Západočeská univerzita v Plzni
Fakulta aplikovaných věd

Isogeometrická analýza pro modelování nestlačitelného turbulentního proudění

Ing. Eva Turnerová

disertační práce
k získání akademického titulu doktor
v oboru Aplikovaná matematika

Školitel: Doc. Ing. Marek Brandner, Ph.D.
Katedra matematiky

Plzeň 2019

University of West Bohemia
Faculty of Applied Sciences

Isogeometric analysis for incompressible turbulent flow

Ing. Eva Turnerová

doctoral thesis
submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy (Applied Mathematics)

Supervisor: Doc. Ing. Marek Brandner, Ph.D.
Department of Mathematics

Pilsen 2019

Abstrakt

Tato práce se zabývá aplikací isogeometrické analýzy na úlohy nestlačitelného turbulentního proudění. Proudění newtonovské nestlačitelné tekutiny je popsané pomocí Navierových–Stokesových rovnic. Navzdory rychlému vývoji numerickým metod pro přímou numerickou simulaci (DNS - Direct Numerical Simulation) proudění a zejména simulaci pohybu velkých vírů (LES - Large Eddy Simulation) jsou středované Navierovy–Stokesovy rovnice (RANS – Reynolds–Averaged Navier–Stokes) stále základním nástrojem pro numerické řešení turbuletního proudění. Důvodem jsou nízké výpočetní nároky. Celé spektrum turbulentních vírů je totiž modelováno. Soustava středovaných rovnic není ale uzavřená a musí být doplněna takzvaným modelem turbulence.

Řešení konvektivně dominantních úloh obsahuje mezní vrstvy, což jsou úzké oblasti, ve kterých se prudce mění hodnota řešení. Tyto mezní vrstvy lze je ovšem velmi obtížně vyřešit vzhledem k tomu, že je šířka mezní vrstvy často užší než je velikost výpočetní sítě. Výsledným efektem je značný nárůst numerických (nefyzikálních) oscilací, které způsobují ztrátu stability a přesnosti diskrétního řešení. Našim cílem je tedy zajistit stabilitu řešení nestlačitelného proudění při vysokých Reynoldsových číslech, která se vyskytují v reálných aplikacích.

K redukci numerických oscilací se používají metody jejichž cílem je vylepšení stability diskrétního řešení, ale bez ztráty přesnosti. Přístupy, které byly původně navržené pro metodu konečných prvků, využíváme pro isogeometrickou diskretizaci. SUPG (streamline upwind/Petrov-Galerkin) a SOLD (spurious oscillations at layers diminishing) metody jsou standardní přístupy používané pro metody založené na spojité Galerkinově metodě. Stabilizační metody přidávají umělou vazkost, jejíž množství regulujeme vhodnou volbou stabilizačního parametru. Přestože existuje mnoho variant stabilizačních parametrů, žádná z nich není optimální pro obecnou úlohu. V této práci se věnujeme studiu vlivu různých stabilizačních parametrů na výsledné numerické řešení.

Stabilizační metody nejprve aplikujeme na obyčejnou časově závislou advekčně difúzní rovnici řešenou na víceplátových oblastech s pravidelnou i nepravidelnou křivočarou sítí. Závěry a pozorování z této části jsou aplikované na mnohem obtížnější časově závislý advekčně difúzně reakční turbulentní model s dominantním advekčním a reakčním členem a nekonstantními koeficienty. Numerické experimenty prokazují vliv zvoleného stabilizačního parametru a také vliv třídy spojitosti bázových funkcí na výsledné diskrétní řešení.

Klíčová slova:

Nestlačitelné proudění, časově závislá advekčně difúzně reakční rovnice, Navierovy–Stokesovy rovnice, RANS rovnice, turbulentní model, isogeometrická analýza, spojitá Galerkinova metoda, B-spline bázové funkce, multipatch, křivočará síť, nekonstantní koeficienty, numerické oscilace, stabilizace, SUPG, SOLD, stabilizační parametr

Abstract

This doctoral thesis deals with application of isogeometric analysis to incompressible turbulent flow problems. The Navier–Stokes equations are the basis for computational modeling of the flow of an incompressible Newtonian fluid. Despite the development of the numerical methods for direct numerical simulation and especially simulation of the large eddies, Reynolds–Averaged Navier–Stokes (RANS) approach based on the time-averaging of the Navier–Stokes equations is the most common method to describe turbulent flow so far. The reason is the lower memory requirements since the whole range of the scales is modeled and only the effect of the turbulence on the mean flow behavior is considered. The fundamental problem of the solution of the RANS equations is to close the problem by a model of turbulence.

Since IgA is continuous Galerkin-based method, the numerical solution of advection dominated problems is usually polluted by spurious (unphysical) oscillations, which causes loss of accuracy and stability. This type of problems contain sharp layers, where the solution gradients are very large. These layers usually arise from the discontinuous, non-smooth or rapidly changing data, which are typical for the turbulent flow, or generally for advection dominated problems. We focus on keeping the stability of the solution at high Reynolds numbers, which is significant in practical applications.

The stabilization techniques are investigated which improve the stability, however, without degrading accuracy. The approaches originally intended for the finite element method are employed for isogeometric discretization including the streamline upwind/Petrov-Galerkin (SUPG) method and the spurious oscillations at layers diminishing (SOLD) method. The amount of the added numerical diffusion is controlled by a suitable choice of the stabilization parameter. However, the formulas for the parameters are not optimal for general problem and hence the influence of various parameters is studied.

We employ the stabilization methods first for a simple time-dependent advection-diffusion equations on multipatch domains with both regular and irregular curvilinear meshes. The observations are applied to more complicated advection–reaction dominated system of unsteady advection–diffusion–reaction turbulence model with non-constant coefficients. The rate of stabilization is not affected only by the parameters of the given stabilization method but also by the choice of the regularity of the basis functions.

Keywords:

Incompressible flow, time-dependent advection–diffusion–reaction equations, Navier–Stokes equations, RANS equations, turbulence model, high-order spatial discretization, isogeometric analysis, continuous Galerkin method, B-spline basis function, multipatch, curvilinear meshes, tensor-product refinement, small diffusion, non-constant coefficients, spurious oscillations, stabilization, SUPG, SOLD, crosswind diffusion, isotropic diffusion, stabilization parameter

Acknowledgement

The research presented in this doctoral thesis would not have been achievable without support that I received from many people. I would like to express my deepest appreciation to my supervisor Doc. Ing. Marek Brandner, Ph.D., for all his guidance, patience and motivation throughout my study. His expertise was a source of indispensable help. I would also like to thank Doc. Ing. Bohumír Bastl, Ph.D., for his assistance in the theory and application regarding isogeometric analysis, fluid flow and related issues.

Many thanks belong to all my colleagues from No. 678727 and No. 19-04006S projects for pleasant cooperation during the past six years.

Last but certainly not least, I would like to thank my family, who have always supported me in any way they could.

I acknowledge the financial support of European Union's Horizon 2020 research and innovation programme under grant No. 678727 and Czech Science Foundation GA ČR (Grantová agentura České Republiky) grant No. 19-04006S.

Declaration

I do hereby declare that to my best knowledge, this doctoral thesis is my original work and that any ideas, techniques, or any other materials included are fully acknowledged with the standard referencing practices.

....
Eva Turnerová

Contents

1 Introduction

Fluid mechanics is the study of fluids, motion of the fluid and the forces acting on them. This science belongs to the most challenging areas of research (in both the theory and the numerical implementation). The development of the fluid mechanics is influenced by the requirement of the modern technologies for deeper understanding of the behavior of the real fluids and the most accurate numerical simulations. Some of the fields of application are meteorology, biological sciences (e.g. the study of blood flow), mechanical engineering (e.g. marine and wind/water turbine engineering), astrophysics.

Computational fluid dynamics (CFD) is a branch of fluid mechanics that uses numerical mathematics to analyze and solve problems that involve fluid flows. Modern engineering industry requires to improve numerical simulations to achieve as real behavior of the fluid as possible. Therefore, this provides higher and higher demands on the mathematical research to develop more sophisticated numerical methods with high accuracy and reasonable computational cost. However, the fluid flow is mostly turbulent in engineering applications, which is still very difficult to simulate, and special attention has to be invested if we are interested in turbulent fluid flow.

There is no clear and accurate definition of the turbulence and even the current researchers differ in their description of the turbulence and they follow distinct authors, [32, 116, 10]. The turbulence is the property of the flow such that the turbulent flow consists of the whole spectrum of the turbulent eddies, [30]. The motion of the fluid particles is unorganized, the trajectories and eddies intersect each other, which leads to chaotic flow with rapid variations of the velocity and pressure. Even if the flow is chaotic, the same structures and flow characteristics of the turbulent flow arise for unchanging boundary conditions. As a result, the turbulence can be considered as quasi-deterministic process and the motion of the fluid is supposed to be described by the Navier–Stokes equations (see e.g. [116, 12, 10, 114, 105, 37]).

The Navier–Stokes equations are used to model wide range of problems such as weather, water and some air flow and study of blood flow. System of (d+1) equations has to be solved to simulate fluid flow in d-dimensional space. These equations are intensively studied from both a numerical and a theoretical point of view. The existence, uniqueness and stability of weak solution is not complete, because the uniqueness of the weak solution is not proved for three dimensional nonstationary problem on arbitrarily long time interval for any initial condition, [37, 17].

The turbulent flow can be simulated by solving directly the Navier–Stokes equations for the given boundary and initial conditions, which is known as direct numerical simulation (DNS). However, it means that the whole range of the scales (sizes of the eddies) must be resolved up to the smallest scales of the flow [116, 84]. Then the number of grid points in spatial discretization has to be proportional to $Re^{9/4}$, where Re is the Reynolds number, otherwise the simulation becomes unstable. The computer storage and runtime requirements increase with increasing Reynolds number. Thus, the direct approach can be applied only for low Reynolds number flows in simple geometries.

Turbulent flow is extensively studied in engineering practice and big effort is made to produce techniques which describe turbulent flow for a wide range of applications of the external and internal flows. The most common approaches are RANS (Reynolds Average Navier–Stokes) and LES (Large Eddy Simulation).

RANS approach is probably the most widely used method for simulation of the turbulent flow. All turbulent scales are modeled using RANS, [32, 82, 116]. It is based on Reynolds decomposition, i.e. the velocity and pressure variables are decomposed into a mean (time–averaged/ensemble averaged) component and the fluctuating component and the Navier–Stokes equations are integrated over a time interval. The attention is focused only on the mean flow and the effects of the turbulence on the mean flow and the turbulent fluctuations are not resolved in detail, which is often sufficient in practice. The time–averaging (or ensemble averaging [82]) causes that extra unknown terms appear (Reynolds stresses).

Boussinesq hypothesis is usually applied to approximate the Reynolds stresses (cf. [116, 32, 39, 40, 90]), which gives arise to the turbulence models. Various turbulence models have been proposed to satisfy different types and conditions of the fluid flow. The improvement and development of new turbulence models are still the subject of extensive research resulting in increasing accuracy of the turbulent flow predictions, which is closely related to the development in computer technology. In modern engineering applications, the commonly used turbulence models are e.g. the one–equation Spalart–Allmaras and two–equation SST $k - \omega$ turbulence models, [116, 71, 96].

The third mathematical model is the LES approach, which can be classified between DNS and RANS methods considering the memory requirements. The LES equations are derived from the Navier–Stokes equations such that only the motion of the large eddies is simulated. Small eddies are filtered and their motion is modeled by some subgrid–scale model (SGS), [10, 116].

Another approaches have been developed to avoid the problems of the three above mentioned methods. The grid refinement is required near the walls for, say, more accurate LES simulations, which makes this approach barely applicable even on the largest supercomputers so far. Therefore, LES remains still very limited in practice. The so called hybrid DES (Detached eddy simulation) has been developed, [33, 115, 90]. Because of time consuming LES simulation, RANS approach is employed in near wall regions and the rest of the flow is simulated by LES. The RANS method enables the use of coarser grid near the boundary, which reduces the computational cost of DES. However, the difficulty of DES is to ensure the smooth transition from RANS to LES model.

It was already mentioned that the classical LES is derived by the space filtering of the Navier–Stokes equations with an explicit SGS model. Then the continuous problem is discretized using some numerical method. Contrarily, the approach of a relatively new ILES (Implicit Large Eddy Simulation) method is different, [43, 80]. The numerical method is applied to the original equations (unfiltered Navier–Stokes) and the truncation error of the discretization is then used to model the effects of the unresolved scales. Hence, the additional SGS model is no longer needed. The numerical dissipation resulting from the discretization scheme is referred to as an implicit SGS model. Although, the approach of

ILES is simple, and can be applied for general problems, it is not commonly used in the turbulence modeling community so far, probably because of the lack of understanding of the relation between the numerical scheme, the SGS model and the LES technique. ILES approach was applied with a number of different schemes (e.g. finite volume methods, finite difference methods, spectral methods, discontinuous Galerkin method etc). Simulations based on ILES give satisfactory results with accurate predictions, but the computational requirements are high similarly to the classical LES method and hence the application of ILES is limited to low Reynolds number flows.

Variational MultiScale (VMS) method [51, 10, 11] avoids some problems of LES, e.g. the definition of the appropriate boundary condition for the large scales, [51, 56]. The VMS simulation is based on the variational projections of the large scales into appropriate spaces. The advantage is that the VMS approach covers both turbulence modeling and stabilization, which has to be added separately in case of RANS or LES if necessary. But this approach is relatively new and it is still the subject of research.

An important issue is the numerical treatment of the equations in regions near the boundary for the simulation of the turbulent flow. The wall–functions and the low–Reynolds number method are two possibilities for the near wall treatment, [14]. The wall function approach is based on the so called law of the wall, [116, 32]. The idea is to apply the near wall behavior of the turbulent variables described by algebraic expressions instead of resolving it. The advantage is that a coarser grid can be used and thus the lower memory requirements are needed. On the other hand, the equations are solved through all the boundary layers using low–Reynolds number method and thus a very fine grid is necessary near the solid walls.

The study of the weak imposition of Dirichlet boundary conditions is introduced in [7]. The Dirichlet conditions are not satisfied explicitly, but extra terms are added to the variational formulation to enforce the conditions weakly as Euler–Lagrange conditions. It has been shown that the weak imposition of Dirichlet boundary conditions gives similar results to the wall function technique, [7]. The idea was later modified, [8], as a combination of weakly imposed boundary conditions with residual based turbulence modeling. Moreover, weakly imposed boundary conditions provide more accurate results compared to the classical approach, [8]. Another advantage is the ability to reduce the numerical oscillations, [7, 8].

