgabenstellung

Bewertung von Hyper-Reduktionsverfahren im Kontext von CFD-basierten intrusiven Modellen reduzierter Ordnung

Es wurden bereits einige Techniken für Modelle reduzierter Ordnung (ROMs) präsentiert und intensiv von der Forschungsgemeinschaft der Luft- und Raumfahrttechnik mit dem Ziel untersucht, aufwändige CFD Analysen durch approximative Modelle zu ersetzen, ohne in Folge dessen, deutliche Verschlechterungen in der Vorhersagegenauigkeit hervorzurufen. Von besonderem Interesse sind intrusive ROM-Techniken, welche auf CFD Lösern basieren, um eine hochgenaue Abbildung der Physik in ihren Formulierungen beizubehalten. Am Institut für Aerodynamik und Strömungstechnik des DLR wurde ein parametrisches, intrusives Modell reduzierter Ordnung basierend auf dem DLR CFD Löser TAU erfolgreich implementiert und umfassend für verschiedene stationäre und instationäre Anwendungsfälle eingesetzt. Dennoch hat der TAU Löser einige wesentliche Einschränkungen, die eine effektive Implementierung von aktuellen ROM-Techniken, insbesondere Hyper-Reduktionsalgorithmen, verhindern. Aufgrund eines geringen Bestandes an Literaturquellen zur sinnvollen Anwendung von Hyper-Reduktionsverfahren, ist es schwierig, den realen Nutzen (falls vorhanden) hinsichtlich der Reduktion von Rechenzeiten für solche Methoden abzuschätzen. Das primäre Ziel dieser Masterarbeit ist es, eine effektive Schnittstelle zwischen dem bestehenden, in der SMARTy-Bibliothek des DLR implementierten ROM-Algorithmus und dem neuen CFD Löser CODA zu implementieren, welche es ermöglicht, den Nutzen von Hyper-Reduktionsverfahren für hochdimensionale Probleme der Luft- und Raumfahrt zu bewerten. Der Student wird sich zuerst mit der Formulierung und Implementierung des Modells reduzierter Ordnung vertraut machen, um, basierend auf der FlowSimulator Simulationsumgebung, den existierenden ROM Programmcode mit dem CFD Löser CODA speicherbezogen zu verbinden. Anschließend wird die Schnittstelle für die Nutzung von modernen Hyper-Reduktionsverfahren erweitert, zu denen die discrete empirical interpolation method (DEIM) und ein fortgeschrittener Greedy-Algorithmus gehören. Die resultierende Schnittstelle wird abschließend mit dem Ziel, die Leistungsfähigkeit der ROM-Techniken zu beurteilen, anhand industriell relevanter Anwendungsfälle (2D und möglicherweise 3D) getestet.

Abstract

The aircraft design and optimization process relies on an extensive numbers of computations for a wide range of parameters defining flight conditions, mass cases or shape variations. As the deployment of high-fidelity methods like computational fluid dynamics (CFD) is still too expensive for such multi-query scenarios, reduced order models (ROMs) are a popular approach to reduce the computational costs while retaining sufficient accuracy levels. ROMs are usually based on a low-dimensional representation of the full order model (FOM), that is utilized to map a set of input parameters to an approximative solution of the FOM. This thesis investigates a physics-based ROM that seeks an optimized representation of the FOM in a proper orthogonal decomposition (POD) reduced space by minimizing the steady residual obtained from a CFD solver. The so called least squares ROM (LSQ-ROM) is extended by a consistent hyperreduction, which aims for a reduction of the entries of the residual vector that is minimized during the prediction. In particular, hyperreduction has been proposed by former studies on LSQ-ROM in order to decouple the algorithm computational complexity from the problem size. However, limitations within the DLR's CFD solver TAU prevent the consistent application of the hyperreduction. The CFD solver "*CFD for ONERA, DLR and AIRBUS*" (CODA), which is currently under development, allows the implementation and investigation of a consistent hyperreduction that effectively removes the dependency on the original problem size. The main goal of this thesis is the implementation and the performance assessment of a consistent hyperreduction for the LSQ-ROM that is coupled with the solver CODA and based on a reduced CFD mesh. The reduced mesh is identified by a set of hyperreduction indices selected by the discrete empirical interpolation method (DEIM) and missing point estimation (MPE) and allows a direct reduction of the effort for the residual evaluation in the CFD solver CODA. For an assessment of the performance of the hyperreduction with respect to accuracy and prediction time, the consistent hyperreduction implementation is applied to the steady flow prediction of two 2D test cases in the subsonic and transonic regime and one 3D test case in the transonic regime. It can be shown that the implemented hyperreduction effectively reduces the time for the predictions while causing only minor accuracy deterioration. The results highlight that the new hyperreduction is superior to the former implementation of the hyperreduction. In particular, for high reduction levels, the consistent hyperreduction becomes significantly faster than the former one with speed-up factors of around 5 for the 2D test cases and up to 25 for the 3D test case.

Contents

List of figures	iii
List of tables	vi
1 Introduction	7
1.1 The role of reduced order modelling	7
1.2 The LSQ-ROM	9
1.3 A consistent hyperreduction	9
1.4 Scientific objective	10
1.5 Outline of the thesis	10
2 Theoretical foundations	12
2.1 Full order model: CFD-based solution algorithm	12
2.1.1 Discretization of the governing equations	13
2.2 Proper Orthogonal Decomposition	15
2.3 Thin plate spline interpolation	19
2.4 Hyperreduction techniques	20
2.4.1 Discrete Empirical Interpolation Method	21
2.4.2 Missing Point Estimation	22
2.4.3 Artificial hyperreduction indices	27
2.5 Optimization algorithms	27
2.5.1 Nonlinear Least-squares problem	29
2.5.2 Levenberg-Marquardt method	31
2.6 Algorithm of the LSQ-ROM	33
2.6.1 Offline stage	33
2.6.2 Prediction stage	36
3 Implementation and validation approaches	39
3.1 Software tools and simulation framework	39
3.1.1 SMARTy	39
3.1.2 CFD solver: TAU and CODA	40
3.1.3 FlowSimulator environment	40
3.2 Implementations within SMARTy and CODA	41
3.2.1 LSQ-ROM method with CODA as the flow solver	42
3.2.2 Hyperreduction for the LSQ-ROM with CODA	43
3.3 Validation and investigation test cases	47
3.3.1 Meshes for the application in the CFD solver	48
3.3.2 Settings for the application of the LSQ-ROM	50
3.3.3 Prediction error metrics	51
4 Results	53
4.1 Validation of the LSQ-ROM coupled with CODA	53
4.1.1 NLR7301: subsonic flow	53
4.1.2 RAE2822: transonic flow	60
4.1.3 Assessment of the validation results	68
4.2 Investigation of the hyperreduction	69
4.2.1 NLR7301: subsonic flow	69

4.2.2	RAE2822: transonic flow	80
4.2.3	NASA-CRM: three-dimensional, transonic flow	92
5	Conclusion	107
	Bibliography	114
A	NLR7301	115
B	RAE2822	118
C	NASA-CRM	122

List of Figures

2.1	Different stencils for the residuum in cell i .	15
2.2	Schematic representation of the offline phase of the LSQ-ROM.	35
2.3	Schematic representation of the prediction phase of the LSQ-ROM.	38
3.1	The FlowSimluator environment.	41
3.2	CODA class diagram.	42
3.3	Submesh extraction for the hyperreduction of the LSQ-ROM.	44
3.4	Submesh generated from a base mesh for the NLR7301 airfoil.	46
3.5	Mesh and geometry of the NLR7301 airfoil.	48
3.6	Mesh and geometry of the RAE2822 airfoil.	49
3.7	Mesh and geometry of the NASA-CRM configuration.	50
4.1	Design space for the prediction of the subsonic flow around the NLR7301.	54
4.2	NLR7301: Mean and POD modes of Density from CODA (left) and TAU (right) solutions.	55
4.3	NLR7301: Three POD modes of Density from CODA (left) and TAU (right) solutions.	56
4.4	NLR7301: Relative information content of POD bases.	57
4.5	NLR7301: CFD reference solutions at $M = 0.25$ and $\alpha = 6^\circ$.	58
4.6	NLR7301: LSQ-ROM solutions for $M = 0.25$ and $\alpha = 6^\circ$ and a compression rate of 0.05.	58
4.7	NLR7301: Relative error in pressure for a predicted solution with 2, 10 and 39 POD modes.	59
4.8	NLR7301: Relative prediction time and objective function value over compression rate.	60
4.9	Design space for the prediction of the transonic flow around the RAE2822.	61
4.10	RAE2822: Mean and first two POD modes of Density from CODA (left) and TAU (right) solutions.	62
4.11	RAE2822: Third to fifth POD mode of Density from CODA (left) and TAU (right) solutions.	63
4.12	RAE2822: Relative information content of POD bases.	64
4.13	RAE2822: CFD reference solutions at $M = 0.75$ and $\alpha = 3^\circ$.	65
4.14	RAE2822: Solutions predicted with the LSQ-ROM at $M = 0.75$ and $\alpha = 3^\circ$ and with a compression rate of 1.0.	65
4.15	RAE2822: Relative error in pressure of the LSQ-ROM at $M = 0.75$ and $\alpha = 3^\circ$ and with a compression rate of 1.0.	66
4.16	RAE2822: Pressure distributions of the LSQ-ROM prediction at $Ma = 0.75$ and $\alpha = 3^\circ$.	67
4.17	RAE2822: Relative prediction time and objective function value over compression rate.	67
4.18	NLR7301: Heatmaps for the validation of the test point.	70
4.19	NLR701: Absolute relative error in pressure of the test point pediction.	71
4.20	NLR7301: Coefficient of determination for different compression rates.	72
4.21	NLR7301: Coefficient of determination for solutions of the LSQ-ROM and TPS for different compression rates.	73
4.22	NLR7301: Submeshes utilized for the consistent hyperreduction implementation.	74

4.23	2D: Relative number of submesh cells depending on the relative number of selected indices.	75
4.24	NLR7301: Influence of the hyperreduction level on the time for the evaluation of the residual.	76
4.25	NLR7301: Relative prediction times for different compression rates.	77
4.26	NLR7301: Relative prediction times per objective function evaluation for different compression rates.	78
4.27	NLR7301: Speed-up factor for the consistent hyperreduction.	79
4.28	RAE2822: Heatmaps for the validation of the test point.	81
4.29	RAE2822: Absolute relative error in pressure for predictions with different reduction levels.	82
4.30	RAE2822: Pressure distributions for different reduction levels of the LSQ-ROM.	83
4.31	RAE2822: Coefficient of determination for the total energy field and the surface pressure for different compression rates.	84
4.32	RAE2822: Visualization of the cells referring to sets of hyperreduction indices.	85
4.33	RAE2822: Coefficient of determination for solutions of the LSQ-ROM and POD + TPS for different compression rates.	85
4.34	RAE2822: Pressure distributions from POD + TPS.	86
4.35	RAE2822: Residual fields for the POD + TPS and LSQ-ROM predictions.	87
4.36	RAE2822: Pressure distributions from the LSQ-ROM and POD + TPS for an extrapolation point.	89
4.37	RAE2822: Relative prediction times for different compression rates.	90
4.38	RAE2822: Influence of the hyperreduction level on the normalized prediction time.	91
4.39	Design space for the prediction of the transonic flow around the CRM.	93
4.40	NASA-CRM: Heatmaps for the validation of the test point.	94
4.41	NASA-CRM: Pressure distributions of the FOM and the unreduced LSQ-ROM at $M = 0.85$ and $\alpha = 3.0^\circ$	95
4.42	NASA-CRM: Pressure distributions for a slice at 50% of the wing span.	95
4.43	NASA-CRM: Error of the surface pressure coefficient for a compression rate of 1.0.	96
4.44	NASA-CRM: Error of the surface pressure coefficient for a compression rate of 0.5.	97
4.45	NASA-CRM: Coefficient of determination for the surface pressure for different compression rates.	98
4.46	NASA-CRM: Coefficient of determination for the surface pressure of the LSQ-ROM and POD+TPS for different compression rates.	99
4.47	NASA-CRM: Error of the surface pressure coefficient for solutions of the LSQ-ROM and POD+TPS.	100
4.48	NASA-CRM: Submeshes utilized for the consistent hyperreduction.	101
4.49	3D: Relative number of submesh cells depending on the relative number of selected indices.	102
4.50	NASA-CRM: Time spent for the evaluation of the residual / objective function.	103
4.51	NASA-CRM: Relative prediction times for different compression rates.	104
4.52	NASA-CRM: Speed-up factors of the consistent hyperreduction for different compression rates.	105
A.1	NLR7301: Coefficient of determination for the predicted density field.	116
A.2	NLR7301: Coefficient of determination for the predicted field of the momentum in x-direction.	116
A.3	NLR7301: Coefficient of determination for the predicted field of the momentum in z-direction.	117

B.1	RAE2822: Coefficient of determination for the predicted density field. . . .	119
B.2	RAE2822: Coefficient of determination for the predicted field of the momentum in x-direction.	119
B.3	RAE2822: Coefficient of determination for the predicted field of the momentum in z-direction.	120
B.4	RAE2822: Speed-up factors for the two hyperreduction implementations. .	120
B.5	RAE2822: Relative error of the Total Energy for the POD + TPS and LSQ-ROM predictions.	121
C.1	NASA-CRM: pressure distributions at 25% of the wing span for a LSQ-ROM with a compression rate of 1.0.	123
C.2	NASA-CRM: pressure distributions at 75% of the wing span for a LSQ-ROM with a compression rate of 1.0.	123
C.3	NASA-CRM: pressure distributions at 25% of the wing span for a LSQ-ROM with a compression rate of 0.5.	124
C.4	NASA-CRM: pressure distributions at 75% of the wing span for a LSQ-ROM with a compression rate of 0.5.	124

List of Tables

2.1	DEIM algorithm	22
2.2	Fast MPE algorithm	26

Chapter 1

Introduction

1.1 The role of reduced order modelling

The design, control and optimization processes in nearly every technical field rely on methods, which have developed in accuracy and complexity at the same time. Let it be the generally known progress in image and signal processing, modelling of chemical or biological systems or the analysis of heat transfer and fluid dynamics. The systems and the amount of data has become larger and more complex and coincidentally the effort to obtain insights in form of accurate and robust simulations [22].

An illustrative example for the rising effort and complexity in system analysis in the field of aerospace engineering is the growing importance of high fidelity CFD simulations in the design and optimization of aircraft. In order to reduce the resources required by expensive experimental campaigns and to reduce the uncertainty of critical design choices, high-fidelity, numerical methods like CFD become more relevant. Therefore, to reach sufficient accuracy the models and methods used in CFD become more sophisticated and the overall problem size can easily reach hundreds of millions of degrees of freedom. At the same time, there is a trend towards an application of CFD in earlier phases of the design process. In combination with the tremendous number of flight conditions tested during design or iterations during an optimization process, the restriction to high-fidelity methods becomes infeasible. Even with the capability of substantial parallelization, highly resolved CFD simulations can take hours to days for a single flight point and geometry. Investigating a whole flight envelope for different geometries and masses can require hundreds of these simulations and thus relying on massive computing power, which becomes too costly for preliminary design or detailed optimization problems [44]. So, even for steady simulations the application of CFD in a multi-query scenario can lead to computational times not sufficient for fast design decisions. Therefore, processes like multidisciplinary design analysis and optimization (MDAO) not only in the field of aerospace have to include methods, which are a compromise between sufficient prediction accuracy and acceptable time requirements.

An important role in overcoming the difficulty in achieving sufficient accuracy in a relatively small amount of time is played by surrogates. Surrogates are models that aim for reducing the computational effort in computationally costly applications, such as aircraft design, without causing a substantial degradation in the result accuracy. The broad range of surrogates can be divided into three different classes [55]. One class is constituted by data-fit models, which rely on interpolation or regression of simulation data. By fitting the model to the data, the model is trained to predict the behaviour of the underlying system based on some input parameters. Gaussian processes and neural networks are currently the most popular methods, which are applied to various aerodynamic problems like the prediction of pressure distributions or inverse geometrical airfoil design [11], [65], [78]. The second type of surrogates are hierarchical models that are based on the original system but apply simplifying approaches e.g for the physics or the discretization. These models reach from simplified-physics models, that e.g solve the potential equations instead

of the Navier-Stokes equations for the prediction of flow fields to multigrid approaches that are also utilized in CFD [50], [14]. The last category of surrogates is represented by ROMs. A reduced order model is a “characterization of a physical process, such that the essential behaviours of the process are captured with a relatively small number of DOFs” [17]. Projecting the underlying FOM onto an appropriate subspace is one common possibility to obtain a suitable reduced representation of the FOM for a ROM. In most cases, the application of a ROM can be divided into two stages. First, in a so-called “offline stage” the model is constructed based on training data from the FOM including all computations that can be performed without knowledge of the specific queries, the model will be used for.

These relatively high up front costs leads to a reduced representation of the system that is used in the second stage, i.e. the “online stage” or “prediction stage”, to predict the system response at different conditions. Since the prediction is based on the reduced representation, the online stage has relatively low evaluation cost and can thus effectively be used in a multi-query context. This thesis focuses on ROMs because they are extensively used for aerodynamic problems. This group of surrogates can be further divided into the groups of intrusive and non-intrusive ROMs. While intrusive ROMs retain a connection to the underlying FOM even in the online stage, non-intrusive models lack such incorporation of characteristics of the original system. The advantage of non-intrusive ROMs is that the FOM is run as a black-box during the offline stage only to collect the model training data. However, as the online stage is not bound to any representation of the system behaviour, the predictions can lead to undetected violations of the underlying physics. A common way of constructing a non-intrusive ROM is to represent the system output in a suitable low-order basis, which can be derived by various methods like rational interpolation methods or POD [55]. After that, the reduced basis is utilized to find a mapping from the input parameters to an approximation of the FOM output. Such a mapping can be realized e.g by any interpolation method. Models based on a POD basis and an interpolation for the estimation of the POD coefficients were extensively investigated e.g for the prediction of flows around airfoils and whole aircraft [72].

Intrusive models, on the other hand, keep a linkage to the system behaviour by incorporating a FOM representation into the model. Because of that connection to the FOM, intrusive model complexity often scales with the size of the original problem, therefore preventing an effective reduction of the computational effort or requiring special methods to circumvent the issue. Nevertheless, the intrusive character of such models ensures a connection to the physics of the system and facilitates the derivation of error estimates to assess the approximation quality of the model. One of the most established intrusive ROMs in the field of aerodynamics is the well-known Galerkin-projection [13] [36], often combined with a POD base. POD is applied to the training data collected during the offline stage in order to obtain a reduced representation of the system behavior. Then, the resulting orthogonal basis of a POD is applied to approximate the state quantities of the equations, often partial differential equations (PDEs), defining the FOM and it can serve at the same time as the low-dimensional subspace, the equations are projected onto. Through the Galerkin projection, the PDE can be transferred into a small system of ordinary differential equations (ODEs). However, this reduction method falls short in the presence of general nonlinear terms that remain dependent of the original problem size. In order to circumvent this difficulty, several methods were proposed to reduce the evaluation costs of the nonlinear term. These methods are also known as hyperreduction methods [7], which play an important role within this thesis and are presented in detail in section 2.4. Another famous intrusive ROM that is also based on some kind of hyperreduction is the GNAT method [10] [41]. The GNAT method is based on a Petrov-Galerkin projection of the system’s PDE, which is represented in a low-dimensional subspace. This results in a minimization problem, which is solved using Gauss-Newton and gappy POD [59] for the Approximated Tensors within the algorithm. Finally, another popular ROM within the German Aerospace Center (DLR) research group is the LSQ-ROM, which relies

on the direct minimization of the FOM residual in a low-dimensional POD subspace in order to determine the prediction POD coefficients. The LSQ-ROM and its acceleration through application of the hyperreduction approach is part of this thesis and presented in more detail in the next section.

1.2 The LSQ-ROM

The ROM investigated in this thesis is the least squares ROM. It is an intrusive, POD-based ROM that contains, in contrast to a non-intrusive ROM based on POD plus interpolation, an additional minimization process, which represents the linkage to the equations of the FOM. Hereinafter and without loss of generality, the description concentrates on an aerodynamic application with a CFD solver as a FOM. However, the same approach can be applied to different disciplines like stress analysis in structural engineering. For an aerodynamic application, the LSQ-ROM predicts the solution of the flow field, which is normally the result of a CFD computation. For the construction of the model, a POD basis is generated from set of CFD solutions, which span over the defined parameter space and represent the training snapshots; examples of common parameter for aerodynamic application cases are Mach number, Reynolds number or angles defining the flow direction. However, even shape parameters for the geometry are conceivable as input parameters. POD identifies a proper low-dimensional representation of the parameter space, which is used during the online stage for the prediction of a solution that is based on a linear combination of POD modes. The required POD coefficients are obtained in the LSQ-ROM technique by an optimization process that aims for an improved solution by minimizing the residual from the CFD solver with respect to the POD coefficients. As the residual can be understood as the (nonlinear) deviation from the numerically exact solution, a significant reduction of it should correspond to more accurate solution.

There are multiple investigations in the literature for steady aerodynamic problems that compare the accuracy of the LSQ-ROM with the solution obtained from a pure interpolation [63], investigate the extrapolation capabilities [62] or test a LSQ-ROM which is subject to aerodynamic constraints [64]. Especially for the prediction of extrapolation points and through incorporation of additional constraints into the optimization process, the intrusive characteristics of the LSQ-ROM lead to substantial improvements compared to non-intrusive interpolation based methods. As investigations indicated good accuracy in combination with significant time savings compared to the CFD solution, the LSQ-ROM was further extended towards unsteady simulations. The first promising results for the prediction of gust-load cases were shown in [50] and [3].

However, some of these studies highlight how the intrusive formulation of LSQ-ROM leads to a computational complexity that scales with the size of the original problem (e.g. the number of cells in the CFD grid). For this reason, an additional hyperreduction technique is usually integrated into the LSQ-ROM to pose a second reduction step. More specifically, hyperreduction application to the residual vector of the FOM promises high time saving potentials as the residual vector is still linked to the size of the original problem and a hyperreduction removes this dependency by evaluating the residual vector only for a subset of cells.

1.3 A consistent hyperreduction

The hyperreduction implemented so far, is not exhausting its full potential, as the reduction of the residual vector is performed after its full evaluation, which reduces solely the computational cost of linear algebra operations in the optimizer. Therefore, the

LSQ-ROM is still depending on the original problem size. In order to exploit the full acceleration potential of the hyperreduction approach, it is necessary to reduce the residual vector for its evaluation in the CFD solver. However, prior applications of the LSQ-ROM within DLR were based on DLR's CFD solver TAU, which does not have the capabilities to successfully perform a consistent hyperreduction approach.

This work instead makes use of the new CFD solver CODA, which is currently developed at the DLR together with other partners, to implement a consistent hyperreduction that computes the residual only for a subset of cells. As before, this subset is identified by the DEIM [68] and MPE [61]. The approach to compute the reduced residual is based on the construction of a reduced mesh that contains only the cells that are needed to correctly evaluate the residuals corresponding to the subset of hyperreduction indices. All residual evaluations are then performed solely on this submesh whose size scales with the number of indices.

Thus, the costs for the evaluation of the residual can be expected to scale with the size of the submesh and not with the original problem size anymore. Since the effort for the residual evaluation is linearly dependent on the size of the mesh or to be more precise on the number of faces contained in the mesh, the additional acceleration through application of the proposed hyperreduction is expected to linearly scale with the size of the submesh.

1.4 Scientific objective

The work aims for an assessment of the effectiveness and prediction accuracy of the chosen approach for a consistent hyperreduction not only for realistic reduction levels in the LSQ-ROM context, but also in a direct comparison with the former hyperreduction. Therefore, a successful implementation of the consistent hyperreduction is expected to significantly reduce the prediction time of the LSQ-ROM in its absolute magnitude, but also when comparing it with the computational cost of LSQ-ROMs based on the former hyperreduction implementation. In addition to the acceleration potential, the influence of the hyperreduction on the accuracy needs to be examined to assess the compromise between acceleration and accuracy.

For that, tests are conducted for the prediction of steady, turbulent flows around 2D airfoils in subsonic and transonic regimes as well as for a more realistic 3D aircraft configuration. The LSQ-ROM prediction processes are performed with different combinations of the reduction level for the reduced space and the hyperreduction. The analysis of the resulting error levels and the required computational time makes it possible to assess the implemented hyperreduction under realistic conditions, which should lead to realistic recommendations for the choice of proper reduction levels for future applications of the LSQ-ROM.

1.5 Outline of the thesis

After the presented motivation and background for the development of the LSQ-ROM in this chapter, the second chapter outlines the required theoretical background for the application of the LSQ-ROM. At first, some fundamental approaches within in the field of CFD are introduced, in order to assess their implications on the LSQ-ROM. These information mainly focus around approaches for the spatial discretization that determine the residual that is utilized in the LSQ-ROM. Following, basic methods from the field of reduced order modelling are introduced, that constitute the LSQ-ROM. The main components are POD, the hyperreduction technique, which is the MPE together with the

DEIM, and the optimization algorithm in form of the Levenberg-Marquardt (LM) algorithm. With the definition of these methods at hand, the second chapter is concluded by a description of the algorithm of the LSQ-ROM including the newly implemented hyperreduction method. For the consistent implementation of the hyperreduction an estimation for its acceleration potential is given that is verified during the investigations. The following chapter presents the software tools utilized in this work and the environment they are operating in. Moreover, the implementations for the application of the LSQ-ROM coupled with CODA and for the realization of the submesh-based hyperreduction are presented in more detail. After that, the geometries and meshes of the test cases are shown as well as settings for the CFD solver and LSQ-ROM for the subsequent presentation of the investigation results. The chapter for that is split into a section for the validation of the coupling of the LSQ-ROM with CODA and a section for the results concerning the consistent hyperreduction. The validation includes the comparison of LSQ-ROM predictions for a coupling with TAU and CODA on the basis of two 2D test cases at subsonic and transonic conditions. In the second section, that focuses on the hyperreduction, an additional 3D case is part of the examinations. For each test case, the influence of the hyperreduction on the accuracy and the acceleration of the LSQ-ROM is investigated. This also includes considerations of the POD reduction, the residual minimization procedure and assessments of the performance of the consistent hyperreduction with respect to the former hyperreduction. The findings of all the investigation are summarized in the final chapter besides some remarks concerning current limitations and future improvements for the consistent hyperreduction.

Chapter 2

Theoretical foundations

The following chapter introduces the most important methods and equations for the application of the LSQ-ROM method as a reduced order model for a CFD-based FOM. After a brief description of the governing equations of the FOM and their discretization within the CFD solver, the key techniques for the ROM are presented. Two reduction methods, which are part of the LSQ-ROM method, are described in detail; first, POD for the identification of a reduced basis and second hyperreduction techniques to reduce the evaluation costs of nonlinear terms. Moreover, a spline based interpolation technique is presented. Next, the general optimization problem will be described and some basic approaches to solve it. The special case of Least-squares problem arising from the LSQ-ROM method and the applied algorithm to solve it, is presented in the same manner. Having all the key elements of the LSQ-ROM method at hand, an outline of the overall algorithm is given in the final section of this chapter.

2.1 Full order model: CFD-based solution algorithm

Defining the behaviour of a continuous fluid, the Navier-Stokes equations form the governing set of equations, which needs to be solved in order to obtain the solution of a specific flow problem. The set of nonlinear partial differential equations is based on the conservation laws for the mass, energy and the momentum in all three spatial dimensions. Assembling the conservative variables ρ (mass), $\rho\mathbf{v}$ (momentum) and ρE (energy) in the vector \mathbf{w} and integrating over the flow domain Ω , gives the Navier-Stokes equations in their integral form [34]

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{w}(\mathbf{x}, t) d\Omega + \int_{\partial\Omega} \mathbf{F}_c(\mathbf{w}) - \mathbf{F}_v(\mathbf{w}) ds = \int_{\Omega} \mathbf{Q}(\mathbf{w}) d\Omega \quad (2.1)$$

with

$$\mathbf{w} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}$$

where u, v, w are the components of the velocity vector \mathbf{v} . eq. (2.1) consists of the term for temporal change of the conservative variables, the convective flux vector \mathbf{F}_c , the viscous flux vector \mathbf{F}_v and the sources \mathbf{Q} . The explicit formulation for the fluxes and the sources can be found in [34]. As this system of equations is not closed, there are at least two additional relations needed. Usually for gaseous fluids, the first is the perfect gas law

$$p = \rho R_i T \quad (2.2)$$

to obtain an additional equation for the pressure p and the temperature T . The second is the relation between the inner energy e and the temperature of a perfect gas.

$$e = c_v T \quad (2.3)$$

The specific gas constant R_i and the heat capacity c_v are defined by the chemical composition of the gas. They are often assumed to be constant, which is a good approximation for non-reactive flows below hypersonic flow speeds. Although fluxes and source terms include several other substance properties like the viscosity or the heat conduction coefficient, these can be modelled with the knowledge of the quantities of state and the specific type of the fluid.

Despite having a closed form for the description of a continuous flow at hand, there is an additional difficulty when discretizing it for the numerical solution of a turbulent flow. Since turbulent flows consist of flow phenomena ranging from the size of the whole flow domain down to a microscopic scale [26], a simulation resolving even the smallest scales of the turbulence becomes intractable for industrial application cases. A common approach is to perform a Reynolds-averaging of the Navier-Stokes equation, resulting in the Reynolds averaged Navier-Stokes (RANS) equations. This results in a time averaged influence of the turbulent fluctuations, which is represented in the RANS equations by the Reynolds-stress tensor τ_{turb} . As this tensor contains six unknown fluctuation components, an additional approach is needed to model this stress tensor. Often times, the Boussinesq hypothesis is used for modelling approaches like the well known one equation model of Spalart and Allmaras [71]. The Boussinesq hypothesis claims an analogy of the turbulent stress tensor to the shear stress tensor but with an additional free parameter, the turbulent dynamic viscosity μ_t . The turbulent model of Spalart and Allmaras (SA model) consists of a transport equation for the turbulent viscosity ν_t , which needs to be solved additionally to the five equations for the conservation of mass, momentum and energy. However, this sixth equation can be discretized in a similar way like the others and can either be solved loosely coupled with defined update intervalls between the equations or strongly coupled as one system of equations.

2.1.1 Discretization of the governing equations

Until now there is no analytical solution to the full Navier-Stokes equations or the RANS equations known. Therefore, a classic approach for the computation of an approximative solution corresponding to a specific flow problem, defined by its boundary conditions, is to discretize the equations separately in the spatial and temporal dimension. A common strategy for the spatial discretization is the Finite-Volume approach, which divides the spatial flow domain into smaller control volumes. For every of these volume elements, often also referred to cells, one assumes a constant distribution of \vec{W} in every element and conservation like it is defined by the equations. The purpose of an iterative solution process is to drive the imbalances for the conservation laws in every element towards zero. For the definition of the volume elements, there are two established ways irrespective of the mesh being structured or unstructured: node-centered and cell-centered. With the node-centered approach (used in TAU), the volume element is constructed around each node of the mesh, whereas with the cell-centered approach (used in CODA), the volume elements are equal to the elements of the original mesh. As a result of this, one differs between the original mesh, which is a discrete representation of the flow domain, and the computational mesh that the CFD solver operates on. Moreover, the computational mesh is stored in a face-based manner, meaning that geometrical quantities needed for the integration over the volume elements are stored indirectly through the face elements forming the surface of every volume. This enables the computation of surface integrals efficiently via a loop over all faces.

The presented Finite-Volume discretization allows to replace the volume integrals in eq. (2.1) with multiplications of the cell volume and the surface integrals with sums over all faces of an element. These equation manipulations result in the semi-discretized formulation of eq. (2.1) for an arbitrary volume cell i :

$$\frac{\partial}{\partial t} \mathbf{w}_i \Omega_i + \sum_{j=1}^{n_i} (\mathbf{F}_c)_j \Delta \mathbf{s}_j - \sum_{j=1}^{n_i} (\mathbf{F}_v)_j \Delta \mathbf{s}_j = \mathbf{q}_i \Omega_i \quad (2.4)$$

All the terms representing the spatial discretization are normally combined in the residual R_i , which consists of the conservation imbalances for cell i . The global formulation for this is given by

$$\frac{\partial \mathbf{W}}{\partial t} = -\frac{1}{\Omega} \mathbf{R}(\mathbf{W}), \quad (2.5)$$

where \mathbf{R} is the residual vector, \mathbf{W} the vector of all conservative variables and Ω the mass matrix. This formulation is the starting point for the frequently used method of lines that treats spatial and temporal discretization separately [34] and enables the application of different discretization strategies for the spatial and temporal domain of the RANS or Navier-Stokes equations. Frequently used methods for the temporal discretization are Runge-Kutta time-stepping schemes, which are often applied in case of stationary problems via a pseudo time step [70]. For stationary problems, which are the focus of this work, the aim of the time-stepping scheme is to drive the residual vector towards zero, which automatically leads to $\frac{\partial \mathbf{W}}{\partial t} \approx 0$ and therefore to the solution of the stationary problem. However, for a further insight into the wide range of CFD solution algorithms the reader is referred to [34]. The same is true for the various discretization schemes of the convective flux, viscous flux and the sources, which form the residual. Nevertheless, some remarks are given because of the importance of the residual vector within the LSQ-ROM method.

An important feature of a discretization scheme is the stencil it implies. For example, the computation of the fluxes across the surfaces of a cell requires quantities at the surface. Since in a Finite-Volume approach the conservative variables are only given in the cell centers, many discretization schemes use an approximation of the flux based on the values of the neighbouring cells. As a result of the summation of all fluxes (see eq. (2.4)) for the residual, the residual R_i becomes dependent of the direct neighbourhood $N(i)$. The neighbourhood $N(i)$ forms the stencil of cell i and is defined by all cells, that share a face with this cell. Numerical schemes that give a discretization with a convergence order that is higher than one can even require an extended stencil. The approximative flux across a face is then also influenced by the neighbours of the face neighbouring cells, and the residual even by its second neighbours.

$$\text{basic stencil:} \quad \mathbf{R}_i = \mathbf{R}_i(N(i)) \quad (2.6)$$

$$\text{extended stencil:} \quad \mathbf{R}_i = \mathbf{R}_i(N(N(i))) \quad (2.7)$$

Figure 2.1 illustrates different stencils for the residual in an arbitrary cell i . It is possible to see that the number of cells in the stencil is not only influenced by the type of the stencil but also by the cell types, which defines the number of direct neighbours. Another important point concerning the residual is that every change of the underlying spatial discretization influences the solution of the flow problem. As the stationary solution is the root of the residual function, one has to expect a different solution from a different residual definition. At the same time, when separating spatial and temporal discretization, one can adapt the latter without influencing the final, stationary solution.

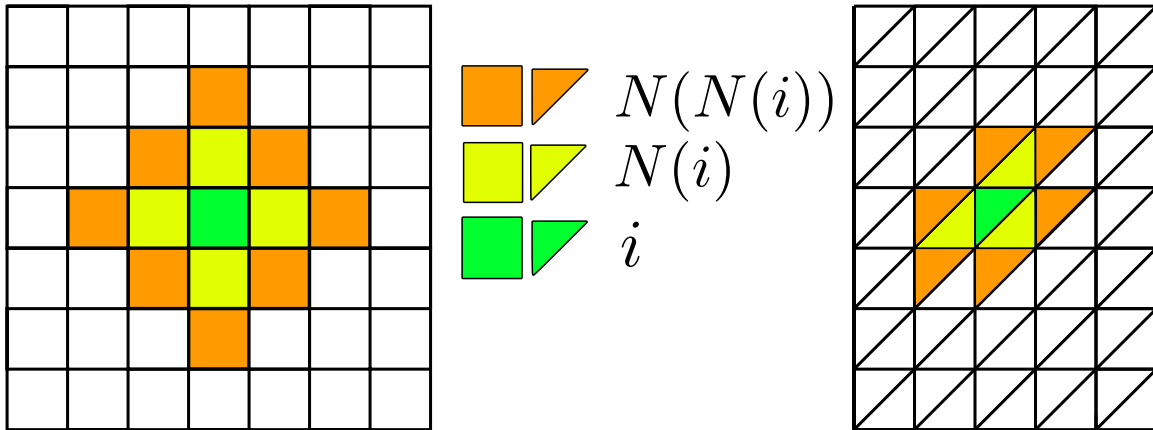


Figure 2.1: Different stencils for the residuum in cell i .

2.2 Proper Orthogonal Decomposition

The proper orthogonal decomposition (POD) has become one of the most widely used techniques in the field of reduced order modeling with various application cases. Often used for defining a reduced basis in projection-based reduced models, its applicability is unrestricted by the underlying system. Therefore, POD, also known as Principal Component Analysis (PCA) in the field of statistics, can be applied to linear and nonlinear problems. With its first application for the analysis of turbulent flows [54], the decomposition is utilised in various fields like image processing [6], chemical reaction systems [67], aeroelasticity [73] or structural mechanics [69]. The underlying idea is that the points of a given dataset lie sufficiently close to a manifold of lower dimensionality [12]. If the dataset has indeed a lower intrinsic dimensionality than the individual data points, the aim is to find a proper description of the defining subspace. Thus the complexity of the dataset can be reduced without pruning the most relevant information. POD decomposes a dataset into a set of orthogonal modes, that contain a decreasing amount of information of the original dataset. The decreasing content of information of the specific modes can then be utilised by neglecting a certain number of modes. Cutting off just the modes with the lowest information content ensures that the remaining modes are still able to adequately describe the behaviour of the system the original dataset was generated from. There are several approaches to derive the POD modes. The following derivation is based on a linear projection that minimizes the error between the data points of the dataset and their projection onto the subspace, spanned by the POD modes.

Given the dataset denoted by the matrix $\mathbf{Y} \in \mathbb{R}^{n \times m}$, which contains the data points $\mathbf{y}_i \in \mathbb{R}^n$ as column vectors, the projection of an arbitrary vector \mathbf{y} onto the d -dimensional subspace $\mathbb{V} \subset \mathbb{R}^n$ is defined by

$$\tilde{\mathbf{y}} = \sum_{j=1}^d \langle \mathbf{y}, \mathbf{u}_j \rangle \mathbf{u}_j \quad (2.8)$$

where $\langle \cdot, \cdot \rangle$ represents the dot product and \mathbf{u}_j the normalised j -th basis vector of the reduced basis, represented by the matrix $\mathbf{U} \in \mathbb{R}^{n \times d}$. The squared summed error between the column vectors \mathbf{y}_i of \mathbf{Y} and their projections $\tilde{\mathbf{y}}_i$ is defined as

$$E = \sum_{i=1}^m |\mathbf{y}_i - \tilde{\mathbf{y}}_i|^2. \quad (2.9)$$

This error can be minimized through a proper choice of the d basis vectors \mathbf{u}_j which are also referred as the POD modes. Since the basis vector should minimize the projection

error, they have to meet the following condition [12]

$$\frac{\partial E(\mathbf{U})}{\partial \mathbf{u}_j} = 0. \quad (2.10)$$

The arising Least-squares problem

$$\min_{\mathbf{u}_1, \dots, \mathbf{u}_d \in \mathbb{R}^n} \left| \mathbf{y} - \sum_{j=1}^d \langle \mathbf{y}, \mathbf{u}_j \rangle \mathbf{u}_j \right|^2 = \min_{\alpha \in \mathbb{R}^n} |\mathbf{y} - \mathbf{U}\alpha|^2 \quad (2.11)$$

with $\alpha = \mathbf{U}^T \mathbf{y}$ illustrates that the projection defined by the matrix \mathbf{U} is an orthogonal projection (in case of orthonormal basis vectors \mathbf{u}_i) yielding the best approximation of \mathbf{y} in the reduced space \mathbf{U} . Assuming normalized basis vectors, the minimization problem can be transferred into a simpler maximization problem, which aims for obtaining the basis vector \mathbf{u}_j giving the best approximation to all vectors of \mathbf{Y} . Therefore, instead of minimizing the projection error, the projection of all vectors \mathbf{y}_i onto the basis vectors is maximized:

$$\max_{\mathbf{u} \in \mathbb{R}^n} \sum_{i=1}^n \sum_{j=1}^d |\langle \mathbf{y}_i, \mathbf{u}_j \rangle|^2 \quad (2.12)$$

Solving the maximization problem requires a constraint through a Lagrange multiplier λ for the term $1 - |\mathbf{u}_j|^2$, which prevents the trivial solution $u_j \rightarrow \infty$ for the optimization.

$$L(u, \lambda) = \sum_{i=1}^n \left(\sum_{j=1}^d |\langle \mathbf{y}_i, \mathbf{u}_j \rangle|^2 + \sum_{j=1}^d \lambda (1 - |\mathbf{u}_j|^2) \right) \quad (2.13)$$

Differentiating the above expression for the maximization with respect to the arbitrary basis vector u_k yields the following (cf. [66]):

$$\frac{\partial L(\mathbf{u}, \lambda)}{\partial \mathbf{u}_k} = \frac{\partial}{\partial \mathbf{u}_k} \left(\sum_{i=1}^n \left(\sum_{j=1}^d |\langle \mathbf{y}_i, \mathbf{u}_j \rangle|^2 + \sum_{j=1}^d \lambda (1 - |\mathbf{u}_j|^2) \right) \right) \quad (2.14)$$

$$= 2 \left(\sum_{i=1}^n \langle \mathbf{y}_i, \mathbf{u}_k \rangle \mathbf{y}_i^T - \lambda \mathbf{u}_k \right) = 0 \quad (2.15)$$

Thus, for every \mathbf{u}_k with $k = 1, \dots, d$, the following eigenvalue problem holds

$$\mathbf{Y}\mathbf{Y}^T \mathbf{u}_k = \lambda \mathbf{u}_k. \quad (2.16)$$

This eigenvalue problem needs to be solved in order to obtain the basis that minimizes the projection error of \mathbf{Y} onto the subspace \mathbb{V} . It follows that the basis vectors of the reduced space \mathbf{U} are exactly the eigenvectors of the positive definite matrix $(\mathbf{Y}\mathbf{Y}^T)$. This matrix is called the Gramian matrix, which has n non-negative eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$, because the Gramian is positive-definite and of rank $r = \text{rank}(\mathbf{Y})$ [31].

For the case that \mathbf{U} forms a reduced space with $d < m$, the displacement between a vector \mathbf{y} and its approximation in the reduced space $\hat{\mathbf{y}}$ lies in the space orthogonal to the reduced subspace. The error of the projection can be quantified by the sum of the eigenvalues corresponding to the vectors laying orthogonal to the subspace [12].

$$\sum_{i=1}^m \left| \mathbf{y}_i - \sum_{j=1}^d \langle \mathbf{y}_i, \mathbf{u}_j \rangle \mathbf{u}_j \right|^2 = \sum_{j=d+1}^m \lambda_j \quad (2.17)$$

This information can be utilized to define a measure for the amount of information captured by the reduced subspace. A often used definition is the relative information content (RIC)

$$\text{RIC} = \frac{\sum_{i=1}^d \lambda_i}{\sum_{i=1}^m \lambda_i} \quad (2.18)$$

which can be understood as the ratio of information modeled by the subspace to the total information contained in the system \mathbf{Y} . In the context of fluid flows, where \mathbf{Y} contains information about the velocity of the flow, the eigenvalues correspond to the fluid's kinetic energy associated with the POD mode of the specific eigenvalue [5]. Considering that the eigenvalues are sorted, a reduction of system \mathbf{Y} by $m - d$ dimensions neglects exactly those modes which contain the smallest amount of information of the original system. POD modes d in the reduced basis divided by the maximum number of POD modes m :

$$\text{compression rate} = \frac{d}{m} \quad (2.19)$$

So a compression rate of 1.0 refers to a full POD basis, while lower compression rates refer to reduced POD bases.

Computation of the POD modes To compute the proper orthogonal decomposition of \mathbf{Y} and obtaining the reduced space from it, one has to compute the eigenvectors and eigenvalues of $\mathbf{Y}\mathbf{Y}^T$. Since common algorithms to compute the eigenvalue decomposition scale cubic with the largest dimension of the matrix [27], the effort to compute the eigenvalue decomposition (EVD) of $\mathbf{Y}\mathbf{Y}^T \in \mathbb{R}^{n \times n}$ would be $\sim n^3$. This method is called the *direct method* [35]. In case of large data vectors $y \in \mathbb{R}^n$, which contain for example the solution of a flow field with thousands of values, this method becomes infeasible. However, since the rank r of \mathbf{Y} is $r = \min(n, m)$, one has to compute only r pairs of eigenvector and eigenvalues while: $u_{j=r+1}^{\max(n)} = 0$. Therefore [43] proposed in case of $n < m$ to compute the EVD of the matrix $\mathbf{Y}^T\mathbf{Y} \in \mathbb{R}^{m \times m}$, leading to the so called *method of snapshots*. To obtain the same eigenvectors as the *direct method*, the eigenvectors \mathbf{v}_i from the *method of snapshots* have to be rescaled in the following way [60].

$$\mathbf{U} = \mathbf{Y}\mathbf{V}\sqrt{\mathbf{\Lambda}}^{-1} \quad (2.20)$$

Herein $\mathbf{V} \in \mathbb{R}^{m \times d}$ is the matrix with the column-wise stacked eigenvectors \mathbf{v}_i and $\mathbf{\Lambda} \in \mathbb{R}^{d \times d}$ is a square matrix containing the corresponding eigenvalues on the main diagonal. The connection between the two presented methods of computing the POD of the matrix \mathbf{Y} can be found by application of the singular value decomposition (SVD) to \mathbf{Y} , which represents another method of obtaining the eigenvectors and eigenvalues of the matrix $\mathbf{Y}\mathbf{Y}^T$. The *Singular Value Decomposition* decomposes a real matrix $\mathbf{C} \in \mathbb{R}^{n \times m}$:

$$\mathbf{C} = \mathbf{L}\mathbf{\Sigma}\mathbf{R}^T \quad (2.21)$$

into an $n \times n$ orthogonal matrix \mathbf{L} , an $m \times m$ orthogonal matrix \mathbf{R} , and an $n \times m$ diagonal matrix $\mathbf{\Sigma}$ with the diagonal entries $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$, with $r = \min(n, m)$, which are called singular values of \mathbf{C} [74]. Since all $\{\sigma_i\}_{r+1}^{\max(n,m)}$ of $\mathbf{\Sigma}$ are 0, one can derive a more compact form of the SVD, which is called *thin SVD* or *compact SVD*, which increases the efficiency of algorithms computing the decomposition of a matrix \mathbf{C} . Neglecting all columns of the matrices \mathbf{L} and \mathbf{R} that correspond to singular values equal to zero, the *thin SVD* takes the form [74]:

$$\mathbf{C} = [\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_r] \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r) [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r]^T \quad (2.22)$$

Applying the SVD of \mathbf{Y} to $\mathbf{Y}\mathbf{Y}^T$ results in the following expression:

$$\mathbf{Y}\mathbf{Y}^T = (\mathbf{L}\mathbf{\Sigma}\mathbf{R}^T) (\mathbf{L}\mathbf{\Sigma}\mathbf{R}^T)^T = (\mathbf{L}\mathbf{\Sigma}\mathbf{R}^T) (\mathbf{R}\mathbf{\Sigma}^T\mathbf{L}^T) = \mathbf{L}\mathbf{\Sigma}\mathbf{\Sigma}^T\mathbf{L}^T \quad (2.23)$$

Comparing the last term with the EVD

$$\mathbf{Y}\mathbf{Y}^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \quad (2.24)$$

shows that \mathbf{L} of the SVD corresponds to the matrix of the POD modes (eigenvectors) \mathbf{U} , while the term $\mathbf{\Sigma}\mathbf{\Sigma}^T$ represents the matrix of the eigenvalues $\mathbf{\Lambda}$. Therefore, the SVD constitutes a third option to calculate the POD, besides the *direct method* and the *method of snapshots*.

Another point to mention is the common approach to center the dataset \mathbf{Y} around its mean value $\bar{\mathbf{y}}$, which is calculated by

$$\bar{\mathbf{y}} = \frac{1}{m} \sum_{i=1}^m \mathbf{y}_i \quad (2.25)$$

and is then subtracted column-wise from \mathbf{Y} . The main reason for this step is often the desire of similar magnitudes for the entries within a snapshot \mathbf{y}_i and its POD modes \mathbf{U} , which are usually further processed in the framework of a ROM. Although this centering influences the physical interpretation of the POD modes, the overall procedure stays the same. However, it is essential to notice, that such centering lowers the rank of \mathbf{Y} by one.

