UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Complexity reduction in parametric flow problems via Nonintrusive Proper Generalised Decomposition in OpenFOAM

by
Vasileios Tsiolakis

ADVERTIMENT La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del repositori institucional UPCommons (http://upcommons.upc.edu/tesis) i el repositori cooperatiu TDX (http://www.tdx.cat/) ha estat autoritzada pels titulars dels drets de propietat intel·lectual **únicament per a usos privats** emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei UPCommons o TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a UPCommons (*framing*). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del repositorio institucional UPCommons (http://upcommons.upc.edu/tesis) y el repositorio cooperativo TDR (http://www.tdx.cat/?locale-attribute=es) ha sido autorizada por los titulares de los derechos de propiedad intelectual **únicamente para usos privados enmarcados** en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio UPCommons No se autoriza la presentación de su contenido en una ventana o marco ajeno a UPCommons (*framing)*. Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the institutional repository UPCommons (http://upcommons.upc.edu/tesis) and the cooperative repository TDX (http://www.tdx.cat/?locale-attribute=en) has been authorized by the titular of the intellectual property rights **only for private uses** placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading nor availability from a site foreign to the UPCommons service. Introducing its content in a window or frame foreign to the UPCommons service is not authorized (framing). These rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.

Complexity reduction in parametric flow problems via Nonintrusive Proper Generalised Decomposition in OpenFOAM

by

Vasileios Tsiolakis

Doctoral Thesis

Universitat Politècnica de Catalunya

Swansea University

November 2019

Disclaimer

The results, opinions and conclusions of this dissertation are not necessarily those of the Volkswagen AG.

Die Ergebnisse, Meinungen und Schlüsse dieser Dissertation sind nicht notwendigerweise die der Volkswagen AG.

Abstract

The present thesis explores the viability of the proper generalised decomposition (PGD) as a tool for parametric studies in a daily industrial environment. Starting from the equations modelling incompressible flows, the separated formulation of the equations, the development of a parametric solver, the implementation in a commercial computational fluid dynamics (CFD) software, OpenFOAM, and a numerical validation are presented.

The parametrised Stokes and Oseen flows are used as an initial step to test the applicability of the PGD to flow problems. The rationale for the construction of a separable approximation is described and implemented in OpenFOAM. For the numerical validation of the developed strategy analytical test cases are solved. Then, the parametrised steady laminar incompressible Navier-Stokes equations are considered. The nonintrusive implementation of PGD in OpenFOAM is formulated, focusing on the seamless integration of a reduced order model (ROM) in the framework of an industrially validated CFD software. The proposed strategy exploits classical solution strategies in OpenFOAM to solve the PGD spatial iteration, while the parametric one is solved via a collocation approach. Such nonintrusiveness represents an important step towards the industrialisation of PGD-based approaches. The capabilities of the methodology are tested by applying it to benchmark tests in the literature and solving a parametrised flow control problem in a realistic geometry of interest for the automotive industry.

Finally, the PGD framework is extended to turbulent Navier-Stokes problems. The separable form of an industrially popular turbulence model, namely Spalart-Allmaras model, is formulated and a PGD strategy for the construction of a parametric turbulent eddy viscosity is devised. Different implementation possibilities in the nonintrusive PGD for parametrised Navier-Stokes equations are explored and the proposed strategy is applied to well-documented turbulent flow control benchmark cases in both two and three dimensions.

Keywords: Reduced order models, proper generalised decomposition, parametrised incompressible flows, parametrised turbulent flows, incompressible Navier-Stokes, Spalart-Allmaras turbulence model, OpenFOAM, nonintrusive, finite volume, parametrised flow-control

Acknowledgements

I would like to thank all those who have contributed to the completion of this thesis. I would like to thank my thesis advisors, by order of seniority, Antonio Huerta, Professor Polytechnic University of Catalonia, Carsten Othmer, Project Leader Volkswagen Group Innovation, Ruben Sevilla, Professor Swansea University and Matteo Giacomini, Postdoctoral Fellow Polytechnic University of Catalonia, for giving me the opportunity to carry out such challenging research and investing time and effort throughout these past few years.

I am sincerely thankful to the thesis committe, Oubay Hassan, Professor Swansea University, Pedro Diez, Professor Polytechnic University of Catalonia, Juan J. Alonso, Professor Stanford University, Gianluigi Rozza, Professor Scuola Internazionale Superiore di Studi Avanzati and Xesús Nogueira, University of A Coruña.

I am grateful my colleagues in Volkswagen AG, Group Innovation, Simulation Methods for creating a friendly and enjoyable working environment. I would like to thank the AdMoRe ITN for the chance to interact and collaborate with brilliant scientists as well as the rest of the Early Stage Researchers of the network, with whom I shared this unique experience. Last but not least, I would like to thank my family for their moral support and encouragement along the duration of the thesis.

This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Actions (Grant agreement No. 675919) that financed the Ph.D. fellowship.

Contents

Chapter 1

Introduction

In the daily industrial environment computational fluid dynamics (CFD) has become a key component of the design pipeline. With robustness highly valued, simulations of incompressible flows are performed on a daily basis to solve different problems both in automotive and aeronautical industries. The most widely spread CFD methodology is the finite volume (FV) method [67, 108, 73, 7, 38, 100]. Using this technique, numerically evaluated quantities of interest (e.g. drag and lift) have proved to match reasonably well experimental results while recent efforts towards better parallelisation [110, 43] and implementation of higher-order schemes [26, 76, 75, 93, 42] have yielded further improvement in both the efficiency and accuracy of the technique.

Depending on the nature of the problem, the automotive industry has adopted a range of methodologies. For internal flow simulations the Reynolds Averaged Navier Stokes (RANS) method is employed, as is for external aerodynamics when turnaround time is of the essence. Otherwise, Detached Eddy Simulation [103, 59] (DES), Improved Delayed Detached Eddy simulation [109, 112] (IDDES) and lattice Boltzmann [17, 97] methods have their own use-cases in the computation of vehicle external aerodynamics. The turbulence models most widely used are the $k - \omega$ SST [71], applicable to both internal and external aerodynamics and the Spalart-Allmaras turbulence model [102] (SA), used solely for external aerodynamics.

In recent years, the optimisation pipeline of the automotive industry has seen the introduction of tools based on the continuous adjoint method [79]. Such tools simplify the aerodynamic optimisation process by computing sensitivity maps, quantifying the effect of the design variables (typically shape) to the quantities of interest (cost functional of drag, lift, etc.). Adjoint solvers have been developed for both RANS and DES [60, 81, 82], as well as lattice Boltzmann [18], making their use in the industrial environment as seamless as possible.

1.1 Problem statement

Despite the advanced tools and the maturity of the method, the design and optimisation cycles in a production environment are still characterised by the large amount of configurations of the same problem that require testing. Varying boundary conditions, physical properties of the fluid and geometry of the domain within a prescribed range are typical examples of interest. In this context, such parameters act as extra-coordinates of a high-dimensional partial differential equation (PDE). The computational cost of such parametric studies currently represents the major limitation to the application of simulation-based engineering techniques in a daily industrial environment. It is well-known that the computational complexity of approximating the PDEs describing the problems under analysis increases exponentially with the number of parameters considered.

Dimensionality reduction via active subspaces [80] has been considered for the simplification of parametric studies. Reduced order models [19], including reduced basis (RB) [6, 48, 47, 94, 16, 57, 70], proper orthogonal decomposition (POD) [58, 62, 8, 10, 5, 65, 11] and hierarchical model reduction (HiMod) [89, 90, 2, 52], have also been proposed to reduce the computational burden of parametric analysis and optimisation for several physical problems, including incompressible flows. The aforementioned techniques rely on an *a posteriori* reduction based on snapshots computed as solutions of the full-order model for different values of the parameters under analysis. An alternative approach is represented by PGD [3, 20, 21, 22, 23]. This method features an *a priori* reduction, using a separable approximation of the solution, which depends explicitly on the parameters under analysis. In this context, during an offline phase, a reduced basis is constructed with no *a priori* knowledge of the solution, whereas efficient online evaluations of the generalised solution are performed by simple interpolation in the parametric space. See [33, 4, 114, 72, 101, 30, 56] for several applications of PGD to different physical problems. Moreover, special attention has been devoted to the study of convective phenomena and convection stabilisation using PGD-based separated representations [45, 44]. Of course, parametric solutions involving the geometry of the domain, which are beyond the contributions of this thesis, are of great interest in an industrial environment for the speed up of shape design and optimisation procedures. A separated solution of a PDE involving parametrised geometries was first proposed in [4] and further investigated in [114, 101, 30]. Such technique can be extended to the PGD algorithm for OpenFOAM proposed in this work via the definition of two nested computational meshes, one with the FV cells used for the flow computation and one featuring macrocells for the description of the geometry. In the context of geometrically parametrised problems, it is worth mentioning that a seamless integration of PGD and computer aided design (CAD) tools has been recently proposed in [99] to exploit the potential of parametric representation of curves and surfaces via the control points of nonuniform rational B-splines (NURBS).

In the context of flow problems, model reduction techniques based on Galerkin projection have been extensively studied in the literature [27, 69, 113]. In this framework, several strategies have been proposed to construct the trial basis, using POD [55], RB [53] or the empirical interpolation method [32]. Concerning incompressible Navier-Stokes equations, in [68, 104] supremiser stabilisations techniques have been investigated to couple the FV method with POD to solve parametrised turbulent flow problems. Coupled strategies, where the turbulent Navier-Stokes are approximated using classical POD with data-driven models and non-intrusive reduced order modelling (NIROM) with machine learning techniques are proposed in [54, 111]. In [13, 15, 12, 66], alternative non-linear projection methods have been discussed. More recently, special attention has also been devoted to FV-based structure-preserving ROMs for conservation laws [14].

Another key aspect for the application of simulation-based techniques to industrial problems is the capability of the proposed methods to provide verified and certified results. This problem has been classically treated by equipping numerical methods with reliable and fully-computable *a posteriori* error estimators using equilibrated fluxes [29, 36, 37] and flux-free approaches [85, 24, 84] to control the error of the solution as well as of quantities of interest [78, 83, 86, 63, 1, 74, 41, 40]. Nonetheless, these approaches require intrusive modifications of existing computational libraries and may not be feasible in the context of commercial software. Hence, although the effort of the academic community in this direction, such solutions have not been successfully and widely integrated in codes utilised by the industry. More recently, to circumvent this issue, great effort has been devoted to nonintrusive implementations in which novel numerical methodologies are externally coupled to existing commercial and open-source software used in industry on a daily basis. Some contributions in this direction have been successfully proposed coupling PGD with Abaqus® for mechanical problems [115] and PGD with SAMTECH® for shape optimisation problems [25]. For flow problems, the coupling of POD and OpenFOAM has been discussed in [9, 104]. This work explores the viability of a nonintrusive integration of the PGD framework in OpenFOAM for the solution of parametrised incompressible Navier-Stokes problems in both the laminar and turbulent regime. The

RANS method offers a convenient first step towards an easily adopted PGD scheme in the industry, seeing how it is used in the majority of simulations. Similarly, the Spalart-Allmaras turbulence model, being a widely spread single equation model, presents both a relevant and convenient entry point for parametrised turbulent problems.

1.2 Contributions and outline

The main contributions of this thesis are:

1. Implementation of a PGD solver for parametric Stokes and Oseen problems in OpenFOAM.

2. The *nonintrusive PGD* method for the solution of parametrised flow problems. This method offers a straightforward and flexible implementation procedure in industrial CFD software packages and brings the otherwise complex PGD solver closer to the industry.

3. The implementation of the PGD method in OpenFOAM. The resulting `pgdFoam` algorithm relies on OpenFOAM [107] CFD libraries and exploits the incompressible flow solver `simpleFoam` for the spatial iteration of the alternating direction scheme.

4. Application of the *nonintrusive PGD* to flow control problem and realistic geometries of industrial interest.

5. The PGD solver for the Spalart-Allmaras turbulence model and the method to solving parametrised Navier-Stokes problems residing in the

turbulent regime. Developing a PGD methodology to approximate parameter dependent turbulent viscosity and including it in the `pgdFoam` algorithm is a significant milestone towards PGD applications to parametric studies of industrial flows.

6. The approximation of convection-dominated parametric Navier-Stokes flow problems with PGD.

The thesis, not including the introduction and conclusions, is organised in three main chapters summarised below. The first chapter presents a first study of the viability of PGD for the solution of parametrised flow problems using OpenFOAM. More precisely, Chapter 2 recalls the flow problem and introduces the PGD rationale. The strategy for the construction of a separable approximation of velocity and pressure for parametrised Stokes flow and linearised Navier-Stokes (Oseen) flow is described. This chapter concludes with numerical validation of the aforementioned strategy. Analytical test cases are used to verify the theoretical order of convergence.

Chapter 3 introduces the concept of the 'nonintrusive PGD' method. Using the parametrised laminar Navier-Stokes flow, the strategy to assembling problems solvable using standard OpenFOAM solvers is described. Other alternatives are also discussed. The second part of the chapter focuses on applications of the proposed nonintrusive PGD to both two- and three-dimensional cases. The method's validity is tested with a benchmark case, before transitioning to parametrised flow control problems.

The final chapter considers the application of PGD in more realistic scenarios, where the parameters affect turbulence. First, in Chapter 4 the PGD formulation and strategy to solving the parametrised Spalart-Allmaras turbulence model is recalled. The communication between this PGD structure and the nonintrusive PGD is analysed, proposing a methodology to tackle the parametrised turbulent Navier-Stokes problem. Then, the proposed method is applied to flow control problems of significantly high Reynolds number (from $100,000$ to $1,000,000$) to verify that the PGD approximation can recover the flow structure changes that the full-order solver predicts.

Chapter 2

The proper generalised decomposition for linear flow problems

This chapter aims at introducing the PGD rationale for flow problems and exploring its applicability in OpenFOAM. Initially, a general form of the flow governing equations is recalled, along with a brief introduction of the finite volume solver chosen. Focusing on linear problems, the construction of a separable approximation is detailed in steps. Following the PGD rationale, the parametrised Stokes flow problem is formulated and the PGD strategy for the construction of a separable approximation is recalled. Next, a linearised convection term is considered, yielding the parametrised Oseen flow problem and the PGD strategy is adapted accordingly. Finally, test cases are used for the numerical validation of the implemented PGD methodology.

2.1 A cell-centred finite volume approximation using OpenFOAM

In this section, a linearised form of the steady Navier-Stokes equations for the simulation of incompressible viscous laminar flows in d spatial dimensions is recalled.

Let $\Omega \subset \mathbb{R}^d$, $\partial\Omega{=}\Gamma_D \cup \Gamma_N$ be an open bounded domain with disjoint Dirichlet, Γ_D, and Neumann, Γ_N, boundaries. The flow problem under analysis consists of computing the velocity field $\boldsymbol{u}$ and the pressure p such that

$$\begin{cases} \boldsymbol{\nabla}\cdot(\boldsymbol{u}\otimes\boldsymbol{a}) - \boldsymbol{\nabla}\cdot(\nu\boldsymbol{\nabla}\boldsymbol{u}) + \boldsymbol{\nabla}p = \boldsymbol{s} & \text{in } \Omega, \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 & \text{in } \Omega, \\ \boldsymbol{u} = \boldsymbol{u}_D & \text{on } \Gamma_D, \\ \boldsymbol{n}\cdot(\nu\boldsymbol{\nabla}\boldsymbol{u}{-}p\mathbf{I}_d) = \boldsymbol{t} & \text{on } \Gamma_N, \end{cases} \tag{2.1}$$

where the first equation describes the balance of momentum and the second one the conservation of mass. In Equation (2.1), $\boldsymbol{a}$ is a divergence-free velocity field, $\boldsymbol{s}$ represents a volumetric source term, $\nu{>}0$ is the dynamic viscosity and $\mathbf{I}_d$ is the $d{\times}d$ identity matrix. On the Dirichlet boundary Γ_D, the value $\boldsymbol{u}_D$ of the velocity is imposed, whereas on Γ_N the pseudo-traction $\boldsymbol{t}$ is applied. From the modelling point of view, inlet surfaces and physical walls are described as Dirichlet boundaries with an imposed entering velocity profile and a homogeneous datum, respectively, whereas outlet surfaces feature homogeneous Neumann boundary conditions. For the sake of simplicity and without loss of generality, Γ_N is henceforth assumed to be an outlet boundary, that is a null $\boldsymbol{t}$ is considered.

Based on the choice of the velocity field $\boldsymbol{a}$, two linear flow regimes can be recovered

- $\boldsymbol{a} = \boldsymbol{0}$ resulting in the Stokes flow.
- $\boldsymbol{a} = \boldsymbol{u}_0$ where $\boldsymbol{u}_0$ is an a priori known divergence-free velocity field, resulting in the Oseen flow.

The formulation of a FV scheme for the incompressible linearised Navier-Stokes equations is briefly recalled to introduce the notation needed for the high-dimensional parametrised problem developed in all following Sections.

The domain Ω is partitioned in N nonoverlapping cells V_i, $i{=}1,\dots,N$ such that $\Omega{:=}\bigcup_{i=1}^{N} V_i$ and $V_i{\cap}V_j{=}\emptyset$, for $i{\neq}j$. The FV discretisation is constructed starting from the integral formulation of Equation (2.1), namely find $(\boldsymbol{u},p)$, constant on each cell V_i, $i{=}1,\dots,N$, such that $\boldsymbol{u}{=}\boldsymbol{u}_D$ on Γ_D and it holds

$$\begin{cases} \displaystyle\int_{V_i} \boldsymbol{\nabla}{\cdot}(\boldsymbol{u}{\otimes}\boldsymbol{a})\, dV - \int_{V_i} \boldsymbol{\nabla}{\cdot}(\nu\boldsymbol{\nabla}\boldsymbol{u})\, dV + \int_{V_i} \boldsymbol{\nabla} p\, dV = \int_{V_i} \boldsymbol{s}\, dV, \\ \displaystyle\int_{V_i} \boldsymbol{\nabla}{\cdot}\boldsymbol{u}\, dV = 0. \end{cases} \tag{2.2}$$

OpenFOAM implements a cell-centred FV rationale in which piecewise constant approximations are sought for velocity and pressure in each cell of the computational mesh and the degrees of freedom of the discretised problem are located at the centroid of each finite volume. Employing Gauss's theorem, the integrals in Equation (2.2) are rewritten in terms of fluxes over the boundaries of the cells and approximated using classical central differencing schemes [7, 38]. The resulting solution strategy relies on the SIMPLE algorithm which belongs to the family of fractional-step projection methods [87, 31]. A brief description of this method for the Navier-Stokes problem, the handling of the nonlinearity in the convection term, as well as typical discretisation is provided in B.

Note that this thesis aims to develop a nonintrusive PGD strategy to approximate parametrised Navier-Stokes flows using OpenFOAM. For the solution of the Stokes and Oseen flows, modified versions of the SIMPLE solver were introduced in OpenFOAM. By definition the corresponding PGD strategies are intrusive. A more detailed view on the Stokes and Oseen solver adaptations can be found in Appendix B.1 and B.2, respectively.

2.2 The proper generalised decomposition (PGD) rationale

Consider now the case in which the user-prescribed data in Equation (2.1), i.e. the viscosity coefficient, the source term and the boundary conditions, depend on a set of parameters $\boldsymbol{\mu} \in \boldsymbol{\mathcal{I}} \subset \mathbb{R}^M$, with M being the number of parameters. More presicely, the set $\boldsymbol{\mathcal{I}}$ describing the range of admissible parameters can be defined as the Cartesian product of the domains of the M parameters, namely, $\boldsymbol{\mathcal{I}}{:=}\mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_M$ with $\mu_i \in \mathcal{I}_i$ for $i{=}1,\dots,M$. Within this context, $\boldsymbol{\mu}$ is treated as a set of additional independent variables, or parametric coordinates, instead of problem parameters. For the purpose of discretisation, each interval $\mathcal{I}_i$ is subdivided in N_μ subintervals. The unknown pair $(\boldsymbol{u}, p)$ is thus sought in a high-dimensional space described by the independent variables $(\boldsymbol{x}, \boldsymbol{\mu}) \in \Omega\times\boldsymbol{\mathcal{I}}$ and fulfils the following parametrised equations on each cell V_i, $i{=}1,\dots,N$

$$\begin{cases} \displaystyle\int_{\boldsymbol{\mathcal{I}}}\int_{V_i} \boldsymbol{\nabla}\cdot(\boldsymbol{u}\otimes\boldsymbol{a})\, dV\, d\boldsymbol{\mathcal{I}} - \int_{\boldsymbol{\mathcal{I}}}\int_{V_i} \boldsymbol{\nabla}\cdot(\nu\boldsymbol{\nabla}\boldsymbol{u})\, dV\, d\boldsymbol{\mathcal{I}} \\ \displaystyle\qquad\qquad + \int_{\boldsymbol{\mathcal{I}}}\int_{V_i} \boldsymbol{\nabla} p\, dV\, d\boldsymbol{\mathcal{I}} = \int_{\boldsymbol{\mathcal{I}}}\int_{V_i} \boldsymbol{s}\, dV\, d\boldsymbol{\mathcal{I}}, \\ \displaystyle\qquad\qquad \int_{\boldsymbol{\mathcal{I}}}\int_{V_i} \boldsymbol{\nabla}\cdot\boldsymbol{u}\, dV\, d\boldsymbol{\mathcal{I}} = 0. \end{cases} \tag{2.3}$$

In the following sections, a step-by-step approach will be used to recall the rationale for the construction of a separated solution for parametrised flow problems. The Stokes flow system of equations will be used to introduce the main components of the strategy, while the parametrised Oseen flow problem will be expressed as an adaptation of the Stokes PGD.

The PGD constructs an approximation $(\boldsymbol{u}^n_{\texttt{PGD}}, p^n_{\texttt{PGD}})$ of the solution $(\boldsymbol{u}, p)$ of Equation (2.3) in terms of a sum of n separable functions, or modes. Each mode is the product of functions depending solely on one of the arguments $\boldsymbol{x}, \mu_1, \mu_2, \dots, \mu_M$. For the sake of readability and without loss of generality, only the spatial coordinates $\boldsymbol{x}$ and the parametric ones $\boldsymbol{\mu}$ are henceforth separated.

Following [30], the so-called *single parameter approximation* is detailed. That is, for each mode, a unique scalar parametric function $\phi(\boldsymbol{\mu})$ is considered for all the variables and the resulting separated form of the unknowns is

$$\begin{cases} \boldsymbol{u}^n_{\text{PGD}}(\boldsymbol{x},\boldsymbol{\mu}) = \boldsymbol{u}^{n-1}_{\text{PGD}}(\boldsymbol{x},\boldsymbol{\mu}) + \sigma^n_u\, \boldsymbol{f}^n_u(\boldsymbol{x})\, \phi^n(\boldsymbol{\mu}), \\ p^n_{\text{PGD}}(\boldsymbol{x},\boldsymbol{\mu}) = p^{n-1}_{\text{PGD}}(\boldsymbol{x},\boldsymbol{\mu}) + \sigma^n_p\, f^n_p(\boldsymbol{x})\, \phi^n(\boldsymbol{\mu}). \end{cases} \tag{2.4}$$

where the superindex n denotes the, *a priori* unknown, number of terms in the PGD expansion and the positive scalar coefficients σ^n_u and σ^n_p represent the amplitude of the n-th mode for velocity and pressure, respectively. These coefficients are obtained normalising the modal functions, namely

$$\sigma^n_u := \|\boldsymbol{f}^n_u\| \text{ and } \sigma^n_p := \|f^n_p\|,$$

with $\|\phi^n\| = 1$. Appropriate user-defined norms on the spatial and parametric domains are introduced for each function. For all the simulations in Section 3.3, the $\mathcal{L}_2$ norm has been considered for normalisation.

Remark 1. *The normalisation coefficients play a critical role in checking the convergence of the PGD algorithm and may be used as quantitative stopping criterion in the PGD enrichment procedure described in Section 2.3.*

For a discussion on alternative formulations of the separation in Equation (2.4), involving both scalar and vector-valued parametric functions the interested reader is referred to [30]. Henceforth and except in case of ambiguity, the dependence of the modes on $\boldsymbol{x}$ and $\boldsymbol{\mu}$ is omitted.

Considering a linearised approach to compute each new mode, Equation (2.4) can be rewritten as the following *predictor-corrector single param-*

eter approximation

$$\begin{cases} \boldsymbol{u}^n_{\text{PGD}} = \widetilde{\boldsymbol{u}}^n_{\text{PGD}} + \sigma^n_u \delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}} = \boldsymbol{u}^{n-1}_{\text{PGD}} + \sigma^n_u \boldsymbol{f}^n_u \phi^n + \sigma^n_u \delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}}, \\ p^n_{\text{PGD}} = \widetilde{p}^n_{\text{PGD}} + \sigma^n_p \delta\widetilde{p}^n_{\text{PGD}} \ = p^{n-1}_{\text{PGD}} + \sigma^n_p f^n_p \phi^n + \sigma^n_p \delta\widetilde{p}^n_{\text{PGD}}, \end{cases} \tag{2.5}$$

where $\widetilde{\boldsymbol{u}}^n_{\text{PGD}} {:=} \boldsymbol{u}^{n-1}_{\text{PGD}} + \sigma^n_u \boldsymbol{f}^n_u \phi^n$ and $\widetilde{p}^n_{\text{PGD}} {:=} p^{n-1}_{\text{PGD}} + \sigma^n_p f^n_p \phi^n$ account for the $n{-}1$ previously computed terms and a prediction of the current mode. More precisely, $(\sigma^n_u \boldsymbol{f}^n_u \phi^n, \sigma^n_p f^n_p \phi^n)$ play the role of predictors in the computation of the n-th mode, whereas $(\sigma^n_u \delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}}, \sigma^n_p \delta\widetilde{p}^n_{\text{PGD}})$ are the corresponding correctors featuring the variations Δ in the spatial and parametric functions, namely

$$\begin{cases} \delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}} := \Delta \boldsymbol{f}_u \phi^n + \boldsymbol{f}^n_u \Delta\phi + \Delta \boldsymbol{f}_u \Delta\phi, \\ \delta\widetilde{p}^n_{\text{PGD}} := \Delta f_p \phi^n + f^n_p \Delta\phi + \Delta f_p \Delta\phi. \end{cases} \tag{2.6}$$

Note that the last term in Equation (2.6) represents a high-order variation which is henceforth neglected. As for the classical *single parameter approximation*, σ^n_u and σ^n_p represent the amplitudes of the n-th velocity and pressure modes. That is, setting $\|\phi^n + \Delta\phi\| = 1$, they are defined as

$$\sigma^n_u := \|\boldsymbol{f}^n_u + \Delta \boldsymbol{f}_u\| \text{ and } \sigma^n_p := \|f^n_p + \Delta f_p\|.$$

2.3 Predictor-corrector alternating direction scheme for parametrised Stokes flow

In order to compute $(\boldsymbol{u}^n_{\texttt{PGD}}, p^n_{\texttt{PGD}})$ in Equation (2.5), a greedy algorithm is implemented. The first PGD mode $(\boldsymbol{u}^0_{\texttt{PGD}}, p^0_{\texttt{PGD}})$ is arbitrarily chosen to fulfil the Dirichlet boundary conditions of the problem and the n-th mode is successively computed assuming that term $n-1$ is available [22, 23]. Some variations of this strategy based on Arnoldi-type iterations have been investigated in [77, 105]. In this section, the parametrised Stokes flow system of equations is introduced and the concept behind the alternating direction scheme used to compute the PGD modes is described. A key assumption for the application of this method is the separability of the data. For the sake of simplicity and without any loss of generality, the separated form of the viscosity coefficient, see e.g. [114], is reported

$$\nu(\boldsymbol{x}, \boldsymbol{\mu}) := \psi(\boldsymbol{\mu}) D(\boldsymbol{x}) = \sum_{i=1}^{n_\nu} \psi_{1,i}(\mu_1) \cdots \psi_{M,i}(\mu_M) D_i(\boldsymbol{x}), \tag{2.7}$$

and analogous separations are considered for all the parametric data in the problem under analysis.

2.3.1 Separable form of the parametrised Stokes flow

The parametrised Stokes flow equations can be recovered from Equation (2.3) by setting the advective velocity field $\boldsymbol{a} = \boldsymbol{0}$

$$\begin{cases} -\int_{\boldsymbol{\mathcal{I}}}\int_{V_i} \boldsymbol{\nabla}\cdot(\nu\boldsymbol{\nabla}\boldsymbol{u})\,dV\,d\boldsymbol{\mathcal{I}} + \int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}p\,dV\,d\boldsymbol{\mathcal{I}} = \int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{s}\,dV\,d\boldsymbol{\mathcal{I}}, \\ \int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot\boldsymbol{u}\,dV\,d\boldsymbol{\mathcal{I}} = 0. \end{cases} \tag{2.8}$$

By plugging (2.5) into (2.8) and gathering the unknown increments $(\sigma_u^n\delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}}, \sigma_p^n\delta\widetilde{p}^n_{\text{PGD}})$ on the left-hand side while leaving on the right-hand side the residuals computed using the previous modes $(\boldsymbol{u}^{n-1}_{\text{PGD}}, p^{n-1}_{\text{PGD}})$ and the predictions $(\sigma_u^n\boldsymbol{f}_u^n\phi^n, \sigma_p^n f_p^n\phi^n)$ of the current one, the following equations are obtained

$$\begin{cases} -\int_{\boldsymbol{\mathcal{I}}}\psi\int_{V_i}\boldsymbol{\nabla}\cdot(D\boldsymbol{\nabla}(\sigma_u^n\delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}}))\,dV\,d\boldsymbol{\mathcal{I}} + \int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}(\sigma_p^n\delta\widetilde{p}^n_{\text{PGD}})\,dV\,d\boldsymbol{\mathcal{I}} = \mathcal{R}_u, \\ \int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot(\sigma_u^n\boldsymbol{\delta}\widetilde{\boldsymbol{u}}^n_{\text{PGD}})\,dV\,d\boldsymbol{\mathcal{I}} = \mathcal{R}_p, \end{cases} \tag{2.9}$$

where the residuals are defined as

$$\begin{aligned} \mathcal{R}_u &:= \mathcal{R}_u(\boldsymbol{u}^{n-1}_{\text{PGD}}, p^{n-1}_{\text{PGD}}, \sigma_u^n\boldsymbol{f}_u^n, \sigma_p^n f_p^n, \phi^n) = \mathcal{R}_u(\widetilde{\boldsymbol{u}}^n_{\text{PGD}}, \widetilde{p}^n_{\text{PGD}}) \\ &= \int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{s}\,dV\,d\boldsymbol{\mathcal{I}} \\ &\quad + \int_{\boldsymbol{\mathcal{I}}}\psi\int_{V_i}\boldsymbol{\nabla}\cdot(D\boldsymbol{\nabla}\widetilde{\boldsymbol{u}}^n_{\text{PGD}})\,dV\,d\boldsymbol{\mathcal{I}} - \int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\widetilde{p}^n_{\text{PGD}}\,dV\,d\boldsymbol{\mathcal{I}}, \end{aligned} \tag{2.10}$$

$$\begin{aligned} \mathcal{R}_p &:= \mathcal{R}_p(\boldsymbol{u}^{n-1}_{\text{PGD}}, \sigma_u^n\boldsymbol{f}_u^n, \phi^n) = \mathcal{R}_p(\widetilde{\boldsymbol{u}}^n_{\text{PGD}}) \\ &= -\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot\widetilde{\boldsymbol{u}}^n_{\text{PGD}}\,dV\,d\boldsymbol{\mathcal{I}}. \end{aligned} \tag{2.11}$$

As classical in ROMs [88, 95], an affine dependence of the forms in (2.9), (2.10) and (2.11) on the parameters is required to construct the PGD approximation. The spatial (respectively, parametric) component of each mode is thus computed by restricting Equation (2.9) to the tangent manifold associated with the spatial (respectively, parametric) coordinate. Following from Equation (2.6) and setting a fixed value for the parametric function ϕ^n, the pair $(\sigma_u^n \Delta \boldsymbol{f}_u, \sigma_p^n \Delta f_p)$ is determined by solving a purely spatial PDE. Recall that the PGD alternating direction scheme handles homogeneous Dirichlet boundary conditions at each iteration of the spatial solver [23], whereas inhomogeneous data are treated by the first arbitrary PGD mode $(\boldsymbol{u}_{\text{PGD}}^0, p_{\text{PGD}}^0)$ introduced above. In a similar fashion, the increment $\Delta\phi$ is computed as the solution of an algebraic system of equations in the parameter $\boldsymbol{\mu}$ while the spatial functions $(\sigma_u^n \boldsymbol{f}_u^n, \sigma_p^n f_p^n)$ are considered known.