The development of the numerical discretization methods for the fluid flow simulation is fundamental in practice. The goal is to find a numerical method, which is the most efficient, and a numerical solution of the given problem, which is the most accurate. There are many numerical methods which can be applied to approximate the solution of incompressible fluid flow, for example finite difference methods, finite volume methods, finite element methods, spectral methods etc. Finite difference method (FDM) [101, 83] is one of the oldest, it is simple to implement, but it is rarely applied for fluid flow simulation because of the difficult employment to the complex geometries. Also, very fine grid usually has to be applied to get sufficient accuracy.

On the other hand, finite volume method (FVM) is a common approach and it is frequently used in (commercial) CFD codes because it follows the continuous conservation laws of physics, see e.g. [116, 101]. FVM is also easy to implement, it requires only low memory, even in the case of turbulent simulation where finer grids are used. Further, FVM is robust and flexible. One of the drawbacks is that too much diffusion can be added using methods of lower (first and second) order. However, if the high-order FVM is employed, the size of reconstruction stencils is increasing, which becomes computationally expensive.

The finite element method (FEM) [101, 83] is also commonly used for solving fluid flow simulation. Nevertheless, this method is not so popular in the CFD community because of higher memory requirements [104]. FEM is able to deal relatively easily with problems that are defined on complex geometries. This method is one of the Galerkin approaches which are based on weak formulation of the problem and polynomial approximation of the solution is usually used.

The discontinuous Galerkin method (DG) is widely used in practice, e.g. for solving compressible and incompressible flow with high Reynolds numbers, but also for the shallow water equations, turbomachinery or magneto–hydrodynamics, [23, 22, 5, 87]. The DG methods combines features of the Galerkin finite element method and the Riemann solver based approach (applied to discontinuous piecewise polynomial approximations). Thus, the solution is represented as a FEM approximation within each element and the advection terms are resolved similarly to FVM. It results in a high order accurate solution. The DG method also benefits in its ability to handle with the complicated geometries, hanging nodes and nonconforming meshes. Besides, the importance of this method is appreciated in situations with steep gradients or shocks.

Isogeometric analysis is a recently developed spline modification of the finite element method based on B–spline/NURBS objects (cf. e.g. [50, 27, 54]). The triangular/tetrahedral meshes typically used in FEM are replaced by meshes composed of parts of NURBS surfaces/volumes representing a computational domain. This is suitable for consequent computations, because it allows to avoid the time–consuming step of generating triangular/tetrahedral meshes. Moreover, since the discretization of a computational domain is always exact, this approach reduces errors in the computational analysis and it is also suitable for formulation of automatic shape optimization algorithms.

Another approaches which offer new possibilities for turbulent flow computations can be introduced, e.g. Johnson and co-workers [46, 48] use an adaptive finite element method with a duality–based a posteriori error control to predict drag and lift accurately with slip boundary conditions. Then the thin boundary layers do not have to be resolved, which leads to relatively low memory requirements. This theory follows from their resolution of the d'Alembert's paradox, which is in contradiction to commonly accepted Prandtl's suggestions that the drag and lift emanate from viscous boundary layers even with vanishing viscosity.

The disadvantage of all the mentioned discretization methods is that they suffer from two main sources of instabilities. One is an incompatibility of pressure and velocity such that the discretization of velocity and pressure has to satisfy special requirements. Indeed, the so called inf–sup condition (or Ladyženskaja–Babuška–Brezzi condition (LBB)) has to be fulfilled to guarantee the stability, [116, 36, 35, 104, 12]. Not satisfied LBB condition leads to instability which usually appears as oscillations, primarily in the pressure approximation. In practice, it is common to apply an extension of the classical Taylor–Hood finite elements (see e.g. [116, 36, 35]) for the isogeometric analysis to satisfy the inf–sup condition (106), i.e., to choose an unequal order of the spatial discretization for the velocity and pressure approximations, [15, 89, 17]. If the approximations of the equal order are rather used, the additional stabilization, the Pressure Stabilized/Petrov–Galerkin (PSPG) method [110, 12], has to be implemented.

The other source of instability is the possible dominance of the advective term over the diffusion term in the problem of Navier–Stokes equations, i.e. in case of the turbulent flow. Also, the system of RANS equations is closed with the turbulence model, which is the advection dominated system of time–dependent advection–diffusion–reaction equations with the source term. The reaction term can be also significant in some parts of the domain and thus the problem becomes reaction dominated, too.

The advection dominated phenomena are characterized by skew-symmetric differential operators and the numerical solutions of this type of problems contain sharp layers (contact discontinuities, shock waves, boundary layers). The width of the sharp layers is usually smaller than the mesh size and thus the layers cannot be resolved properly. If the standard finite difference method, finite volume method or continuous Galerkin method are used, it leads to unwanted spurious (nonphysical) oscillations in the numerical solution, which causes loss of accuracy and stability.

Many stabilization techniques have been proposed to remove (or to diminish) the spurious oscillations without leading to excessive smearing of discontinuities or layers. Linear and nonlinear schemes are distinguished (we call the scheme linear if it is linear and applied to a linear partial differential equation).

There are many FD and FV linear approaches, e.g. upwind, QUICK (Quadratic Upstream Interpolation for Convective Kinematics) and QUICKEST (QUICK with Estimated Streaming Terms) schemes (cf. e.g. [116, 88]). Other schemes are based on the continuous or discontinuous Galerkin methodology (DGFEM, SUPG, GLS, CIP, LPS etc.). Nevertheless, linear schemes are neither monotone nor positive (we recall the Godunov barrier theorem for the pure advection case).

The nonlinear schemes based on FDM, FVM and DGFEM are adaptive. They are solution dependent and based on different techniques (flux limited schemes, flux correction transport, slope limited schemes, TVD schemes, MUSCL scheme, ENO and WENO schemes [88, 4]). There are also many nonlinear stabilization techniques used within the continuous Galerkin method. These techniques are called spurious oscillation at layer diminishing (SOLD) schemes, [59, 60, 57]. The aim of these approaches is to remove spurious oscillations in the multidimensional case (by adding an stabilization term to a linear stabilization scheme). These nonlinear stabilizations generally improve solutions obtained by linear methods, but they are not usually able to remove oscillations completely. A typical representative of the SOLD method is crosswind artificial diffusion (cf. e.g. [58, 59, 18, 28, 24]).

To obtain an oscillation-free solution, a genuinely nonlinear scheme has to be used (i.e., schemes that are adaptive and nonlinear even in one dimension for linear problem), which is e.g. the flux-corrected transport (FCT) as the most common scheme, see [72, 68, 86, 55].

Another important issue is the time discretization of the unsteady Navier–Stokes or RANS problem. However, the turbulent motion of the fluid is very fast and it changes chaotically together with the eddies of various length scales. Thus, the turbulent simulation requires very accurate approximation in time. The goal is to apply the time discretization method which gives solution of certain level of precision in acceptable computational time.

The explicit Euler method is the simplest and fastest time discretization method, which is however limited by the small time step sizes resulting in high number of iterations. Moreover, the continuity equation cannot be exactly satisfied using the explicit method, which leads to increasing error with the number of time steps, [17]. The implicit Euler and semi-implicit Crank–Nicholson methods are more popular for the practical applications, [17, 105]. The advantage is a possible choice of the higher time step size. But the resulting discrete problem is nonlinear. Crank–Nicholson is a second order method which yields the higher order of convergence, [17].

The so called operator splitting methods have been developed to ensure unconditional stability and high accuracy (cf. e.g. [17, 34, 29, 35]). The aim of these methods is to decouple the problem associated with the nonlinear advective term, into several subproblems which are easy to solve. The standard operator splitting methods are two–stage scheme (Peaceman–Rachford scheme), [34], and three–stage scheme (θ–scheme), [17, 105]. The two–stage scheme is a second order accurate solving one nonlinear problem and one generalized Stokes problem. However, it is not asymptotically stable. The three–stage variant eliminates this difficulty and it becomes unconditionally stable under certain set of parameter θ such that two generalized Stokes problems and only one nonlinear problem have to be solved.

The spatial and time discretizations of the Navier–Stokes equations usually lead to the nonlinear system of equations. Newton's and Picard's methods are two classical linearization procedures, [35, 99, 75]. Solving the Navier–Stokes problem thus requires nonlinear iteration such that linearized problem is solved at every step. It produces a sequence of approximate solutions which converge to the solution of the weak formulation. Because the iteration process is applied, an initial guess has to be chosen. The Newton's method gives quadratic convergence, but the disadvantage is that the initial guess needs to be closer and closer to the exact solution as the Reynolds number increases, [35].

The linearized system of equations can be then solved by direct method in each iteration step, [99, 75]. However, simulation of turbulent flow implies large number of degrees of freedom and the system thus involves thousands (or even millions) of equations. Although direct methods are robust and accurate they are almost not feasible, especially in 3D.

The alternative is to apply iterative methods which approximate the solution with required accuracy. There are three main classes of iterative methods: stationary iterative methods, nonstationary iterative methods (especially Krylov subspace methods) and multilevel iterative correction techniques including the multigrid method, [35, 99]. The stationary iterative methods form simple process which is repeated in each iteration. Although these methods are simple to derive and implement, the convergence is guaranteed only in the limit case, [99]. The main representatives are the Jacobi method, the Gauss–Seidel method or the SOR relaxation method, [75].

On the other hand, Krylov subspace methods are commonly used for turbulent flow problems. They form special basis (Krylov sequence) as the initial residual times powers of matrix of the system, [99, 108, 75, 77, 35]. The approximate solution minimize the residual in appropriate norm according to the formed Krylov subspace basis. The typical representative is the conjugate gradient method (CG) for symmetric positive definite matrix. Other Krylov methods are the generalized minimal residual method (GMRES) and the biconjugate gradient method (BiCG) for indefinite problems. The GMRES method is usually restarted after several iterations such that the last solution from the previous process is used as a new initial vector. The reason is the increasing storage requirements and the loss of the orthogonality with the higher number of iterations, [118].

The Navier–Stokes system is solved usually with the Krylov subspace methods which require a good preconditioner to achieve fast convergence in practice. The reason is the dependence of the Krylov subspace methods on the spectrum of the system matrix. Because the discretization of Navier–Stokes equations leads to a system of equations with nonsymmetric matrix, iterative methods such as preconditioned GMRES need to be used, [35]. One type of preconditioners are the algebraic preconditioners, based on an ILU factorization, [35, 99]. However, because the absence of the pressure variable in the continuity equation causes zero block in the complete matrix of the system, special attention has to be invested to avoid the breakdown of ILU.

The recent research tends to investigate block preconditioners, as another class of preconditioners, which are suitable for Navier– Stokes problem, [100, 66, 99]. In general, the block preconditioners are based on separation of the velocity and pressure. The convergence depends on the approximation of the Schur complement which arises from the LDU decomposition of the complete system matrix. The combination of block factorization with various approximation of the Schur complement leads to methods such as GCR preconditioner, LSC preconditioner and a suitable approximation of Schur matrix provides the commonly used SIMPLE preconditioner, [35, 99].

To sum up, large system of linear equations must be solved in each iteration of the linearized Navier–Stokes problem, which leads to time–consuming process. Moreover, the continuity equation is not dependent on the pressure variable at all. Thus the so called SIMPLE (Semi–Implicit Method for Pressure–Linked Equations) was developed, originally for FVM [93], to separate problem for velocity and pressure, [66, 94, 116]. A modified variant is e.g. SIMPLER (SIMPLE Revised) or MSIMPLER (Modified SIMPLER), [119], for which the experiments provide better convergence behavior. Finally, PISO (Pressure Implicit with Splitting of Operator) algorithm is an extension of SIMPLE for nonstationary problems, [52, 53, 116].

SIMPLE-like algorithms are iterative. First, the momentum equation is solved as a predictor to compute velocity, where pressure from the previous iteration step is used. However, velocity field does not satisfy the incompressibility condition. The continuity equation is rewritten and the so called pressure correction equation is derived to be used as a correction step. The pressure variable is computed solving the pressure correction equation which leads to the correction of velocity afterwards, i.e. the correction step makes the predicted velocity field satisfying the continuity equation. This two step procedure is repeated until the solution converges. However, the convergence is often not achieved and thus the underrelaxation is applied for velocity and pressure which leads to slow convergence, [66]. The scheme is still popular and is used in many commercial packages. Nevertheless as mentioned above, the SIMPLE-like methods can be used as preconditioners for the Krylov methods, which results in faster convergence, [103, 99].

The work of this thesis is motivated by the problem of automatic shape optimization of runner blades in water turbines in order to improve utility quantities as the turbine efficiency. The flow in the hydraulic turbine is assumed to be very fast and thus the turbulent behavior of the flow is expected. Isogeometric analysis is a very powerful tool for the optimization processes since it gives the possibility to represent the geometry exactly. Moreover, it allows to avoid the time-consuming step of mesh generation and manual treatment of the mesh refinement.

RANS equations closed with the SST turbulence model are solved for the numerical modeling of turbulent incompressible fluid flow. The backward Euler method is applied for the time discretization and the obtained nonlinear discrete problem is linearized using the Picard's method. The resulting linear system of equations is solved by direct method in each iteration.

If the SST turbulence model is employed, the wall distance (needed for the evaluation of the coefficients of the SST model) is computed by solving additional Poisson equation and the wall distance is then approximated using an algebraic expression.

There are not many studies aimed at modeling turbulent incompressible fluid flow using isogeometric analysis. Most of them are focused on the use of LES or VMS techniques, [6]. There are few papers devoted to the isogeometric analysis of the RANS equations supplemented by a two–equation turbulence model, e.g. [92]. More studies devoted to the turbulent compressible fluid flow deal with the combinations of the RANS equations and discontinuous Galerkin isogeometric approximation, [5, 64].

The extension of the classical stabilization approaches for finite elements are investigated employing isogeometric analysis with higher order B-spline basis functions on multipatch domains and irregular curvilinear meshes including local refinement (within the meaning of tensor product refinement).

The numerical difficulties of the advection dominated problems are introduced for a simple advection–diffusion (AD) equation. The SUPG and SOLD stabilization methods are formulated and employed for the AD equation first. Then, the observations are applied to stabilization of the more complicated advection dominated turbulence model. But the study has to be extended since the turbulence model is also reaction dominated. The numerical experiments are presented in two–dimensions, for which the solution of the RANS equations usually provides satisfactory results (cf. e.g. [40]).

The fluid flow simulation is obtained from an in-house isogeometric incompressible flow solver implemented in a framework of the G+Smo (Geometry + Simulation modules) library for both 2D and 3D problems. G+Smo is an open source, object-oriented, templated C++ library.

2 Incompressible fluid dynamics

Liquids, gases and plasmas are the states of matter which are referred as fluids. Fluids do not have their own permanent shape because of their large particle motions, but they deform (flow) under any applied forces. Shape of liquids is dependent on the container in which they are located, compared to the gasses, which fill the entire space of any container.