Interpretation and application of POD modes Having multiple methods at hand to compute the proper orthogonal decomposition of an arbitrary dataset, its of particular interest to interpret the meaning of the orthogonal modes. In general, without any specific information of the system from which the dataset or snapshots were generated, the modes define the directions of a subspace, which give the best approximation of the original dataset. Considering that the modes are ordered by their content of information with respect to the dataset, analysing the first few dominant POD modes could possibly reveal the most important features of the whole dataset. The condition for that is a rapidly decreasing information content along the modes, as the dataset not necessarily lies on a manifold, which can be approximated with only a few POD modes. However, in many application cases one can observe such a decrease [5]. With the information about the directions of the most important features, a powerful technique in analysing a snapshot is to project it onto the POD modes (reduced space in case of already neglected modes). The snapshot can then be reconstructed by a linear combination of the projections onto the orthonormal system spanned by the POD modes. This allows to identify the most prominent characteristics of the snapshot due to the normally high contribution of the projections of the dominant modes. Given the orthonormal POD modes \mathbf{U} , the projection of a snapshot \mathbf{y}_i is defined by the coefficient vector \mathbf{a} , whose components are

$$a_i = \mathbf{y}_i^T \mathbf{u}_i. \quad (2.26)$$

The coefficients can be understood as the representation of the snapshot in the orthonormal system. As long as the orthonormal basis \mathbf{U} has the full rank of the dataset \mathbf{Y} , the projection followed by the reconstruction yields the exact snapshot. The coefficients of every projected snapshot \mathbf{y}_i can be combined in the coefficient matrix $\mathbf{A} \in \mathbb{R}^{m \times d}$

$$\mathbf{A} = \mathbf{Y}^T \mathbf{U} \Rightarrow \mathbf{Y} = \mathbf{U} \mathbf{A}^T \quad (2.27)$$

The full set of snapshots can then be reconstructed by multiplication of the POD modes with the transposed of the coefficient matrix. This property is due to the orthonormality of the modes ($\Rightarrow \mathbf{A}^T = \mathbf{A}^{-1}$). The information of a snapshot \mathbf{y}_i , which is held by the k -th POD mode can then be computed in the following way:

$$\mathbf{y}_i^k = (\mathbf{a}_i)_k \mathbf{u}_k \quad (2.28)$$

The reconstruction of the snapshot results in the sum over the contributions of all modes:

$$\mathbf{y}_i = \sum_{k=1}^d \mathbf{y}_i^k = \sum_{k=1}^d (\mathbf{a}_i)_k \mathbf{u}_k \quad (2.29)$$

2.3 Thin plate spline interpolation

The Thin plate spline (TPS) interpolation is part of the investigated LSQ-ROM and is utilized for the generation of a suited set of initial POD coefficients for the optimization process (section 2.6). TPS is an interpolation and smoothing method based on splines represented by radial basis functions. These splines are defined in correspondence to the mechanical analogy of the bending of a thin plate [33], whose shape is defined by the spline function:

$$z = f(x, y) \quad (2.30)$$

As every interpolation method, TPS is based on a set of m known data values z_i at the control points $x_i, y_i; i = 1, \dots, m$. The basic characteristic of the thin plate splines is their definition according to an minimum energy principle. They have to minimize the external energy E_{ext} , which is induced by the displacement of the spline representation of the thin plate from the control point values z_i . So, the sum of all squared displacements constitutes the external energy as defined by

$$E_{\text{ext}} = \sum_{i=1}^m (z_i - f(x_i, y_i))^2 \quad (2.31)$$

This term needs to be minimized in order to find a solution for the spline based interpolation function $f(x, y)$. It is possible, to control the smoothness of the spline function by adding a second term to the energy minimization problem, which represents the internal bending energy of the plate function $f(x, y)$. The internal energy E_{int} is defined by [39]

$$E_{\text{int}} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 dx dy \quad (2.32)$$

and can be adapted by an scaling factor α to control its contribution to the total energy E_{tot} .

$$E_{\text{tot}} = E_{\text{ext}} + \alpha E_{\text{int}} \quad (2.33)$$

When α takes the value 0.0, TPS is an interpolation spline, otherwise it is an approximation function or regression method. The minimization problem of the total energy can be determined by setting its first derivative to 0. As a result, the Euler-Lagrange differential equation is obtained, which has the fundamental solution

$$f(\mathbf{r}) = \mathbf{r}^2 \ln(\mathbf{r}) \quad (2.34)$$

where \mathbf{r} defines the distance of two points in \mathbb{R}^2 [39]:

$$r_i^2 = (x - x_i)^2 + (y - y_i)^2 \quad (2.35)$$

Therefore, the interpolation spline in its representation of radial basis functions takes the form

$$f(x, y) = \sum_{i=1}^m b_i r_i^2 \ln(r_i) \quad (2.36)$$

and the approximation or regression function with $\alpha \neq 0.0$ is defined by

$$f(x, y) = \sum_{i=1}^m b_i r_i^2 \ln(r_i) + d_0 + d_1 x + d_2 y \quad (2.37)$$

The definition for the interpolation spline has m free parameters that need to be determined, while the approximation function includes three additional parameters; d_0, d_1 and d_2 . These parameters can be found by solving the linear system

$$\mathbf{N}\mathbf{c} = \mathbf{z} \quad (2.38)$$

which includes the matrix $\mathbf{N} \in \mathbb{R}^{(m+3) \times (m+3)}$

$$\mathbf{N} = \begin{bmatrix} \alpha & a_{12} & a_{13} & \dots & a_{1n} & 1 & x_1 & y_1 \\ a_{21} & \alpha & a_{23} & \dots & a_{2n} & 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & \alpha & 1 & x_n & y_n \\ 1 & 1 & 1 & \dots & 1 & 0 & 0 & 0 \\ x_1 & x_2 & x_3 & \dots & x_n & 0 & 0 & 0 \\ y_1 & y_2 & y_3 & \dots & y_n & 0 & 0 & 0 \end{bmatrix} \quad (2.39)$$

and the vector \mathbf{c} containing all the desired parameters on the left side and on the right side the vector \mathbf{z} of the values at all the control points.

$$\mathbf{c} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \\ d_0 \\ d_1 \\ d_2 \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_n \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.40)$$

For the construction of \mathbf{N} , the entries a_{ij} are computed by

$$a_{ij} = r_{ij}^2 \ln(r_{ij}). \quad (2.41)$$

After solving this linear system of equations, the interpolation or regression function $f(x, y)$ is fully defined by the coefficient vector \mathbf{c} . The formulation for the interpolation spline in eq. (2.36) or the regression function in eq. (2.37) can then be evaluated at an arbitrary point (x_i, y_i) to obtain an value of z .

2.4 Hyperreduction techniques

In the field of reduced order modeling the term *Hyperreduction techniques* is often referred to techniques, that aim to accelerate the computation or evaluation of nonlinearities [7]. Especially in projection based ROMs that are based on a Galerkin projection, nonlinearities pose a fundamental problem for the model reduction. This is because the inner

product of potential nonlinear terms prevents a reduction of the model complexity, that is independent from the complexity of the full order model [68]. To circumvent this bottleneck several methods have been proposed, which reduce the cost of the evaluation of nonlinearities by reducing the query points for the nonlinear term. The approaches to find a suitable approximation for the full nonlinear term reach from (empirical) interpolation methods [45], [68], [48] over regression based methods [38], [61], [59], [23] to collocation / domain decomposition methods [16], [15], [53]. In the following sections two well established methods within the field of numerical flow simulations are presented. These methods have been widely used and were already investigated in the papers referring to previous activities concerning the LSQ-ROM method. In the context of compressible fluid flows, the FOM is described by the Navier-Stokes equations, which consist of several nonlinear terms for the description of fluxes and sources. These nonlinear terms constitute the need for hyperreduction techniques for efficient ROMs in the field of aerodynamics.

2.4.1 Discrete Empirical Interpolation Method

The DEIM is the discrete variant of the empirical interpolation method (EIM), described in [45]. The EIM was developed to reduce the complexity of non-affine functions within partial differential equations. The idea is to represent the non-affine term by a reduced-basis expansion. For that, several empirical interpolation functions are defined, via a proper choice of interpolation points, to represent the reduced basis. Because the EIM identifies the interpolation points through optima in a continuous spatial domain, the DEIM includes modifications to make this technique applicable to discrete representations of a spatial domain. Thereby, the DEIM has been successfully utilized as a hyperreduction technique.

Given a spatial domain, discretized by a set of n grid points, we assume an already reduced space $\mathbb{V} \subset \mathbb{R}^n$, spanned by $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m] \in \mathbb{R}^{n \times m}$. The reduced space for the solutions within the spatial domain can be produced e.g. through POD, that was described in section 2.2. The aim of the DEIM is an approximation of the function $\mathbf{f}(U)$, projected onto the reduced space, which fullfills certain error bounds [68]. Defining the approximation of \mathbf{f} by a linear combination in the reduced space by

$$\mathbf{f} \approx \tilde{\mathbf{f}} \approx \mathbf{U}\mathbf{c} \quad (2.42)$$

leads to an overdetermined system for the coefficient vector $\mathbf{c} \in \mathbb{R}^m$. The determination of \mathbf{c} follows through selection of m distinguished rows from the basis vectors of the reduced space. These rows or their indices can be understood as the interpolation points for the approximation of \mathbf{f} . The row selection can be described by a mask matrix $\mathbf{P} \in \mathbb{R}^{n \times m}$, that contains m unit vectors \mathbf{e}_{j_i} with $j_i \in \mathbf{J} = \{j_1, \dots, j_m\} \subset \{1, \dots, n\}$ as columns. A multiplication with \mathbf{P}^T from the left picks m rows out of the matrix it were multiplied to. Applied to the approximation of $\tilde{\mathbf{f}}$, the overdetermined system for the coefficients \mathbf{c} reduces to

$$\mathbf{P}^T \tilde{\mathbf{f}} \approx \mathbf{P}^T \mathbf{U}\mathbf{c}. \quad (2.43)$$

With this equation together with eq. (2.42), the approximated function $\tilde{\mathbf{f}}$ can expressed as

$$\tilde{\mathbf{f}} = \mathbf{U} (\mathbf{P}^T \mathbf{U})^{-1} \mathbf{P}^T \mathbf{f}. \quad (2.44)$$

This formulation makes clear that the effort for the evaluation of the nonlinear function $\mathcal{O}(m)$ becomes independent from the size of the original problem $\mathcal{O}(n)$. The main task in the preprocessing is the determination of the appropriate interpolation indices for the specific reduced basis \mathbf{U} . The algorithm to select these indices is described in table 2.1 and refers to the one in [68]. The entry into the algorithm is the first basis vector of the reduced space. Its maximum absolute entry defines the first selected index. From

Algorithm: Discrete Empirical Interpolation Method

```

1   $[r_{\max}, j_1] \leftarrow \max(|\vec{u}_1|)$ 
2   $\mathbf{U} \leftarrow \mathbf{u}_1, \mathbf{P} \leftarrow \mathbf{e}_{j_1}, \mathbf{J} \leftarrow \{j_1\}$ 
   for:  $l = 2, \dots, m$ 
3     solve:  $\mathbf{P}^T \mathbf{U} \mathbf{c} = \mathbf{P}^T \mathbf{u}_l$ 
4      $\mathbf{r} \leftarrow \mathbf{u}_l - \mathbf{U} \mathbf{c}$ 
5      $[r_{\max}, j_l] \leftarrow \max(|\mathbf{r}|)$ 
6      $\mathbf{U} \leftarrow [\mathbf{U}, \mathbf{u}_l], \mathbf{P} \leftarrow [\mathbf{P}, \mathbf{e}_{j_l}], \mathbf{J} \leftarrow \{\mathbf{J}, j_l\}$ 

```

Table 2.1: DEIM algorithm

that on, all remaining $m - 1$ indices are chosen by the steps 3-5 in the algorithm. Step 3 includes the computation of a small scale linear system for the coefficients $\mathbf{c} \in \mathbb{R}^{l-1}$, where the left side corresponds to the sampled nonlinear term of eq. (2.43). The solution coefficients lead to an approximation of the basis vector \mathbf{u}_l , which is exact at the current set of indices $\{j_1, \dots, j_{l-1}\}$, like interpolations are exact at their sampling points. In the next step of the algorithm, the residual or the error between the current processed basis vector \mathbf{u}_l and its approximation in the current basis $\{\mathbf{u}\}_{l=1}^{l-1}$ is computed. The error is then used in step 5 to determine the next index, which is the index of the residual with the highest absolute value. Therefore, the newly added index in step 6 is the one that would result in the highest interpolation error between the former basis and its added basis vector. To guarantee that the interpolation indices are nonrepeated and that $\mathbf{P}^T \mathbf{U}$ is always nonsingular, the vectors $\{\mathbf{u}\}_{l=1}^m$ have to be linear independent, which is assured by a POD basis. Since the DEIM approximation is uniquely determined by its underlying projection basis \mathbf{U} , the overall accuracy of the DEIM is influenced and determined by the chosen basis. In [68] an additional bound for the stepwise growth of the error during the process of indices determination is given, which results in a restriction of the overall error introduced by the discrete empirical interpolation method. One drawback of the DEIM is the restriction of the number of selectable indices to the number of basis vectors. However, this can be utilized for other hyperreduction methods, which rely on a start set of indices like the Missing Point Estimation, which is described next.

2.4.2 Missing Point Estimation

Missing point estimation is a nonlinear reduction method to select a subset of a spatial domain in order to minimize the error between the exact solution of a certain physical model and the solution reconstructed within a reduced subspace. This results in an optimization problem that seeks the subset for which the alias sensitivity is minimal [52]. Because of the high complexity of this combinatorial optimization problem, the authors in [52] proposed a heuristic approach for the selection of a subset, that is somehow close to the optimum set. The foundation for this method can be found in the *gappy POD* [59] and techniques originating from the field of signal processing and the problem of signal reconstruction. First applications of the MPE were demonstrated for a ROM for the incompressible Navier-Stokes equations based on a galerkin projection. However, the MPE proved successful also for compressible aerodynamic problems and in general as a hyperreduction method. The efficiency and accuracy of the original algorithm for the MPE was improved in [61] by a *Accelerated Greedy MPE*, whose characteristics shall be

presented after a description of the underlying combinatorial optimization problem. The MPE is assumed to operate in a projection-based ROM to further reduce the model complexity by lowering the number of degrees of freedom in case of a spatial problem. Although a POD basis is not essential for the application of the MPE, a subspace, which is constructed by POD modes, is assumed for the following derivation of the MPE method. The subset of the spatial domain is identified by a mask matrix $\mathbf{P} = (\mathbf{e}_{j_1}, \dots, \mathbf{e}_{j_s}) \in \mathbb{R}^{n \times s}$, that consists of s column-wise stacked unit vectors \mathbf{e}_{j_i} with j_i being an index of the subset of indices $\mathbf{J} = \{j_1, \dots, j_s\} \subset \{1, \dots, n\}$. Multiplying a vector \mathbf{y} with the transposed of \mathbf{P} , selects the rows j_1, \dots, j_s of \mathbf{y} . In a usual application like a spatial discretization, these indices refer to certain points, nodes or cells. Similar to section 2.2, the state vector of the modeled system is described by the vector \mathbf{y} and the reduced subspace is spanned by the d POD modes, which are represented by the matrix $\mathbf{U} \in \mathbb{R}^{n \times d}$. In order to find the optimum set of indices of the MPE, the following Least-squares problem needs to be solved [61]

$$\min_{\alpha \in \mathbb{R}^n} \left| \mathbf{P}^T \mathbf{y} - \mathbf{P}^T \mathbf{U} \mathbf{U}^T \mathbf{y} \right|^2, \quad (2.45)$$

which is the masked version of the projection problem in section 2.2. In contrast to the DEIM, presented in section 2.4.1, the MPE relies on a predefined set of start indices, whose cardinality is larger than the number of basis vectors, so that the Least-squares problem is not rank-deficient. Whereas the solution of the unmasked Least-squares problem is the orthogonal projection of \mathbf{y} onto the subspace, spanned by \mathbf{U} , the optimum projection for the masked problem is

$$\tilde{\mathbf{y}} = \mathbf{\Pi} \mathbf{y} \quad (2.46)$$

with the projection operator $\mathbf{\Pi}$

$$\|\mathbf{\Pi}\| = \mathbf{U}^T \left(\mathbf{U}^T \mathbf{P} \mathbf{P}^T \mathbf{U} \right)^{-1} \mathbf{U}^T \mathbf{P} \mathbf{P}^T. \quad (2.47)$$

This formulation can be easily obtained by applying the general solution of a linear Least-squares problem to the masked problem. From this, it is possible to derive a bound for the error between the exact solution \mathbf{y} and its masked projection $\hat{\mathbf{y}}$:

$$\|\mathbf{y} - \hat{\mathbf{y}}\| \leq \|\mathbf{\Pi}\| \|\mathbf{y} - \mathbf{U} \mathbf{U}^T \mathbf{y}\|. \quad (2.48)$$

The norm of the projection operator directly controls the maximum error of the masked projection. By using an SVD of the projection operator and specifying the norm as the spectral radius, the factor for the error bound takes the form

$$\|\mathbf{\Pi}\| = \frac{1}{\sigma_{\min}(\mathbf{P}^T \mathbf{U})} \quad (2.49)$$

Since \mathbf{P} is the only variable parameter (constant POD basis assumed), minimization of the error introduced by the masked projection means maximizing the minimum eigenvector of the matrix $\mathbf{P}^T \mathbf{U} \in \mathbb{R}^{s \times d}$. The application of an exhaustive greedy search for this problem leads to an inefficient algorithm, because it has to add indices successively by computing for all potential new mask matrices $\mathbf{P}_{s+1} = (\mathbf{P}, \mathbf{e}_{j_{s+1}})$ the minimum eigenvalue and comparing them to each other. In [61] another error bound is proposed with the goal of restricting the residuals of the projected solution $\tilde{\mathbf{y}}$ and the masked projected solution $\hat{\mathbf{y}}$. This bound is defined by

$$\|\mathbf{\Pi}\| = \frac{1}{\sigma_{\min}(\mathbf{P}^T \mathbf{L})} \quad (2.50)$$

with \mathbf{L} being the left matrix of the compact SVD of \mathbf{U} as it was introduced in section 2.2. When searching for the optimum set of indices by minimizing one of the two presented error bounds, one has to solve almost n SVDs for every step adding a new index to

the current subset (*exhaustive greedy algorithm*). To accelerate the index selection, the algorithm from [61] is extended by different screening criteria that reduce the number of potential new indices to a smaller set with the best values according to the specific screening criteria.

The developed screening criteria are based on specific characteristics of a rank-one SVD update problem, which needs to be solved to obtain the minimum eigenvalue of $\mathbf{P}_{s+1}^T \mathbf{U}$. When decomposing this matrix with the compact SVD $\mathbf{P}_{s+1}^T \mathbf{U} = \mathbf{L} \mathbf{\Sigma} \mathbf{R}^T$, the following formulation can be found:

$$\left(\mathbf{P}_{s+1}^T \mathbf{U}\right)^T \left(\mathbf{P}_{s+1}^T \mathbf{U}\right) = \mathbf{R}_s \left(\mathbf{\Sigma}_s^2 + \mathbf{v} \mathbf{v}^T\right) \mathbf{R}_s^T \quad (2.51)$$

where the subscript s indicates matrices of the SVD of the mask matrix \mathbf{P}_s before the rank-one update. The vector \mathbf{v}

$$\mathbf{v} = \mathbf{R}_s^T \mathbf{u}_{j_{s+1}}^T \in \mathbb{R}^d$$

is therefore the row with the index j_{s+1} of the matrix $\mathbf{U} \mathbf{R}_s$. The potential vectors \mathbf{v}_{j_i} for the rank-one update problem are called candidate vectors and are the key objective of the screening criteria. When examining eq. (2.51) it is clear, that the term

$$\mathbf{M} = \mathbf{D} + \mathbf{v} \mathbf{v}^T \quad \text{with: } \mathbf{D} = \mathbf{\Sigma}_s^2$$

constitutes the eigenvalues of $\mathbf{P}^T \mathbf{U}$ after the rank-one update. The candidate vector \mathbf{v} is the only modification to the original eigenvalue problem and therefore used for an effective screening of all potential rank-one updates. An important property of a rank-one update is that the eigenvalues λ_i of the updated problem and the eigenvalues d_i of the original problem are ordered in the following way [32]

$$\lambda_1 \geq d_1 \geq d_2 \geq \dots \geq d_{p-1} \geq \lambda_p \geq d_p \quad (2.52)$$

(assuming that λ_i and d_i $i = 1, \dots, d$ are in descending order). Therefore, the smallest eigenvalue of the updated problem λ_d is bound by the intervall $[d_{p-1}, d_p]$, which is important when someone aims to maximize the smallest eigenvalue in order to minimize the error bound for the masked projection.

The characteristic polynomial of \mathbf{M} leads to the expression [24]

$$f(\lambda) = 1 + \sum_{i=1}^p \frac{v_i^2}{d_i - \lambda}, \quad (2.53)$$

which can be differentiated under the assumption that \mathbf{D} is strictly positive definite [61]. The partial derivative of the eigenvalue λ_k by v_i is

$$\partial_i \lambda_k(\mathbf{v}) = \frac{-2}{\|\mathbf{q}_k\|^2} \frac{v_i}{d_i - \lambda_k(\mathbf{v})} \quad (2.54)$$

with \mathbf{q}_k being the eigenvector of \mathbf{M} , that corresponds to the k -th eigenvalue. From this, it is possible to already derive some important properties of a candidate vector for the minimum eigenvalue λ_p . So for all entries $v_i : i < p$ ($d_i \geq \lambda_p$) λ_p is decreasing for values $v_i > 0$ and increasing for values $v_i < 0$. The opposite is true for the last entry of the candidate vector $v_p : i = p$ ($d_p \leq \lambda_p$). For the largest growth in σ_{\min} , the last component of \mathbf{v} should be as large as possible while the remaining entries should be as small as possible.

The proposed screening criteria make use of an approximation of the exact eigenvalue λ_p , which is the p -th root of eq. (2.53). To obtain a good estimation $\lambda_{p,0}$, [61] approximates

eq. (2.53) with the help of two constants, either

$$c_{up} = 1 + \sum_{i=1}^{p-2} \frac{v_i^2}{d_i - d_{p-1}} \quad (2.55)$$

or

$$c_{lo} = 1 + \sum_{i=1}^{p-2} \frac{v_i^2}{d_i - d_p}, \quad (2.56)$$

depending on how close λ_p is either to d_{p-1} or d_p . Inserting one of the terms into eq. (2.53) leads to a quadratic equation in λ_p which can be solved straightforward. With this, an estimate for the distance between the updated last eigenvalue λ_p and the second last eigenvalue of the original problem d_{p-1} is

$$0 < d_{p-1} - \lambda_p < d_{p-1} - \lambda_{p,0} \\ = \frac{d_{p-1} - d_p}{2} - \frac{1}{2c_{up}} \left(v_{p-1}^2 + v_p^2 - \sqrt{(\beta_p - v_p^2)^2 + (\beta_p + v_{p-1}^2)^2 + 2v_{p-1}^2 v_p^2 - \beta_p^2} \right) \quad (2.57)$$

$$\text{with } \beta_p = c_{up}(d_{p-1} - d_p).$$

This equation states the evaluation criterium for every candidate vector. A potential greedy algorithm would always pick the candidate vector, which implies the largest growth in λ_p and leads, because of eq. (2.57), to the smallest distance to the second last eigenvalue of the original problem. Picking up a simplified form of eq. (2.57) from [61]

$$0 < d_{p-1} - \lambda_p < \frac{v_{p-1}^2(d_{p-1} - d_p)}{v_p^2 - c_{up}(d_{p-1} - d_p)} \quad (2.58)$$

indicates that this is achieved by increasing the last entry of \vec{v} and/or decreasing the second last entry as far as possible. However, when performing this strategy multiple times, a problem can occur, because the upper limit of λ_p , which is d_{p-1} , does not increase with the same amount. In contrast, d_{p-1} will change minimally for candidate vectors with entries $v_{p-1} \rightarrow 0$, which is due to [32]

$$(\mathbf{D} + \mathbf{v}\mathbf{v}^T)\mathbf{e}_i = \mathbf{D}\mathbf{e}_i + \mathbf{v}v_i \stackrel{v_i \rightarrow 0}{\approx} d_i. \quad (2.59)$$

To circumvent a very slow growth behaviour, a suggested improvement is a compromise between greedily driving λ_p towards its upper limit and targeting an increase of the upper limit itself for the current iteration. The criterium from eq. (2.57) can be applied to any eigenvalue λ_{p-k} with minor modifications:

$$d_{p-k-1} - \lambda_{p-k} < \frac{d_{p-k-1} - d_{p-k}}{2} - \frac{1}{2c_{k,up}} \left(v_{p-k-1}^2 + v_{p-k}^2 - \text{sgn}(c_{k,up}) \sqrt{(\beta_{p-k} - v_{p-k}^2)^2 + (\beta_{p-k} + v_{p-k-1}^2)^2 + 2v_{p-k-1}^2 v_{p-k}^2 - \beta_{p-k}^2} \right) \quad (2.60)$$

$$k = 1, \dots, p-1$$

Therefore, it is possible to target any growth limiting eigenvalue. So for example, after multiple iterations, which targeted λ_p , λ_{p-1} could limit further growth of the smallest eigenvalue. Therefore, one of the proposed algorithms in [61] switches between targeting the smallest and the second smallest eigenvalue. By targeting for every third iteration the second smallest value, the upper limit is continuously increased to enable further growth

Algorithm: Fast MPE with "module-three" target switching

```

1   $\bar{\mathbf{J}}_s = \{1, \dots, n\} \setminus \mathbf{J}_s = \{j_1, \dots, j_s\}$ 
2  while:  $|\mathbf{J}_s| \leq n_{\max}$ 
3      if:  $\text{mod}(s, 3) \leq 1$ 
4           $k_t = 0$ 
5      else:
6           $k_t = 1$ 
7           $c_{\text{opt}} = d_{p-k-1} - d_{p-k}$ 
8          for:  $i = s + 1, \dots, n$ 
9               $\vec{v} \leftarrow \vec{v}_{j_i}$ 
10              $c_s \leftarrow$  (screening criterium according to eq. (2.60) with  $k = k_t$ )
11             if:  $c_s < c_{\text{opt}}$ 
12                  $c_{\text{opt}} = c_s, \quad j_{\text{opt}} = j_i$ 
13              $\mathbf{J}_{s+1} = \mathbf{J}_s \cup \{j_{\text{opt}}\}, \quad \bar{\mathbf{J}}_{s+1} = \bar{\mathbf{J}}_s \setminus \{j_{\text{opt}}\}$ 
14   $s = s + 1$ 
    
```

Table 2.2: Fast MPE algorithm

of the smallest eigenvalue. The basic steps of this algorithm are outlined in table 2.2. The second algorithm proposed in [61] targets recursively the eigenvalues that are indicated to limit further growth of the next smaller eigenvalue. This recursive problem is incorporated into the following criterium

$$\frac{d_{p-k} - d_{p-k+1}}{d_{p-k}} < \tau \quad k = 1, \dots, p - 1 \quad (2.61)$$

to identify the upper bound, which is too constraining for the future growth of λ_{p-k} . The parameter $\tau \in (0, 1)$ can be set individually as there are no general guidelines for its value. When this criterium is fulfilled, the interval $(d_{p-k+1} - d_{p-k})$ is considered as too small for further growth and the upper bound of λ_{p-k} , which is the next larger eigenvalue λ_{p-k-1} , is targeted instead of λ_{p-k} itself. In that way the smallest eigenvalue is targeted as long as its upper is bound not too close to limit the future growth. Both algorithms lead to satisfactory results with respect to acceleration and error limits. Due to its simpler application, which does not need any experience for the parameter τ , the first algorithm was utilized for this work.

According to this, the algorithm relies solely on a starting set of indices \mathbf{J} of a cardinality $> d$ and the parameter n_{\max} that defines the total number of indices to identify by the MPE. The starting set can be found for example by a preliminary application of the DEIM algorithm. This proposed surrogate for the full rank-one update of the eigenvalue problem has only computational costs which are linear in n (length of basis vectors) and quadratic in d (number of basis vectors) for every iteration. This leads to a significant reduction in computational time, as the exhaustive greedy algorithm, which solves the full rank-

one update problem, has computational costs which scale cubic with d . Investigations in [61] show that this acceleration comes at minor costs concerning the accuracy, since the accelerated algorithm picked in most of the cases the same candidate vector / index as the exhaustive algorithm.

2.4.3 Artificial hyperreduction indices

During the investigation of the hyperreduction methods it became clear that the hyperreduction indices selected by the MPE + DEIM need to be extended because often they do not include a sufficient number of boundary cell indices. This can lead to problems, since most boundary conditions are defined in a local way so that the imposed boundary conditions only affects cells and their stencils, which are directly connected to the boundary. Because of that, a residual which is not evaluated for those cells is not linked to the boundary anymore, although the states may highly deviate from the imposed condition. This can lead to problems during the optimization process of a LSQ-ROM, which can converge towards local minima at training points, as the training points may be defined solely by input parameters imposed at a boundary not included in the set of hyperreduction indices. To circumvent such issues, the set of hyperreduction indices for the LSQ-ROM should include a sufficient number of cells / indices at boundaries referring to the input parameter. In case of the angle of attack and the Mach number, which are utilized in this work, the corresponding boundary is the farfield of the flow domain, where the outer state is imposed based on flow angles and velocity. The importance of such boundary cells for the success of the hyperreduction was already shown in [2], which also emphasized that farfield cells are suited for the reduction of the residual. Nevertheless, the specific boundary that needs to be considered for additional boundary indices depends on the type of input parameter. While the angle of attack corresponds to the farfield boundary, a shape parameter of an airfoil would correspond to the wall boundary at the airfoil. So, the boundary that provides the additional indices has to be chosen with respect to the input parameters or the design space respectively. The required number of additional boundary indices is probably depending several factors like the number of hyperreduction indices or the initial condition of the optimization problem so as a rule of thumb a small percentage of all hyperreduction indices should correspond to boundary indices, just like a full residual includes a small percentage of residuals at the boundary.

2.5 Optimization algorithms

The investigated LSQ-ROM method relies on the minimization of the nonlinear residual of the discretized governing equations. As this minimization poses a nonlinear Least-squares problem, which is a special case of a general optimization problem, the following section will describe important strategies for the solution of such problems. Although the investigated LSQ-ROM method harnessed solely the well-known LM method, the underlying methods, which it is based on, shall be presented briefly, too.

The general optimization problem can be described as the search for a local minimizer $\mathbf{x}^* \in \mathbb{R}^n$ of a real-valued function $F(\mathbf{x})$. The objective function $F(\mathbf{x})$ is assumed to be at least twice differentiable, since this is a common requirement for established gradient based solution methods. Moreover, the problem shall be unconstrained with no conditions imposed on x and the minimizer \mathbf{x}^* is in general only a local minimizer [9]:

$$\min_{\mathbf{x}} F(\mathbf{x}) = F(\mathbf{x}^*) \leq F(\mathbf{x}) \quad \text{for all } \mathbf{x} : \quad \|\mathbf{x} - \mathbf{x}^*\| < \delta, \quad \delta > 0 \quad (2.62)$$

While it is often possible to find such a local minimizer, the identification of the global minimum becomes infeasible for arbitrary functions $F(\mathbf{x})$ and an unrestricted parameter

space for \mathbf{x} . A minimum needs to meet two conditions for optimality. First, the necessary condition

$$F'(\mathbf{x}^*) = 0 \quad (2.63)$$

to identify an extremum and second, the sufficient condition that $F''(\mathbf{x}^*)$ is strictly positive definite. Only points satisfying both of these conditions are local minimizer.

The gradient of $F(\mathbf{x})$, which is $F'(\mathbf{x})$ or $\nabla F(\mathbf{x})$, is the derivative of $F(\mathbf{x})$ with respect to every component of \mathbf{x} , whereas the second order derivative is $F''(\mathbf{x})$ or $\Delta F(\mathbf{x})$.

All presented minimization methods are iterative *Descent Methods*, which enforce the descending condition $F(\mathbf{x}_{k+1}) < F(\mathbf{x}_k)$ for every iteration $k = 1, \dots, n$. A *Descent Method* is built up on three generical steps [37]:

1. find a descent direction \mathbf{h}_d ;
2. find a step length α in the found descent direction, giving a sufficient decrease for $F(\mathbf{x})$;
3. apply the step to the current minimizer: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{h}_d$;

Iteratively running through these steps drives the parameter vector x continuously towards a local minimizer. The number of iterations needed for this procedure is heavily influenced by the method-specific convergence order p , which is defined by

$$\|\mathbf{x}_{k-1} - \mathbf{x}^*\| \leq K \|\mathbf{x}_k - \mathbf{x}^*\|^p \quad 0 \leq K \leq 1. \quad (2.64)$$

Hence, a method with a convergence order of 2 decreases the error of the next iterated solution quadratically and is therefore called quadratically-convergent. Another noticeable point is the dependence of the finale solution x^* of the starting point x_0 . The iterative process is influenced by x_0 in such a complex way, that does not guarantee, that the found local minimizer x^* is the one closest to x_0 [37]. This is why even small changes of the starting point for the iterative process can lead to totally different local minima.

Steepest Descent Method

The Steepest Descent Method describes a globally convergent algorithm, that is capable of finding a local minimum [9]. Its iterative procedure takes the classic form

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathbf{h}_{sd} \quad (2.65)$$

wherein \mathbf{h}_{sd} is the descent direction, which is simply the negative gradient of the objective function:

$$\mathbf{h}_{sd} = -F'(\mathbf{x}) = -\nabla F(\mathbf{x}). \quad (2.66)$$

Because $F'(\mathbf{x})$ scales with the magnitude of $F(\mathbf{x})$, the step length α needs to be chosen appropriately to ensure a continuous decrease in $F(\mathbf{x})$. To find such an appropriate step length, a popular class of methods are *Trust Region Methods* that require – on top of the optimization problem – the solution of a Trust Region sub problem, generally defined by:

$$\mathbf{h}_{tr} = \min_{\|\mathbf{h}\| \leq \Delta} \{m(\mathbf{h})\}. \quad (2.67)$$

For that, a quadratic model $m(\mathbf{h})$ for the objective function $F(\mathbf{x})$ is given by

$$m(\mathbf{h}) = F(\mathbf{x}) + \mathbf{h}^T F'(\mathbf{x}) + \frac{1}{2} \mathbf{h}^T F''(\mathbf{x}) \mathbf{h}. \quad (2.68)$$

This model is assumed to to be a sufficiently accurate approximation $F(\mathbf{x})$ within the radius Δ of the Trust Region. A detailed description of this problem and the various

techniques to solve it are beyond the scope of this thesis, but can be found e.g in [9]. After having found a suitable step length, the Steepest Descent step (see eq. (2.65)) can be applied. The overall method has a good performance in the initial stage of the iterative process for many problems, but converges only linearly in the final stage [37].

Newton's Method

Newton's method is a locally convergent method, that requires its initial iterate \mathbf{x}_0 to be sufficiently close to \mathbf{x}^* , so that the sufficient optimality condition holds. The method is normally used to solve systems of nonlinear equations by finding the root of the function

$$m(\mathbf{x} + \mathbf{h}) = F(\mathbf{x}) + \mathbf{h}^T F'(\mathbf{x}). \quad (2.69)$$

By setting this formula equal to zero, the iterative procedure definition of Newton's method becomes:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - F'(\mathbf{x}_k)^{-1} F(\mathbf{x}_k). \quad (2.70)$$

In the context of the minimization problem, Newton's method can be roughly understood as the application of eq. (2.70) to $F'(\mathbf{x})$ instead of $F(\mathbf{x})$. Or to be more precise: The new iterate \mathbf{x} of Newton's method converges to the minimizer of the local quadratic model $m(\mathbf{x} + \mathbf{h})$ of $F(\mathbf{x})$ [9]. This leads to

$$m'(\mathbf{x} + \mathbf{h}) = 0 = F'(\mathbf{x}) + F''(\mathbf{x})\mathbf{h} \quad (2.71)$$

and from that to the equation for the descent direction \mathbf{h}_{nm}

$$F''(\mathbf{x})\mathbf{h}_{nm} = -F'(\mathbf{x}). \quad (2.72)$$

In contrast to the Steepest Descent method, the received direction needs no scaling as it is independent of the magnitude of $F'(\mathbf{x})$. Therefore, it is possible to set the step length α simply to 1, such that

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_{nm}. \quad (2.73)$$

However, one has to make sure that \mathbf{h}_{nm} is actually a direction of descent, which is only ensured if the Hessian $F''(\mathbf{x})$ is positive definite. This is the case when \mathbf{x} is sufficiently close to the local minimum. Within this neighbourhood of the minimum, Newton's method converges quadratically towards the local minimum [56].

2.5.1 Nonlinear Least-squares problem

As mentioned in the beginning of this section, the LSQ-ROM method requires the solution of a nonlinear Least-squares problem, which is an intermediate stage between nonlinear equations and optimization problems. This perspective becomes clear by defining the Least-squares problem, which is basically the search for an optimum solution of an overdetermined nonlinear system. It can be solved by the minimization of the objective function $F(x)$ for the overdetermined system

$$\mathbf{A}(\mathbf{x}) = \mathbf{b} \quad \mathbf{b} \in \mathbb{R}^m, \quad \mathbf{x} \in \mathbb{R}^n \quad (2.74)$$

and its error function

$$f(\mathbf{x}) = \mathbf{b} - \mathbf{A}(\mathbf{x}) \quad f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^m \quad (2.75)$$

with $m \geq n$. From that, the objective function can be defined as the squared sum of all elements of $f(\mathbf{x})$ [37]:

$$F(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^m (f_i(\mathbf{x}))^2 = \frac{1}{2} f(\mathbf{x})^T f(\mathbf{x}). \quad (2.76)$$

Given this specific formulation of the objective function, the former mentioned first order derivative is [37]

$$F'(\mathbf{x}) = \sum_{i=1}^m f_i(\mathbf{x}) \frac{\partial f_i}{\partial x_j}(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T f(\mathbf{x}) \quad (2.77)$$

with $\mathbf{J} \in \mathbb{R}^{n \times m}$ being the Jacobian matrix consisting of the derivatives of every element of $f(\mathbf{x})$ with respect to every element of \mathbf{x} . The second derivative of $F(\mathbf{x})$ has the form

$$F''(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) + \sum_{i=1}^m f_i(\mathbf{x}) f_i''(\mathbf{x}) \quad (2.78)$$

and is also referred as the Hessian \mathbf{H} . It has to be noted, that the full evaluation of the Hessian of $F(\mathbf{x})$ requires m evaluations of Hessians of $f(\mathbf{x})$. This is the reason why classic Newtons method would be too costly for the solution of the minimization problem arising from the overdetermined system in eq. (2.74).

Gauss-Newton Method

The large cost of a exact Hessian matrix needed by the classic Newton method require some sort of simplification for the Hessian in order to enable a Newton like method with quadratic convergence order. The Gauss-Newton method approaches this task by approximating the error function $f(\mathbf{x})$ by a linear model

$$f(\mathbf{x} + \mathbf{h}) = f(\mathbf{x}) + \mathbf{J}(\mathbf{x})\mathbf{h}. \quad (2.79)$$

Inserted into eq. (2.76), it follows

$$F(\mathbf{x} + \mathbf{h}) = \frac{1}{2} f(\mathbf{x} + \mathbf{h})^T f(\mathbf{x} + \mathbf{h}) = F(\mathbf{x}) + \mathbf{h}^T \mathbf{J}(\mathbf{x})^T f(\mathbf{x}) + \frac{1}{2} \mathbf{h}^T \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) \mathbf{h}. \quad (2.80)$$

The Gauss-Newton method minimizes this term instead of $F(x)$ within the Newton method [37]. Hence, the Gauss-Newton method operates with

$$F'(\mathbf{x} + \mathbf{h}) = \mathbf{J}(\mathbf{x})^T f(\mathbf{x}) + \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) \mathbf{h} \quad (2.81)$$

and the Hessian

$$F''(\mathbf{x} + \mathbf{h}) = \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}). \quad (2.82)$$

Comparing $F''(\mathbf{x} + \mathbf{h})$ and $F''(\mathbf{x})$, illustrates that the Gauss-Newton methods just neglects the costly term of $F''(\mathbf{x})$ and transfers eq. (2.72) into

$$\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) \mathbf{h}_{gn} = -\mathbf{J}(\mathbf{x})^T f(\mathbf{x}). \quad (2.83)$$

So, in the Gauss-Newton algorithm there is solely the Jacobian needed to solve for the descent direction \mathbf{h}_{gn} . Some remarks concerning the evaluation of the Jacobian are given in the next section in combination with a detailed description of the damped version of the Gauss-Newton method. The simplifications made through the linearization of $f(\mathbf{x})$ make the computation of the Hessian feasible but on the other hand one can not expect to generally obtain the same quadratic convergence as for the classic Newtons method.

Similar behaviour requires $F''(x+h)$ to be a good approximation of $F''(\mathbf{x})$, which should only be the case for \mathbf{x} being close to \mathbf{x}^* and under the assumption of weak nonlinearities ($f_i'' \approx 0$):

$$F'(\mathbf{x} + \mathbf{h}) \approx F'(\mathbf{x}), \quad F''(\mathbf{x} + \mathbf{h}) \approx F''(\mathbf{x})$$

2.5.2 Levenberg-Marquardt method

The LM method, originally proposed by [40] and [19], can be understood as a damped Gauss-Newton method, combining the method of the Steepest Descent with the Gauss-Newton method. The formula for the descent direction of the LM method is given by [56] ($\mathbf{J} = \mathbf{J}(\mathbf{x})$):

$$(\mu \mathbf{I} + \mathbf{J}^T \mathbf{J}) \mathbf{h}_{lm} = -\mathbf{J}^T f(\mathbf{x}). \quad (2.84)$$

While the right hand side is the same as for Steepest Descent and the Gauss-Newton method, the left hand side consists of the term $\mathbf{J}^T \mathbf{J}$ originating from the Gauss-Newton method and the term $\mu \mathbf{I}$ being the scaled left hand side of the Steepest Descent method. The pure Gauss-Newton method can be regained by setting the damping factor or LM parameter μ to 0.

$$\mathbf{h}_{lm} = -(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T f(\mathbf{x})$$

Instead, setting $\mu \rightarrow \infty$ eq. (2.84) yields a scaled step into the direction of the steepest descent:

$$\mathbf{h}_{lm} = -\frac{1}{\mu} \mathbf{J}^T f(\mathbf{x})$$

As the damping factor is added to the diagonal elements of $\mathbf{J}^T \mathbf{J}$, a proper control of μ can ensure that the matrix $(\mu \mathbf{I} + \mathbf{J}^T \mathbf{J})$ is positive definite and thus \mathbf{h}_{lm} a descent direction. With μ steering the characteristics of the LM method, it is possible to use the global convergence of a steepest-descent like method in the early stage of the iterative process and the quadratic convergence of a newton like method in the final stage of the iterative process. To make advantage of that, a strategy is needed in order to properly increase and decrease μ during the iterative process. In [37], the described method to control μ follows the idea of the aforementioned trust region method. It is important to underline that the damping factor not only influences the descent direction but also the step size. This gives the possibility to reduce the number of parameters within the trust region problem to only one. The key of the control strategy is the gain ratio ρ_g . It is ratio of the actual gain in $F(\mathbf{x})$ induced by the step \mathbf{h}_{lm} to the gain predicted by the quadratic model $m(\mathbf{h})$ for $F(\mathbf{x})$. Given the model function

$$m(\mathbf{h}) = F(\mathbf{x}) + \mathbf{h}_{lm}^T \mathbf{J}(\mathbf{x})^T f(\mathbf{x}) + \frac{1}{2} \mathbf{h}_{lm}^T (\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x})) \mathbf{h}_{lm} \quad (2.85)$$

the gain ratio takes the form [37]

$$\rho_g = \frac{F(\mathbf{x}) - F(\mathbf{x} + \mathbf{h})}{m(0) - m(\mathbf{h})} = \frac{F(\mathbf{x}) - F(\mathbf{x} + \mathbf{h})}{\frac{1}{2} \mathbf{h}_{lm}^T (\mu \mathbf{h}_{lm} - \mathbf{J}^T f(\mathbf{x}))}. \quad (2.86)$$

For the case that ρ_g takes value close to 1, e.g. $\rho_g > \rho_{g,\text{high}} = 0.75$, the approximation of $F(\mathbf{x} + \mathbf{h})$ by the model is in good agreement with the actual value and the damping factor μ can be decreased to drive the algorithm towards the Gauss-Newton method. On the other side, if ρ_g is close to 0, e.g. $\rho_g < \rho_{g,\text{low}} = 0.25$ the model provides a poor approximation and the damping factor should be increased. An increased damping factor

leads to a descent direction closer to the direction of steepest descent and simultaneously reduces the step length. The given limits, which define a decrease, increase or no change in the damping factor are just recommendations from [37] and can be chosen problem dependent.

Computation of the Jacobian So far, it was assumed for all the presented methods, that the Jacobian already exists. However, the computation of an accurate Jacobian is not trivial. In most cases the analytical formulation is not known or extremely computationally expensive. When having the full source code at hand, automatic differentiation [4] may be a practical approach to obtain the exact Jacobian from the differentiation of the underlying code for the evaluation of $f(\mathbf{x})$. Unfortunately, in case of a black-box function $f(\mathbf{x})$ this approach falls short. The common way to obtain the Jacobian from a black-box function is by computing it by finite difference. Using this method, the function $f(\mathbf{x})$ is derived with respect to each parameter x_i through a small perturbation of the corresponding parameter. A derivative via a forward difference is obtained from

$$\frac{\partial f(\mathbf{x})}{\partial x_j} = \frac{f(\mathbf{x} + h\mathbf{e}_j) - f(\mathbf{x})}{h}, \quad (2.87)$$

where h is the magnitude of the perturbation and \mathbf{e}_j the j -th unit vector. Hence, for $\mathbf{J} \in \mathbb{R}^{n \times m}$, there are m function evaluations needed for the full Jacobian. The error of a forward difference is $\mathcal{O}(h + \frac{\epsilon_f}{h})$ with ϵ_f being the error caused by the limited number representation ($\approx 10^{-16}$ for 8-Byte floating numbers). Because of this, the perturbation magnitude needs to be chosen with some care, since the error can not be driven to zero by just further decreasing the perturbation value. Although there is no general rule for choosing h , in several application cases a constant value of 10^{-6} , which lays between the analytical optimum for a forward and a central difference [25], led to satisfactory results [70]. A generally more accurate approximation can be expected from a central difference like

$$\frac{\partial f(\mathbf{x})}{\partial x_j} = \frac{f(\mathbf{x} + h\mathbf{e}_j) - f(\mathbf{x} - h\mathbf{e}_j)}{2h}. \quad (2.88)$$

with an error of $\mathcal{O}(h^2 + \frac{\epsilon_f}{h})$. However, this requires twice the number of function evaluations than the forward difference.

Since the evaluation of $f(\mathbf{x})$ is often computational expensive, a recomputation of the Jacobian in every step of the LM algorithm can become performance limiting for the overall process. A way to circumvent this drawback is by computing the Jacobian in the first iteration and to successively update it during the following iterations. Broyden's Rank One Update [8] provides a way to obtain an approximation of the Jacobian \mathbf{J}_{k+1} based on the Jacobian of the prior step. The update mechanism is based on solving

$$\mathbf{J}_{k+1}\mathbf{v} = \mathbf{J}_k\mathbf{v}$$

for every vector \mathbf{v} that is orthogonal to the step $\mathbf{h}_{lm} = \mathbf{x}_{k+1} - \mathbf{x}_k$. This leads to the formulation:

$$\mathbf{J}_{k+1} = \mathbf{J}_k + \left(\frac{1}{\mathbf{h}^T\mathbf{h}} (f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) - \mathbf{J}_k\mathbf{h}) \right) \mathbf{h}^T. \quad (2.89)$$

In case of expensive function evaluations, the Broyden Update drastically reduces the computational costs for the evaluation of the Jacobian. However, it is not ensured that the updated Jacobian is a sufficient approximation of the real Jacobian. To counteract a deterioration of the Jacobian during the iterative process, it is possible to periodically recompute the full Jacobian via finite difference or just selected derivatives based on a measure of their approximation quality [37].

Stopping criteria In order to stop the infinite iterative process when the solution has reached a certain level of accuracy, an adequate stopping criteria has to be defined. To restrict the maximum number of iterations, the iterative process is halted when:

$$n_{\text{iter}} \geq n_{\text{max}}. \quad (2.90)$$

The process should also be stopped when the change in the objective function $F(x)$ is small enough to assume the proximity to a local minimum (since $F'(x^*) = 0$).

$$\|\mathbf{J}(\mathbf{x})^T f(\mathbf{x})\| \leq \epsilon_1 > 0 \quad (2.91)$$

One can also define a fallback criterium for the case that ϵ_1 is chosen too small causing rounding errors to prevent a proper stop. Hence, it should be checked that the step size is larger than a certain threshold:

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \leq \epsilon_2 (\|\mathbf{x}_k\| + \epsilon) \quad (2.92)$$

2.6 Algorithm of the LSQ-ROM

The purpose of this section is to give a basic overview of the algorithm that forms the LSQ-ROM method as it was proposed in [63], [62] for steady simulations and in [50] and [3] for unsteady simulations. Its description is divided into the offline stage and the online / prediction stage. During the offline stage, the CFD solver is queried to generate the training data (I), the POD basis is computed (II) and the prerequisites for the hyperreduction like the index selection are prepared (III). All these parts are presented roughly in the following paragraph. Since the new hyperreduction implementation within this method is subject of the following chapter, a detailed description of it can be found there. The prediction stage is based on the minimization of the residual of the CFD solver. This minimization process leads to a coefficient vector for the POD basis, that defines a solution for which the residual has a local minimum. Including the residual in the ROM preserves the connection of the reduced model to the physics of the FOM and thus gives the ROM an intrusive character. Since a residual with a value of 0 corresponds to the exact solution of the FOM, the value of the residual obtained from the minimization process is automatically an error measure for the predicted solution of the LSQ-ROM method.

2.6.1 Offline stage

The offline stage can be split into three main steps, which need the results of the previous step as its input. Since every step relies on its own algorithms, their brief description follows the sequential process chain of the offline stage. An illustration of the overall process is given in fig. 2.2 consisting of the three steps: Generation of snapshots (I), computation of the POD basis (II) and the preparation step for the hyperreduction (III).

I: Generation of snapshots The main part in the process chain is the generation of the training snapshots, which are solutions of the flow field computed by the CFD solver. Usually, the CFD algorithm has multiple free parameters including specific schemes for the flux discretization that need to be defined. These parameters can be divided into settings for the spatial discretization, which influence the final converged solution and for the time discretization, which solely influence the iterative process. And as part of the spatial discretization, boundary conditions have to be defined for the boundaries of

the solution domain. The independent variables ψ , which define the design point for the training snapshots, are also passed to the solver. Typical variables can be physical quantities like the Mach number or theoretical even geometrical quantities like airfoil shape parameters. The investigated parameter for this thesis are the Mach number and the angle of attack, giving:

$$\psi_i = \begin{pmatrix} M_i \\ \alpha_i \end{pmatrix}$$

for the i -th sample point. All the sample points are generated from a Design of Experiments, given for example by a multidimensional Halton series [30]. So, every training snapshot refers to a CFD solution computed for a specific set of (physical) parameters ψ_i . The combination of all the solutions \mathbf{y} or \mathbf{W}^1 into the snapshot matrix \mathbf{Y} presents the input for the next step for the offline stage of the LSQ-ROM.

Although the origin of the training snapshots is not of importance for the generation of the ROM, it is essential that the residual definition for the prediction stage refers to the definition used for the generation of the training data. Therefore, one has to pass the same settings for the spatial discretization to the CFD solver for the residual evaluation and the generation of the training data, as these settings influence the overall solution (see section 2.1.1).

II: Proper Orthogonal Decomposition Next, the snapshot matrix $\mathbf{Y} \in \mathbb{R}^{n \times d}$, is passed as an input to the POD. The decomposition is computed by an EVD or SVD with a prior centering of all snapshots around their mean value $\bar{\mathbf{y}}$. This yields a basis of $m - 1$ orthogonal POD modes, which are combined in the matrix \mathbf{U} . Further outputs are the coefficient matrix \mathbf{A} , the singular values σ_i and the eigenvalues λ_i , respectively. The latter are necessary for the first reduction tier of the model. After computing the full POD basis, a reduction can be performed by neglecting a certain number of the least important modes. This number needs to be defined either directly by r_m , which is the number of modes to retain for the reduced basis, or indirectly by the *RIC*. When passing a value for the *RIC*, the model will comprise as many modes as needed to reach at minimum the defined *RIC*. Besides the reduced basis \mathbf{U} , which is passed to the last step of the offline stage, the other outputs of the POD are stored for later use during the prediction stage and for the reconstruction of solutions.