Note that at each iteration of the alternating direction scheme, $\widetilde{\boldsymbol{u}}_{\text{PGD}}^n$ is known and may be expressed in separated form as $\sum_{m=1}^n \sigma_u^m \boldsymbol{f}_u^m \phi^m$. Thus, exploiting the separated structure of the unknowns and the affine parametric decomposition of the involved integral forms, the numerical complexity of the high-dimensional PDE is reduced by alternatively solving for the spatial and the parametric unknowns, as detailed in the next subsections.

Remark 2. *By restricting Equation (2.9) to the tangent manifold in the spatial (respectively, parametric) direction, the integral forms are multiplied by ϕ^n (respectively, $(\sigma_u^n \boldsymbol{f}_u^n, \sigma_p^n f_p^n)$). This is equivalent to the projection of the high-dimensional PDE to the tangent manifold discussed for PGD in the context of finite element approximations [30]. In the framework of FV* *discretisations, the finite element test functions are set equal to* 1 *to retrieve the classical integral form of the PDE under analysis. Hence, the restriction of the high-dimensional PDE to the tangent manifold introduces a factor* ϕ^n *multiplying the integral forms in Equations* (2.9), (2.10) *and* (2.11) *for the spatial iteration. Similarly, in the parametric iteration, a factor* $\sigma_u^n \boldsymbol{f}_u^n$ *appears in the integral form of the momentum equation and in the residual* $\mathcal{R}_u$*, whereas* $\sigma_p^n f_p^n$ *multiplies the integrand terms in the mass conservation equation and in the residual* $\mathcal{R}_p$*.*

2.3.2 The spatial iteration

First, the parametric function ϕ^n is fixed and the increments $(\sigma_u^n \varDelta \boldsymbol{f}_u, \sigma_p^n \varDelta f_p)$ are determined by solving a spatial PDE. More precisely, restricting Equation (2.9) to the tangent manifold in the spatial direction, a pair $(\sigma_u^n \varDelta \boldsymbol{f}_u, \sigma_p^n \varDelta f_p)$, constant element-by-element, is sought such that in each cell V_i, $i{=}1,\dots,N$ it holds

$$\begin{cases} -\alpha_2 \displaystyle\int_{V_i} \boldsymbol{\nabla}\cdot(D\boldsymbol{\nabla}(\sigma_u^n \varDelta \boldsymbol{f}_u))\, dV + \alpha_3 \int_{V_i} \boldsymbol{\nabla}(\sigma_p^n \varDelta f_p)\, dV = R_u^n, \\ \qquad\qquad\qquad\qquad\qquad\qquad \alpha_3 \displaystyle\int_{V_i} \boldsymbol{\nabla}\cdot(\sigma_u^n \varDelta \boldsymbol{f}_u)\, dV = R_p^n, \end{cases} \tag{2.12}$$

where R_u^n and R_p^n are the spatial residuals associated with the discretisation of the momentum and mass equations, respectively, and each coefficient α_i, $i{=}1,2$ depends solely on the parametric function ϕ^n and on the data of the problem

$$\alpha_2 := \int_{\boldsymbol{\mathcal{I}}} [\phi^n]^2 \, \psi \, d\boldsymbol{\mathcal{I}}, \quad \alpha_3 := \int_{\boldsymbol{\mathcal{I}}} [\phi^n]^2 \; d\boldsymbol{\mathcal{I}}. \tag{2.13}$$

Note that given the separable form of (2.10)-(2.11), an efficient implementation of the right-hand side of the spatial iteration may be devised and the corresponding FV discretisation is obtained. A detailed description of the residuals acting as linear functionals on the right-hand side of Equation (2.12) is provided in A.

The PDE of Equation (2.12) can be solved in Ω using the FV discretisation scheme libraries of OpenFOAM yielding the correction to the velocity and pressure modes.

It is reminded that for any particularisation of the parameters the solution of the full-order Stokes flow problem is carried out by a SIMPLE-based Stokes solver as implemented in OpenFOAM (see Appendix B.1).

2.3.3 The parametric iteration

In the parametric step, the value of the previously computed spatial functions is fixed $(\boldsymbol{f}_u^n, f_p^n) \leftarrow (\sigma_u^n \boldsymbol{f}_u^n + \Delta \boldsymbol{f}_u, \sigma_p^n f_p^n + \Delta f_p)$ and the parametric increment $\Delta\phi$ acts as unknown. Within the *single parameter approximation* rationale, a unique scalar function depending on $\boldsymbol{\mu}$ is sought. Following the strategy described for the spatial iteration, Equation (2.9) is restricted to the parametric direction of the tangent manifold and $\Delta\phi$ is computed by solving the following algebraic equation

$$(-a_2\psi + a_3)\,\Delta\phi = r_u^n + r_p^n, \tag{2.14}$$

where r_u^n and r_p^n are the parametric residuals associated with the discretisation of the momentum and mass equations, respectively, and each coefficient a_i, $i{=}1,2$ depends solely on the spatial functions $(\sigma_u^n \boldsymbol{f}_u^n, \sigma_p^n f_p^n)$ and on the data of the problem, namely

$$\begin{cases} a_2 := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \left[\boldsymbol{\nabla} \cdot (D \boldsymbol{\nabla} (\sigma_u^n \boldsymbol{f}_u^n)) \right] dV, \\ a_3 := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \boldsymbol{\nabla} (\sigma_p^n f_p^n)\, dV + \int_{V_i} \sigma_p^n f_p^n \boldsymbol{\nabla} \cdot (\sigma_u^n \boldsymbol{f}_u^n)\, dV. \end{cases} \tag{2.15}$$

The unknown $\Delta\phi$ is discretised at the nodes of the parametric domain $\boldsymbol{\mathcal{I}}$ and the resulting algebraic equation is solved via a collocation method. Similarly to the spatial iteration, the separable form of (2.10)-(2.11) is exploited to perform computationally efficient pointwise evaluations of the residuals at the nodes of $\boldsymbol{\mathcal{I}}$. The complete derivation of the separated form of the right-hand side is detailed in A.

2.4 Predictor-corrector alternating direction scheme for parametrised Oseen flow

At this step, and moving towards the Navier-Stokes system, a linearised convection term is considered. Setting $\boldsymbol{a} = \boldsymbol{u}_0$ in Equation (2.3), with $\boldsymbol{u}_0$ a user prescribed divergence-free velocity field, yields the Oseen system of equations

$$
\begin{cases}
\displaystyle\int_{\mathcal{I}}\int_{V_i} \boldsymbol{\nabla}\cdot(\boldsymbol{u}\otimes\boldsymbol{u}_0)\, dV\, d\boldsymbol{\mathcal{I}} - \int_{\mathcal{I}}\int_{V_i} \boldsymbol{\nabla}\cdot(\nu\boldsymbol{\nabla}\boldsymbol{u})\, dV\, d\boldsymbol{\mathcal{I}} \\
\qquad\qquad\qquad\qquad + \displaystyle\int_{\mathcal{I}}\int_{V_i} \boldsymbol{\nabla} p\, dV\, d\boldsymbol{\mathcal{I}} = \int_{\mathcal{I}}\int_{V_i} \boldsymbol{s}\, dV\, d\boldsymbol{\mathcal{I}}, \\
\qquad\qquad\qquad\qquad \displaystyle\int_{\mathcal{I}}\int_{V_i} \boldsymbol{\nabla}\cdot\boldsymbol{u}\, dV\, d\boldsymbol{\mathcal{I}} = 0.
\end{cases}
\tag{2.16}
$$

Similarly to the Stokes flow, a SIMPLE-based solver for the full-order Oseen flow problem has been implemented in OpenFOAM (see Appendix B.2).

As observed from a comparison between equations (2.8) and (2.16) the Oseen flow system is identical to the Stokes flow system described above with the exception of the linearised convection term. As such, the main point of interest will be the handling of the extra term and the required adaptation of the alternating direction scheme.

It must be noted that the assumption of separability of data discussed in Section 2.3 stands for the Oseen flow. Much like viscosity (Equation (2.7)) the following separable form of the prescribed velocity field is assumed

$$
\boldsymbol{u}_0(\boldsymbol{x},\boldsymbol{\mu}) := \xi(\boldsymbol{\mu})\boldsymbol{F}(\boldsymbol{x}) = \sum_{i=1}^{n_{\boldsymbol{u}_0}} \xi_{1,i}(\mu_1)\cdots\xi_{M,i}(\mu_M)\boldsymbol{F}_i(\boldsymbol{x}), \tag{2.17}
$$

2.4.1 Separable form of the linearised convection term

By plugging the predictor-corrector approximation (2.6) in (2.16) the contribution of the convection term to the left- and right-hand side, respectively, reads:

$$
\begin{aligned}
&\mathcal{LHS}: \int_{\boldsymbol{\mathcal{I}}} \xi \int_{V_i} \boldsymbol{\nabla}\cdot(\sigma_u^n \delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}} \otimes \boldsymbol{F})\, dV\, d\boldsymbol{\mathcal{I}} \\
&\mathcal{RHS}: \ -\int_{\boldsymbol{\mathcal{I}}} \xi \int_{V_i} \boldsymbol{\nabla}\cdot(\widetilde{\boldsymbol{u}}^n_{\text{PGD}} \otimes \boldsymbol{F})\, dV\, d\boldsymbol{\mathcal{I}}.
\end{aligned}
\tag{2.18}
$$

Augmenting the separable form of the Stokes system (2.9) with the aforementioned convection terms yields the separable Oseen system

$$
\begin{cases}
\displaystyle\int_{\boldsymbol{\mathcal{I}}} \xi \int_{V_i} \boldsymbol{\nabla}\cdot(\sigma_u^n \delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}} \otimes \boldsymbol{F})\, dV\, d\boldsymbol{\mathcal{I}} - \int_{\boldsymbol{\mathcal{I}}} \psi \int_{V_i} \boldsymbol{\nabla}\cdot(D\boldsymbol{\nabla}(\sigma_u^n \delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}}))\, dV\, d\boldsymbol{\mathcal{I}} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \displaystyle + \int_{\boldsymbol{\mathcal{I}}} \int_{V_i} \boldsymbol{\nabla}(\sigma_p^n \delta\widetilde{p}^n_{\text{PGD}})\, dV\, d\boldsymbol{\mathcal{I}} = \mathcal{R}_u, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \displaystyle \int_{\boldsymbol{\mathcal{I}}} \int_{V_i} \boldsymbol{\nabla}\cdot(\sigma_u^n \delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}})\, dV\, d\boldsymbol{\mathcal{I}} = \mathcal{R}_p,
\end{cases}
\tag{2.19}
$$

where the residuals are defined as

$$
\begin{aligned}
\mathcal{R}_u :=& \, \mathcal{R}_u(\boldsymbol{u}^{n-1}_{\text{PGD}}, p^{n-1}_{\text{PGD}}, \sigma_u^n \boldsymbol{f}_u^n, \sigma_p^n f_p^n, \phi^n) = \mathcal{R}_u(\widetilde{\boldsymbol{u}}^n_{\text{PGD}}, \widetilde{p}^n_{\text{PGD}}) \\
=& \int_{\boldsymbol{\mathcal{I}}} \int_{V_i} \boldsymbol{s}\, dV\, d\boldsymbol{\mathcal{I}} - \int_{\boldsymbol{\mathcal{I}}} \xi \int_{V_i} \boldsymbol{\nabla}\cdot(\widetilde{\boldsymbol{u}}^n_{\text{PGD}} \otimes \boldsymbol{F})\, dV\, d\boldsymbol{\mathcal{I}} \\
&+ \int_{\boldsymbol{\mathcal{I}}} \psi \int_{V_i} \boldsymbol{\nabla}\cdot(D\boldsymbol{\nabla}\widetilde{\boldsymbol{u}}^n_{\text{PGD}})\, dV\, d\boldsymbol{\mathcal{I}} - \int_{\boldsymbol{\mathcal{I}}} \int_{V_i} \boldsymbol{\nabla}\widetilde{p}^n_{\text{PGD}}\, dV\, d\boldsymbol{\mathcal{I}},
\end{aligned}
\tag{2.20}
$$

$$
\begin{aligned}
\mathcal{R}_p :=& \, \mathcal{R}_p(\boldsymbol{u}^{n-1}_{\text{PGD}}, \sigma_u^n \boldsymbol{f}_u^n, \phi^n) = \mathcal{R}_p(\widetilde{\boldsymbol{u}}^n_{\text{PGD}}) \\
=& -\int_{\boldsymbol{\mathcal{I}}} \int_{V_i} \boldsymbol{\nabla}\cdot\widetilde{\boldsymbol{u}}^n_{\text{PGD}}\, dV\, d\boldsymbol{\mathcal{I}}.
\end{aligned}
\tag{2.21}
$$

2.4.2 The spatial iteration

Restricting Equation (2.19) to the tangent manifold in the spatial direction the spatial increments of the n-th mode are sought

$$\begin{cases} \alpha_1 \displaystyle\int_{V_i} \boldsymbol{\nabla}\cdot\left(\sigma_u^n \varDelta \boldsymbol{f}_u \otimes \boldsymbol{F}\right) dV - \alpha_2 \displaystyle\int_{V_i} \boldsymbol{\nabla}\cdot\left(D \boldsymbol{\nabla}(\sigma_u^n \varDelta \boldsymbol{f}_u)\right) dV & \\ \qquad\qquad + \alpha_3 \displaystyle\int_{V_i} \boldsymbol{\nabla}(\sigma_p^n \varDelta f_p)\, dV = R_u^n, & \\ \qquad\qquad \alpha_3 \displaystyle\int_{V_i} \boldsymbol{\nabla}\cdot(\sigma_u^n \varDelta \boldsymbol{f}_u)\, dV = R_p^n, & \end{cases} \tag{2.22}$$

where R_u^n and R_p^n are the spatial residuals associated with the discretisation of the momentum and mass equations, respectively, and each coefficient α_i, $i{=}1,\dots,3$ depends solely on the parametric function ϕ^n and on the data of the problem

$$\alpha_1 := \int_{\boldsymbol{\mathcal{I}}} [\phi^n]^2 \, \xi \, d\boldsymbol{\mathcal{I}}, \quad \alpha_2 := \int_{\boldsymbol{\mathcal{I}}} [\phi^n]^2 \, \psi \, d\boldsymbol{\mathcal{I}}, \quad \alpha_3 := \int_{\boldsymbol{\mathcal{I}}} [\phi^n]^2 \, d\boldsymbol{\mathcal{I}}. \tag{2.23}$$

A detailed description of the residuals acting as linear functionals on the right-hand side of Equation (2.22) is provided in A.

The PDE of Equation (2.22) features a structure similar to the original Oseen problem in the spatial domain Ω and can thus be solved using numerical schemes found in OpenFOAM.

Remark 3. *It has to be noted at this points that with the use of a fictitious viscosity $\hat{\nu} = \frac{\alpha_2}{\alpha_3} D$ and a modified user-prescribed velocity $\hat{\boldsymbol{a}} = \frac{\alpha_1}{\alpha_3} \boldsymbol{F}$ an Oseen flow problem can be assembled and the discretisation is thus performed using the Oseen FV solver implemented in OpenFOAM, see Appendix B.2. This* *point will be revisited in the chapter regarding the nonintrusive PGD for Navier-Stokes problems.*

2.4.3 The parametric iteration

Much like the parametric step of the PGD for parametrised Stokes flow described in 2.3.3 the value of the previously computed spatial functions is fixed $(\boldsymbol{f}_u^n, f_p^n) \leftarrow (\sigma_u^n \boldsymbol{f}_u^n + \Delta \boldsymbol{f}_u, \sigma_p^n f_p^n + \Delta f_p)$ and the parametric increment $\Delta\phi$ acts as the unknown. Equation (2.19) is restricted to the parametric direction of the tangent manifold and $\Delta\phi$ is computed by solving the following algebraic equation

$$(a_1 \xi - a_2 \psi + a_3)\, \Delta\phi = r_u^n + r_p^n, \tag{2.24}$$

where r_u^n and r_p^n are the parametric residuals associated with the discretisation of the momentum and mass equations, respectively, and each coefficient a_i, $i{=}1,\dots,3$ depends solely on the spatial functions $(\sigma_u^n \boldsymbol{f}_u^n, \sigma_p^n f_p^n)$ and on the data of the problem, namely

$$\begin{cases} a_1 := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \left[\boldsymbol{\nabla} \cdot (\sigma_u^n \boldsymbol{f}_u^n \otimes \boldsymbol{F}) \right] dV \\ a_2 := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \left[\boldsymbol{\nabla} \cdot (D \boldsymbol{\nabla} (\sigma_u^n \boldsymbol{f}_u^n)) \right] dV, \\ a_3 := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \boldsymbol{\nabla} (\sigma_p^n f_p^n)\, dV + \int_{V_i} \sigma_p^n f_p^n \boldsymbol{\nabla} \cdot (\sigma_u^n \boldsymbol{f}_u^n)\, dV. \end{cases} \tag{2.25}$$

The complete derivation of the separated form of the right-hand side is detailed in A.

2.5 Numerical validation

In this section, the proposed PGD scheme for linear problems implemented in OpenFOAM is validated using numerical examples. Starting with the parametrised Stokes flow, a test case with known analytical solution is considered to verify the optimal convergence rate of the high-dimensional FV approximation of the velocity and pressure fields, measured in the $\mathcal{L}_2(\Omega\times\boldsymbol{\mathcal{I}})$ norm, for a multi-parametric anisotropic viscosity coefficient. In the Oseen framework, the optimal rate of convergence is sought using a well documented benchmark case where the viscosity coefficient is the parameter. In this context, special emphasis is given to the additional error introduced by the PGD, highlighting the range of applicability of the discussed reduced-order strategy in terms of expected accuracy of the parametric solution.

2.5.1 Parametrised Stokes flow

Consider a spatial domain $\Omega{=}[0,1]^2$ with anisotropic viscosity that is controlled by $M = 2$ parameters. More specifically, the dependence reads

$$\nu(\boldsymbol{x}, \mu_1, \mu_2) = \mu_1(1-y) + \mu_2 y$$

with $\mathcal{I}_1 \in$ [1, 2] and $\mathcal{I}_2 \in$ [3, 4], both discretised with the same number of uniform intervals. The spatial domain is discretised with a family of cartesian meshes of quadrilateral cells.

Introducing the following external body forces

$$\boldsymbol{s}(\boldsymbol{x},\mu_1,\mu_2) = \begin{pmatrix} (-\mu_1\mu_2+\mu_1^2)(6x(1-x)-1) \\ (\mu_1\mu_2-\mu_2^2)(6y(1-y)-1) \end{pmatrix}$$

velocity and pressure can be analytically expressed as

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x},\mu_1,\mu_2) &= \begin{pmatrix} \mu_1(1-y)+\mu_2 x \\ -\mu_1 x-\mu_2 y \end{pmatrix} \\ p(\boldsymbol{x},\mu_1,\mu_2) &= (-\mu_1\mu_2+\mu_1^2(x^2(3-2x)) + (\mu_1\mu_2-\mu_2^2)y^2(3-2y). \end{aligned} \tag{2.26}$$

Using the same normalised characteristic length h_{μ_1}, h_{μ_2} and h_x, listed in Table 2.1, a uniform mesh refinement study is run.

h	8.3×10^{-2}	4×10^{-2}	2×10^{-2}	1×10^{-2}	5×10^{-3}	2.5×10^{-3}

Table 2.1: Normalised characteristic length of both the spatial and parametric discretisations.

In order to run the PGD an affine separation of the data, as detailed in Section 2.3, is required. The separable form of the anisotropic viscosity consists of two modes

$$\begin{aligned} D_1(\boldsymbol{x}) &= 1-y\,, & \psi_1(\mu_1) &= \mu_1\,, & \psi_1(\mu_2) &= 1 \\ D_2(\boldsymbol{x}) &= y\,, & \psi_2(\mu_1) &= 1\,, & \psi_2(\mu_2) &= \mu_2\ . \end{aligned} \tag{2.27}$$

while the Dirichlet velocity datum is imposed via three modes, with respective null pressure fields

$$
\begin{aligned}
\boldsymbol{f}_u^1(\boldsymbol{x}) &= \begin{pmatrix} x + y(y-1)x(x-1) \\ -y \end{pmatrix} , & f_p^1(\boldsymbol{x}) = 0 \\
\boldsymbol{f}_u^2(\boldsymbol{x}) &= \begin{pmatrix} 1-y \\ -x - y(y-1)x(x-1) \end{pmatrix} , & f_p^2(\boldsymbol{x}) = 0 \\
\boldsymbol{f}_u^3(\boldsymbol{x}) &= \begin{pmatrix} -(x^2-1)2y + x(x-1)^2(y-1)y^3 \\ (y^2-1)2x - x(x-1)^2(y-1)y^3 \end{pmatrix} , & f_p^3(\boldsymbol{x}) = 0 \\
&\phi^1(\mu_1) = 1 \ , \ \phi^2(\mu_1) = \mu_1 \ , \ \phi^3(\mu_1) = 1, \\
&\phi^1(\mu_2) = \mu_2 \ , \ \phi^2(\mu_2) = 1 \ , \ \phi^3(\mu_1) = 1 \ .
\end{aligned}
\tag{2.28}
$$

Each individual PGD is run until either the relative amplitude of the computed mode drops below a threshold or N modes have been computed. More specifically, the relative amplitude criterion is $\eta_{(u,p)} \leq 10^{-5}$, where $\eta_{(u,p)}$ accounts for the relative amplitude of both the velocity and pressure modes, such that

$$
\eta_{(u,p)} := \sqrt{\left(\frac{\sigma_u^n}{\sum_{m=1}^n \sigma_u^m} \right)^2 + \left(\frac{\sigma_p^n}{\sum_{m=1}^n \sigma_p^m} \right)^2} . \tag{2.29}
$$

and the maximum number of modes is $N = 20$.

The $\mathcal{L}_2(\Omega \times \boldsymbol{\mathcal{I}})$ error between the PGD approximation $(\boldsymbol{u}_{\texttt{PGD}}^n, p_{\texttt{PGD}}^n)$ computed and the analytical solution $(\boldsymbol{u}, p)$ as a function of the characteristic mesh size h_x is displayed in Figure 2.1. The optimal first-order convergence rate for pressure and second-order one for velocity are obtained.

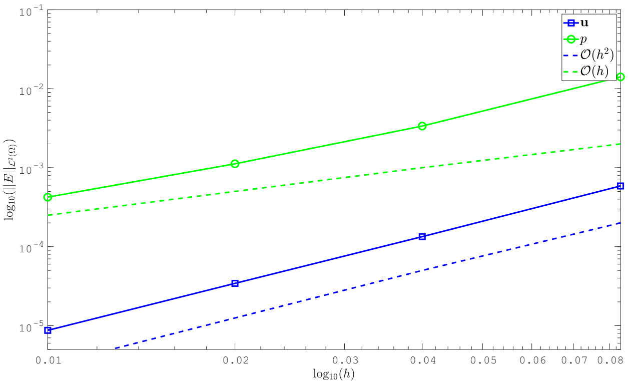


Figure 2.1: Optimal convergence of the $\mathcal{L}_2(\Omega\times\boldsymbol{\mathcal{I}})$ error of the PGD approximation of the analytical Stokes flow with parametrised anisotropic viscosity using two parameters with respect to the exact solution as a function of the characteristic mesh size h.

2.5.2 Parametrised Oseen flow

Consider the Kovasznay flow [61] for a parametrised viscosity $\nu(\mu)=\mu$. The analytical solution is

$$\begin{aligned} \boldsymbol{u}(x,y,\mu) &= \left(1-e^{\lambda(\mu)x}\cos(2\pi y),\, \tfrac{\lambda(\mu)}{2\pi}e^{\lambda(\mu)x}\sin(2\pi y)\right) \\ p(x,y,\mu) &= \tfrac{1}{2}\left(1-e^{2\lambda(\mu)x}\right)+C \end{aligned} \tag{2.30}$$

where the constant C is determined by fixing a reference value for the pressure field in one point of the domain, whereas λ is a function of the parametrised viscosity and changes when the Reynolds number is modified, namely,

$$\lambda(\mu) = \tfrac{1}{2\nu(\mu)} - \sqrt{\tfrac{1}{(2\nu(\mu))^2}+4\pi^2}.$$

The parameter μ is sought in the space $\mathcal{I}=[5\times10^{-3}, 10^{-2}]$, which is discretised with uniform intervals. The corresponding values of the Reynolds number span from 100 to 200. The spatial domain $\Omega{=}[-1,1]^2$ is discretised with a family of Cartesian meshes of quadrilateral cells. The characteristic lengths h_x and h_μ of the spatial and parametric discretisations, respectively, are provided in Table 2.2.

h_x	8.3×10^{-2}	$4{\times}10^{-2}$	$2{\times}10^{-2}$	$1{\times}10^{-2}$	$5{\times}10^{-3}$	$2.5{\times}10^{-3}$
h_μ	$2{\times}10^{-2}$	$1{\times}10^{-2}$	$5{\times}10^{-3}$	$2.5{\times}10^{-3}$	$1.25{\times}10^{-3}$	$6.25{\times}10^{-4}$

Table 2.2: Normalised characteristic lengths of the spatial and parametric discretisations.

A convergence study under uniform mesh refinement is performed for the linearised Navier-Stokes equations using the meshes described in Table 2.2. In this context, a convective field $\boldsymbol{a}$ given by the analytical expression of the Kovasznay velocity is introduced in Equation (2.1). As detailed in Section 2.4, an affine separation of the data is required to run PGD. Thus, the convective field $\boldsymbol{a}$ is separated *a priori* considering the first four terms of the Taylor expansion of $e^{\lambda x}$ in the analytical form of the velocity, see Equation (2.30). For $\mu{=}10^{-2}$, the relative $\mathcal{L}_2(\Omega)$ error of the resulting separated velocity field with respect to the exact one is $4.3{\times}10^{-3}$ and, consequently, a target error of 10^{-2} in the spatial discretisation is considered for the following convergence study. Moreover, the Dirichlet boundary datum $\boldsymbol{u}_D$ requires five modes to be described in a separated form.

The $\mathcal{L}_2(\Omega{\times}\boldsymbol{\mathcal{I}})$ error between the PGD approximation $(\boldsymbol{u}^n_{\texttt{PGD}}, p^n_{\texttt{PGD}})$ computed using fifteen modes and the high-dimensional analytical solution $(\boldsymbol{u}, p)$ as a function of the characteristic mesh size h_x is displayed in Figure 2.2. The optimal first-order convergence rate for pressure and second-order one for velocity are obtained.

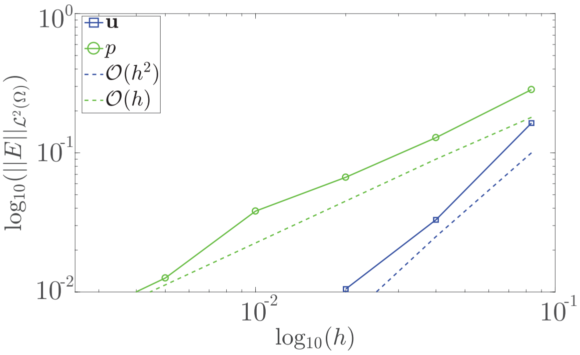
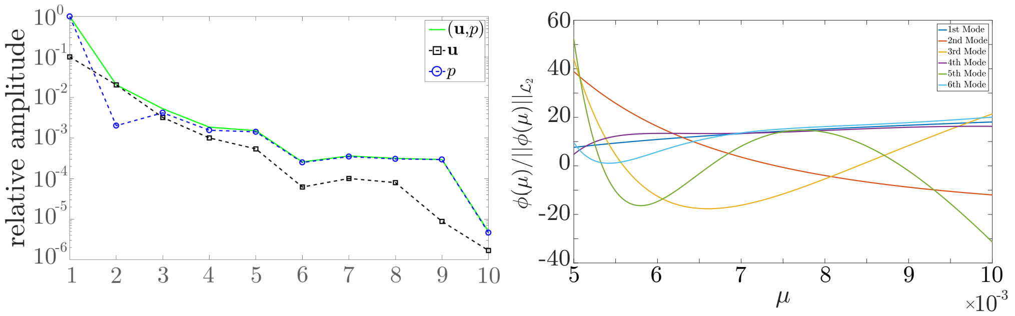


Figure 2.2: Optimal convergence of the $\mathcal{L}_2(\Omega\times\boldsymbol{\mathcal{I}})$ error of the PGD approximation of the Kovasznay flow with parametrised viscosity with respect to the exact solution as a function of the characteristic mesh size h_x.

Similarly to section 2.5.1, the stopping criterion is chosen to be $\eta_{(u,p)} \leq 10^{-5}$ (2.29). In Figure 2.3(a), the evolution of the amplitude $\eta_{(u,p)}$, η_u and η_p is displayed for the finest mesh described in Table 2.2. After ten computed modes, the stopping criterion is fulfilled and the PGD enrichment stops.

As previously mentioned, five terms are required to describe the Dirichlet boundary conditions in a separated form. Henceforth, only the computed modes, starting from the sixth term of the PGD approximation are displayed. In Figures 2.3(b), the first six normalised computed parametric modes are displayed. The corresponding computed spatial modes for pressure and velocity are presented in Figure 2.4 and 2.5, respectively.

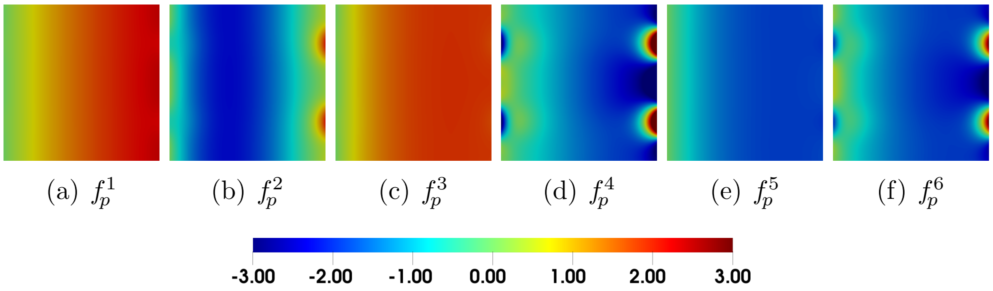
The PGD approximation $(\boldsymbol{u}^n_{\texttt{PGD}}, p^n_{\texttt{PGD}})$ using n=1, 3, 10 terms is compared to the analytical solution for the case of Re=200, in Figure 2.6.



(a) Amplitude of the computed modes

(b) Computed parametric modes

Figure 2.3: PGD approximation of the Kovasznay flow with parametrised viscosity. (a) Relative amplitude of the computed modes $\boldsymbol{f}_u^m$ (black), f_p^m (blue) and the combined amplitude of $(\boldsymbol{f}_u^m, f_p^m)$ according to Equation (**??**). (b) First six normalised computed parametric modes.



(a) f_p^1

(b) f_p^2

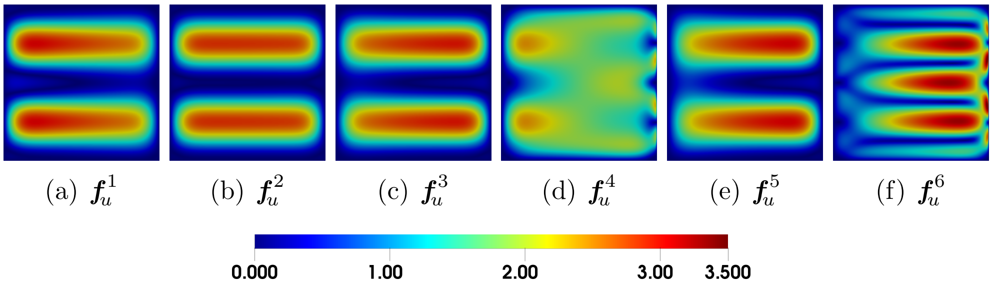
(c) f_p^3

(d) f_p^4

(e) f_p^5

(f) f_p^6

Figure 2.4: PGD approximation of the Kovasznay flow with parametrised viscosity. First six computed spatial modes f_p^m, $m = 1, \ldots, 6$ for pressure.