Fluid flow is concerned by a discipline called fluid dynamics, which describes the properties of the fluid flow, such as velocity, pressure, density and temperature, through empirical laws of conservation. The mentioned fluid flow properties are generally functions of space and time, i.e. consider a fluid moving in a domain $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$ during any time interval $[0, \bar{t}]$, where d is dimension of the domain and $\bar{t} > 0$ is an upper bound of the time interval of interest. Then $\mathbf{u} = \mathbf{u}(\mathbf{x}, t) = (u_1(\mathbf{x}, t), \dots, u_d(\mathbf{x}, t))$, $p = p(\mathbf{x}, t)$, $\rho = \rho(\mathbf{x}, t)$, $T = T(\mathbf{x}, t)$ denotes the velocity, pressure, density and temperature of the fluid flow at every position $\mathbf{x} = (x_1, \dots, x_d) \in \Omega$ and time $t \in (0, \bar{t})$.

The flow velocity $\mathbf{u}$ of a fluid is a vector field, which describes the motion of the fluid in the direction of each axis of the coordinate system. The flow speed is then the length of the flow velocity vector $\|\mathbf{u}\|$.

The density of the fluid ρ is defined as its mass per unit volume. As considered above, the fluid density is generally function of space and time, because the density of any fluid varies with changes in pressure or temperature. In this context, we are talking about compressibility of the fluids, which is a measure of relative volume changes, i.e. changes of density ρ of the fluid. All fluids are to some extent compressible, [116, 20]. Gasses can be easily compressed, however, the compressibility of liquids is so small that the changes in density are negligible at a particular temperature and thus liquids are generally considered as incompressible fluids with constant density. The flow of any incompressible fluid is then considered as the incompressible flow. On the other hand, the flow of the compressible fluids (gasses in our considerations) is not always compressible. If the speed of the fluid is low enough, the density of the fluid remains nearly constant and the flow of the compressible fluid can be also considered as the incompressible flow. But for higher speeds, the density of the fluid locally changes, the flow becomes more complicated and the compressible flow is observed.

The compressibility of a flow is usually related to the value of the so called Mach number, [116]. Mach number is a dimensionless quantity representing the ratio of the flow velocity to the local speed of sound. The flow of a compressible fluid can be approximated as incompressible at low Mach numbers, whereas compressible flow is considered at higher Mach numbers. The limit of the Mach number between the compressible and incompressible flow is not easy to determine, because the limit varies with temperature for example and thus the limit must be determined for every case separately. However, we restrict to the incompressible fluids in the rest of the thesis as our aim is to study the flow of the liquids and thus more details about the compressible flow can be found e.g. in [116, 17].

The so called Bernoulli's principle is valid for incompressible flows. It describes the change of the fluid pressure p in response to the change of the flow speed. Specifically, the fluid pressure decreases when the flow speed increases, which is mathematically expressed by

$$\frac{1}{2}\rho v^2 + h\rho g + p = \text{constant}, \tag{1}$$

where v is the fluid flow speed at a point, h is the elevation of the point of a reference plane and g is the gravity acceleration. In other words, the conservation of energy holds, i.e. the sum of the kinetic energy, potential energy and pressure energy remains constant. The Bernoulli's equation can be modified for the compressible flow. For more details about the Bernoulli's principle, the reader is referred to [116, 47].

Another important property of fluids is the viscosity (specifically, the dynamic viscosity denoted by μ), which represents the resistance of the fluids to deformation caused by the shear stress denoted by τ. The shear stress acts tangentially to the surface and it arises when some particle layers of the fluid are moving at different velocities than another layers of the fluid. An obvious example is the slower movement of the fluid particles near a wall of the domain, compared to the faster fluid flow far from the wall. Thus, the viscosity can be simply expressed as friction between the particles of the fluid and the shear stress is needed to overcome the friction between the particle layers to keep the fluid moving. Informally, we talk about the thickness of the fluid, e.g. honey is thicker than the water, thus honey has higher viscosity than water.

Fluids with nonzero viscosity are called viscous, on the other hand, a fluid with zero viscosity is called inviscid (also known as non-viscous or ideal) fluid that has no resistance to the shear stress. Inviscid fluids are usually known as superfluids. The concept of the superfluids was created at the theoretical level, but this state was relatively recently discovered for helium cooled below a certain temperature.

In fluid dynamics, it is common to work with a dynamic viscosity which is defined as the ratio of the dynamic viscosity ν to the density ρ

$$\nu = \frac{\mu}{\rho}. \tag{2}$$

A fluid is said to be Newtonian fluid if it behaves according to the Newton's law of viscosity, which describes linear relation between the shear stress τ and the strain rate denoted as $\mathbf{e}$ and defined by

$$\mathbf{e} = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right). \tag{3}$$

Strain rate describes the rate of the deformation of the fluid over time, see [20, 90, 116]. Note that the strain rate is usually defined as tensor by

$$e_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{4}$$

Then the Newton's law of viscosity of the incompressible fluid is written in the tensor form as

$$\tau_{ij} = 2\mu e_{ij} \tag{5}$$

and the matrix form

$$\tau = 2\mu\mathbf{e} = \mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right), \tag{6}$$

where viscosity μ is the proportionality constant. An example of the Newtonian fluid is the water with a constant viscosity. The second group of fluids are non-Newtonian fluids, which does not follow the linear Newton's law of viscosity, but the shear stress is dependent on the strain rate non-linearly. Thus, the viscosity of the non-Newtonian fluids is not constant, but non-linear or it is dependent on deformation history or even on time, see [116, 47].

Generally, we assume unsteady flow, i.e., the quantities of the fluid are time dependent, whereas they do not depend on time at any point in the steady flow. In practice, most of the flows are unsteady. But the cyclic behavior can be often observed and the steady flow can be usually assumed after initial unsteady flow development, if the boundary conditions and source term are steady.

Assume a fluid with high viscosity that flows with very low speed. If the trajectories of the fluid particles are well ordered and they move in parallel layers without mixing, a laminar flow is observed. In the fluid dynamics, the laminar flow occurs when the viscous (or friction) forces dominate the inertial forces. The ratio of the inertial and viscous forces is know as the Reynolds number, which is expressed by, see e.g. [35, 10, 17],

$$Re = \frac{UL}{\nu}, \tag{7}$$

where L is a characteristic length scale of the domain Ω and U is the characteristic velocity scale (mean velocity). The characteristic length scale is size of the geometry also called as a characteristic dimension. The characteristic length scale is usually determined as the internal diameter. For example of the flow in the pipe, it is the pitch of the pipe walls and for the flow over an obstacle, L is the width of the obstacle. Certain rules are also applied for more complicated shapes of the geometry.

The Reynolds number is thus a dimensionless quantity of the flow that is used to predict the behavior of the fluid flow. Low Reynolds numbers, where viscous forces are dominant, characterize laminar flow. If the inertial forces acting on the fluid are big enough compared to the viscous forces that dampen the disturbances and instabilities of the flow and the so called critical Reynolds number is exceeded, the laminar flow becomes a turbulent flow, [32, 37], which is the result of the random swirls, the turbulent eddies.

However, no universal limits of the critical Reynolds number exist. For instance, the flow in the pipe is laminar for $Re < 2100$ and it becomes turbulent with the Reynolds number greater than 4000. The critical Reynolds number that characterize the transition between the laminar and turbulent flow thus occurs at $2100 < Re < 4000$ for the pipe flow. Another example is the flow over an obstacle, for which the laminar flow exists up to $Re \approx 20$.

The majority of the flows are turbulent in the nature or in the engineering applications, and the turbulent behavior occurs in all the fluids if the inertial forces dominate over the viscous forces as already mentioned. The turbulence is the property of the flow (not of the fluid) such that the turbulent flow consists of the whole spectrum of the turbulent eddies

[30] whose sizes are known as the turbulent scales. The motion of the fluid particles is unorganized, the trajectories and eddies intersect each other, which leads to chaotic flow with rapid variations of the velocity and pressure.

A number of experiments of the laminar and turbulent flows in simple geometries were illustrated in the literature, see e.g. [30]. It was shown according to the experiments that the averaged velocity distributions of the turbulent flow highly correspond to the laminar steady case. The averaged variables are called mean component of the turbulent flow. Moreover, a chaotic part of motion was found, called fluctuating component. Thus in general, the turbulent flow is considered as the sum of the mean $\bar{\mathbf{u}}(\mathbf{x})$ and fluctuating $\mathbf{u}'(\mathbf{x},t)$ components, i.e.,

$$\mathbf{u}(\mathbf{x},t) = \bar{\mathbf{u}}(\mathbf{x}) + \mathbf{u}'(\mathbf{x},t). \tag{8}$$

This way of decomposition of the flow variables is known as a statistical approach, which is a natural way as the turbulence is unsteady, chaotic (random) and extremely sensitive to the perturbation in the initial data. A statistical measure is an average, which can be understood in two ways. First way is the time–averaging over a sufficiently large time interval $[t_0, t_0+T]$, where $t_0 \geq 0$. The mean quantity is then defined by

$$\bar{\mathbf{U}}(\mathbf{x}) = \lim_{T\to\infty} \frac{1}{T} \int_{t_0}^{t_0+T} \mathbf{U}(\mathbf{x},t)\,\mathrm{d}t, \tag{9}$$

where $\bar{\mathbf{U}} = [\bar{\mathbf{u}}, \bar{p}]^T$ are the mean components and $\mathbf{U}' = [\mathbf{u}', p']^T$ are the fluctuating components. Note that t_0 can be assumed to be equal to zero, because the turbulence repeats itself over a period of time with the invariant boundary conditions. Substituting the decomposition of the velocity and pressure (8) into the definition of the mean component (9) implies that the averaging of the fluctuating component equals to zero, i.e.,

$$\begin{aligned}
\bar{\mathbf{U}}(\mathbf{x}) &= \lim_{T\to\infty} \frac{1}{T} \int_0^T \mathbf{U}(\mathbf{x},t)\,\mathrm{d}t \\
&= \lim_{T\to\infty} \frac{1}{T} \int_0^T \left[\bar{\mathbf{U}}(\mathbf{x}) + \mathbf{U}'(\mathbf{x},t)\right]\,\mathrm{d}t \\
&= \lim_{T\to\infty} \frac{1}{T} \int_0^T \bar{\mathbf{U}}(\mathbf{x})\,\mathrm{d}t + \lim_{T\to\infty} \frac{1}{T} \int_0^T \mathbf{U}'(\mathbf{x},t)\,\mathrm{d}t \\
&= \bar{\mathbf{U}}(\mathbf{x}) + \lim_{T\to\infty} \frac{1}{T} \int_0^T \mathbf{U}'(\mathbf{x},t)\,\mathrm{d}t \\
&= \bar{\mathbf{U}}(\mathbf{x}) + \overline{\mathbf{U}'(\mathbf{x},t)}
\end{aligned}$$

and thus $\overline{\mathbf{U}'(\mathbf{x},t)} = 0$. Note that the mean component is no longer a function of time and it follows that all time derivatives of $\bar{\mathbf{U}}(\mathbf{x})$ are zero. However, if the turbulent flow is unsteady, time–averaging cannot be used and it has to be replaced by ensemble averaging

(where the notation is used identical for simplicity, because the unsteady flow is assumed in the subsequent interpretation)

$$\bar{\mathbf{U}}(\mathbf{x}, t) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \mathbf{U}_n(\mathbf{x}, t). \tag{10}$$

The concept is to imagine a sequence of flow realizations $\mathbf{U}_n(\mathbf{x}, t)$ for which the problem is identical (e.g. boundary conditions, geometry). The realizations create flows that vary considerably from each other. An average over a large set of such realizations is an ensemble average, which based on the ergodic hypothesis, see e.g. [82, 30, 32, 116, 85] for more details.

A basic properties of the fluid flow were mentioned above, however, no clear and accurate definition of the turbulence exists. There is a long list of attempts at a definition by the researchers in turbulence, but some of them are controversial or incomplete. Even the current researchers differ in their description of the turbulence and they follow distinct authors. One of the most accurate definitions can be attributed to Bradshaw [13]: "*Turbulence is a three–dimensional time–dependent motion in which vortex stretching causes velocity fluctuations to spread to all wavelengths between a minimum determined by viscous forces and a maximum determined by the boundary conditions of the flow.*"

Indeed, the largest scales are of the order of the flow geometry and they transfer the kinetic energy k from the mean flow, where the kinetic energy is produced. Because the viscous forces are negligible in the large scales, the kinetic energy increases (which follows from the fluid deformation - rotation and translation - similarly to solids) and the smaller scale is then necessarily generated such that the kinetic energy is transferred from the larger eddies to the smaller ones. The process is continuously repeated until the motion of the smallest eddies (molecular or the so called Kolmogorov scales) is stable, where the viscous forces become larger than the inertial forces and the energy is dissipated into the thermal energy (heat, but it does not play any essential role). The dissipation (amount of energy that is dissipated or the rate of destruction of the turbulent kinetic energy) is denoted by ϵ. The described process is called the energy cascade and is analyzed in details in [13, 32, 30].

Near–wall behavior

Most of the flows in the nature and engineering practice are influenced by an adjacent wall. The velocity of the fluid flow equals zero at the wall, but it is nonzero at any distance from the wall. The free stream velocity is assumed far from the wall and the transitional part between the zero and free stream velocity is known as a boundary layer or near-wall region, where the variation of the flow is usually intensive and hence the gradients and velocity changes are the largest here.

The near-wall region is divided into three areas, the viscous sub–layer closest to the wall, next the buffer layer and the logarithmic layer, [116, 117, 14]. The non–dimensional wall distance y^+ is traditionally used in the boundary layer theory to divide the boundary layers and it is for the wall bounded flows defined in the following way

$$y^+ = \frac{u_\tau y}{\nu}, \tag{11}$$

where u_τ is the friction velocity defined below and y is the distance to the nearest wall. Then the width of the inner layers are usually considered according to, see e.g. [14],

- viscous sub–layer $0 < y^+ < 5$
- buffer layer $5 < y^+ < 30$
- logarithmic layer $30 < y^+ < 200$

In the viscous sub–layer, the velocity profile increases linearly with the wall distance (i.e. $u^+ = y^+$) and the behavior of the fluid is dominated by the viscous effects. Moreover, the advection and pressure gradients are almost negligible. In the buffer layer, the viscous and turbulent effects are both important and the pressure gradient and advection cannot be neglected. Moreover, the production of the turbulent quantities gain maximum value in the buffer layer. Hence, the modelling of the velocity behavior is very difficult in this area. In the last logarithmic layer, the turbulent effects are dominant and the velocity profiles show a logarithmic variation, [90, 116, 20, 123, 121], which is given by the so called law of the wall (or logarithmic law) which estimates the velocity profile as

$$u^+ = \frac{1}{\kappa} \ln \left(y^+\right) + B, \tag{12}$$

where $\kappa \approx 0.41$, $B \approx 5.1$ and u^+ is dimensionless velocity defined by the shear (friction) velocity u_τ, [20, 123], as

$$u^+ = \frac{\bar{u}}{u_\tau}, \quad u_\tau = \sqrt{\frac{\tau_w}{\rho}}, \tag{13}$$

where ρ is density of the fluid, τ_w is the wall shear stress and $\bar{u}$ is the velocity parallel to the wall. The derivation of the logarithmic law can be found e.g. in [65, 14].