III: Hyperreduction preparations The hyperreduction method poses the second reduction method within the LSQ-ROM method and relies on the reduced POD basis \mathbf{U} resulting from the previous step. Although the reduction is executed only during the prediction stage, its definition and preparations can be conducted during the offline stage. For that, in a first step, a proper set of indices (J_s) has to be identified. In all investigations the Fast MPE algorithm from section 2.4.2 is used after exploiting the DEIM to obtain the first d indices. Another input is the maximum number of indices n_{\max} , which defines the cardinality of J_s . Because of the restrictions for a index specific evaluation of the residual by the CFD solver, the indices can not be used directly but need to be further processed. They are input to the extraction of a reduced mesh or submesh that allows the computation of the residual for a smaller set of cells. Since every residual depends on its stencil (cf. section 2.1), this submesh contains all cells identified by J_s plus the cells belonging to the neighbourhood of these cells. Origin of this extraction is of course the mesh, the CFD solver is operating on for the training snapshots from step (I). The role and the need for this submesh is explained in more detail in the following chapter together with a description of the new implementations needed for this kind of hyperreduction.

¹The solutions / snapshots were previously defined by \mathbf{Y} or \mathbf{y} . However, to correspond to the usual CFD notation, \mathbf{W} and \mathbf{y} are used analogously

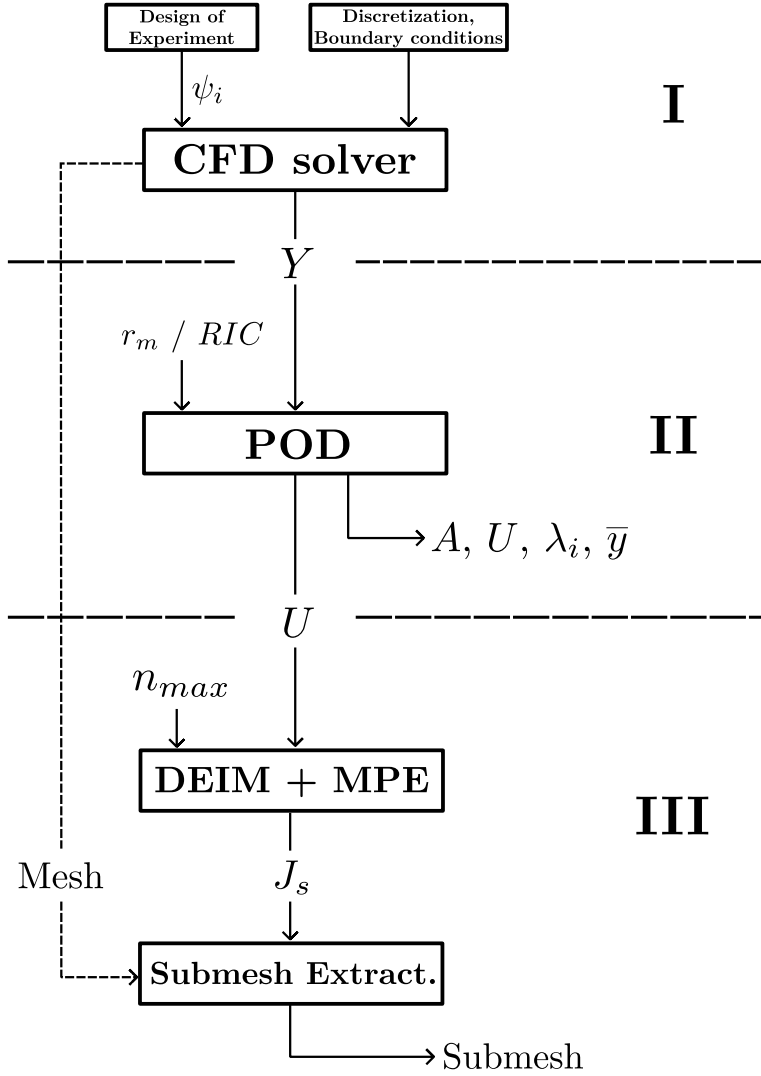


Figure 2.2: Schematic representation of the offline phase of the LSQ-ROM.

2.6.2 Prediction stage

Based on the reduced model from the previously explained offline stage, the prediction of a solution proceeds as it is outlined in fig. 2.3. From a specified prediction point ψ_p , a initial solution for the POD coefficient vector \mathbf{a} is generated by an interpolation method. Although the applied interpolation method was fixed, it is essential to keep in mind that the initial solution can influence the final solution (cf. section 2.5). The initial coefficients \mathbf{a}_0 are then used to compute an approximative solution \mathbf{W} . within the subspace \mathbb{V} , spanned by the POD modes \mathbf{U} (cf. section 2.2).

$$\mathbf{W} = \overline{\mathbf{W}} + \mathbf{U}\mathbf{a} \quad (2.93)$$

Because the snapshots were centered around their mean, the linear combination ($\mathbf{U}\mathbf{a}$) of the POD modes is added to that mean to obtain the approximated solution. This solution can be passed to the CFD solver, which computes the residual for that state. CFD solver and POD are queried by the Levenberg-Marquardt algorithm while it optimizes the coefficient vector \mathbf{a} based on the objective function $F(\mathbf{a})$.

$$\min_{\mathbf{a} \in \mathbb{R}^d} F(\mathbf{a}) = \min_{\mathbf{a} \in \mathbb{R}^d} \|\mathbf{R}(\overline{\mathbf{W}} + \mathbf{U}\hat{\mathbf{a}})\|_2 \quad (2.94)$$

The LM algorithm minimizes the squared sum of every residual entry, which can be understood as the search for a local minimum for the sum over all violations of the conservation laws. A local minimizer \mathbf{a}^* is assumed, when one of the defined stopping criteria is met (cf. section 2.5.2) and the objective function is at a local minimum. It is essential to notice that the chosen algorithm includes the residual of every conservative variable and treats them equally. As the residuals scale with the surface area of the individual elements, it becomes clear that some sort of normalization, e.g by the volumes, is required to prevent overrating of larger cells. Besides the residual evaluation for the objective function, the residual is also required for the computation of the Jacobian, which is evaluated via finite difference. The first step in the LM algorithm computes the Jacobian via full finite difference, while all the following iterations exploit the Broyden update, which is promoted by finite difference in the direction of a poorly approximated derivative. The whole algorithm follows the description of the Levenberg-Marquardt method in section 2.5.2.

Hyperreduction application To evaluate the potential of the new hyperreduction method, a comparison with its old implementation from [50] and [3] shall be presented. The main idea of a hyperreduction is to reduce the time needed to evaluate the nonlinear residual. As the required evaluation on only a specific set of cells is usually not part of a CFD solver, the former approach was to reduce the residual after its evaluation. Although this approach does not reduce the evaluation time in the CFD solver, it still lowers the costs of the optimization problem. When applying such subsequent residual reduction, the optimizer operates on a masked residual, leading to the formal minimization problem:

$$\min_{\mathbf{a} \in \mathbb{R}^d} F(\mathbf{a}) = \min_{\mathbf{a} \in \mathbb{R}^d} \|\mathbf{P}^T \mathbf{R}(\overline{\mathbf{W}} + \mathbf{U}\mathbf{a})\|_2 \quad (2.95)$$

where \mathbf{P} is the mask matrix. The mask matrix reduces the full residual to a reduced residual \mathbf{R}_{red} by solely picking the residual entries defined by the hyperreduction indices J_s .

$$\mathbf{R}_{\text{red}}(\overline{\mathbf{W}} + \mathbf{U}\mathbf{a}) = \mathbf{P}^T \mathbf{R}(\overline{\mathbf{W}} + \mathbf{U}\mathbf{a}) \quad (2.96)$$

Consistent hyperreduction The aim of this thesis is the implementation and investigation of a hyperreduction that is reducing the actual evaluation of the nonlinear residual.

To achieve this, a reduced mesh or submesh is constructed that serves as an additional mesh that is only used for the evaluation of the selected residuals J_s . The reduced mesh is defined by another set of cells J_p , which differs from J_s because of the way the residual is evaluated in the CFD solver (see section 3.2.2). Therefore, the reconstruction of the state is done for the reduced mesh, leading to the reduced state vector $\hat{\mathbf{W}} \in \mathbb{R}^{n_{\text{red}}}$, $n_{\text{red}} = n_c \cdot n_p$, ($n_c =$ number of conservative variables, $n_p = |J_p|$) which can be formally obtained from a masked POD basis with

$$\hat{\mathbf{W}} = \hat{\mathbf{P}}^T \overline{\mathbf{W}} \quad (2.97)$$

and

$$\hat{\mathbf{U}} = \hat{\mathbf{P}}^T \mathbf{U}. \quad (2.98)$$

The mask matrix $\hat{\mathbf{P}} \in \mathbb{R}^{n \times n_p}$ selects every row that corresponds to J_p and hence to a cell in the reduced mesh. By applying the mask matrix directly to the POD basis instead of performing the hyperreduction after the reconstruction with the full POD basis, the effort for the reconstruction of the states is reduced and proportional to the hyperreduction level. The hyperreduction level shall be defined by the ratio of cells not included in the submesh to the number of cells in the base mesh.

$$\text{hyperreduction level} = \frac{\text{number of cells not in the reduced mesh}}{\text{number of cells in the base mesh}} = \frac{n - n_p}{n} \quad (2.99)$$

Its values range from 0 for an reduced mesh that is equal to the base mesh to 1.0 for an empty reduced mesh. It is expected that the reduced number of cells in the submesh translates to a proportional reduction in residual evaluation computational cost, although the residual evaluation effort is more precisely determined by the number of faces. As the residual evaluation is performed on the reduced mesh, the minimization from eq. (2.94) becomes:

$$\min_{\mathbf{a} \in \mathbb{R}^d} F(\mathbf{a}) = \min_{\mathbf{a} \in \mathbb{R}^d} \|\hat{\mathbf{R}}(\hat{\mathbf{W}} + \hat{\mathbf{U}}\mathbf{a})\|_2 \quad (2.100)$$

So, the optimization includes the residual $\hat{\mathbf{R}}$ as a reduced black-box function instead of the full residual \mathbf{R} . From there, it is easy to see that the hyperreduction is directly acting on the computationally expensive evaluation of the nonlinear residual. As a result, the overall acceleration potential is based on the reduced effort for the reconstruction of the states $\hat{\mathbf{W}}$, the evaluation of the objective function / residual and the computations in the LM algorithm. Since the effort for the reconstruction of the states and the residual evaluation is roughly $\mathcal{O}(n)$, the time savings by application of the consistent hyperreduction should be proportional to the hyperreduction level.

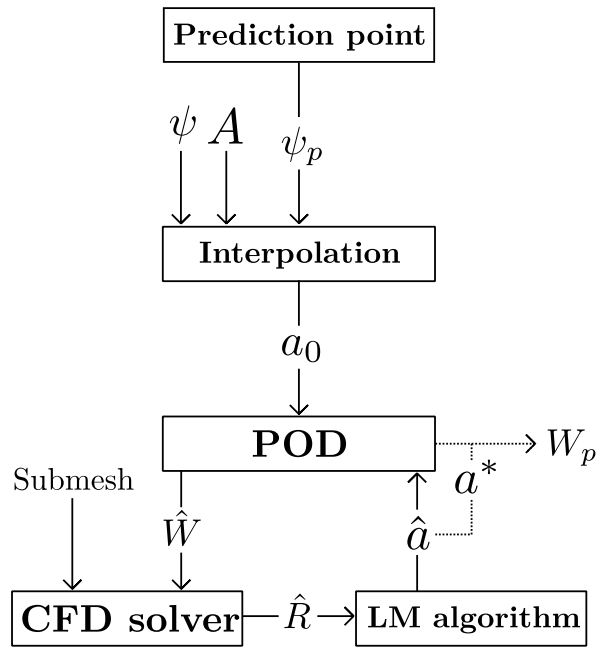


Figure 2.3: Schematic representation of the prediction phase of the LSQ-ROM.

Chapter 3

Implementation and validation approaches

The subsequent chapter shall give some insights into the framework the LSQ-ROM is operating in, as the method is based on independent software products, which are coupled through a HPC framework. After presenting the basic structure of this environment, the next sections explain how the LSQ-ROM is implemented in the DLR-Surrogate Modeling for Aero Data Toolbox in Python (SMARTy) and coupled with the CODA flowsolver, which is currently under development at Airbus, DLR and ONERA. From that, a detailed description of the implemented hyperreduction is given with its corresponding changes in SMARTy and CODA. To validate and investigate the new hyperreduction within the LSQ-ROM method coupled with CODA, multiple aerodynamic test cases with different geometries, in various flow regimes are explored. The last section presents these test cases and summarizes the most important settings for the generation of the training data and the reduced order model; results are presented in the following chapter (chapter 4).

3.1 Software tools and simulation framework

3.1.1 SMARTy

The DLR-Surrogate Modeling for Aero Data Toolbox in Python is a software package developed at DLR to make data-driven techniques like regression models or dimensionality reduction techniques available for aerodynamic tasks. However, as purely data-driven methods can be applied rather independently from their context, SMARTy can also be utilized in other areas like thermodynamics or propulsion modeling [57]. SMARTy is programmed in Python and follows the approach of an Application Programming Interface (API). Its modular structure allows to compose large and complex models from its various building blocks such as design of experiments, surrogate modeling, dimensionality reduction, flowsolver access and optimization algorithms. To increase the performance for expensive computations, these tasks are implemented in Cython, which can be additionally parallelized through OpenMP. A special feature of SMARTy is its close linkage to the CFD solver TAU and CODA, which are presented in the upcoming sections together with the framework that couples all the LSQ-ROM tools. The available interface between SMARTy and CFD solvers simplifies the implementation of physics-based methods like intrusive reduced order modeling (e.g. the LSQ-ROM used in this thesis) and the application of data-driven turbulence modeling to the physical equations of CFD.

3.1.2 CFD solver: TAU and CODA

TAU is a CFD solver, whose development already started in the 90s at DLR in the form of a modern software system for the computation of viscous and inviscid flows for steady and unsteady cases [18]. TAU can cover subsonic to hypersonic flows around complex geometries on hybrid unstructured grids. Although its main use case is the prediction of flows around complex aircraft-type configurations, many extensions emerged from research and industry contributions [18]. TAU implements a Finite-Volume scheme with a node-based approach for the storage of the flow variables. Various different upwind and central schemes are available for the discretization of the fluxes from Euler, Navier-Stokes or RANS equations. For the solution of the RANS equations, several turbulence models are implemented like the one equation model of Spalart and Allmaras as well as Reynolds stress models. The solution techniques for steady state problems are explicit Runge-Kutta schemes or low-cost implicit schemes, which both can be accelerated by available techniques like local time stepping or multigrid [29]. TAU features many further capabilities like mesh deformation and adaption or transition modeling, which are also described in [18] and documented in [29]. With over twenty years of active usage both in research and industry, TAU is highly validated within a wide range of application fields.

However, during the years more sophisticated solution algorithms and discretization approaches emerged, which can not be implemented straightforward into TAU's architecture. Therefore, a new CFD solver named "*CFD for ONERA, DLR and AIRBUS*" (CODA) is currently under development [75]. CODA is the computational fluid dynamics (CFD) software being developed as part of a collaboration between the French Aerospace Lab ONERA, the German Aerospace Center (DLR), Airbus, and their European research partners. CODA is jointly owned by ONERA, DLR and Airbus. In contrast to TAU, CODA is designed to natively operate in the FlowSimulator environment (section 3.1.3), whereas TAU is originally a standalone software. Some of the main development goals of CODA are new higher order discretization approaches like Discontinuous Galerkin Methods together with implicit solution algorithms and advanced multigrid methods [28]. Besides such methods which are still part of active research, CODA already features the established Finite-Volume discretization with a cell-centered scheme for the solution of the Navier-Stokes equations and its most common derivatives. Upwind schemes as well as central schemes are available for the discretization of the fluxes. Further alterations focus on structural and algorithmic adaptations to improve the performance on high performance computing (HPC) clusters with their parallel architecture. Although CODA is still under active development, its core functionalities are already available, validated and used for this work.

3.1.3 FlowSimulator environment

Since the LSQ-ROM method is mainly implemented in the SMARTy package, the overall implementation relies on an additional CFD solver like TAU or CODA. Such a coupling of different software tools, which normally introduce their own software architecture and interface, requires an efficient and aligned framework. The LSQ-ROM is, as a method of SMARTy, part of the FlowSimulator environment. The FlowSimulator provides an open framework for the integration of various software tools with the main focus on multi-disciplinary simulations, e.g. for MDAO of an aircraft [46]. In such application cases, fully evolved software tools from various areas of the aircraft design process are required, therefore creating the need for a computational infrastructure able to effectively connect these tools to each other. The FlowSimulator, developed by Airbus and several other partners like DLR, tackles these tasks [42]. Figure 3.1 gives a brief overview of the architecture of the FlowSimulator, which is based on the FlowSimulator DataManager (FSDM) as a common data handler for the tools linked through this framework. Tools

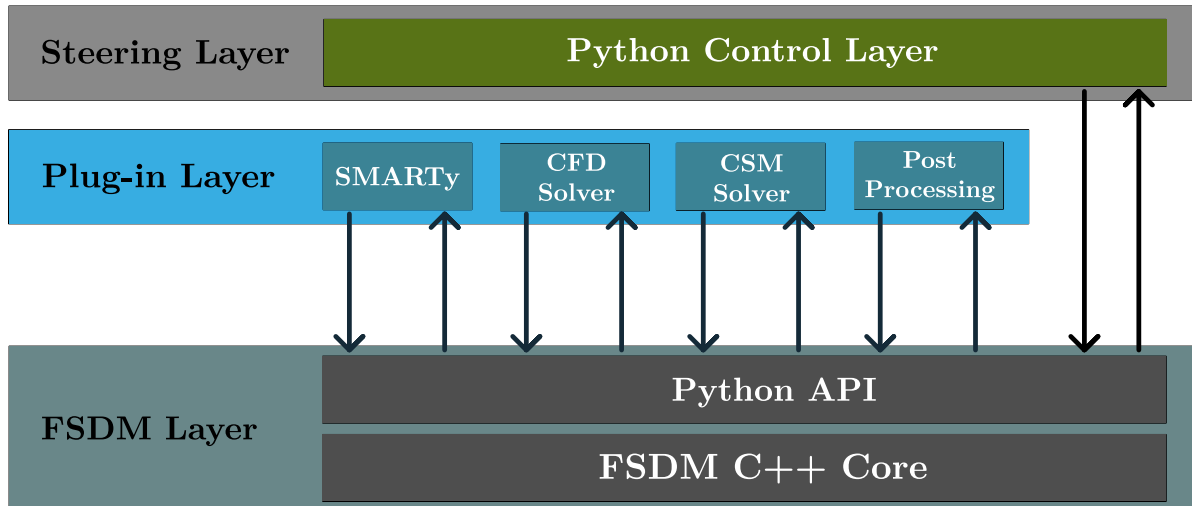


Figure 3.1: The FlowSimluator environment.

like SMARTy are part of the plug-in layer, that forms a common environment for the integration of applications. Besides SMARTy and the CFD solvers TAU and CODA, various other tools are available or can be potentially integrated in the plug-in layer; e.g. a Computational Structural Mechanics (CSM) solver and additional post processing tools could be introduced in case of an aeroelastic problem. Of particular value for this work are plug-ins to manipulate meshes and to import and export the data stored for these meshes to usual file formats. All the different plug-ins can be steered via the control layer by using e.g. python control scripts, whereas their interactions are enabled either through the python API of the FSDM or directly through the C++ core. In order to support parallel applications, the FSDM is fully compatible with highly parallel cluster architectures, which are normally distributed memory systems [46].

SMARTy offers in-memory views of data-structures in the FSDM to reduce the number of unnecessary copy operations, thereby improving the efficiency while relying on other software tools connected to the FSDM [57]. Therefore, the data structures can be easily accessed in other tools like a CFD-solver. Within the scope of the LSQ-ROM method, the control is mostly done from the SMARTy side as it provides problem adapted control methods for the CFD solver. Some of these methods are part of the implementations carried out for the investigations of this thesis, and they are described in the next section.

3.2 Implementations within SMARTy and CODA

The main part of this section describes the idea and the basic structure of the implemented hyperreduction for the LSQ-ROM method in SMARTy. So far, the LSQ-ROM was only coupled to TAU. Therefore, additional implementations are required to enable the usage of CODA for the LSQ-ROM. Hence, some remarks concerning these changes are given in advance. Finally, the adaptations to the preprocessor of CODA and to the LSQ-ROM in SMARTy are summarized. These adaptations become necessary because of the chosen hyperreduction approach.

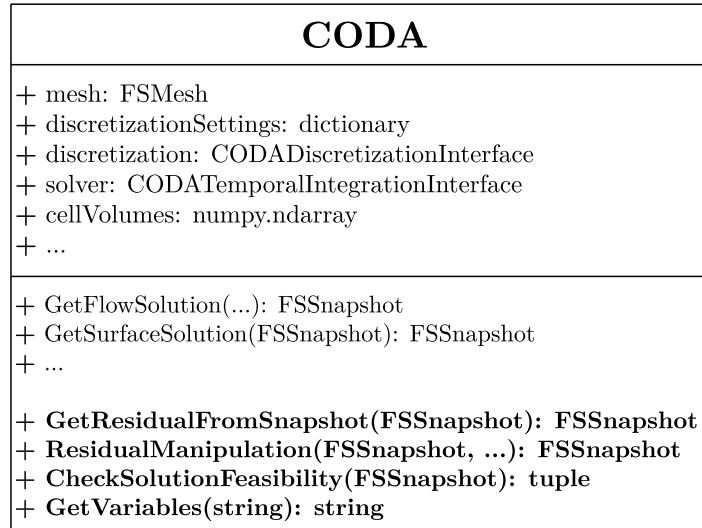


Figure 3.2: CODA class diagram.

3.2.1 LSQ-ROM method with CODA as the flow solver

The implementations for the coupling of the LSQ-ROM with CODA can draw on an already existing rudimentary interface class to CODA. In order to guarantee the interchangeability of the two CFD solvers (TAU and CODA) within SMARTy functionalities, the two corresponding SMARTy modules have to share the same fundamental structure for public Python methods. In particular, the following functionalities are added to the existing CODA module in SMARTy in order to successfully reach the final objective of this thesis research:

- computation of the residual in CODA based on a snapshot, which holds an arbitrary field solution and corresponds to the mesh / spatial discretization in CODA
- manipulation of the residual vector obtained from CODA (e.g normalization, selection of specific entries, ...)
- evaluation of a CODA flow solution with respect to physical meaningful values (e.g non-negative density)
- additional helper methods

A simplified class diagram of the CODA interface class in SMARTy can be seen in fig. 3.2 where the new functionalities introduced during the thesis work are in bold characters. Together with some code restructuring, these changes enable the application of the LSQ-ROM method coupled with the flow solver CODA. This code state is used for the validation and comparison of the LSQ-ROM based on CODA with the already existing LSQ-ROM based on TAU.

It is important to mention that the application is restricted to steady state simulations as the unsteady LSQ-ROM would require further adaptations, which are beyond the scope of this work. Moreover, it is not clear if or to which extent it is applicable to CODA utilizing a multistage Runge-Kutta scheme for the computation of unsteady simulations instead of the Backwards-Difference scheme applied in TAU.

3.2.2 Hyperreduction for the LSQ-ROM with CODA

As it is stated in the introduction (cf. section 1.2), the hyperreduction technique in the LSQ-ROM was already investigated for unsteady aerodynamic problems in [3] and [50]. These investigations showed a significant acceleration of the ROM by application of some sort of hyperreduction, without causing a perceivable deterioration in the prediction accuracy when the reduction levels are appropriately chosen. However, the solver TAU that has been used in the previous LSQ-ROM applications does not have the capability to compute the residual on a subset of entries or cells, and therefore the reduction of the residual was performed after its full evaluation. Therefore the main effort, which is the actual evaluation of the nonlinear residual, is not reduced. As a consequence, all the time savings are solely due to a reduction of the optimization problem, which then operates on a smaller subset of the residual. Giving up the main acceleration potential in form of a residual evaluation on a smaller subset, a true hyperreduction is sought, that is independent of the total problem size $\mathcal{O}(n)$. With CODA it becomes possible to implement a residual evaluation that is just performed on a subset of cells. The subset is initially identified by means of one of the hyperreduction methods previously described in section 2.4 and further processed to obtain a reduced mesh allowing the evaluation of a reduced residual. Given the aforementioned promising results with TAU and the estimations given in section 2.6.2, a further acceleration of the LSQ-ROM is expected by such a hyperreduction implementation.

The basic idea behind the new implemented hyperreduction is to compute the reduced residual by reducing the mesh and computing the full residual on this reduced mesh. This new mesh is called a submesh from here on as it is based on a subset of cells (indices) of the original mesh. While the original mesh is used to generate the training data and to obtain the entire solution after the optimization process for the POD coefficients, the submesh is generated purely for the residual evaluation. One of the reasons for this approach is the reduction of the interference with CODA to a minimum. This is because the submesh can be constructed outside of CODA in the FlowSimulator environment from where it can be passed to CODA as an input for the subsequent transfer into a computational mesh. This is achieved in a preprocessing step, which is fairly independent from the solution algorithms of CODA. So any changes do not effect the overall algorithm for the residual evaluation, which is favourable because the loops for the evaluation of the residual are highly optimized as they play a crucial role for the overall performance of a CFD solver. Other conceivable approaches for the hyperreduction, that directly interfere in these loops e.g. to evaluate only the fluxes for the selected residuals, are likely to significantly deteriorate the overall performance, which could offset the benefits from a reduced number of flux evaluations. A further advantage of the mesh-based approach is that the parallelization through mesh partitioning is unaffected. The domain defined by the submesh can still be split into subdomains and distributed to multiple processes, although one has to expect a loss in parallelization efficiency due to the size of the mesh and its separation into various unconnected parts. And finally, all the operations for the extraction of a submesh can be executed independently from the CFD solver, in the future even purely in the FSDM, as the mesh is already stored on the C++ level of the FSDM.

The procedure to extract such a submesh is sketched in fig. 3.3, which gives a schematic overview of the process. The whole extraction process is defined in a method of the Extractor-class, that contains e.g. helper methods and methods to visualize the extracted meshes. Required information for the extraction are the base mesh, the definitions of the boundary conditions J_{BC} , the size of the residual stencil p_d resulting from the spatial discretization, and the set of selected indices J_s (corresponding to volume elements or residual entries). The boundary conditions are defined by a set of marker values, which uniquely identify for every boundary face of the base mesh the imposed boundary condition at this face. So, when having a face-based representation of the original mesh at hand, the next step is to gather all faces belonging to a specific marker and therefore representing this boundary condition. As the second input from the CFD solver, the stencil,

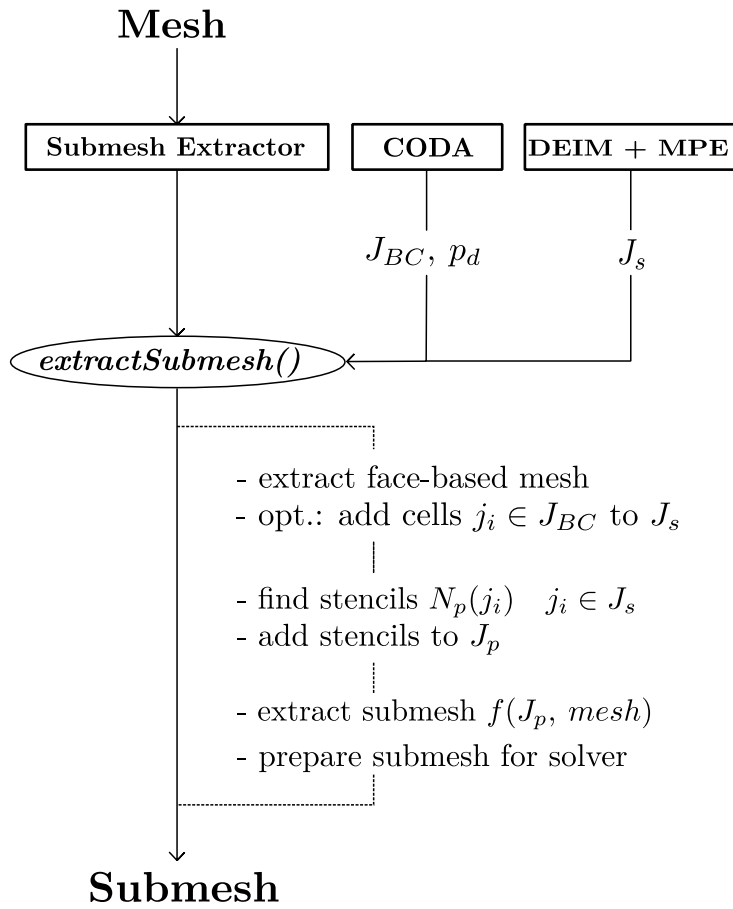


Figure 3.3: Submesh extraction for the hyperreduction of the LSQ-ROM.

which is determined by the spatial discretization of the solver (cf. section 2.1), defines the neighbourhood of every cell identified by the indices J_s . This neighbourhood has to be included in addition to J_s , since the residuals rely on the state values of their neighbours. The indices J_s are selected by the DEIM and MPE as described in section 2.4.

With the described inputs, the extraction process starts by obtaining the face-based representation of the original mesh. This face-based mesh is provided by the FSDM and includes every face of the mesh. The faces are defined by the indices of the cells, which share this face, and a value for the marker in case of a face at a boundary. From this, it is possible to retain the direct neighbours of every cell, which allows to recursively identify arbitrary stencil sizes for every cell. In addition to the selected cells from J_s , it is also possible to append cells at certain boundaries. These boundary cells are not chosen by the selected hyperreduction methods, but by a user input, that identifies the type of the boundary condition and the maximum amount of cells that are added for the specific boundary condition. The reason for this is elaborated in section 2.4.3. These cells and their stencils are added to J_s . Since the stencils are already defined by p_d , the next step is to add all cells included in $N_p(j_i)$ with $j_i \in J_s$ to the set of cells J_p , that identifies the submesh. J_p is then utilized to extract the corresponding geometrical information defining the submesh. The submesh still needs some preparation steps like the partitioning, which are executed right after its extraction in order to pass it directly to CODA.

To illustrate a submesh resulting from this procedure, fig. 3.4 displays three different sections of a submesh, generated from a mesh around a NLR7301 airfoil with 100 primarily selected cells and all farfield cells in addition. The cells are coloured depending on the neighbourhood they are belonging to. All the green coloured cells $i \in J_s$ are the ones that are selected via DEIM and MPE or selected because they consist of at least one boundary face. Yellow cells instead are direct neighbours of green ones, whereas orange cells belong to the neighbours of neighbours J_s . In (b) the submesh can be seen at the leading edge of the airfoil including several stencils without any connection to the rest of the mesh. A closer view of (b) is shown in (c) illustrating some stencils which are connected through cells belonging to multiple neighbourhoods. The cells of section (d) are included by the optional addition of boundary cells, which are in this example all the cells (and their neighbourhoods) adjoint to the farfield of the flow domain. When emphasizing that solely green cells are referring to the hyperreduced residual, it becomes clear that the effort for the residual computation is not only driven by the number of selected cells, but also by the stencils defined by the numerical schemes and the cell type. In a 2D case with a common second order scheme, a hexaeder cell implies 12 additional cells and in a 3D case even 24. However, because of the properties of the selection methods, it is likely that the submesh consists of clusters of cells reducing the number of pure "stencil cells". Furthermore, as the residual is computed by loop over all faces, the crucial parameter for a reduced effort is the number of faces in the submesh compared to the number in the original mesh. Another good estimate for the acceleration potential of a submesh residual computation is therefore the ratio of the total number of faces in both meshes. For the applied numerical schemes, it is possible to further reduce the effort by neglecting all outer faces of the cells of the outer most neighbourhood. However, the described approach still has some overhead, because the mesh still includes faces that are not necessary for the correct evaluation of the residuals from J_s . All the faces of the nearest neighbours that are not also a face of a cell from J_s , are still part of the flux evaluation, although these terms are not required. These faces are solely required for the construction of gradients or laplacians.

Preprocessor changes in CODA Before transferring a mesh into its CFD solver specific representation as a computational mesh, every solver should check if the passed mesh forms a consistent discretization of the spatial domain. As CODA also includes such consistency checks, it is not possible to directly compute a residual on a submesh like

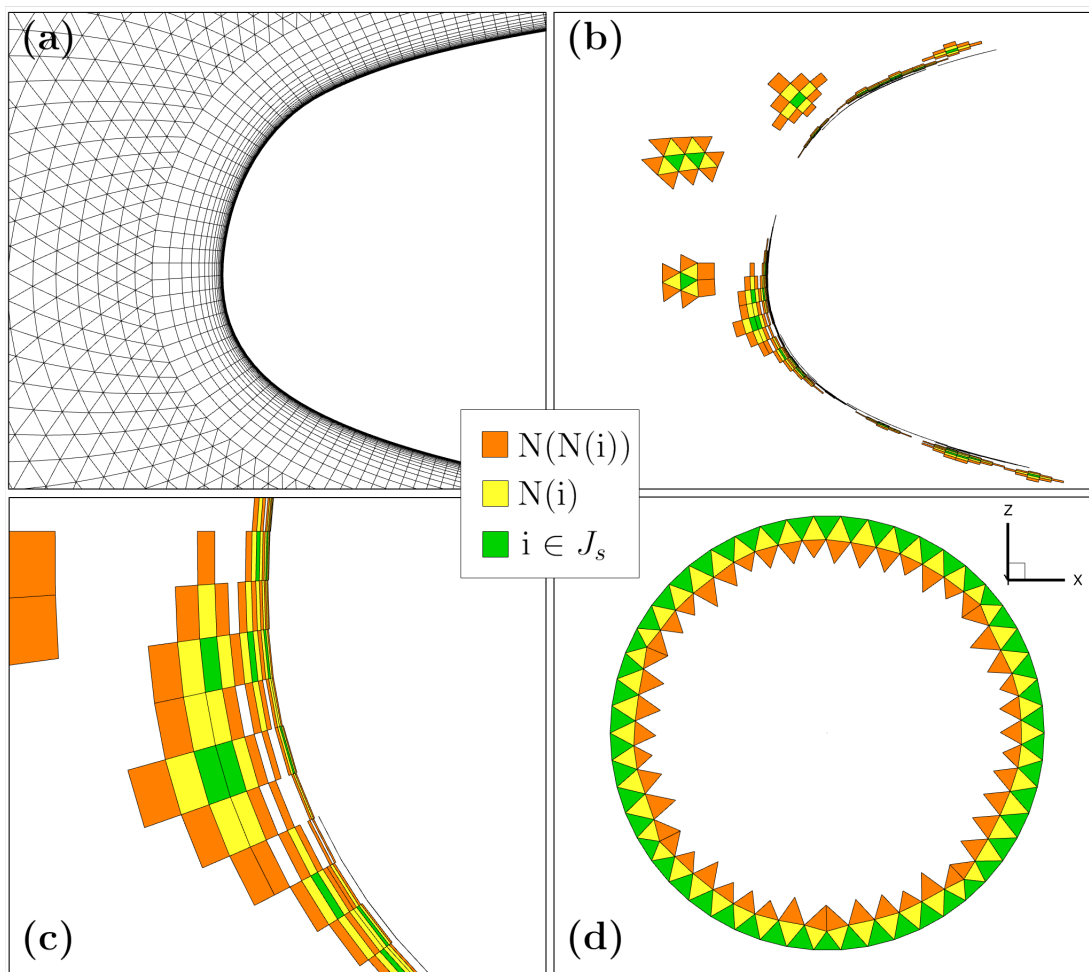


Figure 3.4: Submesh generated from a base mesh for the NLR7301 airfoil.

it is presented in the prior section. This is due to cells like in fig. 3.4 (a), that are not connected to either other cells or a boundary. The cells of the outer neighbourhood consist of faces, which are lacking a second cell as these cells were originally inner faces. Therefore, it is essential to ensure that such faces are just skipped when the faces for the face-based computational mesh are collected. This adaption together with turning off some consistency checks (e.g. checks for the correct number of faces for every cell) are sufficient to let CODA successfully operate on a submesh. As a result, the residual computation on the submesh provides correct solutions only for the primarily selected cells J_s and J_{BC} . Residual values from any other cell are incorrect as they are missing information from neighbouring cells or even some of their faces and their corresponding fluxes. Specific to the employed turbulence model of Spalart and Allmaras, which relies on the distance to the nearest solid wall for every cell, the changes involved also a security check, that these distances are imported from the values of the original mesh. As the submesh could lack some faces to a wall, a recomputation for the submesh would result in inaccurate values for the wall distance.

Adaptations of the LSQ-ROM implementation The approach to implement the hyperreduction based on a submesh requires modifications to the implementations of the LSQ-ROM method. This paragraph identifies the most important issues for an efficient implementation and it describes how these tasks are solved within this work. Nevertheless, future developments could tackle these issues in a more structured way. First, one has to organize the switching between the submesh and the mesh. The full mesh is required during the offline stage of the LSQ-ROM method and when computing the full solution based on the optimized POD coefficients, whereas the submesh is required in between these steps during the LM algorithm. However, this can be easily done on the control level by a method that repeats all preparation steps for the usage of the mesh in the CFD solver and the LSQ-ROM. The second issue is the efficient determination of the solution for the submesh. Referring to fig. 2.3, it can be seen that every updated set of POD coefficients \mathbf{a} leads to a recomputation of the state \mathbf{W} . In the POD basis \mathbf{U} , the effort for this is of $\mathcal{O}(n)$ and the obtained state still needs to be restricted to the intersection of submesh and original mesh. Because of that, it is more efficient to reconstruct the state for the submesh directly from a suitable POD base. A possible solution to this is to store a second POD, which is just a masked copy of the original POD. The masked basis $\hat{\mathbf{U}}$, which contains only entries corresponding to the s cells in the submesh, allows to compute the new state $\hat{\mathbf{W}}$ at costs $\mathcal{O}(s)$. The last challenge in this context is the identification of the valid residuals from the submesh residual vector. For that, the indices of the cells referring to J_s are needed, which can be obtained e.g. from the submesh extraction process. Only these residual entries are relevant for the evaluation of the objective function. A practicable solution is a simple indexing of the full submesh residual vector. Future extensions could include a recomputation of the hyperreduction indices during the prediction, which would also lead to a new submesh extraction. Since the submesh extraction is in almost every case faster than the selection of the indices, the extraction should not be the critical part performance-wise.

3.3 Validation and investigation test cases

A brief presentation of the meshes, the residual discretization and the settings for the LSQ-ROM is given in this section. These settings are chosen for all the investigations as long as it is not stated otherwise. Especially, the information about the discretization in the CFD solvers are kept general without going too much into the wide field of discretization approaches in CFD.

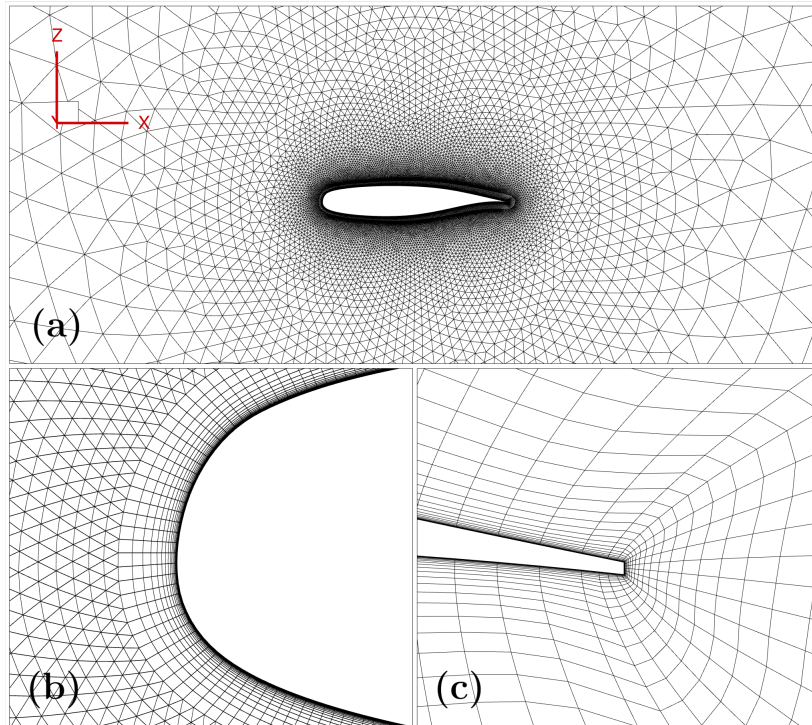


Figure 3.5: Mesh and geometry of the NLR7301 airfoil.

3.3.1 Meshes for the application in the CFD solver

This paragraph presents the meshes used to obtain the full CFD solutions and to perform the residual computations. They also constitute the base mesh for the construction of a submesh resulting from the hyperreduction. When comparing results of the LSQ-ROM based on different flow solvers like CODA and TAU, it is essential to keep in mind that the same mesh leads to different computational meshes for these solvers. This is due to the different approaches on how to form the finite volumes. While CODA follows a cell-centered approach, TAU uses a node-based approach. Therefore, the usage of the same mesh leads to different discretizations even when using the same numerical schemes.

2D airfoil: NLR7301 The NLR7301 airfoil is chosen for investigations in the subsonic and transonic flow regime. The airfoil has become a common case for numerical investigations of CFD solvers and was already considered in prior investigations of the LSQ-ROM method. Its geometry and the mesh used for the CFD solver are shown in fig. 3.5. In (a) the whole airfoil is given, while (b) presents a closer view of the leading edge and the hexahedron based mesh near the airfoil. The hexahedrons form a O-type mesh around the entire airfoil, which consists of highly stretched cells to resolve the gradients resulting from the boundary layer. From (c) one can see the trailing edge, which is not sharp and chopped instead. Besides the hexahedrons near the airfoil, the mesh consists purely of prisms. In total, there are 17313 hexahedron cells and 23609 prismatic cells. Assuming five unknowns per cell, like it is the case for the RANS equations with a one-equation turbulence model, the flow problem comprises 204610 degrees of freedom. However, these assertions are only correct for the cell-centered approach pursued in CODA. In TAU, whose computational grid is node-based, the same mesh results in a flow problem with 147205 unknowns as the mesh consists only of 29441 nodes.

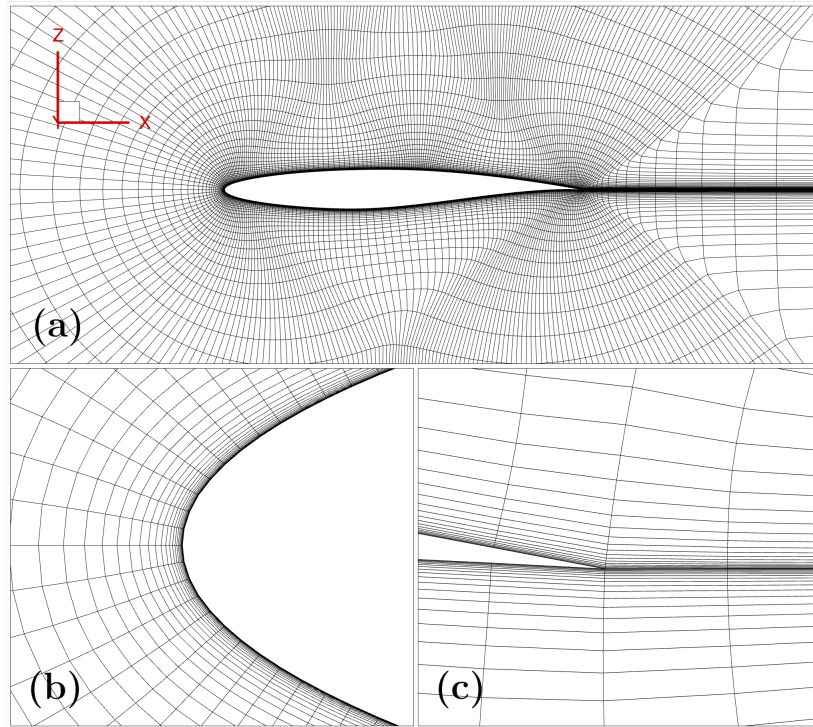


Figure 3.6: Mesh and geometry of the RAE2822 airfoil.

2D airfoil: RAE2822 Besides the NLR7301, the well-known airfoil RAE2822 is part of the investigations. It has been extensively studied in experimental investigations [21] and numerical investigations in the field of CFD [58] [70]. In particular, the mesh utilized in this work was generated and used for the studies in [70]. It constitutes a medium resolved mesh out of a sequence of successively refined grids. Several sections of this mesh are shown in fig. 3.6. Section (a) displays the hexahedron based mesh around the airfoil, while (b) and (c) exhibit closer views of the mesh at the leading edge and the trailing edge respectively. The hexahedrons in the proximity of the airfoil are highly stretched for a good resolution of the boundary layer. Resulting from the structured mesh generation, the cells behind the sharp trailing edge also embody high anisotropies. In total, the mesh consists of 20480 hexahedrons. Therefore, the flow problem modelled by the RANS equations plus a one equation turbulence model has a total number of unknowns of 102400 in case of the cell-centered approach in CODA. For the node-based approach in TAU, the flow is described by 104160 degrees of freedom as the mesh is composed by 20832 nodes.

3D airplane configuration: NASA-CRM In order to examine the implemented hyper-reduction for an industrial relevant 3D case, the NASA common research model (CRM) serves as a test case for a generic configuration of a civil transport plane. The CRM is an established geometry for numerical investigations of CFD codes [49]. It was part of several Drag Prediction Workshops [20], so that many results and adapted meshes can be found in [20]. The mesh, that is used in this work and shown in fig. 3.7, also originates from one of the workshops. The full geometry, as given in (a) together with the mesh, resembles fuselage and wing of current transonic aircraft configurations. Since the configuration is assumed to be symmetric, only one half of the configuration is part of the discretized flow domain in the shape of a hemisphere. A detailed view of the mesh around the nose of the CRM is given in (b), whereas (c) displays the slice through the mesh around the swept wing of the configuration. The mesh is purely based on hexahedrons, which are stretched around the solid CRM body to increase the number of cells in the

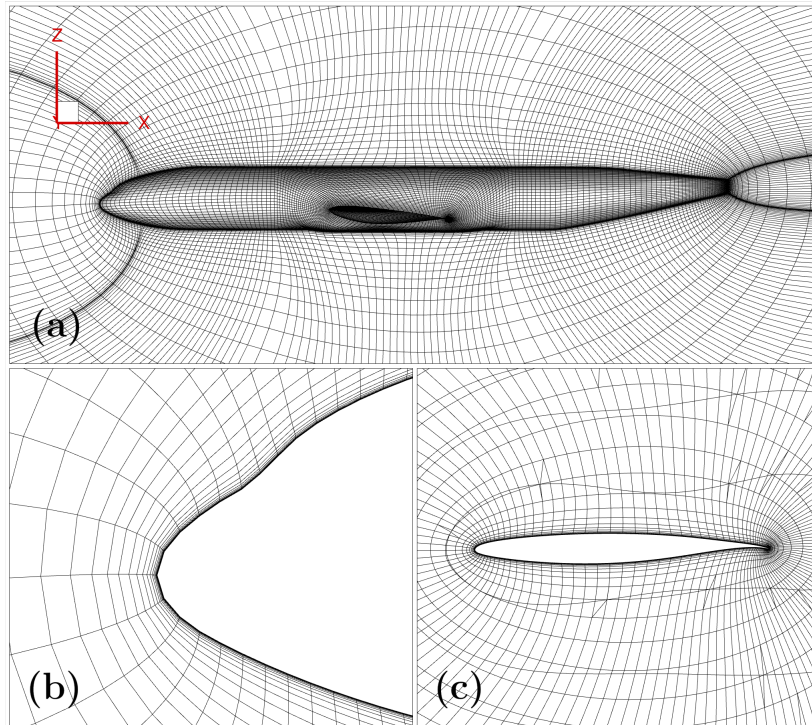


Figure 3.7: Mesh and geometry of the NASA-CRM configuration.

area of boundary layers. Summed up, the relatively coarse mesh for the CRM consists of 638976 cells. Modelling the flow through the RANS equations and the turbulence model of Spalart and Allmaras leads to a problem with 3833856 unknowns solved by CODA. This test case is not investigated with TAU.

3.3.2 Settings for the application of the LSQ-ROM

The section gives a brief description of the settings and input parameters for the CFD solver and the LSQ-ROM and which options are chosen for the investigations in chapter 4.

Discretization schemes in the CFD solvers

TAU and CODA have different native discretization schemes. While TAU's main discretization scheme for the discretization of the convective fluxes is a central scheme [58], CODA's development started with the famous upwind Roe scheme [51]. Although the central scheme became recently available also in CODA, its performance in CODA can not compete with the chosen Roe scheme. That is why different schemes are applied for both solver within this thesis. The Roe scheme in CODA is based on a reconstruction of states to obtain a discretization scheme that is of second order. As this reconstruction process can lead to instabilities e.g at strong discontinuities, a limiter is applied for the transonic flow cases [77]. Additionally, a positivity preservation of quantities, that need to stay in the physical meaningful range, is utilized to increase the robustness of solution process. The computation of the gradients is done by the Green-Gauss approach [34]. In TAU, a central schemes with scalar dissipation is applied, which is automatically a second order scheme and does not need any further reconstruction of states. Stabilization of this second order scheme is accomplished with a pressure switch [58]. The gradients are also

reconstructed through application of the Green-Gauss approach. It is mentionable that the gradient computation in TAU and CODA is different, although the used approach is the same. Because of the different ways the finite volumes are formed in TAU and CODA, the gradients and every numerical scheme relying on the volumes leads to varying results between the solvers.

LSQ-ROM settings

Based on the CFD solutions as the snapshot data, the POD basis is computed by an eigenvalue decomposition as described in section 2.2. Before that, the snapshots are centered around their mean. This reduced basis is then utilized to select the hyperreduction indices via DEIM and MPE algorithms presented in section 2.4. The algorithm is stopped as soon as the predefined number of indices is reached. The number of selected indices and the number of included POD modes define the reduction levels for the model and are often part of parameter variations for the investigations in chapter 4. A higher number of settings need to be chosen for the online stage of the LSQ-ROM. For the estimation of the start coefficients for the LM algorithm, a Thin Plate Spline interpolation is chosen (cf. section 2.3), which is augmented by an additional constant trend function and scaling of the control points to the unit hypercube. The following optimization of the coefficients relies on the LM algorithm as outlined in section 2.5. For that, the full Jacobian is constructed by a forward finite difference once at the beginning of the optimization and then updated via Broyden update, whose approximation quality is improved by a gradient computation based on finite difference. The residuals are scaled by the square root of the cell volumes to properly adjust the weighting of the residuals, whose magnitude is depending on the cell size. The native scaling of the residuals is the full cell volume, however, this scaling can lead the optimization process towards solutions that are not in line with the farfield boundary condition. The reason for this is that common meshes consists of way more cells around the airfoil than at the farfield, so that the summed residual is dominated by entries corresponding to cells far from the exterior state. Therefore, violations of the imposed farfield conditions can be counterbalanced by smaller residual reductions at the airfoil. Additional options like the selection of specific residual variables for the evaluation of the objective function value and the scaling of it with its initial value are not considered, but could serve as a starting point for further tuning of the prediction quality of the LSQ-ROM.