(a) $\boldsymbol{f}_u^1$

(b) $\boldsymbol{f}_u^2$

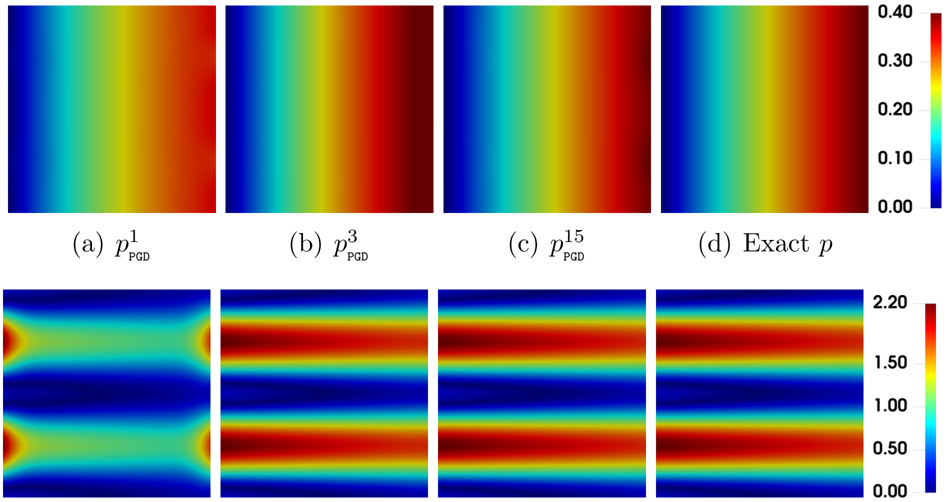
(c) $\boldsymbol{f}_u^3$

(d) $\boldsymbol{f}_u^4$

(e) $\boldsymbol{f}_u^5$

(f) $\boldsymbol{f}_u^6$

Figure 2.5: PGD approximation of the Kovasznay flow with parametrised viscosity. First six computed spatial modes $\boldsymbol{f}_u^m$, $m = 1, \ldots, 6$ for velocity.



(a) p^1_{PGD} (b) p^3_{PGD} (c) p^{15}_{PGD} (d) Exact p

(f) $\boldsymbol{u}^1_{\text{PGD}}$ (g) $\boldsymbol{u}^3_{\text{PGD}}$ (h) $\boldsymbol{u}^{15}_{\text{PGD}}$ (i) Exact $\boldsymbol{u}$

Figure 2.6: Comparison of the PGD approximation to the analytical solution of the Kovasznay flow for Re $= 200$, that is $\mu = 10^{-2}$. Pressure (top) and velocity (bottom) using 1, 3 and 15 modes.

Chapter 3

The nonintrusive PGD for parametrised Navier-Stokes flow

Having validated the OpenFOAM implementation of PGD for linear flow problems the laminar Navier-Stokes flow is considered. As aforementioned, a bottleneck to the adoption of PGD in the daily industrial environment rests in the complexity and intrusiveness of the method. Initially, the construction of a separable approximation of the parametrised Navier-Stokes flow problem is recalled. During the derivation of the PGD formulation the concept of the nonintrusive PGD is introduced and the implementation of the nonintrusive PGD method in OpenFOAM is described. Finally, two- and three-dimensional parametrised problems are solved to test the proposed method.

3.1 Predictor-corrector alternating direction scheme for parametrised Navier-Stokes flow

This section expands on the separability of the laminar Navier-Stokes flow as well as the feasibility of a nonintrusive implementation in OpenFOAM. Setting $\boldsymbol{a} = \boldsymbol{u}$ in Equation (2.3) the higher dimensional Navier-Stokes system of equations reads

$$\begin{cases} \displaystyle\int_{\boldsymbol{\mathcal{I}}}\int_{V_i} \boldsymbol{\nabla}\cdot(\boldsymbol{u}\otimes\boldsymbol{u})\, dV\, d\boldsymbol{\mathcal{I}} - \int_{\boldsymbol{\mathcal{I}}}\int_{V_i} \boldsymbol{\nabla}\cdot(\nu\boldsymbol{\nabla}\boldsymbol{u})\, dV\, d\boldsymbol{\mathcal{I}} \\ \displaystyle\qquad\qquad\qquad\qquad + \int_{\boldsymbol{\mathcal{I}}}\int_{V_i} \boldsymbol{\nabla} p\, dV\, d\boldsymbol{\mathcal{I}} = \int_{\boldsymbol{\mathcal{I}}}\int_{V_i} \boldsymbol{s}\, dV\, d\boldsymbol{\mathcal{I}}, \\ \displaystyle\qquad\qquad\qquad\qquad \int_{\boldsymbol{\mathcal{I}}}\int_{V_i} \boldsymbol{\nabla}\cdot\boldsymbol{u}\, dV\, d\boldsymbol{\mathcal{I}} = 0. \end{cases} \tag{3.1}$$

As is observed, in a similar fashion to the parametrised Oseen flow, the nonlinear convection term needs to be accounted for.

Remark 4. *It is at this point reminded that all previous assumptions of separability still stand.*

By plugging the predictor-corrector approximation (2.6) in (3.1) and gathering the unknown increments $(\sigma_u^n \delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}}, \sigma_p^n \delta\widetilde{p}^n_{\text{PGD}})$ on the left-hand side while leaving on the right-hand side the residuals computed using the previous modes $(\boldsymbol{u}^{n-1}_{\text{PGD}}, p^{n-1}_{\text{PGD}})$ and the predictions $(\sigma_u^n \boldsymbol{f}_u^n \phi^n, \sigma_p^n f_p^n \phi^n)$ of the current one, the following equations are obtained

$$\left\{\begin{aligned}
&\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot(\sigma_u^n\delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}}\otimes\sigma_u^n\delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}})\,dV\,d\boldsymbol{\mathcal{I}}\\
&\quad+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot(\sigma_u^n\delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}}\otimes\widetilde{\boldsymbol{u}}^n_{\text{PGD}})\,dV\,d\boldsymbol{\mathcal{I}}+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot(\widetilde{\boldsymbol{u}}^n_{\text{PGD}}\otimes\sigma_u^n\delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}})\,dV\,d\boldsymbol{\mathcal{I}}\\
&\quad-\int_{\boldsymbol{\mathcal{I}}}\psi\int_{V_i}\boldsymbol{\nabla}\cdot(D\boldsymbol{\nabla}(\sigma_u^n\delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}}))\,dV\,d\boldsymbol{\mathcal{I}}+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}(\sigma_p^n\delta\widetilde{p}^n_{\text{PGD}})\,dV\,d\boldsymbol{\mathcal{I}}=\mathcal{R}_u,\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot(\sigma_u^n\delta\widetilde{\boldsymbol{u}}^n_{\text{PGD}})\,dV\,d\boldsymbol{\mathcal{I}}=\mathcal{R}_p,
\end{aligned}\right. \tag{3.2}$$

where the residuals are defined as

$$\begin{aligned}
\mathcal{R}_u:=&\,\mathcal{R}_u(\boldsymbol{u}^{n-1}_{\text{PGD}},p^{n-1}_{\text{PGD}},\sigma_u^n\boldsymbol{f}_u^n,\sigma_p^n f_p^n,\phi^n)=\mathcal{R}_u(\widetilde{\boldsymbol{u}}^n_{\text{PGD}},\widetilde{p}^n_{\text{PGD}})\\
=&\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{s}\,dV\,d\boldsymbol{\mathcal{I}}-\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot(\widetilde{\boldsymbol{u}}^n_{\text{PGD}}\otimes\widetilde{\boldsymbol{u}}^n_{\text{PGD}})\,dV\,d\boldsymbol{\mathcal{I}}\\
&+\int_{\boldsymbol{\mathcal{I}}}\psi\int_{V_i}\boldsymbol{\nabla}\cdot(D\boldsymbol{\nabla}\widetilde{\boldsymbol{u}}^n_{\text{PGD}})\,dV\,d\boldsymbol{\mathcal{I}}-\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\widetilde{p}^n_{\text{PGD}}\,dV\,d\boldsymbol{\mathcal{I}},
\end{aligned}\tag{3.3}$$

$$\begin{aligned}
\mathcal{R}_p:=&\,\mathcal{R}_p(\boldsymbol{u}^{n-1}_{\text{PGD}},\sigma_u^n\boldsymbol{f}_u^n,\phi^n)=\mathcal{R}_p(\widetilde{\boldsymbol{u}}^n_{\text{PGD}})\\
=&-\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot\widetilde{\boldsymbol{u}}^n_{\text{PGD}}\,dV\,d\boldsymbol{\mathcal{I}}.
\end{aligned}\tag{3.4}$$

3.1.1 The spatial iteration

First, the parametric function ϕ^n is fixed and the increments $(\sigma_u^n \varDelta \boldsymbol{f}_u, \sigma_p^n \varDelta f_p)$ are determined by solving a spatial PDE. More precisely, restricting Equation (3.2) to the tangent manifold in the spatial direction, a pair $(\sigma_u^n \varDelta \boldsymbol{f}_u, \sigma_p^n \varDelta f_p)$, constant cell-by-cell, is sought such that in each cell V_i, $i{=}1,\dots,N$ it holds

$$\left\{\begin{aligned} &\alpha_0 \int_{V_i} \boldsymbol{\nabla}\cdot(\sigma_u^n \varDelta \boldsymbol{f}_u \otimes \sigma_u^n \varDelta \boldsymbol{f}_u)\, dV \\ &\quad + \int_{V_i} \boldsymbol{\nabla}\cdot\Big(\sigma_u^n \varDelta \boldsymbol{f}_u \otimes \sum_{m=1}^{n} \alpha_1^m \sigma_u^m \boldsymbol{f}_u^m\Big) dV \\ &\qquad + \int_{V_i} \boldsymbol{\nabla}\cdot\Big(\sum_{m=1}^{n} \alpha_1^m \sigma_u^m \boldsymbol{f}_u^m \otimes \sigma_u^n \varDelta \boldsymbol{f}_u\Big) dV \\ &\qquad\quad - \alpha_2 \int_{V_i} \boldsymbol{\nabla}\cdot(D \boldsymbol{\nabla}(\sigma_u^n \varDelta \boldsymbol{f}_u))\, dV + \alpha_3 \int_{V_i} \boldsymbol{\nabla}(\sigma_p^n \varDelta f_p)\, dV = R_u^n, \\ &\qquad\qquad\qquad\qquad\qquad\qquad \alpha_3 \int_{V_i} \boldsymbol{\nabla}\cdot(\sigma_u^n \varDelta \boldsymbol{f}_u)\, dV = R_p^n, \end{aligned}\right. \tag{3.5}$$

where R_u^n and R_p^n are the spatial residuals associated with the discretisation of the momentum and mass equations, respectively, and each coefficient α_i, $i{=}0,\dots,3$ depends solely on the parametric function ϕ^n and on the data of the problem

$$\begin{aligned} \alpha_0 &:= \int_{\boldsymbol{\mathcal{I}}} [\phi^n]^3 \, d\boldsymbol{\mathcal{I}}, & \alpha_2 &:= \int_{\boldsymbol{\mathcal{I}}} [\phi^n]^2 \, \psi \, d\boldsymbol{\mathcal{I}}, \\ \alpha_1^m &:= \int_{\boldsymbol{\mathcal{I}}} [\phi^n]^2 \, \phi^m \, d\boldsymbol{\mathcal{I}}, & \alpha_3 &:= \int_{\boldsymbol{\mathcal{I}}} [\phi^n]^2 \, d\boldsymbol{\mathcal{I}}. \end{aligned} \tag{3.6}$$

Note that given the separable form of (3.3)-(3.4), an efficient implementation of the right-hand side of the spatial iteration may be devised and the corresponding FV discretisation is obtained. A detailed description of the residuals acting as linear functionals on the right-hand side of Equation (3.5) is provided in A.

The terms in Equation (3.5) feature a structure similar to the original incompressible Navier-Stokes problem in the spatial domain Ω, see Equation (3.1). The discretisation is thus performed using the cell-centred FV method implemented in OpenFOAM, see Section 2.1. The main difference is represented by the second and third integrals on the left-hand side of the momentum equation (3.5). In order to preserve the nonintrusiveness of the discussed PGD approach, Equation (3.5) is modified by introducing a relaxation in the SIMPLE iterations to treat these two integrals in an explicit way as part of the right-hand side of the momentum equation, leading to

$$\left\{\begin{aligned}
&\alpha_0\int_{V_i}\boldsymbol{\nabla}\cdot(\sigma_u^n\Delta\boldsymbol{f}_u\otimes\sigma_u^n\Delta\boldsymbol{f}_u)dV\\
&\quad-\alpha_2\int_{V_i}\boldsymbol{\nabla}\cdot(D\boldsymbol{\nabla}(\sigma_u^n\Delta\boldsymbol{f}_u))dV+\alpha_3\int_{V_i}\boldsymbol{\nabla}(\sigma_p^n\Delta f_p)dV=R_u^n\\
&\qquad\qquad\qquad-\int_{V_i}\boldsymbol{\nabla}\cdot\Big(\sum_{m=1}^{n}\alpha_1^m\sigma_u^m\boldsymbol{f}_u^m\otimes\sigma_u^{k-1}\Delta\boldsymbol{f}_u^{k-1}\Big)dV\\
&\qquad\qquad\qquad-\int_{V_i}\boldsymbol{\nabla}\cdot\Big(\sigma_u^{k-1}\Delta\boldsymbol{f}_u^{k-1}\otimes\sum_{m=1}^{n}\alpha_1^m\sigma_u^m\boldsymbol{f}_u^m\Big)dV,\\
&\qquad\qquad\qquad\alpha_3\int_{V_i}\boldsymbol{\nabla}\cdot(\sigma_u^n\Delta\boldsymbol{f}_u)dV=R_p^n,
\end{aligned}\right.\tag{3.7}$$

where the index $k-1$ is associated with the last computed increment $\sigma_u^{k-1}\Delta\boldsymbol{f}_u^{k-1}$ in the SIMPLE algorithm, see B. It is straightforward to observe that Equation (3.7) now features the same structure as the original Navier-Stokes problem (3.1) for which the SIMPLE algorithm is designed. Hence, the resulting solver for the spatial iteration does not require any modification of the existing OpenFOAM routines and is nonintrusive with respect to `simpleFoam`.

Remark 5. *The first integral in Equation* (3.7) *solely introduces a higher-order perturbation which, upon convergence of the SIMPLE iterations, is negligible. Hence, an alternative formulation of the momentum equation within the proposed PGD algorithm relies on neglecting such higher-order contribution. The convection term is thus linearised using the last computed approximation* $\sum_{m=1}^{n} \alpha_1^m \sigma_u^m \boldsymbol{f}_u^m$ *of the unknown velocity field, namely*

$$\begin{cases} \displaystyle\int_{V_i} \boldsymbol{\nabla}\cdot\Big(\sigma_u^n \Delta\boldsymbol{f}_u \otimes \sum_{m=1}^{n} \alpha_1^m \sigma_u^m \boldsymbol{f}_u^m\Big) dV \\ \displaystyle -\alpha_2 \int_{V_i} \boldsymbol{\nabla}\cdot(D\boldsymbol{\nabla}(\sigma_u^n \Delta\boldsymbol{f}_u)) dV + \alpha_3 \int_{V_i} \boldsymbol{\nabla}(\sigma_p^n \Delta f_p) dV = R_u^n \\ \displaystyle \qquad\qquad\qquad\qquad - \int_{V_i} \boldsymbol{\nabla}\cdot\Big(\sum_{m=1}^{n} \alpha_1^m \sigma_u^m \boldsymbol{f}_u^m \otimes \sigma_u^{k-1} \Delta\boldsymbol{f}_u^{k-1}\Big) dV, \\ \displaystyle \qquad\qquad\qquad\qquad \alpha_3 \int_{V_i} \boldsymbol{\nabla}\cdot(\sigma_u^n \Delta\boldsymbol{f}_u) dV = R_p^n. \end{cases} \tag{3.8}$$

The left-hand side of Equation (3.8) *mimics the SIMPLE strategy to solve the linearised version of the Navier-Stokes equations because it substitutes the unknown convection field with a relaxation. This approach converges to the same solution of* (3.7) *(and this has been verified numerically). Nevertheless it is slightly intrusive in the context of* ***simpleFoam*** *as it requires the implementation of the linear convection term. Hence, Equation* (3.7) *is preferred for the PGD spatial iteration.*

3.1.2 The parametric iteration

In the parametric step, the value of the previously computed spatial functions is fixed $(\boldsymbol{f}_u^n, f_p^n) \leftarrow (\sigma_u^n \boldsymbol{f}_u^n + \Delta \boldsymbol{f}_u, \sigma_p^n f_p^n + \Delta f_p)$ and the parametric increment $\Delta\phi$ acts as unknown. Within the *single parameter approximation* rationale, a unique scalar function depending on $\boldsymbol{\mu}$ is sought.

As noted in the previous Section, the correction introduced by the high-order term in the momentum equation is negligible upon convergence. Following the strategy described for the spatial iteration and neglecting this term in the restriction of Equation (3.2) to the parametric direction of the tangent manifold, $\Delta\phi$ is computed by solving the following algebraic equation

$$\left(\sum_{m=1}^{n} a_1^m \phi^m - a_2 \psi + a_3 \right) \Delta\phi = r_u^n + r_p^n, \tag{3.9}$$

where r_u^n and r_p^n are the parametric residuals associated with the discretisation of the momentum and mass equations, respectively, and each coefficient a_i, $i{=}1, \dots, 3$ depends solely on the spatial functions $(\sigma_u^n \boldsymbol{f}_u^n, \sigma_p^n f_p^n)$ and on the data of the problem, namely

$$\begin{cases} a_1^m := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \left[\boldsymbol{\nabla} \cdot (\sigma_u^n \boldsymbol{f}_u^n \otimes \sigma_u^m \boldsymbol{f}_u^m) \right] dV \\ \qquad\quad + \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \left[\boldsymbol{\nabla} \cdot (\sigma_u^m \boldsymbol{f}_u^m \otimes \sigma_u^n \boldsymbol{f}_u^n) \right] dV, \\ a_2 := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \left[\boldsymbol{\nabla} \cdot (D \boldsymbol{\nabla} (\sigma_u^n \boldsymbol{f}_u^n)) \right] dV, \\ a_3 := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \boldsymbol{\nabla} (\sigma_p^n f_p^n) \, dV + \int_{V_i} \sigma_p^n f_p^n \boldsymbol{\nabla} \cdot (\sigma_u^n \boldsymbol{f}_u^n) \, dV. \end{cases} \tag{3.10}$$

The unknown $\Delta\phi$ is discretised at the nodes of the parametric domain $\boldsymbol{\mathcal{I}}$ and the resulting algebraic equation is solved via a collocation method. Similarly to the spatial iteration, the separable form of (3.3)-(3.4) is exploited to perform computationally efficient pointwise evaluations of the residuals at the nodes of $\boldsymbol{\mathcal{I}}$. The complete derivation of the separated form of the right-hand side is detailed in A.

3.2 A nonintrusive implementation of the proper generalised decomposition in OpenFOAM

A critical aspect to make ROM strategies suitable for application on a daily basis in an industrial environment is their nonintrusiveness with respect to existing solution methodologies validated by companies. Thus, in order to solve parametrised flow problems using OpenFOAM, the proposed PGD algorithm is designed to be nonintrusive with respect to the `simpleFoam` solver as described above. As discussed in Section 2.3, inhomogeneous Dirichlet boundary conditions are treated by means of a spatial mode computed using the full-order solver, whereas the corresponding parametric mode is set equal to 1 (Algorithm 1 - Step 1). Then, the enrichment process is started and at each iteration of the alternating direction scheme a spatial mode is computed using `simpleFoam` (Algorithm 1 - Steps 7 to 10) and a linear system is solved to determine the corresponding parametric term (Algorithm 1 - Steps 11 to 14). The alternating direction iterations stop when the computed corrections $\varDelta \boldsymbol{f}_{\diamond}$, $\varDelta \phi$ are negligible with respect to the amplitudes $\sigma^n_{\diamond}$, σ_{ϕ} of the current mode for $\diamond = u, p$ and the residuals $\varepsilon^r_{\circ}$ are sufficiently small for $\circ = u, p, \phi$ (Algorithm 1 - Steps 6 and 15). The global enrichment strategy ends when the amplitude of the current mode $\sigma^n_{\diamond}$ is negligible with respect to the first one $\sigma^1_{\diamond}$ for $\diamond = u, p$ (Algorithm 1 - Step 3).

Remark 6. *Alternative criterions may be considered to stop the greedy algorithm, e.g. when the magnitude of the last mode normalised with respect to the sum of the amplitudes of all the computed terms is lower than a user-*

defined tolerance $\eta_\diamond^\star$, namely

$$\sigma_\diamond^n < \eta_\diamond^\star \sum_{m=1}^{n} \sigma_\diamond^m, \quad \textit{for } \diamond = u, p.$$

Algorithm 1 `pgdFoam`: a nonintrusive PGD implementation in OpenFOAM

```
Require: Tolerances $\eta_\diamond^\star$ for the greedy algorithm. Tolerances $\eta_\circ$ for the amplitudes and $\eta_\circ^r$ for the residuals in the alternating direction iteration. Typical values $\mathrm{typ}_\circ$ for the residuals of the spatial and parametric problems. $\diamond = u, p$ and $\circ = u, p, \phi$.
Compute boundary condition modes: the spatial mode is solution of (2.2) using simpleFoam and the parametric mode is equal to 1.
Set $n \leftarrow 1$ and initialise the amplitudes of the spatial modes $\sigma_\diamond^1 \leftarrow 1$.
while $\sigma_\diamond^n > \eta_\diamond^\star \sigma_\diamond^1$ do
    Set $k \leftarrow 0$, the parametric predictor $\phi^n \leftarrow 1$ and the spatial predictors $(\boldsymbol{f}_u^n, f_p^n)$ using the last computed modes.
    Initialise $\varepsilon_\circ \leftarrow 1$, $\varepsilon_\circ^r \leftarrow \mathrm{typ}_\circ$.
    while $\varepsilon_\circ > \eta_\circ$ or $\varepsilon_\circ^r > \eta_\circ^r$ do
        Compute the spatial residuals (A.9) and coefficients (3.6).
        Solve the spatial Navier-Stokes problem (3.7) using simpleFoam.
        Normalise the spatial predictors: $\sigma_\diamond^n \leftarrow \|\sigma_\diamond^n \boldsymbol{f}_\diamond^n + \Delta \boldsymbol{f}_\diamond\|$.
        Update the spatial predictors: $\boldsymbol{f}_\diamond^n \leftarrow (\sigma_\diamond^n \boldsymbol{f}_\diamond^n + \Delta \boldsymbol{f}_\diamond)/\sigma_\diamond^n$.
        Compute the parametric residual (A.11) and coefficients (3.10).
        Solve the parametric linear system (3.9).
        Normalise the parametric predictor: $\sigma_\phi \leftarrow \|\phi^n + \Delta\phi\|$.
        Update the parametric predictor: $\phi^n \leftarrow (\phi^n + \Delta\phi)/\sigma_\phi$.
        Update stopping criterions: $\varepsilon_\diamond \leftarrow \|\Delta \boldsymbol{f}_\diamond\|/\sigma_\diamond^n$, $\varepsilon_\phi \leftarrow \|\Delta\phi\|/\sigma_\phi$, $\varepsilon_\circ^r \leftarrow \|r_\circ\|$.
        Update the alternating direction iteration counter: $k \leftarrow k + 1$.
    end while
    Update the mode counter: $n \leftarrow n + 1$.
end while
```

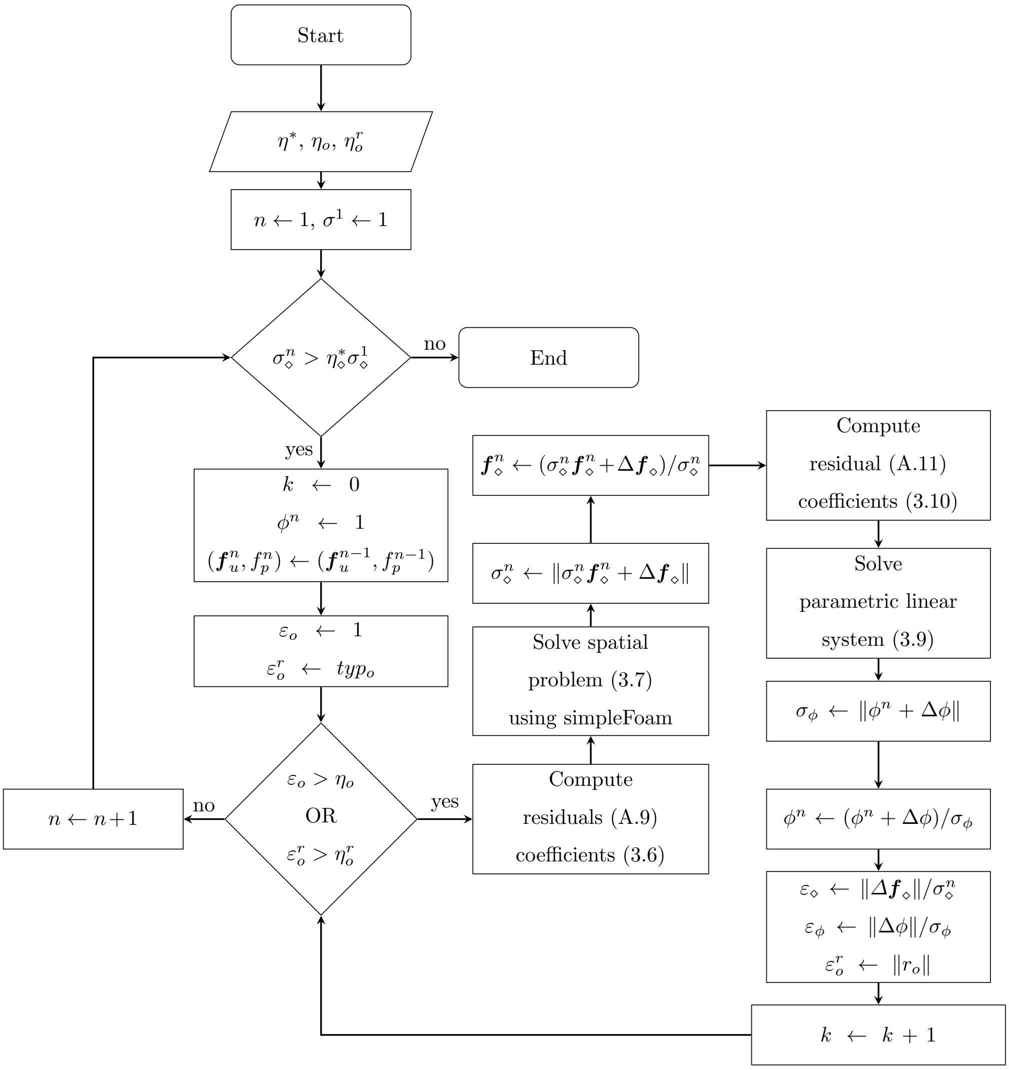

Start
η^*, η_o, η_o^r
$n \leftarrow 1$, $\sigma^1 \leftarrow 1$
$\sigma_\diamond^n > \eta_\diamond^* \sigma_\diamond^1$
no
End
yes
$k \leftarrow 0$
$\phi^n \leftarrow 1$
$(\boldsymbol{f}_u^n, f_p^n) \leftarrow (\boldsymbol{f}_u^{n-1}, f_p^{n-1})$
$\varepsilon_o \leftarrow 1$
$\varepsilon_o^r \leftarrow typ_o$
$\varepsilon_o > \eta_o$
OR
$\varepsilon_o^r > \eta_o^r$
no
$n \leftarrow n+1$
yes
Compute
residuals (A.9)
coefficients (3.6)
Solve spatial
problem (3.7)
using simpleFoam
$\sigma_\diamond^n \leftarrow \|\sigma_\diamond^n \boldsymbol{f}_\diamond^n + \Delta \boldsymbol{f}_\diamond\|$
$\boldsymbol{f}_\diamond^n \leftarrow (\sigma_\diamond^n \boldsymbol{f}_\diamond^n + \Delta \boldsymbol{f}_\diamond)/\sigma_\diamond^n$
Compute
residual (A.11)
coefficients (3.10)
Solve
parametric linear
system (3.9)
$\sigma_\phi \leftarrow \|\phi^n + \Delta\phi\|$
$\phi^n \leftarrow (\phi^n + \Delta\phi)/\sigma_\phi$
$\varepsilon_\diamond \leftarrow \|\Delta \boldsymbol{f}_\diamond\|/\sigma_\diamond^n$
$\varepsilon_\phi \leftarrow \|\Delta\phi\|/\sigma_\phi$
$\varepsilon_o^r \leftarrow \|r_o\|$
$k \leftarrow k+1$

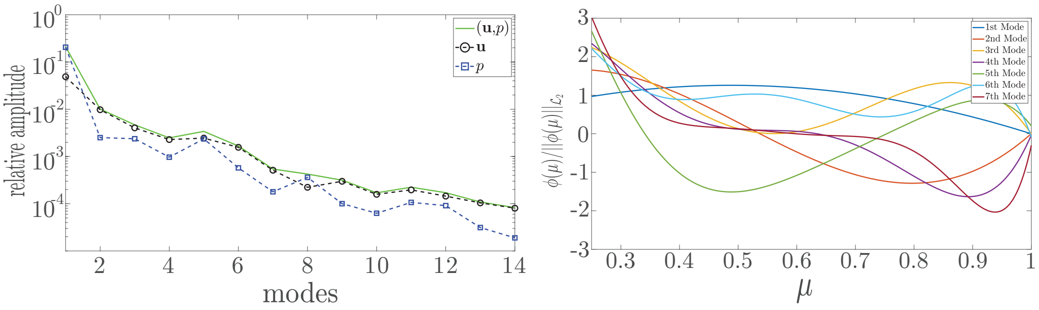
Figure 3.1: Flowchart of the nonintrusive `pgdFoam` algorithm. Legend: $\diamond = u, p$ and $\circ = u, p, \phi$.

3.3 Applications of the nonintrusive PGD methodology

In this section, the proposed methodology is applied to two- and three-dimensional geometries. First, a classical benchmark test for incompressible flow solvers, namely the nonleaky lid-driven cavity, is studied parametrising the imposed velocity of the lid in a range of values of the Reynolds number spanning from 1,000 to 4,000. Next, the nonintrusive PGD methodology is applied to laminar flow-control problems. The first problem is a two-dimensional case of academic interest, while the second is a real-life case provided by Volkswagen AG.

3.3.1 The cavity with parametrised lid velocity

In this section, the classical benchmark problem of the nonleaky lid-driven cavity is studied [39]. The unitary square $\Omega=[0,1]^2$ is considered as spatial domain and homogeneous Dirichlet boundary conditions are imposed on the lateral and bottom walls. On the top wall, a velocity $\boldsymbol{u}_{\text{lid}}(\boldsymbol{x},\mu)=400\mu\boldsymbol{u}_{\text{lid}}(\boldsymbol{x})$ is enforced, where the parameter $\mu \in [0.25,1]$ acts as a scaling factor of the maximum velocity of the lid, whereas $\boldsymbol{u}_{\text{lid}}(\boldsymbol{x})$ is a velocity profile featuring two ramps on the top-left and top-right corners of the domain to account for the change between null and maximum velocity. As classical in the literature treating the lid-driven cavity example, for $x \in [0,0.06]$ and $x \in [0.94,1]$, the horizontal component of the lid velocity changes linearly from 0 to 400μ and vice versa. The dynamic viscosity is set to $\nu=0.1\,\text{m}^2/\text{s}$ and the values



(a) Amplitude of the computed modes

(b) Computed parametric modes

Figure 3.2: PGD of the cavity flow with parametrised lid velocity. (a) Relative amplitude of the computed modes $\boldsymbol{f}_u^m$ (black), f_p^m (blue) and the combined amplitude of $(\boldsymbol{f}_u^m, f_p^m)$ according to Equation (**??**). (b) First seven normalised computed parametric modes.

considered for the Reynolds number span from 1,000 to 4,000.

The mode handling the boundary conditions is obtained as a full-order solution of the Navier-Stokes equations using the `simpleFoam` algorithm for a lid velocity computed using μ=1, that is for a maximum horizontal velocity of 400 m/s. The corresponding parametric boundary condition mode is set to be linearly evolving from μ=0.25 to μ=1, that is $\phi(\mu)$=μ.