The shear stress is caused by movement of the boundary layers in any fluid, where the velocity losses its speed because of the viscous forces dominating close to the wall. Then the shear velocity u_τ represents a rate of deceleration of the free stream velocity far from the boundary towards the boundary. Here the shear velocity also acts as a scaling parameter to get dimensionless law of the wall. The wall shear stress τ_w is defined by

$$\tau_w = \mu \left.\frac{\partial u}{\partial y}\right|_{y=0}. \tag{14}$$

But it can be approximated by

$$\tau_w = \frac{1}{2} C_f \rho U^2, \tag{15}$$

where U is the free stream velocity (usually taken outside the boundary layer or at inlet) and C_f is the skin shear coefficient. In our case, the skin shear coefficient is considered as

$C_f = 0.079 \cdot Re^{-\frac{1}{4}}$ for internal flows and Re is Reynolds number for the given problem, [121].

The fluid flow near the boundary is quite sensitive to the influence of the changes of the geometry or the magnitude of the pressure gradient (e.g. the backward facing step, wing of the aircraft, blade of the turbine, where the flows from the suction and pressure sides encounter), which can lead to the detachment of the flow from the surface and the eddies are formed. Such a behaviour of the flow is called flow separation. Specifically, the separation occurs in the viscous layer, where the velocity is close to zero and the pressure gradient is positive and increases in the direction of the flow, which is said to be the adverse pressure gradient. As the speed of the flow in the viscous layer decelerates, the adverse pressure gradient can affect the layer more significantly. If the pressure gradient is large enough, the velocity becomes zero and then reverses its direction, which leads to the separation of the flow from the wall. The location where the velocity equals zero in the boundary layer is called the point of separation.

The flow separation has importance in the engineering practice, since it can result in increased pressure drag. Hence, the effort of the research of hydrodynamics and aerodynamics is to optimize the shape in order to delay the flow separation and preserve the flow attached as long as possible. For more details and discussion about the flow separation, the reader is referred to [47, 90, 39].

Because of the difficult chaotic (often also termed as random) behaviour of the turbulent flow, special attention has to be invested if we are interested in turbulent fluid flow simulation. Even if the flow is chaotic, the same structures and flow characteristics of the turbulent flow arise if the boundary conditions do not change. As a result, the turbulence can be considered as quasi-deterministic processes and the mean flow of the fluid can be described with the help of the statistical laws. Moreover, the fluid flow is considered as continuum and thus the motion of the fluid is supposed to be described by the Navier–Stokes equations, [20, 30, 13], which is also validated by the results of the numerical simulation. The Navier–Stokes equations are based on the physical principles, which are introduced in the following section 2.1.

2.1 Governing equations

The governing equations are based on the conservation laws of fluid mechanics together with the constitutive relations, [30]. We restrict ourselves to the incompressible and Newtonian fluids, thus the physical principles necessary for the derivation are

- conservation of mass, i.e., continuity equation,
- Newton's second law, i.e., conservation of the momentum,
- Newton's law of viscosity (6), which is the constitutive law.

The instantaneous state of the fluid of the infinitesimal volume δV (fluid element) is considered in the rectangular element (control volume or control element) with edges $\mathrm{d}x$, $\mathrm{d}y$, dz in the three coordinate directions. As the density of the incompressible fluid is constant, the mass of the fluid in the control element is supposed to be constant at the given moment in time. The conservation of mass then states that the rate of change of mass in a control volume equals to the amount of fluid flowing across its boundary of the control volume over time. Which is mathematically expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{\delta V} \rho \,\mathrm{d}V = -\int\limits_{\partial V} \rho \mathbf{u}\cdot\mathbf{n}\,\mathrm{d}\Gamma, \tag{16}$$

where $\mathbf{n}$ is a unit outward normal to the boundary of the control volume ∂V. If we apply the divergence theorem to the right–hand side, we get

$$\frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{\delta V} \rho \,\mathrm{d}V = -\int\limits_{\delta V} \rho \nabla\cdot\mathbf{u}\,\mathrm{d}V. \tag{17}$$

As the density of the incompressible fluid is constant in the space and time, the left-hand side of mass conservation (17) equals zero, i.e.,

$$0 = -\rho\int\limits_{\delta V} \nabla\cdot\mathbf{u}\,\mathrm{d}V, \tag{18}$$

from which follows the continuity equation for the incompressible fluid

$$\nabla\cdot\mathbf{u} \;=\; 0, \tag{19}$$

which is therefore valid whether the flow is changing in time or not. The derivation of the conservation of momentum is more difficult and can be found e.g. in [30, 13] in details. In this work, only reduced steps are provided. The conservation of momentum states that the rate of change of momentum of the fluid in the control volume equals to the sum of forces acting on the fluid. The rate of change of momentum is the product of the mass and acceleration of the fluid

$$\int\limits_{\delta V} \rho\left(\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right)\mathrm{d}V. \tag{20}$$

The forces acting on the incompressible Newtonian viscous fluid are the external forces $\mathbf{f} = \mathbf{f}(\mathbf{x})$ (e.g. magnetic field or gravity), pressure forces (in the direction of the inward normal) and shear forces (stress gradients caused by the deformation of the fluid)

$$\int\limits_{\delta V} \rho\mathbf{f}\,\mathrm{d}V + \int\limits_{\partial V} p(-\mathbf{n})\,\mathrm{d}\Gamma + \int\limits_{\delta V} \mu\Delta\mathbf{u}\,\mathrm{d}V, \tag{21}$$

where $p = p(\mathbf{x}, t)$ is a scalar function describing pressure of the fluid. Note that the pressure term can be modified using the divergence theorem as

$$\int\limits_{\partial V} p(-\mathbf{n})\,\mathrm{d}\Gamma = -\int\limits_{\delta V} \nabla p\,\mathrm{d}V$$

and represents the normal stresses. The last shear term follows from the shear stresses (representing tangential stresses or viscous forces) acting in any direction at every point on the surface ∂V

$$\int\limits_{\partial V} \tau \cdot \mathbf{n} \, \mathrm{d}\Gamma = \int\limits_{\delta V} \nabla \cdot \tau \, \mathrm{d}V,$$

where τ is the shear stress tensor defined by (6). Then the shear forces are given by

$$\begin{aligned} \nabla \cdot \tau &= \nabla \cdot (2\mu\epsilon) \\ &= \mu \nabla \cdot \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \\ &= \mu \left(\nabla \cdot \nabla \mathbf{u} + \nabla \cdot (\nabla \mathbf{u})^T \right) \\ &= \mu \left(\nabla^2 \mathbf{u} + \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)^T \underbrace{(\nabla \cdot \mathbf{u})}_{=0} \right) \\ &= \mu \Delta \mathbf{u}. \end{aligned} \tag{22}$$

Taking into account the previous relations, the conservation of momentum has the form

$$\int\limits_{\delta V} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) \, \mathrm{d}V = \int\limits_{\delta V} \rho \mathbf{f} \, \mathrm{d}V - \int\limits_{\delta V} \nabla p \, \mathrm{d}V + \int\limits_{\delta V} \mu \Delta \mathbf{u} \, \mathrm{d}V. \tag{23}$$

Since δV is any volume, the previous form results in the nonstationary Navier–Stokes equations together with the continuity equation that complete the system of equations for the four variables $\mathbf{u} = (u_1, u_2, u_3)$ and p such that

$$\begin{aligned} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) &= \rho \mathbf{f} - \nabla p + \mu \Delta \mathbf{u} \quad \text{in } \Omega \times (0, T), \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{in } \Omega \times (0, T). \end{aligned} \tag{24}$$

As already mentioned, the equations describe both laminar and turbulent motion. Although the system of equations (24) is a closed system of equations in the case of the laminar flow, the turbulent flow is much more complicated and the velocity and pressure variables in (24) should be understood as instantaneous values of the chaotic (random) non-stationary variables.

3 Mathematical modeling

The mathematical model of the fluid flow is based on the conservation laws (24). The first and natural approach is to formulate the mathematical problem as the system of equations representing the conservation laws (24) together with an appropriate boundary and initial conditions, i.e.,

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= \mathbf{f} - \nabla p + \nu \Delta \mathbf{u} \quad \text{in } \Omega \times (0,T), \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{in } \Omega \times (0,T), \end{aligned} \tag{25}$$

where $\nu = \mu/\rho$ is kinematic viscosity and $p = p(\mathbf{x}, t)$ is kinematic pressure given as dynamic pressure p divided by density ρ of the fluid p/ρ, but again denoted by p for simplicity. The initial boundary value Navier–Stokes problem is given as a system of $d+1$ equations (25) together with initial and mixed boundary conditions

$$\begin{aligned} \mathbf{u}(\mathbf{x}, 0) &= \mathbf{u}_0(\mathbf{x}), && \mathbf{x} \in \Omega, \\ \mathbf{u} &= \mathbf{g}, && \text{in } \partial\Omega_D \times [0,T], \\ \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - \mathbf{n} p &= \mathbf{0}, && \text{in } \partial\Omega_N \times [0,T]. \end{aligned} \tag{26}$$

The boundary condition at the boundary part $\partial\Omega_N$ is known as the “do-nothing" condition. Note that the boundary condition for velocity on any stationary solid surface is zero Dirichlet boundary condition $\mathbf{u} = \mathbf{0}$, which is known as the no-slip condition. A couple $(\mathbf{u}, p)$ is called a classical solution of the nonstationary Navier–Stokes problem with the nonhomogeneous boundary conditions, if

$$\mathbf{u} \in C^2\left(\bar{\Omega} \times [0,T]\right), \quad p \in C^1\left(\bar{\Omega} \times [0,T]\right) \tag{27}$$

satisfying (25) - (26).

Sometimes it is convenient to consider a simplification of the Navier–Stokes equations (e.g. for the computation of the initial condition or simulation of the laminar flow), which is presented in the paragraphs below. First assume stationary case, i.e. the time derivative $\frac{\partial \mathbf{u}}{\partial t}$ is zero. Moreover, if the fluid viscosity is sufficiently large and the velocity small, the diffusion term $\nu\Delta\mathbf{u}$ dominates over the advective term $\mathbf{u} \cdot \nabla \mathbf{u}$ in Navier–Stokes equations, then the advective term can be neglected. This implies simplified linear problem, which together with continuity equation and boundary conditions form Stokes problem, [35],

$$\begin{aligned} -\nu \Delta \mathbf{u} + \nabla p &= \mathbf{f}, \quad \text{in } \Omega, \\ \nabla \cdot \mathbf{u} &= 0, \quad \text{in } \Omega, \\ \mathbf{u} &= \mathbf{g}, \quad \text{on } \partial\Omega_D, \\ \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - \mathbf{n} p &= \mathbf{0}, \quad \text{on } \partial\Omega_N. \end{aligned} \tag{28}$$

Stokes problem describes slow flow of very viscous fluid. For example, the flow of blood in some parts of the body or flow of lava can be simulated by Stokes problem. The classical solution of the Stokes problem are functions $\mathbf{u}$ and p such that

$$\mathbf{u} \in C^2\left(\bar{\Omega}\right), \quad p \in C^1\left(\bar{\Omega}\right), \tag{29}$$

which satisfy (28). The solution of the Stokes problem is often used as initial condition for nonstationary case.

The stationary Stokes problem was assumed to simulate very viscous flow of the fluid such that the advective term was neglected. If the stationary case is still assumed but the viscous term does not dominate over the advective term, the advective term cannot be ignored and the stationary Navier–Stokes problem [35, 37] is considered

$$\begin{aligned} -\nu\Delta\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p &= \mathbf{f}, \quad \text{in } \Omega, \\ \nabla\cdot\mathbf{u} &= 0, \quad \text{in } \Omega, \\ \mathbf{u} &= \mathbf{g}, \quad \text{on } \partial\Omega_D, \\ \nu\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - \mathbf{n}p &= \mathbf{0}, \quad \text{on } \partial\Omega_N. \end{aligned} \tag{30}$$

The classical solution $(\mathbf{u}, p)$ satisfy (29) and (30). Navier–Stokes problem (30) is a system of nonlinear equations which can cause certain difficulties using numerical methods. To avoid the nonlinearity, Navier–Stokes equations can be linearized which leads to the so called Oseen problem

$$\begin{aligned} -\nu\Delta\mathbf{u} + \mathbf{w}\cdot\nabla\mathbf{u} + \nabla p &= \mathbf{f}, \quad \text{in } \Omega, \\ \nabla\cdot\mathbf{u} &= 0, \quad \text{in } \Omega, \\ \mathbf{u} &= \mathbf{g}, \quad \text{on } \partial\Omega_D, \\ \nu\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - \mathbf{n}p &= \mathbf{0}, \quad \text{on } \partial\Omega_N, \end{aligned} \tag{31}$$

where $\mathbf{w}$ is a given function such that $\nabla\cdot\mathbf{w} = 0$ is fulfilled in Ω, [37]. Classical solution must satisfy again (29) and (31).

Neumann condition was assumed on part of the boundary for all the flow problems. However, Dirichlet boundary condition could be considered everywhere on the boundary, e.g. for flow in cavity. In such a case, necessary condition must be satisfied for solvability of the problems, i.e. integrating the continuity equation over Ω and using divergence theorem gives

$$0 = \int\limits_{\Omega} \nabla\cdot\mathbf{u}\,\mathrm{d}\Omega = \int\limits_{\partial\Omega} \mathbf{u}\cdot\mathbf{n}\,\mathrm{d}\Gamma - \int\limits_{\Omega} \mathbf{u}\cdot\underbrace{\nabla 1}_{=0}\,\mathrm{d}\Omega = \int\limits_{\partial\Omega} \mathbf{g}\cdot\mathbf{n}\,\mathrm{d}\Gamma.$$

Then assuming inflow $\partial\Omega^+$ and outflow $\partial\Omega^-$ part of the boundary gives a compatibility condition

$$\int\limits_{\partial\Omega^+} \mathbf{g}\cdot\mathbf{n}\,\mathrm{d}\Gamma + \int\limits_{\partial\Omega^-} \mathbf{g}\cdot\mathbf{n}\,\mathrm{d}\Gamma = 0. \tag{32}$$

It means that the volume of the incompressible fluid flowing in the domain must be equal to the volume of the fluid leaving the domain. In this thesis, flow problems of inflow/outflow type are considered, thus (32) is automatically satisfied assuming the Neumann condition (26) on the outflow part and the mass is conserved. It is the reason why the Neumann condition in (26) is also called natural outflow. Moreover, pressure solution is unique in such a case, otherwise pressure is unique up to a hydrostatic constant assuming Dirichlet condition everywhere on the boundary.