3.3.3 Prediction error metrics

The accuracy of the predictions with the LSQ-ROM are assessed based on different error metrics depending on the desired level of detail for the error analysis. A highly detailed view on the error of a prediction can be obtained, when computing the relative error field for a certain variable $\mathbf{z} \in \mathbb{R}^n$ e.g the pressure. For that, the difference between the predicted solution of the LSQ-ROM and the reference CFD solution is normalized by the values of the reference solution. So, the relative error is defined by:

$$\text{relative error of } \mathbf{z} = 100 \cdot \frac{\mathbf{z}_{\text{LSQ-ROM}} - \mathbf{z}_{\text{CFD}}}{\mathbf{z}_{\text{CFD}}} \quad (3.1)$$

However, in some cases a scalar measure for the assessment of the prediction accuracy is required to examine and compare the accuracy of a larger number of predictions. Therefore, to obtain a quick overview of the prediction quality, it is possible to utilize the coefficient of determination (CoD) as an integral measure for the agreement of the predicted solution with the reference solution of the FOM. The coefficient of determination

is applied to every value in the flow field for the variable z , so it can be computed by:

$$\text{CoD} = 1 - \frac{\sum_{i=1}^n (z_{i,\text{CFD}} - z_{i,\text{pred}})^2}{\sum_{i=1}^n (z_{i,\text{CFD}} - \bar{z})^2}. \quad (3.2)$$

The higher the agreement is between the prediction and the reference solution the closer the coefficient of determination is to 1.0. For predictions deviating from the reference solution, the coefficient of determination is decreasing towards 0; for really high errors eventually even below 0. However, as the predictions with the LSQ-ROM are normally small for major parts of the flow field, the coefficient of determination usually exhibits values very close to one. Because of that, a second integral error metric is used in some cases. It is defined by root of the sum over all squared differences between the predicted quantity z and the value from the CFD solution and termed as RSSE.

$$\text{RSSE} = \sqrt{\sum_{i=1}^n (z_i - z_{i,\text{CFD}})^2} \quad (3.3)$$

A perfect agreement between the two solutions results in $\text{RSSE} = 0.0$, while any deviation between them increases the value for this error metric.

Chapter 4

Results

The depiction of the investigation results is split into two sections. The first part presents the results of a validation study for the implementation that enabled the application of the LSQ-ROM method together with the new flowsolver CODA without considering the hyperreduction. In particular, results from the LSQ-ROM coupled with TAU, which were investigated in relatively high detail, serve as comparison for the LSQ-ROM with CODA. Although both CFD solvers rely on different discretizations, outputs from the overall model are expected to lead to comparable results. The investigations for this comparison include two different 2D cases, one in a subsonic flow regime and one in the transonic regime. The same cases are further examined in the second part of this chapter, which focuses on the findings for the hyperreduction, including a comparison of the former hyperreduction implementation and the new consistent implementation. Besides the 2D cases, a 3D test case is investigated to assess the performance of the hyperreduction for relevant industrial configurations.

4.1 Validation of the LSQ-ROM coupled with CODA

The comparison of the results of the LSQ-ROM coupled with TAU or CODA is first carried out for the subsonic flow around the NLR7301 airfoil and then for the transonic flow around the RAE2822 airfoil. The LSQ-ROM prediction of a flow field is conducted for both cases and both CFD solvers, including the full offline stage required to generate the training data. The settings for the generation of the training snapshots are summarized for both solvers together with the settings for the optimizer in section 3.3.2.

4.1.1 NLR7301: subsonic flow

The starting point of the LSQ-ROM is the definition of the design space, which is spanned by the parameters Mach number and the angle of attack (alpha). The ranges for these parameters are:

$$\text{Mach} = [0.2, 0.4] \quad \text{angle of attack} = [-4^\circ, 8^\circ]$$

Within the given ranges, the training points ψ_i are defined by a two-dimensional Halton sequence that leads to a pseudo-random and deterministic distribution in the parameter space. The locations of the training points are illustrated in fig. 4.1 as blue dots. The red dot indicates the arbitrary test case at the conditions

$$\text{Mach} = 0.25, \quad \text{angle of attack} = 6^\circ$$

where a prediction of the flow is performed with the LSQ-ROM coupled with both TAU and CODA. The CFD solutions are based on the mesh presented in section 3.3.1 and are computed for a Reynolds number of $1.7 \cdot 10^6$. After computing the set of training snapshots

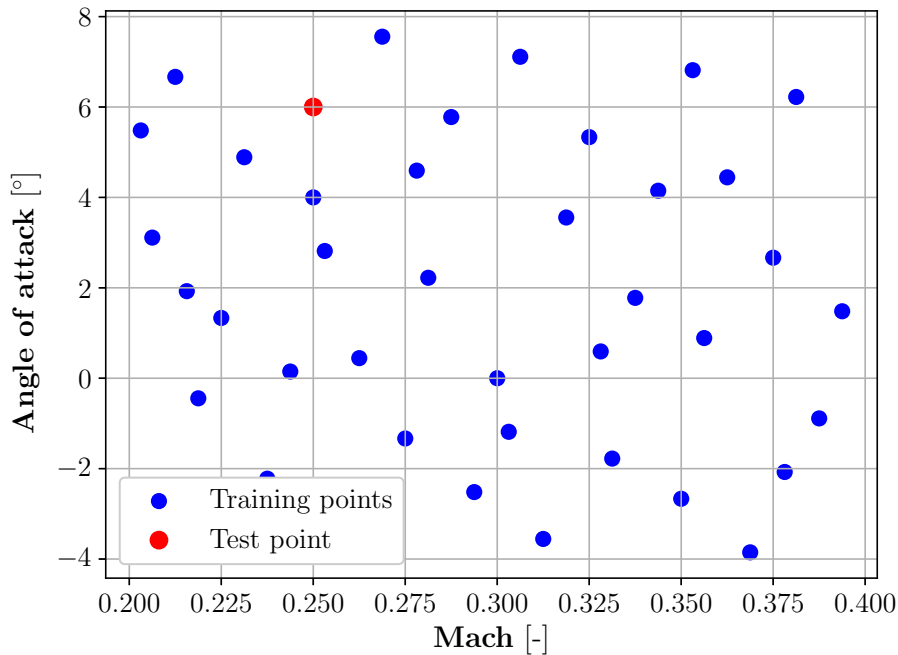


Figure 4.1: Design space for the prediction of the subsonic flow around the NLR7301.

both for TAU and CODA and defining the related POD subspaces, the POD modes can be visualized to analyze the information content represented by the most relevant modes. A comparison of the mean of all snapshots and the two most relevant POD modes are given in fig. 4.2. The left figures correspond to modes of the snapshots generated with CODA, while the right ones refer to the TAU solutions. Three additional POD modes are displayed in fig. 4.3. While the first row of fig. 4.2 shows the mean value of the density fields of the training snapshots, all the further subfigures in fig. 4.2 and fig. 4.3 illustrate POD modes, which are centered around the mean. The mean density fields are quite similar for the snapshots from CODA and TAU. The first three modes, which are also the three modes with the highest information content, capture primarily density variations around the nose at the airfoil. This is because of the varying location of the stagnation point for the snapshots with different angles of attack. When comparing the modes between the two data sets from CODA (left) and TAU (right), it is possible to see that the first two modes are in good agreement, while the modes with a lower information content (3, 4 and 5) have some differences. However, this is not surprising as these modes represent less important flow features and the flow solutions from CODA and TAU are not perfectly equal due to their differing approaches for the spatial discretization. Because of that, the information content in the training data differs to a limited extent and therefore also its representation in a POD basis.

An overview of the varying information content of the POD modes is shown in fig. 4.4 where the relative information content of the POD basis is plotted as a function of different number of modes included in the model. For both POD models, the *RIC* shows a very similar trend, although the differences become larger with increasing *RIC*. This meets the expectation resulting from the comparison of some of the POD modes (e.g. in fig. 4.2), which feature similar characteristics and should therefore refer to similar eigenvalues, too. It is remarkable that only 5 POD modes are necessary to construct a model that contains more than 99.99% of the information content of the total POD basis. The fact that a small number of modes is sufficient to capture a major part of the flow phenomena within the design space is the main motivation that led to the development of POD-based ROMs.

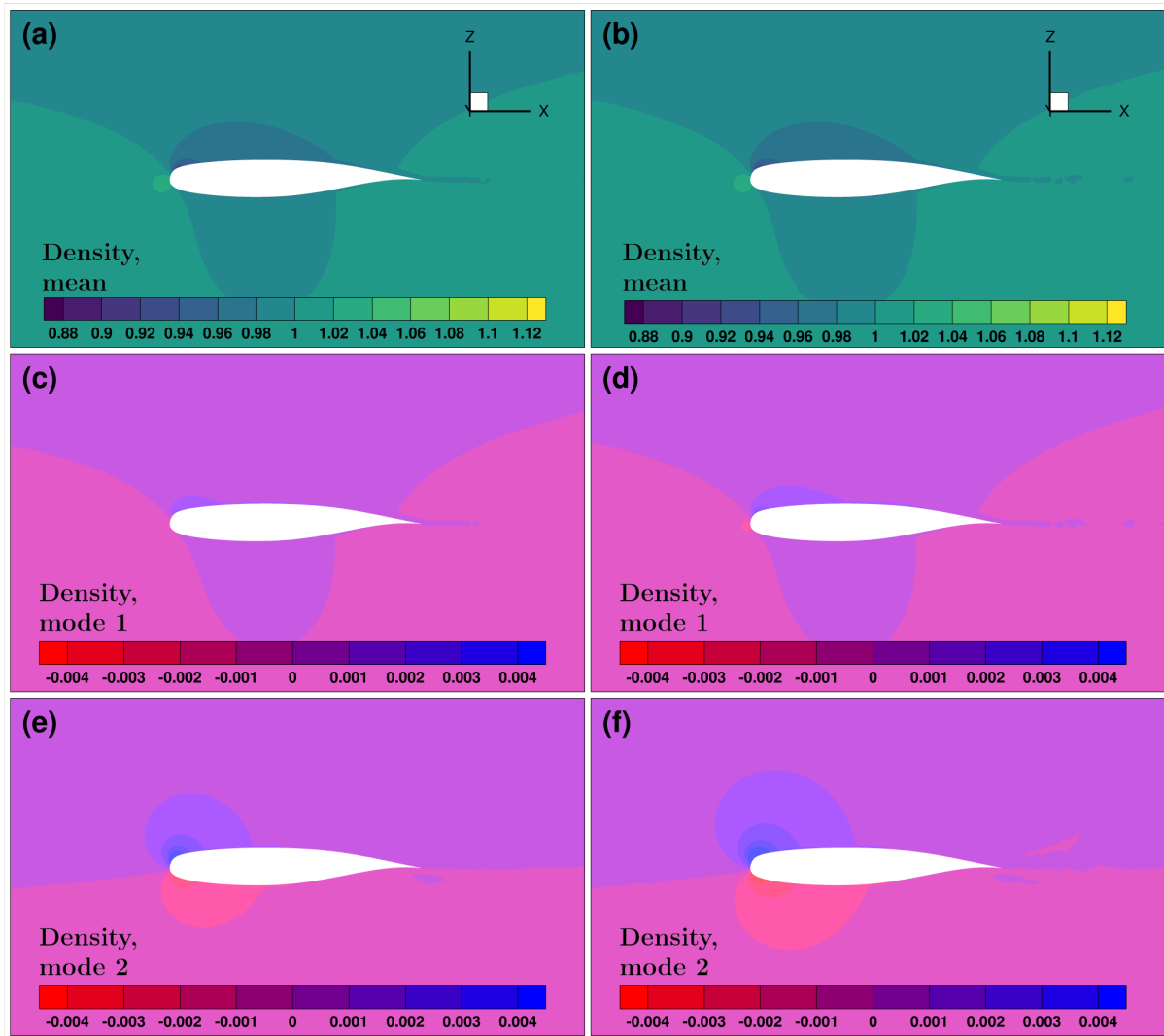


Figure 4.2: NLR7301: Mean and POD modes of Density from CODA (left) and TAU (right) solutions.

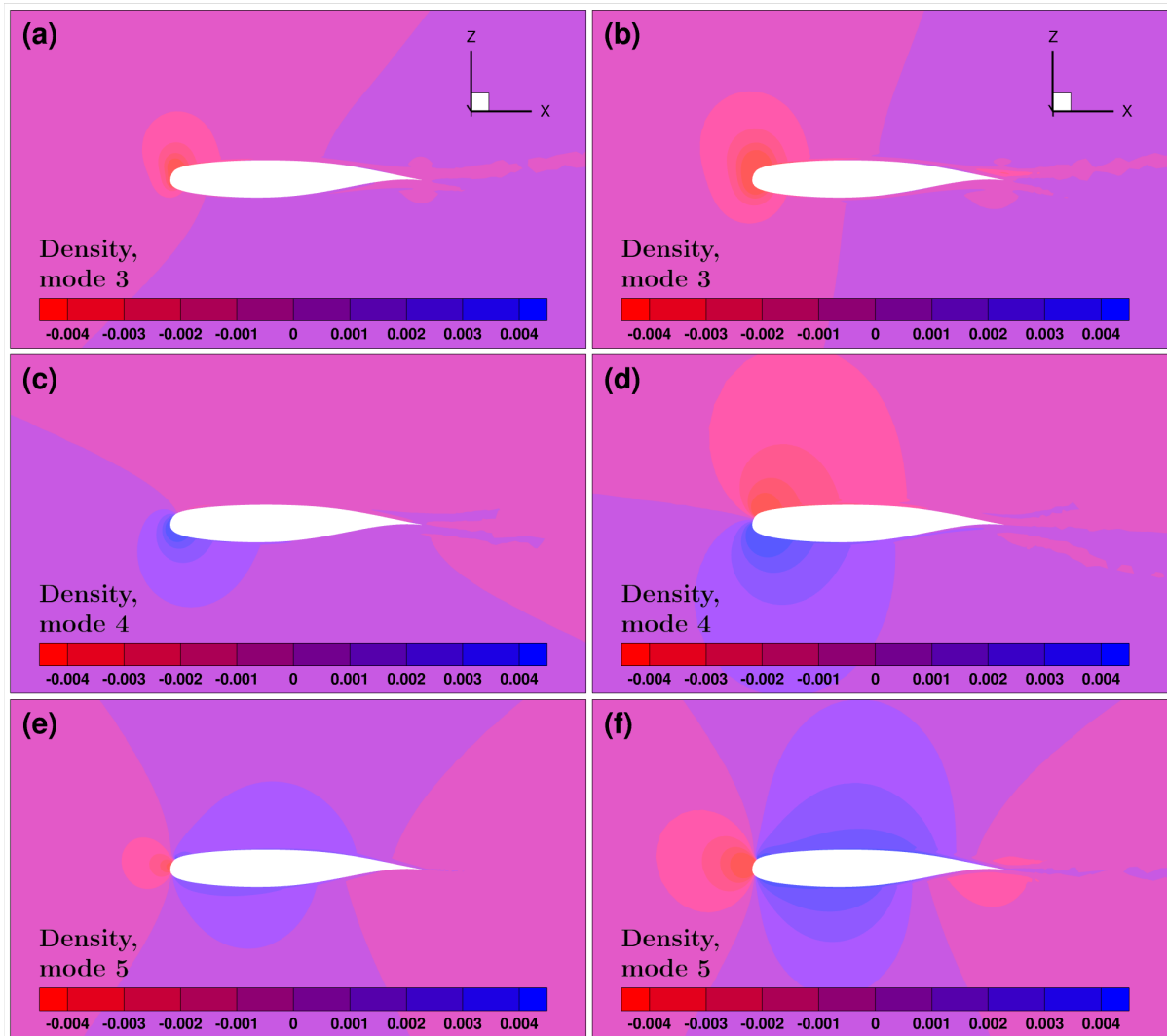


Figure 4.3: NLR7301: Three POD modes of Density from CODA (left) and TAU (right) solutions.

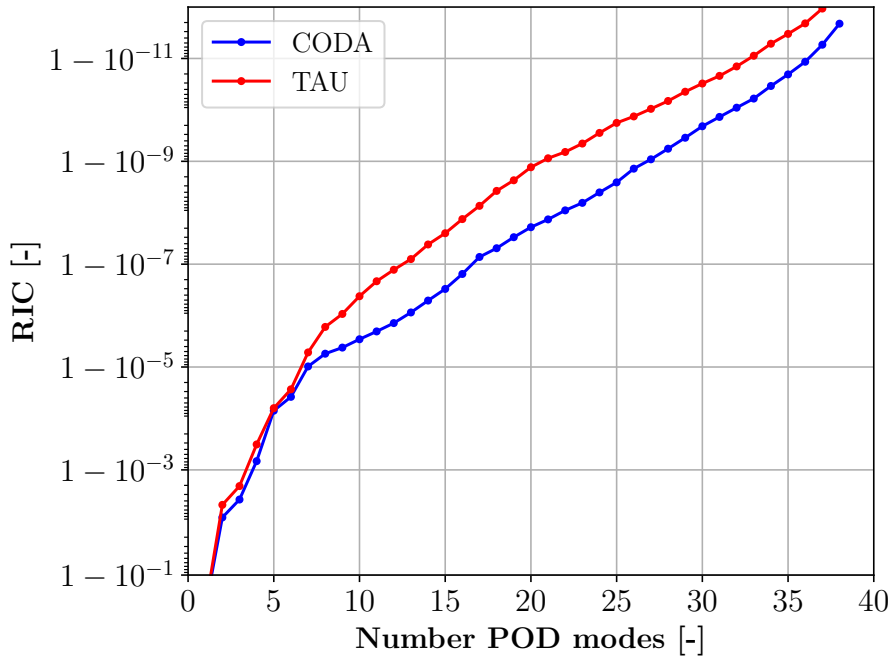


Figure 4.4: NLR7301: Relative information content of POD bases.

Prediction of the test point Figure 4.5 shows the Mach flow field of the two reference CFD solutions for the test case (Mach = 0.25, angle of attack = 6°) obtained with CODA and TAU. Both flow fields depict reasonable characteristics with a stagnation point at the lower side of the airfoil around the nose and the expected suction and pressure side. The two solutions are in good agreement to each other showing only minor differences for the location of the iso-Mach lines. Applying the LSQ-ROM with a compression rate of 0.05 for the prediction of this test case results in the solution for the Mach field shown in fig. 4.6. Although the illustrated solution is solely based on the mean value and two additional POD modes, the overall flow field is already well approximated, when comparing it to the corresponding reference solution. Hence, subsonic flow, at least for this test case, is suited to be approximated in a low-dimensional subspace spanned by a small number of POD modes.

To quantify the deviations between predicted LSQ-ROM solutions and reference CFD solutions, fig. 4.7 presents the relative error between these solutions, which is defined according to eq. (3.1). The relative error in pressure is given in fig. 4.7 side by side for LSQ-ROM solutions based on CODA (a, c and e) and TAU (b, d and f) for different numbers of POD modes. For an approximation with two modes (a and b) both solutions depict errors which are below 5%. The highest deviations for the pressure are located at the leading edge of the airfoil, where the flow field has the highest gradients and is strongly influenced by the input parameter α . This pattern can still be seen for the solution based on ten POD modes (c and d). Moreover, the addition of POD modes improves the overall accuracy and decreases the maximum errors below 0.1% and even further for the solutions including all POD modes (e and f). The comparison of these results with respect to the used CFD solver shows similar trends for the accuracy of the predicted solutions and its underlying POD basis. Besides the accuracy, the potential time savings are of interest when comparing the LSQ-ROM coupled with the new CFD solver CODA. Figure 4.8 reports the plot of the relative prediction time as a function of the compression rate, where the relative prediction time is the time needed for the online stage of the LSQ-ROM normalized by the computational time of the reference CFD solution. Furthermore, fig. 4.8 shows also the influence of the compression rate on

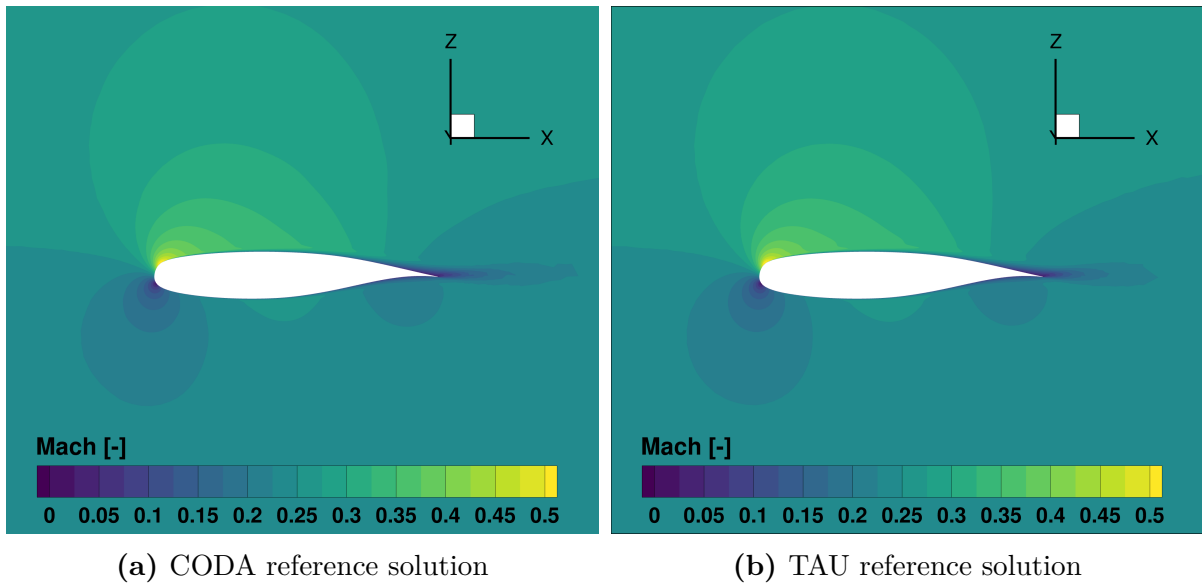


Figure 4.5: NLR7301: CFD reference solutions at $M = 0.25$ and $\alpha = 6^\circ$.

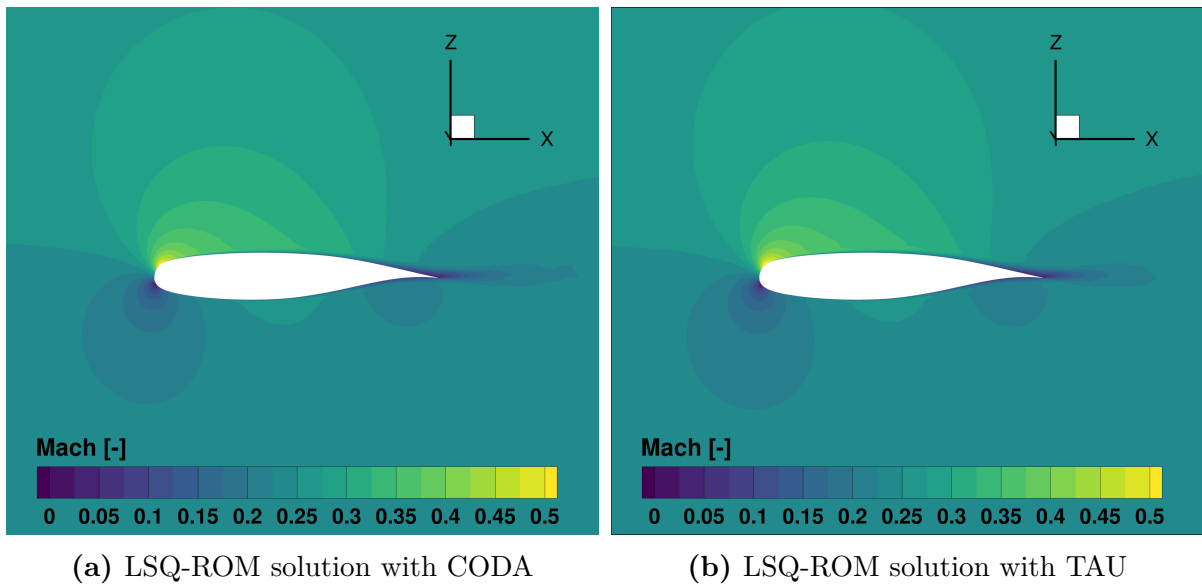


Figure 4.6: NLR7301: LSQ-ROM solutions for $M = 0.25$ and $\alpha = 6^\circ$ and a compression rate of 0.05.

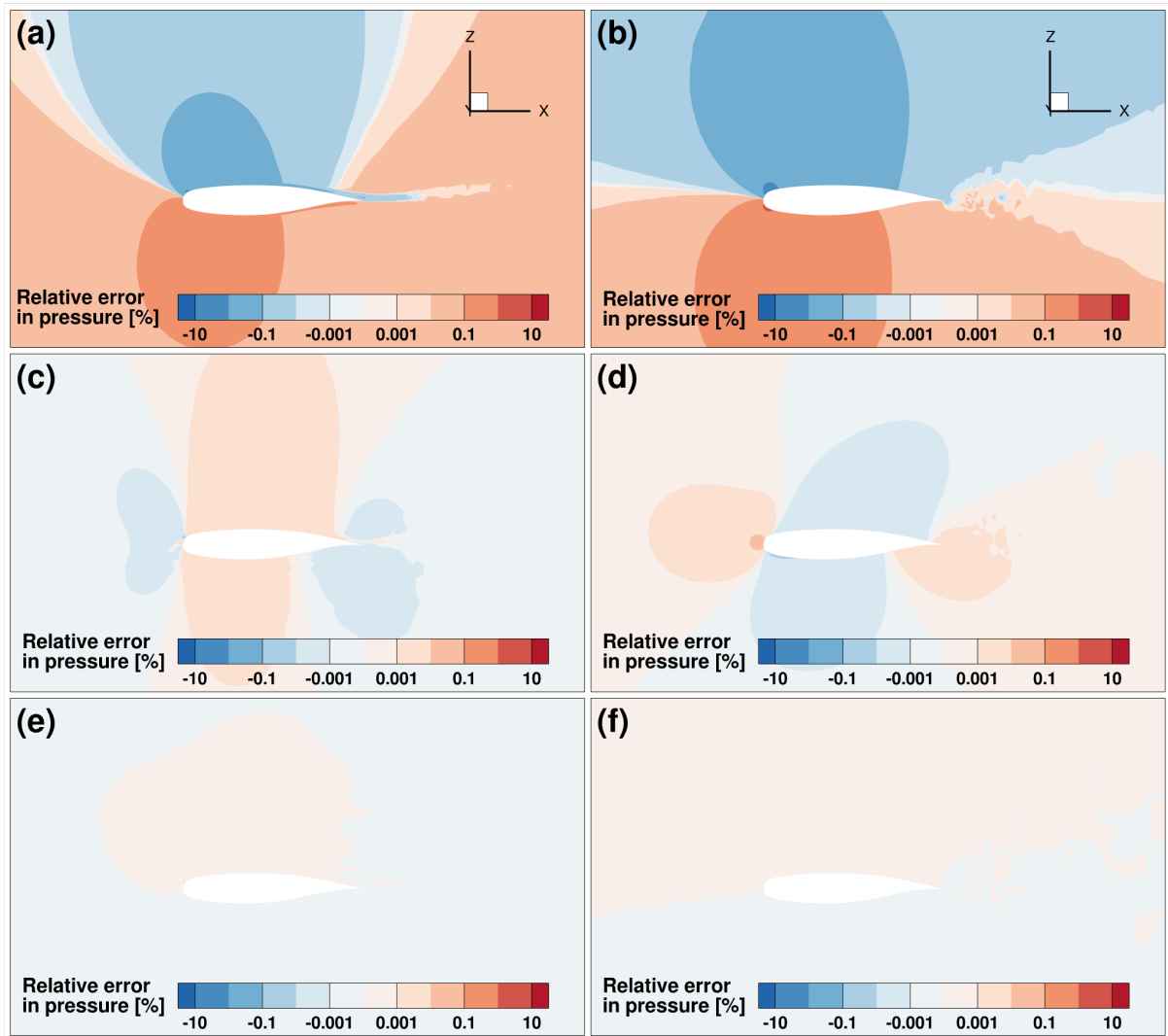


Figure 4.7: NLR7301: Relative error in pressure for a predicted solution with 2, 10 and 39 POD modes.

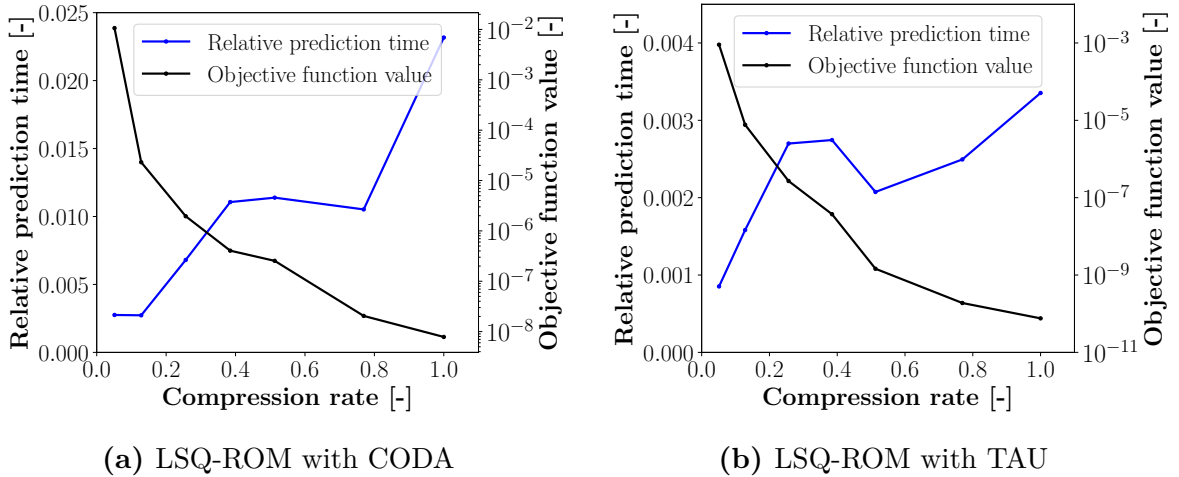


Figure 4.8: NLR7301: Relative prediction time and objective function value over compression rate.

the final objective function value of the residual minimization process. From fig. 4.8 it can be seen that the prediction time generally increases with a higher compression rate, which is expected as a higher compression rate corresponds to higher number of POD modes resulting in an increased effort per iteration in the LM algorithm. Moreover, the effort for the construction of the Jacobian increases and more iterations are usually required to converge to a local minimum because of the larger problem dimensionality. Besides some outliers due to varying number of iterations in the optimization process, both LSQ-ROM results confirm this trend. In addition, a comparison between the values of the relative prediction time for the LSQ-ROM with CODA and TAU seems to indicate that the online stage relying on TAU is significantly faster than the counterpart with CODA. This result is mainly caused by the fact that the computation of the reference solution with TAU took almost 2.5 times more time than with CODA (the computational time for reference CFD solution is at the denominator of the relative prediction time). It is also important to report that the residual computation in TAU is based on a relatively low effort central scheme, while the residual computation in CODA utilizes a more expensive upwind scheme.

The final observation about fig. 4.8 is that the objective function value, which is the summed squared residual, is monotonically decreasing for an increasing compression rate. This is the expected behaviour for the optimizer since considering more POD modes leads to extra degrees of freedom for the approximation of the solution, and therefore potentially to a solution that further decreases the residual.

4.1.2 RAE2822: transonic flow

The same analyses presented for the subsonic case are completed also for the prediction of transonic flows around the 2D airfoil RAE2822. The airfoil geometry and the corresponding mesh for its discretization are described in section 3.3.1. In contrast to the subsonic regime, the transonic regime is influenced by strong nonlinear flow phenomena like compression shocks that need to be captured by the solutions of the LSQ-ROM. Therefore, the next results prove that the LSQ-ROM coupled with CODA behaves similar to the LSQ-ROM with TAU in the presence of strong nonlinearities. First of all, the design space for the training points is defined by a Halton sequence for the parameters Mach

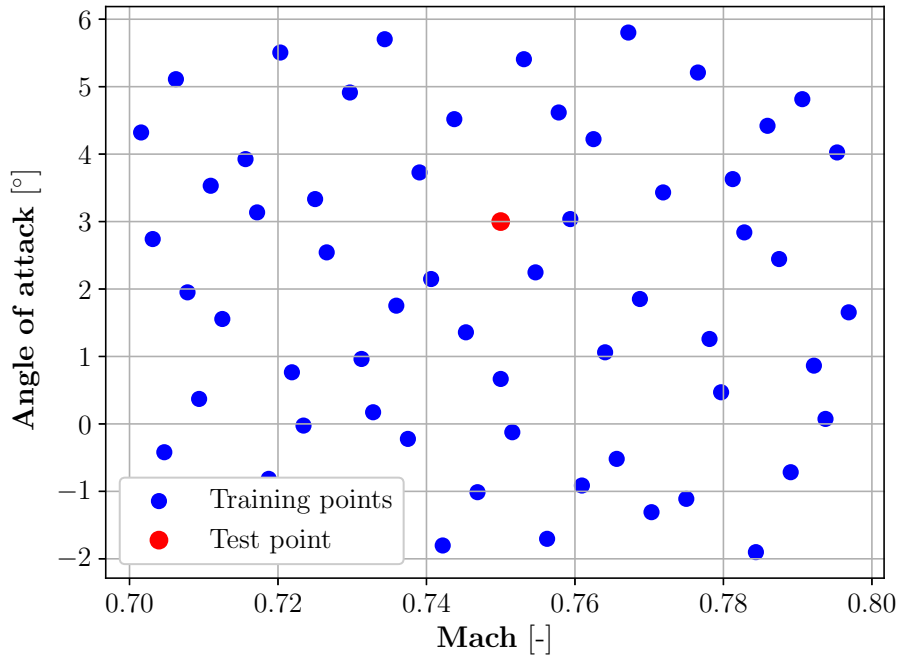


Figure 4.9: Design space for the prediction of the transonic flow around the RAE2822.

number and angle of attack in the ranges:

$$\text{Mach} = [0.7, 0.8] \quad \text{angle of attack} = [-2^\circ, 6^\circ]$$

The total number of points is 60, and their location in the two dimensional design space are illustrated in fig. 4.9. Besides the blue colored training points, the test point at

$$\text{Mach} = 0.75 \quad \text{angle of attack} = 3.0^\circ$$

is represented in the figure as a red dot. For all points, a Reynolds number of $6.5 \cdot 10^6$ is chosen and the resulting snapshots are used to perform a POD, whose modes are displayed in fig. 4.10 and fig. 4.11 that compare side by side the mean and the first modes resulting from the CODA (left) and the TAU (right) snapshots. The mean density fields show again a good agreement featuring low density at the upper suction side and higher densities at the nose near the stagnation point and at the lower pressure side of the airfoil. The first POD mode features two relatively large density fluctuations, which are a negative one at the leading upper side and a positive one at the leading lower side. All further modes possess strong fluctuations at the rear upper side, where a shock is located for a large majority of the training points. Because the position and characteristic of this shock is quite sensitive to both Mach number and angle of attack, the POD modes have to comprise information of this important flow feature. In addition, the shock strength and location does not linearly dependent on global parameters such a the angle of attack. However, with focus on the comparison between the results based on CODA and TAU, it can be seen that the first five POD modes include reasonable flow characteristics and their basic patterns match well. This aspect is particularly important when considering the different discretization schemes utilized in CODA and TAU, which should lead to deviations in the flow solutions particularly in regions of large gradients. The amount of information captured by a specific number of POD modes is illustrated by the relative information content in fig. 4.12 for both POD bases. First, its obvious to see that an incorporation of additional modes increases the RIC of the model. This increase is very similar for the two bases, indicating a good agreement of the information included in

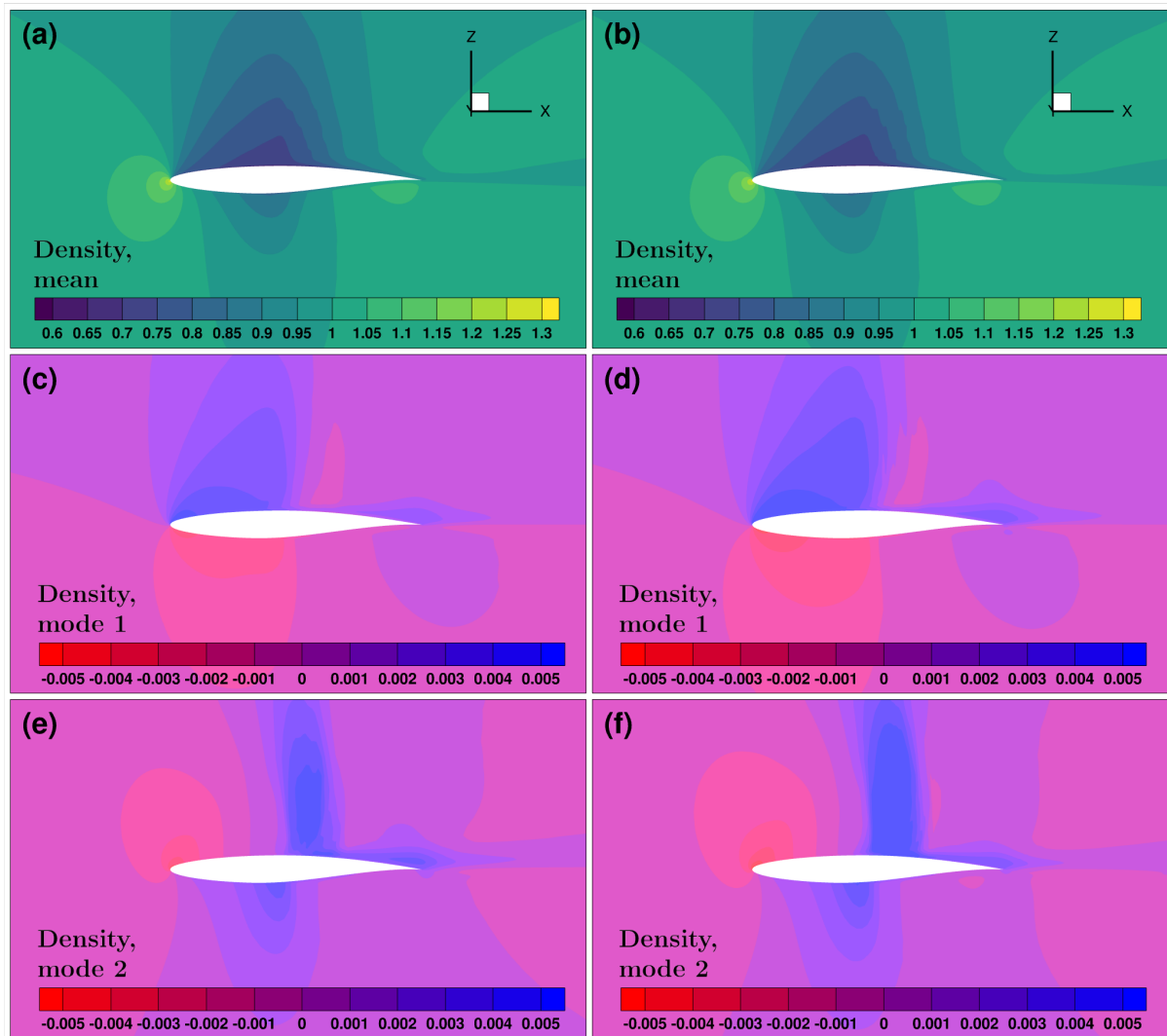


Figure 4.10: RAE2822: Mean and first two POD modes of Density from CODA (left) and TAU (right) solutions.

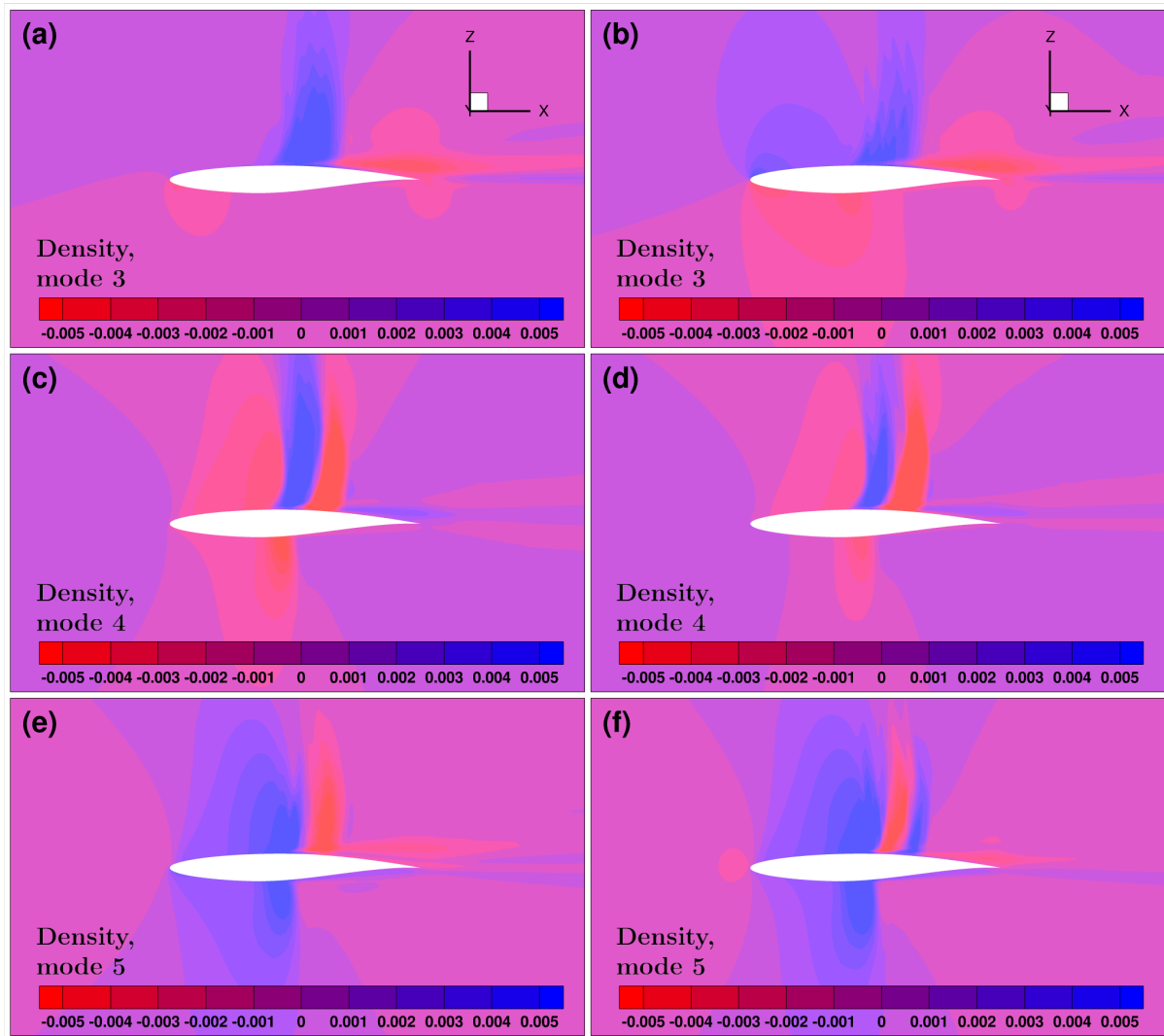


Figure 4.11: RAE2822: Third to fifth POD mode of Density from CODA (left) and TAU (right) solutions.

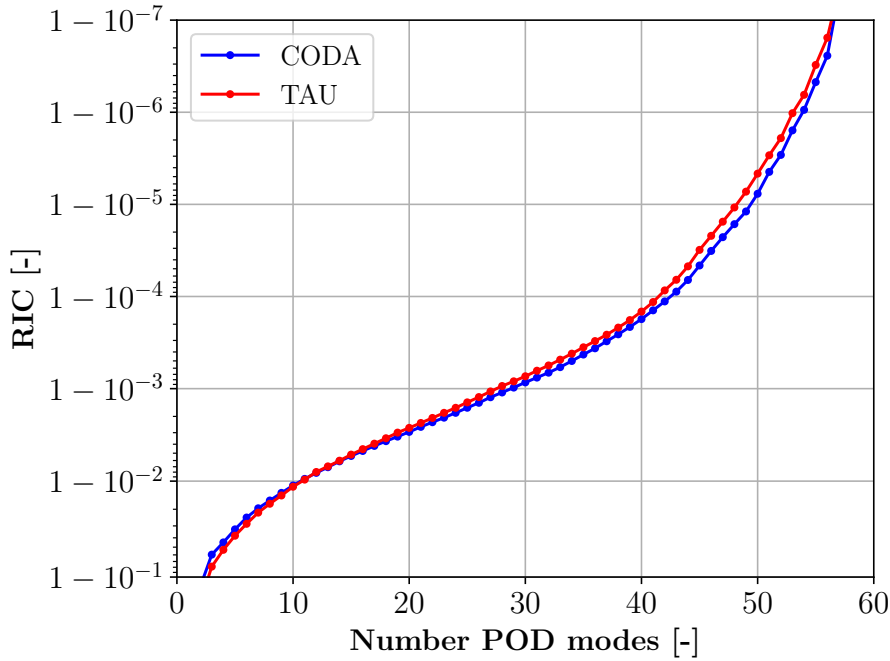


Figure 4.12: RAE2822: Relative information content of POD bases.

every mode. When comparing the values of the RIC with the ones from the subsonic case in fig. 4.4, it becomes clear that a given number of modes includes less information in the transonic case than in the subsonic condition. In fact, only 5 modes out of 39 are needed to capture 99.99% of the information of the POD basis in the subsonic regime, while around 42 of 59 modes are required in the transonic regime. This behaviour was already observed in [3] and is most probably due to the presence of compression shocks that induce strongly nonlinear behaviour, which requires more POD modes due to the linear combination of the modes.

Prediction of the test point The comparison of the CFD reference solutions for the test point at Mach = 0.75 and $\alpha = 3^\circ$ given in fig. 4.13 shows similar results for the Mach field from CODA (left) and TAU (right). Both solutions embody a compression shock at around 60% of the chord length that decelerates the flow from supersonic speed of around Mach = 1.35 to subsonic speeds of around 0.85. At the same time, the flow at the pressure side of the airfoil stays subsonic. Behind the shock, a wake develops with a thickening boundary layer. These phenomena can be found in both solutions and only minor differences are present around the location of the shock and in the wake of the airfoil. So, as already described for the comparison of the POD modes, the flow field solutions from CODA and TAU give comparable and reasonable results. The LSQ-ROM is now applied to predict the flow at the test conditions leading to the predicted Mach field in fig. 4.14 for a compression rate of 1.0. In contrast to the subsonic case, the predicted solution has some noticeable deviations, when compared to the reference solution, even when including all POD modes.

These disparities are mostly localized around the compression shock, as it can be seen from fig. 4.15. The magnitude of the maximum error is around 20% for the prediction with LSQ-ROM + CODA and around 25% for LSQ-ROM + TAU. This underprediction of the pressure is related to the shock location as the LSQ-ROM solution predicts the location a bit more towards the rear. Hence, the predicted pressure is lower at the location of the shock of the CFD solution. The rest of the flow field has lower deviations, which are

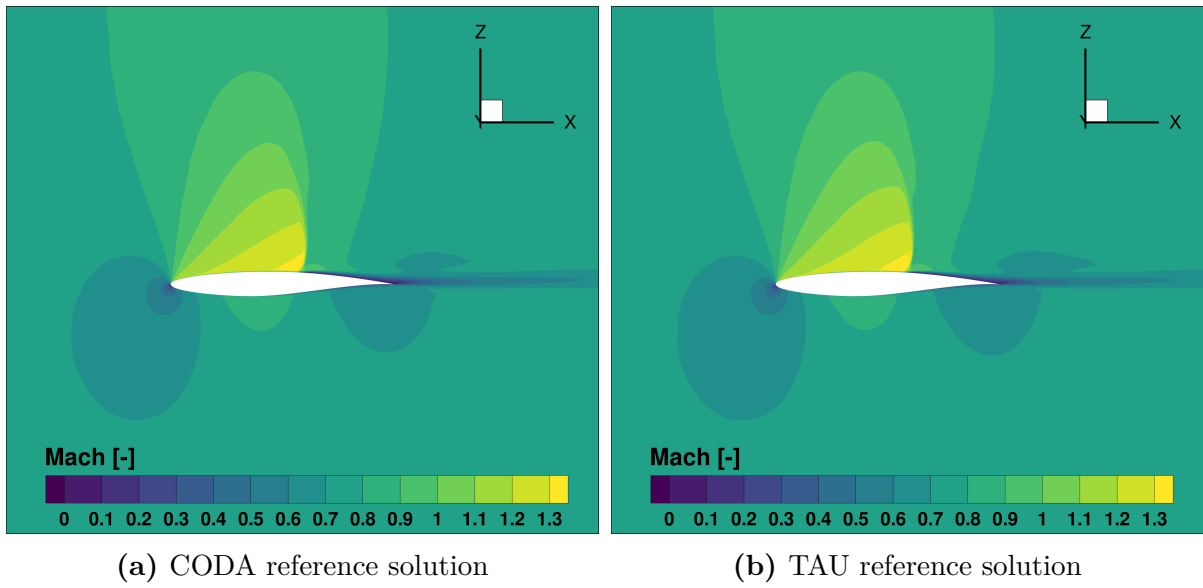


Figure 4.13: RAE2822: CFD reference solutions at $M = 0.75$ and $\alpha = 3^\circ$.