Following the rationale described in the previous section, two different stopping criterions are considered for the PGD enrichment strategy, namely $\eta_{(u,p)} \leq 10^{-3}$ and $\eta_{(u,p)} \leq 10^{-4}$. Figure 3.2(a) displays the relative amplitude of the computed modes. Note that the first stopping point is achieved after seven computed modes, whereas fourteen terms are required to fulfil the lower tolerance. The corresponding computed parametric modes are presented on Figure 3.2(b). It is worth noting that all the computed parametric modes are close or equal to 0 for μ=1. This is due to the fact that the boundary conditions of the problem are imposed by means of a full-order solution

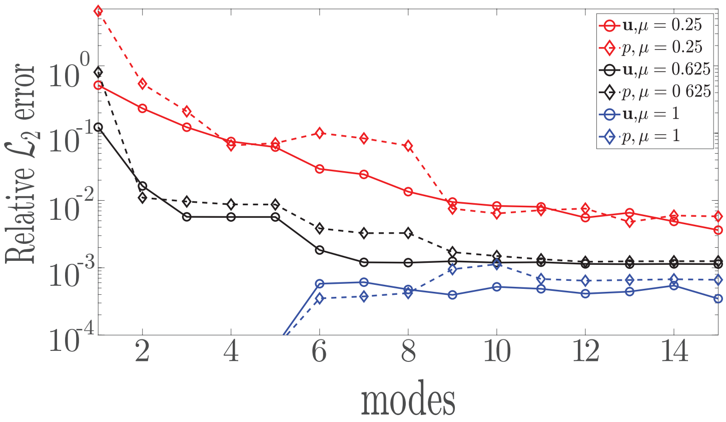


Figure 3.3: Relative $\mathcal{L}_2(\Omega)$ errors of the PGD approximation of the cavity flow with parametrised lid velocity with respect to the full-order solution as a function of the global number of modes (i.e. boundary conditions and computed) utilised in the PGD expansion.

computed for the maximum value of μ in the parametric space. Hence, the case of μ=1 is accurately described by the PGD approximation using solely the mode obtained via `simpleFoam`.

Now, the online evaluations of the PGD approximation of the velocity and pressure fields for different values of the parameter μ are compared to the full-order solutions computed using `simpleFoam`. The corresponding relative $\mathcal{L}_2(\Omega)$ errors are presented in Figure 3.3 as a function of the number of modes utilised in the PGD approximation. The boundary condition mode and the first seven computed modes, for which $\eta_{(u,p)} \leq 10^{-3}$, provide a good approximation of both velocity and pressure and limited corrections are introduced by the following modes until the stopping criterion of 10^{-4} is fulfilled. For the case μ=1, a small error of the order of 10^{-4} appears starting from the fifth computed mode, i.e. n=6. This is due to the fact that the boundary condition mode already captures all the features of the flow, being

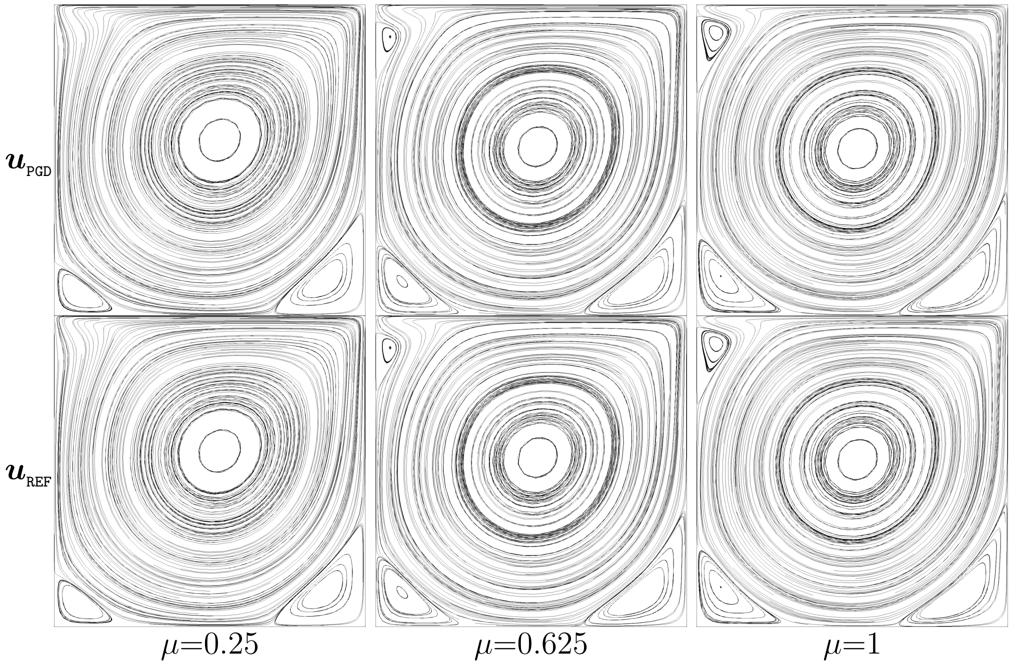


Figure 3.4: Comparison of the PGD approximation (top) and the full-order solution (bottom) of the parametrised lid-driven cavity problem for μ=0.25, μ=0.625 and μ=1, corresponding to a maximum velocity of the lid of 100, 250 and 400 m/s, respectively.

a full-order solution of the Navier-Stokes equations as previously mentioned.

A qualitative comparison of the reduced-order and full-order solutions of the parametrised lid-driven cavity problem is displayed in Figure 3.4. The PGD approximations for μ=0.25, μ=0.625 and μ=1 are presented as long as their corresponding simulations obtained using `simpleFoam`. The cases under analysis are associated with a maximum horizontal velocity of the lid of 100, 250 and 400 m/s, respectively. It is worth noting that `pgdFoam` is able to capture the topological changes of the flow with great accuracy, managing to identify location and size of the vortices, as well as their appearance and disappearance according to the values of the Reynolds number considered in the analysis.

3.3.2 Applications to parametrised laminar flow control problems

Dynamically controlling the features of a flow is a challenging problem with several high-impact applications including, e.g., drag minimisation, stall control and aerodynamic noise reduction [34, 28, 51]. A major bottleneck to the design of flow control devices is represented by the large number of simulations involved in the tuning of the control loop. In this section, the potential of the described nonintrusive PGD implementation in OpenFOAM is demonstrated for parametrised flow control problems. Two- and three-dimensional internal flows with jets are studied. Specifically, a parametric study involving the peak velocity of the jets as extra-coordinate of the problem is considered to test the proposed PGD methodology.

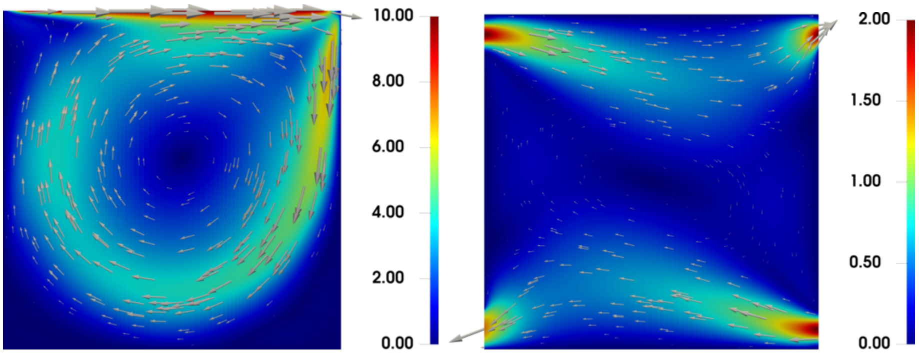
3.3.2.1 Lid-driven cavity with parametrised jet velocity

Consider the nonleaky lid-driven cavity problem introduced in Section 3.3.1. The lid velocity is defined with two linear ramps, increasing from 0 to 10 m/s on the top-left corner and decreasing correspondingly on the top-right one. Three jets of size 0.12 m are introduced on the vertical walls, two on the right wall and one on the left, respectively. The parametrised velocity of the jets is $\boldsymbol{u}_{\text{jet}}(\boldsymbol{x},\mu)=\mu\boldsymbol{u}_{\text{jet}}(\boldsymbol{x})$, where the maximum velocity is controlled by the parameter $\mu \in [0,1]$ and the profile $\boldsymbol{u}_{\text{jet}}(\boldsymbol{x})$ is defined as

$$\boldsymbol{u}_{\text{jet}}(x,y)=\begin{cases}\Big(-1+\cos\big(2\pi y/0.12\big),0\Big) & \text{for } x=1,\ y\in[0,0.12],\\ \Big(1-\cos\big(2\pi(y-0.88)/0.12\big),0\Big) & \text{for } x=1,\ y\in[0.88,1],\\ \Big(1-\cos\big(2\pi(y-0.88)/0.12\big),0\Big) & \text{for } x=0,\ y\in[0.88,1].\end{cases} \tag{3.11}$$

An outlet boundary is added on the left vertical wall for $y \in [0,0.12]$ and a free-traction condition is enforced. The dynamic viscosity is set to $\nu{=}0.01\,\text{m}^2/\text{s}$, therefore the corresponding Reynolds number is Re=1,000.

The boundary conditions of the problem are enforced through two modes computed as full-order solutions via `simpleFoam` as shown in Figure 3.5: the first one, for $\mu{=}0$, corresponds to lid velocity of 10 m/s and inactive jets; the second one, for $\mu{=}1$, is associated with the maximum velocity of the jets and a zero velocity of the lid. The corresponding parametric modes for the boundary conditions are set to $\phi(\mu){=}1$ and $\phi(\mu){=}\mu$, respectively.



(a) B.C. mode for the lid (b) B.C. mode for the jets

Figure 3.5: Cavity flow with parametrised jet velocity. Spatial boundary condition modes for velocity.

Following the rationale previously discussed, the PGD enrichment process is stopped when $\eta_{(u,p)} \leq 10^{-4}$. In Figure 3.6, the generalised solution computed by the PGD is interpolated in different points of the parametric space under analysis and compared with the corresponding full-order solutions provided by `simpleFoam`. The flows for μ=0.1, μ=0.3, μ=0.5, μ=0.7, μ=0.8 and μ=1 are displayed, covering a wide range of flow regimes in the cavity. It is worth noting that the discussed reduced-order strategy is able to capture the topological changes in the flow features and accurately reproduce the appearance and disappearance of vortices localised in different regions of the domain.

The accuracy of the PGD approximation with respect to the full-order solution is also verified by computing the relative $\mathcal{L}_2(\Omega)$ error of the spatial discretisation while enriching the modal description of the solution. More precisely, Figure 3.7 shows that using two boundary condition modes and six computed modes all approximations achieve a plateau with relative errors of 10^{-2} or lower.

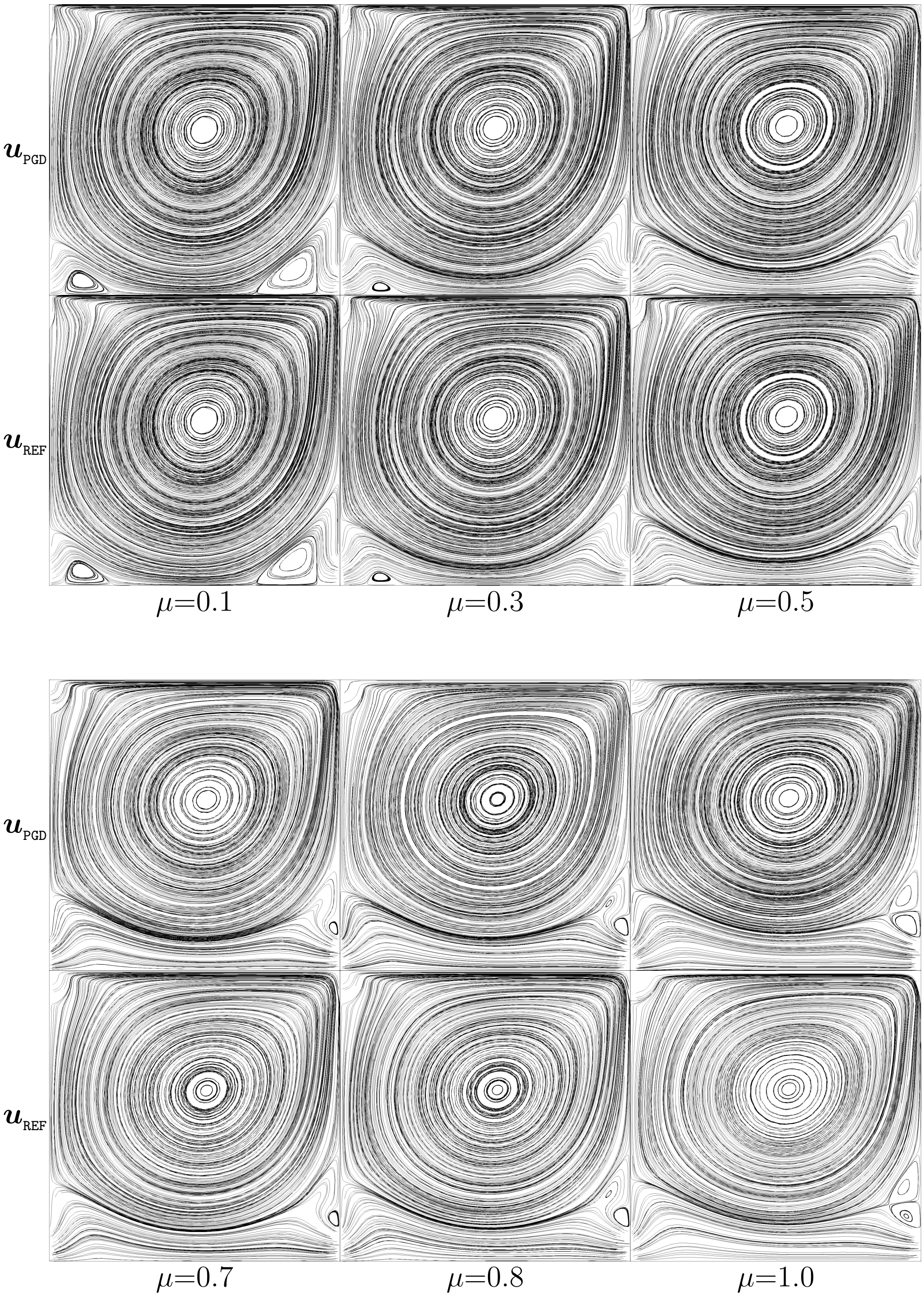


Figure 3.6: Comparison of the PGD approximation (top) and the full-order solution (bottom) of the lid-driven cavity with jets for μ=0.1, μ=0.3, μ=0.5, μ=0.7, μ=0.8 and μ=1, corresponding to a maximum velocity of the jets spanning from 0.2 to 2 m/s.

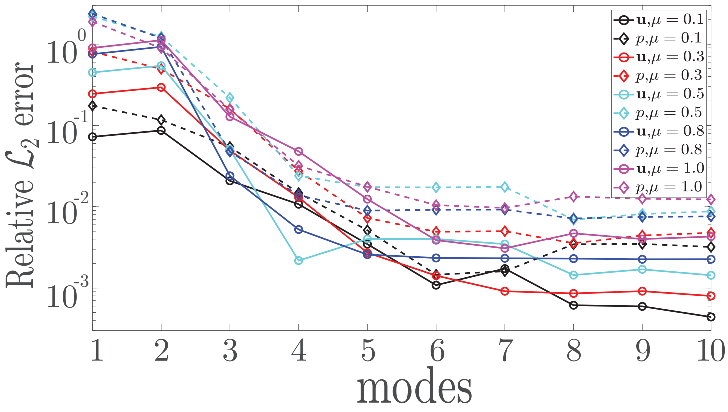


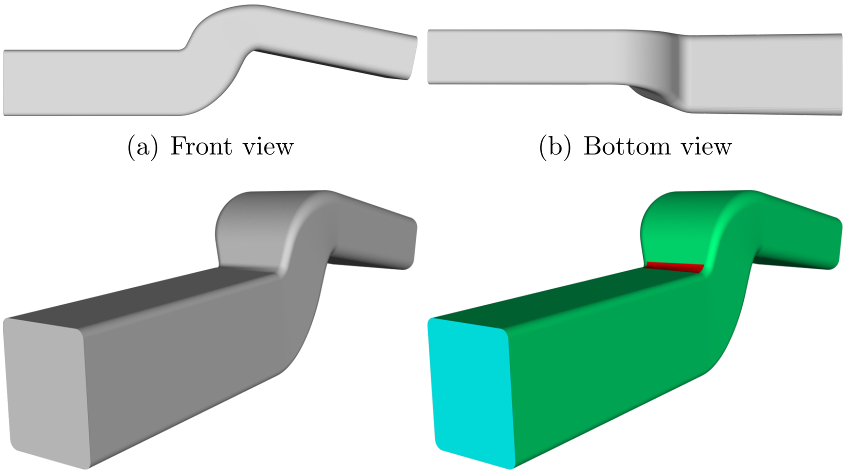
Figure 3.7: Relative $\mathcal{L}_2(\Omega)$ error of the PGD approximation of the cavity flow with parametrised jet velocity with respect to the full-order solution as a function of the global number of modes (i.e. boundary conditions and computed) utilised in the PGD expansion, for μ=0.1, μ=0.5, μ=0.7 and μ=1.

3.3.2.2 S-Bend with flow control driven by a jet

In this section, the proposed PGD methodology is applied to a flow control problem using a three-dimensional geometry of industrial interest. The model of a heating, ventilation and air conditioning (HVAC) duct section provided by Volkswagen AG is shown in Figure 3.8. A jet is introduced on the red patch, at the first bend of the duct. The velocity profile of the jet is a sinusoidal function defined on the reference planar square $[0, 1]^2$ as

$$u_{\hat{y}}(\hat{x}, \hat{z}) = 0.0375(1 - \cos(-2\pi\hat{x}))(1 - \cos(2\pi\hat{z})) \tag{3.12}$$

and pointing in the direction $\hat{y}$ orthogonal to the plane $(\hat{x}, \hat{z})$. The parametrisation is constructed as a scaling of the jet velocity from u_y=$-0.015\,\text{m/s}$, i.e. blowing, to suction with u_y=$0.15\,\text{m/s}$. A single parameter μ is introduced and the parametric domain considered for the analysis is $\mathcal{I}$=$[-0.1, 1]$. Note that



(a) Front view

(b) Bottom view

(c) Perspective view

(d) Patches of the duct

Figure 3.8: Geometrical model of the S-Bend. On the bottom-right image, the jet patch is highlighted in red.

this problem is especially challenging due to the change of sign in the interval of parametric values considered leading to different physical phenomena. The remaining boundary conditions feature homogeneous velocity on all the lateral walls, a parabolic velocity profile with mean value $\boldsymbol{u}$=$(0.83, 0, 0)$m/s on the inlet and a free-traction on the outlet. The dynamic viscosity is set to ν=$1.588 \times 10^{-5}\,\mathrm{m}^2/\mathrm{s}$ and the corresponding value of the Reynolds number is Re=280. The quantity of interest in this problem is the pressure drop computed along the duct.

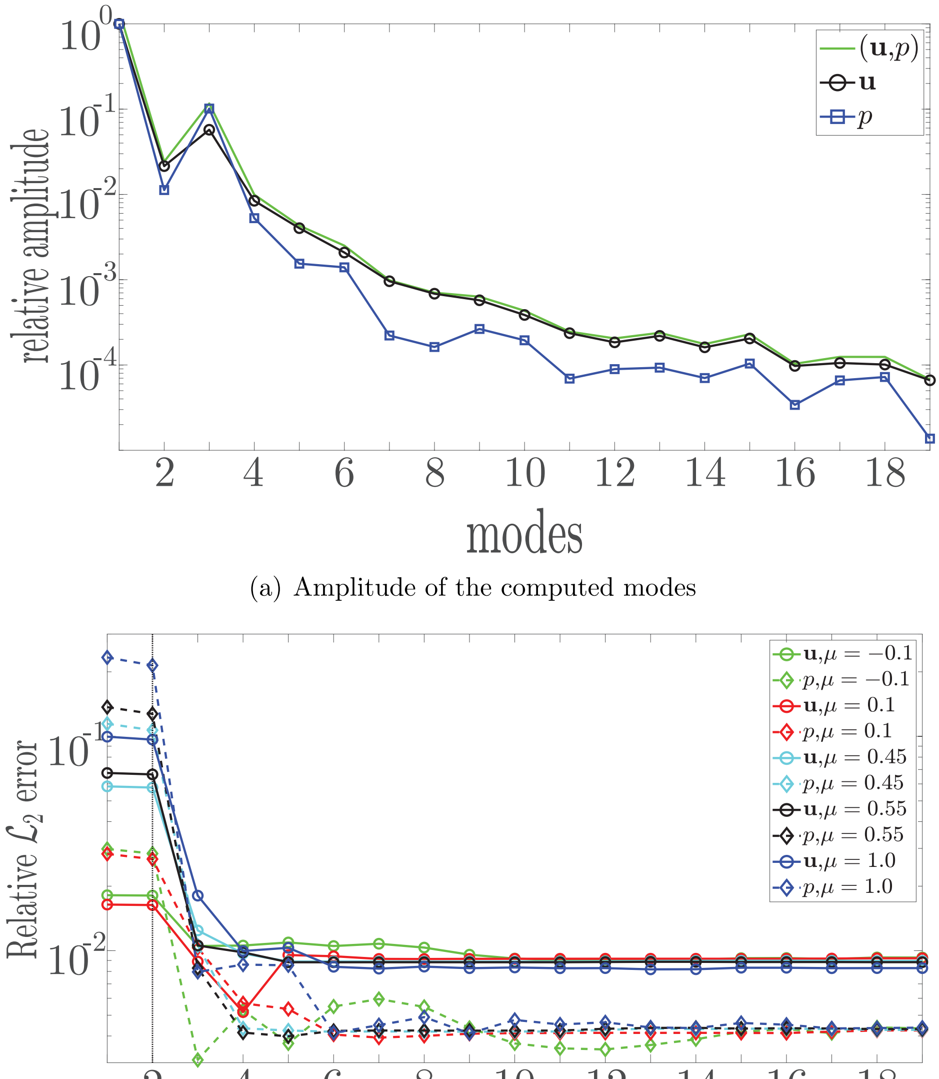
As previously done for the lid-driven cavity with jets, two modes to account for the boundary conditions are computed using `simpleFoam`. The first mode is a full-order solution corresponding to the case of inactive jet and given inlet parabolic profile; the second one, is obtained setting a zero inlet velocity and a jet of maximum velocity u_y=$0.15\,\mathrm{m/s}$. The corresponding parametric modes are $\phi(\mu)$=1 and $\phi(\mu)$=μ, respectively.

Setting a tolerance of 10^{-3}, `pgdFoam` computes five modes before fulfilling the stopping criterion for $\eta_{(u,p)}$, see Equation (2.29), whereas seventeen computed modes are required for the amplitude to drop at 10^{-4}, as displayed in Figure 3.9(a).

The PGD approximation obtained is compared with the full-order solutions given by `simpleFoam` for the values $\mu{=}-0.1$, $\mu{=}0.45$ and $\mu{=}1$ of the parameter under analysis. In Figure 3.9(b), the relative $\mathcal{L}_2(\Omega)$ error for these configurations is reported. The numerical experiments confirm that an accuracy of 10^{-2} is achieved, for all parameters, using two computed modes additionnally to the two terms accounting for the boundary conditions. Note that the first computed mode is one order of magnitude more relevant than the following ones (Fig. 3.9(a)) and after two computed modes only limited corrections are introduced to the existing PGD approximation. It is worth observing that a comparable accuracy of the PGD solution is achieved throughout the whole parametric space for both velocity and pressure.

A qualitative comparison of the pressure and velocity fields computed using the PGD solution interpolated in different points of the parametric interval $\boldsymbol{\mathcal{I}}$ and the corresponding full-order discretisations is presented in Figures 3.10 and 3.11.

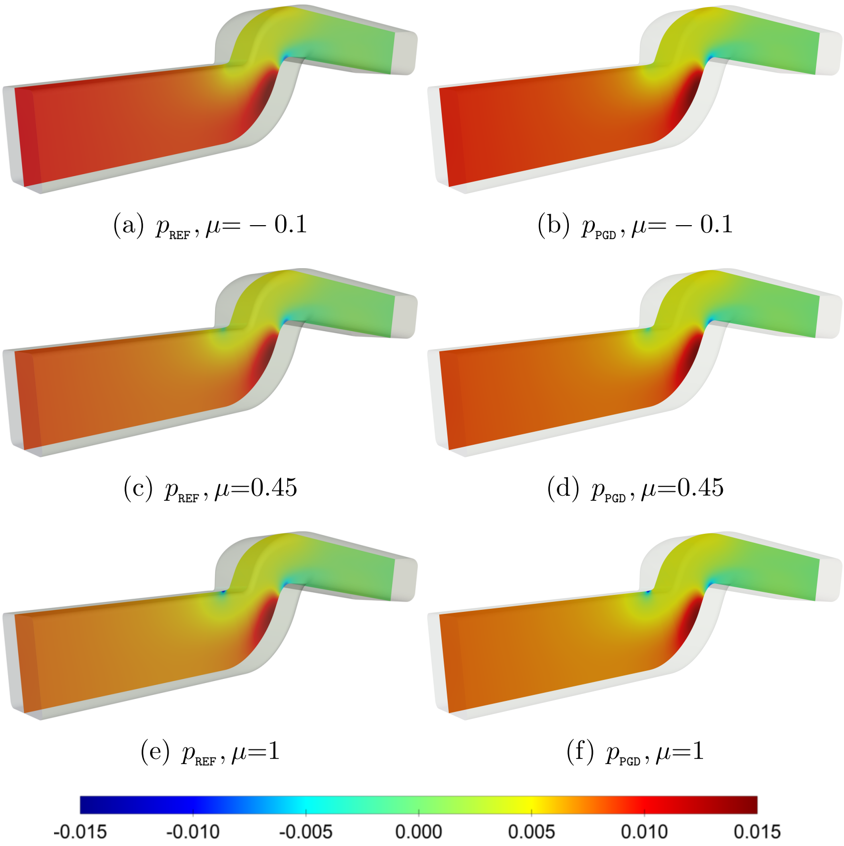
As mentioned at the beginning of this section, the quantity of engineering interest in the analysis of this problem is the pressure drop computed along the duct. Considering a zero reference pressure on the outlet, the weighted



(a) Amplitude of the computed modes

(b) Relative $\mathcal{L}_2(\Omega)$ error

Figure 3.9: Internal flow in the S-Bend with parametrised jet velocity. (a) Relative amplitude of the computed modes $\boldsymbol{f}_u^m$ (black), f_p^m (blue) and the combined amplitude of $(\boldsymbol{f}_u^m, f_p^m)$ according to Equation (**??**). (b) Relative $\mathcal{L}_2(\Omega)$ error of the PGD approximations of pressure and velocity with respect to the full-order solutions for different values of μ.



(a) $p_{\text{REF}}, \mu{=}-0.1$

(b) $p_{\text{PGD}}, \mu{=}-0.1$

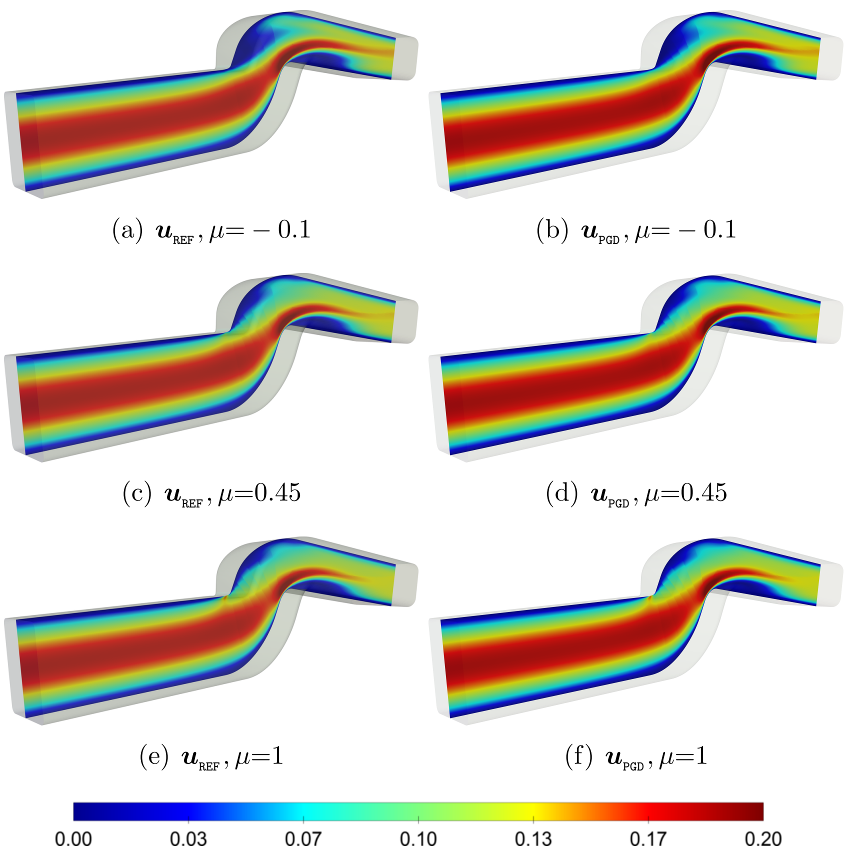
(c) $p_{\text{REF}}, \mu{=}0.45$

(d) $p_{\text{PGD}}, \mu{=}0.45$

(e) $p_{\text{REF}}, \mu{=}1$

(f) $p_{\text{PGD}}, \mu{=}1$

Figure 3.10: Comparison of the PGD approximation (right) and the full-order solution (left) of the pressure field of the internal flow in the S-Bend with a jet configuration of $\mu{=}-0.1$, $\mu{=}0.45$ and $\mu{=}1$, corresponding to a jet which spans from blowing at maximum velocity 0.015 m/s to suction with peak velocity 0.15 m/s.



(a) $\boldsymbol{u}_{\text{REF}}, \mu{=}-0.1$

(b) $\boldsymbol{u}_{\text{PGD}}, \mu{=}-0.1$

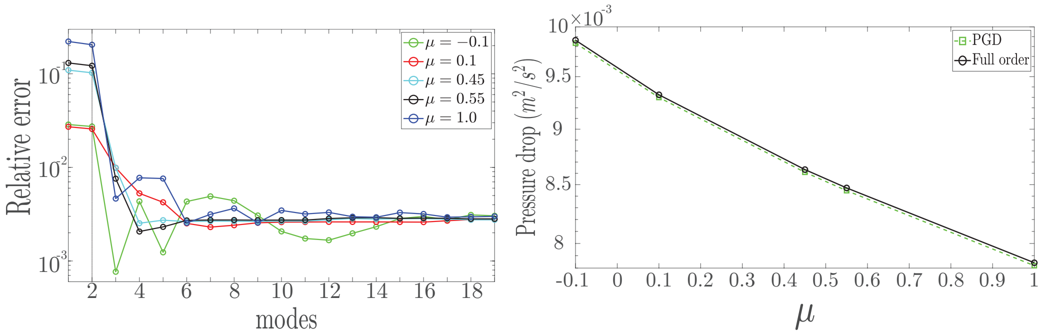
(c) $\boldsymbol{u}_{\text{REF}}, \mu{=}0.45$

(d) $\boldsymbol{u}_{\text{PGD}}, \mu{=}0.45$

(e) $\boldsymbol{u}_{\text{REF}}, \mu{=}1$

(f) $\boldsymbol{u}_{\text{PGD}}, \mu{=}1$

Figure 3.11: Comparison of the PGD approximation (right) and the full-order solution (left) of the velocity field of the internal flow in the S-Bend with a jet configuration of $\mu{=}-0.1$, $\mu{=}0.45$ and $\mu{=}1$, corresponding to a jet which spans from blowing at maximum velocity 0.015 m/s to suction with peak velocity 0.15 m/s.



(a) Error of the pressure drop

(b) Pressure drop

Figure 3.12: PGD approximation of the internal flow in the S-Bend with parametrised jet velocity. (a) Relative error of the pressure drop enriching the PGD modal approximation. (b) Pressure drop with respect to the maximum jet velocity.

average pressure drop is defined as

$$p_{\text{drop}} := \frac{1}{A_{\text{in}}} \sum_{i=1}^{N_{\text{in}}} A_i p_i, \tag{3.13}$$

where A_{in} is the area of the inlet surface, N_{in} the number of faces S_i, $i{=}1,\dots,N_{\text{in}}$ on the inlet patch and p_i, A_i are the pressure and area on the face S_i, respectively. For $\mu{=}-0.1$, $\mu{=}0.1$, $\mu{=}0.45$, $\mu{=}0.55$ and $\mu{=}1$, the pressure drop is evaluated interpolating the generalised PGD solution in the corresponding values of the parametric space and using the full-order solver `simpleFoam`. Figure 3.12(a) presents the convergence history of the error in the pressure drop as a function of the number of modes in the PGD approximation. It is straightforward to observe again that using the modes accounting for the boundary conditions and two computed modes is sufficient to capture the flow features of a wide range of parameters, with a maximum error in the pressure drop below 10^{-2}. Moreover, by comparing the pressure drop with respect to the maximum velocity of the jet for different configurations with the corresponding values provided by the full-order solver, the capability of the discussed reduced-order strategy to accurately capture the evolution of a quantity of interest throughout the range of values of the parameter μ is confirmed (Fig. 3.12(b)). Finally, these experiments highlight that the proposed PGD algorithm is able to provide an accurate approximation of the solution both in terms of a global measure of the error in the domain, see the $\mathcal{L}_2$ error in Figure 3.9(b), and of a localised quantity of interest as the pressure drop in Figure 3.12(a).