Sometimes it is convenient to imagine a fluid with zero viscosity, the so called ideal or inviscid fluid (discussed already in the Section 2), which has no resistance between the fluid layers, thus there are no shear stresses and the Navier–Stokes equations lose the viscous term and they become the Euler equations

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= \mathbf{f} - \nabla p, \quad \text{in } \Omega \times (0,T), \\ \nabla \cdot \mathbf{u} &= 0, \quad \text{in } \Omega \times (0,T). \end{aligned} \tag{33}$$

Formally, the Euler equations are obtained by setting $\nu = 0$ in (25). However, Navier–Stokes equations simulate viscous flow and the Euler equations simulate inviscid flow which satisfy different boundary conditions as no boundary layer arises with zero viscosity. Thus, there is no friction and the fluid can slide along the boundaries. Then, the so called slip boundary condition is imposed

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad \text{on } \partial\Omega, \tag{34}$$

where $\mathbf{n}$ denotes the outer normal to the boundary. In the past, the Euler equations were mostly solved because of their lower memory requirements, which are much higher for solving Navier–Stokes problem. The reason is that the Euler equations are simpler for numerical simulation as no boundary layers remain with zero viscosity. Certainly, this approach does not give physically satisfactory results in many cases. An example is the flow around the air profile, where the friction losses could not be evaluated from the velocity and pressure fields computed from the Euler equations, however, the lift coefficient was able to determine quite reliably from the pressure distribution on the air profile.

Because of the nonlinear character of the partial differential equations, which produces the turbulent behaviour, no general solution of the Navier–Stokes problem was found so far. More precisely, the existence and uniqueness of the solution of the Navier–Stokes problem in two dimensions was already proved, however, the proof is incomplete for the three-dimensional unsteady case so far. The fundamental question, whether the three-dimensional solution exists on arbitrarily long time interval for any initial condition, remains unresolved. However, the theory about the existence of the solution is not the objective of the thesis, thus the reader is referred to [37].

Currently, the solution of the fluid flow is based on numerical solution of the Navier–Stokes equations. The most common approaches are Direct Numerical Simulation (DNS), Reynolds–Averaged Navier–Stokes equations (RANS) and Large Eddy Simulation (LES). The first clear choice is the direct numerical simulation in which the Navier–Stokes equations (25) are directly numerically solved for the given boundary and initial conditions. However, it means that the whole range of the scales must be resolved till the smallest scales of the flow [116, 84]. Then the number of grid points in spatial discretization has to be proportional to $Re^{9/4}$, i.e. the grid spacing has to be smaller than the smallest turbulent length scale, otherwise the simulation becomes unstable. Moreover, the time step has to be sufficiently small to resolve the movement of the fastest fluctuations. Thus, the computer storage and runtime requirements increase with increasing Reynolds number. Despite the rapid progress of the computers, we are still not able to cover high memory requirements of DNS. Thus, the direct approach is possible to apply only for the low Reynolds number flows in simple geometries and the complex technical applications will not be able to simulate for some time.

An alternative to the refinement of the spatial and temporal discretization for fluid flow simulation with higher Reynolds number are stabilization methods. The advection term dominates over the diffusion term in the Navier–Stokes equations with increasing Reynolds number. Thus, stabilization techniques were developed to improve the stability of the advection dominated problems. Nevertheless, they are usually not sufficient in engineering problems if the Reynolds number is too high. The problem of stabilization is discussed later in the thesis in Section 4.3. On the other hand, if the employed stabilization is sufficient, it is not clear how to interpret the resulting numerical solution. Some of the researchers follow the concept of the ILES/MILES methods based on the analogy of the numerical dissipation resulting from the numerical scheme and SGS model in the LES technique, [43, 80].

Nevertheless, for many engineering applications and for the comparison of the simulations with the real measurements, the smallest scales and fluctuations are not the point of interest. Thus in the case of a fully developed turbulent flow, certain simplifications can be involved. The Reynolds-Averaged Navier-Stokes approach (so called statistical approach) is based on the Reynolds decomposition (8) and solving of the equations of the fluid motion for the mean variables. RANS equations have become a basic tool for the numerical solution of the turbulent flow. Also, this thesis is devoted to the turbulent flow simulated by the RANS equations, which are introduced in the following section in more details.

The third mathematical model for turbulence is the LES approach, [10, 116, 123], which can be classified between DNS and RANS methods considering the memory requirements. The LES equations are derived from the Navier–Stokes equations such that only the motion of the larger eddies is simulated whereas the rest of the small eddies is filtered and their motion is modeled by the so called subgrid model. Note that the small eddies are filtered according to the size of the grid cells, i.e. those that are smaller than the width of the grid cell (width of the filter).

3.1 Reynolds–Averaged Navier–Stokes equations

Despite the development of the numerical methods for direct numerical simulation and especially simulation of the large eddies, Reynolds–Averaged Navier–Stokes approach based on the averaging of the Navier–Stokes equations is the most common method to describe turbulent flow so far. The reason is the lower memory requirements as the whole range of the scales is modeled and only the effect of the turbulence on the mean flow behaviour is considered, [32, 82, 116, 123, 20].

As already mentioned in the previous section, RANS equations are time–averaged (or ensemble averaged, see above) equations of motion of the fluid. The idea is to separate the flow variables into the mean component and the fluctuating component, which is the Reynolds decomposition. Thus, the velocity and pressure solution $\mathbf{U}(\mathbf{x}, t) = [\mathbf{u}(\mathbf{x}, t), p(\mathbf{x}, t)]^T$ of the nonstationary Navier–Stokes problem is expressed as

$$\mathbf{U}(\mathbf{x}, t) = \bar{\mathbf{U}}(\mathbf{x}) + \mathbf{U}'(\mathbf{x}, t), \tag{35}$$

where $\bar{\mathbf{U}} = [\bar{\mathbf{u}}, \bar{p}]^T$ are the mean components and $\mathbf{U}' = [\mathbf{u}', p']^T$ are the fluctuating components. The mean component was already defined in Section 2 by the time–averaging (9) in the case of flow whose mean values are stationary or by the ensemble averaging (10) in the case of nonstationary mean flow. As we are interested in the time dependent process, the ensemble averaging is considered.

Note that the continuity equation was used during the derivation of the momentum equation and the diffusion term $\nabla \cdot \tau$ representing shear forces was rewritten as $\mu\Delta\mathbf{u}$. However, it is necessary to derive the RANS equations with the whole shear stress, i.e. let us consider the Navier–Stokes equations in the form

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= -\nabla p + \frac{1}{\rho} \nabla \cdot \tau, \\ \nabla \cdot \mathbf{u} &= 0. \end{aligned} \tag{36}$$

Now we substitute the decomposition (35) into the nonstationary Navier–Stokes equations (36) and after the ensemble averaging, it yields

$$\begin{aligned} \frac{\partial \bar{\mathbf{u}}}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} &= -\nabla \bar{p} + \frac{1}{\rho} \nabla \cdot \tau - \overline{\mathbf{u}' \cdot \nabla \mathbf{u}'}, \\ \nabla \cdot \bar{\mathbf{u}} &= 0. \end{aligned} \tag{37}$$

Detailed derivation can be found in [30, 116, 32, 39, 82]. As already mentioned, the turbulent fluctuations $\mathbf{u}'$, p' are not resolved anymore, but the attention is focused only on the mean flow $\bar{\mathbf{u}}$, $\bar{p}$ and the effects of the turbulence on the mean flow. Compared to the system of Navier–Stokes equations with the continuity equation, the averaged Navier–Stokes equations (37) includes new unknown term

$$\overline{\mathbf{u}' \cdot \nabla \mathbf{u}'}, \tag{38}$$

which is writen in the tensor notation as

$$\overline{\mathbf{u}' \cdot \nabla u_i'} = \nabla \cdot \overline{\mathbf{u}' u_i'} = \frac{\partial}{\partial x_j} \overline{u_j' u_i'}$$

applying the continuity equation, where

$$-\overline{u_i' u_j'} \tag{39}$$

are the so called Reynolds stresses acting on the fluid. Then, the averaged Navier–Stokes equations can be rewritten in the tensor notation as

$$\begin{aligned} \frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j}(\bar{u}_i \bar{u}_j) &= -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\frac{1}{\rho}\tau_{ij} - \overline{u_i' u_j'}\right), \\ \frac{\partial \bar{u}_i}{\partial x_i} &= 0. \end{aligned} \tag{40}$$

Thus, we can observe that the momentum of the fluid depends not only on the shear stress, but also on the Reynolds stresses, which represent the transfer of the momentum influenced by the turbulent fluctuations, for more details see e.g. [30].

The Reynolds stress tensor is symmetric, i.e., 3 terms ($\overline{u_1' u_1'}$, $\overline{u_2' u_1'}$ and $\overline{u_2' u_2'}$) in 2D and six terms in 3D are unknown. Thus, the fundamental problem of the solution of the averaged Navier–Stokes equations is to close the momentum equations. In this context, we talk about models of turbulence closing the system. One approach is to express the equations for the Reynolds stresses, which results in the time–consuming solution of three equations in 2D or six equations in 3D in addition. The derivation of the Reynolds stress equations follows from the Navier–Stokes equations and can be found e.g. in [32, 20]. However, the exact Reynolds stress equations cannot be directly used to close the averaged Navier–Stokes equations (40), because they includes another unknown terms (dependent on the velocity and pressure fluctuations). Thus, the unknown terms must be approximated if the transport Reynolds stress equations are used to close the RANS system.

Nevertheless, the second group of the turbulence models is still the most commonly applied. The idea of these models is based on the assumption that the Reynolds stress tensor is proportional to the strain rate tensor, S_{ij}

$$S_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right). \tag{41}$$

Then, the so called Boussinesq hypothesis is applied

$$-\overline{u_i' \cdot u_j'} = 2\nu_T S_{ij} - \frac{2}{3}\delta_{ij}k, \tag{42}$$

i.e.

$$-\overline{u_i' \cdot u_j'} = \nu_T \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}k, \tag{43}$$

where ν_T is the proportionality constant called eddy viscosity or turbulent viscosity, δ_{ij} is Kronecker delta and k is turbulent kinetic energy (see more e.g. in [32, 39, 40, 90, 116]). Note that the strain rate tensor S_{ij} defined by (41) corresponds to the definition (4) of the tensor e_{ij}. However, the strain rate of the mean flow is commonly denoted by S_{ij} in the literature. The unknown term $\overline{\mathbf{u}' \cdot \nabla \mathbf{u}'}$ is then expressed as

$$-\overline{\mathbf{u}' \cdot \nabla \mathbf{u}'} = \nabla \cdot \left(\nu_T \left(\nabla \bar{\mathbf{u}} + (\nabla \bar{\mathbf{u}})^T\right) - \frac{2}{3}k\mathbf{I}\right), \tag{44}$$

where $\mathbf{I}$ is an identity matrix. The Boussinesq hypothesis was the first approach that reduced the modeling process from searching the six Reynolds stress terms needed to determine the eddy viscosity and turbulent kinetic energy. The Boussinesq eddy viscosity concept was developed as an analogy to the viscous stresses acting in the laminar flows, i.e. the molecular viscosity creates the momentum transfer caused by molecular diffusion, then the eddy viscosity acts as a quantity modeling the transfer of momentum caused by turbulent eddies. Thus, the eddy viscosity represents the fluid resistence in the turbulent flows and then it varies significantly from one point to another in the flow. [40, 20]

Applying the Boussinesq hypothesis into the equations (40), the final form of the Reynolds–Averaged Navier–Stokes equations is written

$$
\begin{aligned}
\frac{\partial \bar{\mathbf{u}}}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} &= -\frac{2}{3}\nabla k - \nabla \bar{p} + \nabla \cdot \left[(\nu + \nu_T)(\nabla \bar{\mathbf{u}} + \nabla \bar{\mathbf{u}}^T) \right], \\
\nabla \cdot \bar{\mathbf{u}} &= 0.
\end{aligned}
\tag{45}
$$

Boussinesq approach provides simplification which allows to simulate the effects of the turbulence to the mean flow with relatively low memory requirements. Let us notice also the weakness of the Boussinesq assumption. The Reynolds stresses are not generally co–linear with the mean velocity gradients and thus the Boussinesq assumption is not valid in general, such as in case of flows with strong curvature, rapidly accelerated or decelerated. In spite of this weakness, the eddy viscosity models are commonly used since more accurate approaches are still too time consuming. But one should be aware and take it into consideration while deciding on an appropriate turbulence model for a given problem.

Note that the Boussinesq hypothesis does not close the averaged equations (45), but the turbulent viscosity and turbulent kinetic energy are needed to compute to close the system. A wide range of the approaches have been developed to approximate the unknown quantities ν_T and k, which lead to different turbulence models, also known as eddy viscosity models. The turbulence models vary from relatively simple algebraic models to more complex models, e.g. the one equation models and two equation models. However, the eddy viscosity models are only linear models, since the Boussinesq hypothesis assumes the co–linearity of the Reynolds stresses and the strain rate. A brief introduction is shown in the following sections, but the two equation models are discussed in more details, because they are used in the numerical experiments in the thesis.

3.2 Eddy viscosity turbulence models

The following is a brief overview of commonly employed models in modern engineering applications. The turbulence model categories consist of the number of additional equations solved to compute the eddy viscosity and to close the RANS equations.

3.2.1 Algebraic models

The simplest turbulence models, also referred as zero–equation models, are models that do not require the solution of any additional equations, [123, 116, 20]. The eddy viscosity is calculated directly from the averaged flow variables.

The algebraic models are based on an additional Prandtl's mixing length concept, such that in the simplest wall–bounded flow problems, the eddy viscosity is given by

$$\nu_T = l_m^2 \left| \frac{\partial \bar{\mathbf{u}}}{\partial y} \right| , \tag{46}$$

where y is the wall normal direction and l_m is the so called mixing length. The mixing length is referred as a distance that the turbulent vortex travels across the boundary layer before the vortex vanishes by mixing.

The mixing length is usually difficult to determine for individual problem. In case of the free shear flows, i.e., the flow is not bounded by any solid surface, the mixing length is constant across a layer and proportional to the width of the layer. However, it is not valid for the flows near a solid surface. The mixing length is consistent with the law of the wall (12), but the law of the wall has a good estimate only in the logarithmic layer. Then, the mixing length varies in the viscous sublayer or in the outer layer. The Baldwin-Lomax model and Cebeci-Smith model provide modification such that the mixing length is specified over the entire layers, read more details e.g. in [20, 17, 85].

On account of the mentioned simplicity of the model, algebraic turbulence models yield reasonable results for flows such as the fully developed pipe and channel flow or boundary layer flow with reasonable pressure gradients. The algebraic models can also be used for simulation in domains with simple geometries or in initial phase of simulation where no difficulties are expected in the flow and where more complicated turbulence models would take more computational time. However, the algebraic models are applicable only for planar problems and the flow should not include separated regions where the algebraic models are unable to take the flow history effects into account. Moreover, algebraic model requires fine grid near the walls (the first grid point from the wall should be located for $y^+ \leq 1$).