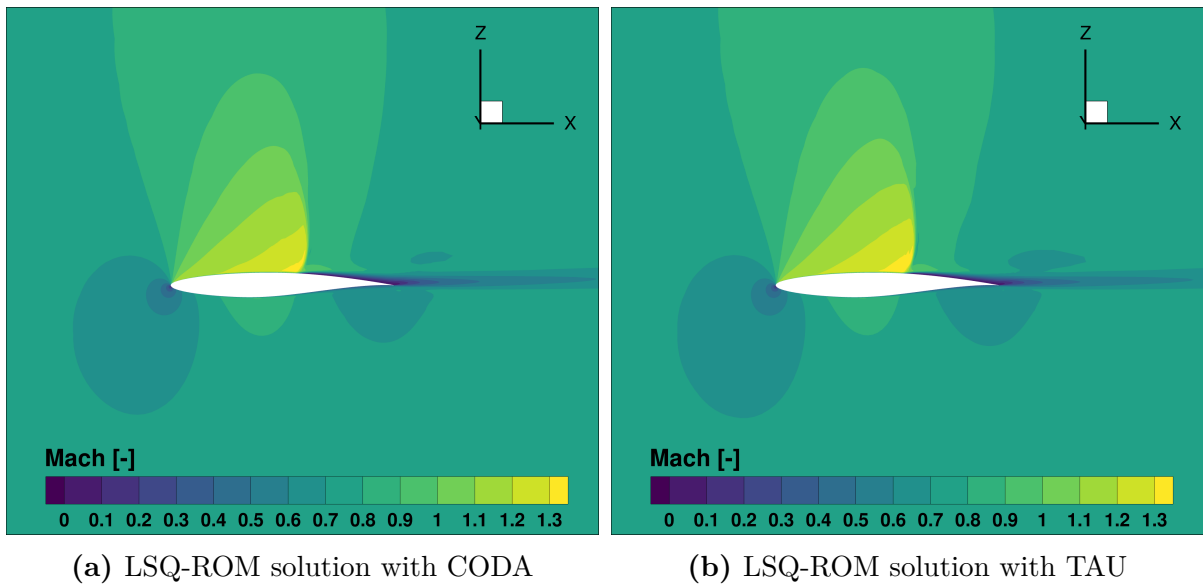
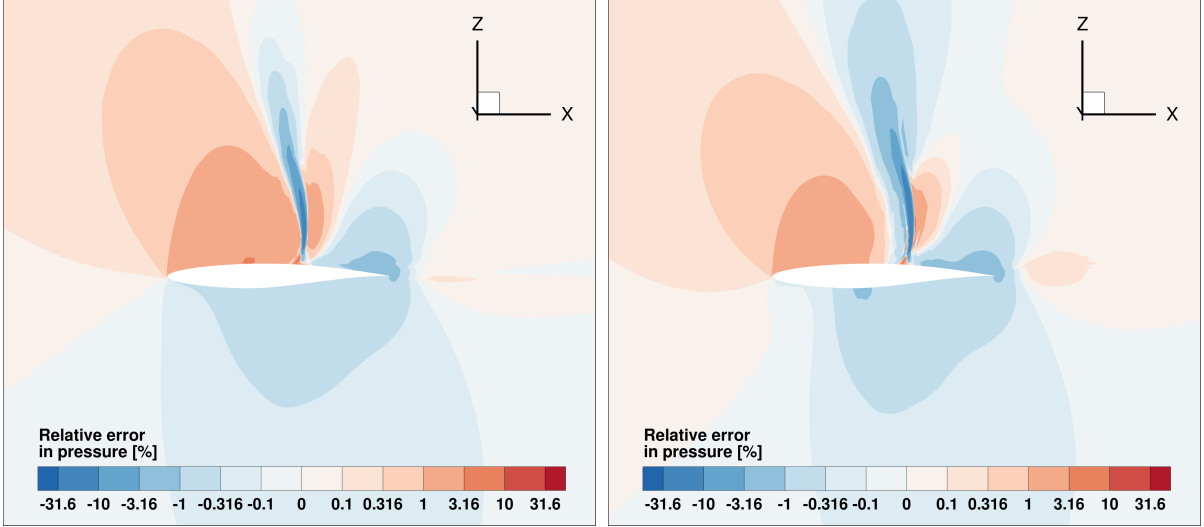


Figure 4.14: RAE2822: Solutions predicted with the LSQ-ROM at $M = 0.75$ and $\alpha = 3^\circ$ and with a compression rate of 1.0.



(a) Relative error of the LSQ-ROM solution with CODA (b) Relative error of the LSQ-ROM solution with TAU

Figure 4.15: RAE2822: Relative error in pressure of the LSQ-ROM at $M = 0.75$ and $\alpha = 3^\circ$ and with a compression rate of 1.0.

mostly below 5%. To compare the accuracy of the method for different levels of reduction through restriction of the included POD modes, fig. 4.16 illustrates the pressure coefficient at the surface of the airfoil for different compression rates. The pressure coefficient C_p is defined as

$$C_p = \frac{p(\mathbf{x}) - p_\infty}{q_\infty} \quad (4.1)$$

and describes the pressure difference between the local pressure $p(\mathbf{x})$ and the ambient pressure p_∞ normalized by the ambient dynamic pressure q_∞ :

$$q_\infty = \rho_\infty \frac{\|\mathbf{v}_\infty\|_2^2}{2}.$$

From fig. 4.16 its possible to see that the surface solution of the LSQ-ROM meets the reference solution quite well for the high compression rates of 1.0 and 0.8. So, although the deviations in the flow field are relatively high for this case, as seen from fig. 4.15, the pressure distributions are still highly accurate. However, when lowering the number of POD modes, the prediction of the shock location and strength becomes worse. While the solution of the LSQ-ROM coupled with CODA gives even for a compression rate of 0.6 a sufficient shock prediction, the optimized solution from LSQ-ROM with TAU already fails to capture this phenomena. This can be related to the optimization process in the LSQ-ROM not finding a proper minimum for the residual, which is also supported by the behaviour of the objective function values illustrated in fig. 4.17b. It can be seen that the objective function value is increasing by a full order of magnitude when lowering the compression rate from 0.8 to 0.6. This indicates higher errors as the objective function value is the squared summed error of the residuals, which refer to the violation of the conservation laws. Nevertheless, despite the shock, the overall pressure distribution still matches the characteristic of the reference solution. The behaviour of the objective function in fig. 4.17a and fig. 4.17b follows the expected decreasing trend when increasing the number of POD modes for the model. At the same time, the relative prediction time, which is the time needed for LSQ-ROM prediction stage normalized by the time for the CFD solution, shows an increasing trend as the computational effort raises for

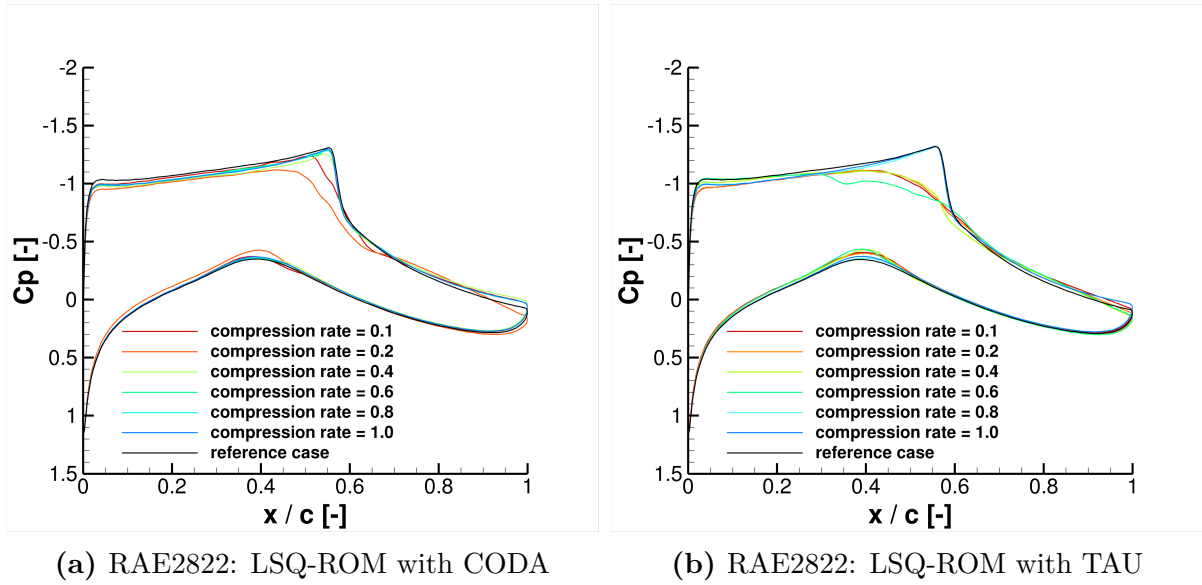


Figure 4.16: RAE2822: Pressure distributions of the LSQ-ROM prediction at $Ma = 0.75$ and $\alpha = 3^\circ$.

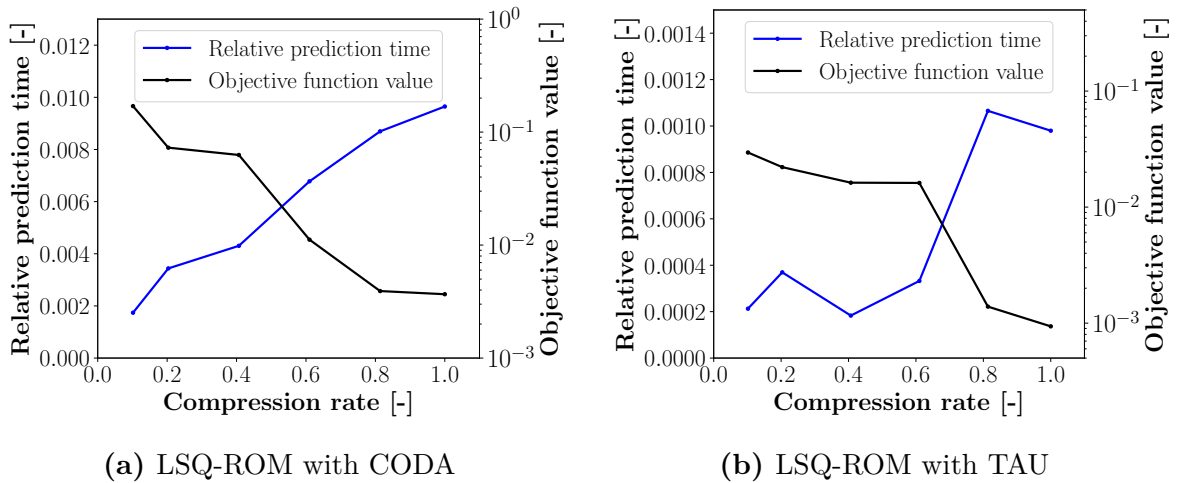


Figure 4.17: RAE2822: Relative prediction time and objective function value over compression rate.

a higher number of POD modes. Comparing the relative prediction time between the models coupled either with CODA or TAU indicates a high difference in the prediction time. However, this is explained by the huge difference in the computational time for the CFD solutions. As anisotropic grids like the one for the RAE2822 rely on implicit solvers for a fast convergence, TAU's explicit solver struggled with convergence and hence, required nine times more time than CODA's implicit solver. So, the absolute prediction times are in good agreement with each other.

4.1.3 Assessment of the validation results

The presented results confirm the functionality of the LSQ-ROM method when coupled with the new CFD solver CODA. The results used for the comparison are obtained with LSQ-ROM connected with TAU, i.e. a framework that was already part of several investigations. So, a proper implementation of LSQ-ROM coupled with CODA is expected to give results that are comparable to the ones from the LSQ-ROM + TAU as long as both CFD solvers yield similar flow solutions. This is tested by comparing several POD modes, which are defined by the snapshots referring to the training points computed by the respective flow solver. The training data in both the subsonic and transonic regime lead to comparable POD bases as it is confirmed through a comparison of several modes and the information content contained in all the modes. Hence, it is possible to conclude that both subspaces contain similar information to approximate solutions with a low dimensional representation of the design space. The presented analysis compares also the accuracy of the LSQ-ROM predictions based on both CODA and TAU solutions and residual evaluations. The subsonic test case exhibits very similar prediction errors for all investigated reduction levels not only in their magnitude but also for their structure within the whole flow domain. For the transonic case, the LSQ-ROM solutions show good agreement for the lower reduction levels (higher compression rate), while the coupling with CODA leads to a better prediction of the shock location and strength for the higher reduction levels in comparison to the results obtained with TAU. The reasons behind this are assumed to be a different behaviour during the optimization process, which relies on slightly different POD modes together with a different residual definition and ends in a different local minimum. Further results for the final objective function value from the optimizer verify that the minimization process is able to find local minima, which improve when including a higher number of POD modes. The analysis of the time required for the prediction of a LSQ-ROM solution is presented for different compression rates. Results show that the prediction time increases, as expected, for higher compression rates and its absolute value is comparable for the LSQ-ROM coupled with both CODA and TAU. Compared to the time of the reference CFD solution, the LSQ-ROM + CODA is still multiple orders of magnitude faster, although this comparison is highly influenced by the chosen solver settings, which is also the reason why the LSQ-ROM + TAU seems to perform better when considering the relative prediction time. All these summarized outcomes prove the correct behaviour of the LSQ-ROM coupled with CODA and establish the foundation for the main investigations that focus on the reimplemented hyperreduction method, which is presented in the next section.

4.2 Investigation of the hyperreduction

The section includes results about the hyperreduction technique and the analyses to determine its influence on the accuracy and acceleration within the LSQ-ROM coupled with CODA. The main goal is to give insights into the behaviour and potential acceleration improvements compared to the former implementation of the hyperreduction. In addition to the two 2D test cases, which are also part of section 4.1, a 3D case is included in the investigations to test whether the 2D findings can be transferred to 3D. For all three test cases, the first part contains a presentation of results that allow an assessment of the prediction quality for different levels of hyperreduction. After that a comparison of the time savings of the new, consistent implementation of the hyperreduction with the former implementation is given and finally an assessment of proper reduction levels for the POD basis and the hyperreduction, which are a compromise between acceleration and accuracy deterioration.

4.2.1 NLR7301: subsonic flow

The upcoming results are based on the same training data used for the validation analysis of the LSQ-ROM + CODA (cf. section 4.1). The test point is again defined by:

$$\text{Mach} = 0.25 \quad \alpha = 6.0^\circ$$

However, in contrast to the validation results that should solely prove comparability of the results of the LSQ-ROM coupled with CODA or TAU, this section tries to assess the absolute prediction quality. Therefore, additional nine test points are randomly chosen in the design space to avoid that the selected test point is exceptionally well suited for a prediction with the LSQ-ROM than other points. To check whether the test point with $\text{Mach} = 0.25 \quad \alpha = 6.0^\circ$ can be used as a representative case for the investigations of the hyperreduction, fig. 4.18 presents two heatmaps that give an overview of the performance of the LSQ-ROM for different prediction points. The predictions rely on a full POD basis, and therefore represent the model without reduction, which should be the baseline for an assessment of further model reductions. In fig. 4.18a the colouring of the test points refers to the objective function value from the optimizer, which is the summed squared residual, and therefore it represents an indicator for the quality of the approximated solution. The figures also include the training points as grey dots so the predictions can be seen in the context of proximity to the sampling points, whose prediction would be nearly exact. In fig. 4.18a the objective function values are normalized by the maximum value of all validation points and the test point. Given this relative description, it is possible to see that the prediction of the solution at the test point with $\text{Mach} = 0.25, \quad \alpha = 6.0^\circ$ leads to an minimized value that ranks among the higher values for the objective function value. And as expected, there is a connection of these values to the overall accuracy of the predicted solution, that is illustrated in fig. 4.18b in form of relative summed squared error (cf. eq. (3.3)) for the total energy field. So, when comparing the relative values for this error measure with the objective function values, a relatively low objective function value corresponds also to a relatively low error in the prediction. Based on this discussion, the test point with $\text{Mach} = 0.25, \quad \alpha = 6.0^\circ$ is among the points with lower accuracy, and therefore the analysis of the related flowfield results is expected to be rather conservative. For this reason, the results presented in the rest of this section refer to the flow condition of $\text{Mach} = 0.25, \quad \alpha = 6.0^\circ$.

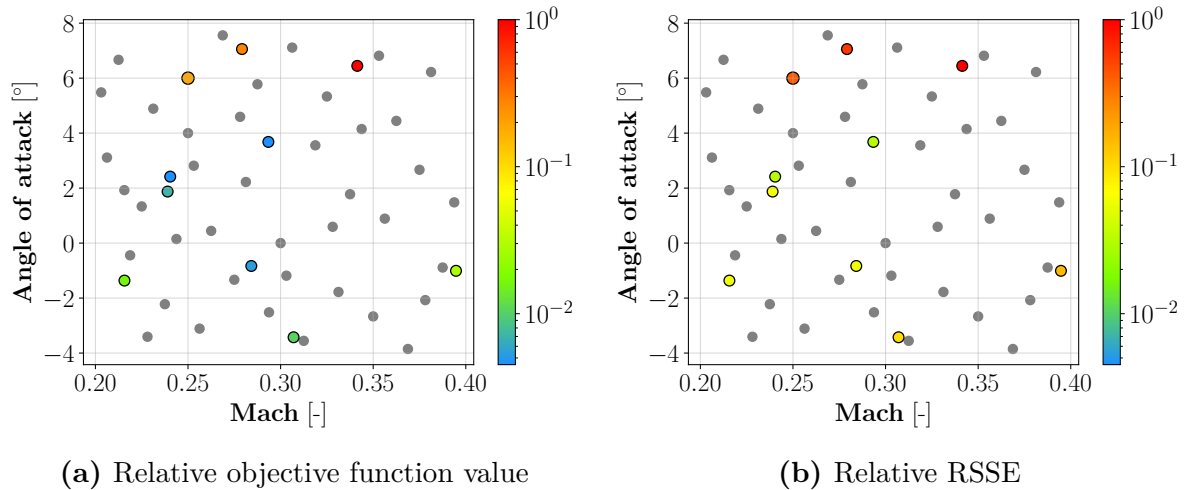


Figure 4.18: NLR7301: Heatmaps for the validation of the test point.

Accuracy for different reduction levels

In section 4.1.1 it is already shown that the LSQ-ROM is able to predict highly accurate subsonic solutions with negligible errors for the whole flow domain. However, the influence of the two reduction methods integrated in the LSQ-ROM is first investigated in this section. Figure 4.19 presents the absolute relative error of the predicted pressure field for four different combinations of compression rate and hyperreduction indices. From (a) and (b), it is possible to see that the error field is only slightly influenced when decreasing the number of hyperreduction indices. The maximum errors are in both cases around 3%, which can be decreased when taking more POD modes into account. Although the compression rate of 0.1 refers to solely 4 POD modes, the maximum errors can be decreased by more than an order of magnitude, which can be seen from (c) and (d). The relative number of hyperreduction indices can again be decreased down to 0.5% without losing too much accuracy. In order to assess the accuracy of multiple combinations of reduction levels for the POD basis and the hyperreduction, the coefficient of determination is utilized as an integral metric to measure the agreement between the predicted flow field and the reference solution from the CFD solver. Its definition is given by eq. (3.2). Figure 4.20 displays the coefficient of determination for the field of the total energy for different compression rates and relative number of indices. The relative number of indices corresponds to the number of hyperreduction indices normalized with the total number of cells (maximum number of indices). So, these two parameters define the levels of reduction for the two available reduction methods of the LSQ-ROM. It is possible to see from fig. 4.20 that the number of selected indices has relatively small influence on the coefficient of determination. Solely for extremely low numbers of indices a significant deterioration of the prediction quality seems to appear. However, this decline is less pronounced than the accuracy degradation caused by a reduction of the POD modes. Lowering the compression rate leads to a clear decrease in the values for the coefficient of determination, which becomes more distinct when decreasing the compression rate. These trends for the total energy field are also present for the other conservative variables (cf. appendix A), and therefore these findings are representative of the overall prediction accuracy. In conclusion, it is possible to state that, for the specific scenario, feasible reduction levels for the LSQ-ROM seem to be a compression rate of 0.1 and a relative number of selected indices of 0.5%. Although the reduction levels are relatively high, the field solution of the absolute error in pressure in fig. 4.19 (d) exhibits low errors. Even the highest errors are only at around 0.1%, which is definitely sufficient for a reduced order model application. At the same time the high reduction levels promise a distinct

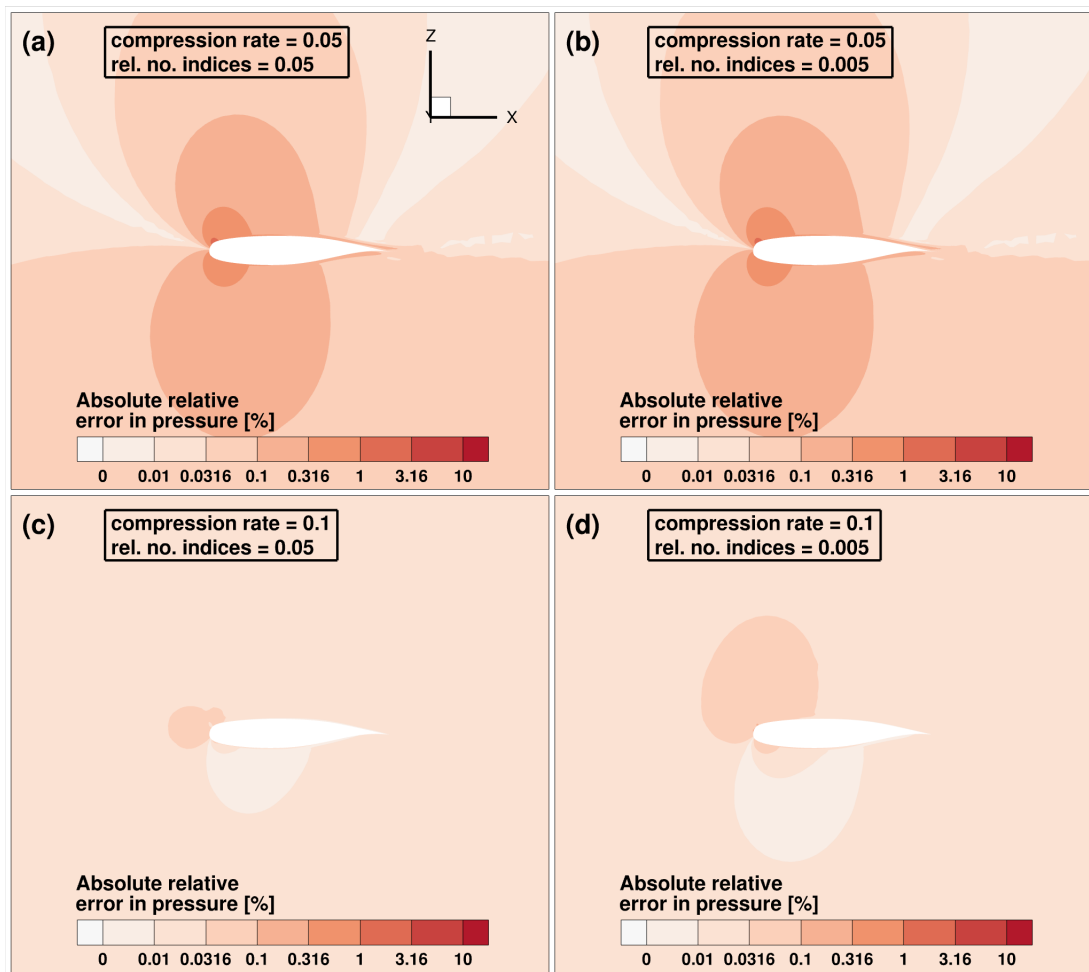


Figure 4.19: NLR701: Absolute relative error in pressure of the test point prediction.

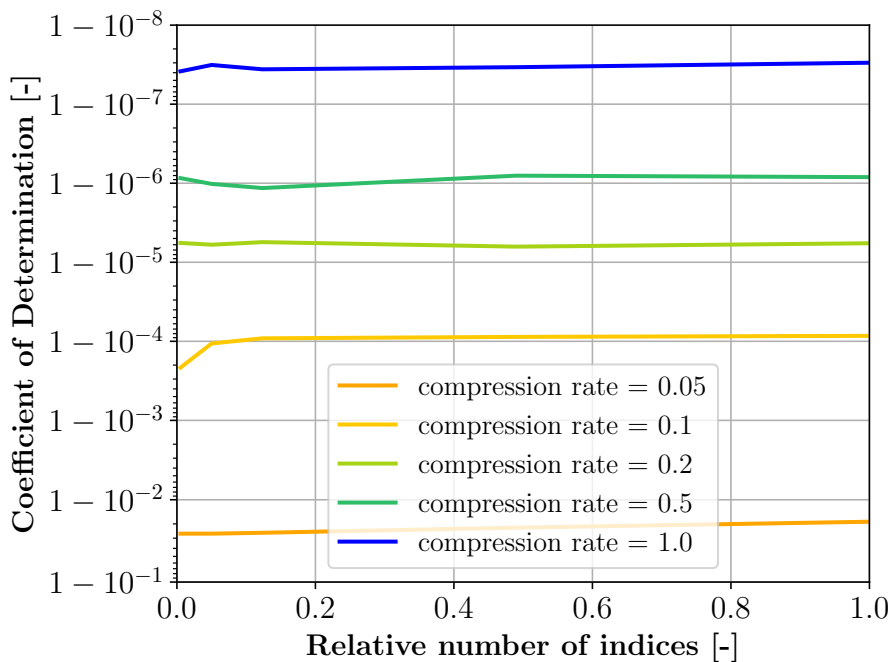


Figure 4.20: NLR7301: Coefficient of determination for different compression rates.

acceleration of the prediction in the online stage.

Comparison with the interpolated solution As described in section 2.6.2, the POD coefficients are estimated in LSQ-ROM by means of the LM line search minimizer using as initial condition the POD coefficient returned by an interpolation method. This refers to a start solution that is based on a POD + Interpolation (POD+I), which can be seen as an alternative ROM to the LSQ-ROM. The comparison of the POD+I model with the LSQ-ROM was already investigated in [62] for extrapolation points and different interpolation models. However, to examine the impact of the residual minimization process on the accuracy, the solutions of the LSQ-ROM from fig. 4.20 are compared to the interpolated solutions. The utilized interpolation method is based on Thin Plate Splines (TPS) (cf. section 2.3). Interpolating the POD coefficients allows to reconstruct a solution the same way as for LSQ-ROM solution. Although this interpolation is independent of the hyperreduction, for an easier comparison the coefficient of determination of the interpolated solutions is presented in fig. 4.21 like in fig. 4.20. Increasing the compression rate above 0.1, does not lead to any significant improvement with respect to coefficient of determination. It is possible to see that the interpolated solutions lead to lower agreement with the reference flow solution than the LSQ-ROM solutions. An exception is the solution for the lowest compression rate of 0.05. For that, the solution with TPS interpolation leads to a higher coefficient of determination than the LSQ-ROM solution. However, this is not the case for all conservative variables, as the coefficients of determination for the momentum are still higher after the optimization. This comparison between interpolated and LSQ-ROM solutions confirms the expected behaviour that the minimization of the residual leads in general to a higher agreement to the reference solution. Nevertheless, it is important to keep in mind that the residual is a nonlinear function of the conservative variables, which means that a reduction in the residual does not automatically refers to a more accurate solution. In fact, the minimization is done on the squared sum of the residual that is a global quantity, and therefore it lacks information of the underlying complex field solution. Together with the observation of a decreased coefficient of determination

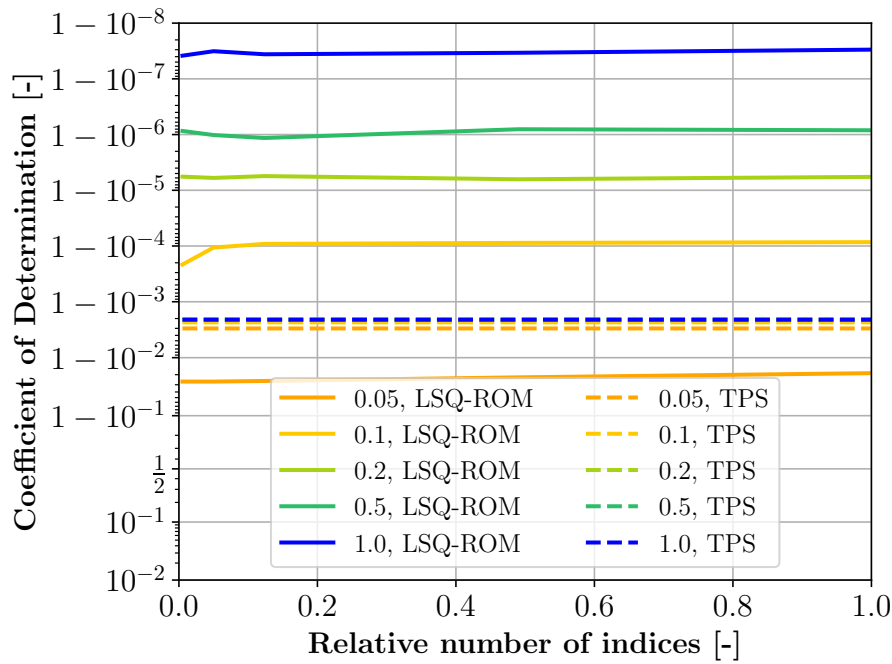


Figure 4.21: NLR7301: Coefficient of determination for solutions of the LSQ-ROM and TPS for different compression rates.

for one of the optimized solution, it can be assumed that a reduced summed residual does not necessarily leads to an "global" improvement for all variables.

Assessment of the submesh-based Hyperreduction in 2D

Before assessing the acceleration potential achieved by the consistent hyperreduction implementation, some remarks concerning their different approaches must be given in order to define ways to fairly compare results based on both methods. While the former implementation can be realized by a simple indexing of the full residual, the consistent implementation utilizes a submesh, which is defined not only by the selected hyperreduction indices but also by the dependency of the residuals on neighbouring cells. Several submeshes are illustrated in fig. 4.22 for different numbers of indices and a model including all POD modes of the subsonic design space. The submesh in fig. 4.22 (a) is based on 100 selected indices (+ five farfield cell indices). The colouring of the cells refers to primarily selected cells (green), direct neighbours (yellow) and neighbours of neighbours (orange). Especially in (b) and (c) it is possible to see that the cells concentrate around the airfoil with the focus on the nose, where relatively strong variations of the flow field can be expected due to the varying location of the stagnation point within the design space. Another important fact is that almost every primarily selected cell for the submesh in (a) leads to twelve additional cells, since the selected cells and their stencil are in most cases disconnected from each other. This is an important consideration for the assessment of the computational effort of the submesh-based hyperreduction as the residuals need to be evaluated for every cell in the submesh, and not only for the cells identified by the hyperreduction indices. Therefore, the computational effort for the residual evaluation scales with the size of the submesh and just indirectly with the number of indices. However, for the former hyperreduction it is sufficient to express the level of reduction in terms of the number of indices as it directly refers to the reduction of the residual (after its evaluation). To allow a fair comparison of both implementations, the number of indices is

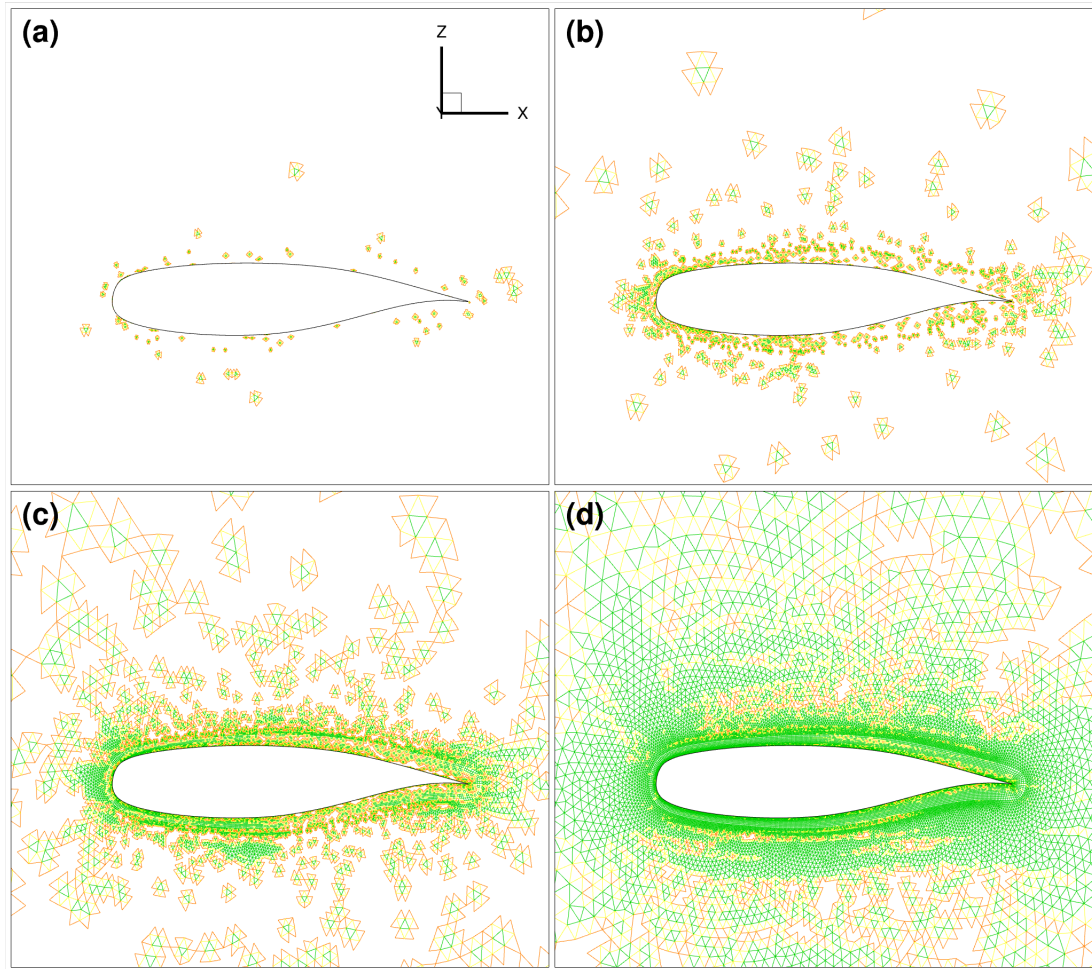


Figure 4.22: NLR7301: Submeshes utilized for the consistent hyperreduction implementation.

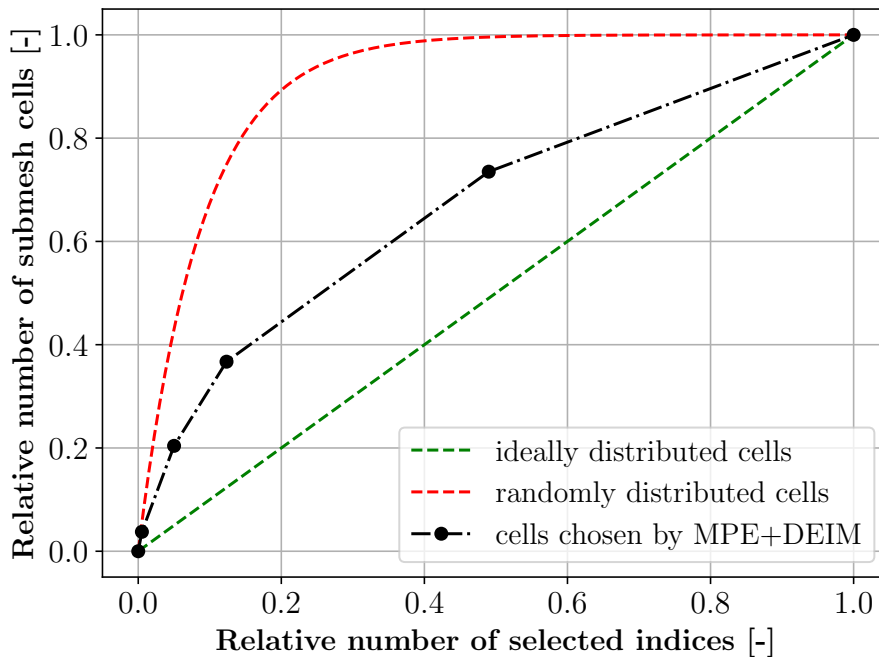


Figure 4.23: 2D: Relative number of submesh cells depending on the relative number of selected indices.

chosen for a measure of the reduction level, although it is not directly a measure for the effort in the new, consistent hyperreduction. As a result, the relation between the number of hyperreduction indices and the resulting submesh size becomes important to assess the acceleration potential and acceleration estimations from section 2.6 for the consistent hyperreduction.

In order to illustrate the connection between the relative number of selected indices and the resulting size of the submesh, fig. 4.23 shows some relations between the two measures for indices selected by the MPE + DEIM, for randomly distributed cells / indices, and the ideal but unrealistic case that all selected cells form a compact cluster. For the small numbers of selected indices the number of cells in the submesh is near the values for the randomly distributed cells. That is expected when looking at the submeshes in fig. 4.22 (a) and (b) where most of the stencils are not connected, therefore resulting in a negligible amount of shared neighbor cells. For increasing numbers of indices this progression deviates from the one for the randomly distributed cells as the indices start to cluster (cf. fig. 4.22 (c) and (d)). The clustering is observed for many applications of the MPE and often undesired but in this case it is important for a better scaling of the residual evaluations on the submesh. Without any clustering, it would take only 20% of all indices to produce a submesh with 90% of the size of the original mesh. Besides the clustering, the relation between indices and submesh is also influenced by the cell types and the dimensionality of the mesh. The 2D mesh around the NLR7301 consists mostly of tetrahedrons, which only imply at maximum nine additional cells, while the additional hexahedrons (cf. section 3.3.1) imply twelve cells. For a three-dimensional case a hexahedron would lead to 25 cells, which would automatically lead to a faster growth of the submesh when increasing the number of indices. However, this issue is investigated in more detail for the 3D case in section 4.2.3.

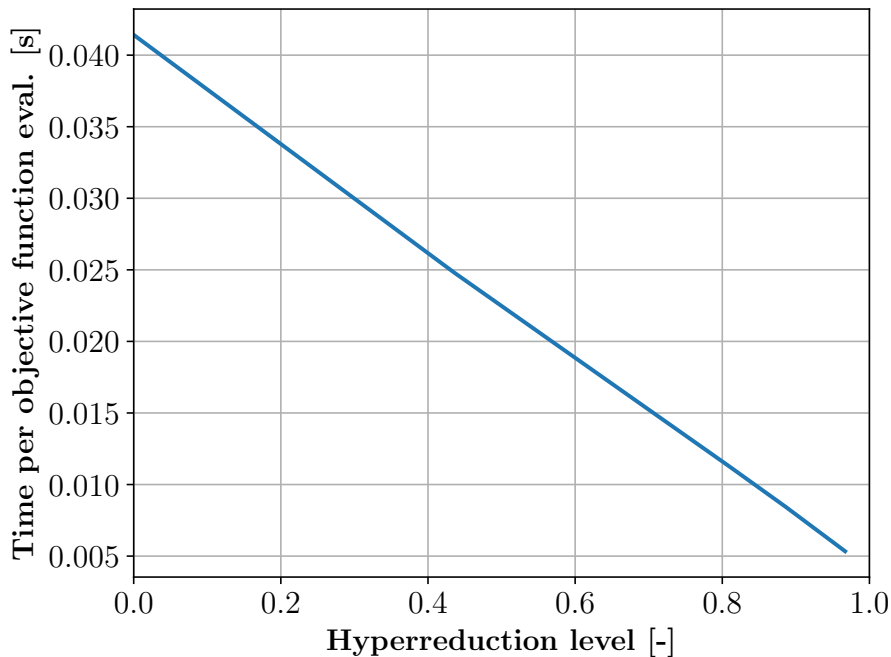


Figure 4.24: NLR7301: Influence of the hyperreduction level on the time for the evaluation of the residual.

Acceleration potential for different reduction levels

Before comparing the acceleration potential of the two hyperreduction implementations at different reduction levels, the estimation for the savings in the computational effort for the evaluation of the submesh residuals from section 2.4 is examined. While the former hyperreduction implementation does not allow a reduction of the residual evaluation, the evaluation on the submesh should be proportional to the number of cells and therefore to the hyperreduction level from eq. (2.99). This aspect is confirmed in fig. 4.24 which shows an almost linear relation between the time spent for the evaluation of the objective function / residual and the hyperreduction level. For the case, that the time for the residual evaluation is plotted directly over the the number of hyperreduction indices, the linear scaling is lost. Instead, the behaviour would be influenced by the relation between the number of indices and the resulting submesh size as it is illustrated in fig. 4.23. Moreover, the same scaling can be expected for the effort for the reconstruction of the state from the POD basis as it also linearly scales with size of the submesh. A similar trend can be seen in fig. 4.25a, which presents the relative prediction time (i.e. the prediction time divided by the time required for the CODA simulation) for different reduction levels for the POD basis and the hyperreduction. It is possible to see, that this trend is present for every compression rate of the model, which confirms not only the effectiveness of the hyperreduction but also the time saving estimations applied to the whole online stage of the LSQ-ROM. So, a prediction without any reduction for the POD basis and without hyperreduction takes around 1% of the time required for the FOM. Taking only 0.5% of all potential residual indices into account allows to reduce the relative prediction down to around 0.1%, which is a reduction of one order of magnitude. Similar reductions in prediction time are present for the models with a decreased number of POD modes. As expected, the reduction of included POD modes also reduces the prediction time, because of the lower costs in the optimizer and for the reconstruction of the states. The minimum relative prediction time can be obtained for the highest reduction levels, here at a compression rate of 0.05 and for a relative number of indices of 0.005. With these

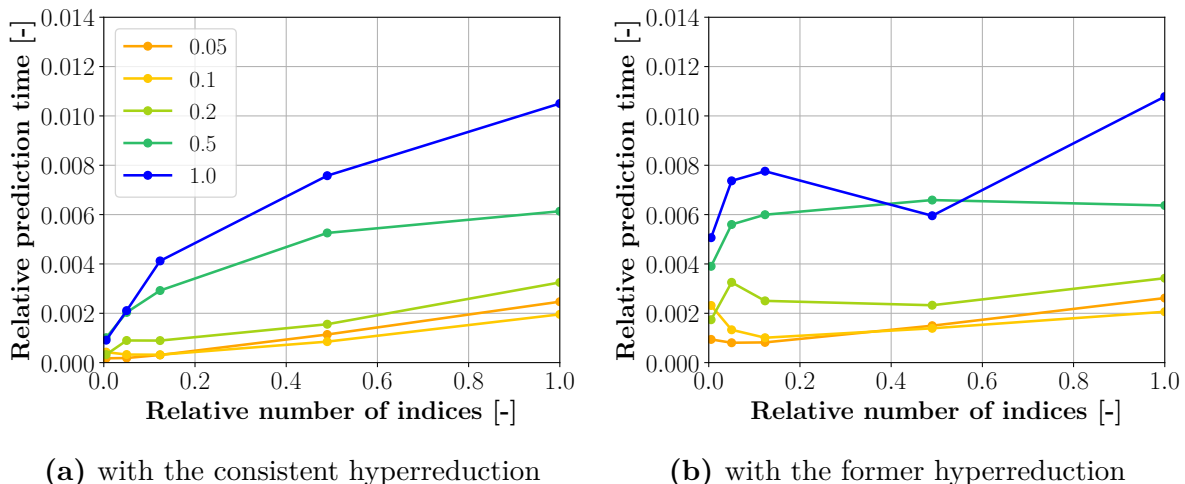


Figure 4.25: NLR7301: Relative prediction times for different compression rates.

settings, the ROM is around 5700 times faster than the FOM. However, the necessary compromise between gain in acceleration and a loss of accuracy is discussed in the end of this section. Despite the convincing results with respect to the acceleration potential with the consistent hyperreduction, a comparison with the former implementation is needed to review the actual gain of the new submesh-based implementation. For that, fig. 4.25b illustrates the relative prediction time for different reduction levels but with the former implementation of the hyperreduction. In this case, there is still a reduction in prediction time, when reducing the relative number of indices. Nevertheless, the time saving is less pronounced than the ones in fig. 4.25a. Reducing the number of indices, leads to a maximum reduction by a factor of 2.5 (compression rate 0.05). So, the fastest model with the former hyperreduction leads to a maximum speed-up factor compared to the FOM of around 1200 in contrast to the factor of 5700 for the consistent hyperreduction.

An even clearer view of the difference is visible, when normalizing the relative prediction time with the number of residual evaluations to remove the dependency on the fluctuating number of iterations in the optimizer. In fig. 4.26 this relative prediction time per objective function evaluation is displayed side by side for both implementations. The normalized relative prediction times in fig. 4.26a almost perfectly agree with the expectations concerning the time saving through reduction of the POD modes and the hyperreduction. It is remarkable, how well the decline for lower numbers of indices corresponds to the behaviour in fig. 4.23, and that an illustration with respect to the hyperreduction level would show the expected linear decline in prediction time. At the same time, fig. 4.26b, outlines a minor time saving potential in the online stage per objective function evaluation when using the former hyperreduction. As the time savings are even less distinct in this representation, it becomes clear that the accelerations seen in fig. 4.25b partially originate from a smaller number of optimizer iterations.

To assess the speed-up that can be obtained from a LSQ-ROM with the consistent hyperreduction compared to the former hyperreduction, fig. 4.27 displays the speed-up factor, which is the prediction time with the former hyperreduction divided by the time required with the consistent hyperreduction. At a relative number of indices of 1.0 both implementations should lead to similar prediction times and thus to a speed-up factor of 1.0, as the residual evaluation is performed for a full mesh even for the consistent implementation. Considering some variations due to statistical changes of the load of the workstation, these values show good agreement. When reducing the number of hyperreduction indices, the speed-up factor increases, except for the model with a compression rate of 1.0 and a relative number of indices of 0.5. This is because of a slightly higher number of iterations

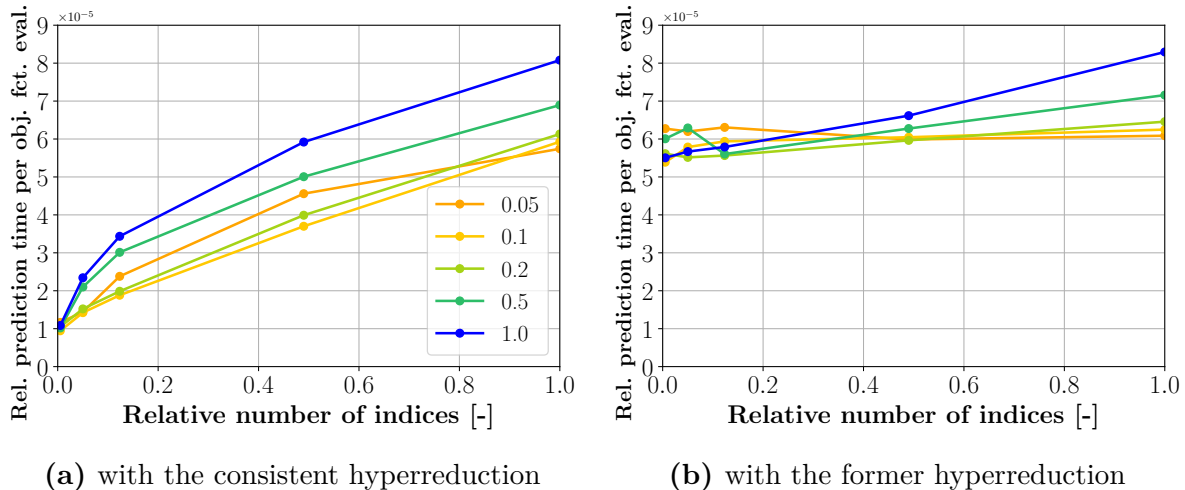


Figure 4.26: NLR7301: Relative prediction times per objective function evaluation for different compression rates.

in the optimizer, due to a relatively high sensitivity of the stopping criteria to numerical errors. The increase of the speed-up factor is more distinct in the region of lower numbers of hyperreduction indices, as the submesh size is reduced more effectively. So, especially for high hyperreduction levels the new consistent hyperreduction becomes superior to the former one with maximum speed-up factors between 3.8 and 5.5.

Assessment of the results

The NLR7301 test case confirms several important expectations for the LSQ-ROM for predictions in the subsonic regime together with promising results for the newly implemented hyperreduction. The prediction of the test case exhibits low errors within the total flow field compared to the reference solution of the FOM. The findings from fig. 4.4 already indicate that the design space can be well approximated by a small number of POD modes. The results for the coefficient of determination in fig. 4.20 support this, although increasing the number of POD modes allows to further decrease the errors of the LSQ-ROM. An application of the hyperreduction has only small impact on the overall accuracy of the prediction. Solely very high levels of reduction show a significant accuracy deterioration. From these results, it is possible to conclude that, choosing proper reduction levels allows to obtain a model that promises a high acceleration while still yielding a prediction with sufficient accuracy.

Additionally, the optimized LSQ-ROM solution is compared to the interpolated solution (POD+TPS) used as initial condition of the LSQ-ROM optimizer. This comparison shows that the minimization of the summed residual can further increase the accuracy of the interpolated solution. Although the minimization process acts solely on the sum of all squared residual entries, which automatically focuses on the minimization of the highest entries, it still manages to increase the overall agreement of the predicted solution with the CFD solution. However, as it is shown in some cases, such a residual minimization does not necessarily lead to a better overall accuracy. This happens because the minimization of a global quantity can possibly reduce a small number of relatively high residuals while increasing all other residuals. Moreover, the residual is even in the subsonic regime not linearly dependent on the state variables. So, it can not be expected that a reduction of a residual is directly proportional the accuracy of the state variables in the same cell.

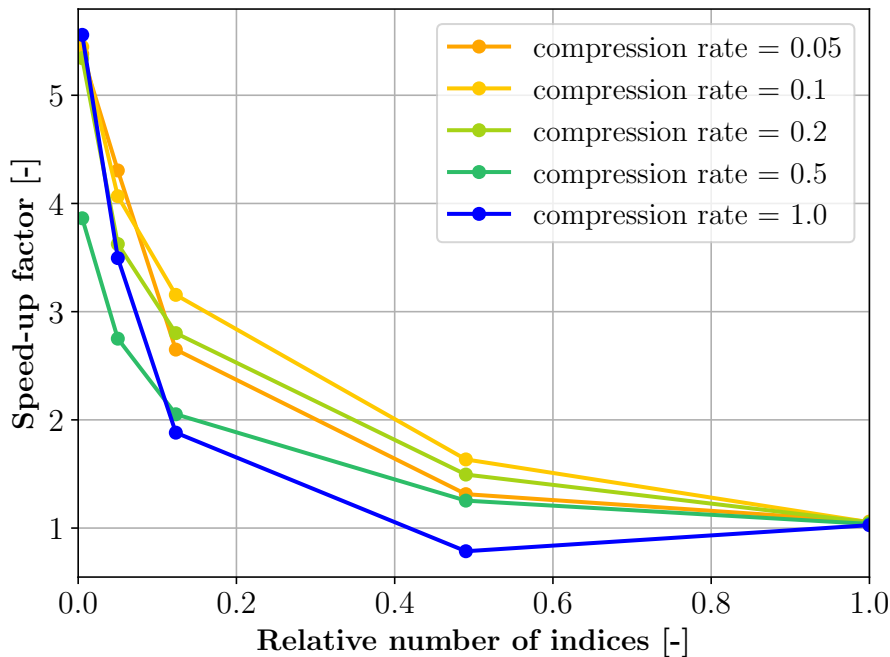


Figure 4.27: NLR7301: Speed-up factor for the consistent hyperreduction.

The investigation of the submeshes for the new hyperreduction implementation illustrates the connection between the number of hyperreduction indices and the resulting size of the submesh. The frequently observed clustering of the indices selected by the MPE leads to a smaller growth of the submeshes. So, the illustrated case of randomly distributed cell indices is flattened, which leads to smaller submeshes for the real sets of indices, and therefore to higher potential time savings. These time savings are clearly visible, when applying the submesh-based hyperreduction. The highest reduction levels allow accelerations that reduce the time spend in the online stage by an order of magnitude. At the same time, the estimations from section 2.4 for the reduction in effort can be confirmed by the time spent for the residual evaluations and even in the overall prediction time. Most importantly, the former implementation is clearly outperformed by the consistent hyperreduction. The reduction of the prediction time is also present for the former implementation, but particularly for the high reduction levels the gap between both implementations becomes clear. The LSQ-ROM with the consistent hyperreduction reaches speed-up factors of around 5700 compared to the FOM, whereas the former hyperreduction reaches only a factor of 1200. So, given the option to use a model with a high hyperreduction level, the new hyperreduction implementation can accelerate the whole online stage of the LSQ-ROM by a factor up to 5.5 compared to the former hyperreduction for this specific test case.