Chapter 4

The PGD rationale for the Spalart-Allmaras turbulence model

Industrial cases typically reside in the turbulent regime. It is therefore critical to devise a PGD framework suitable for application to flow problems characterised by high Reynolds number. Applying the nonintrusive PGD methodology detailed in Chapter 3 requires knowledge of the effects of the parameters on turbulence. Therefore, focusing on building a fully *a priori* method using PGD, a strategy for the efficient computation of a separable approximation of the turbulent eddy viscosity is required.

In this chapter, firstly, the Spalart-Allmaras turbulence model is recalled and a PGD strategy for the construction of a separable Spalart-Allmaras viscosity field is described. Then a method to efficiently implement the computation of the Spalart-Allmaras in a PGD enrichment process is proposed. Finally, two- and three-dimensional test cases are solved for the validation of the proposed method.

4.1 The Spalart-Allmaras turbulence model

In a daily industrial environment where robustness, stability and speed are of the utmost importance, Reynolds-Averaged-Navier-Stokes methods are widely popular. In the RANS framework turbulence is introduced via the turbulent viscosity coefficient ν_t, and the steady incompressible turbulent Navier-Stokes system now reads

$$\begin{cases} \boldsymbol{\nabla}\cdot(\boldsymbol{u}\otimes\boldsymbol{u}) - \boldsymbol{\nabla}\cdot((\nu+\nu_t)\boldsymbol{\nabla}\boldsymbol{u}) + \boldsymbol{\nabla}p = \boldsymbol{s} & \text{in } \Omega, \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 & \text{in } \Omega, \\ \boldsymbol{u} = \boldsymbol{u}_D & \text{on } \Gamma_D, \\ \boldsymbol{n}\cdot(\nu\boldsymbol{\nabla}\boldsymbol{u} - p\mathbf{I}_d) = \boldsymbol{t} & \text{on } \Gamma_N \end{cases} \tag{4.1}$$

Considering an effective viscosity $\nu_{eff} := \nu + \nu_t$ the similarity between laminar and turbulent Navier-Stokes system becomes more apparent. The sole change is the extra step required to evaluate the effective viscosity at each step (Appendix B). As aforementioned, for a variety of external aerodynamics cases the turbulence model of choice is the Spalart-Allmaras [102].

The Spalart-Allmaras turbulence model, is a single equation model, meaning it offers a convenient entry point to building a PGD solver. Under some common assumptions in the automotive field, the classical Spalart-Allmaras model is simplified, leading to the following steady transport equation for $\tilde{\nu}$

$$\begin{cases} \boldsymbol{\nabla} \cdot (\tilde{\nu} \otimes \boldsymbol{u}) - \dfrac{1}{\sigma} \boldsymbol{\nabla} \cdot [(\nu + \tilde{\nu}) \boldsymbol{\nabla} \tilde{\nu}] - \dfrac{C_{b2}}{\sigma} |\boldsymbol{\nabla} \tilde{\nu}|^2 & \\ \qquad\qquad = C_{b1} \tilde{S} \tilde{\nu} - C_{w1} f_w \left(\dfrac{\tilde{\nu}}{y} \right)^2 & \text{in } \Omega, \\ \tilde{\nu} = \tilde{\nu}_D & \text{on } \Gamma_D, \\ \boldsymbol{n} \cdot \boldsymbol{\nabla} \tilde{\nu} = 0 & \text{on } \Gamma_N, \end{cases} \tag{4.2}$$

where $\tilde{\nu}$ is a fictitious viscosity, from now on refered to as state viscosity, $\tilde{\nu}_D$ is the state viscosity datum on the Dirichlet boundaries and y is the distance to the closest wall. The terms $\tilde{S}$ and f_w are given by

$$\tilde{S} = S + f_{v2}(\chi, f_{v1}) \frac{\tilde{\nu}}{(\kappa y)^2} , \quad f_w = g \sqrt[6]{\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6}} ,$$

with

$$\begin{aligned} g &= r + C_{w2} \left(r^6 - r \right) , & r &= \frac{\tilde{\nu}}{\tilde{S} (\kappa y)^2} , \\ S &= \sqrt{2 \Omega^2} , & \Omega &= \frac{1}{2} \left(\nabla \boldsymbol{u} - \nabla \boldsymbol{u}^T \right) , \\ f_{v2} &= 1 - \frac{\chi}{1 + \chi f_{v1}} , & f_{v1} &= \frac{\chi^3}{\chi^3 + C_{v1}^3} \\ \chi &= \frac{\tilde{\nu}}{\nu} \end{aligned}$$

and coefficients $\sigma = \frac{2}{3}$, $\kappa = 0.41$, $C_{b1} = 0.1355$, $C_{b2} = 0.622$, $C_{v1} = 7.1$, $C_{w2} = 0.3$, $C_{w3} = 2$ and $C_s = 0.3$. It is here noted that the OpenFOAM implementation omits the tripping term f_{t2}.

The turbulent eddy viscosity is computed in terms of $\tilde{\nu}$ using the expression

$$\nu_t = \tilde{\nu} f_{v1} \ . \tag{4.3}$$

For the sake of readability in the coming sections Equation 4.2 is broken down to components. The first term represents convection, the second and third diffusion and lastly, the first and second terms on the right-hand side represent a production and destruction term, respectively.

4.2 The PGD rationale for the SA model

The concept of constructing a separable approximation of the state viscosity is similar to the one described in Chapters 2 and 3. Assuming the same discretisation as in Chapter 2 the high-dimensional form of the steady Spalart-Allmaras equation reads

$$\begin{aligned}&\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot(\tilde{\nu}\otimes\boldsymbol{u})\,dV\,d\boldsymbol{\mathcal{I}}-\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{1}{\sigma}\boldsymbol{\nabla}\cdot[(\nu+\tilde{\nu})\boldsymbol{\nabla}\tilde{\nu}]\,dV\,d\boldsymbol{\mathcal{I}}\\ &-\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{C_{b2}}{\sigma}|\boldsymbol{\nabla}\tilde{\nu}|^2\,dV\,d\boldsymbol{\mathcal{I}}=\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}C_{b1}\tilde{S}\tilde{\nu}\,dV\,d\boldsymbol{\mathcal{I}}-\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}C_{w1}f_w\left(\frac{\tilde{\nu}}{y}\right)^2\,dV\,d\boldsymbol{\mathcal{I}}\end{aligned}\tag{4.4}$$

Considering a single parameter μ affecting the turbulent viscosity, the separable form of $\tilde{\nu}$ using m modes reads

$$\widetilde{\nu}^{m}_{\text{PGD}}=\widetilde{\nu}^{m-1}_{\text{PGD}}(\boldsymbol{x},\mu)+\sigma^m_\nu\,B^m(\boldsymbol{x})\,\xi^m(\mu)\ ,$$

and employing a predictor-corrector scheme yields

$$\widetilde{\nu}^{m}_{\text{PGD}}=\bar{\nu}^{\,m}_{\text{PGD}}+\sigma^m_\nu\delta\bar{\nu}^{\,m}_{\text{PGD}}=\widetilde{\nu}^{m-1}_{\text{PGD}}+\sigma^m_\nu B^m\xi^m+\sigma^m_\nu\delta\bar{\nu}^{\,m}_{\text{PGD}},\tag{4.5}$$

where $\bar{\nu}^m_{\text{PGD}}$ and $\delta\bar{\nu}^m_{\text{PGD}}$ notes the prediction of the current approximation and its corresponding variation, not to be confused with $\widetilde{\nu}^m_{\text{PGD}}$, the final prediction after convergence of m modes.

For the sake of readability and without loss of generality, a constant physical viscosity ν independent from the problem parameters is considered.

Plugging (4.5) in (4.4), and using the PGD approximation for velocity, the separable weak form of the Spalart-Allmaras equation is derived in parts. Note that n and m, which are the number of modes used for the separation of velocity and state viscosity, respectively, are not necessarily the same. Separation of the convection term yields

$$\begin{aligned}
&\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot(\tilde{\nu}\otimes\boldsymbol{u})\,dV\,d\boldsymbol{\mathcal{I}}\approx \\
&\quad\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot(\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\otimes\widetilde{\boldsymbol{u}}^n_{\text{PGD}})\,dV\,d\boldsymbol{\mathcal{I}}+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot\left(\bar{\nu}^m_{\text{PGD}}\otimes\widetilde{\boldsymbol{u}}^n_{\text{PGD}}\right)\,dV\,d\boldsymbol{\mathcal{I}}\ ,
\end{aligned}\tag{4.6}$$

separation of the two diffusion terms yields

$$\begin{aligned}
&\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{1}{\sigma}\boldsymbol{\nabla}\cdot((\nu+\tilde{\nu})\boldsymbol{\nabla}\tilde{\nu})\,dV\,d\boldsymbol{\mathcal{I}}\approx \\
&\quad\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{1}{\sigma}\boldsymbol{\nabla}\cdot\left(\nu\boldsymbol{\nabla}\left(\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\right)\right)\,dV\,d\boldsymbol{\mathcal{I}} \\
&\quad+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{1}{\sigma}\boldsymbol{\nabla}\cdot\left(\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\boldsymbol{\nabla}\left(\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\right)\right)\,dV\,d\boldsymbol{\mathcal{I}} \\
&\quad+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{1}{\sigma}\boldsymbol{\nabla}\cdot\left(\bar{\nu}^m_{\text{PGD}}\boldsymbol{\nabla}\left(\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\right)\right)\,dV\,d\boldsymbol{\mathcal{I}} \\
&\quad+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{1}{\sigma}\boldsymbol{\nabla}\cdot\left(\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\boldsymbol{\nabla}\bar{\nu}^m_{\text{PGD}}\right)\,dV\,d\boldsymbol{\mathcal{I}} \\
&\quad+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{1}{\sigma}\boldsymbol{\nabla}\cdot\left(\left(\nu+\bar{\nu}^m_{\text{PGD}}\right)\boldsymbol{\nabla}\bar{\nu}^m_{\text{PGD}}\right)\,dV\,d\boldsymbol{\mathcal{I}}\ ,
\end{aligned}\tag{4.7a}$$

$$\begin{aligned}&\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{C_{b2}}{\sigma}|\boldsymbol{\nabla}\tilde{\nu}|^2\,dV\,d\boldsymbol{\mathcal{I}}\approx\\&\quad\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{C_{b2}}{\sigma}|\boldsymbol{\nabla}\left(\sigma_{\nu}^{m}\delta\bar{\nu}_{\text{PGD}}^{m}\right)|^2\,dV\,d\boldsymbol{\mathcal{I}}+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}2\frac{C_{b2}}{\sigma}\boldsymbol{\nabla}\left(\sigma_{\nu}^{m}\delta\bar{\nu}_{\text{PGD}}^{m}\right)\cdot\boldsymbol{\nabla}\bar{\nu}_{\text{PGD}}^{m}\,dV\,d\boldsymbol{\mathcal{I}}\\&\quad+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{C_{b2}}{\sigma}\boldsymbol{\nabla}\bar{\nu}_{\text{PGD}}^{m}\cdot\boldsymbol{\nabla}\bar{\nu}_{\text{PGD}}^{m}\,dV\,d\boldsymbol{\mathcal{I}},\end{aligned}\tag{4.7b}$$

and finally, separation of the production and destruction terms yields

$$\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}C_{b1}\tilde{S}\tilde{\nu}\,dV\,d\boldsymbol{\mathcal{I}}\approx\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}C_{b1}\tilde{S}\sigma_{\nu}^{m}\delta\bar{\nu}_{\text{PGD}}^{m}\,dV\,d\boldsymbol{\mathcal{I}}+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}C_{b1}\tilde{S}\bar{\nu}_{\text{PGD}}^{m}\,dV\,d\boldsymbol{\mathcal{I}},\tag{4.8a}$$

$$\begin{aligned}&\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}C_{w1}f_w\left(\frac{\tilde{\nu}}{y}\right)^2\,dV\,d\boldsymbol{\mathcal{I}}\approx\\&\quad\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}C_{w1}f_w\left(\frac{\sigma_{\nu}^{m}\delta\bar{\nu}_{\text{PGD}}^{m}}{y}\right)^2\,dV\,d\boldsymbol{\mathcal{I}}+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}2C_{w1}f_w\frac{\sigma_{\nu}^{m}\delta\bar{\nu}_{\text{PGD}}^{m}\,\bar{\nu}_{\text{PGD}}^{m}}{y^2}\,dV\,d\boldsymbol{\mathcal{I}}\\&\quad+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}C_{w1}f_w\left(\frac{\bar{\nu}_{\text{PGD}}^{m}}{y}\right)^2\,dV\,d\boldsymbol{\mathcal{I}}.\end{aligned}\tag{4.8b}$$

Plugging the Equations 4.6-4.8b into Equation 4.4 and keeping the variation terms on the left-hand side while transferring to the right-hand side all terms depending solely on the current prediction of the state viscosity, yields the separable form of the high-dimensional SA turbulence model

$$
\begin{aligned}
&\int_{\boldsymbol{\mathcal{I}}}\int_{V_i} \boldsymbol{\nabla}\cdot(\sigma^m_\nu \delta\bar{\nu}^m_{\text{PGD}}\otimes\tilde{\boldsymbol{u}}^n_{\text{PGD}})\,dV\,d\boldsymbol{\mathcal{I}} - \int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{1}{\sigma}\boldsymbol{\nabla}\cdot\left(\nu\boldsymbol{\nabla}(\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}})\right)\,dV\,d\boldsymbol{\mathcal{I}} \\
&-\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{1}{\sigma}\boldsymbol{\nabla}\cdot\left(\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\,\boldsymbol{\nabla}\left(\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\right)\right)\,dV\,d\boldsymbol{\mathcal{I}} \\
&-\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{1}{\sigma}\boldsymbol{\nabla}\cdot\left(\bar{\nu}^m_{\text{PGD}}\,\boldsymbol{\nabla}\left(\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\right)\right)\,dV\,d\boldsymbol{\mathcal{I}} \\
&-\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{1}{\sigma}\boldsymbol{\nabla}\cdot\left(\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\,\boldsymbol{\nabla}\bar{\nu}^m_{\text{PGD}}\right)\,dV\,d\boldsymbol{\mathcal{I}} - \int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{C_{b2}}{\sigma}|\boldsymbol{\nabla}\left(\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\right)|^2\,dV\,d\boldsymbol{\mathcal{I}} \\
&-\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}2\frac{C_{b2}}{\sigma}\boldsymbol{\nabla}\left(\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\right)\cdot\boldsymbol{\nabla}\bar{\nu}^m_{\text{PGD}}\,dV\,d\boldsymbol{\mathcal{I}} - \int_{\boldsymbol{\mathcal{I}}}\int_{V_i}C_{b1}\tilde{S}\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\,dV\,d\boldsymbol{\mathcal{I}} \\
&+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}C_{w1}f_w\left(\frac{\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}}{y}\right)^2\,dV\,d\boldsymbol{\mathcal{I}} + \int_{\boldsymbol{\mathcal{I}}}\int_{V_i}2C_{w1}f_w\frac{\sigma^m_\nu\delta\bar{\nu}^m_{\text{PGD}}\,\bar{\nu}^m_{\text{PGD}}}{y^2}\,dV\,d\boldsymbol{\mathcal{I}} = R^m_\nu,
\end{aligned}
\tag{4.9}
$$

where the residual is defined as

$$
\begin{aligned}
R^m_\nu :=\mathcal{R}_\nu :=\ &\mathcal{R}_\nu(\tilde{\boldsymbol{u}}^n_{\text{PGD}},\tilde{\nu}^{m-1}_{\text{PGD}},\sigma^m_\nu\bar{\nu}^m_{\text{PGD}}) \\
=&-\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\boldsymbol{\nabla}\cdot\left(\bar{\nu}^m_{\text{PGD}}\otimes\tilde{\boldsymbol{u}}^n_{\text{PGD}}\right)\,dV\,d\boldsymbol{\mathcal{I}} \\
&+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{1}{\sigma}\boldsymbol{\nabla}\cdot\left(\left(\nu+\bar{\nu}^m_{\text{PGD}}\right)\boldsymbol{\nabla}\bar{\nu}^m_{\text{PGD}}\right)\,dV\,d\boldsymbol{\mathcal{I}} \\
&+\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}\frac{C_{b2}}{\sigma}\boldsymbol{\nabla}\bar{\nu}^m_{\text{PGD}}\cdot\boldsymbol{\nabla}\bar{\nu}^m_{\text{PGD}}\,dV\,d\boldsymbol{\mathcal{I}} + \int_{\boldsymbol{\mathcal{I}}}\int_{V_i}C_{b1}\tilde{S}\bar{\nu}^m_{\text{PGD}}\,dV\,d\boldsymbol{\mathcal{I}} \\
&-\int_{\boldsymbol{\mathcal{I}}}\int_{V_i}C_{w1}f_w\left(\frac{\bar{\nu}^m_{\text{PGD}}}{y}\right)^2\,dV\,d\boldsymbol{\mathcal{I}}.
\end{aligned}
\tag{4.10}
$$

Remark 7. *From equations (4.9) and (4.10) can be observed that no separable form has been assumed for the terms f_w and $\tilde{S}$. The expressions for f_w and $\tilde{S}$, as recalled in Section 4.2, feature a non-linear dependence to the parameter, making them inefficient to reconstruct in a separable form. Thus a relaxation of the Spalart-Allmaras model is considered where they depend* *solely on the last known approximation of viscosity. What is more, due to their complexity it is more efficient at this point to reconstruct and use their high-dimensional form, $\overline{f}_w(\boldsymbol{x},\mu)$ and $\overline{S}(\boldsymbol{x},\mu)$, rather than attempt a separation that would result in extra loss of accuracy. More precisely, using the last known state viscosity PGD approximation the process of assembling the high-dimensional form of the aforementioned terms reads*

$$\overline{S} = S + \overline{f}_{v2}\frac{\widetilde{\nu}^{m-1}_{\scriptscriptstyle PGD}}{(\kappa y)^2}\,,\quad \overline{f}_w = \overline{g}\sqrt[6]{\frac{1+C_{w3}^6}{\overline{g}^6+C_{w3}^6}}\,, \tag{4.11}$$

where

$$\overline{f}_{v2} = 1 - \frac{\overline{\chi}}{1+\overline{\chi}\,\overline{f}_{v1}}\,, \qquad \overline{f}_{v1} = \frac{\overline{\chi}^3}{\overline{\chi}^3 + C_{v1}^3}\,, \qquad \overline{\chi} = \frac{\widetilde{\nu}^{m-1}_{\scriptscriptstyle PGD}}{\nu}$$

and

$$\overline{g} = \overline{r} + C_{w2}\overline{r}^6 - \overline{r}\,,\quad \overline{r} = \frac{\widetilde{\nu}^{m-1}_{\scriptscriptstyle PGD}}{\overline{S}(\kappa y)^2}.$$

Similarly, upon convergence, the high-dimensional approximation of the turbulent viscosity from the separable state viscosity (Eq.(4.3)) is assembled as follows

$$\nu_{t_{pgd}} = \tilde{f}_{v1}\widetilde{\nu}^{m}_{\scriptscriptstyle PGD}\,, \tag{4.12}$$

with

$$\tilde{f}_{v1} = \frac{\tilde{\chi}^3}{\tilde{\chi}^3 + C_{v1}^3}\,, \qquad \tilde{\chi} = \frac{\widetilde{\nu}^{m}_{\scriptscriptstyle PGD}}{\nu}.$$

Note that in the formulations above, bar variables note relaxations of the Spalart-Allmaras model, whereas tilde variables express the dependence on the converged PGD approximation and should not be confused. In the following *sections, for the sake of readability, whenever $\tilde{S}$ and f_w are used in the PGD formulation, they represent the relaxation terms $\overline{S}$ and $\overline{f}_w$.*

4.2.1 Spatial iteration

In classical alternating direction fashion Equation (4.9) is restricted to the spatial direction of the tangent manifold yielding the PDE of Equation (4.13). Using the numerical schemes implemented in OpenFOAM the 5^{th} term of (4.9) is handled explicitly as part of the residual of (4.13). The residual R_ν^m can be efficiently computed using the separable form of the already computed approximation (Appendix A).

$$\begin{aligned}
&\int_{V_i} \boldsymbol{\nabla} \cdot \left(\sigma_\nu^m \Delta B^m \otimes \sum_{k=1}^{n} \alpha_1^k \sigma_u^k \boldsymbol{f}_u^k \right) dV \\
&\quad - \beta_3 \int_{V_i} \frac{1}{\sigma} \boldsymbol{\nabla} \cdot \left(\nu \boldsymbol{\nabla} \left(\sigma_\nu^m \Delta B^m \right) \right) dV \\
&\quad - \beta_2 \int_{V_i} \frac{1}{\sigma} \boldsymbol{\nabla} \cdot \left(\sigma_\nu^m \Delta B^m \boldsymbol{\nabla} \left(\sigma_\nu^m \Delta B^m \right) \right) dV \\
&\quad - \int_{V_i} \frac{1}{\sigma} \boldsymbol{\nabla} \cdot \left(\sum_{j=1}^{m} \beta_4^j \sigma_\nu^j B^j \boldsymbol{\nabla} \left(\sigma_\nu^m \Delta B^m \right) \right) dV \\
&\quad - \beta_2 \int_{V_i} \frac{C_{b2}}{\sigma} \left| \boldsymbol{\nabla} \left(\sigma_\nu^m \Delta B^m \right) \right|^2 dV \\
&\quad - 2 \int_{V_i} \frac{C_{b2}}{\sigma} \boldsymbol{\nabla} \left(\sigma_\nu^m \Delta B^m \right) \cdot \boldsymbol{\nabla} \left(\sum_{j=1}^{m} \beta_4^j \sigma_\nu^j B^j \right) dV \\
&\quad - \beta_5 \int_{V_i} C_{b1} \sigma_\nu^m \Delta B^m \, dV + \beta_6 \int_{V_i} C_{w1} \left(\frac{\sigma_\nu^m \Delta B^m}{y} \right)^2 dV \\
&\quad + 2 \int_{V_i} C_w \frac{\sigma_\nu^m \Delta B^m}{y^2} \sum_{j=1}^{m} \beta_7^j \sigma_\nu^j B^j \, dV = R_\nu^m ,
\end{aligned} \tag{4.13}$$

where the coefficients β_i with $i = 1, \dots, 7$ are

$$
\begin{aligned}
\beta_1^k &= \int_{\boldsymbol{\mathcal{I}}} (\xi^m)^2 \, \phi^k \, d\boldsymbol{\mathcal{I}}, & \beta_2 &= \int_{\boldsymbol{\mathcal{I}}} (\xi^m)^3 \, d\boldsymbol{\mathcal{I}}, \\
\beta_3 &= \int_{\boldsymbol{\mathcal{I}}} (\xi^m)^2 \, d\boldsymbol{\mathcal{I}}, & \beta_4^j &= \int_{\boldsymbol{\mathcal{I}}} (\xi^m)^2 \, \xi^j \, d\boldsymbol{\mathcal{I}}, \\
\beta_5 &= \int_{\boldsymbol{\mathcal{I}}} \tilde{S} \, (\xi^m)^2 \, d\boldsymbol{\mathcal{I}}, & \beta_6 &= \int_{\boldsymbol{\mathcal{I}}} f_w \, (\xi^m)^3 \, d\boldsymbol{\mathcal{I}}, \\
\beta_7^j &= \int_{\boldsymbol{\mathcal{I}}} f_w \, (\xi^m)^2 \, \xi^j \, d\boldsymbol{\mathcal{I}},
\end{aligned}
\tag{4.14}
$$

with $j = 1, \dots, m$ and $k = 1, \dots, n$. Due to the high-dimensional form of f_w and $\tilde{S}$ (Remark 7), coefficients β_5, β_6 and β_7 represent spatial functions.

Remark 8. *The second, fifth and eighth terms in equation (4.13) represent high-order variations. Omitting these terms leads to stability issues.*

4.2.2 Parametric iteration

Fixing the computed spatial mode and restricting (4.9) to the parametric direction of the tangent manifold, the parametric increment is computed from the following algebraic equation

$$
\begin{aligned}
&- \left(\frac{1}{\sigma} b_2 + \frac{C_{b2}}{\sigma} b_6 - C_{w1} b_9 \right) (\Delta\xi^m)^2 \\
&+ \left(\sum_{k=1}^{n} b_1^k \phi^k - \frac{1}{\sigma} \left(b_3 + \sum_{j=1}^{m} \left(b_4^j \xi^j + b_5^j \xi^j \right) \right) \right) \Delta\xi^m \\
&- \left(2 \sum_{j=1}^{m} \frac{C_{b2}}{\sigma} b_7^j \xi^j + C_{b1} b_8 - 2 C_{w1} \sum_{j=1}^{m} b_{10}^j \xi^j \right) \Delta\xi^m = r_\nu^m,
\end{aligned}
\tag{4.15}
$$

where the coefficients b_i with $i = 1, \ldots, 10$ are computed as follows

$$
\begin{aligned}
b_1^k &= \int_{V_i} \sigma_{\nu}^{m} B^m \boldsymbol{\nabla} \cdot \left(\sigma_{\nu}^{m} B^m \otimes \sigma_u^k \boldsymbol{f}_u^k \right) \, dV, \\
b_2 &= \int_{V_i} \sigma_{\nu}^{m} B^m \boldsymbol{\nabla} \cdot \left(\sigma_{\nu}^{m} B^m \boldsymbol{\nabla} \left(\sigma_{\nu}^{m} B^m \right) \right) \, dV, \\
b_3 &= \int_{V_i} \sigma_{\nu}^{m} B^m \boldsymbol{\nabla} \cdot \left(\nu \boldsymbol{\nabla} \left(\sigma_{\nu}^{m} B^m \right) \right) \, dV, \\
b_4^j &= \int_{V_i} \sigma_{\nu}^{m} B^m \boldsymbol{\nabla} \cdot \left(\sigma_{\nu}^{j} B^j \boldsymbol{\nabla} \left(\sigma_{\nu}^{m} B^m \right) \right) \, dV, \\
b_5^j &= \int_{V_i} \sigma_{\nu}^{m} B^m \boldsymbol{\nabla} \cdot \left(\sigma_{\nu}^{m} B^m \boldsymbol{\nabla} \left(\sigma_{\nu}^{j} B^j \right) \right) \, dV, \\
b_6 &= \int_{V_i} \sigma_{\nu}^{m} B^m |\boldsymbol{\nabla} \left(\sigma_{\nu}^{m} B^m \right)|^2 \, dV, \\
b_7^j &= \int_{V_i} \sigma_{\nu}^{m} B^m \boldsymbol{\nabla} \left(\sigma_{\nu}^{m} B^m \right) \cdot \boldsymbol{\nabla} \left(\sigma_{\nu}^{j} B^j \right) \, dV, \\
b_8 &= \int_{V_i} \sigma_{\nu}^{m} B^m \tilde{S} \sigma_{\nu}^{m} B^m \, dV, \\
b_9 &= \int_{V_i} \sigma_{\nu}^{m} B^m f_w \left(\frac{\sigma_{\nu}^{m} B^m}{y} \right)^2 \, dV, \\
b_{10}^j &= \int_{V_i} \sigma_{\nu}^{m} B^m f_w \frac{\sigma_{\nu}^{m} B^m \sigma_{\nu}^{j} B^j}{y^2} \, dV.
\end{aligned}
\tag{4.16}
$$

Remark 7 is recalled for integrals including f_w and $\tilde{S}$, therefore coefficients b_8, b_9 and b_{10} represent parametric functions. The residual is described in detail in Appendix A.

4.3 Implementing turbulence in the nonintrusive PGD scheme in OpenFOAM

The structure of the PGD solver for the Spalart-Allmaras turbulence model is similar to the one presented for the laminar Navier-Stokes system in 3.2. The 0^{th} mode will impose the boundary conditions as is common in the PGD methodology. In this work, the state viscosity $\tilde{\nu}$ boundary condition modes are acquired from the respective boundary condition modes for velocity and pressure. The enrichment process is then initiated and at each iteration of the alternating direction scheme a spatial and parametric mode is computed (Algorithm 2 - Steps 7 to 14). The alternating direction iterations stop when the computed corrections $\varDelta B$, $\varDelta\xi$ are negligible with respect to the amplitudes σ_ν^m, σ_ξ of the current mode (Algorithm 2 - Steps 6 and 15). Similar to the laminar Navier-Stokes problem, the global enrichment strategy ends when the amplitude of the current mode σ_ν^m is negligible with respect to the first one σ_ν^1 (Algorithm 2 - Step 3). Remark 6 is recalled to remind that various valid stopping criteria can be used.

Implementing Algorithm 2 in the non-intrusive PGD scheme introduced in Section 3.2 requires but a few adaptions of Algorithm 1, owing to the addition of the parameter dependent turbulent viscosity coefficient. As aforementioned, in this work, the high-dimensional form of the turbulent viscosity is reconstructed, as it simplifies the implementation and the computation procedure.

Since the PGD strategy introduced during the study of the steady lam-

Algorithm 2 `pgdSA`: an OpenFOAM-based PGD library to solve the parametric Spalart-Allmaras turbulence model

```
Require: Tolerances $\eta_\nu^\star$ for the greedy algorithm. Tolerances $\eta_\circ$ for the
    amplitudes and $\eta_\circ^r$ for the residuals in the alternating direction iteration.
    $\circ = B, \xi$.
Compute boundary condition modes: the spatial mode is solution of (4.2)
    corresponding to the simpleFoam solution for the velocity and pressure
    mode and the parametric mode is equal to 1.
Set $m \leftarrow 1$ and initialise the amplitudes of the spatial mode $\sigma_\nu^1 \leftarrow 1$.
while $\sigma_\nu^m > \eta_\nu^\star \sigma_\nu^1$ do
    Set $k \leftarrow 0$, the parametric predictor $\xi^m \leftarrow 1$ and the spatial predictor
        $B^m$ using the last computed modes.
    Initialise $\varepsilon_\circ \leftarrow 1$, $\varepsilon_\circ^r \leftarrow \mathrm{typ}_\circ$.
    while $\varepsilon_\circ > \eta_\circ$ or $\varepsilon_\circ^r > \eta_\circ^r$ do
        Compute the spatial residuals (A.13) and coefficients (4.14).
        Solve the spatial PDE (4.13).
        Normalise the spatial predictors: $\sigma_\nu^m \leftarrow \|\sigma_\nu^m B^m + \varDelta B\|$.
        Update the spatial predictors: $B^m \leftarrow (\sigma_\nu^m B^m + \varDelta B)/\sigma_\nu^m$.
        Compute the parametric residual (A.15) and coefficients (4.16).
        Solve the parametric linear system (4.15).
        Normalise the parametric predictor: $\sigma_\xi \leftarrow \|\xi^m + \varDelta\xi\|$.
        Update the parametric predictor: $\xi^m \leftarrow (\xi^m + \varDelta\xi)/\sigma_\xi$.
        Update stopping criterions: $\varepsilon_\diamond \leftarrow \|\varDelta B\|/\sigma_\nu^m$, $\varepsilon_\xi \leftarrow \|\varDelta\xi\|/\sigma_\xi$, $\varepsilon_\circ^r \leftarrow \|r_\circ\|$.
        Update the alternating direction iteration counter: $k \leftarrow k + 1$.
    end while
    Update the mode counter: $m \leftarrow m + 1$.
end while
```

inar Navier-Stokes remains largely the unaffected, only required changes to the diffusion term are recalled. The artificial diffusion term in the spatial iteration of the laminar PGD strategy (3.5) is modified to include the coefficient dependent on turbulence. More specifically, the new diffusion term reads

$$-\int_{V_i} \boldsymbol{\nabla}\cdot [(\alpha_3 D + \alpha_{\tilde{\nu}})\boldsymbol{\nabla}\,(\sigma_u^n \Delta \boldsymbol{f}_u)]\; dV \tag{4.17}$$

where $\alpha_{\tilde{\nu}}$ is the spatial function

$$\alpha_{\tilde{\nu}}(\boldsymbol{x}) := \int_{\boldsymbol{\mathcal{I}}} (\phi^n)^2\, \nu_t \, d\boldsymbol{\mathcal{I}}$$

and ν_t is computed as described in Equation (4.12), using the PGD approximation of the turbulent viscosity described in Algorithm 2.