Several modifications have been proposed to improve the weakness of the models, e.g. the Johnson–King model improves the algebraic models in case of separated flows by solving an extra ordinary differential equation (sometimes called a half–equation model). For more details and discussion of the algebraic turbulence models the reader is referred e.g. to [116].

3.2.2 One–equation models

As the engineering applications required higher accuracy with increasing capacity of the computers, the algebraic models lost their popularity and thus the one equation models were developed to improve the turbulent flow predictions by solving one additional transport partial differential equation such that the latest one equation models are still widely applied.

Almost all of the one–equation models are based on solving the transport equation for the turbulent kinetic energy k, from which the eddy viscosity is approximated as a square root of the kinetic energy times a length scale l

$$\nu_T = l\sqrt{k}. \tag{47}$$

The drawback of the one equation models are the empirical relations needed to approximate the turbulent length scale l. The turbulent length scale is a quantity describing the size of the large eddies in the turbulent flow and thus it was usually intuitively estimated as a size smaller than the dimension of the geometry.

However, new turbulence models were developed such that the turbulent length scale may not be explicitly derived, e.g. Baldwin-Barth model or Spalart-Allmaras model. The Spalart–Allmaras model has become very popular in engineering practice, because it provides good results even in the boundary layers. Although the Spalart–Allmaras model is still pretty dependent on the empirical relations, it is numerically stable with reasonable accuracy.

A drawback of the Spalart–Allmaras model can be the requirement to calculate the wall distance, primarily in the 3D geometry. Usually, an additional differential equation is solved to approximate the distance to the nearest wall rather than search algorithms, which can be difficult to apply in complex geometries. The three well known differential equations used to compute wall distance are the Eikonal, Hamilton–Jacobi and Poisson equations. More details about wall distance theory is discussed in the following section or can be found e.g. in [113]. The Spalart–Allmaras model was developed for flow of the compressible fluid around the profile of the wing of the aircraft, thus it is not an appropriate model for our purposes of the incompressible fluid flow in the bounded domains.

3.2.3 Two–equation models

The two–equation models are an extension of the one–equation models, where the algebraic computation of the turbulent length scale dependent on the wall distance is a drawback. Most of the two equation models solve a transport equation for the turbulent kinetic energy k and the purpose of the second transport equation is to replace the calculation of the turbulent length scale l by another turbulent quantity, usually the dissipation ϵ, turbulent specific dissipation ω or some equivalent quantity, which is related to the length scale. The two equation models became standard models in engineering practice and research, thus a large number of two equation models have been derived and development of new models or improvements of the current models are still the subjects of extensive research.

The variety of the models give us an opportunity to choose the most appropriate model for a wide range of the flows. However, the fluid flow behavior is necessary to predict properly and thus to understand the formulation of the two equation models and their assumptions. For example, some turbulence models are valid in the boundary layer (low Reynolds number (LRN) models) and other are valid in the outer regions (high Reynolds number (HRN) models) etc. Thus, it is essential to predict if the separation region can appear in the flow, if the flow near the walls is important to simulate in details or other characteristics of the flow.

Since the averaged flow is simulated solving RANS equations, the basic assumption is that the turbulence scales are proportional to the mean flow scales (i.e. the Boussinesq approximation (43), where the eddy viscosity ν_T is the proportionality constant between the Reynolds stresses and the mean strain rate), which is also important for derivation of the turbulence models. The validity of the Boussinesq assumption was briefly discussed in Section 3.1, from which it follows that the turbulence models have difficulties to predict rapidly accelerated or decelerated flows, like fast rotating flows, stagnation flows or flows significantly affected by the curvature of the geometry. Therefore, the assumptions should not be strongly violated. Despite the assumptions for the Boussinesq hypothesis, other assumptions are also considered for two–equation models, which are not mentioned here, but these assumptions are described in detail in [116, 20]. The most popular k–ϵ, k–ω and SST models are introduced in the following.

$k - \epsilon$ turbulence model

The motivation to derive the two–equation models is to improve the mixing length model and thus to eliminate the algebraic prescription of the turbulent length scale, as already mentioned. The k–ϵ model was the first turbulence model which closed the RANS equations by solving two additional transport equations for the turbulent kinetic energy k and turbulent dissipation ϵ.

Obviously, the first variable k describes the energy of the eddies in the turbulent flow and it is quantified by average of the normal Reynolds stresses (i.e. trace of the Reynolds stress tensor)

$$k = \frac{1}{2}\overline{u_i' u_i'}, \tag{48}$$

which is in 3D

$$k = \frac{1}{2}\left(\overline{u_1' u_1'} + \overline{u_2' u_2'} + \overline{u_3' u_3'}\right).$$

As already mentioned in Section 2, the turbulent kinetic energy is produced from the mean flow, it is transferred through the energy cascade and dissipated in the smallest eddies. The exact equation for turbulent kinetic energy corresponding to the described process is derived from the equations for the Reynolds stresses assuming $i = j$ to satisfy (48). Then, the exact equation for k, which is derived in detail in [20, 32], is expressed as

$$\frac{\partial k}{\partial t} + \frac{\partial}{\partial x_j}(\bar{u}_j k) = -\overline{u_i' u_j'}\frac{\partial \bar{u}_i}{\partial x_j} - \frac{\partial}{\partial x_j}\left(\frac{1}{2}\overline{u_i' u_i' u_j'} + \delta_{ij}\overline{p' u_i'}\right) + \frac{\partial}{\partial x_j}\left(\nu \frac{\partial k}{\partial x_j}\right) - \nu \overline{\frac{\partial u_i'}{\partial x_j}\frac{\partial u_i'}{\partial x_j}}, \tag{49}$$

where the left hand side represents the change in time and advection as the transport of the turbulent kinetic energy through the mean flow. The terms on the right hand side represents:

- production P_k, i.e., amount of the turbulent kinetic energy generated by the mean velocity gradients (interaction with the mean flow)

$$P_k = -\overline{u_i'u_j'}\frac{\partial \bar{u}_i}{\partial x_j}, \tag{50}$$

- turbulent diffusion caused by the motion of the turbulence energy influenced by the velocity and pressure fluctuations

$$-\frac{\partial}{\partial x_j}\left(\frac{1}{2}\overline{u_i'u_i'u_j'} + \delta_{ij}\overline{p'u_i'}\right), \tag{51}$$

- viscous diffusion caused by the molecular motion of the turbulent kinetic energy

$$\frac{\partial}{\partial x_j}\left(\nu\frac{\partial k}{\partial x_j}\right), \tag{52}$$

  which affects only the motion in the vicinity of the walls.

- dissipation ϵ of the turbulent kinetic energy caused by the viscous forces, i.e., the mean rate of transfer of the kinetic energy of the smallest eddies into the thermal energy

$$\epsilon = \nu\overline{\frac{\partial u_i'}{\partial x_j}\frac{\partial u_i'}{\partial x_j}}. \tag{53}$$

  Larger amount of the energy transferred into the thermal energy implies the larger velocity gradients.

However, several terms on the right hand side of k equation (49) are unknown. These terms need to be approximated, which is based on the physical or experimental knowledge. First, note that the production term includes the Reynolds stresses. According to the Boussinesq hypothesis, the production term of the turbulent kinetic energy of the incompressible fluid is given by the relation

$$P_k = -\overline{u_i'u_j'}\frac{\partial \bar{u}_i}{\partial x_j} = \nu_T\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)\frac{\partial \bar{u}_i}{\partial x_j} - \frac{2}{3}\delta_{ij}k\frac{\partial \bar{u}_i}{\partial x_j}. \tag{54}$$

The incompressible fluid is considered in this thesis, for which holds the continuity equation and $\delta_{ij} = 0$ for $i \neq j$, then the relation for the production term can be simplified

$$P_k = \nu_T\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)\frac{\partial \bar{u}_i}{\partial x_j}. \tag{55}$$

The assumptions for the approximation of the turbulent diffusion term are more complicated and can be found e.g. in [20, 116]. But the approximation is based on the gradient diffusion transport mechanism, according to which the diffusion increases with the increasing gradient of the quantity. Moreover, the analogy to the viscous diffusion term is considered, which leads to the modeled relation for the turbulent diffusion

$$D_k = \frac{\partial}{\partial x_j}\left(\frac{\nu_T}{\sigma_k}\frac{\partial k}{\partial x_j}\right), \tag{56}$$

where σ_k is a model constant called Prandtl–Schmidt number. The last dissipation term is approximated by an algebraic relation, if the one–equation turbulence model is used. However, if the two–equation turbulence model is chosen, the dissipation ϵ of the kinetic energy is computed by solving an additional transport equation. In the case of k–ϵ turbulence models, the second transport equation is derived just for the dissipation ϵ. Then, the modeled k-equation can be written in the matrix notation as

$$\frac{\partial k}{\partial t} + \bar{\mathbf{u}} \cdot \nabla k = P + \nabla \cdot \left[\left(\nu + \frac{\nu_T}{\sigma_k}\right)\nabla k\right] - \epsilon. \tag{57}$$

The exact transport equation for the dissipation ϵ can be derived similarly to the turbulent kinetic energy from the Navier–Stokes equations, which is described e.g. in [121]. However, this ϵ-equation is far more complicated than the exact equation for the turbulent kinetic energy and involves multiple correlations of the fluctuating quantities that are very difficult or even impossible to measure and approximate. Therefore, a modeled transport equation for the dissipation ϵ was proposed based on physical reasoning and similarity to the k equation and is written as

$$\frac{\partial \epsilon}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \epsilon = C_{\epsilon 1}\frac{\epsilon}{k}P_k + \nabla \cdot \left[\left(\nu + \frac{\nu_T}{\sigma_\epsilon}\right)\nabla \epsilon\right] - C_{\epsilon 2}\frac{\epsilon^2}{k}, \tag{58}$$

where P_k is the production of the turbulent energy k given by (55). The system of k–ϵ equations (57)-(58) is known as the standard k–ϵ turbulence model with the model constants

$$\sigma_k = 1.0, \quad \sigma_\epsilon = 1.3, \quad C_{\epsilon 1} = 1.44, \quad C_{\epsilon 2} = 1.92. \tag{59}$$

The turbulent viscosity ν_T needs to be expressed to close the RANS equations using the Boussinesq approximation. In the case of one–equation turbulence model, the turbulent viscosity was determined by the relation (47), i.e., $\nu_T = k^{1/2} l$ with the turbulent length scale l. As already mentioned, the one–equation turbulence models are usually solved for the turbulent kinetic energy k such that the same transport equation (57) is solved as in the case of two–equation turbulence models. Thus, the dissipation ϵ is approximated by an algebraic relation to close the k equation. Usually, it is given by

$$\epsilon = C_\mu \frac{k^{\frac{3}{2}}}{l}. \tag{60}$$

This relation is used in two–equation turbulence model to approximate the eddy viscosity as

$$\nu_T = k^{\frac{1}{2}} l = C_\mu \frac{k^2}{\epsilon}, \tag{61}$$

where $C_\mu = 0.09$. It should be noted that the model constants in the k–ϵ turbulence model (59) and C_μ were determined based on a simple flow cases, empirical procedure or by the numerical optimization. The numerical optimization approach means that the constant is changing until the turbulence model obtains reasonable agreement with an experiment. Also, certain assumptions are necessary to include, e.g., validity of the law of the wall, where the advection and viscous diffusion can be neglected or the equilibrium of the turbulent energy (the production of the turbulent kinetic energy equals dissipation rate). More details about the determination of the model constants can be found in [20, 32].

Note that various two–equation models were developed and thus various coefficients were determined depending on which application was considered for the given turbulence model. Thus, a usual source of numerical difficulties is the application of the turbulence model on another type of the turbulent flow than the model was derived.

The mentioned standard $k - \epsilon$ turbulence model is applicable only in the sufficient distance from the wall. The modification of the standard model has to be included or the so called wall functions have to be used, when applied to a wall bounded flow. The wall function approach is based on the law of the wall introduced in Section 2. The idea is to apply the near wall behaviour of the turbulent variables instead of resolve it. The first computational cell is then placed outside the walls in the logarithmic layer, see [116, 31, 14].

Also, it has been shown that the $k - \epsilon$ model is useful for flows with relatively small pressure gradients. In the opposite case, the behaviour of turbulent dissipation ϵ causes certain problems near the boundary of the domain. The discussion about the validity of the standard $k - \epsilon$ turbulence model can be found in [20, 32, 121].

$k - \omega$ turbulence model

Numerical problems appear by solving k–ϵ turbulence model (57)-(58) in the boundary layer. Thus, the Wilcox's k–ω model 1988 was proposed [121], where the so called *specific dissipation* ω is used to approximate the length scale l instead of the dissipation ϵ. The specific dissipation represents the rate at which the dissipation occurs and is expressed as

$$\omega = \frac{\epsilon}{k}. \tag{62}$$

The original k–ω turbulence model is given by

$$\begin{aligned} \frac{\partial k}{\partial t} + \bar{\mathbf{u}} \cdot \nabla k &= P_k + \nabla \cdot \left[\left(\nu + \sigma^* \nu_T \right) \nabla k \right] - \beta^* k \omega, \\ \frac{\partial \omega}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \omega &= \alpha \frac{\omega}{k} P_k + \nabla \cdot \left[\left(\nu + \sigma \nu_T \right) \nabla \omega \right] - \beta \omega^2, \end{aligned} \tag{63}$$

where the production of the turbulent energy P_k is defined by the same relation (55) as for the k–ϵ model. The constants were estimated similarly to the k–ϵ model and they are usually determined as $\sigma^* = 0.5$, $\sigma = 0.5$, $\beta^* = 0.09$, $\alpha = \frac{5}{9}$ and $\beta = \frac{3}{40}$, see [121], and the eddy viscosity is given by the relation

$$\nu_T = \frac{k}{\omega}. \tag{64}$$

Note that the source term in the ω equation includes the fraction with the turbulent kinetic energy in the denominator. However, the source term can be modified as

$$\begin{aligned}
\alpha \frac{\omega}{k} P_k &= \alpha \frac{\omega}{k} \nu_T \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \frac{\partial \bar{u}_i}{\partial x_j} \\
&= \alpha \frac{\omega}{k} \frac{k}{\omega} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \frac{\partial \bar{u}_i}{\partial x_j} = \\
&= \alpha \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \frac{\partial \bar{u}_i}{\partial x_j}.
\end{aligned} \tag{65}$$

Generally, the $k - \omega$ model provides better predictions contrarily to the $k - \epsilon$ turbulence model for the wall bounded flows with adverse pressure gradient, because the k–ω model behaves as the law of the wall in the boundary layer. Then, there is no need to apply the wall functions. But the model is sensitive to the boundary conditions for ω which can negatively influence the flow behavior in the outer layer.