Recommendation for proper reduction levels Some suited reduction levels are already given in section 4.2.1. The proposed POD compression rate of 0.1 and a restriction of the residual to around 0.5% of all entries leads to a model that predicts the pressure field with relative errors of at most 0.1%. It is assumed that these relatively high reduction levels are restricted to flow regimes that do not include strong nonlinear flow phenomena like compression shocks or separation bubbles. As POD based ROMs are known to struggle with the approximation of such nonlinearities [17], which are depending in a nonlinear way on the input parameter, the recommendations are not universal. It is expected that a design space including such nonlinear phenomena requires a higher density of training points, and an effective and accurate ROM can not be obtained using the same reduction

level of this test case. Nevertheless, for design spaces similar to the presented one, the proposed reduction levels should be reasonable in terms of accuracy. The time savings resulting from the reduction can be read from fig. 4.25a. While the model without any reduction takes around 1% of the time for the FOM, the model reduction leads to a relative prediction time of only 0.0004, which is a speed-up of more than 2300 compared to the FOM. The LSQ-ROM with the former hyperreduction implementation instead has a speed-up of around 430, emphasizing the advantage of the consistent hyperreduction. For high levels of hyperreduction, the consistent implementation can easily outperform the former implementation by factors of 5.

4.2.2 RAE2822: transonic flow

The transonic test case is based on the design space and training data already presented in section 4.1.2 for the validation of the LSQ-ROM coupled with CODA. In contrast to the subsonic test case for the NLR7301, the design space for the LSQ-ROM for the RAE2822 covers the transonic flow regime. The distinction between these two regimes is important, because with the occurrence of supersonic flow speeds, more complex flow phenomena arise that often times need additional treatments for their computation. In order to account for this increased complexity in the physics that has to be represented by the model, the design space is covered with a higher number of training points compared to the one for the subsonic flow. The test point for the prediction with the LSQ-ROM is defined as in section 4.1.2 by:

$$\text{Mach} = 0.75 \quad \alpha = 3.0^\circ$$

Similarly to the subsonic test case, additional validation points are computed within the design space. These validation points are presented in fig. 4.28 in form of two heatmaps that illustrate the objective function value and the square root of the summed squared difference between the predicted field of the total energy and the reference field solution as defined in eq. (3.3). Both measures are divided by the maximum value of the validation points or the test point and correspond to a LSQ-ROM based on the full POD basis. From fig. 4.28a, it is possible to see that the test point exhibits the largest minimized objective function value. Figure 4.28b displays the integral error measure for the total energy field and shows that the performance of the different points with respect to the residual does not automatically corresponds to this error measure. Points that lead to high values for the residual exhibit in some cases smaller errors than points with smaller objective function values. So, the test point has the highest value for the minimized residual but not for the error. Although there still seems to be a connection between the two measures, the differences are more distinct than in the subsonic case (cf. fig. 4.18). Possible reasons for that are discussed later in this section. Nevertheless, the test point is assumed to be suited to be a representative point for the predictions in the design space and for further investigations.

Accuracy for different reduction levels

Section 4.1.2 presented some general results concerning the accuracy of the LSQ-ROM for predictions in the transonic flow regime. An important outcome is that the prediction of transonic flow fields becomes more complex and less accurate in the presence of nonlinearities like shocks, therefore requiring more POD modes to capture the flow field including a compression shock. Despite the increase of the relative errors in the flow field, the models with an adequately chosen reduction level for the POD basis are still able to properly predict quantities like the surface pressure distribution of the airfoil. Based on

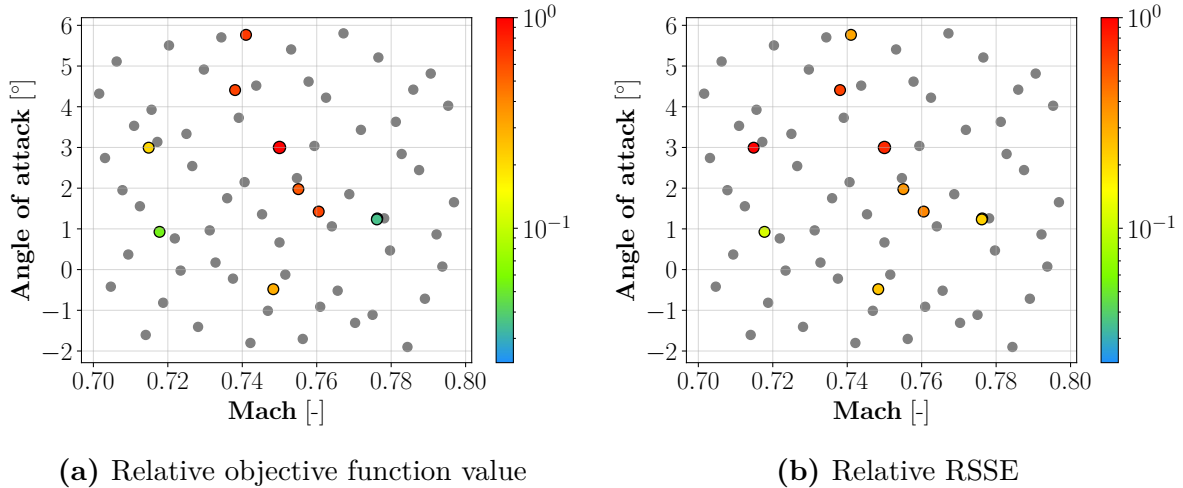


Figure 4.28: RAE2822: Heatmaps for the validation of the test point.

these results, the second reduction method, the hyperreduction, and its influence on the accuracy of the LSQ-ROM is investigated in this section. Before assessing the accuracy with respect to global measures like the coefficient of determination (cf. eq. (3.2)), some selected field solutions of the relative error between the LSQ-ROM solution and the FOM solution are presented in fig. 4.29.

From fig. 4.29 (a), it is possible to see that even a LSQ-ROM without any reduction of the POD basis or hyperreduction predicts the pressure field with errors up to 20.%. These high error values are concentrated around the shock location (cf. fig. 4.13 (left)), indicating that the relatively high errors result from the difficulties to capture the compression shock at the suction side of the airfoil. Nevertheless, the high errors do not span the full shock front, as the pressure very close to the airfoil surface is still in good agreement with the reference solution. The same can be observed for the rest of the pressure field, which has relative low errors around 1.0%. A reduction of the residual vector via hyperreduction only leads to minor changes for the pressure errors, as it can be seen from fig. 4.29 (b). Although the number of considered residual entries is reduced by 75%, this reduction level does not induces further deviations from the FOM. On the contrary, a reduction of the POD basis to a compression rate of 0.4 increases mainly the maximum errors around the shock. Without any hyperreduction, the relative errors increase up to around 24%, while the application of the hyperreduction further increases the error up to 30%. In both cases, the higher errors around the shock location reach closer to the airfoil surface. However, the rest of the pressure field is hardly influenced. When comparing the influence of the hyperreduction on the accuracy for the two different POD bases, the loss in accuracy is more distinct for the reduced POD basis.

The interaction between reduction of the POD basis and hyperreduction is examined in more detail in fig. 4.30a and fig. 4.30b, which display the surface pressure for the predicted solutions at different levels of hyperreduction. These pressure distributions are illustrated by the pressure coefficient C_p (cf. eq. (4.1)) along the airfoil surface. Figure 4.30a refers to a model with a full POD basis. The comparison of the various pressure distributions illustrates a good agreement with the reference solution of the FOM. As the illustrations from fig. 4.29 already suggested, the accuracy of the prediction is only slightly influenced by the extend of the hyperreduction. Even for the highest level of hyperreduction (only 0.5% of all residual entries included), the model is able to sufficiently reproduce the reference pressure distribution. It is remarkable that the highest deviations for all predicted pressure distributions are not located at the shock location between 55% and 60% of the chord length, but mostly at the suction side apart from the shock location.

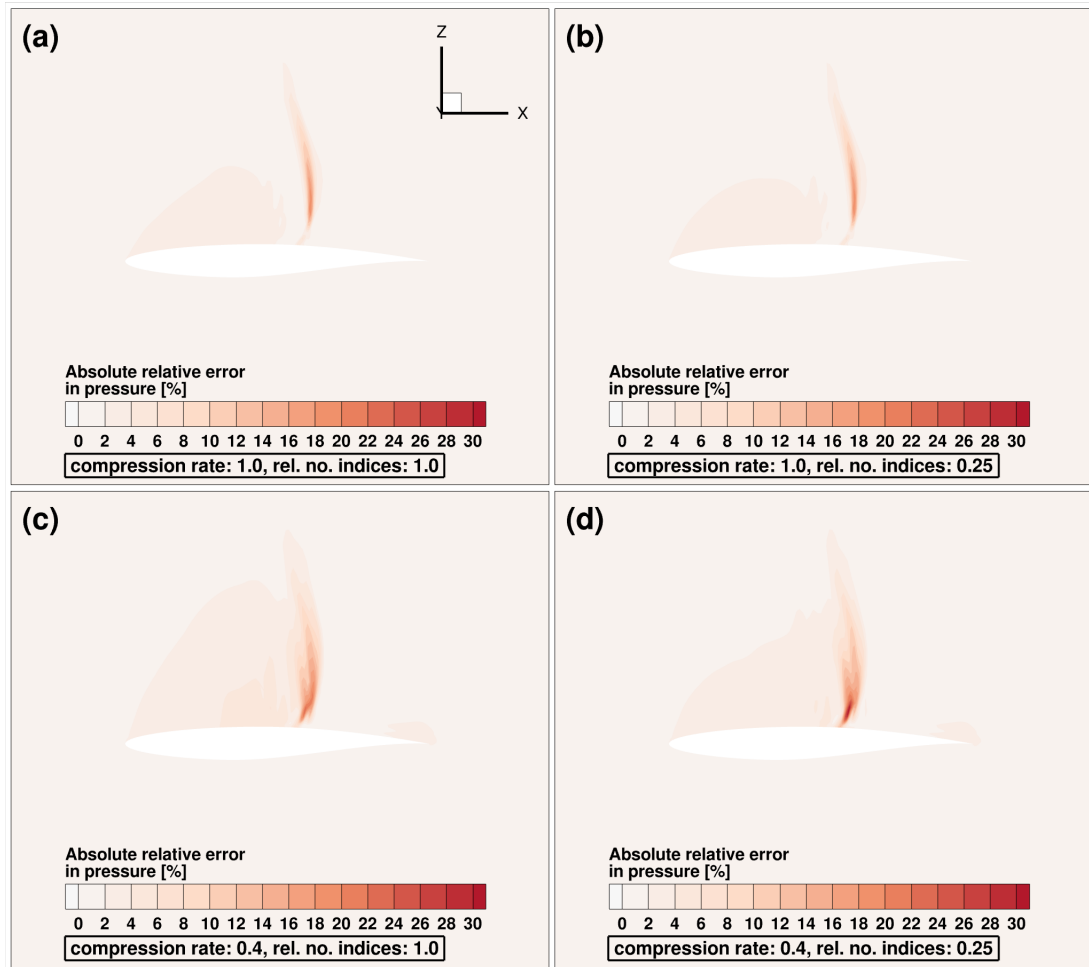


Figure 4.29: RAE2822: Absolute relative error in pressure for predictions with different reduction levels.

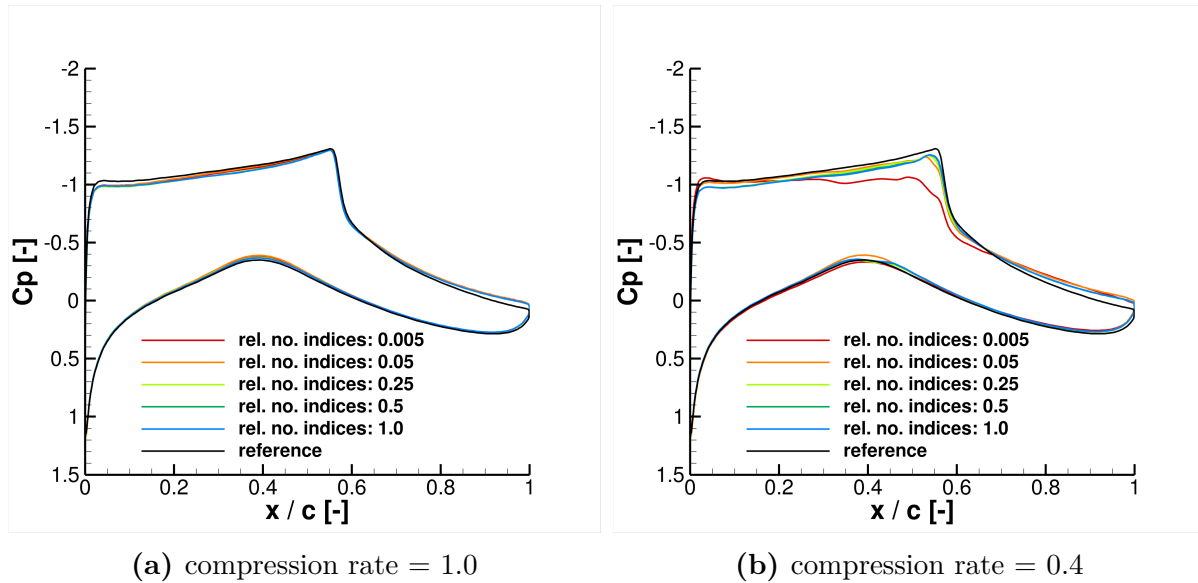


Figure 4.30: RAE2822: Pressure distributions for different reduction levels of the LSQ-ROM.

In fact, the shock is captured almost perfectly, while the rest of pressure distribution contains some offsets from the reference solution.

Figure 4.30b instead displays the pressure distributions for LSQ-ROMs with a compression rate of 0.4. Besides the model with the lowest level of hyperreduction, all the models lead to comparable pressure distributions. The main deviations can be found at the suction side in front of the shock, where the pressure is constantly overestimated, while the pressure at the rear is underestimated. The shock location is in good agreement with the reference solution, whereas the pressure minimum in front of the shock is predicted too high. The overall deviations are larger than for the models with the full POD basis (cf. fig. 4.30a) but exhibit a similar behaviour. The sensitivity of the accuracy follows the findings from the relative errors in the pressure fields, as the hyperreduction seems to influence the pressure distributions more distinct for models with a POD basis with a lower compression rate.

In order to obtain a broader view on the influence of the two reduction methods on the model accuracy, the coefficient of determination is utilized as an integral measure of the agreement of the predicted solution with the FOM. In fig. 4.31a, the coefficient of determination for the field of the total energy is illustrated for various combinations of reduction levels, while fig. 4.31b illustrates this coefficient for the surface pressure. From fig. 4.31a, it is possible to see that the influence of both reduction methods drastically increases when the reduction level is further increased from a certain critical level. So, a reduction of the POD basis down to a compression rate of 0.6 has only a minor influence on the coefficient of determination, at least for relatively high number of relative indices. But when decreasing the compression rate to 0.4, the coefficient of determination drops from around 0.998 to 0.993. This behaviour can also be observed for the coefficient of determination for the surface pressure in fig. 4.31b and other mean flow variables (cf. appendix B). The level of the hyperreduction has a similar influence on this error measure as it stays nearly constant for the low levels of hyperreduction (high relative number of indices), but significantly decreases, when exceeding a certain level for the hyperreduction. This trend seems to be more pronounced for models, which act on a reduced POD basis. Therefore, the models with a compression rate of 1.0 and 0.8 can rely on a low number of residual indices without losing to much of the agreement with the FOM. The models with

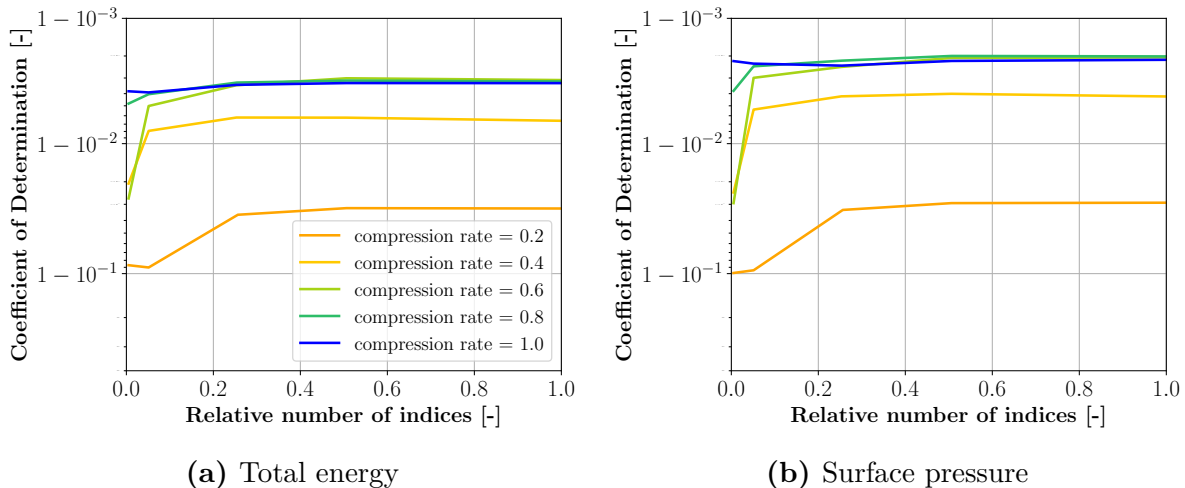


Figure 4.31: RAE2822: Coefficient of determination for the total energy field and the surface pressure for different compression rates.

compression rates below 0.8 face a distinct decrease for the coefficient of determination, when increasing the level of the hyperreduction too far (0.5% of all indices). These findings can be directly transferred to the coefficient of determination for the surface pressure. Moreover, the higher sensitivity of models with a reduced POD basis for the application of the hyperreduction is in line with the results from fig. 4.29 and fig. 4.30b. A possible reason for this behaviour is the influence of the POD basis on the set of selected indices, so that the missing information in a reduced POD basis lead to less favourable indices for the representation of the full residual vector. Figure 4.32 illustrates the indices / cells that are picked by the MPE + DEIM for a POD basis with a compression rate of 1.0 (a) and 0.4 (b) and the number of indices set to 5000. These reduction levels refer to the models leading to the errors shown in (b) and (d) from fig. 4.29. It can be seen that the set of indices cover similar locations around the airfoil like the nose, the rear suction side and the wake. However, there are clear differences e.g. at the rear pressure side, which is hardly covered for the reduced POD basis in contrast to the set of indices for the full POD basis. Instead, more indices are selected at the nose and in the wake of the airfoil. So, it becomes clear that the reduction of the POD basis has a distinct impact on the chosen set of residual indices and therefore directly influencing the performance of the hyperreduction.

Comparison with the interpolated solution To investigate the influence of the residual minimization on the model accuracy, the previous results are compared to the purely interpolated solution (POD+I), which serves as initial condition of the optimization process. The interpolation of the POD coefficients is based on the TPS interpolation method as stated in section 3.3.2, which are then used to reconstruct the interpolated solution. Both the LSQ-ROM and the TPS interpolation are compared in fig. 4.33 with respect to the coefficient of determination for the total energy field. As the interpolated solutions are only depending on the POD basis, they are constituted by the straight lines. The solutions for the compression rates of 0.4 to 1.0 exhibit similar values for the coefficient of determination, so their lines lay beneath the dark blue one. Comparing the results from the TPS interpolation with the ones from the LSQ-ROM leads to the unexpected conclusion that the interpolated solutions lead to higher agreement with the FOM with respect to the coefficient of determination than the LSQ-ROM solutions. This outcome is somehow surprising, as the optimization in the LSQ-ROM seems to decrease the agreement with the FOM, although the overall residual is decreased during this process. In

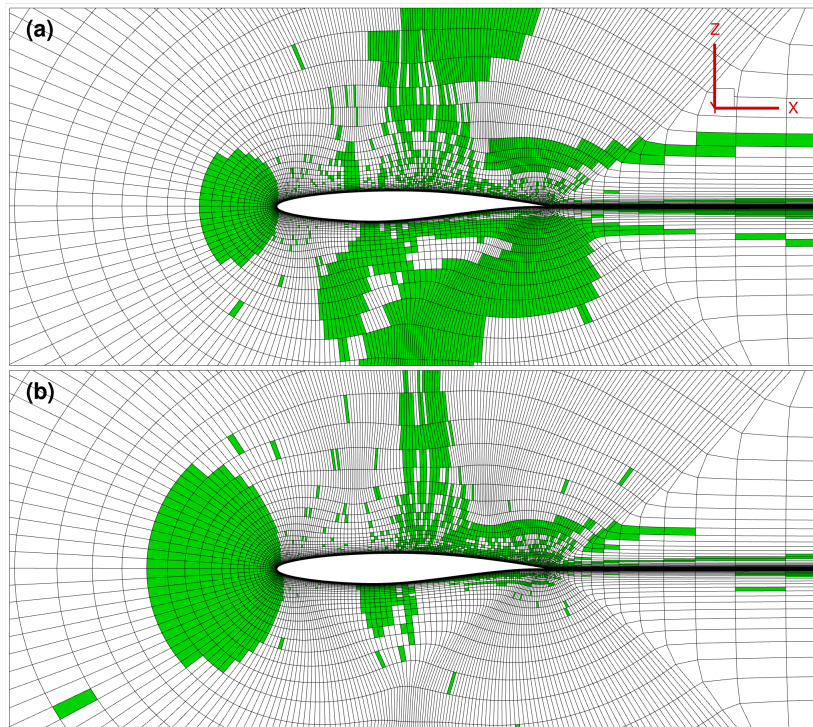


Figure 4.32: RAE2822: Visualization of the cells referring to sets of hyperreduction indices.

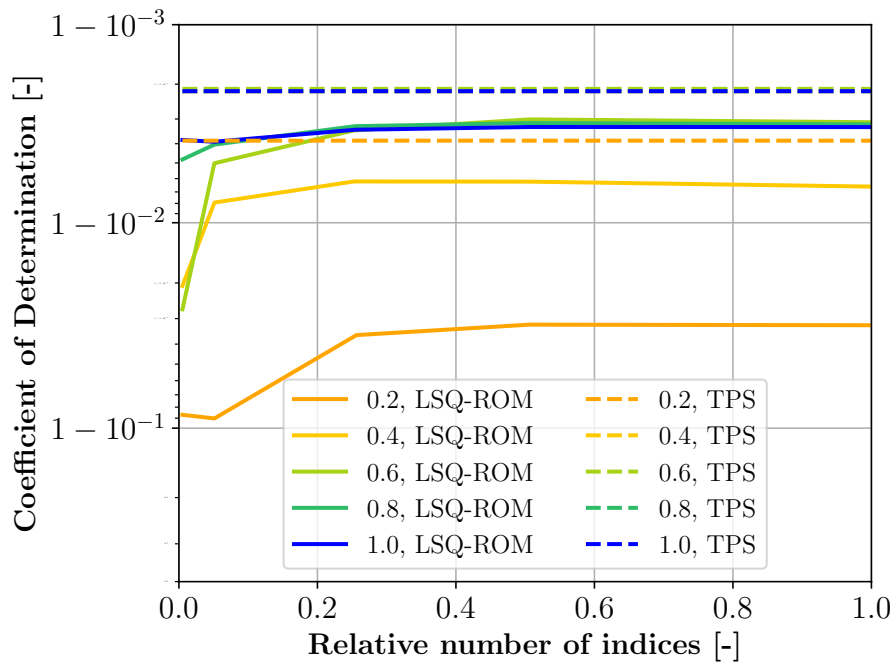


Figure 4.33: RAE2822: Coefficient of determination for solutions of the LSQ-ROM and POD + TPS for different compression rates.

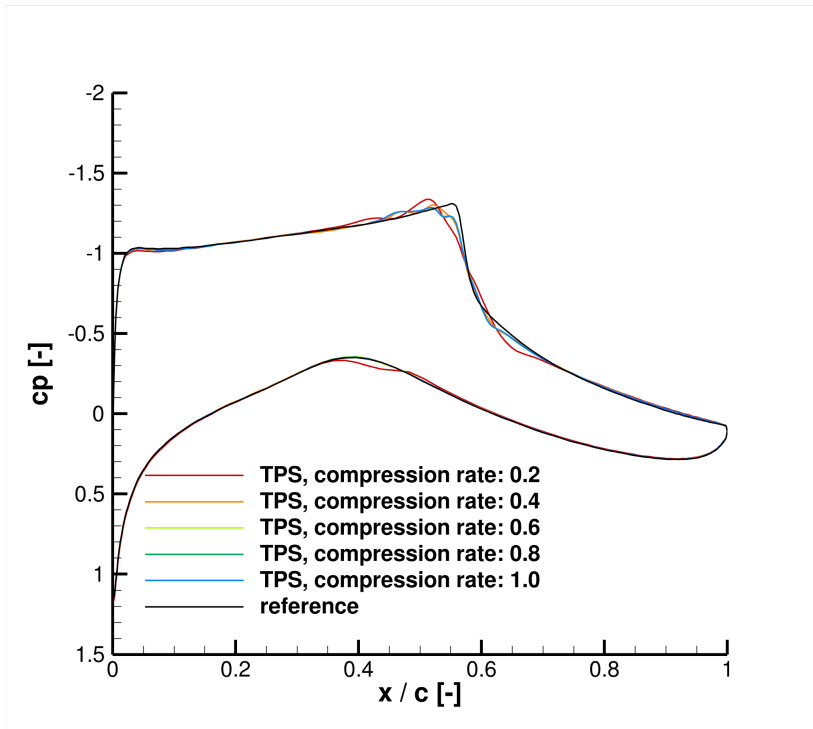


Figure 4.34: RAE2822: Pressure distributions from POD + TPS.

order to find an explanation for this behaviour, the pressure distributions from the predicted solution with the TPS interpolation are shown in fig. 4.34 for varying compression rates. It can be seen that the interpolated solutions lead to pressure distributions almost perfectly following the reference solution. Beside the solution referring to a compression rate of 0.2, the pressure distributions show only minor differences. The superior agreement with the FOM compared to the approximations with LSQ-ROM e.g. in fig. 4.30a is present for almost all parts of the surface. However, the pressure around and at the shock is predicted more precisely by the LSQ-ROM. The TPS interpolation instead leads to some overshoots before and undershoots behind the actual shock location. Even the interpolation with the full POD basis (blue line) is not able to accurately predict the steep gradients induced by the compression shock. Summarizing, the residual minimization in the LSQ-ROM leads to a better agreement with the FOM for the solution around the shock, while it is simultaneously worsened for the rest of the solution. This connection becomes clearer, when examining the residual field of the interpolated (start) solution and the one from the optimized solution. In fig. 4.35 the residuals for the total energy are displayed for the TPS interpolated solution (a and c) and the LSQ-ROM solution (b and d) without any model reduction. The residuals are divided by the square root of the cell volumes, which refers to their usage in the optimizer. This formulation is differing from the way the residuals are treated in the CFD solver, where the residuals are normalized exactly by the cell volume. From the residual fields of the interpolated solution, it is possible to see that the highest residuals are located at the rear suction side around the compression shock. So, the highest residuals are referring to highest deviations of the interpolated solution to the FOM, which are located around the shock. This is implied by the outlined pressure distributions from fig. 4.34, but it is also confirmed when comparing the relative errors of the total energy field of the POD+TPS (a) and LSQ-ROM (b) solution (cf. fig. B.5). After the optimization (cf. fig. 4.35 (d)), the residuals around the shock are mostly reduced by at least one order of magnitude, while the residuals e.g. at the leading suction side and at the farfield boundary (cf. fig. 4.35 (b)) increased. These increased residuals seem to be connected to the offsets of the pressure distributions in

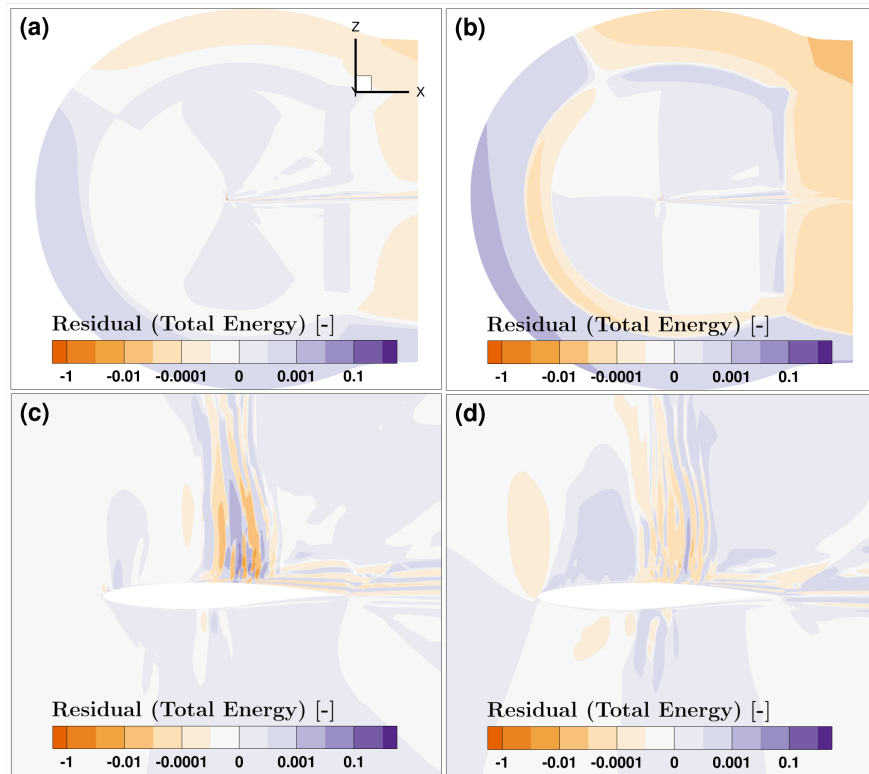


Figure 4.35: RAE2822: Residual fields for the POD + TPS and LSQ-ROM predictions.

fig. 4.30a, whereas the residual reduction around the shock may refer to the improved agreement with the reference solution around the shock, both for the surface pressure and the field solution of the total energy. Nevertheless, it is important to keep in mind that low residuals in individual cells are not automatically equal to low errors in these cells, because the residuals are influenced in a complex way by their neighborhood and can not be assessed independently from each other. The comparison of the residual fields and their corresponding deviations from the reference solution clarifies the surprising results for the coefficient of determination. The residual minimization process aims on reducing the sum of all squared residual. This does not only imply an optimization of a global quantity for the prediction of a whole flow field but also the (over-) weighting of the largest residuals due to their squaring. So, in case of strong nonlinearities like the compression shock (and the flow separation behind it) even relatively small errors in the solution can cause residuals that dominate the sum over all squared residuals. As a result, the minimization process can achieve a reduction of the objective function by focusing these high but localized residuals, although the majority of all residuals is negatively influenced by this. So the intrusive character of the LSQ-ROM and its connection to the FOM ensures, that important flow features are still in correspondence with the underlying physics, although the costs for this may be a loss in accuracy for the remaining domain. Similar findings concerning the prediction quality of the LSQ-ROM compared to the POD+TPS can also be found in [2], where the TPS interpolation outperformed the LSQ-ROM for the given test cases. However, investigations in [62] were able to show that the LSQ-ROM becomes beneficial for predictions points that lie outside of the design space and are therefore hard to predict with interpolation models. In these cases, the incorporation of the underlying physics ensures the proximity to the correct solution.

Extrapolation capability of the LSQ-ROM and TPS interpolation To confirm the superior extrapolation capabilities of the LSQ-ROM, an additional test point is chosen

outside of the design space, which is still defined by

$$\text{Mach} = [0.7, 0.8] \quad \text{angle of attack} = [-2^\circ, 6^\circ]$$

and illustrated in fig. 4.9. The extrapolation point is specified by

$$\text{Mach} = 0.75, \quad \text{angle of attack} = 6.5^\circ.$$

Its prediction shall be briefly examined on the basis of the pressure distributions for two LSQ-ROM and two POD+TPS models with different compression rates. They are illustrated in fig. 4.36 along the reference solution from the FOM. It can be seen that the reference solution features again a compression shock, which is located at the suction side at around 35% of the chord length. The LSQ-ROM predicts the overall pressure distributions with relatively small errors at the pressure side and at the leading suction side. The main deviations can be found for the prediction of the shock location that has an offset of around 5% of the chord length. Further but smaller deviations underestimate the pressure coefficient at the rear. The accuracy of this prediction is persuasive considering that the shock location is very sensitive and strongly influenced by the angle of attack, whose value is not part of the design space. As the shock location is moving to the leading edge with increasing angle of attack, the POD basis lacks the modes that could capture such a flow phenomena at the correct location. With respect to this, the prediction quality of the LSQ-ROM is impressive, especially when comparing it to the solution from the POD+TPS model. The interpolated solutions exhibit once again undershoots of the pressure coefficient behind the shock, whose magnitude is more distinct than for the previous test point. Similar to the LSQ-ROM predictions and probably also resulting from the missing information in the POD basis, the shock location is displaced. Moreover, the pressure distributions contain offsets at the leading suction and pressure side. So, the interpolated solution is clearly outperformed by the LSQ-ROM through incorporation of the residual. Although these finding can not be generalized, they are in line with the outcomes from [62] and confirm the advantage of the intrusive character of the LSQ-ROM for extrapolation cases.

Acceleration potential for different reduction levels

Although the transonic test case leads to different results for the prediction accuracy than the subsonic case, the general steps of the algorithms are not affected. Therefore, the results with respect to the acceleration capabilities of the submesh-based hyperreduction are not discussed in detail like it is done in section 4.2.1. For example, the connection of the number of selected indices to the submesh size is only slightly influenced by the different mesh, which now solely contains hexahedrons, and the different sets of indices, which are based on differing POD modes but still tend to form clusters (cf. fig. 4.32). The resulting flattening growth rate of the submesh can also be found for the RAE2822 test case. Its influence can be seen in fig. 4.37a, which displays the relative prediction time for different levels of reduction of the POD basis and hyperreduction. For that, the time spent in the online stage of the LSQ-ROM is divided by the time required by the CFD simulation. For every line corresponding to a specific compression rate, a distinct reduction of the prediction time can be observed, when reducing the relative number of indices for the evaluation of the residual. This reduction is enhanced when fewer hyperreduction indices are utilized. This trend is a bit blurred by the influence of the total number of optimizer iterations on the prediction time. Comparing the relative prediction of the models with no hyperreduction (relative number of indices = 1.0) to the ones with the highest level of hyperreduction (relative number of indices = 0.005) shows speed-up factors of 4.0 up to 18.0 solely due to hyperreduction. On average, the speed-up is similar to the one observed for the subsonic case (cf. section 4.2.1). The reduction of the POD basis also accelerates the online stage. However, the compression rates have

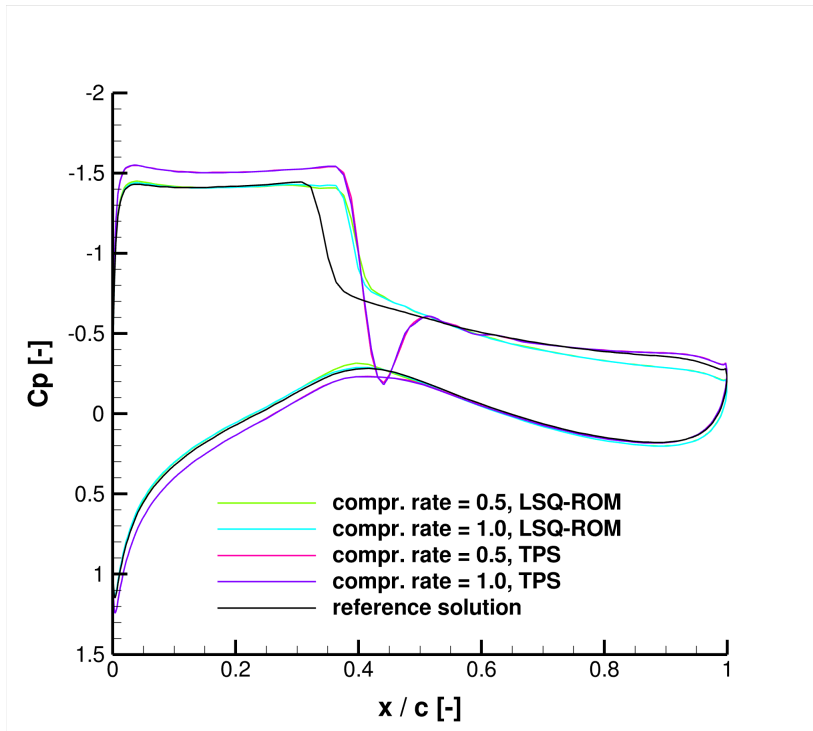


Figure 4.36: RAE2822: Pressure distributions from the LSQ-ROM and POD + TPS for an extrapolation point.

to be kept higher than for the subsonic case, so this kind of reduction has less effect on the prediction time. When utilizing the unreduced LSQ-ROM, it is possible to predict a solution in roughly 1% of the time needed for the FOM. Through the influence of the described reduction models, the gap to the FOM can be further increased up to speed-up factors of around 4000 for the model with a compression rate of 0.4 and a relative number of residual indices of 0.005.

The presented results can be compared to the results of the former hyperreduction implementation in fig. 4.37b, in order to assess the benefit of the new and consistent hyperreduction implemented for this thesis. The corresponding results also show a trend towards a reduction in the prediction time for both reduction methods. Nevertheless, the acceleration by the former implementation of the hyperreduction is lower. Comparing the prediction times for the models utilizing the full residual with the models relying solely on a small subset of 0.5% of all entries, reveals speed-ups of 1.3 to around 4.0. These accelerations are expectedly smaller than for the models relying on the consistent hyperreduction. Especially for the relatively low number of included indices, the residual evaluation on a submesh leads to distinct time savings compared to the former hyperreduction. So, the fastest LSQ-ROM model becomes 800 times faster than the FOM, while with the consistent hyperreduction the fastest model is 4000 times faster. The speed-up factors between both hyperreduction implementations show similar behaviour and values like the ones from the subsonic case (cf. fig. 4.27); their depiction can be found in appendix B. These results confirm the findings from the subsonic case, which is reasonable as the LSQ-ROM algorithm and its underlying hyperreduction do not change its fundamental behaviour for the transonic case.

It is interesting to see the behaviour of the prediction time with the consistent hyperreduction, when normalizing it by the number of objective function evaluations and plotting it over the hyperreduction level instead of the relative number of indices. By this, the influence of the number of iterations and the influence of the first jacobian computation within

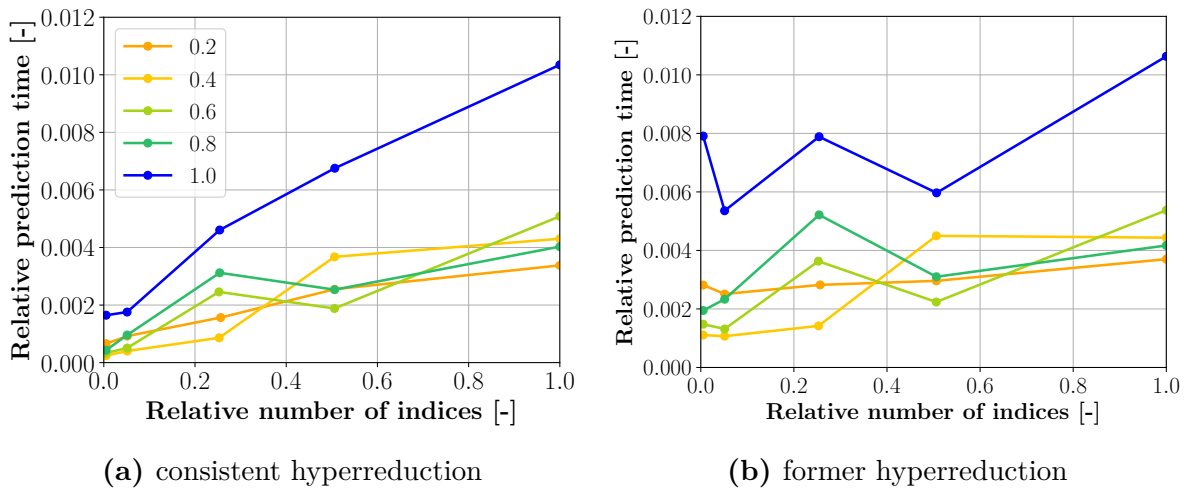


Figure 4.37: RAE2822: Relative prediction times for different compression rates.

the optimizer are resolved. At the same time, the usage of the hyperreduction level as a parameter allows to test the estimations from section 2.4 that suggest a linear scaling of acceleration potential of the new consistent hyperreduction implementation. Figure 4.38 shows that indeed the acceleration seems to scale in nearly linear fashion. The normalized relative prediction time flattens a bit with increasing hyperreduction level. However, this might be explained by the growing influence of some offset costs for the optimization process like the interpolation of the initial POD coefficients. So, the acceleration of the LSQ-ROM online stage can be estimated by the hyperreduction level as defined by eq. (2.99).

Assessment of the results

The investigations of the LSQ-ROM for the predictions of transonic flow fields show some important differences to the predictions in the subsonic regime. While the subsonic design space is sufficiently captured by a small number of POD modes, the transonic design space includes additional flow phenomena like shocks and shock induced separations that require more POD modes. This is due to the nonlinear characteristic of these flow phenomena, which are depending in a nonlinear way on the parameters defining the transonic design space. Therefore, the POD struggles capturing nonlinearities like varying shock locations [17], and the LSQ-ROM requires a higher number of POD modes for the prediction in the transonic regime. The relative information content of several reduced POD bases is plotted in fig. 4.12, where it is evident that several additional POD modes are required to reach the same RIC as for the POD basis for the subsonic regime. In addition to that, even POD modes with small contributions to the total information content can include important information for the low dimensional representation of the FOM [17]. The results shown for the RAE2822 test case support the assumed difficulties, as the accuracy of the predicted flow fields can not match the results obtained for the subsonic case. Although the flow fields exhibit relatively high errors in the proximity of shocks, the pressure distributions still show good agreement with the FOM. To capture all the important flow phenomena, the LSQ-ROM requires POD bases that include at least 40% of all modes. This value can not be understood as a general benchmark, as the error induced by the POD representation is affected by the density and location of the training points in the design space [47], but it still clarifies the effect of the transonic phenomena in the design space. The hyperreduction instead can be driven to rather high levels of reduction. While models with low compression rates seem to be more affected by high

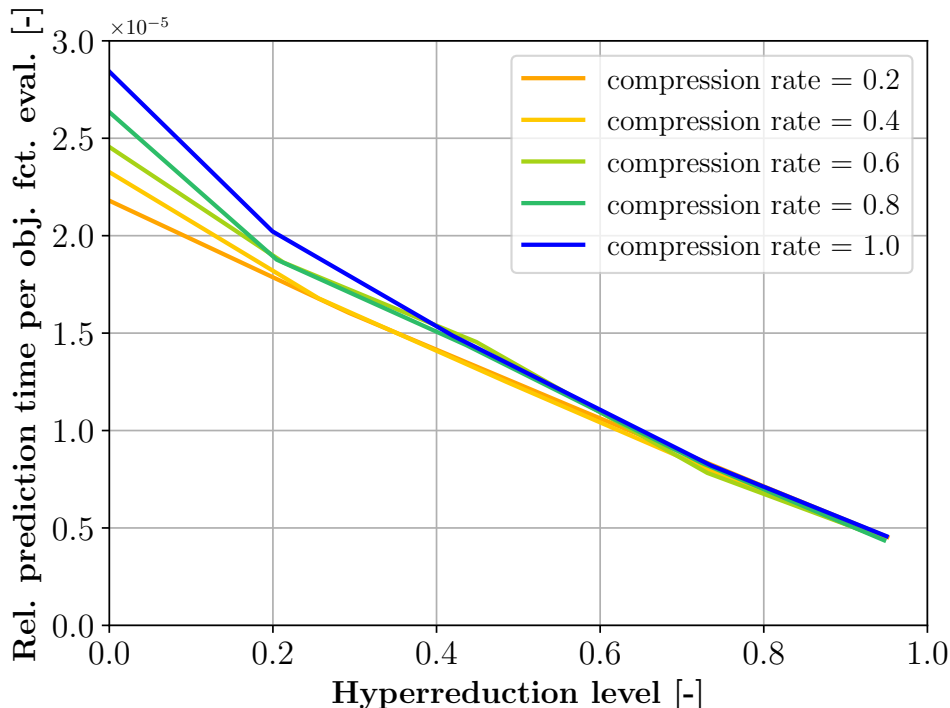


Figure 4.38: RAE2822: Influence of the hyperreduction level on the normalized prediction time.

levels of the hyperreduction, the accuracy of models with high compression rates is only slightly influenced by it. So for example, the overall agreement of the predicted flow field with the FOM is better for a model with no reduction of the POD basis but the highest level of hyperreduction than for a model with a reduced POD basis but no hyperreduction. For future investigations it might be of interest to examine the influence of the POD basis on the quality of the hyperreduction indices.

Additional investigations of the accuracy lead to the unexpected results that the the overall flow field can be approximated more precisely by the POD+TPS model, whose interpolated solution serves as an initial solution for the optimization process in the LSQ-ROM. It can be shown that the reason for this is the minimization of high residuals in the region of the predicted shock. So, while the interpolated solutions do not enforce any relation to the physically correct behaviour at such strongly nonlinearities, the intrusive character of the LSQ-ROM assures a better agreement with the FOM at such locations. The LSQ-ROM is able to accurately predict the magnitude and location of the shock, whereas the interpolated solution exhibits unphysical over- and undershoots. However, the enforcement of the residual reduction at the shock comes at a loss in accuracy for the remaining flow field. The superior prediction quality of POD+TPS the in terms of the whole flow domain are also observed in the investigation in [2]. Nevertheless, the incorporation of the residual minimization should prevent the occurrence of local error spikes as their resulting residuals are effectively reduced during the optimization process. So with this kind of error damping large over- or undershoots are settled, which becomes even more obvious for the examined prediction of an extrapolation point. Herein, the interpolated solution is clearly outperformed by the LSQ-ROM not only for the surface pressure at the shock but almost for the whole airfoil surface. So, when the interpolation falls short due to a lack of sampling points, the LSQ-ROM still leads to sufficient predictions. The extrapolation capability of the LSQ-ROM was already investigated in [62], which gives similar results for the comparison of the LSQ-ROM and POD+TPS.

In terms of prediction time, the investigation confirms the main findings of the subsonic test case. The consistent submesh-based hyperreduction method effectively reduces the time spent during the online stage by reducing the effort for the evaluation of the residual and for the reconstruction of the states from the POD coefficients. It is shown that the potential acceleration is estimated correctly by the linear dependency on the hyperreduction level, which is based on the submesh size. As the growth rate of the submesh is high for relatively small sets of hyperreduction indices and flattens towards larger sets of indices, the new hyperreduction implementation overproportionally benefits from small set of indices. So, especially for small numbers of indices, the consistent hyperreduction is superior to the former implementation. The results for this test case suggest speed-up factors of the online stage, which are up to five times higher than for predictions with the former hyperreduction. Together with the outcome from the subsonic test case, the newly implemented hyperreduction proves its effectiveness and its advantage over the former implementation for the prediction of two dimensional flow fields.

Recommendation for proper reduction levels The findings for the transonic test case suggest that it is less critical for the prediction accuracy to apply a stronger hyperreduction than to reduce to POD basis. For the given test case, lower compression rates of e.g. 0.4 for the POD basis can be chosen. However, for such reduction levels the potential of the hyperreduction can not be fully exploited without degrading the prediction. Since high hyperreduction levels are preferred to take advantage of the acceleration of the consistent hyperreduction, rather high compression rates of 0.6 or 0.8 are recommended to use hyperreduction levels that solely use 0.5% of all residual entries. Proper reduction levels for the POD basis are depending on the design space and the distribution of training points in it. Therefore, a generalized recommendation can not be given. However, with respect to the hyperreduction it is assumed that the POD basis needs to be able to accurately represent the design space as the POD basis directly influences the hyperreduction indices. So, when choosing a proper POD basis, it should be possible to utilize rather high levels of hyperreduction. In case of the investigated test case, the proposed reduction levels (compression rate = 0.8, relative number of indices = 0.005) still lead to a distinct acceleration of the online stage and simultaneously retain high accuracy for important measures like the surface pressure distribution. Compared to an LSQ-ROM without any reduction, the whole prediction process can be accelerated by a factor of around 24 leading to a speed-up of around 2300 compared to the FOM. In contrast, the application of the former hyperreduction leads to for the same reduction levels only to a speed-up of 500.

4.2.3 NASA-CRM: three-dimensional, transonic flow

The NASA-CRM generic civil aircraft configuration and its mesh, which discretizes the three-dimensional domain, are shown in section 3.3.1. The aim of the investigations for the flow around the NASA-CRM is to check if the results for the two-dimensional problem can be transferred to a three-dimensional problem that is more representative of a potential industrial application of LSQ-ROM. The dimension influences not only the problem size n , but also the residual computation and the construction of the three-dimensional submesh for the consistent hyperreduction. In addition to that, the flow around a full aircraft is more complex than the flow around a simple airfoil. Therefore, it becomes necessary to confirm the previous outcomes by additional investigations for a 3D test case. The investigations are based on a design space that is once again defined by the Mach number and angle of attack in the ranges:

$$\text{Mach} = [0.75, 0.95] \quad \text{angle of attack} = [-2^\circ, 6^\circ]$$

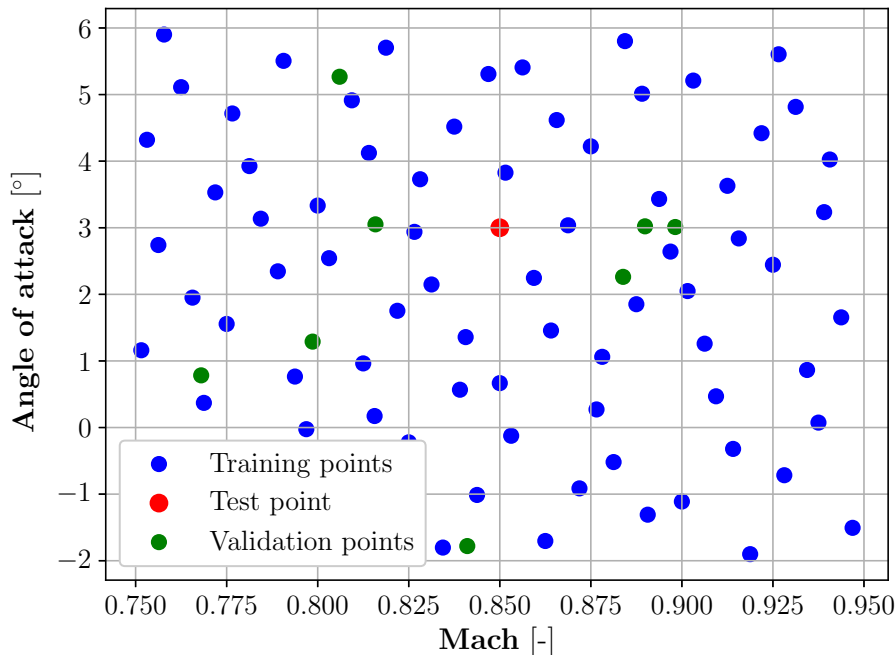


Figure 4.39: Design space for the prediction of the transonic flow around the CRM.