Similarly the residual (Appendix A.3) will now include the following extra diffusion term

$$\sum_{m=1}^{n} \int_{V_i} \boldsymbol{\nabla}\cdot (\alpha_{\tilde{\nu}}^m \boldsymbol{\nabla}\,(\sigma_u^m \boldsymbol{f}_u^m))\; dV \tag{4.18}$$

where $\alpha_{\tilde{\nu}}$ is the spatial function

$$\alpha_{\tilde{\nu}}^m(\boldsymbol{x}) := \int_{\boldsymbol{\mathcal{I}}} \phi^n \phi^m \nu_{t_{pgd}}\, d\boldsymbol{\mathcal{I}}.$$

The parametric direction is treated similarly with an extra term defined on the left-hand and right-hand side of Equation (3.9), respectively

$$\begin{aligned} \mathcal{LHS} &: -\, a_{\tilde{\nu}} \Delta\phi\ , \\ \mathcal{RHS} &: \sum_{m=1}^{n} a_{\tilde{\nu}}^m \phi^m , \end{aligned} \tag{4.19}$$

where $a_{\tilde{\nu}}$ and $a_{\tilde{\nu}}^m$ are the following parametric functions

$$
\begin{aligned}
a_{\tilde{\nu}}(\mu) &:= \int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \left[\boldsymbol{\nabla} \cdot \left(\nu_{t_{pgd}} \boldsymbol{\nabla} \left(\sigma_u^n \boldsymbol{f}_u^n \right) \right) \right] dV \ , \\
a_{\tilde{\nu}}^m(\mu) &:= \sum_{m=1}^{n} \int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \left[\boldsymbol{\nabla} \cdot \left(\nu_{t_{pgd}} \boldsymbol{\nabla} \left(\sigma_u^m \boldsymbol{f}_u^m \right) \right) \right] dV \ .
\end{aligned}
$$

The PGD solver structure for the turbulent Navier-Stokes flow problem aims to imitate the segregated approach of the SIMPLE algorithm when coupled with a turbulence model. The first step consists of computing velocity and pressure. In a subsequent step and using the computed velocity field the turbulence model is called to update the turbulent viscosity. Finally, the algorithm loops back to the computation of a velocity and pressure solution, using the new turbulent viscosity. The PGD methodology emulates this structure by enriching the velocity and pressure approximation until a user-prescribed, exponentially decreasing relative amplitude criterion has been met. Once the relative amplitude drops below said threshold, the SA PGD solver (Algorithm 2) is called to computed a new approximation for the turbulent viscosity. Using the new approximation for viscosity, the Navier-Stokes PGD is reset and called to compute velocity and pressure modes until the new amplitude criterion is met. For the sake of simplicity, let us call the Navier-Stokes PGD solver *PGD-NS* and the Spalart-Allmaras PGD solver *PGD-SA*.

Remark 9. *The number of turbulent viscosity updates and effectively the cost and accuracy of the method is controlled via the prescribed threshold of the relative amplitude. In this work the threshold is exponentially decreased,* *initialised at 10% and dropped by 1 order of magnitude with each passing update*

$$\eta_{turb} = 10^{-(i_t+1)} \quad \textit{for } i_t \geq 0 \tag{4.20}$$

where i_t is the iterator of SA PGD calls. That has proven to be an effective compromise between cost and accuracy, while smaller steps lead to better stability.

This strategy is described in Algorithm 3, where the steps of Algorithms 1 and 2 have been condensed for the sake of readability.

Algorithm 3 turbulent `pgdFoam`: a PGD solver for the parametric turbulent Navier-Stokes problem

Require: Tolerances of Algorithms 1 and 2. $\diamond = u, p, B$ and $\circ = u, p, B, \phi, \xi$

```
Compute boundary condition modes: the spatial mode is solution of (4.1) using simpleFoam and the Spalart-Allmaras turbulence model and the parametric modes are equal to 1.
Set $n \leftarrow 1$, $m \leftarrow 1$ and initialise the amplitudes of the spatial modes $\sigma_\diamond^1 \leftarrow 1$. Initialise viscosity update counter $i_t = 0$.
while $\sigma_\diamond^n > \eta_\diamond^\star \sigma_\diamond^1$ do
    Compute $\sigma_u^n \boldsymbol{f}_u^n, \sigma_p^n f_p^n, \phi^n$ using PGD-NS
    if $\sigma_\diamond^n < \eta_{turb}$ then
        Call PGD-SA library 2. Update $\widetilde{\nu}_{\text{PGD}}$ approximation.
        Increment viscosity update counter $i_t \leftarrow i_t + 1$.
        Update the node counter: $n \leftarrow 0$.
    end if
    Update the mode counter: $n \leftarrow n + 1$.
end while
```

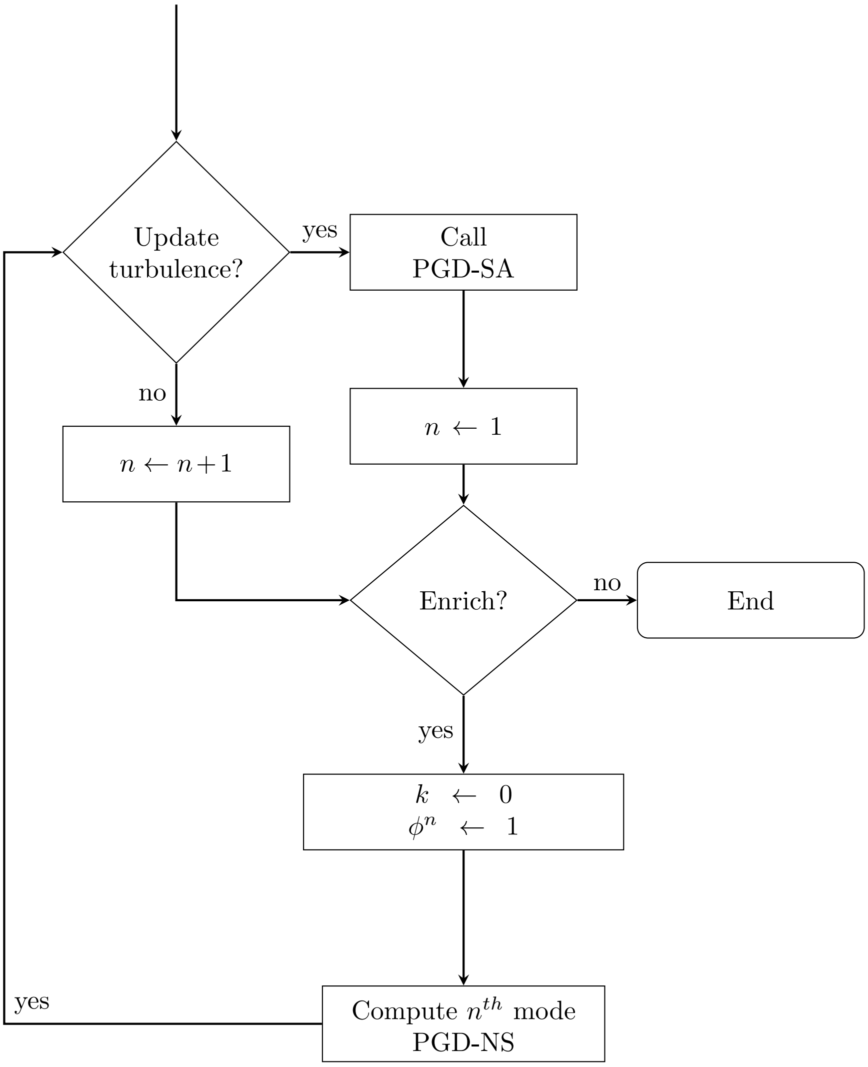


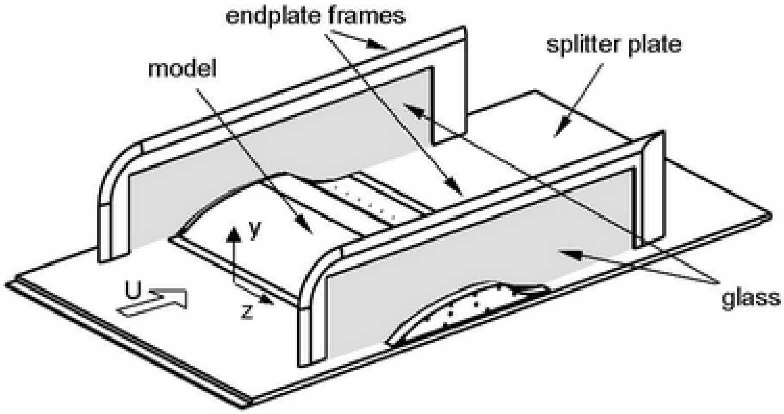
Figure 4.1: Flowchart of the structure to approximation parametrised turbulent Navier-Stokes flow via PGD (Algorithm 3).

4.4 Applications of turbulent Navier-Stokes flow-control problems

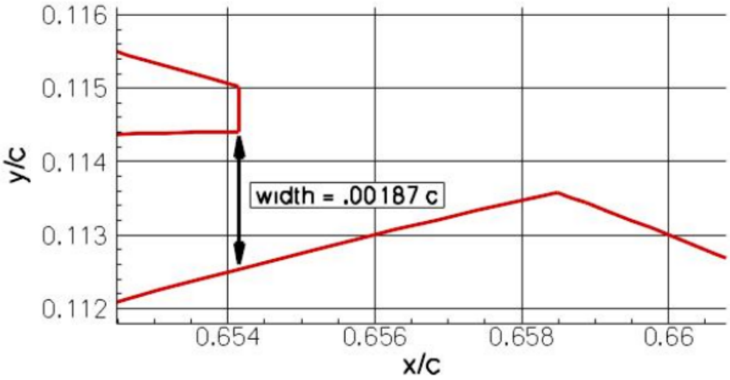
In this section the PGD methodology described in Section 4.3 is applied to turbulent flow control problems. More specifically, the NASA wall-mounted hump is studied, a turbulence modelling validation test case. The experimental set-up consists of a Glauert-Goldschmied type body mounted between two endplates on a splitter plate (Fig. 4.2(a)). As was studied in [98] the characteristic length is the length of the bump ($c = 0.42$ m). Flow separation control is introduced via a suction jet acting through a plenum leading to a slot at 65% of the chord (Fig. 4.2(b)). The proposed jet mass flow rate is 0.01518kg/s throughout the experimental 5.842-meter-span slot. With an inlet flow velocity of 34.6 m/s and a kinematic viscosity $\nu = 1.55274 \cdot 10^{-5}$ m^2/s the Reynolds number is approximately $Re = 936,000$.

The quantity of interest is the effect of the jet on the flow separation and reattachment points. Both experimental and CFD validation tests have been run [46, 96], showcasing the quasi-two-dimensional nature of the problem, despite the small three-dimensional effects introduced by the endplates. It has also been discussed that RANS solutions, regardless of the turbulence model used, tend to overestimate the recirculation bubble showing reattachment positions between 10 and 20% further downstream than the experiment [96].

The following sections introduce the parametric case, as well as a two- and three-dimensional variant of the NASA wall mounted hump which will be solved using the proposed PGD methodology.



(a)



(b)

Figure 4.2: a) Graphical representation of the experimental set-up for the the wall-mounted hump. The model is mounted on a splitter plate with glass endplates on the sides. b) A jet acts at approximately 65% of the hump chord. A 2-D visualisation of that jet slot. [Source: https://cfdval2004.larc.nasa.gov/case3.html]

4.4.1 Parametric study of the NASA wall-mounted hump

Considering how RANS methods tend to overestimate the reattachment point, the main goal of the reduced order model is to properly estimate the behaviour of the full-order solution instead of the experimental data. Such assumption provides freedom to the case set-up. More specifically, for the sake of reducing computational cost a set of changes are proposed to the original test case. Starting with the two-dimensional case and in order to reduce the size of the computational domain, the version without the plenum is chosen. A patch is defined on the part of the hump that was originally the jet slot and a sinusoidal velocity profile normal to the patch face is imposed. Similarly to the flow control cases in section 3.3, a single parameter, μ, controls the jet peak velocity. More specifically when the jet patch is discretised using i faces

$$\boldsymbol{u}_i^{jet}(\boldsymbol{x},\mu) = \mu U_0 \frac{1 - cos(2\hat{x}_i\pi)}{2}\boldsymbol{n}_i \tag{4.21}$$

where U_0 is the peak velocity of the jet and is chosen such that the mass flow ratio between inlet and jet is similar to the one proposed in the original case, $\hat{x}$ is the normalised coordinate such that $\hat{x} \in [0\,,1]$ and $\boldsymbol{n}_i$ is the face normal vector.

The three-dimensional case is the product of extrusion of the two-dimensional domain in the third (z) direction by 0.8 chord lengths. The three-dimensional version of the jet boundary condition reads

$$\boldsymbol{u}_i^{jet}(\boldsymbol{x},\mu) = \mu U_0 \frac{(1 - cos(2\hat{x}_i\pi))(1 - cos(2\hat{z}_i\pi))}{4}\boldsymbol{n}_i \tag{4.22}$$

where

$$\hat{z} = \begin{cases} 0 & \text{for } z \leq 0.4C \\ \frac{z-\min(z)}{\max(z)-\min(z)} & \text{for } 0.4C < z < 0.6C \\ 0 & \text{for } z \geq 0.6C \end{cases} \tag{4.23}$$

In both cases the parametric domain is defined as $\mathcal{I} = [0.1,\, 1]$.

4.4.2 The parametric 2-D NASA wall-mounted hump

The computational domain of the two-dimensional wall-mounted hump is displayed in Fig.4.3 and consists of $114,000$ cells. It extends 6.39c upstream, 5c downstream and has a height of 1c. In order to avoid singularities at the inlet a parabolic ramp-up of the velocity is introduced, where starting from null velocity at $y = 0$m the maximum velocity is achieved at $y = 0.02$m. A similar ramp-up is used for the state viscosity at the inlet, where the freestream value is chosen to be $\tilde{\nu} = 3\nu$. The remaining boundary conditions feature homogeneous velocity and state viscosity on the bottom wall/hump, symmetry on the top wall, free traction on the outlet and zero Neumann state viscosity on the jet patch. Using a mesh with $y^+ < 1$ on the hump, no wall treatment is required for turbulent viscosity.

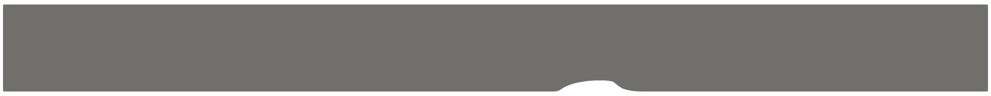


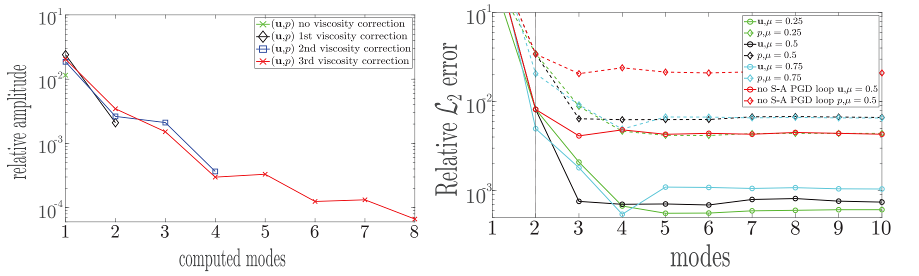
Figure 4.3: Visual representation of the 2-dimensional computational domain for the wall-mounted hump. The domain extends 6.39c upstream, 5c downstream and has a height of 1c.

The exact boundary conditions are imposed using two modes computed using `simpleFoam` and the Spalart-Allmaras turbulence model. The first mode corresponds to the full-order solution when the jet acts at 10% mass flow rate ($\mu = 0.1$) and the second to the case of maximum jet ($\mu = 1$). The parametric functions corresponding to the first and second spatial modes are $\phi(\mu) = 1 - \mu$ and $\phi(\mu) = \mu$, respectively.

The enrichment process tolerance is set at 10^{-4}, while the turbulence enrichment tolerance is set at 10^{-2}. The criterion to enter the viscosity update loop is described in detail in Section 4.3. The NS PGD loop reaches the desired tolerance after 8 computed modes and 3 SA PGD calls. Each SA PGD loop reached the prescribed tolerance within 3 computed modes. The relative amplitude of the modes, as the turbulent viscosity was being updated, is reported in Figure 4.4(a).

The approximation of the PGD methodology is compared to the full-order solution using `simpleFoam` for $\mu = 0.25$, $\mu = 0.5$ and $\mu = 0.75$, respectively. In Figure 4.4(b) the $\mathcal{L}_2$ error for these configurations is reported. For comparison the error when no SA PGD loops are called is included. It can be observed how without updating viscosity the error is stagnated from the first couple of modes and is higher by almost one order of magnitude.

The main point of interest is the effect of the suction jet in the reattachment point. Focusing on the recirculation bubble, a comparison of the skin friction coefficient between the PGD approximation and the full-order solution is reported in 4.5(a). For comparison both the PGD approximation using only the boundary condition modes ($n = 2$) and using all 8 computed



(a) (b)

Figure 4.4: a) Relative amplitude of the computed PGD modes. Each time the relative amplitude drops by one order of magnitude, the Spalart-Allmaras PGD loop is initiated to update the state viscosity modes. The previously computed velocity and pressure modes are recomputed for the updated turbulent eddy viscosity. b) Relative $\mathcal{L}_2$ error of the computed PGD approximation after the last state viscosity update. For comparison the relative $\mathcal{L}_2$ error of the PGD approximation for $\mu = 0.5$ when no Spalart-Allmaras PGD loop is included is included.

modes ($n = 10$) are included. The skin friction coefficient is evaluated as

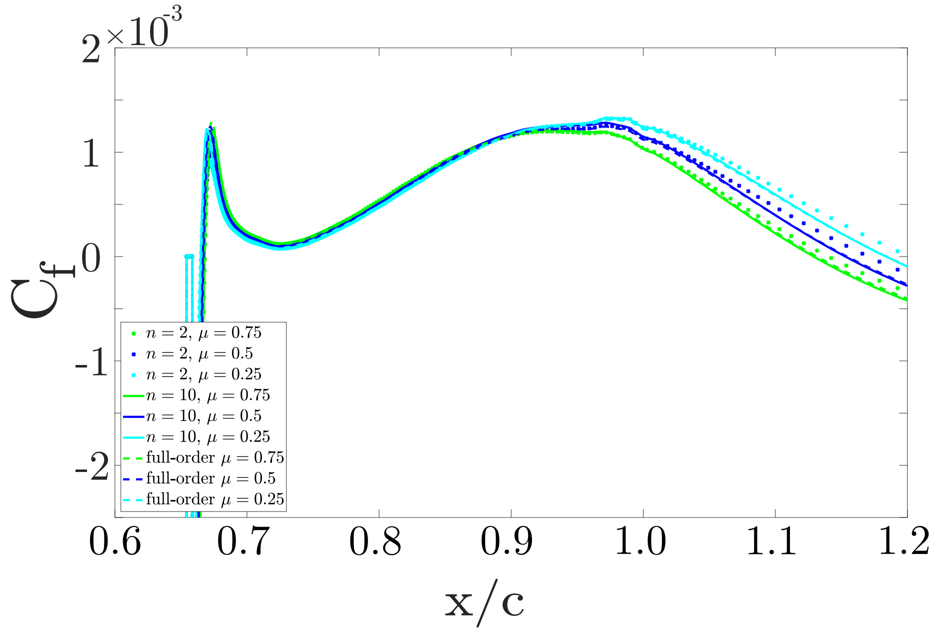
$$C_f := \frac{2\tau_w}{\rho U_\infty^2}$$

where τ_w is the wall shear stress and is defined as

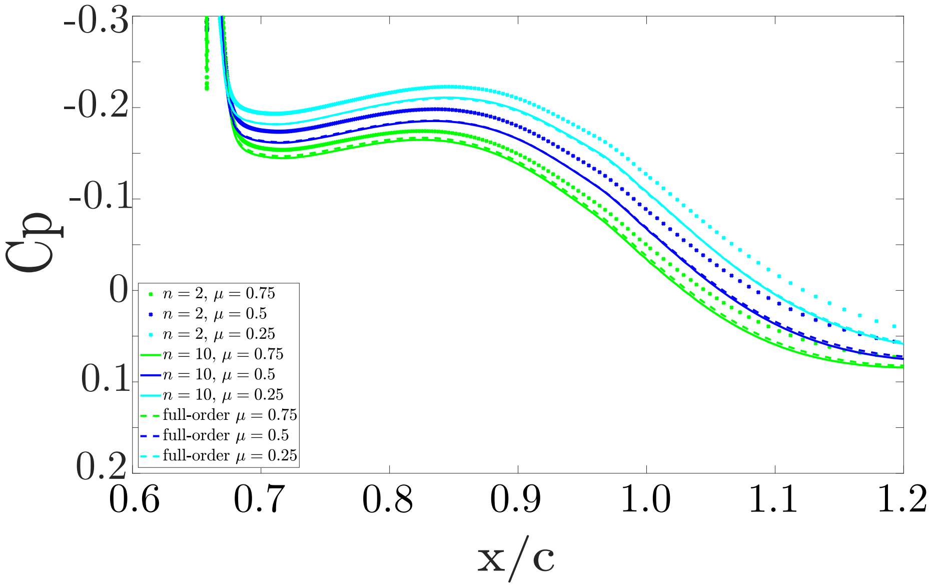
$$\tau_w := \mu \left(\frac{\partial \boldsymbol{u}}{\partial y} \right)_{y=0} .$$

Similarly, in Figure 4.5(b) a comparison of the coefficient of pressure of the approximation to the full-order is reported. It is important to note how closely the proposed PGD methodology follows the full-order solution and how well it picks up the reattachment position for all three parametric values (Table 4.1).

A qualitative comparison of the velocity, pressure and state viscosity fields approximated with PGD and computed with the full-order solver are pre-



(a)



(b)

Figure 4.5: a) Comparison of the skin friction coefficient between PGD approximation and full-order solution for the three prescribed states, $\mu = 0.25$, $\mu = 0.5$ and $\mu = 0.75$. b) Coefficient of pressure along the bottom wall and hump. The PGD approximation for $\mu = 0.25$, $\mu = 0.5$ and $\mu = 0.75$ is compared to the corresponding `simpleFoam` solution.

μ	Full-Order	PGD
0.25	1.184 c	1.183 c
0.50	1.154 c	1.156 c
0.75	1.131 c	1.129 c

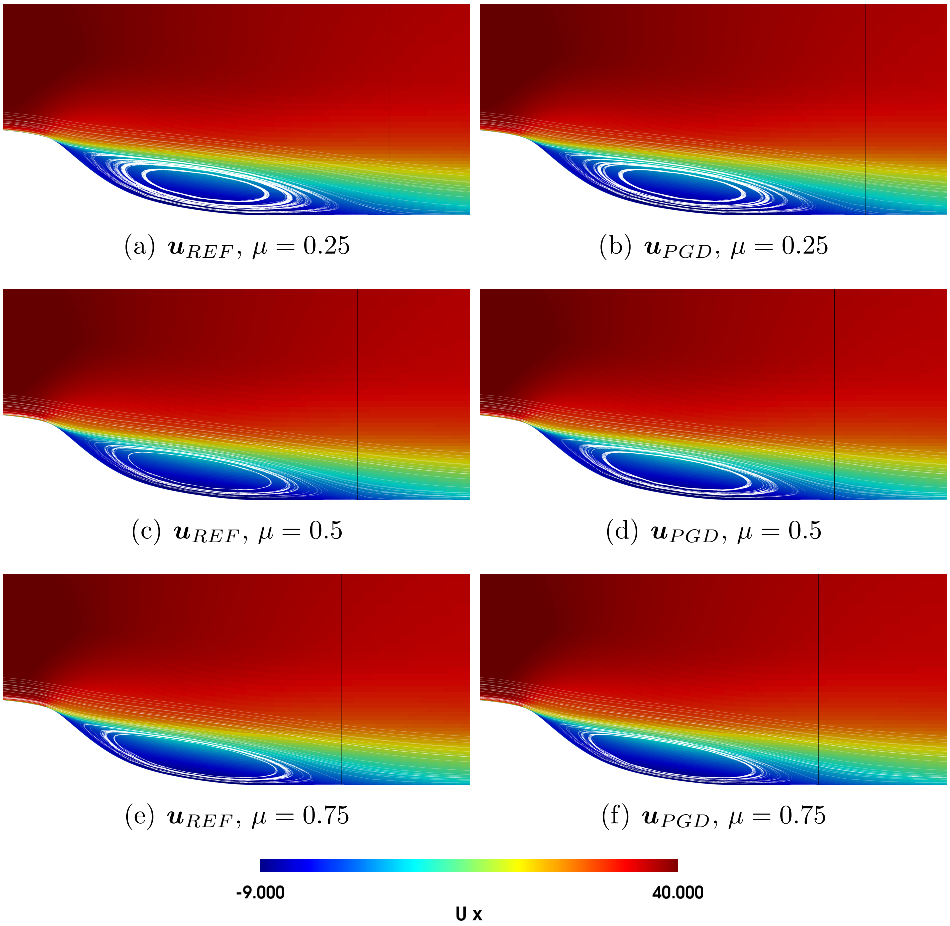
Table 4.1: Comparison between the reattachment point computed with `simpleFoam` and approximated with the proposed PGD methodology for three different parameter values.

sented in Figures 4.6, 4.7 and 4.8.

4.4.3 The parametric 3-D NASA wall-mounted hump

As described in Section 4.4.1 the three-dimensional domain is the product of extrusion of the two-dimensional version in the z direction with a displacement length of 0.8 c (Fig.4.9). In order to further reduce the computational cost of the three-dimensional case the Reynolds number was lowered, while remaining in the turbulent range and allowing for a much coarser mesh. More precisely the inlet and jet velocity is scaled down to 10% of the 2D case and the new Reynolds number is $Re = 93,600$, which is still within the turbulent range. The z direction is discretised with 72 cells, resulting in a domain of 2.34 million cells.

The jet boundary condition is described in detail in section 4.4.1. The rest of the boundary conditions are similar to the two-dimensional case, including the ramp-up of the velocity and state viscosity on the inlet. Slip is imposed to the two extra boundary patches.


(a) $\boldsymbol{u}_{REF}$, $\mu = 0.25$

(b) $\boldsymbol{u}_{PGD}$, $\mu = 0.25$

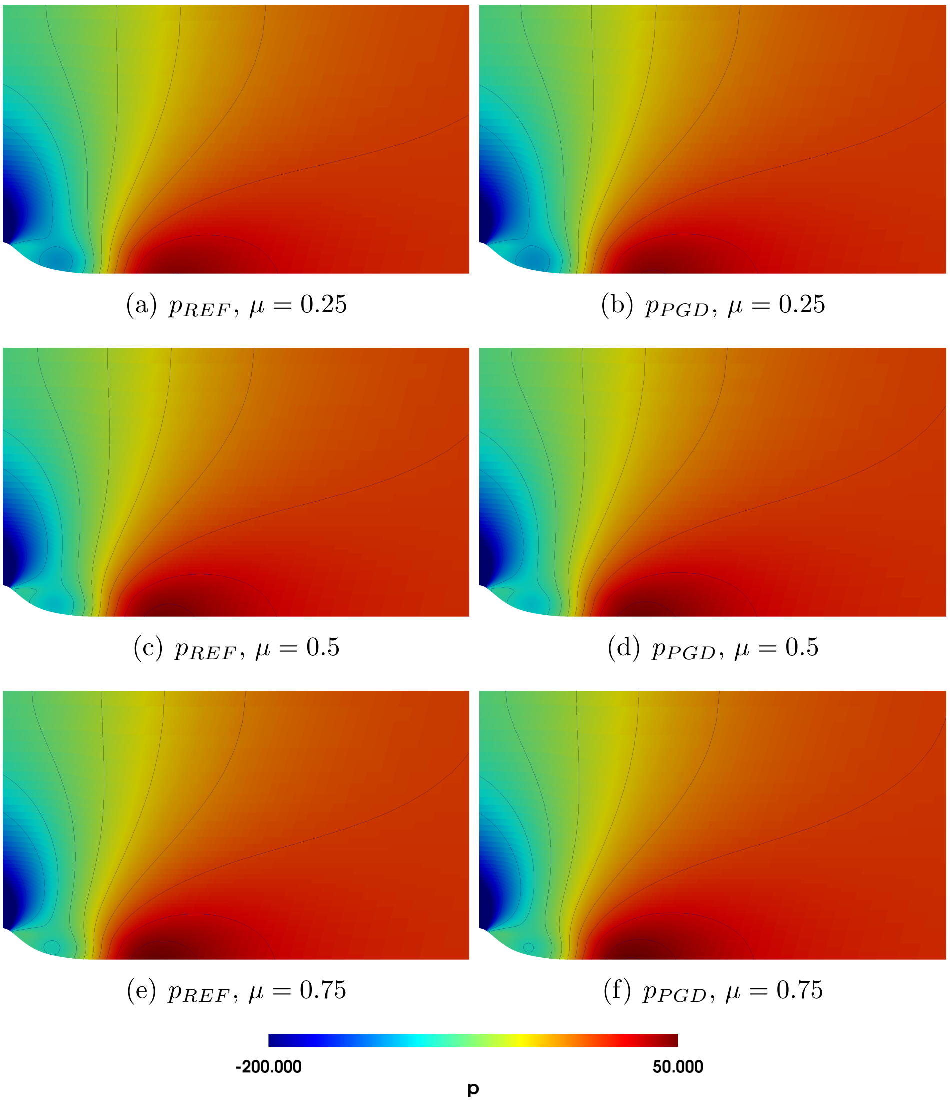
(c) $\boldsymbol{u}_{REF}$, $\mu = 0.5$

(d) $\boldsymbol{u}_{PGD}$, $\mu = 0.5$

(e) $\boldsymbol{u}_{REF}$, $\mu = 0.75$

(f) $\boldsymbol{u}_{PGD}$, $\mu = 0.75$

Figure 4.6: Visual comparison of the recirculation bubble between `simpleFoam` solution and PGD approximation for $\mu = 0.25$ (top), $\mu = 0.5$ (middle) and $\mu = 0.75$ (bottom). The reattachment point is noted with a vertical line.



(a) p_{REF}, $\mu = 0.25$

(b) p_{PGD}, $\mu = 0.25$

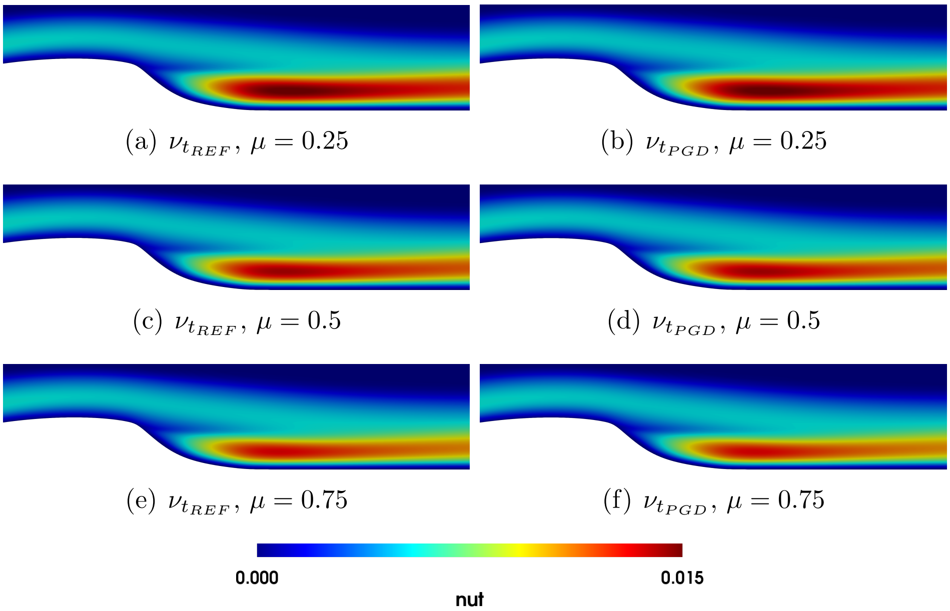
(c) p_{REF}, $\mu = 0.5$

(d) p_{PGD}, $\mu = 0.5$

(e) p_{REF}, $\mu = 0.75$

(f) p_{PGD}, $\mu = 0.75$

Figure 4.7: Visual comparison of the pressure field in the detachment area between `simpleFoam` solution and PGD approximation for $\mu = 0.25$ (top), $\mu = 0.5$ (middle) and $\mu = 0.75$ (bottom).