Several modifications of the k–ω turbulence model were developed to improve the predictions of the turbulent quantities and their behavior in the outer layer, [123, 14]. The Wilcox k–ω variant (specifically derived for the computations in the whole domain up to the walls) is used for the numerical computations in this thesis and is written as

$$\begin{aligned}
\frac{\partial k}{\partial t} + \bar{\mathbf{u}} \cdot \nabla k &= P_k - \beta^* \omega k + \nabla \cdot \left[\left(\nu + \sigma_k \alpha^* \frac{k}{\omega} \right) \nabla k \right], \\
\frac{\partial \omega}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \omega &= \gamma \frac{\omega}{k} P_k - \beta \omega^2 + \nabla \cdot \left[\left(\nu + \sigma_\omega \alpha^* \frac{k}{\omega} \right) \nabla \omega \right] + \frac{\sigma_d}{\omega} \nabla k \cdot \nabla \omega,
\end{aligned} \tag{66}$$

where the source term P_k is again defined by (55), but the eddy viscosity is expressed for this k–ω variant as

$$\nu_T = \alpha^* \frac{k}{\hat{\omega}}, \tag{67}$$

where

$$\hat{\omega} = \max \left(\omega, C_{lim} \sqrt{\alpha^* \frac{2 S_{ij} S_{ij}}{\beta_0^*}} \right). \tag{68}$$

The source term in the ω-equation can be modified similarly to (65), but with the eddy viscosity (67)

$$\gamma \frac{\omega}{k} P = 2\gamma \frac{\omega}{k} \nu_T S_{ij} \frac{\partial \bar{u}_i}{\partial x_j} = 2\gamma \alpha^* \frac{\omega}{k} \frac{k}{\hat{\omega}} S_{ij} \frac{\partial \bar{u}_i}{\partial x_j} = 2\gamma \alpha^* S_{ij} \frac{\omega}{\hat{\omega}} \frac{\partial \bar{u}_i}{\partial x_j} \tag{69}$$

Thus, the fraction $\frac{\omega}{k}$ completely vanishes only if the maximum value of the relation (68) equals to the specific turbulent dissipation ω. The coefficients needed for the LRN k–ω equation (66), eddy viscosity (67) and $\hat{\omega}$ (68) are given by

$$\alpha^* = \frac{\alpha_0^* + \frac{Re_T}{R_k}}{1 + \frac{Re_T}{R_k}} \qquad \beta^* = \beta_0^* \left(\frac{\frac{100}{27}\beta_0^* + \left(\frac{Re_T}{R_\beta}\right)^4}{1 + \left(\frac{Re_T}{R_\beta}\right)^4} \right) \qquad \gamma = \frac{13}{25}\frac{1}{\alpha^*}\left(\frac{\alpha_0 + \frac{Re_T}{R_\omega}}{1 + \frac{Re_T}{R_\omega}} \right)$$

$$Re_T = \frac{1}{\nu}\frac{k}{\omega} \qquad \beta = \beta_0 f_\beta \qquad f_\beta = \frac{1 + 85\chi_\omega}{1 + 100\chi_\omega} \qquad \chi_\omega = \left| \frac{\Omega_{ij}\Omega_{jk}S_{ki}}{(\beta^*\omega)^3} \right|$$

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} - \frac{\partial \bar{u}_j}{\partial x_i} \right) \qquad \sigma_d = \begin{cases} 0, & \text{for } (\nabla k \cdot \nabla \omega \leq 0) \\ \frac{1}{8}, & \text{for } (\nabla k \cdot \nabla \omega > 0) \end{cases}$$

The strain rate S_{ij} defined by (41) (whose magnitude is the largest near the wall and it is close to zero in the core of the vortex), Ω_{ij} is the vorticity (whose magnitude is the largest in the core of the vortex or in the boundary layer) and the remaining constants are determined for the LRN case as

$$\sigma_k = 0.6 \qquad \sigma_\omega = 0.5 \qquad C_{lim} = \frac{7}{8}$$

$$R_\beta = 8 \qquad R_k = 6 \qquad R_\omega = 2.61$$

$$\alpha_0 = \frac{1}{9} \qquad \alpha_0^* = \frac{1}{3}\beta_0 \qquad \beta_0 = 0.0708 \qquad \beta_0^* = 0.09$$

Although the two–equation models perform well for a wide range of the flows in engineering practice, these models should be used with caution in case of coarser grids and their application is still limited, e.g., flows with sudden changes or separation. The advantages and disadvantages of the $k - \epsilon$ and $k - \omega$ models are discussed for example in [20, 14] in details.

According to the mentioned properties of $k - \omega$ and $k - \epsilon$ turbulence models, we can summarize that the $k - \epsilon$ model usually causes problems near the boundary and gives reasonable behaviour in the outer layer. On the other hand, $k - \omega$ model is usable through all the layers near the walls. Thus, the shear stress transport (SST) model was developed to combine the mentioned two equation models, which tends to be satisfactory. The near wall layers are modeled by $k - \omega$ and it switches to $k - \epsilon$ turbulence model in the outer region to avoid sensitivity problems of $k - \omega$ model, see e.g. [90, 123].

The wall distance y (i.e. the normal distance to the nearest wall) of a given point is used to decide when the SST model switches to $k - \omega$ model near the walls or to $k - \epsilon$ model far from the walls. Using the distance y, it is possible to formulate the SST turbulence model as a system of two equations again, i.e.

$$\begin{aligned} \tfrac{\partial k}{\partial t} + \bar{\mathbf{u}} \cdot \nabla k &= P_k + \nabla \cdot \left[\left(\sigma_k \nu_T + \nu \right) \nabla k \right] - \beta^* k \omega, \\ \tfrac{\partial \omega}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \omega &= \alpha S^2 + \nabla \cdot \left[(\sigma_\omega \nu_T + \nu) \nabla \omega \right] - \beta \omega^2 + 2\left(1 - F_1\right) \sigma_{\omega 2} \tfrac{1}{\omega} \nabla k \cdot \nabla \omega, \end{aligned} \tag{70}$$

where

$$\begin{aligned} F_1 &= \tanh \left(\left[\min \left[\max \left(\tfrac{\sqrt{k}}{\beta^* \omega y}, \tfrac{500\nu}{y^2 \omega} \right), \tfrac{4\sigma_{\omega 2} k}{CD_{k\omega} y^2} \right] \right]^4 \right), \\ CD_{k\omega} &= \max \left(2\rho \sigma_{\omega 2} \tfrac{1}{\omega} \nabla k \cdot \nabla \omega, 10^{-10} \right), \\ P_k &= \min \left(\nu_T f, 10 \beta^* k \omega \right), \end{aligned} \tag{71}$$

the constant ν is the given kinematic viscosity, $\beta^* = \frac{9}{100}$, $\sigma_{\omega 2} = 0.856$. The values of the remaining parameters σ_k, σ_ω, α and β are dependent on the wall distance y. Let ϕ_1 and ϕ_2 be two given parameters. Then define a parameter ϕ, whose value depend on the wall distance y, such that it varies between the given parameters ϕ_1, ϕ_2 as

$$\phi = \phi_1 F_1 + \phi_2 \left(1 - F_1\right). \tag{72}$$

This relation is applied to calculate appropriate values of the parameters σ_k, σ_ω, α and β using

$$\begin{aligned} \sigma_{k1} &= 0.85, & \sigma_{k2} &= 1, \\ \sigma_{\omega 1} &= 0.5, & \sigma_{\omega 2} &= 0.856, \\ \alpha_1 &= \tfrac{5}{9}, & \alpha_2 &= 0.44, \\ \beta_1 &= \tfrac{3}{40}, & \beta_2 &= 0.0828, \end{aligned} \tag{73}$$

where the parameter $\sigma_{\omega 2}$ was already given above. Since the two equation model switches according to the wall distance y, the eddy viscosity has to be dependent on the wall distance too. The form of the eddy viscosity is given by

$$\nu_T = \tfrac{k}{\max\left(\omega, S F_2\right)}, \tag{74}$$

where

$$F_2 = \tanh \left(\left[\max \left(\tfrac{2\sqrt{k}}{\beta^* \omega y}, \tfrac{500\nu}{y^2 \omega} \right) \right]^2 \right). \tag{75}$$

4 Numerical solution

The following paragraphs are devoted to the numerical solution of the stationary Navier–Stokes equations. Isogeometric analysis is considered in the experiments, which is based on Galerkin approach of discretization. However, the discrete spaces cannot be chosen arbitrarily to keep the stability of numerical solution. The discretization leads to nonlinear system for Navier–Stokes problem, which is solved iteratively applying linearization methods introduced later in the chapter.

4.1 Galerkin approach

The Galerkin type methods are based on approximation of the solution of the boundary value problem by a finite combination of linearly independent functions, known as basis functions, i.e. the continuous problem (more exactly weak formulation) is converted to a discrete problem, whose solution is a function of a finite dimensional subspace generated by the basis functions. Certainly, the appropriate choice of the basis functions and the way of determination of the coefficients of the linear combination are crucial and lead to different Galerkin type methods. The most common is finite element method (FEM) which is based on division of the computational domain into a finite number of non-overlapping subdomains that are called finite elements, over which the solution of any problem is approximated by local basis functions, typically polynomials for FEM. However, instead of the finite element method, isogeometric analysis is applied in the numerical experiments, which is usually understood as a modification to FEM. The detailed introduction to the isogeometric analysis is given in Section 4.2.

The Galerkin method is based on the weak formulation of the problem. In evolutionary case, the weak formulation is considered in the semi-discrete form with respect to the spatial variable. In the following paragraphs, the weak formulation is derived and the discretization is applied for the stationary Navier–Stokes problem. The weak formulation and Galerkin approximation of the Reynolds-averaged Navier–Stokes problem closed with the k–ω turbulence model is written in Appendices A and B due to long and complicated formulations. The weak formulation and numerical treatment of the Stokes problem, Oseen problem and unsteady Navier–Stokes problem is not included in the thesis, because its weak and discrete formulation is straightforward considering the derivation of the discrete steady Navier–Stokes and RANS problem.

4.1.1 Weak formulation

Let us remind the stationary boundary Navier–Stokes problem

$$\begin{aligned} -\nu\Delta\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p &= \mathbf{f} \quad \text{in } \Omega, \\ \nabla\cdot\mathbf{u} &= 0 \quad \text{in } \Omega, \\ \mathbf{u} &= \mathbf{g}_D \quad \text{on } \partial\Omega_D, \\ \nu\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - \mathbf{n}p &= \mathbf{g}_N \quad \text{on } \partial\Omega_N. \end{aligned} \tag{76}$$

As already mentioned in Section 3, the classical solution of the steady Navier–Stokes problem are vector function $\mathbf{u}(\mathbf{x},t) : \bar{\Omega} \to \mathbb{R}^d$ and scalar function $p(\mathbf{x},t) : \bar{\Omega} \to \mathbb{R}$ satisfying

(76) and

$$\mathbf{u} \in C^2\left(\bar{\Omega}\right), \quad p \in C^1\left(\bar{\Omega}\right). \tag{77}$$

To derive a weak formulation, the method of weighted residuals is usually used, i.e. the momentum equation in (76) is multiplied by a weighting function $\mathbf{v}$ and the equation of continuity by a weighting function q from appropriate spaces known as test functions. By integrating over the domain Ω, we have

$$\begin{aligned} \int\limits_{\Omega} (-\nu\Delta\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p - \mathbf{f})\cdot\mathbf{v} &= \mathbf{0}, \\ \int\limits_{\Omega} q\nabla\cdot\mathbf{u} &= 0. \end{aligned} \tag{78}$$

This formulation exists provided that the integrals are well defined and the proof can be found e.g. in [35, 37]. If $(\mathbf{u}, p)$ is a classical solution, then (78) is satisfied and equivalent to the strong formulation (76). If the test functions are sufficiently smooth, then the smoothness requirements for $(\mathbf{u}, p)$ can be reduced by using Green's and divergence theorem,

$$\begin{aligned} \nu\int\limits_{\Omega}\nabla\mathbf{u}:\nabla\mathbf{v} + \int\limits_{\Omega}(\mathbf{u}\cdot\nabla\mathbf{u})\mathbf{v} - \int\limits_{\Omega} p\nabla\cdot\mathbf{v} &= \int\limits_{\Omega}\mathbf{f}\cdot\mathbf{v} + \nu\int\limits_{\partial\Omega}(\mathbf{n}\cdot\nabla\mathbf{u})\cdot\mathbf{v} - \int\limits_{\partial\Omega} p\mathbf{n}\cdot\mathbf{v}, \\ \int\limits_{\Omega} q\nabla\cdot\mathbf{u} &= 0, \end{aligned} \tag{79}$$

where $\nabla\mathbf{u} : \nabla\mathbf{v}$ represents the componentwise scalar product, i.e. $\nabla u_1 \cdot \nabla v_1 + \nabla u_2 \cdot \nabla v_2$ in two dimensions. The boundary integrals can be rewritten assuming the boundary conditions in (76)

$$\begin{aligned} \nu\int\limits_{\partial\Omega}(\mathbf{n}\cdot\nabla\mathbf{u})\cdot\mathbf{v} - \int\limits_{\partial\Omega} p\mathbf{n}\cdot\mathbf{v} &= \int\limits_{\partial\Omega_D\cup\Omega_N}\left(\nu\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - \mathbf{n}p\right)\cdot\mathbf{v} \\ &= \int\limits_{\partial\Omega_D}\left(\nu\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - \mathbf{n}p\right)\cdot\mathbf{v} + \int\limits_{\partial\Omega_N}\mathbf{g}_N\cdot\mathbf{v}. \end{aligned} \tag{80}$$

The integrals in (79)-(80) exist if the corresponding gradients and $\mathbf{f}$ are in the Lebesgue space $L^2(\Omega)$. It leads to the following velocity solution space V and test function space V_0

$$\begin{aligned} V &= \{\mathbf{u} \in H^1(\Omega)^d \,|\, \mathbf{u} = \mathbf{g}_D \text{ on } \partial\Omega_D\}, \\ V_0 &= \{\mathbf{v} \in H^1(\Omega)^d \,|\, \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega_D\}, \end{aligned} \tag{81}$$

where d is the spatial dimension. Considering that the test function $\mathbf{v}$ is from the space V_0, the boundary integral over the Dirichlet part $\partial\Omega_D$ in (80) equals zero. The appropriate space for pressure p is $L^2(\Omega)$, since there are no pressure derivatives in (79). Moreover, choosing the pressure test function q from the Lebesgue space $L^2(\Omega)$ leads to the following weak formulation. Let $\nu > 0$, functions $\mathbf{u}$ and p are a weak solution of problem (30) if $\mathbf{u} \in V$, $p \in L^2(\Omega)$ satisfying

$$\begin{aligned} \nu \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} + \int_{\Omega} (\mathbf{u} \cdot \nabla \mathbf{u}) \mathbf{v} - \int_{\Omega} p \nabla \cdot \mathbf{v} &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} + \nu \int_{\partial \Omega_N} \mathbf{g}_N \cdot \mathbf{v}, \\ \int_{\Omega} q \nabla \cdot \mathbf{u} &= 0. \end{aligned} \tag{82}$$

for all $\mathbf{v} \in V_0$ and $q \in L^2(\Omega)$ In the rest of the thesis, we assume $\mathbf{g}_N = \mathbf{0}$ and thus the boundary integral in (82) equals to zero. As already mentioned, if $(\mathbf{u}, p)$ is a classical solution of (76), it also satisfies the weak form (82). The proof of existence and uniqueness of the weak solution is very technical including deep knowledge of the mathematical analysis and thus it is beyond the scope of this thesis, however, the reader is referred to [37] for details.