The specific training points are defined by a two-dimensional Halton sequence and are illustrated in fig. 4.39 together with nine randomly chosen validation points and the test point at:

$$\text{Mach} = 0.85 \quad \text{angle of attack} = 3.0$$

The validation points are utilized to ensure that the test point can be taken as representative point for the design space, so that it is not outperforming all other validation points in terms of accuracy. As for the 2D cases, the test and validation points are compared based on two heatmaps that illustrate the objective function values at the end of the optimization process and the summed squared error of the total energy fields as it is defined in eq. (3.3). The predictions for every point refer to an LSQ-ROM without reduction. For an easier comparison, the values in fig. 4.40 are normalized by the maximum value of all investigated points. Figure 4.40a shows that the test point is not exceptional in terms of the objective function value. In fact, the prediction of the test point leads to an objective function value that ranks among the highest value of the validation points. The same can be observed for the summed error in fig. 4.40b. A reasonable correspondence can be seen between the relative ranking of the objective function values and the respective summed error of the prediction. Similar to the transonic 2D case, the summed residual indicates the magnitude of the prediction error, although the relation is not as clear as for the subsonic test case (cf. fig. 4.18).

Accuracy for different reduction levels

The accuracy of the predicted solution is assessed with respect to the CODA reference solution, which represents the FOM. The pressure coefficient is illustrated in fig. 4.41 for the FOM (top) on the surface of the CRM and for an LSQ-ROM (bottom) that operates on a full POD basis with no hyperreduction. Figure 4.41 shows the upper side of the swept wing and the fuselage of the CRM. At the nose of the configuration a small area of increased pressure can be observed, which is due to the presence of a stagnation point at the nose tip. The wing exhibits over its whole span a low pressure

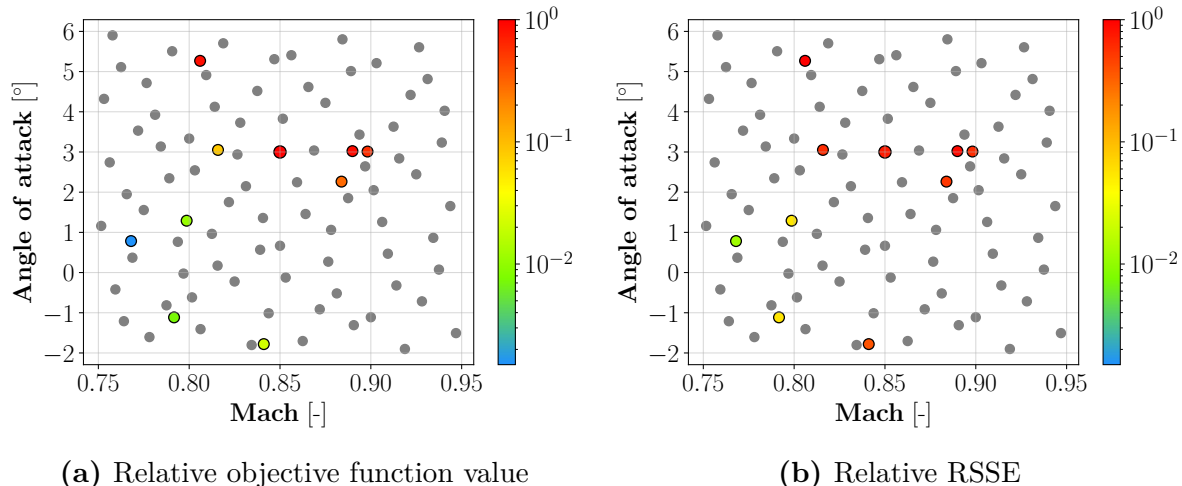


Figure 4.40: NASA-CRM: Heatmaps for the validation of the test point.

region that starts at the leading edge and ends with a compression shock at around 40% to 60% of the respective chord length. It can be seen that the LSQ-ROM is able to predict a pressure distribution that is in good agreement with the FOM. Nevertheless, the location of the shock front is a bit off and further deviations are present at the root and tip of the wing. A more detailed illustration of the accuracy of the LSQ-ROM with respect to various levels of hyperreduction is given in fig. 4.42. Figure 4.42a displays various pressure distributions for a compression rate of 1.0 at a slice in the Z-X plane located at 50% of the wing span. Comparing the pressure distributions of the LSQ-ROMs with the reference solution reveals distinct deviations between the FOM and the ROM. Besides the LSQ-ROM with the lowest number of relative indices for the residual, all LSQ-ROMs exhibit similar predictions of the pressure coefficient. While the pressure side is predicted with a high accuracy, the suction side exhibits differences along the total chord length. In contrast to the two-dimensional transonic case, where the LSQ-ROM is able to precisely predict the pressure around the shock, the 3D case exhibits deviations also at the shock location. Although the shock phenomena is still captured, the magnitude is underestimated and its location is shifted a bit towards the trailing edge. The loss in accuracy compared to the 2D case is somehow expected, because of the more complex flow field, which would probably require a much higher density of training points in order to achieve a similar accuracy. Nevertheless, the influence of the hyperreduction is relatively small, so only the model with the lowest number of indices shows significant differences in front of the compression shock compared to the LSQ-ROMs with higher numbers of indices. When reducing the POD basis to a compression rate of 0.5, the agreement with the reference solution further degrades as it can be seen from fig. 4.42b. The predicted pressure distributions are not able anymore to fully capture the shock at the suction side: they highly overestimate the pressure before the previous shock location and they underestimate it behind this location. This degradation of the predicted solution is not surprising, as the the model with the double amount of POD modes already struggles to accurately capture the pressure distribution at the suction side. The pressure side at this wing location is still captured very precisely, which is probably due to the small influence of the input parameter on the flow at the pressure side. Similar to the results from fig. 4.42a, the pressure distributions are only slightly influenced by the hyperreduction. Merely the highest level for the hyperreduction clearly deviates from the predictions of the all other models. Further pressure distributions for other wing span locations can be found in appendix C. As the presented results indicate that the highest errors along the wing emerge on the suction side, the next depictions focus on the errors of the pressure coefficient at the suction side of the wing. For that, fig. 4.43 illustrates

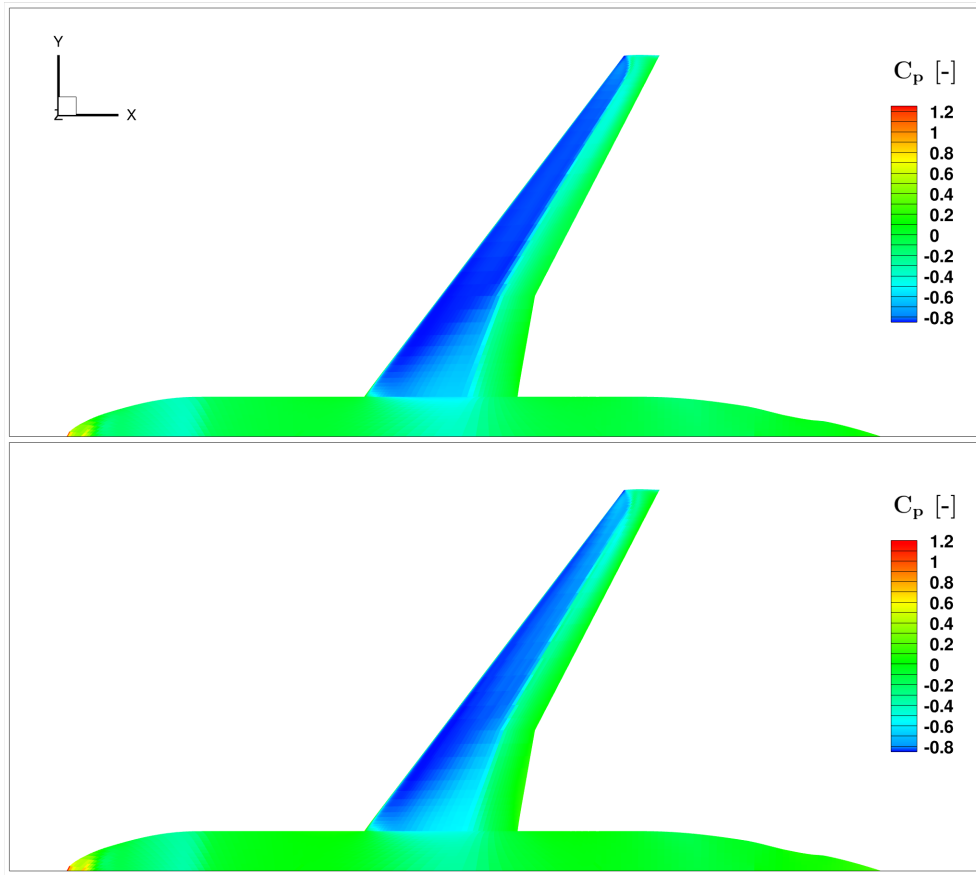
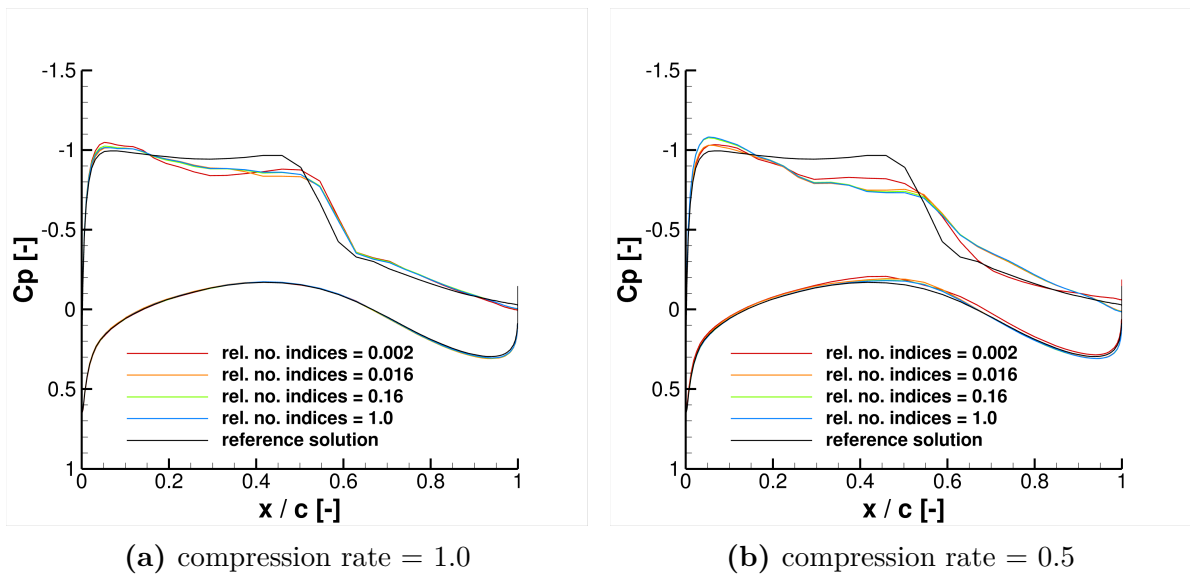


Figure 4.41: NASA-CRM: Pressure distributions of the FOM and the unreduced LSQ-ROM at $M = 0.85$ and $\alpha = 3.0^\circ$.



(a) compression rate = 1.0

(b) compression rate = 0.5

Figure 4.42: NASA-CRM: Pressure distributions for a slice at 50% of the wing span.

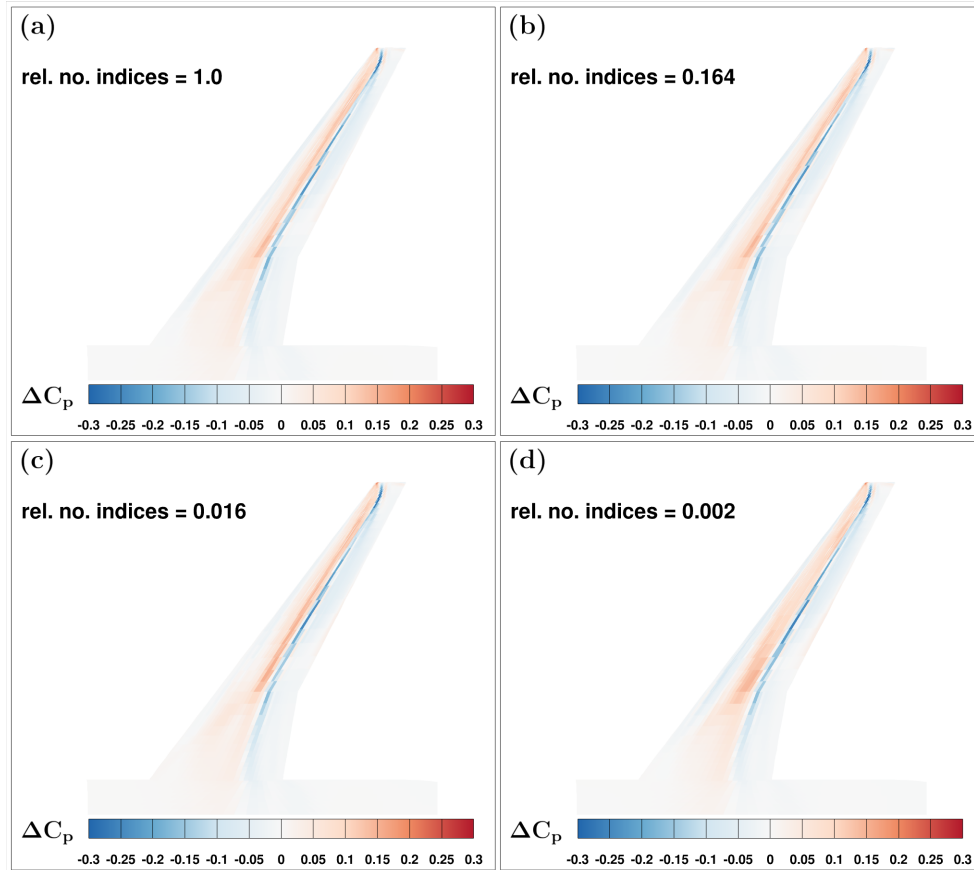


Figure 4.43: NASA-CRM: Error of the surface pressure coefficient for a compression rate of 1.0.

the difference in the pressure coefficient between the specific LSQ-ROM and the reference FOM. The error is shown again for four different levels of the hyperreduction, described by the relative number of residual indices taken into account. It is possible to see that the highest deviations stretch span-wise along the shock front as described for fig. 4.41. So, the overestimation of the pressure before the shock front, which is already observed for the slice section at 50% of the wing span, is present almost along the full wing span. Behind the shock front, the pressure is underestimated leading to small band of negative deviations from the pressure coefficient of the reference solution. This characteristic behaviour of the error can be found for all illustrated model predictions. As for the pressure distributions at the presented slice location, the hyperreduction has a relatively small impact on the predicted surface pressure. Even for the model with the highest level of hyperreduction, the maximum error is only increased for a localized area at the wing tip. The effect of an additional reduction of the POD basis on the predicted surface pressure can be examined from fig. 4.44, which illustrates the pressure predicted by LSQ-ROMs with a compression rate of 0.5 and varying levels for the hyperreduction. Compared to the results from fig. 4.43, the surface solutions used for fig. 4.44 feature much higher deviations from the reference solution. The highest differences reach values of 0.3, while the predictions with full POD basis reached only deviations below 0.2. Nevertheless, the characteristics of the error distribution across the wing stay almost the same, but with an increased magnitude of the error. Also the sensitivity of the solution to the hyperreduction is quite similar to the one for the model with an unreduced POD basis. Anyway, from fig. 4.44 (d) and its comparison to the other error distributions (a,b, and c) it is clear that the influence of the hyperreduction can become substantial when the number of indices is

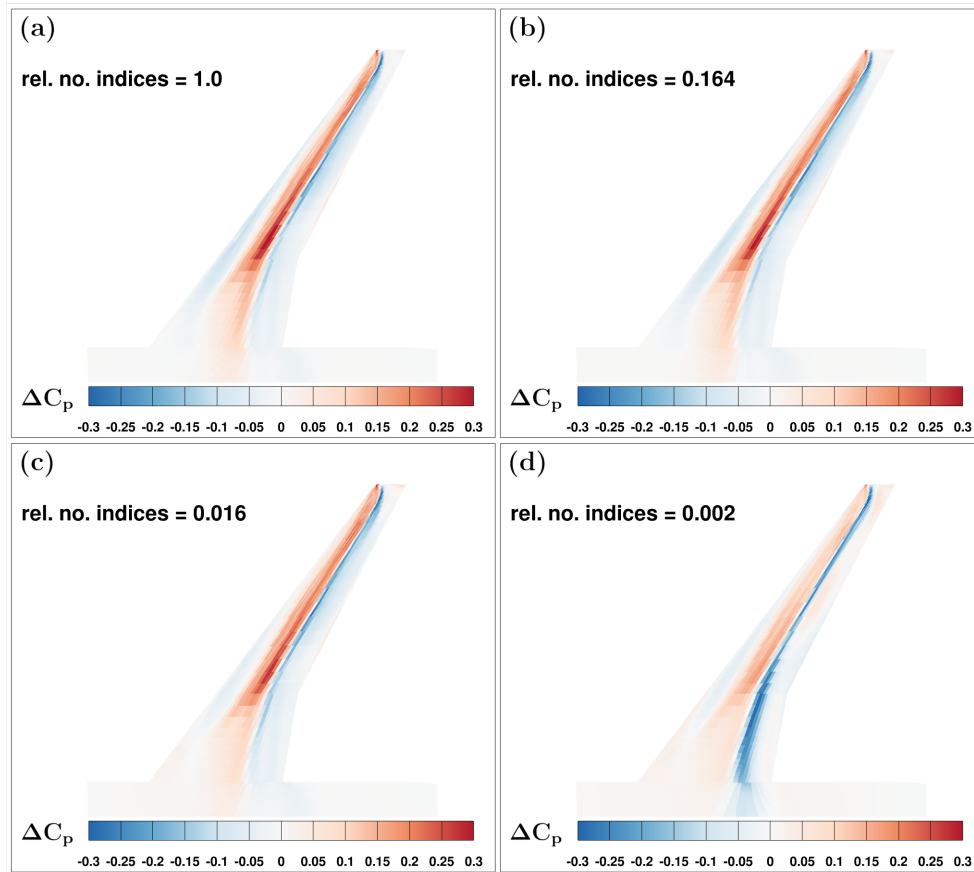


Figure 4.44: NASA-CRM: Error of the surface pressure coefficient for a compression rate of 0.5.

too low. As the underlying solution of (d) clearly deviates from the other solutions in fig. 4.44 and additionally in a more distinct way than fig. 4.43 (d), the LSQ-ROMs with the reduced POD basis seem to be more distinctly influenced by the hyperreduction than the LSQ-ROMs with the full POD basis. This trend is also one of the findings from the investigations of the two-dimensional transonic test case, and it is additionally supported by the illustration of the coefficient of determination for the surface pressure in fig. 4.45, where the coefficient of determination is displayed as a function of the reduction of the POD basis and the hyperreduction. It can be seen that the models with compression rates of 1.0 and 0.5 face solely slight decreases of the agreement of the surface pressure when increasing the level of the hyperreduction; for the compression rate of 0.5 it even increases a bit. In contrast, the decrease is more distinct for the models with lower compression rates. As for the two-dimensional case, it can be assumed that this degradation is caused by the selection of hyperreduction indices, which are less suited for the representation of the whole residual vector.

Comparison with the interpolated solution Similarly to the two-dimensional analyses, the LSQ-ROM for the CRM is compared with the interpolated solution referring to a non-intrusive POD+TPS model; it is important to remember that the interpolated solution is used in LSQ-ROM as a starting condition for the minimization of the residual vector. The comparison from section 4.2.2 shows that the residual minimization of the LSQ-ROM is not automatically leading to a higher agreement of the overall flow field, but rather focuses on reducing the most dominant errors in the flow field. For example, small errors in the proximity of nonlinearities – such as the shocks on the suction side of the CRM wing –

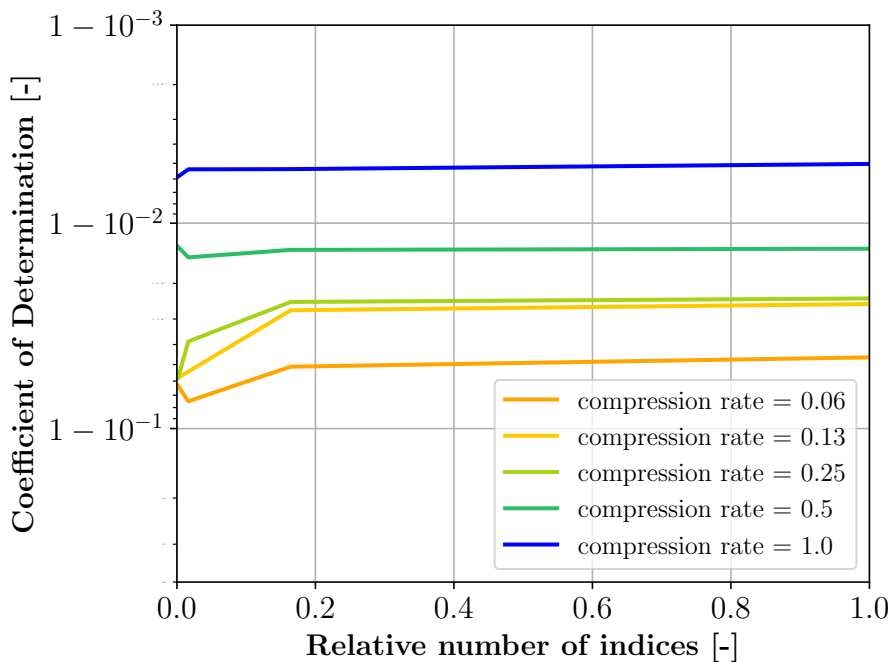


Figure 4.45: NASA-CRM: Coefficient of determination for the surface pressure for different compression rates.

can result in relatively high residuals, which probably become the focus of the residual optimization process. Nevertheless, the LSQ-ROM is expected to improve the accuracy in the regions of the largest residuals, which correspond in a nonlinear fashion to the actual deviations from the FOM. The agreement of the surface pressure is illustrated in fig. 4.46 for the predictions of the LSQ-ROM and the POD+TPS model. As expected, this integral measure for the accuracy of the predicted surface pressure is higher for the interpolated solutions than for the corresponding LSQ-ROM with the same POD basis. Even the POD+TPS model with a compression rate of 0.25 predicts a solution with a higher agreement than the best LSQ-ROM. Before examining the accuracy at the wing, where strong nonlinearities appear due to the shock front, it must be noted that the coefficient of determination from fig. 4.46 is computed for the whole surface of the CRM including the fuselage. In fig. 4.47 the difference in the pressure coefficient is displayed for the LSQ-ROM and the POD+TPS interpolated solution with respect to the FOM. It is remarkable to see that the LSQ-ROM without any reduction is outperforming the interpolation model for the prediction of the surface pressure around the shock front along the wing. The underestimation of the pressure ahead of the shock front is lower after the residual minimization of the LSQ-ROM, especially for the central part of the wing. So, in this case, the residual minimization within the whole flow field is able to reduce the maximum errors at the surface of the CRM. However, for a reduced POD basis with only half of all modes, the LSQ-ROM is not able to effectively reduce the errors at the wing surface. The interpolated solution from (c) instead is very similar to the one in (a), which is already indicated in fig. 4.46 by the small difference in the coefficient of determination. It is somehow unexpected that the errors increase, even the maximum errors, during the residual minimization process, since it should focus on the reduction of the relatively high residuals like it seems to be the case in (b). However, the surface values are just a small extract of the flow field, so the residual minimization is likely to decrease the residuals in cells not associated to the shown surfaces, which still result in a local minimum for the sum of all squared residuals. A speculation about why this behavior occurs for the CRM case but not for the 2D case might be the weighting of the residuals by the square root

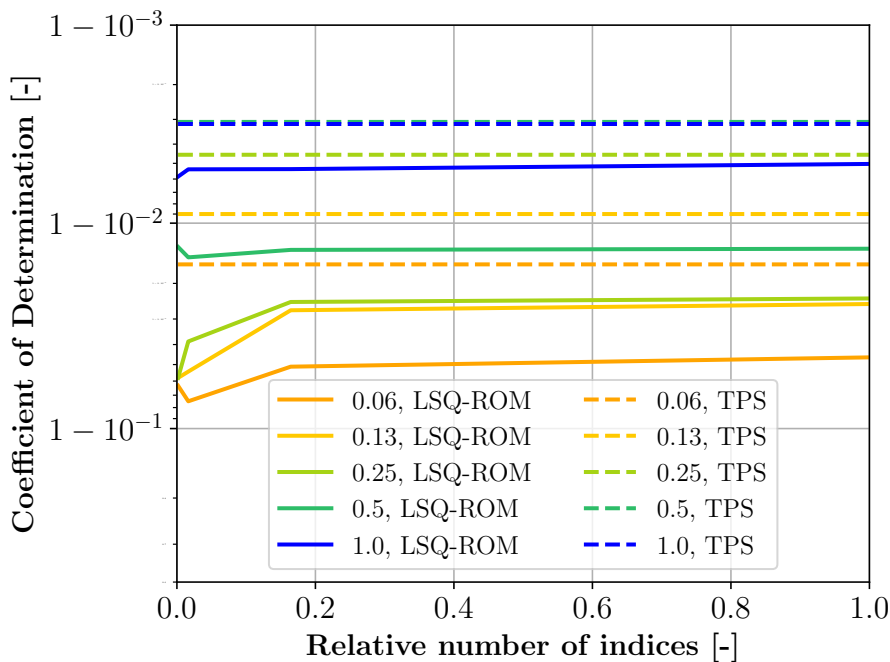


Figure 4.46: NASA-CRM: Coefficient of determination for the surface pressure of the LSQ-ROM and POD+TPS for different compression rates.

of the corresponding cell volume. While the mesh for the RAE2822 has extremely small cells at the surface of the airfoil compared to cells located farther away, the mesh of the CRM does not have such a fine resolution in the neighborhood of the wing. Therefore, the errors directly at the CRM wing are not favoured as much as for the RAE2822 during the residual minimization process.

Assessment of the submesh-based Hyperreduction in 3D

As already mentioned in the beginning of this section, the dimensionality of the predicted flow problem directly influences the performance of the consistent submesh-based hyperreduction due to the necessity to include the stencil of every residual / cell index selected by the MPE and DEIM. These three-dimensional stencils contain more neighbouring cells than the same stencil for a two-dimensional mesh. This fact is important as the acceleration potential of this hyperreduction is determined by the size of the submesh and not directly by the number of hyperreduction indices like for the former implementation. So, the connection between the number of indices and the submesh size is crucial for a potential acceleration of the hyperreduction. Hence, this connection is investigated with respect to the expected changes due to the additional dimension of the flow problem. To evenly compare both hyperreduction implementations, the number of (hyperreduction) indices is still utilized. Figure 4.48 displays a section of two different submeshes of the CRM. While (a) and (c) illustrate only the cells, which directly refer to hyperreduction indices, (b) and (d) additionally show the cells contained in the respective stencils. As the hyperreduction indices are mostly concentrated around the wing of the CRM other parts of the submesh are not visualized. Nevertheless, all the submeshes contain further cells at the farfield boundary and in the wake of the wing and the fuselage. From the comparison of (a) and (b), it becomes clear that the submesh is substantially growing through addition of the stencil cells. The full mesh is purely based on hexahedrons and adding the neighbours and the neighbours of the neighbours to a hexahedron leads to 24

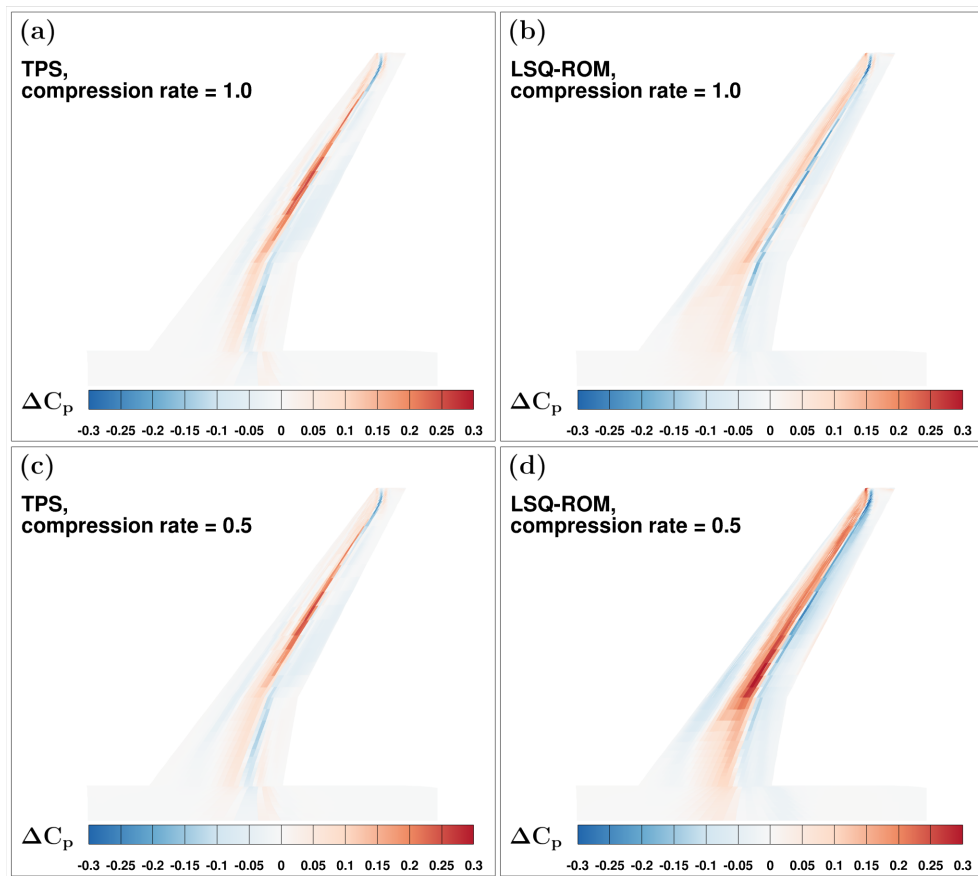


Figure 4.47: NASA-CRM: Error of the surface pressure coefficient for solutions of the LSQ-ROM and POD+TPS.

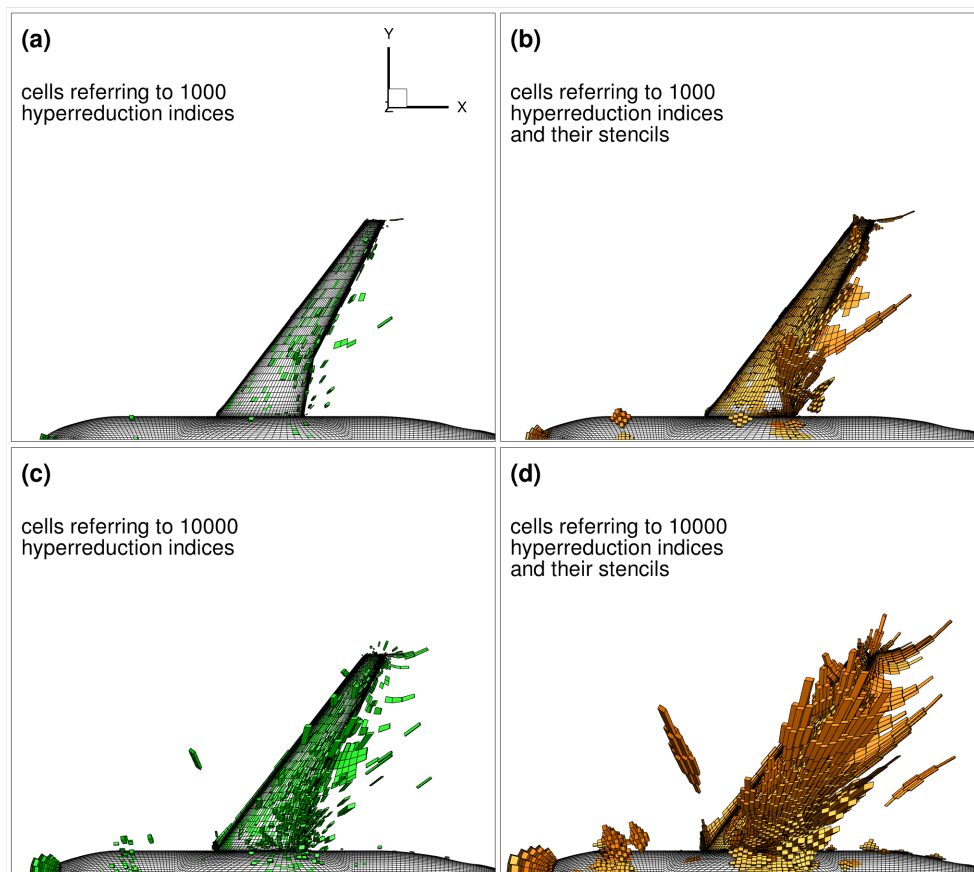


Figure 4.48: NASA-CRM: Submeshes utilized for the consistent hyperreduction.

more cells, which are twice as many as in a two-dimensional case. However, the hyperreduction indices are clustered in small regions of the whole domain as it can be seen from fig. 4.48. As a result, adding another cell with its stencil to an existing mesh implies much less new cells added to the submesh than the full 24. An illustration of this growth rate is given in fig. 4.23. It displays the relative number of cells in the submesh with respect to the relative number of indices picked for its construction. The green line poses the unrealistic but ideal case that all indices are perfectly clustered and thus implying only the cells referring to the hyperreduction index. In contrast to that, the red line is based on randomly distributed hyperreduction indices. The real distribution of the hyperreduction indices for the CRM lays between these two extrema indicating a clustering of the indices, which is also part of the results from fig. 4.48. The pessimistic estimation that all indices are somehow randomly distributed in the domain would lead to extremely fast growing submeshes. In fact, 20% of the total number indices would be enough to obtain a submesh containing more than 99% of the original mesh. Compared to the two-dimensional case, where 20% of the indices would result in submesh containing roughly 90% of all cells (cf. fig. 4.23), this scenario is even worse for the three-dimensional case. Based on the estimations for the acceleration potential of the hyperreduction (cf. section 2.4), this would automatically result in computational times similar to the former implementation. However, the clustering of the hyperreduction indices reduces the growth rate, so that 20% of all indices result in a submesh, which still has only half of the size of the original mesh. The overall behaviour of the growth rate of the submesh is quite comparable to the one presented for the NLR7301 mesh. It has a flattening increase in the submesh size with increasing number of hyperreduction indices, which means that a reduction of the number of indices reduces smaller submeshes more distinct than larger ones. So, to maximize the potential advantage of the consistent hyperreduction over the former hyperreduction, the

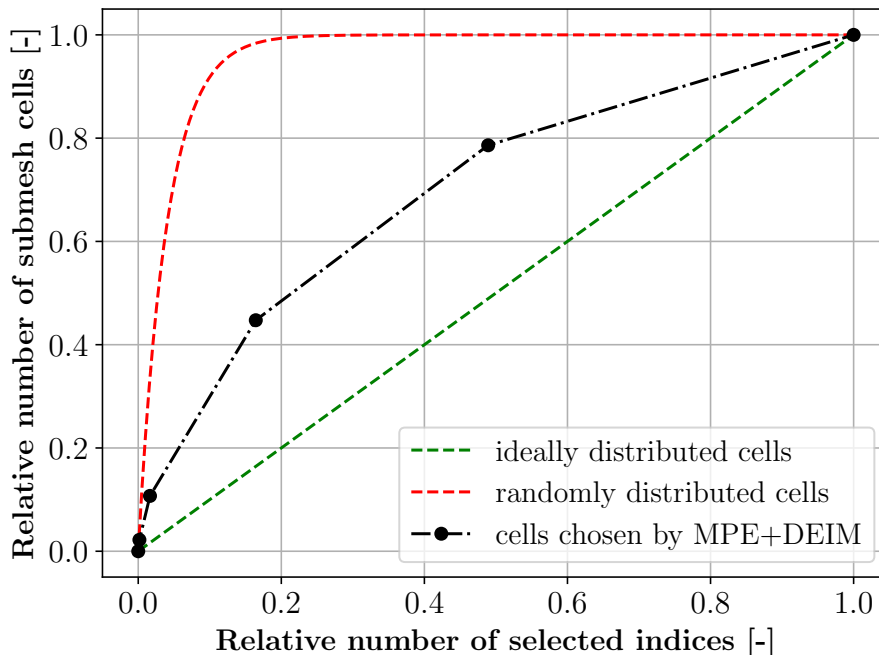


Figure 4.49: 3D: Relative number of submesh cells depending on the relative number of selected indices.

number of indices should be as low as possible.

Acceleration potential for different reduction levels

Although the performance of the new hyperreduction implementation is already confirmed for the 2D test cases in section 4.2.2 and section 4.2.1, an investigation of a three-dimensional problem is still of interest, because of the dependency of the submesh size on the dimension and the increased problem size n . In addition to that, the investigations in [3] revealed for the application of the LSQ-ROM to an unsteady flow around the CRM, that the former hyperreduction becomes less effective for larger meshes, because the main part of the prediction time is spent for the evaluation of the residual and not in the LM algorithm. A direct comparison with the findings for the steady application is not sought, but the upcoming investigations assess the new implementation with respect to the estimations in [3] for the acceleration of a hyperreduction that also acts on the residual evaluation. First, it is verified that the residual evaluation in the CFD solver still scales linearly with the number of cells. For that, the average time spent in the evaluation of the objective function during the prediction stage is displayed in fig. 4.50 with respect to the hyperreduction level. Indeed, the time per objective function evaluation scales linearly with the size of the submesh. This is an important keystone for the effectiveness of the submesh-based hyperreduction, but rather confirms a desirable characteristic of the CFD solver than of the hyperreduction itself. The acceleration potential for the full online stage is illustrated in fig. 4.51a by the relative prediction time. As expected, both reduction methods lead to decrease in the prediction time, which is normalized by the time spent for the FOM. It is possible to see that the decrease of the prediction time with a decreasing number of indices corresponds to the behaviour of the submesh growth in fig. 4.49. Therefore, the reduction in the prediction time is enhanced with the decrease of the number of indices. The relative prediction time for the unreduced LSQ-ROM corresponds to a speed-up factor of around 9 compared to the FOM. Applying the highest

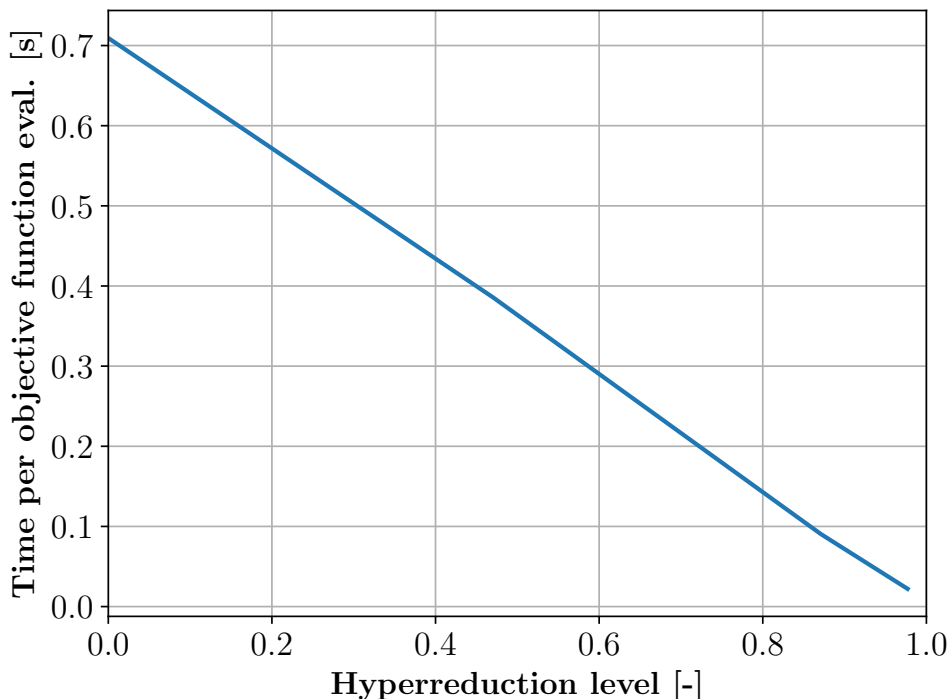


Figure 4.50: NASA-CRM: Time spent for the evaluation of the residual / objective function.

level of hyperreduction (0.16% of all indices) leads to a prediction that is more than 400 times faster than the FOM. So, the solely the hyperreduction can reduce the prediction time by a factor of 48. For the other models with lower compression rates, the hyperreduction accelerates the specific model by a factor between 13 and 51. These factors are multiple times higher than for the transonic 2D case. This is mostly because of the lower hyperreduction level applied in for the CRM case, but also because of the larger problem size n , which increases the share of the time for the residual evaluation on the total prediction time. Although the previous results show that the highest reduction levels for both the POD bases and the hyperreduction do not lead to accurate predictions, their application in the LSQ-ROM would dramatically decrease the prediction time. For the lowest compression rate and number of indices the LSQ-ROM becomes around 5500 times faster than the FOM. In order to assess the benefits of the consistent hyperreduction, the relative prediction times of the LSQ-ROMs with the former hyperreduction are displayed in fig. 4.51b to compare them with the times in fig. 4.51a. The illustrated prediction times decrease for models with a lower compression rate. By applying the former hyperreduction, the prediction time is decreased only slightly. For various models, the prediction time is even increased for higher levels of hyperreduction. However, this is caused by an higher number iterations in the optimizer, which could not be compensated by the hyperreduction. The consistent hyperreduction faces the same trend but is clearly able to still reduce the overall prediction time. With the former implementation the model with a compression rate of 1.0 is only accelerated by a maximum factor of around 6, which is the highest value among all investigated models. In contrast to that, the consistent hyperreduction can reduce the prediction times up to a full order of magnitude. Nevertheless, to obtain such superior acceleration results, it is crucial that the number of indices is extremely low so that the submesh size if effectively reduced. This becomes clear when examining the speed-up factor for the LSQ-ROM for the different hyperreduction implementations. Thereby, the speed-up factor is the prediction time with the former hyperreduction divided by the time with the consistent hyperreduction. From fig. 4.52 it

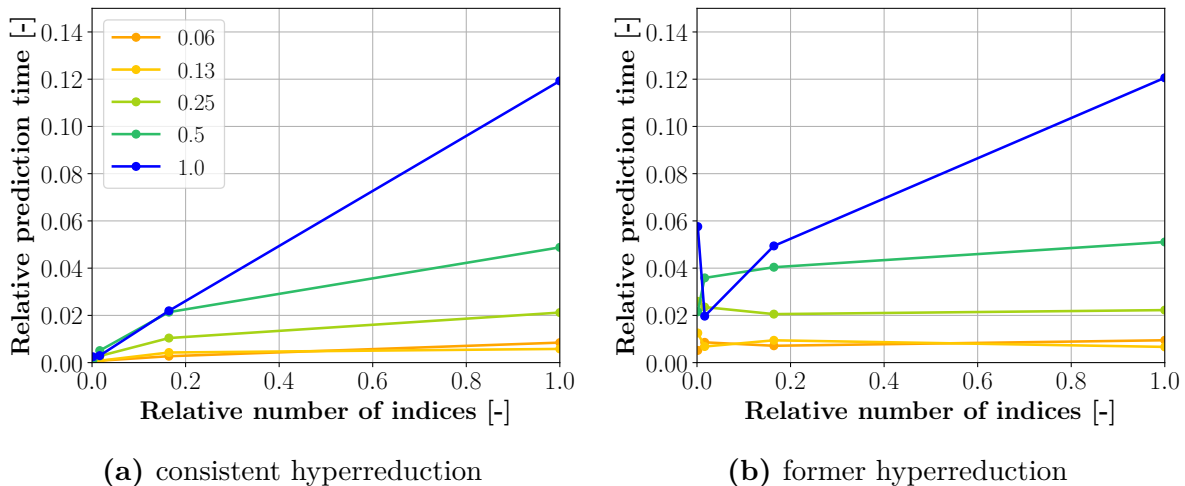


Figure 4.51: NASA-CRM: Relative prediction times for different compression rates.

is possible to see that the speed-up through the consistent hyperreduction dramatically increases for low numbers of hyperreduction indices. When taking 20% of all indices into account the speed-up factors are around 2.5, whereas a further reduction of this number below 0.2% leads to speed-ups between 22 and 28 depending on the actual submesh size. Through comparison of the prediction times of the two hyperreduction implementations it is also possible to assess the acceleration estimations from [3]. These estimations are done in the context of an unsteady LSQ-ROM and they assume an acceleration of the whole online stage by a factor 5 for a model with a relative number of indices of 5% and an POD basis with 60 modes. Despite some differences between the unsteady and steady LSQ-ROM algorithm, the estimated acceleration should be at least of the same order of magnitude. When examining the speed-up factors at the given relative number of indices, it can be seen that all the models exhibit speed-up factors between 6 and 9, which are even a bit higher than the estimated factor of 5 from [3]. So the real speed-up factors are in good agreement with the estimated factor, especially when considering the different context of the estimations.

Assessment of the results

By investigating the LSQ-ROM for the prediction of a transonic, three-dimensional flow, it is possible to confirm important findings from the two-dimensional case and to add further results that apply to three-dimensional problems with a larger problem size. So, as the design space for the CRM is also fully transonic, similar issues concerning the representation of nonlinear shock phenomena by a linear combination of POD modes are expected. Beyond that, the shock is not anymore a two dimensional phenomena, but forms a three-dimensional shock front. Therefore, the POD basis needs to capture more complex flow field affected by extensive areas dominated by compression shocks. The results examining the accuracy of the predicted solutions indeed reveal similar issues for the prediction of the flow including shocks. Even the full POD basis is not able to predict the surface pressure at the wing of the CRM as precisely as the POD basis for the RAE2822. This can be explained by the higher complexity of the flow, but also by the different design spaces. Although the design space for the CRM includes more training points, the range for the Mach number is doubled, so that the chosen number of training points may be not sufficient to reach the the same level accuracy like the prediction for the RAE2822. Nevertheless, the investigated LSQ-ROMs based on the full POD basis are still able to capture main flow characteristics at the surface of the CRM. Particularly,

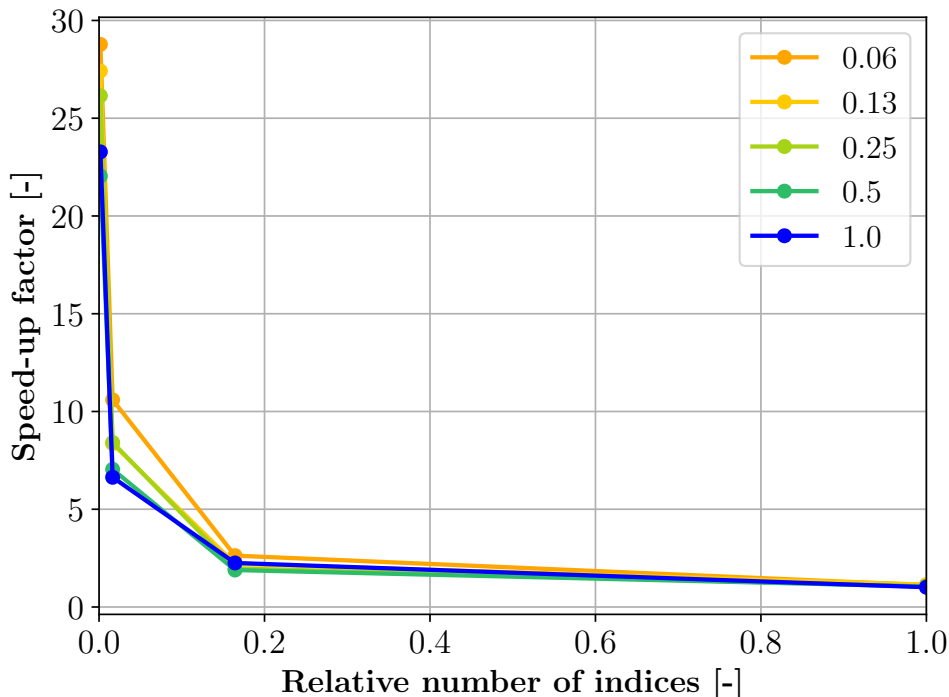


Figure 4.52: NASA-CRM: Speed-up factors of the consistent hyperreduction for different compression rates.

for areas aside the shock dominated flow at the suction side of the wing, the agreement with the FOM is overall acceptable. The highest deviations occur ahead and behind the shock front at the suction side of the wing, which suggests that the POD basis misses information to accurately represent the shock, especially its location. As a consequence, a considerable reduction of the POD is not possible without causing a significant loss in accuracy. Generally, the hyperreduction constitutes a feasible way to reduce the effort for the LSQ-ROM as even extreme reduction levels including less than 2% of all residuals entries do not degrade the solution beyond reasonable limits. Similar to the transonic 2D test case, the hyperreduction can be further extended, when choosing higher compression rates. While the full POD basis allows to reduce the number of indices below 0.2%, the models with lower compression rates are more sensitive and require higher numbers to prevent a further loss in accuracy due to the hyperreduction.

The comparison of the LSQ-ROM with the POD+TPS model – which generates the POD coefficients used as optimization initial condition – confirms the findings from the 2D case. The interpolation model outperforms the LSQ-ROM for the accuracy in a global sense. The residual minimization procedure in the LSQ-ROM seems to reduce localized errors around the shock front, which is therefore captured more accurately by the LSQ-ROM. With respect to the newly implemented hyperreduction, it is shown that the submesh size is not growing as fast as one might expect from the larger stencils in three dimensions. However, the same relative number of hyperreduction indices still leads to a larger mesh in 3D than in 2D. Considering that the acceleration should linearly depend on the size of the submesh, the reduction potential is decreased in 3D, at least for a pure residual evaluation.