(a) $\nu_{t_{REF}}$, $\mu = 0.25$

(b) $\nu_{t_{PGD}}$, $\mu = 0.25$

(c) $\nu_{t_{REF}}$, $\mu = 0.5$

(d) $\nu_{t_{PGD}}$, $\mu = 0.5$

(e) $\nu_{t_{REF}}$, $\mu = 0.75$

(f) $\nu_{t_{PGD}}$, $\mu = 0.75$

Figure 4.8: Visual comparison of the turbulent eddy viscosity in the detached area between `simpleFoam` solution and PGD approximation for $\mu = 0.25$ (top), $\mu = 0.5$ (middle) and $\mu = 0.75$ (bottom).

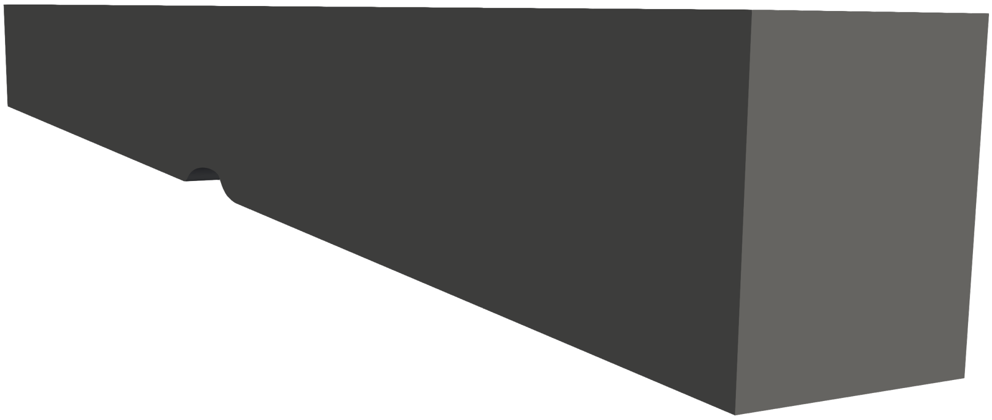


Figure 4.9: Visual representation of the 3-dimensional computational domain for the wall-mounted hump. The domain extends is the result of extrusion of the 2-dimensional domain by 0.8c.

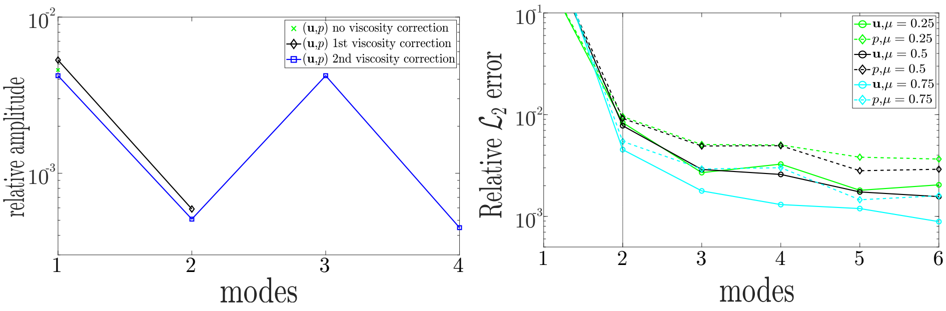
Similar to the 2-dimensional case, the boundary condition modes are the full-order solutions using `simpleFoam` and correspond to the states described by the extrema of the parametric space. More precisely the first boundary condition mode refers to the solution when the jet functions at 10% of its maximum value ($\mu = 0.1$) and the second to the solution with the maximum jet ($\mu = 1$). The parametric functions remain $\phi(\mu) = 1 - \mu$ and $\phi(\mu) = \mu$, respectively.

Further reducing the computational cost of the case, the enrichment process tolerance is increased by half an order of magnitude, set at $5 \cdot 10^{-4}$, while the turbulence enrichment tolerance is kept at 10^{-2}. What is more, the condition to call SA PGD, as recalled in 9, is modified to

$$\eta_{turb} = 10^{-(i_t+2)},$$

resulting in one less SA PGD call. The NS PGD loop reaches the desired tolerance after 4 computed modes and 2 SA PGD loops. Every SA PGD loop reached the prescribed tolerance with 2 computed modes. Much like the results presented in 4.4.2, first, the relative amplitude of the modes, as SA PGD is called to update the turbulent viscosity is reported in Figure 4.10(a). The relative $\mathcal{L}_2$ error of the PGD approximation to the `simpleFoam` solution is presented in Figure 4.10(b).

A qualitative comparison between PGD and full-order is displayed in figure 4.11, a top view plot of the wall shear stress on the bottom wall starting from the jet patch up to approximately 2.5c downstream, and figure 4.12 where the pressure is plotted on the same surface. The comparison between

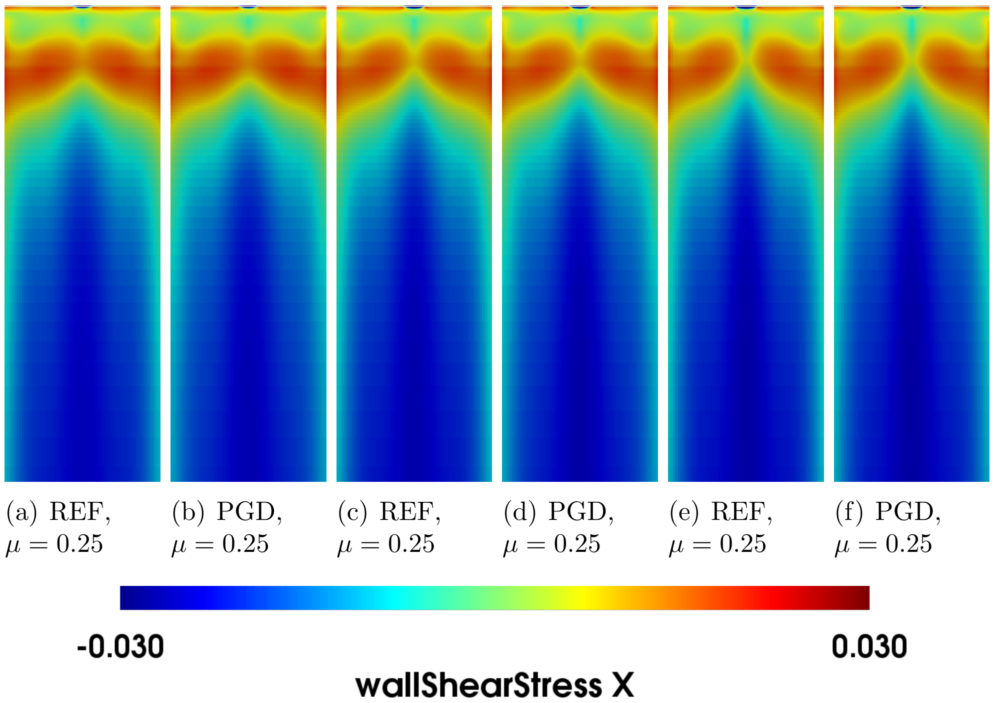


(a)

(b)

Figure 4.10: a) Relative amplitude of the computed PGD modes. Each time the relative amplitude drops by one order of magnitude, the Spalart-Allmaras PGD loop is initiated to update the state viscosity modes. The previously computed velocity and pressure modes are recomputed for the updated turbulent eddy viscosity. b) Relative $\mathcal{L}_2$ error of the computed PGD approximation after the last state viscosity update.

PGD and full-order solver continues in figure 4.13, where the development of the velocity profile for $\mu = 0.75$ is showcased via two slices of the mesh at $x =$ 0.71c and 1.42c, respectively. The velocity streamlines on the recirculation bubble are added to display the ability of the PGD methodology to pick up the effect of the jet on the vortices.



(a) REF, $\mu = 0.25$

(b) PGD, $\mu = 0.25$

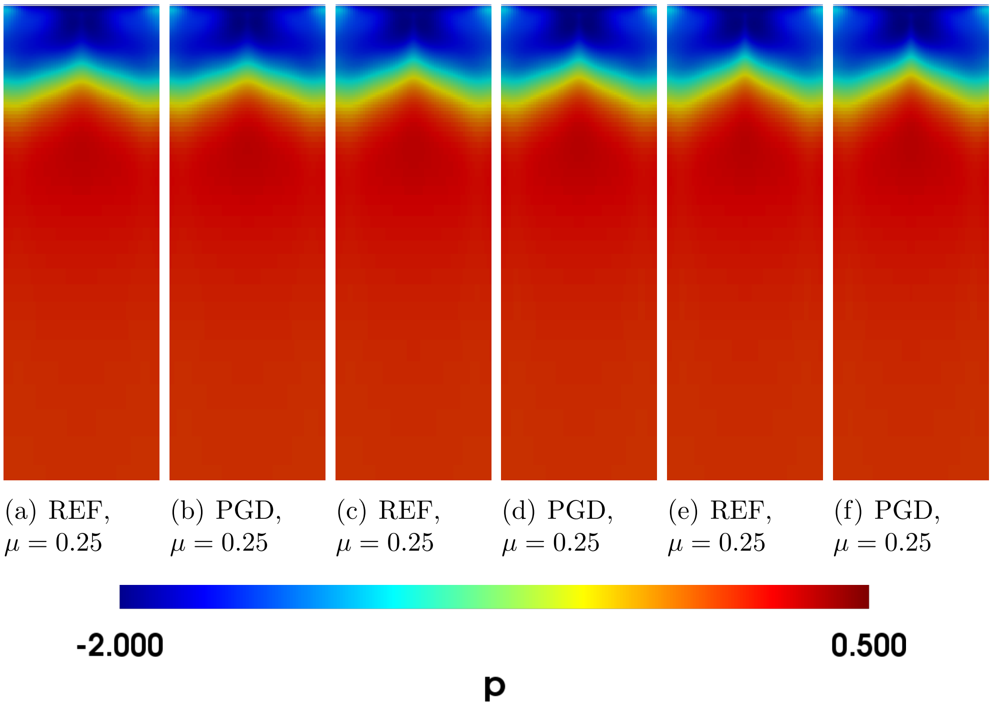
(c) REF, $\mu = 0.25$

(d) PGD, $\mu = 0.25$

(e) REF, $\mu = 0.25$

(f) PGD, $\mu = 0.25$

Figure 4.11: Top view plot of the wall shear stress on the bottom wall, starting from jet patch up to 2.5c downstream, for $\mu = 0.25$, $\mu = 0.5$ and $\mu = 0.75$.


(a) REF, $\mu = 0.25$

(b) PGD, $\mu = 0.25$

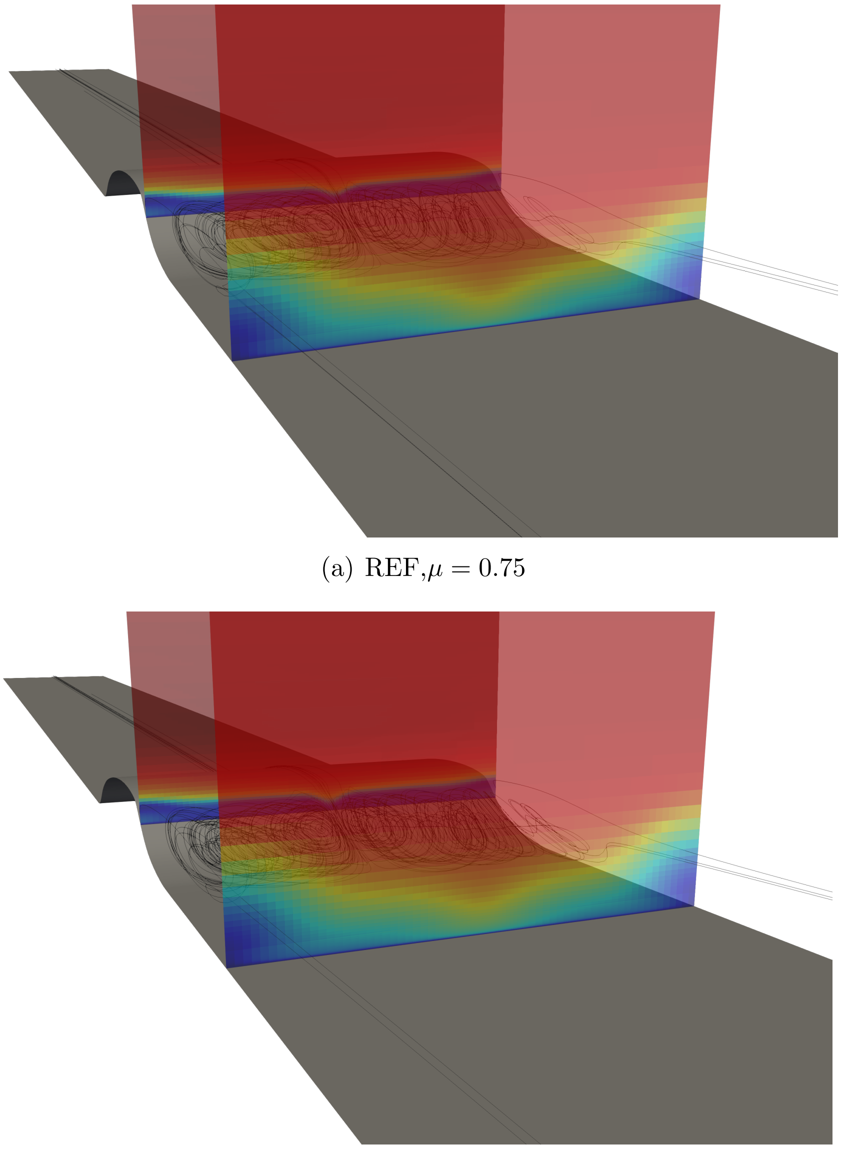
(c) REF, $\mu = 0.25$

(d) PGD, $\mu = 0.25$

(e) REF, $\mu = 0.25$

(f) PGD, $\mu = 0.25$

Figure 4.12: Top view plot of the pressure on the bottom wall, starting from jet patch up to 2.5c downstream, for $\mu = 0.25$, $\mu = 0.5$ and $\mu = 0.75$.



(a) REF,$\mu = 0.75$

(b) PGD,$\mu = 0.75$

Figure 4.13: Velocity streamlines on the recirculation bubble and velocity plot at x = 0.71c and x = 1.42c for μ = 0.75. Notice how the PGD approximation picks up the effect of the jet on the vortices and the development of the velocity profile.

Chapter 5

Concluding remarks

The feasibility of PGD methodologies for the solution of parametrised flow problems has been explored. Parametrised Stokes and Oseen flows were used to build an initial implementation of PGD in OpenFOAM and numerically validate its applicability to flow problems using a manufactured solution to verify the optimal order of convergence of the PGD-FV approximation. Then a nonintrusive PGD implementation in OpenFOAM has been proposed in the context of parametrised incompressible laminar Navier-Stokes flows. The main novelty of such approach is represented by the seamless exploitation of OpenFOAM native SIMPLE solver, making the resulting reduced-order strategy suitable for application in a daily industrial environment. The `pgdFoam` algorithm relies on the industrially-validated solver `simpleFoam` to compute the spatial modes of the solution, whereas the parametric ones are determined via the solution of a linear system of algebraic equations.

The developed strategy has been validated using a classical benchmark test case in the literature of CFD techniques for incompressible flows. Moreover, the potential of the proposed PGD approach to rapidly and accurately simulate incompressible flows for different sets of user-defined parameters has been tested in the context of flow control problems. The `pgdFoam` algorithm has been applied both to an academic test case and an industrial one with a 3D geometrical model provided by Volkswagen AG.

The proposed PGD methodology has proved to be able to compute an accurate reduced basis for the problems under analysis with no *a priori* knowledge of the expected solutions. Moreover, it has shown robustness when dealing with a large range of values of the parameters, accuracy in capturing significant topological changes in the flow features and reliability in evaluating quantities of engineering interest, with an extremely reduced computing time.

Finally, a PGD solver for the Spalart-Allmaras turbulence model was proposed for the approximation of parameter-dependent turbulence fields. A segregated approach to the nonintrusive PGD strategy was proposed, implementing the parametrised S-A solver in the enrichment process. This method allows for the approximation of parametrised turbulent Navier-Stokes flows with no *a priori* knowledge of either the flow or the turbulence fields.

Using a well-documented turbulence model validation test case, the proposed PGD methodology has proved to be able to compute an accurate approximation of high Reynolds number cases. The methodology could closely follow the `simpleFoam` solver coupled with the OpenFOAM-implemented Spalart-Allmaras turbulence model. The effect of flow control on both turbulent viscosity and the flow structure could be reproduced. Quantities of industrial interest, such as skin friction and the coefficient of pressure, were approximated with high accuracy throughout the parametric domain.

Chapter 6

Future works

After developing a nonintrusive proper generalised decomposition methodology for OpenFOAM that is able to tackle parametrised turbulent Navier-Stokes flow problems there are plenty of continuations to be made both from a scientific and practical (i.e. to be used by the industry) point of view. Such improvements shall contribute to the ease of use of the methodology, therefore the adoption rate and deployment in the daily industrial environment.

1. Efficiency:

   A characteristic of PGD, and any ROM for that matter, is the fact that complex problems require a significant number of computed modes to reach the desired approximation accuracy. The more modes in the approximation the more expensive each step becomes. The nonintrusive PGD proposed in this work limits this extra cost to the assembly stages of the spatial and parametric steps (Algorithm 1). The algebraic PGD [35] uses more memory to keep track of the separable operators in a matrix form. Despite the fact that PGD is an efficient tool for multiparametric problems, introducing post- or pre-processing steps in the enrichment process to impose orthogonality between computed modes could reduce the cost and speed-up the convergence process.

2. Turbulence:

   It is multiple times expressed throughout Chapter 4 that certain non-separable terms in the turbulence model are used in their higher dimensional form. This is an obvious limitation of the methodology as is, increasing cost and computational time. Building an accurate separable approximation of these terms would be a significant improvement for two reasons. On one hand these terms depend on the state viscosity approximation and removing the current under-relaxation would speed up the convergence. On the other hand, the computation of high dimensional integrals, especially when considering multiple parameters, introduces a bottleneck that would be lifted.

3. Geometry:

   This methodology was built with flow control problems in mind. However, it offers a great baseline that could be adapted to handle other problems. The first and main entry to a list of such new applications would be for parameter controlled geometry variations. Geometry optimisation via reduced order models based on *a posteriori* separation share a common issue which is the requirement to generate a library of different meshes and compute the corresponding flow. Introducing a separable mapping of a reference geometry [99] could prove an efficient alternative, especially when considering multiple parameters, while retaining the nonintrusiveness of the methodology. Aerodynamic shape optimisation with deterministic methods using nonintrusive PGD could be a quick and streamlined process of the industrial design pipeline.

4. Spatial adaptivity:

   The actual PGD implementation computes global spatial modes. This is reasonable for the first few modes. However, as the number of modes increases, in flow problems, local features are more important. Consequently, spatially adapted PGD strategy seems a reasonable alternative to better capture local features and, at the same time, reduce the computational cost.

Bibliography

[1] M. Ainsworth and R. Rankin. Guaranteed computable bounds on quantities of interest in finite element computations. *Int. J. Numer. Methods Eng.*, 89(13):1605–1634, 2012.

[2] M. C. Aletti, S. Perotto, and A. Veneziani. Himod reduction of advection–diffusion–reaction problems with general boundary conditions. *J. Sci. Comput.*, 76(1):89–119, Jul 2018.

[3] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *J. Non-Newton. Fluid*, 139(3):153 – 176, 2006.

[4] Amine Ammar, Antonio Huerta, Francisco Chinesta, Elías Cueto, and Adrien Leygue. Parametric solutions involving geometry: a step towards efficient shape optimization. *Comput. Methods Appl. Mech. Eng.*, 268:178–193, 2014.

[5] F. Ballarin, A. Manzoni, A. Quarteroni, and G. Rozza. Supremizer stabilization of POD-Galerkin approximation of parametrized steady

incompressible Navier-Stokes equations. *Int. J. Numer. Methods Eng.*, 102(5):1136–1161, 2015.

[6] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Ser. I-Math.*, 339(9):667 – 672, 2004.

[7] T. Barth, R. Herbin, and M. Ohlberger. Finite Volume Methods: Foundation and Analysis. In *Encyclopedia of Computational Mechanics Second Edition*, pages 1–60. American Cancer Society, 2017.

[8] M. Bergmann, C.-H. Bruneau, and A. Iollo. Enablers for robust POD models. *J. Comput. Phys.*, 228(2):516 – 538, 2009.

[9] A. Bertram, C. Othmer, and R. Zimmermann. Towards real-time vehicle aerodynamic design via multi-fidelity data-driven reduced order modeling. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2018.

[10] A. Caiazzo, T. Iliescu, V. John, and S. Schyschlowa. A numerical investigation of velocity-pressure reduced order models for incompressible flows. *J. Comput. Phys.*, 259:598 – 616, 2014.

[11] K. Carlberg. *Model reduction of nonlinear mechanical systems via optimal projection and tensor approximation.* PhD thesis, Stanford University, Dept. of Aeronautics and Astronautics, 2011.

[12] K. Carlberg, M. Barone, and H. Antil. Galerkin v. least-squares Petrov-Galerkin projection in nonlinear model reduction. *Journal of Computational Physics*, 330:693 – 734, 2017.

[13] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.

[14] K. Carlberg, Y. Choi, and S. Sargsyan. Conservative model reduction for finite-volume models. *J. Comput. Phys.*, 371:280 – 314, 2018.

[15] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242:623 – 647, 2013.

[16] S. Chaturantabut and D. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.*, 32(5):2737–2764, 2010.

[17] S. Chen and G. D. Doolen. Lattice Boltzmann method for fluid flows. *Annu. Rev. Fluid Mech.*, 30(1):329–364, 1998.

[18] I. Cheylan, G. Fritz, D. Ricot, and P. Sagaut. Shape optimization using the adjoint lattice boltzmann method for aerodynamic applications. *AIAA Journal*, pages 1–16, 04 2019.

[19] F. Chinesta, A. Huerta, G. Rozza, and K. Willcox. Model Reduction Methods. In Erwin Stein, Rene de Borst, and Thomas J. R. Hughes, editors, *Encyclopedia of Computational Mechanics Second Edition*, volume Part 1 Solids and Structures, chapter 3, pages 1–36. John Wiley & Sons, Ltd., Chichester, 2017.

[20] F. Chinesta, P. Ladevèze, and E. Cueto. A Short Review on Model Order Reduction Based on Proper Generalized Decomposition. *Arch. Comput. Methods Eng.*, 18(4):395, Oct 2011.

[21] F. Chinesta, A. Leygue, F. Bordeu, J. V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar, and A. Huerta. PGD-Based Computational Vademecum for Efficient Design, Optimization and Control. *Arch. Comput. Methods Eng.*, 20(1):31–59, Mar 2013.

[22] Francisco Chinesta, Elias Cueto, and Antonio Huerta. PGD for solving multidimensional and parametric models. In F. Chinesta and P. Ladevèze, editors, *Separated representations and PGD-based model reduction*, volume 554 of *CISM Courses and Lectures*, pages 27–89. Springer, Vienna, 2014.

[23] Francisco Chinesta, Roland Keunings, and Adrien Leygue. *The proper generalized decomposition for advanced numerical simulations. A primer.* Springer Briefs in Applied Sciences and Technology. Springer, Cham, 2014.

[24] R. Cottereau, P. Díez, and A. Huerta. Strict error bounds for linear solid mechanics problems using a subdomain-based flux-free method. *Comput. Mech.*, 44(4):533–547, 2009.

[25] Amaury Courard, David Néron, Pierre Ladevèze, and Ludovic Ballere. Integration of PGD-virtual charts into an engineering design process. *Comput. Mech.*, 57(4):637–651, 2016.

[26] L. Cueto-Felgueroso, I. Colominas, X. Nogueira, F. Navarrina, and M. Casteleiro. Finite volume solvers and Moving Least-Squares approximations for the compressible Navier–Stokes equations on unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 196(45):4712 – 4736, 2007.

[27] A. E. Deane, I. G. Kevrekidis, G. E. Karniadakis, and S. A. Orszag. Low-dimensional models for complex geometry flows: Application to grooved channels and circular cylinders. *Phys. Fluids A: Fluid Dynamics*, 3(10):2337–2354, 1991.

[28] L. Dedè. Optimal flow control for Navier-Stokes equations: drag minimization. *Int. J. Numer. Methods Fluids*, 55(4):347–366, 2007.

[29] P. Destuynder and B. Métivet. Explicit error bounds in a conforming finite element method. *Math. Comp.*, 68(228):1379–1396, 1999.

[30] Pedro Díez, Sergio Zlotnik, and Antonio Huerta. Generalized parametric solutions in Stokes flow. *Comput. Methods Appl. Mech. Eng.*, 326:223–240, 2017.

[31] Jean Donea and Antonio Huerta. *Finite element methods for flow problems.* John Wiley & Sons, Chichester, 2003.

[32] M. Drohmann, B. Haasdonk, and M. Ohlberger. Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM J. Sci. Comput.*, 34(2):A937–A969, 2012.

[33] A. Dumon, C. Allery, and A. Ammar. Proper general decomposition (PGD) for the resolution of Navier-Stokes equations. *J. Comput. Phys.*, 230(4):1387 – 1407, 2011.

[34] R. Duvigneau and M. Visonneau. Optimization of a synthetic jet actuator for aerodynamic stall control. *Comput. Fluids*, 35(6):624 – 638, 2006.

[35] Pedro Díez, Sergio Zlotnik, Alberto García-González, and Antonio Huerta. Algebraic pgd for tensor separation and compression: An algorithmic approach. *Comptes Rendus Mécanique*, 346(7):501–514, July 2018.

[36] A. Ern, A. F. Stephansen, and M. Vohralík. Guaranteed and robust discontinuous galerkin a posteriori error estimates for convection-diffusion-reaction problems. *J. Comput. Appl. Math.*, 234(1):114 – 130, 2010.

[37] A. Ern and M. Vohralík. Polynomial-Degree-Robust A Posteriori Estimates in a Unified Setting for Conforming, Nonconforming, Discon-

tinuous Galerkin, and Mixed Discretizations. *SIAM J. Numer. Anal.*, 53(2):1058–1081, 2015.

[38] R. Eymard, T. Gallouët, and R. Herbin. Finite volume methods. *Handbook of Numerical Analysis*, 7:713 – 1018, 2000. Solution of Equation in $\mathbb{R}^n$ (Part 3), Techniques of Scientific Computing (Part 3).

[39] U. Ghia, K. N. Ghia, and C. T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comput. Phys.*, 48:387–411, 1982.

[40] M. Giacomini. An Equilibrated Fluxes Approach to the Certified Descent Algorithm for Shape Optimization Using Conforming Finite Element and Discontinuous Galerkin Discretizations. *J. Sci. Comput.*, 75(1):560–595, Apr 2018.

[41] Giacomini, M., Pantz, O., and Trabelsi, K. Certified descent algorithm for shape optimization driven by fully-computable a posteriori error estimators. *ESAIM: COCV*, 23(3):977–1001, 2017.

[42] A. J. Gil, J. Bonet, J. Silla, and O. Hassan. A discrete geometric conservation law (DGCL) for a cell vertex finite-volume algorithm on moving domains. *International Journal for Numerical Methods in Biomedical Engineering*, 26(6):770–779, 2010.

[43] A. J. Gil, Z. Zhang, O. Hassan, and K. Morgan. Parallel multigrid detached eddy simulation algorithm for three-dimensional unsteady incompressible flows on unstructured meshes. *Journal of Aerospace Engineering*, 19(4):271–280, October 2019.

[44] D. González, E. Cueto, F. Chinesta, P. Díez, and A. Huerta. Streamline upwind/Petrov-Galerkin-based stabilization of proper generalized decompositions for high-dimensional advection-diffusion equations. *Int. J. Numer. Methods Eng.*, 94(13):1216–1232, 2013.

[45] D. González, L. Debeugny, E. Cueto, F. Chinesta, P. Díez, and A. Huerta. SUPG-based stabilization using a separated representations approach. *Int. J. Mater. Form.*, 3(1):883–886, 2010.

[46] D. Greenblatt, K. B. Paschal, C.-S. Yao, J. Harris, N. W. Schaeffler, and A. E. Washburn. Experimental investigation of separation control, part 1 Baseline and steady suction. *AIAA Journal*, 44(12):2820–2830, 2006.

[47] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: M2AN*, 41(3):575–605, 2007.

[48] M. A. Grepl and A. T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: M2AN*, 39(1):157–181, 2005.

[49] J.-L. Guermond and L. Quartapelle. On stability and convergence of projection methods based on pressure Poisson equation. *Int. J. Numer. Methods Fluids*, 26(9):1039–1053, 1998.

[50] J.-L. Guermond and L. Quartapelle. On the approximation of the unsteady Navier-Stokes equations by finite element projection methods. *Numer. Math.*, 80(2):207–238, 1998.

[51] E. Guilmineau, R. Duvigneau, and J. Labroquère. Optimization of jet parameters to control the flow on a ramp. *C. R. Acad. Sci. Ser. II-Mec.*, 342(6):363 – 375, 2014. Flow separation control.

[52] S. Guzzetti, S. Perotto, and A. Veneziani. Hierarchical model reduction for incompressible fluids in pipes. *Int. J. Numer. Methods Eng.*, 114(5):469–500, 2018.

[53] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: M2AN*, 42(2):277–302, 2008.

[54] S. Hijazi, G. Stabile, A. Mola, and G. Rozza. Data-Driven POD-Galerkin reduced order model for turbulent flows, 2019.

[55] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry.* Cambridge Monographs on Mechanics. Cambridge University Press, 1996.

[56] A. Huerta, E. Nadal, and F. Chinesta. Proper generalized decomposition solutions within a domain decomposition strategy. *Int. J. Numer. Methods Eng.*, 113(13):1972–1994, 2018.

[57] L. Iapichino, A. Quarteroni, and G. Rozza. A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks. *Comput. Methods Appl. Mech. Eng.*, 221-222:63 – 82, 2012.

[58] A. Iollo, S. Lanteri, and J.-A. Désidéri. Stability Properties of POD-Galerkin Approximations for the Compressible Navier-Stokes Equations. *Theor. Comput. Fluid Dyn.*, 13(6):377–396, Mar 2000.

[59] M. Islam, F. Decker, E. de Villiers, A. Jackson, J. Gines, T. Grahs, A. Gitt-Gehrke, and J. Comas i Font. Application of Detached-Eddy Simulation for automotive aerodynamics development. 2009.

[60] G. Karpouzas, E. Papoutsis-Kiachagias, T. Schumacher, E. De Villiers, K.C. Giannakoglou, and C. Othmer. Adjoint optimization for vehicle external aerodynamics. 7:1–7, 01 2016.

[61] L. I. G. Kovasznay. Laminar flow behind a two-dimensional grid. *Proc. Cambridge Phi. Sc.*, 44:58–62, 1947.

[62] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.*, 40(2):492–515, 2002.

[63] F. Larsson, P. Díez, and A. Huerta. A flux-free a posteriori error estimator for the incompressible Stokes problem using a mixed FE formulation. *Comput. Methods Appl. Mech. Eng.*, 199(37-40):2383–2402, 2010.

[64] H. Laval and L. Quartapelle. A fractional-step Taylor–Galerkin method for unsteady incompressible flows. *Int. J. Numer. Methods Fluids*, 11(5):501–513, 1990.

[65] P. Legresley and J. Alonso. Airfoil design optimization using reduced order models based on proper orthogonal decomposition. *Fluids 2000 conference and exhibit*, 08 2000.

[66] P. LeGresley and J. Alonso. Investigation of non-linear projection for POD based reduced order models for aerodynamics. *AIAA*, 2001.

[67] R. J. LeVeque. *Finite volume methods for hyperbolic problems.* Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 2002.

[68] S. Lorenzi, A. Cammi, L. Luzzi, and G. Rozza. POD-Galerkin method for finite volume approximation of Navier-Stokes and RANS equations. *Comput. Methods Appl. Mech. Eng.*, 311:151 – 179, 2016.

[69] X. Ma and G.E. Karniadakis. A low-dimensional model for simulating three-dimensional cylinder flow. *J. Fluid Mech.*, 458:181–190, 2002.

[70] I. Martini, G. Rozza, and B. Haasdonk. Reduced basis approximation and a-posteriori error estimation for the coupled Stokes-Darcy system. *Adv. Comput. Math.*, 41(5):1131–1157, Oct 2015.

[71] F. R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8):1598–1605, August 1994.

[72] D. Modesto, S. Zlotnik, and A. Huerta. Proper generalized decomposition for parameterized Helmholtz problems in heterogeneous and unbounded domains: Application to harbor agitation. *Comput. Methods Appl. Mech. Eng.*, 295:127 – 149, 2015.

[73] K. W. Morton and T. Sonar. Finite volume methods for hyperbolic conservation laws. *Acta Numer.*, 16:155–238, 2007.

[74] I. Mozolevski and S. Prudhomme. Goal-oriented error estimation based on equilibrated-flux reconstruction for finite element approximations of elliptic problems. *Comput. Methods Appl. Mech. Eng.*, 288:127 – 145, 2015.

[75] X. Nogueira, I. Colominas, L. Cueto-Felgueroso, S. Khelladi, F. Navarrina, and M. Casteleiro. Resolution of computational aeroacoustics problems on unstructured grids with a higher-order finite volume scheme. *Journal of Computational and Applied Mathematics*, 234(7):2089 – 2097, 2010.

[76] X. Nogueira, S. Khelladi, I. Colominas, L. Cueto-Felgueroso, J. Parıs, and H. Gømez. High-resolution finite volume methods on unstructured grids for turbulence and aeroacoustics. *Archives of Computational Methods in Engineering*, 18(3):315, Jul 2011.

[77] A. Nouy. Generalized spectral decomposition method for solving stochastic finite element equations: Invariant subspace problem and dedicated algorithms. *Comput. Methods Appl. Mech. Eng.*, 197(51):4718 – 4736, 2008.

[78] J.T. Oden and S. Prudhomme. Goal-oriented error estimation and adaptivity for the finite element method. *Comput. Math. Appl.*, 41(5):735 – 756, 2001.

[79] C. Othmer. Adjoint methods for car aerodynamics. *Journal of Mathematics in Industry*, 4(1):6, June 2014.

[80] Carsten Othmer, Trent W. Lukaczyk, Paul Constantine, and Juan J. Alonso. On active subspaces in car aerodynamics.

[81] E. Papoutsis-Kiachagias and K. Giannakoglou. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications. *Archives of Computational Methods in Engineering*, 12 2014.

[82] E. Papoutsis-Kiachagias, S. Porziani, C. Groth, M. Biancolini, E. Costa, and K. Giannakoglou. *Aerodynamic Optimization of Car Shapes Using the Continuous Adjoint Method and an RBF Morpher*, pages 173–187. 07 2018.

[83] N. Parés, J. Bonet, A. Huerta, and J. Peraire. The computation of bounds for linear-functional outputs of weak solutions to the two-dimensional elasticity equations. *Comput. Methods Appl. Mech. Eng.*, 195(4-6):406–429, 2006.

[84] N. Parés and P. Díez. A new equilibrated residual method improving accuracy and efficiency of flux-free error estimates. *Comput. Methods Appl. Mech. Eng.*, 313(1):785 – 816, 2017.

[85] N. Parés, P. Díez, and A. Huerta. Subdomain-based flux-free a posteriori error estimators. *Comput. Methods Appl. Mech. Eng.*, 195(4-6):297–323, 2006.

[86] N. Parés, P. Díez, and A. Huerta. Exact bounds for linear outputs of the advection-diffusion-reaction equation using flux-free error estimates. *SIAM J. Sci. Comput.*, 31(4):3064–3089, 2009.

[87] S.V Patankar and D.B Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transfer*, 15(10):1787 – 1806, 1972.

[88] Anthony T. Patera and Gianluigi Rozza. Reduced Basis Approximation and A-Posteriori Error Estimation for Parametrized Partial Differential Equations. MIT Pappalardo Graduate Monographs in Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA, 2007.

[89] S. Perotto, A. Ern, and A. Veneziani. Hierarchical Local Model Reduction for Elliptic Problems: A Domain Decomposition Approach. *Multiscale Model. Simul.*, 8(4):1102–1127, 2010.

[90] S. Perotto and A. Veneziani. Coupled model and grid adaptivity in hierarchical reduction of elliptic problems. *J. Sci. Comput.*, 60(3):505–536, Sep 2014.

[91] A. Quarteroni. *Numerical models for differential problems*, volume 16 of *MS&A. Modeling, Simulation and Applications*. Springer, Cham, 2017.

[92] A. Quarteroni, F. Saleri, and A. Veneziani. Factorization methods for the numerical approximation of Navier-Stokes equations. *Comput. Methods Appl. Mech. Eng.*, 188(1–3):505–526, 2000.

[93] L. Remaki, O. Hassan, and K. Morgan. Aerodynamic computations using a finite volume method with an HLLC numerical flux function. *Mathematical Modelling of Natural Phenomena*, 6(3):189 – 212, May 2011.

[94] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Methods Eng.*, 15(3):229, May 2008.

[95] Gianluigi Rozza. Fundamentals of reduced basis method for problems governed by parametrized PDEs and applications. In F. Chinesta and P. Ladevèze, editors, *Separated representations and PGD-based model reduction*, volume 554 of *CISM Courses and Lectures*, pages 153–227. Springer, Vienna, 2014.

[96] C. Rumsey. Reynolds-averaged navier-stokes analysis of zero efflux flow control over a hump model. *AIAA*, 2006.

[97] T. Schütz. *Hucho - Aerodynamik des Automobils.* Springer Vieweg, 2013.

[98] A. Seifert and L. G. pack. Active flow separation control on wall-mounted hump at high Reynolds numbers. *AIAA*, 40(7), 2002.

[99] R. Sevilla, S. Zlotnik, and A. Huerta. Solution of geometrically parametrised problems within a CAD environment via model order reduction. *Comput. Methods Appl. Mech. Eng.*, 358:112631, 2020.

[100] Ruben Sevilla, Matteo Giacomini, and Antonio Huerta. A face-centred finite volume method for second-order elliptic problems. *Int. J. Numer. Methods Eng.*, 115(8):986–1014, 2018.

[101] M. Signorini, S. Zlotnik, and P. Díez. Proper generalized decomposition solution of the parameterized Helmholtz problem: application to inverse geophysical problems. *Int. J. Numer. Methods Eng.*, 109(8):1085–1102, 2017.

[102] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. *AIAA*, January 1992.

[103] P. Spalart, W.H. Jou, M. Strelets, and S. Allmaras. Comments on the feasibility of les for wings, and on a hybrid rans/les approach. 01 1997.

[104] G. Stabile and G. Rozza. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier-Stokes equations. *Comput. Fluids*, 173:273 – 284, 2018.

[105] L. Tamellini, O. Le Maître, and A. Nouy. Model reduction based on proper generalized decomposition for the stochastic steady incompressible Navier-Stokes equations. *SIAM J. Sci. Comput.*, 36(3):A1089–A1117, 2014.

[106] Roger Temam. *Navier-Stokes equations. Theory and numerical analysis.* AMS Chelsea Publishing, Providence, RI, 2001. Corrected reprint of the 1984 edition [North-Holland, Amsterdam, 1984].

[107] The OpenFOAM foundation. OpenFOAM 6.0, 2019. [Accessed 5-February-2019].

[108] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer-Verlag, Berlin, third edition, 2009. A practical introduction.

[109] A. Travin, M. Shur, P. Spalart, M. Kh, and M. Strelets. Improvement of delayed detached-eddy simulation for les with wall modelling. 01 2006.

[110] N. P. Weatherill, O. Hassan, K. Morgan, J. W. Jones, and B. Larwood. Towards fully parallel aerospace simulations on unstructured meshes. *Engineering Computations*, 18(3/4):347–376, October 2019.

[111] D. Xiao, C.E. Heaney, L. Mottet, F. Fang, W. Lin, I.M. Navon, Y. Guo, O.K. Matar, A.G. Robins, and C.C. Pain. A reduced order model for turbulent flows in the urban environment using machine learning. *Building and Environment*, 148:323 – 337, 2019.

[112] C. Zhang, P. C. Bounds, L. Foster, and M. Uddin. Turbulence modeling effects on the CFD predictions of flow over a detailed full-scale sedan vehicle. *Fluids*, 4(3), 2019.

[113] R. Zimmermann, A. Vendl, and S. Görtz. Reduced-order modeling of steady flows subject to aerodynamic constraints. *AIAA Journal*, 52(2):255–266, 2014.

[114] Sergio Zlotnik, Pedro Díez, David Modesto, and Antonio Huerta. Proper Generalized Decomposition of a geometrically parametrized

heat problem with geophysical applications. *Int. J. Numer. Methods Eng.*, 103(10):737–758, 2015.

[115] X. Zou, M. Conti, P. Díez, and F. Auricchio. A nonintrusive proper generalized decomposition scheme with application in biomechanics. *Int. J. Numer. Methods Eng.*, 113(2):230–251, 2018.

Appendix A

Separated representation of the residuals

Consider a separable expression of the source term $\boldsymbol{s}(\boldsymbol{x},\boldsymbol{\mu}){:=}\eta(\boldsymbol{\mu})\boldsymbol{S}(\boldsymbol{x})$.

A.1 Parametrised Stokes flow

Considering the parametrised Stokes flow and for the spatial iteration, the residuals in separated form read as

$$\begin{aligned} R_u^n := &\int_{V_i} \alpha_3 \boldsymbol{S} \, dV + \int_{V_i} \boldsymbol{\nabla}\cdot\Big(D\boldsymbol{\nabla}\Big(\sum_{m=1}^{n} \alpha_4^m \sigma_u^m \boldsymbol{f}_u^m\Big)\Big) \, dV \\ &- \int_{V_i} \boldsymbol{\nabla}\Big(\sum_{m=1}^{n} \alpha_5^m \sigma_p^m f_p^m\Big) \, dV, \end{aligned} \tag{A.1a}$$

$$R_p^n := -\int_{V_i} \boldsymbol{\nabla}\cdot\Big(\sum_{m=1}^{n} \alpha_5^m \sigma_u^m \boldsymbol{f}_u^m\Big) \, dV. \tag{A.1b}$$

where the following expressions for the coefficients are devised

$$\begin{cases} \alpha_3 := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \phi^n \eta \, d\boldsymbol{\mathcal{I}}, \qquad \alpha_4^m := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \phi^n \phi^m \psi \, d\boldsymbol{\mathcal{I}}, \\ \alpha_5 := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \phi^n \phi^m \, d\boldsymbol{\mathcal{I}}. \end{cases} \tag{A.2}$$

For the parametric iteration, the separated expression of the residuals is

$$r_u^n := a_3 \eta + \sum_{m=1}^{n} \left(a_4^m \psi - a_5^m \right) \phi^m, \tag{A.3a}$$

$$r_p := - \sum_{m=1}^{n} a_6^m \phi^m, \tag{A.3b}$$

where the coefficients depend solely on the spatial modes, namely

$$\begin{cases} a_3 := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \boldsymbol{S} \, dV, \\ a_4^m := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \left[\boldsymbol{\nabla} \cdot (D \boldsymbol{\nabla} (\sigma_u^m \boldsymbol{f}_u^m)) \right] dV, \\ a_5^m := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \boldsymbol{\nabla} (\sigma_p^m f_p^m) \, dV, \\ a_6^m := \displaystyle\int_{V_i} \sigma_p^n f_p^n \boldsymbol{\nabla} \cdot (\sigma_u^m \boldsymbol{f}_u^m) \, dV. \end{cases} \tag{A.4}$$

A.2 Parametrised Oseen flow

In the spatial iteration of the parametrised Oseen flow system, the separated form of the residuals, now including the contribution of the linearised convection term read

$$
\begin{aligned}
R_u^n :=& \int_{V_i} \alpha_4 \boldsymbol{S}\, dV - \sum_{m=1}^{n} \alpha_5^m \int_{V_i} \boldsymbol{\nabla}\cdot(\sigma_u^m \boldsymbol{f}_u^m \otimes \boldsymbol{F})\, dV \\
&+ \int_{V_i} \boldsymbol{\nabla}\cdot\Big(D\boldsymbol{\nabla}\Big(\sum_{m=1}^{n} \alpha_6^m \sigma_u^m \boldsymbol{f}_u^m\Big)\Big)\, dV - \int_{V_i} \boldsymbol{\nabla}\Big(\sum_{m=1}^{n} \alpha_7^m \sigma_p^m f_p^m\Big)\, dV,
\end{aligned} \tag{A.5a}
$$

$$
R_p^n := -\int_{V_i} \boldsymbol{\nabla}\cdot\Big(\sum_{m=1}^{n} \alpha_7^m \sigma_u^m \boldsymbol{f}_u^m\Big)\, dV, \tag{A.5b}
$$

where the following expressions for the coefficients are devised

$$
\begin{cases}
\alpha_4 := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \phi^n \eta\, d\boldsymbol{\mathcal{I}}, & \alpha_5^m := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \phi^n \phi^m \xi\, d\boldsymbol{\mathcal{I}}, \\
\alpha_6^m := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \phi^n \phi^m \psi\, d\boldsymbol{\mathcal{I}}, & \alpha_7^m := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \phi^n \phi^m\, d\boldsymbol{\mathcal{I}}.
\end{cases} \tag{A.6}
$$

For the parametric iteration, the separated expression of the residuals is

$$
r_u^n := a_3 \eta + \sum_{m=1}^{n} \left(-a_4^m \xi + a_5^m \psi - a_6^m\right) \phi^m, \tag{A.7a}
$$

$$
r_p := -\sum_{m=1}^{n} a_7^m \phi^m, \tag{A.7b}
$$

where the coefficients depend solely on the spatial modes, namely

$$\begin{cases} a_3 := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \boldsymbol{S} \, dV, \\ a_4^m := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \big[\boldsymbol{\nabla} \cdot (\sigma_u^m \boldsymbol{f}_u^m \otimes \boldsymbol{F}) \big] \, dV, \\ a_5^m := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \big[\boldsymbol{\nabla} \cdot (D \boldsymbol{\nabla} (\sigma_u^m \boldsymbol{f}_u^m)) \big] \, dV, \\ a_6^m := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \boldsymbol{\nabla} (\sigma_p^m f_p^m) \, dV, \\ a_7^m := \displaystyle\int_{V_i} \sigma_p^n f_p^n \boldsymbol{\nabla} \cdot (\sigma_u^m \boldsymbol{f}_u^m) \, dV. \end{cases} \tag{A.8}$$

A.3 Parametrised Navier-Stokes flow

Finally, in the spatial iteration of the parametrised Navier-Stokes flow, the residuals in separable form read as

$$\begin{aligned} R_u^n := & \int_{V_i} \alpha_4 \boldsymbol{S} \, dV - \sum_{m=1}^{n} \sum_{q=1}^{n} \alpha_5^{mq} \int_{V_i} \boldsymbol{\nabla} \cdot (\sigma_u^m \boldsymbol{f}_u^m \otimes \sigma_u^q \boldsymbol{f}_u^q) \, dV \\ & + \int_{V_i} \boldsymbol{\nabla} \cdot \Big(D \boldsymbol{\nabla} \Big(\sum_{m=1}^{n} \alpha_6^m \sigma_u^m \boldsymbol{f}_u^m \Big) \Big) \, dV - \int_{V_i} \boldsymbol{\nabla} \Big(\sum_{m=1}^{n} \alpha_7^m \sigma_p^m f_p^m \Big) \, dV, \end{aligned} \tag{A.9a}$$

$$R_p^n := - \int_{V_i} \boldsymbol{\nabla} \cdot \Big(\sum_{m=1}^{n} \alpha_7^m \sigma_u^m \boldsymbol{f}_u^m \Big) \, dV, \tag{A.9b}$$

where the following expressions for the coefficients are devised

$$\begin{cases} \alpha_4 := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \phi^n \eta \, d\boldsymbol{\mathcal{I}}, & \alpha_5^{mq} := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \phi^n \phi^m \phi^q \, d\boldsymbol{\mathcal{I}}, \\ \alpha_6^m := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \phi^n \phi^m \psi \, d\boldsymbol{\mathcal{I}}, & \alpha_7^m := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \phi^n \phi^m \, d\boldsymbol{\mathcal{I}}. \end{cases} \tag{A.10}$$

For the parametric iteration, the separated expression of the residuals is

$$r_u^n := a_4\eta + \sum_{m=1}^{n}\left(-\sum_{q=1}^{n} a_5^{mq}\phi^q + a_6^m\psi - a_7^m\right)\phi^m, \tag{A.11a}$$

$$r_p := -\sum_{m=1}^{n} a_8^m \phi^m, \tag{A.11b}$$

where the coefficients depend solely on the spatial modes, namely

$$\begin{cases} a_4 := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \boldsymbol{S}\, dV, \\ a_5^{mq} := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \left[\boldsymbol{\nabla}\cdot(\sigma_u^m \boldsymbol{f}_u^m \otimes \sigma_u^q \boldsymbol{f}_u^q)\right] dV, \\ a_6^m := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \left[\boldsymbol{\nabla}\cdot(D\boldsymbol{\nabla}(\sigma_u^m \boldsymbol{f}_u^m))\right] dV, \\ a_7^m := \displaystyle\int_{V_i} \sigma_u^n \boldsymbol{f}_u^n \cdot \boldsymbol{\nabla}(\sigma_p^m f_p^m)\, dV, \\ a_8^m := \displaystyle\int_{V_i} \sigma_p^n f_p^n \boldsymbol{\nabla}\cdot(\sigma_u^m \boldsymbol{f}_u^m)\, dV. \end{cases} \tag{A.12}$$

A.4 Parametrised Spalart-Allmaras turbulence model

The separable form of the Spalart-Allmaras residual in the spatial iteration is evaluated as

$$
\begin{aligned}
R_\nu^m :=& \mathcal{R}_\nu(\widetilde{\boldsymbol{u}}_{\texttt{PGD}}^n, \widetilde{\nu}_{\texttt{PGD}}^{m-1}, \sigma_\nu^m \widetilde{\nu}_{\texttt{PGD}}^m) \\
=& -\sum_{k=1}^{n}\sum_{j=1}^{m} \alpha_8^{k,j} \int_{V_i} \boldsymbol{\nabla} \cdot \left(\sigma_\nu^j B^j \otimes \sigma_u^k \boldsymbol{f}_u^k \right) dV \\
&+ \sum_{j=1}^{m} \alpha_9^j \int_{V_i} \frac{1}{\sigma} \boldsymbol{\nabla} \cdot \left(\sigma_\nu^m \Delta B^m \boldsymbol{\nabla} \sigma_\nu^j B^j \right) dV \\
&+ \sum_{j=1}^{m} \alpha_{10}^j \int_{V_i} \frac{1}{\sigma} \boldsymbol{\nabla} \cdot \left(\nu \boldsymbol{\nabla} \sigma_\nu^j B^j \right) dV \\
&+ \sum_{j=1}^{m}\sum_{l=1}^{m} \alpha_{11}^{j,l} \int_{V_i} \frac{1}{\sigma} \boldsymbol{\nabla} \cdot \left(\sigma_\nu^l B^l \boldsymbol{\nabla} \sigma_\nu^j B^j \right) dV \\
&+ \sum_{j=1}^{m}\sum_{l=1}^{m} \alpha_{11}^{j,l} \int_{V_i} \frac{C_{b2}}{\sigma} \boldsymbol{\nabla} \sigma_\nu^j B^j \cdot \boldsymbol{\nabla} \sigma_\nu^l B^l \, dV + \sum_{j=1}^{m} \int_{\boldsymbol{\mathcal{I}}} \int_{V_i} C_{b1} \alpha_{12}^j \sigma_\nu^j B^j \, dV \\
&- \sum_{j=1}^{m}\sum_{l=1}^{m} \int_{V_i} C_w \alpha_{13}^{j,l} \frac{\sigma_\nu^j B^j \sigma_\nu^l B^l}{y^2} \, dV.
\end{aligned}
\tag{A.13}
$$

where the following expressions for the coefficients are devised

$$
\begin{cases}
\alpha_8^{k,j} := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \xi^m \xi^j \phi^k \, d\boldsymbol{\mathcal{I}}, & \alpha_9^j := \displaystyle\int_{\boldsymbol{\mathcal{I}}} (\xi^m)^2 \, \xi^j \, d\boldsymbol{\mathcal{I}}, \\
\alpha_{10}^j := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \xi^m \xi^j \, d\boldsymbol{\mathcal{I}}, & \alpha_{11}^{j,l} := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \xi^m \xi^j \xi^l \, d\boldsymbol{\mathcal{I}} \\
\alpha_{12}^j := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \xi^m \xi^j \tilde{S} \, d\boldsymbol{\mathcal{I}}, & \alpha_{13}^{j,l} := \displaystyle\int_{\boldsymbol{\mathcal{I}}} \xi^m \xi^j \xi^l f_w \, d\boldsymbol{\mathcal{I}}.
\end{cases}
\tag{A.14}
$$

with α_{12} and α_{13} being spatial functions.

For the parametric iteration, the separated expression of the residuals is

$$\begin{aligned} r_{nu}^{m} :=& -\sum_{k=1}^{n}\sum_{j=1}^{m} a_{11}^{k,j}\xi^{j}\phi^{k} + \sum_{j=1}^{m} a_{12}^{j}\xi^{j} + \sum_{j=1}^{m}\sum_{l=1}^{m} a_{13}^{j,l}\xi^{j}\xi^{l} \\ &+ \sum_{j=1}^{m}\sum_{l=1}^{m} a_{14}^{j,l}\xi^{j}\xi^{l} + \sum_{j=1}^{m} a_{15}^{j}\xi^{j} - \sum_{j=1}^{m}\sum_{l=1}^{m} a_{16}^{j,l}\xi^{j}\xi^{l}, \end{aligned} \tag{A.15}$$

where the coefficients are defined as

$$\left\{\begin{aligned} a_{11}^{k,j} &:= \int_{V_i} \sigma_{\nu}^{m} B^{m} \boldsymbol{\nabla}\cdot\left(\sigma_{\nu}^{j} B^{j} \otimes \sigma_{u}^{k} \boldsymbol{f}_{u}^{k}\right)\, dV, \\ a_{12}^{j} &:= \int_{V_i} \sigma_{\nu}^{m} B^{m} \frac{1}{\sigma} \boldsymbol{\nabla}\cdot\left(\nu \boldsymbol{\nabla} \sigma_{\nu}^{j} B^{j}\right)\, dV, \\ a_{13}^{j,l} &:= \int_{V_i} \sigma_{\nu}^{m} B^{m} \frac{1}{\sigma} \boldsymbol{\nabla}\cdot\left(\sigma_{\nu}^{l} B^{l} \boldsymbol{\nabla} \sigma_{\nu}^{j} B^{j}\right)\, dV, \\ a_{14}^{j,l} &:= \int_{V_i} \sigma_{\nu}^{m} B^{m} \frac{C_{b2}}{\sigma} \boldsymbol{\nabla} \sigma_{\nu}^{j} B^{j} \cdot \boldsymbol{\nabla} \sigma_{\nu}^{l} B^{l}\, dV, \\ a_{15}^{j} &:= \int_{V_i} \sigma_{\nu}^{m} B^{m} C_{b1} \tilde{S} \sigma_{\nu}^{j} B^{j}\, dV, \\ a_{16}^{j,l} &:= \int_{V_i} \sigma_{\nu}^{m} B^{m} C_{w} f_{w} \frac{\sigma_{\nu}^{j} B^{j} \sigma_{\nu}^{l} B^{l}}{y^{2}}\, dV, \end{aligned}\right. \tag{A.16}$$

Appendix B

simpleFoam: the semi-implicit method for pressure linked equations in OpenFOAM

In OpenFOAM, the steady laminar Navier-Stokes equations are approximated by means of an iterative procedure, namely simpleFoam. This algorithm implements the SIMPLE method proposed in [87]. SIMPLE is a fractional-step Chorin-Temam projection method [106, Sect. 3.7] that has been extensively studied in the literature [50, 49]. A description of the relationship between SIMPLE and Chorin-Temam projection methods is detailed in [91, Sect. 17.7, 17.8], whereas the implementation details of simpleFoam are provided in the official OpenFOAM documention [107].

First, an intermediate velocity $\boldsymbol{u}^k$ is computed starting from the momentum equation and neglecting the contribution of pressure, see Equation (B.1a); second, the step involving the incompressibility constraint is rewritten in terms of a Poisson equation for the pressure p, see Equation (B.1b); eventually, a correction is applied to the intermediate velocity field to determine the final value $\boldsymbol{u}$ in Equation (B.1c). Special attention is required to impose the correct set of boundary conditions in each step of the algorithm [64].

$$\begin{cases} \dfrac{\boldsymbol{u}^k - \boldsymbol{u}^{k-1}}{\Delta t} + \boldsymbol{\nabla}\cdot(\boldsymbol{u}^k \otimes \boldsymbol{u}^{k-1}) - \boldsymbol{\nabla}\cdot(\nu \boldsymbol{\nabla} \boldsymbol{u}^k) = \boldsymbol{s} & \text{in } \Omega, \\ \boldsymbol{u}^k = \boldsymbol{u}_D & \text{on } \Gamma_D, \\ \boldsymbol{n}\cdot(\nu \boldsymbol{\nabla} \boldsymbol{u}^k) = \boldsymbol{t} & \text{on } \Gamma_N, \end{cases} \tag{B.1a}$$

$$\begin{cases} \boldsymbol{\nabla}\cdot(\boldsymbol{\nabla} p) = \dfrac{1}{\Delta t} \boldsymbol{\nabla}\cdot\boldsymbol{u}^k & \text{in } \Omega, \\ \boldsymbol{n}\cdot\boldsymbol{\nabla} p = 0 & \text{on } \Gamma_D, \\ \boldsymbol{n} p = \boldsymbol{0} & \text{on } \Gamma_N, \end{cases} \tag{B.1b}$$

$$\boldsymbol{u} = \boldsymbol{u}^k - \Delta t \boldsymbol{\nabla} p. \tag{B.1c}$$

Remark 10. *A variant of the above numerical scheme, known as* incremental *Chorin-Temam projection method, is obtained by adding the term* $-\boldsymbol{\nabla} p^k$ *to the right-hand side of Equation* (B.1a) *to compute the velocity prediction. Hence, the Poisson problem for pressure in Equation* (B.1b) *is solved to compute a pressure variation* Δp*, whereas the correction step in Equation* (B.1c) *remains unchanged, now being* $p{=}p^k{+}\Delta p$*.*

Note that the algorithm in Equation (B.1) may also be rewritten in the framework of an algebraic splitting method [92]. For a complete introduction to the subject, interested readers are referred to [31, Sect. 6.7].

For cases residing in the turbulent regime, an extra step is included in the SIMPLE algorithm, namely the solution of the turbulence model. Similarly to the laminar case, the algorithm computes an intermediate velocity from Equation (B.2a), notice that an effective viscosity is used in the diffusion term, instead of the physical viscosity coefficient. Then, the Poisson equation for pressure p is assembled and solved, Equation (B.2b), and eventually, velocity $\boldsymbol{u}$ is explicitly corrected using the newly computed pressure, Equation (B.2c). Before resetting the loop back to Equation (B.2a), the turbulence model, in this case the Spalart-Allmaras, is solved using Equation (B.3a) to compute a state viscosity, $\tilde{\nu}$, which is in turn used to update the effective viscosity (Equation B.3b).

$$\begin{cases} \dfrac{\boldsymbol{u}^k - \boldsymbol{u}^{k-1}}{\Delta t} + \boldsymbol{\nabla}\cdot(\boldsymbol{u}^k \otimes \boldsymbol{u}^{k-1}) - \boldsymbol{\nabla}\cdot(\nu_{eff}\boldsymbol{\nabla}\boldsymbol{u}^k) = \boldsymbol{s} & \text{in } \Omega, \\ \boldsymbol{u}^k = \boldsymbol{u}_D & \text{on } \Gamma_D, \\ \boldsymbol{n}\cdot(\nu_{eff}\boldsymbol{\nabla}\boldsymbol{u}^k) = \boldsymbol{t} & \text{on } \Gamma_N, \end{cases} \tag{B.2a}$$

$$\begin{cases} \boldsymbol{\nabla}\cdot(\boldsymbol{\nabla}p) = \dfrac{1}{\Delta t}\boldsymbol{\nabla}\cdot\boldsymbol{u}^k & \text{in } \Omega, \\ \boldsymbol{n}\cdot\boldsymbol{\nabla}p = 0 & \text{on } \Gamma_D, \\ \boldsymbol{n}p = \boldsymbol{0} & \text{on } \Gamma_N, \end{cases} \tag{B.2b}$$

$$\boldsymbol{u} = \boldsymbol{u}^k - \Delta t\boldsymbol{\nabla}p, \tag{B.2c}$$

$$\begin{cases} \boldsymbol{\nabla}\cdot(\tilde{\nu}\otimes\boldsymbol{u}) - \dfrac{1}{\sigma}\boldsymbol{\nabla}\cdot[(\nu+\tilde{\nu})\boldsymbol{\nabla}\tilde{\nu}] - \dfrac{C_{b2}}{\sigma}|\boldsymbol{\nabla}\tilde{\nu}|^2 & \\ \qquad\qquad = C_{b1}\tilde{S}\tilde{\nu} - C_{w1}f_w\left(\dfrac{\tilde{\nu}}{y}\right)^2 & \text{in } \Omega, \\ \tilde{\nu} = \tilde{\nu}_D & \text{on } \Gamma_D, \\ \boldsymbol{n}\cdot\boldsymbol{\nabla}\tilde{\nu} = 0 & \text{on } \Gamma_N, \end{cases} \tag{B.3a}$$

$$\begin{cases} \nu_t = f_{v1}\tilde{\nu}, \\ \nu_{eff} = \nu + \nu_t. \end{cases} \tag{B.3b}$$

B.1 `simpleFoam`: a Stokes solver variation

In this section the adaptation of the `simpleFoam` algorithm to solve the Stokes problem is discussed. Considering how the non-linear term is missing from the momentum equilibrium, there is no need to linearise the system. A velocity prediction is computed from equation B.4a and is used to solve the Poisson equation for pressure B.4b. On the final iteration only, the velocity is updated based on the computed pressure B.4c.

$$\begin{cases} \dfrac{\boldsymbol{u}^k - \boldsymbol{u}^{k-1}}{\Delta t} - \boldsymbol{\nabla}\cdot(\nu \boldsymbol{\nabla} \boldsymbol{u}^k) = \boldsymbol{s} & \text{in } \Omega, \\ \boldsymbol{u}^k = \boldsymbol{u}_D & \text{on } \Gamma_D, \\ \boldsymbol{n}\cdot(\nu \boldsymbol{\nabla} \boldsymbol{u}^k) = \boldsymbol{t} & \text{on } \Gamma_N, \end{cases} \tag{B.4a}$$

$$\begin{cases} \boldsymbol{\nabla}\cdot(\boldsymbol{\nabla} p) = \dfrac{1}{\Delta t}\boldsymbol{\nabla}\cdot\boldsymbol{u}^k & \text{in } \Omega, \\ \boldsymbol{n}\cdot\boldsymbol{\nabla} p = 0 & \text{on } \Gamma_D, \\ \boldsymbol{n} p = \boldsymbol{0} & \text{on } \Gamma_N, \end{cases} \tag{B.4b}$$

$$\boldsymbol{u} = \boldsymbol{u}^k - \Delta t \boldsymbol{\nabla} p. \tag{B.4c}$$

B.2 `simpleFoam`: an Oseen solver variation

In this section the adaptation of the `simpleFoam` algorithm to solve the Oseen problem is discussed. Similarly to the Stokes problem, the non-linear term is missing from the momentum equilibrium and there is no need to linearise the system. Initially a pre-scribed velocity field $\boldsymbol{u}_0$ is read to build the linearised convection term. A velocity prediction is computed from equation B.5a and is used to solve the Poisson equation for pressure B.5b. On the final iteration only, the velocity is updated based on the computed pressure B.5c.

$$\begin{cases} \dfrac{\boldsymbol{u}^k - \boldsymbol{u}^{k-1}}{\Delta t} + \boldsymbol{\nabla}\cdot(\boldsymbol{u}^k \otimes \boldsymbol{u}_0) - \boldsymbol{\nabla}\cdot(\nu \boldsymbol{\nabla} \boldsymbol{u}^k) = \boldsymbol{s} & \text{in } \Omega, \\ \boldsymbol{u}^k = \boldsymbol{u}_D & \text{on } \Gamma_D, \\ \boldsymbol{n}\cdot(\nu \boldsymbol{\nabla} \boldsymbol{u}^k) = \boldsymbol{t} & \text{on } \Gamma_N, \end{cases} \tag{B.5a}$$

$$\begin{cases} \boldsymbol{\nabla}\cdot(\boldsymbol{\nabla} p) = \dfrac{1}{\Delta t} \boldsymbol{\nabla}\cdot \boldsymbol{u}^k & \text{in } \Omega, \\ \boldsymbol{n}\cdot\boldsymbol{\nabla} p = 0 & \text{on } \Gamma_D, \\ \boldsymbol{n} p = \boldsymbol{0} & \text{on } \Gamma_N, \end{cases} \tag{B.5b}$$

$$\boldsymbol{u} = \boldsymbol{u}^k - \Delta t \boldsymbol{\nabla} p. \tag{B.5c}$$