The consistent hyperreduction is able to achieve clear reduction of the prediction time. For the lowest number of hyperreduction indices (0.16%), the consistent hyperreduction can accelerate the models without hyperreduction up to a factor of 50. The former

hyperreduction instead reaches only factors of 6 emphasizing the large acceleration advantage of the consistent hyperreduction. The direct comparison of the prediction times for the two hyperreductions exhibits distinct speed-up factors for predictions with the consistent hyperreduction, which can reach values around 25 for very low numbers of the hyperreduction indices. These results confirm the superior performance of the new hyperreduction implementation also in three dimensions. For the same levels of hyperreduction, the speed-ups are similar for the 2D and 3D cases, although an increased problem size should increase the share of the time spent for the residual computation. Anyway, the small number of POD modes limits the effort for the linear algebra operations in the LM algorithm, so its share is relatively small even for the 2D problems. Moreover, the growth behaviour of the submesh counteracts the trend towards a higher effectiveness of the consistent hyperreduction. As a final result, the acceleration estimations from [3] for a consistent hyperreduction are verified. The estimated speed-up factor for such a hyperreduction is in convincing agreement with the one calculated from the presented result.

Recommendation for proper reduction levels As already described, the prediction of the flow around the CRM should utilize a preferably complete POD basis in order to obtain a solution with a sufficient accuracy level. For predictions with a similar accuracy level like the 2D prediction, it seems to be necessary to increase the density of training points in the design space due to the transonic conditions and the dimensionality of the flow field. For the design space utilized in this work, a reduction of the POD basis is not recommended, since the reduction can cause a severe accuracy deterioration. However, a full POD model allows to choose high levels for the hyperreduction, which leads to distinct time savings. For the presented test case and an unreduced POD basis, the number of indices can be reduced down to about 0.2% of all indices. With these reduction levels a similar accuracy like the baseline LSQ-ROM without any reduction can be obtained, while reducing the prediction time by a factor of 48. The resulting ROM leads to a prediction time that is 400 times lower compared to the FOM. In contrast to that, the application of the former hyperreduction would result in a model that is only 17 times faster than the FOM, which is a distinct acceleration, yet the speed-up with the consistent implementation of the hyperreduction is more than 20 times higher.

Chapter 5

Conclusion

Reduced order models (ROMs) are popular techniques that are conceived to reduce the growing computational effort needed for the solution of complex problems in several engineering fields. Besides purely data-driven models like neural networks or interpolation based methods, multiple intrusive ROMs have been proposed and successfully tested like the Galerkin projection, the GNAT method or the LSQ-ROM, which is investigated in this thesis for the prediction of steady-state problems.

More specifically, the LSQ-ROM couples a POD reduced space representation with the direct minimization of the residual vector computed by the full order model (FOM), which usually consists of a numerical Finite-Volume solver for partial differential equation. This link between the LSQ-ROM and the FOM retains the full description of the underlying physics in the prediction framework, but it also causes the ROM model to scale with the original problem size, e.g. the size of the CFD mesh. In order to overcome this limitation, hyperreduction techniques are included into the model in order to reduce the computational cost associated with the residual evaluation. Within the DLR research group, the current implementation of hyperreduction in the LSQ-ROM framework is limited to a reduction of the residual after its evaluation in DLR's CFD solver TAU. So, the full potential benefit of a hyperreduction is not exhausted because of some intrinsic restriction of the available solver TAU. However, the new CFD solver CODA, which is currently under development, allows new approaches for the implementation of a consistent hyperreduction.

This thesis work presents the implementation of a consistent hyperreduction that utilizes a reduced submesh for the evaluation of the residual, and thus effectively removes the dependency on the original problem size. The benefit of this consistent hyperreduction is tested for 2D test cases in subsonic and transonic conditions as well as for a transonic 3D test case. The application of this consistent hyperreduction implementation allows to significantly reduce the time spent in the online stage of the LSQ-ROM, which leads to predictions that are several hundred times faster for the 3D case and several thousand times faster for the 2D cases. For high levels of hyperreduction, the consistent hyperreduction leads to speed-up factors up to 25 compared to the former hyperreduction implementation. Overall, this thesis successfully implements a submesh-based hyperreduction, and the result analyses highlight some key aspects that are briefly described in the following paragraphs.

First of all, it can be shown that the POD basis can be reduced up to relatively high level when operating in subsonic conditions because the resulting aerodynamics do not exhibit strong nonlinearities. However, for the investigated design spaces in the transonic flow regime, an accurate representation of the nonlinear phenomena like shocks requires additional POD modes, i.e. a lower level of reduction. Although the density of training points is more than doubled for the transonic test cases, results show that an inappropriate reduction of the POD basis can easily lead to a distinct accuracy deterioration of the LSQ-ROM.

The second observation is the influence of the POD basis reduction on the effectiveness of the hyperreduction selection process. It has been noticed that the hyperreduction indices

seem less suited for the representation of the full residual when computed from a POD basis that is inappropriate to properly characterize the solution space. As a result, the hyperreduction can cause distinct deterioration of the prediction accuracy when applied together with a reduction of the POD basis that is too high. Thus, the reduction of the POD basis of the LSQ-ROM is critical for the prediction quality, whereas high levels of hyperreduction seem to be applicable as long as the POD basis effectively spans the design space.

With respect to the hyperreduction indices, a preliminary set of analysis revealed that the indices selected by the DEIM and MPE are missing information of the input parameter, which are defined only at the boundary of the computational domain. Therefore, the resulting reduced residual is not penalized when violating the imposed boundary conditions at the farfield boundary. As a consequence, the LSQ-ROM prediction process may converge to a local optimal solution that substantially violates the boundary conditions. The successfully implemented approach to circumvent this issue is to add a small percentage of indices corresponding to cells at the farfield boundaries to the primarily selected hyperreduction indices.

Apart from these accuracy related results, investigations are conducted to assess the performance of the consistent hyperreduction with respect to the expected reduction in computational cost and time. The time for the full online stage decreases according to the lowered effort for the residual evaluation and the reconstruction of the reduced state, which both scale linearly with the submesh size. The consistent hyperreduction allows to reduce the prediction time for all investigated test cases by around an order of magnitude (c.f. section 4.2.1 and section 4.2.2). For the larger 3D case, higher levels for the hyperreduction can be applied leading to even more distinct accelerations, that reduce the prediction time up to a factor of 50 (c.f. section 4.2.3). With this acceleration the LSQ-ROM becomes several hundred to thousand times faster than the FOM. In addition to the hyperreduction, the reduction of the POD basis leads to the expected reduction of the prediction time.

In comparison with the former hyperreduction, the results show that the submesh-based approach is clearly outperforming the former implementation. Although the former hyperreduction also leads to a reduced time spent in the online stage of the LSQ-ROM, the consistent hyperreduction is consistently superior thanks to its direct influence on the residual evaluation which represents the most computationally expensive operation in the LSQ-ROM algorithm. For example, the results show that the highest hyperreduction levels lead to speed-up factors of about 5.0 for the 2D test cases and 25 for the 3D test case. Therefore, the consistent hyperreduction exceedingly benefits from the possibility to reduce the residual to very small subsets.

It is evident that the influence of the two different reduction methods (POD basis reduction and hyperreduction) on the LSQ-ROM prediction accuracy and computational time is the result of a compromise between an acceptable accuracy level and a sufficient acceleration. The choice of proper reduction levels primarily depends on the design space and the flow regime it spans. For steady, subsonic applications of the LSQ-ROM it seems that the POD basis can be reduced to rather high levels without exceeding errors of 1% in the whole flow field. The hyperreduction is barely influencing the accuracy even when combined with reduced POD basis. Therefore, the LSQ-ROM can be accelerated by several orders of magnitude by choosing a low percentage of hyperreduction indices with a POD basis defined by only a few modes to accurately capture the design space. When aiming for an increased accuracy it is recommended to utilize more POD modes as their influence on the accuracy is more distinct than additional residual entries.

In case of a design space in the transonic regime, the POD basis can not be reduced as effectively as in the subsonic regime. The representation of nonlinearities requires a high number of POD modes, and therefore only a limited POD basis reduction can be achieved without excessively worsening the prediction quality. Because a reduction below a certain

level can also negatively influence the performance of the hyperreduction, a reduction of the POD basis should be only considered for design spaces with a sufficient coverage with training points. However, the hyperreduction can be driven towards similar reduction levels like in the subsonic case as long as the POD basis contains a sufficient number of modes.

Future investigations of the LSQ-ROM, particularly for the hyperreduction, could include the examination of different approaches to select the hyperreduction indices. Although the selected indices allow to effectively reduce the residual vector, the set of indices selected by the MPE is not optimal as the indices are selected in a greedy algorithm. There are several other approaches, e.g. for optimal sensor locations based on a Bayesian perspective [1], [76] that could possibly further improve the performance of the hyperreduction indices. More advanced approaches could also utilize information about the residuals, as the indices selected by the DEIM and MPE rely on the POD basis, although the indices should represent the residual, which is only a nonlinear function of the POD basis in this context.

Related to the submesh-based approach further work needs to be done to enable other discretization approaches on the submesh. So far, only the Green-Gauss approach for the gradient reconstruction can be applied for the residual evaluation on the submesh. However, more accurate approaches in CODA are based on the Least-squares approach, which relies on a minimum number of cell neighbours. For the outer cells of a solitary stencil this minimum number is undercut, which prevents the application of these approaches. At the same time it reveals potential for further acceleration of the submesh residual evaluation, since so far, the fluxes and gradients are evaluated for every face in the submesh, although they are solely need for specific cells of the full stencil. This produces overhead, which might be reduced. Nevertheless, it would probably require more invasive changes in the CODA algorithms.

Moreover, the hyperreduction could be further developed towards an application for unsteady predictions to utilize the acceleration potential of the consistent hyperreduction, e.g. for the prediction of gusts like in [3]. Although it is still an open question, how this can be realized in the context of CODA's unsteady time stepping scheme, the persuasive results of the conducted investigations should motivate further research on this topic.

Bibliography

- [1] Alen Alexanderian: „Optimal experimental design for infinite-dimensional Bayesian inverse problems governed by PDEs: a review“. In: *Inverse Problems* 37(4) (2021), p. 043001. ISSN: 0266-5611. DOI: 10.1088/1361-6420/abe10c.
- [2] Alexander Vendl: „Projection-based model order reduction for aerodynamic applications“. dissertation. Technical University Carolo Wilhelmina, Braunschweig, 2013.
- [3] Andrea Garbo and Philipp Bekemeyer: „Unsteady physics-based reduced order modeling for large-scale compressible aerodynamic applications“. In: *Computers & Fluids* 239 (2022), p. 105385. ISSN: 00457930.
- [4] Andreas Griewank: *On Automatic Differentiation*. Argonne, Illinois, 1988.
- [5] Anindya Chatterjee: „An introduction to the proper orthogonal decomposition“. In: *Current Science*. Vol. 78, pp. 808–817.
- [6] Azriel Rosenfeld and Avinash C. Kak: *Digital picture processing*. Computer Science and Applied Mathematics. New York, New York, San Francisco, California, and London, England: Academic Press, 1976. ISBN: 9780323148269.
- [7] B. Brands, D. Davydov, J. Mergheim and P. Steinmann: *Reduced-Order Modelling and Homogenisation in Magneto-Mechanics: A Numerical Comparison of Established Hyper-Reduction Methods*. Mathematical and Computational Applications, 2019.
- [8] C. G. Broyden: „The Convergence of a Class of Double-rank Minimization Algorithms: The New Algorithm“. In: *Journal of the Institute of Mathematics and its Applications* 6 (1970), pp. 222–231.
- [9] C. T. Kelley: *Iterative methods for optimization*. Vol. 18. Frontiers in applied mathematics. Philadelphia, Pa: Society for Industrial and Applied Mathematics (SIAM 3600 Market Street Floor 6 Philadelphia PA 19104), 1999. ISBN: 0-89871-433-8.
- [10] K. Carlberg, C. Farhat, J. Cortial, et al.: „The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows“. In: *Journal of Computational Physics* 242 (2013), pp. 623–647. ISSN: 00219991.
- [11] Christian Sabater, Philipp Stürmer and Philipp Bekemeyer: „Fast Predictions of Aircraft Aerodynamics using Deep Learning Techniques“. In: *AIAA AVIATION 2021 FORUM*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2021. ISBN: 978-1-62410-610-1. DOI: 10.2514/6.2021-2549.
- [12] Christopher M. Bishop: *Pattern recognition and machine learning*. Computer science. New York, NY: Springer, 2006. ISBN: 0-387-31073-8.
- [13] Clarence W. Rowley, Tim Colonius and Richard M. Murray: „Model reduction for compressible flows using POD and Galerkin projection“. In: *Physica D: Nonlinear Phenomena* 189 (2003). ISSN: 01672789.
- [14] D. J. Mavriplis: *Multigrid Techniques for unstructured Meshes*. Hampton, VA.
- [15] D. Ryckelynck: „A priori hyperreduction method: an adaptive approach“. In: *Journal of Computational Physics* 202(1) (2005), pp. 346–366. ISSN: 00219991.

- [16] D. Ryckelynck: „Hyper-reduction of mechanical models involving internal variables“. In: *International Journal for Numerical Methods in Engineering* 77(1) (2009), pp. 75–89. ISSN: 00295981.
- [17] David J. Lucia, Philip S. Beran and Walter A. Silva: „Reduced-order modeling: new approaches for computational physics“. In: *Progress in Aerospace Sciences* 40(1-2) (2004), pp. 51–117. ISSN: 03760421. DOI: 10.1016/j.paerosci.2003.12.001.
- [18] Dieter Schwamborn, Thomas Gerhold and Ralf Heinrich: *The DLR TAU-Code: Recent Applications in Research and Industry*. European Conference on Computational Fluid Dynamics, 2006.
- [19] Donald W. Marquardt: „An algorithm for least-squares estimation of nonlinear parameters“. In: *Journal of the Society for Industrial and Applied Mathematics*. Vol. 11.
- [20] DPW-7. 6.10.2022. URL: <https://aiaa-dpw.larc.nasa.gov/>.
- [21] *Experimental data base for computer program assessment: Report of the Fluid Dynamics Panel Working Group 04*. Neuilly-sur-Seine, 1979.
- [22] Francisco Chinesta, Antonio Huerta, Gianluigi Rozza and Karen Willcox: „Model Order Reduction“. In: *Encyclopedia of Computational Mechanics*.
- [23] Gabriel Dimitriu, Răzvan Ștefănescu and Ionel M. Navon: „Comparative numerical analysis using reduced-order modeling strategies for nonlinear large-scale systems“. In: *Journal of Computational and Applied Mathematics* 310 (2017), pp. 32–43. ISSN: 03770427.
- [24] Gene H. Golub: „Some Modified Matrix Eigenvalue Problems“. In: *SIAM Review* 15(2) (1973), pp. 318–334. ISSN: 0036-1445.
- [25] Heng-Bin An, Ju Wen and Tao Feng: „On finite difference approximation of a matrix-vector product in the Jacobian-free Newton–Krylov method“. In: *Journal of Computational and Applied Mathematics* 236(6) (2011), pp. 1399–1409. ISSN: 03770427.
- [26] Hermann Schlichting and Klaus Gersten: *Grenzschicht-Theorie*. 10th ed. Berlin and Heidelberg: Springer, 2006. ISBN: 3-540-23004-1.
- [27] Holger Wendland: *Numerical Linear Algebra*. Cambridge University Press, 2017. ISBN: 9781107147133. DOI: 10.1017/9781107147133.
- [28] I. Huismann, S. Fechter, and T. Leicht: „HyperCODA – Extension of Flow Solver CODA Towards Hypersonic Flows“. In: *New Results in Numerical and Experimental Fluid Mechanics XIII*. Ed. by A. Dillmann, G. Heller, E. Krämer, et al. Vol. 151. Notes on Numerical Fluid Mechanics and Multidisciplinary Design. Cham: Springer International Publishing, 2021, pp. 99–109. ISBN: 978-3-030-79560-3.
- [29] Institute of Aerodynamics and Flow Technology: *Technical Documentation of the DLR TAU-Code Release 2019.1.0*. 2019.
- [30] J. H. Halton: „On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals“. In: *Numerische Mathematik*. Vol. 2, pp. 84–90.
- [31] James E. Gentle: *Matrix algebra: Theory, computations, and applications in statistics*. Springer texts in statistics. New York, NY: Springer, 2007. ISBN: 978-0-387-70873-7.
- [32] James R. Bunch, Christopher P. Nielsen and Danny C. Sorensen: „Rank-One Modification of the Symmetric Eigenproblem“. In: *Numerische Mathematik*. Vol. 31, pp. 31–48.

- [33] Jean Duchon: „Splines minimizing rotation-invariant semi-norms in Sobolev spaces“. In: *Constructive Theory of Functions of Several Variables*. Ed. by A. Dold, B. Eckmann, W. Schempp, et al. Vol. 571. Lecture Notes in Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 1977, pp. 85–100. ISBN: 978-3-540-08069-5.
- [34] Jiri Blazek: *Computational Fluid Dynamics: Principles and Applications*. 3rd ed. Elsevier Ltd., 2015.
- [35] Julien Weiss: *A Tutorial on the Proper Orthogonal Decomposition*. Reston, Virginia, 2019.
- [36] K. Kunisch and S. Volkwein: „Galerkin proper orthogonal decomposition methods for parabolic problems“. In: *Numerische Mathematik* 90(1) (2001), pp. 117–148. ISSN: 0029-599X.
- [37] K. Madsen, H.B. Nielsen and O. Tingleff: *Methods for non-linear least squares problems*. 2004.
- [38] K. Willcox and J. Peraire: „Balanced Model Reduction via the Proper Orthogonal Decomposition“. In: *AIAA Journal* 40(11) (2002), pp. 2323–2330. ISSN: 0001-1452.
- [39] W. Keller and A. Borkowski: „Thin plate spline interpolation“. In: *Journal of Geodesy* 93(9) (2019), pp. 1251–1269. ISSN: 0949-7714.
- [40] Kenneth Levenberg: „A method for the solution of certain non-linear problems in least-squares“. In: *Quarterly of Applied Mathematics*. Vol. 2, pp. 164–168.
- [41] Kevin Carlberg, Matthew Barone and Harbir Antil: „Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction“. In: *Journal of Computational Physics* 330 (2017), pp. 693–734. ISSN: 00219991.
- [42] Lars Reimer: *The FlowSimulator—A software framework for CFD-related multidisciplinary simulations*. NAFEMS European Conference: Computational Fluid Dynamics (CFD) – Beyond the Solve, 2015.
- [43] Lawrence Sirovich: „Turbulence and the dynamics of coherent structures part I: Coherent Structures“. In: *Quarterly of Applied Mathematics*. Vol. 45, pp. 561–571.
- [44] M. Ripepi, M. J. Verveld, N. W. Karcher, T. Franz, M. Abu-Zurayk, S. Görtz and T. M. Kier: „Reduced-order models for aerodynamic applications, loads and MDO“. In: *CEAS Aeronautical Journal* 9(1) (2018), pp. 171–193. ISSN: 1869-5582.
- [45] Maxime Barrault, Yvon Maday, Ngoc Cuong Nguyen and Anthony T. Patera: „An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations“. In: *Comptes Rendus Mathematique* 339(9) (2004), pp. 667–672. ISSN: 1631073X.
- [46] Michael Meinel and Gunnar Ólafur Einarsson: *The FlowSimulator framework for massively parallel CFD applications*. Para 2010 – State of the Art in Scientific and Parallel Computing, 2010.
- [47] Muruhan Rathinam and Linda R. Petzold: „A New Look at Proper Orthogonal Decomposition“. In: *SIAM Journal on Numerical Analysis* 41(5) (2003), pp. 1893–1925. ISSN: 0036-1429.
- [48] N. C. Nguyen, A. T. Patera and J. Peraire: „A ‘best points’ interpolation method for efficient approximation of parametrized functions“. In: *International Journal for Numerical Methods in Engineering* 73(4) (2008), pp. 521–543. ISSN: 00295981.
- [49] *NASA Common Research Model | providing data worldwide*. 3.01.2023. URL: <https://commonresearchmodel.larc.nasa.gov/>.
- [50] P. Bekemeyer, M. Ripepi, R. Heinrich and S. Görtz: „Nonlinear Unsteady Reduced-Order Modeling for Gust-Load Predictions“. In: *AIAA Journal* 57(5) (2019), pp. 1839–1850. ISSN: 0001-1452.

-
- [51] P. L. Roe: „Approximate Riemann solvers, parameter vectors, and difference schemes“. In: *Journal of Computational Physics* 43(2) (1981), pp. 357–372. ISSN: 00219991.
- [52] Patricia Astrid, Siep Weiland, Karen Willcox and Ton Backx: „Missing Point Estimation in Models Described by Proper Orthogonal Decomposition“. In: *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*. Vol. 53.
- [53] Patrick Allen LeGresley: „Application of Proper Orthogonal Decomposition (POD) to Design Decomposition Methods“. PhD thesis. Stanford University, 2006.
- [54] Peter Benner, Mario Ohlberger, Anthony Patera, Gianluigi Rozza and Karsten Urban: *Model Reduction of Parametrized Systems*. Vol. 17. Cham: Springer International Publishing, 2017. ISBN: 978-3-319-58785-1.
- [55] Peter Benner, Serkan Gugercin and Karen Willcox: „A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems“. In: *Siam Review*. Vol. 57, pp. 483–531.
- [56] Peter Deuffhard: *Newton methods for nonlinear problems: Affine invariance and adaptive algorithms*. Vol. 35. Springer series in computational mathematics. Berlin: Springer, 2004. ISBN: 3-540-21099-7.
- [57] Philipp Bekemeyer, Anna Bertram, Derrick A. Hines Chaves, Mateus Dias Ribeiro, Andrea Garbo, Anna Kiener, Christian Sabater Campomanes, Mario Stradtner, Simon Wassing, Markus Widhalm and Stefan Goertz: „Data-Driven Aerodynamic Modeling Using the DLR SMARTy Toolbox“. In: *AIAA AVIATION 2022 Forum*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2022. ISBN: 978-1-62410-635-4.
- [58] R. C. Swanson and E. Turkel: „On Central-Difference and Upwind Schemes“. In: *Journal of Computational Physics* 101 (1992), pp. 292–306. ISSN: 00219991.
- [59] R. Everson and L. Sirovich: „Karhunen–Loève procedure for gappy data“. In: *Journal of the Optical Society of America A* 12(8) (1995), p. 1657. ISSN: 1084-7529.
- [60] Ralf Zimmermann: *A Comprehensive Comparison of Various Algorithms for Efficiently Updating Singular Value Decomposition Based Reduced Order Models: Report of the Institute of Aerodynamics and Flow Technology*. Braunschweig, 2011.
- [61] Ralf Zimmermann and Karen Willcox: „An Accelerated Greedy Missing Point Estimation Procedure“. In: *SIAM Journal on Scientific Computing*. Vol. 38.
- [62] Ralf Zimmermann and Stefan Görtz: „Improved extrapolation of steady turbulent aerodynamics using a non-linear POD-based reduced order model“. In: *The Aeronautical Journal* 116(1184) (2012), pp. 1079–1100. ISSN: 0001-9240.
- [63] Ralf Zimmermann and Stefan Görtz: „Non-linear reduced order models for steady aerodynamics“. In: *Procedia Computer Science* 1(1) (2010), pp. 165–174. ISSN: 18770509.
- [64] Ralf Zimmermann, Alexander Vendl and Stefan Görtz: „Reduced-Order Modeling of Steady Flows Subject to Aerodynamic Constraints“. In: *AIAA Journal* 52(2) (2014), pp. 255–266. ISSN: 0001-1452.
- [65] S. Ashwin Renganathan, Romit Maulik and Jai Ahuja: „Enhanced data efficiency using deep neural networks and Gaussian processes for aerodynamic design optimization“. In: *Aerospace Science and Technology* 111 (2021), p. 106522. ISSN: 12709638.
- [66] S. Volkwein: *Model reduction using Proper Orthogonal Decomposition*.
- [67] S. Y. Shvartsman, C. Theodoropoulos, R. Rico-Martinez, I. G. Kevrekidis, E. S. Titi and T. J. Mountziaris: „Order reduction for nonlinear dynamic models of distributed reacting systems“. In: *Journal of Process Control* 10(2-3) (2000), pp. 177–184. ISSN: 09591524.
- [68] Saifon Chaturantabut and Danny C. Sorensen: „Nonlinear Model Reduction Via Discrete Empirical“. In: *SIAM Journal on Scientific Computing*. Vol. 32.
-

- [69] Sanjay Lall, Petr Krysl and Jerrold E. Marsden: „Structure-preserving model reduction for mechanical systems“. In: *Physica D: Nonlinear Phenomena* 184(1-4) (2003), pp. 304–318. ISSN: 01672789.
- [70] Stefan Langer: „Preconditioned Newton methods to approximate solutions of the Reynolds averaged Navier-Stokes equations“. Habilitation. Kassel: Universität Kassel.
- [71] Steven R. Allmaras, Forrester T. Johnson and Philippe R. Spalart, ed.: *Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model*. 2012.
- [72] T. Franz, R. Zimmermann, S. Görtz and N. Karcher: „Interpolation-based reduced-order modelling for steady transonic flows via manifold learning“. In: *International Journal of Computational Fluid Dynamics* 28(3-4) (2014), pp. 106–121. ISSN: 1061-8562. DOI: 10.1080/10618562.2014.918695.
- [73] T. Lieu, C. Farhat and M. Lesoinne: „Reduced-order fluid/structure modeling of a complete aircraft configuration“. In: *Computer Methods in Applied Mechanics and Engineering* 195(41-43) (2006), pp. 5730–5742. ISSN: 00457825.
- [74] Thomas S. Shores: *Applied linear algebra and matrix analysis*. Undergraduate texts in mathematics. New York, NY: Springer, 2007. ISBN: 978-0-387-33194-2.
- [75] Tobias Leicht, Daniel Vollmer, Jens Jägersküpper, Axel Schwöppe, Ralf Hartmann, J. Fiedler, T. Schlauch: *DLR-Project DIGITAL-X Next Generation CFD Solver "Flucs"*. Deutscher Luft- und Raumfahrtkongress 2016, 2016.
- [76] Tulay Ercan and Costas Papadimitriou: „Optimal Sensor Placement for Reliable Virtual Sensing Using Modal Expansion and Information Theory“. In: *Sensors (Basel, Switzerland)* 21(10) (2021).
- [77] V. Venkatakrishnan: „On the accuracy of limiters and convergence to steady state solutions“. In: *31st Aerospace Sciences Meeting*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 1993.
- [78] E. Yilmaz and B. German: „A Deep Learning Approach to an Airfoil Inverse Design Problem“. In: *2018 Multidisciplinary Analysis and Optimization Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2018. DOI: 10.2514/6.2018-3420.

Appendix A

NLR7301

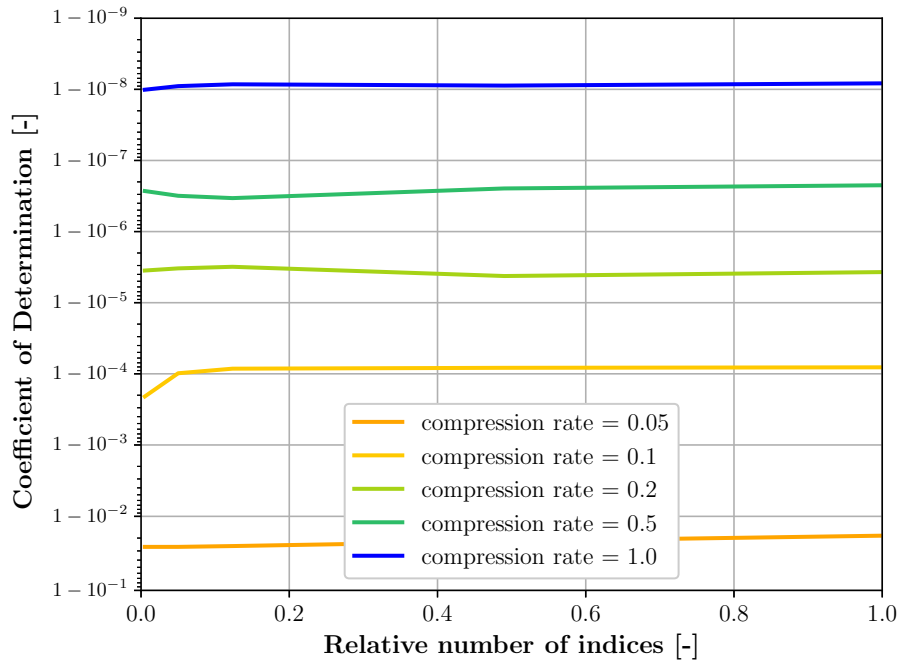


Figure A.1: NLR7301: Coefficient of determination for the predicted density field.

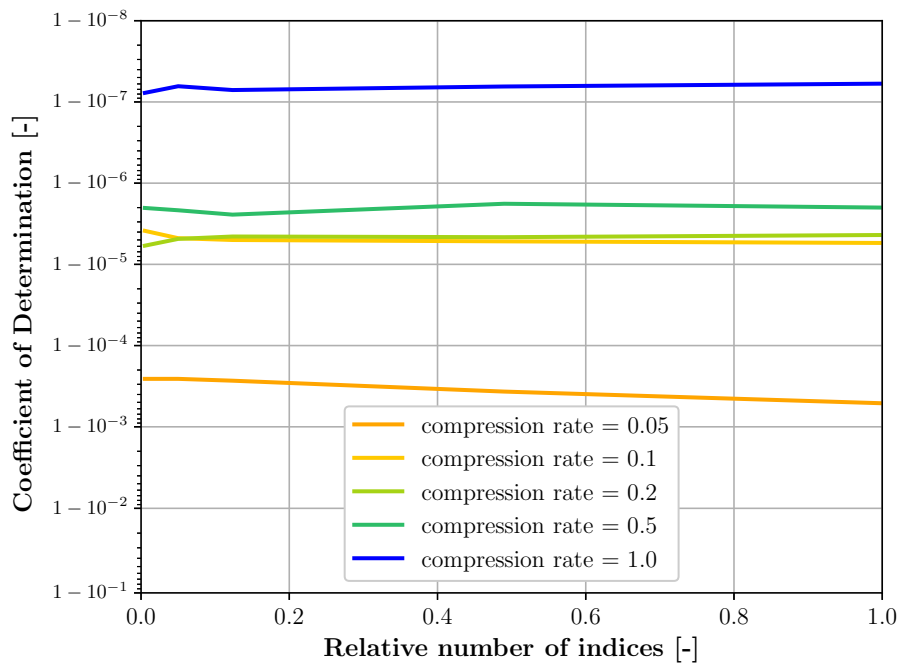


Figure A.2: NLR7301: Coefficient of determination for the predicted field of the momentum in x-direction.

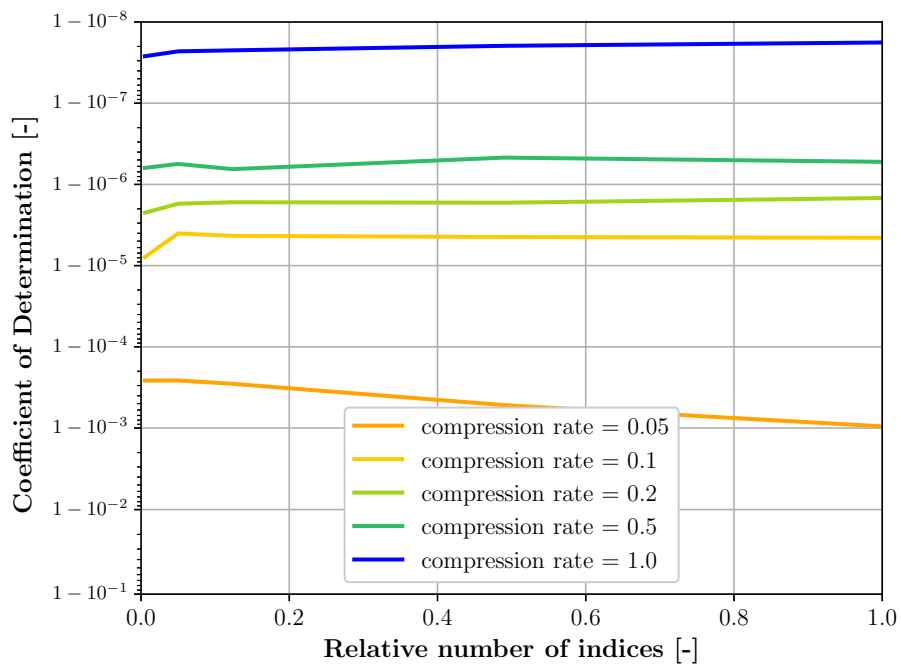


Figure A.3: NLR7301: Coefficient of determination for the predicted field of the momentum in z-direction.

Appendix B

RAE2822

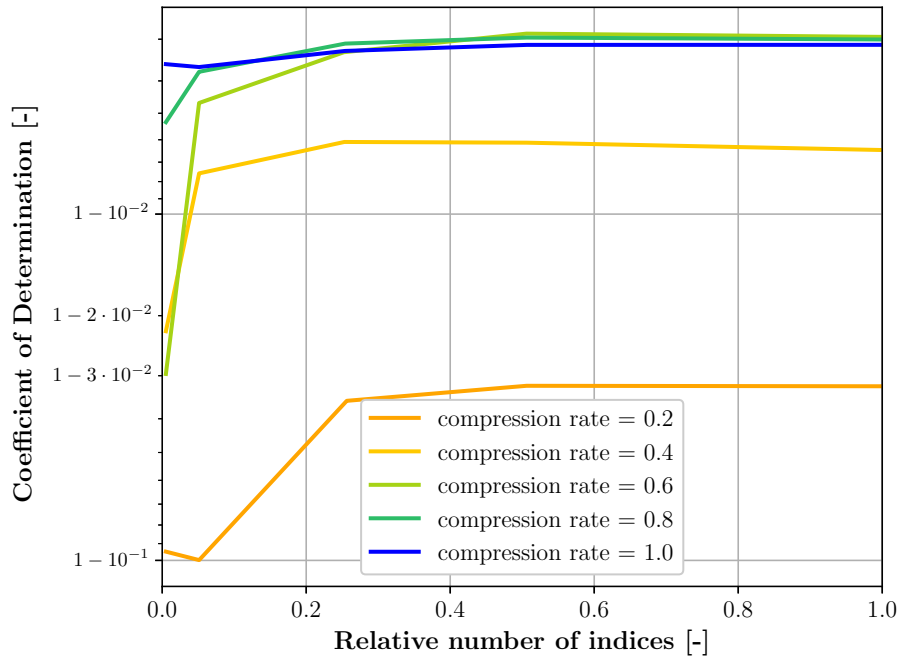


Figure B.1: RAE2822: Coefficient of determination for the predicted density field.

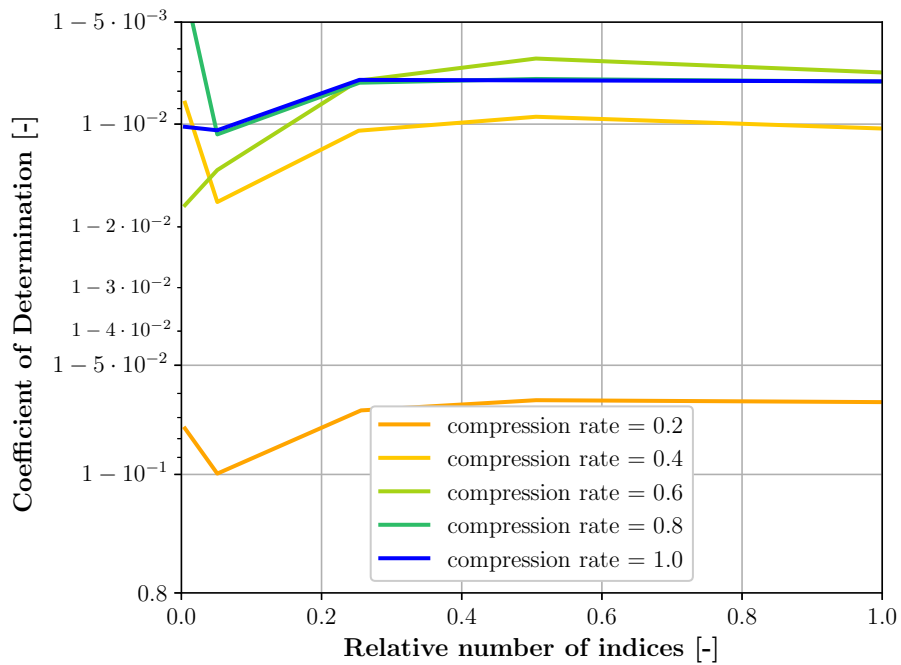


Figure B.2: RAE2822: Coefficient of determination for the predicted field of the momentum in x-direction.

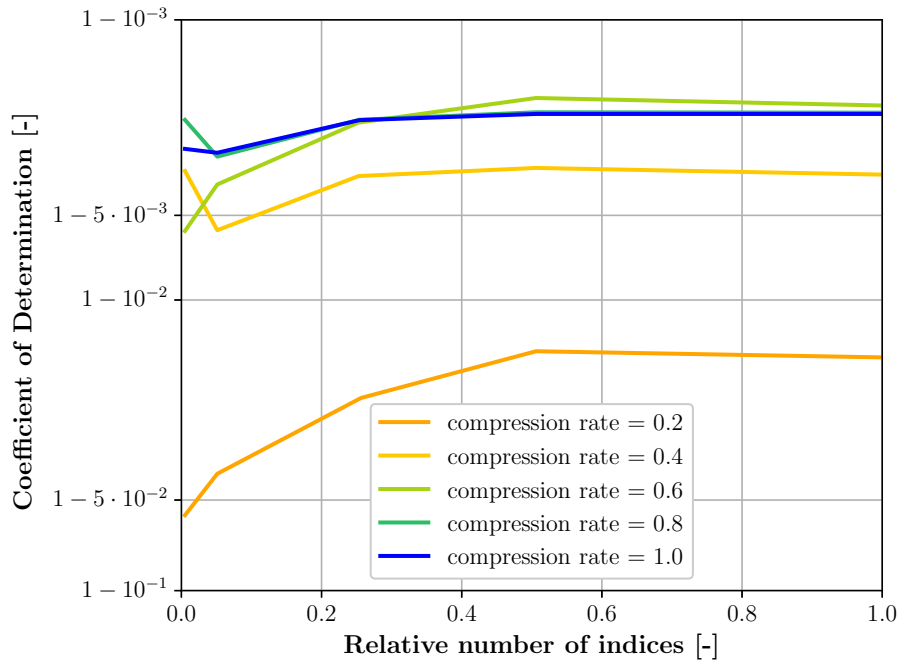


Figure B.3: RAE2822: Coefficient of determination for the predicted field of the momentum in z-direction.

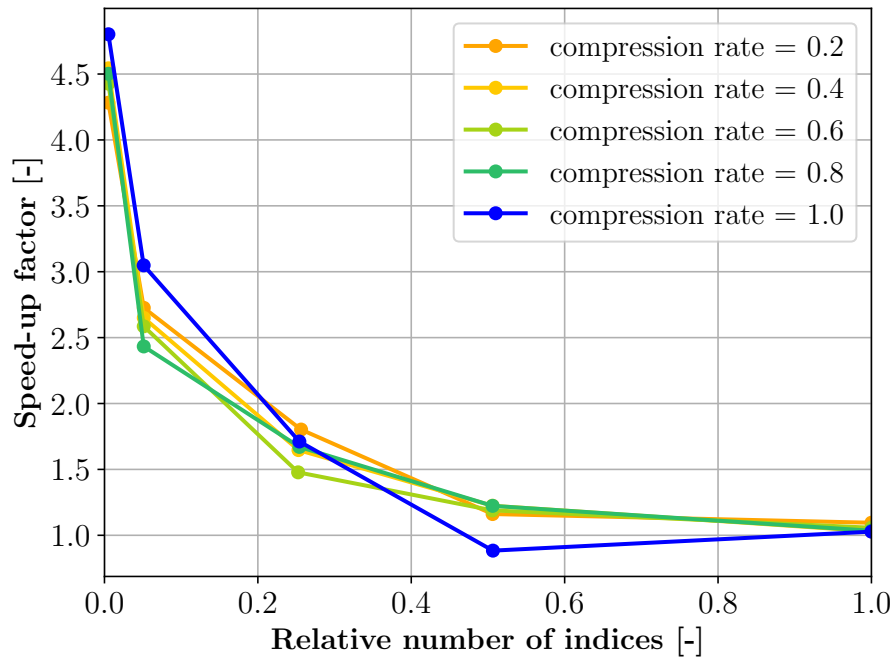


Figure B.4: RAE2822: Speed-up factors for the two hyperreduction implementations.

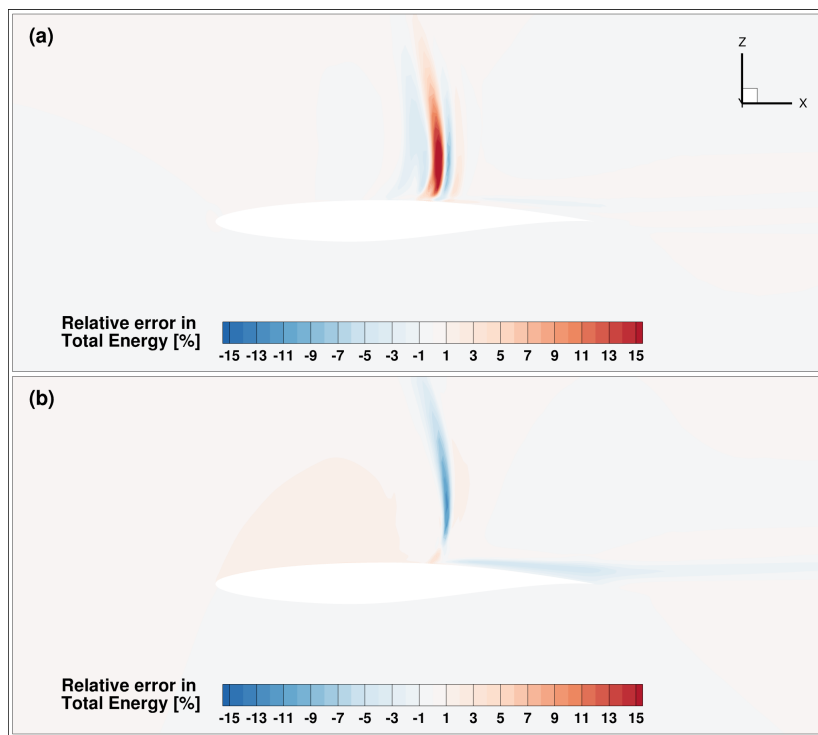


Figure B.5: RAE2822: Relative error of the Total Energy for the POD + TPS and LSQ-ROM predictions.

Appendix C

NASA-CRM

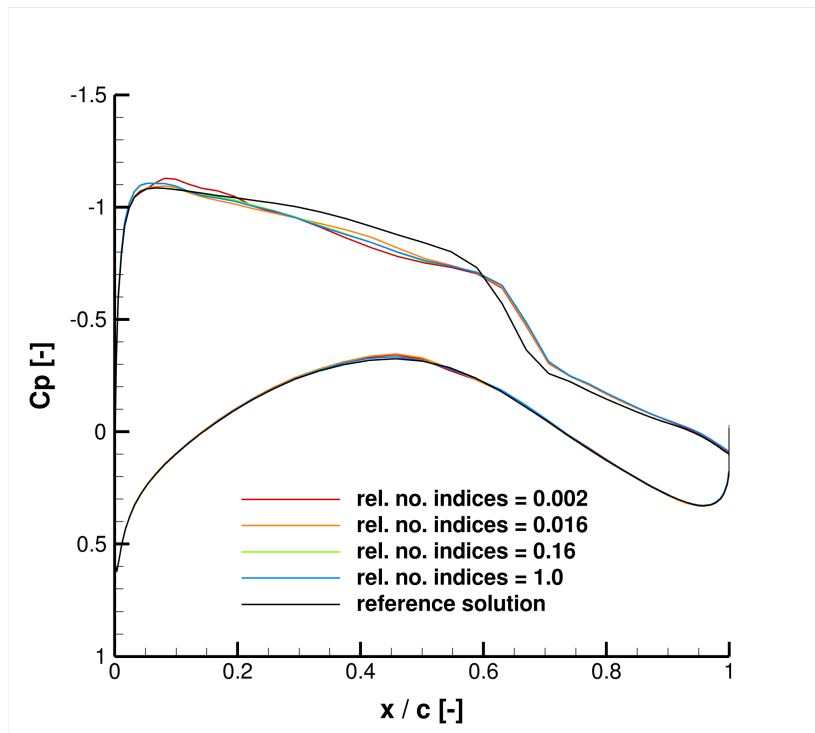


Figure C.1: NASA-CRM: pressure distributions at 25% of the wing span for a LSQ-ROM with a compression rate of 1.0.

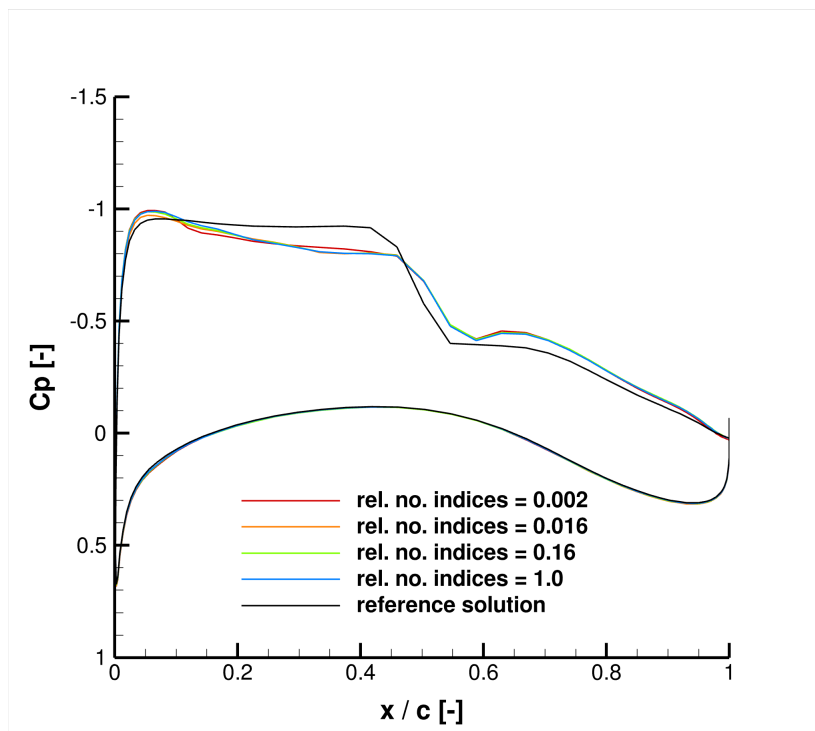


Figure C.2: NASA-CRM: pressure distributions at 75% of the wing span for a LSQ-ROM with a compression rate of 1.0.

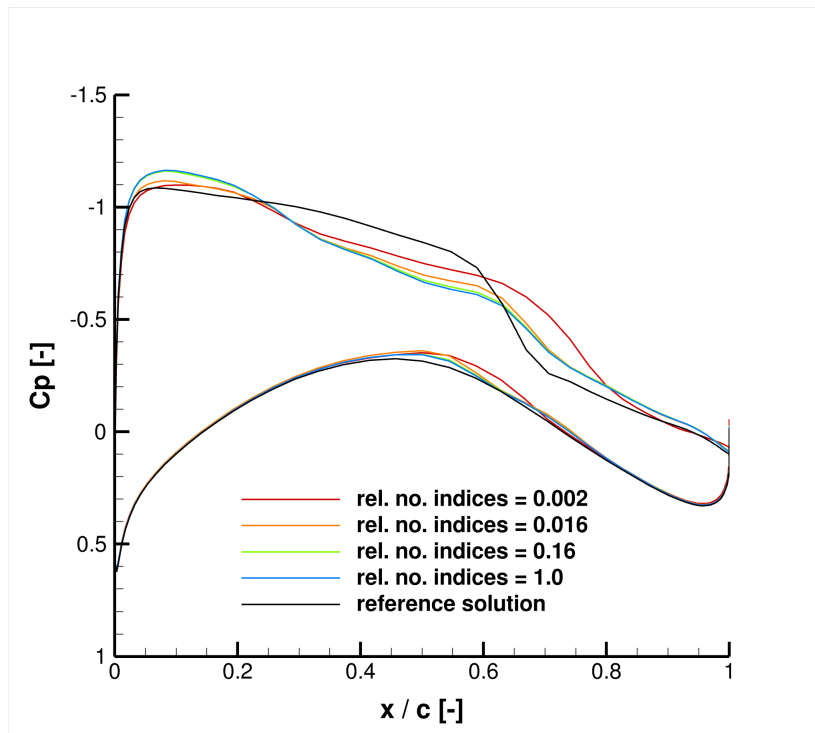


Figure C.3: NASA-CRM: pressure distributions at 25% of the wing span for a LSQ-ROM with a compression rate of 0.5.

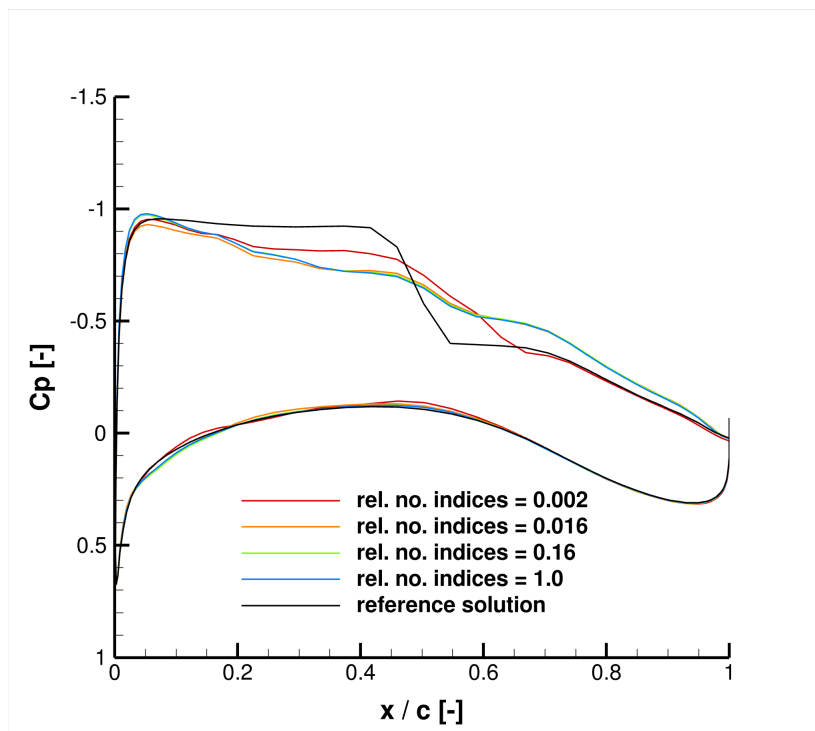


Figure C.4: NASA-CRM: pressure distributions at 75% of the wing span for a LSQ-ROM with a compression rate of 0.5.

Eidesstattliche Erklärung

Hiermit erkläre ich, Cedric Fallet, geb. am 16.09.1996, an Eides statt, die vorliegende Masterarbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben.

C. Fallet

Braunschweig, 13.03.2023