4.1.2 Linearization

Solving nonlinear Navier–Stokes equations requires linearization, such that the iteration process is used and a linearized problem is then solved at every iteration. The derivation is given for the steady case for simplicity, but it is identically applied for unsteady case in every time step, which is written for the RANS equations in Appendix A.

Newton's and Picard's methods are the classical approaches applied for the nonlinear iteration, [35, 99]. Consider an initial guess $(\mathbf{u}^0, p^0) \in V \times L^2(\Omega)$ and a sequence of iterates $(\mathbf{u}^1, p^1)$, $(\mathbf{u}^2, p^2)$, …, $(\mathbf{u}^k, p^k) \in V \times L^2(\Omega)$. Now, let $(\mathbf{u}^k, p^k)$ be known from the previous iteration step. Then the residuals of the weak formulation (82) fulfill

$$\begin{aligned} \mathbf{R}^k &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} - \nu \int_{\Omega} \nabla \mathbf{u}^k : \nabla \mathbf{v} - \int_{\Omega} (\mathbf{u}^k \cdot \nabla \mathbf{u}^k) \mathbf{v} + \int_{\Omega} p^k \nabla \cdot \mathbf{v}, \\ r^k &= - \int_{\Omega} q \nabla \cdot \mathbf{u}^k \, \mathrm{d}\Omega, \end{aligned} \tag{83}$$

for all $\mathbf{v} \in V_0$, $q \in L^2(\Omega)$. Now assume $\mathbf{u}$ and p be the exact solution of the weak form, which is given by the sum of solution of the k-th iteration and the correction

$$\mathbf{u} = \mathbf{u}^k + \delta \mathbf{u}^k, \quad p = p^k + \delta p^k. \tag{84}$$

Substituting (84) into the weak formulation (82) and using equalities (83), the deviation of the solution $(\delta \mathbf{u}^k, \delta p^k) \in V_0 \times L^2(\Omega)$ must satisfy

$$-\nu \int_{\Omega} \nabla \delta \mathbf{u}^k : \nabla \mathbf{v} - \int_{\Omega} (\mathbf{u}^k \cdot \nabla \delta \mathbf{u}^k) \mathbf{v} - \int_{\Omega} (\delta \mathbf{u}^k \cdot \nabla \mathbf{u}^k) \mathbf{v} - \int_{\Omega} (\delta \mathbf{u}^k \cdot \nabla \delta \mathbf{u}^k) \mathbf{v} +$$

$$
\begin{aligned}
+\int\limits_{\Omega} \delta p^k \nabla \cdot \mathbf{v} &= \mathbf{R}^k, \qquad (85)\\
-\int\limits_{\Omega} q \nabla \cdot \delta \mathbf{u}^k \,\mathrm{d}\Omega &= r^k, \qquad (86)
\end{aligned}
$$

for all $\mathbf{v} \in V_0$, $q \in L^2(\Omega)$. Now, we assume that the term quadratic in velocity corrections $\int_\Omega (\delta \mathbf{u}^k \cdot \nabla \delta \mathbf{u}^k)\mathbf{v}$ is negligible and thus it can be omitted. This implies the linear problem for the corrections $\delta \mathbf{u}^k \in V_0$ and $\delta p^k \in L^2(\Omega)$, the so called Newton correction,

$$
\begin{aligned}
-\nu \int\limits_{\Omega} \nabla \delta \mathbf{u}^k : \nabla \mathbf{v} - \int\limits_{\Omega} (\mathbf{u}^k \cdot \nabla \delta \mathbf{u}^k)\mathbf{v} - \int\limits_{\Omega} (\delta \mathbf{u}^k \cdot \nabla \mathbf{u}^k)\mathbf{v} + \int\limits_{\Omega} \delta p^k \nabla \cdot \mathbf{v} &= \mathbf{R}^k,\\
-\int\limits_{\Omega} q \nabla \cdot \delta \mathbf{u}^k &= r^k,
\end{aligned}
$$

for all $\mathbf{v} \in V_0$, $q \in L^2(\Omega)$. After solving this problem, the approximation of the weak solution from the previous iteration can be updated, which defines the next iterate in the sequence, i.e.

$$
\mathbf{u}^{k+1} = \mathbf{u}^k + \delta \mathbf{u}^k, \quad p^{k+1} = p^k + \delta p^k. \qquad (87)
$$

The consistency of the iteration process is discussed in [35]. One disadvantage of the linearization is the choice of the initial guess. Although the Newton method gives quadratic convergence, the initial guess needs to be closer and closer to the exact solution as the Reynolds number increases, [35]. The solution of the linear Stokes problem is possible to choose as the initial guess.

Second approach is the Picard's method [99, 75, 35], which will be applied for solving nonlinear problem in this thesis. Compared to the Newton's method, the term $\int_\Omega (\delta \mathbf{u}^k \cdot \nabla \mathbf{u}^k)\mathbf{v}$ in (85) is also dropped. The nonlinear term is thus linearized using the approximate solution $(\mathbf{u}^k, p^k)$ from the previous iteration step. Then we look for $\mathbf{u}^{k+1} \in V$ and $p^{k+1} \in L^2(\Omega)$, $k = 0, 1, \ldots$, such that for all $\mathbf{v} \in V_0$ and $q \in L^2(\Omega)$ it holds

$$
\begin{aligned}
\nu \int\limits_{\Omega} \nabla \mathbf{u}^{k+1} : \nabla \mathbf{v} + \int\limits_{\Omega} (\mathbf{u}^k \cdot \nabla \mathbf{u}^{k+1})\mathbf{v} - \int\limits_{\Omega} p^{k+1} \nabla \cdot \mathbf{v} &= \int\limits_{\Omega} \mathbf{f} \cdot \mathbf{v},\\
\int\limits_{\Omega} q \nabla \cdot \mathbf{u}^{k+1} &= 0.
\end{aligned} \qquad (88)
$$

Note that each Picard's iteration (88) is the weak formulation of the Oseen problem (31) with $\mathbf{w} = \mathbf{u}^k$. The solution of the Stokes problem (28) is usually chosen to be the initial condition $(\mathbf{u}^0, p^0)$ of the Picard's iteration (88).

As mentioned above, the solution of the Stokes problem can be chosen as an initial guess of the Newton method, but preferable is to solve several iterations using Picard's method and then the Newton's linearization proceeds, because Picard's iterations give closer approximation to the exact solution.

4.1.3 Discrete problem

In the following, we look for an approximation of the weak solution $(\mathbf{u}, p)$ by taking a finite dimensional subspaces of the solution spaces $V_0^h \subset V_0$ and $W^h \subset L^2(\Omega)$ such that an arbitrary function in the solution space can be approximated as linear combination of sufficient number of the basis functions with arbitrary accuracy, i.e. the finite dimensional subspaces are vector spaces generated by a basis φ_i^u, $i = 1, \ldots, n^u$ and φ_i^p, $i = 1, \ldots, n^p$, [104, 101, 83, 37]. Because the function spaces for velocity and pressure are approximated independently, we talk about a mixed approximation. Recall that the Dirichlet boundary condition is assumed for the velocity (76). To ensure the Dirichlet condition, the set of velocity basis is extended by defining additional functions φ_i^u, $i = n^u + 1, \ldots, n_D^u$, which leads to the final velocity finite dimensional vector space $V^h \subset V$. The Galerkin approximations $\mathbf{u}_h \in V^h$, $p_h \in W^h$ are then given as a linear combination of the appropriate basis functions

$$\begin{aligned} \mathbf{u}_h &= \sum_{j=1}^{n^u} \mathbf{u}_j \varphi_i^u + \sum_{j=n^u+1}^{n_D^u} \mathbf{u}_j^* \varphi_j^u, \\ p_h &= \sum_{j=1}^{n^p} p_j \varphi_j^p. \end{aligned} \tag{89}$$

with the unknown coefficients $\mathbf{u}_j \in \mathbb{R}^d$, $j = 1, \ldots, n^u$, $p_j \in \mathbb{R}$, $j = 1, \ldots, n_p$ and fixed coefficients $\mathbf{u}_j^*$, $j = n^u + 1, \ldots, n_D^u$ corresponding to basis functions φ_j^u, $j = n^u + 1, \ldots, n_D^u$ such that the second term in (89) interpolates the Dirichlet boundary condition on $\partial\Omega_D$. The first summation term is an element of the vector subspace V_0^h, i.e. the basis functions $\varphi_1^u, \ldots, \varphi_{n_u}^u$ are zero at Dirichlet part of the boundary. The approximation of the velocity field for the three dimensional flow is

$$\mathbf{u}_h = \sum_{j=1}^{n^u} (u_{1j}, u_{2j}, u_{3j})^T \varphi_i^u + \sum_{j=n^u+1}^{n_D^u} (u_{1j}^*, u_{2j}^*, u_{3j}^*)^T \varphi_j^u. \tag{90}$$

Analogously, function $\mathbf{f}$ can be written as a linear combination of velocity basis functions by

$$\mathbf{f}_h = \sum_{i=1}^{n_D^u} (f_{1i}, f_{2i}, f_{3i})^T \varphi_i^u. \tag{91}$$

For general $\mathbf{f}$, $\mathbf{f}_h$ can be obtained by L^2 projection to the linear space spanned by basis functions $\{\varphi_i^u\}_{1 \le i \le n_D^u}$. To discretize the linearized weak formulation (88) by means of the Galerkin method, the discrete solution and test basis functions are chosen to be identical, i.e. $\mathbf{v}_h \in V_0^h$ and $q_h \in W^h$. Assume that the discrete solution in k–th Picard's iteration $\mathbf{u}_h^k$ is known, then we look for $\mathbf{u}_h^{k+1} \in V^h$ and $p_h^{k+1} \in W^h$ satisfying

$$\nu \int_\Omega \nabla \mathbf{u}_h^{k+1} : \nabla \mathbf{v}_h + \int_\Omega (\mathbf{u}_h^k \cdot \nabla \mathbf{u}_h^{k+1}) \mathbf{v}_h - \int_\Omega p_h^{k+1} \nabla \cdot \mathbf{v}_h = \int_\Omega \mathbf{f}_h \cdot \mathbf{v}_h,$$

$$\int_\Omega q_h \nabla \cdot \mathbf{u}_h^{k+1} \;=\; 0, \tag{92}$$

for all $\mathbf{v}_h \in V_0^h$ and $q_h \in W^h$. Because the velocity and pressure approximations satisfy (89) and (92) holds for all test functions, it must be satisfied for all basis functions from the corresponding finite dimensional subspaces $\varphi_i^u \in V_0^h$ and $\varphi_i^p \in W^h$. Then it follows that (92) is equivalent to finding coefficients $\mathbf{u}_j^{k+1}$, $j = 1, 2, \ldots, n^u$ at every Picard's iteration $k = 0, 1, \ldots$ (remember that $\mathbf{u}_{n_u+1}, \ldots, \mathbf{u}_{n_u+n_D^u}$ are known coefficients corresponding to the Dirichlet condition) and p_j, $j = 1, 2, \ldots, n_p$ such that

$$\begin{aligned}
\sum_{j=1}^{n^u} \mathbf{u}_j^{k+1} \underbrace{\nu \int_\Omega \nabla\varphi_j^u \cdot \nabla\varphi_i^u}_{A_{ij}} + \underbrace{\int_\Omega \varphi_i^u \left(\sum_{m=1}^{n^u} \mathbf{u}_m^k \varphi_m^u \right) \cdot \nabla\varphi_j^u}_{N_{ij}(\mathbf{u}^k)} & \\
- \sum_{j=1}^{n^p} p_j^{k+1} \underbrace{\int_\Omega \varphi_j^p \nabla \cdot \varphi_i^u}_{-B_{1ji},\ldots,-B_{dji}} &= \sum_{i=1}^{n_D^u} \mathbf{f}_{c,i} \underbrace{\int_\Omega \varphi_i^u \varphi_j^u}_{C_{ij}} - \\
&- \sum_{j=n^u+1}^{n_D^u} \mathbf{u}_j^*(A_{ij} + N_{ij}) \\
, & \\
\sum_{j=1}^{n^u} \mathbf{u}_j^{k+1} \underbrace{\int_\Omega \varphi_l^p \nabla \cdot \varphi_j^u}_{B_{1lj},\ldots,B_{dlj}} &= \mathbf{b}_{c4},
\end{aligned}$$

where $i = 1, \ldots, n_u$, $l = 1, \ldots, n_p$ and d is the space dimension. This can be written in the matrix form as the linear system of equations, e.g. in three space dimension $d = 3$,

$$\begin{bmatrix} \mathbf{A} + \mathbf{N}(\mathbf{u}^k) & \mathbf{0} & \mathbf{0} & -\mathbf{B}_1^T \\ \mathbf{0} & \mathbf{A} + \mathbf{N}(\mathbf{u}^k) & \mathbf{0} & -\mathbf{B}_2^T \\ \mathbf{0} & \mathbf{0} & \mathbf{A} + \mathbf{N}(\mathbf{u}^k) & -\mathbf{B}_3^T \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^{k+1} \\ \mathbf{u}_2^{k+1} \\ \mathbf{u}_3^{k+1} \\ \mathbf{p}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{c1} \\ \mathbf{f}_{c2} \\ \mathbf{f}_{c3} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{b}_{c1} \\ \mathbf{b}_{c2} \\ \mathbf{b}_{c3} \\ \mathbf{b}_{c4} \end{bmatrix}, \tag{93}$$

where

$$\begin{bmatrix} \mathbf{b}_{c1} \\ \mathbf{b}_{c2} \\ \mathbf{b}_{c3} \\ \mathbf{b}_{c4} \end{bmatrix} = \begin{bmatrix} (\mathbf{A}^* + \mathbf{N}^*(\mathbf{u}^k)) \cdot \mathbf{u}_1^* \\ (\mathbf{A}^* + \mathbf{N}^*(\mathbf{u}^k)) \cdot \mathbf{u}_2^* \\ (\mathbf{A}^* + \mathbf{N}^*(\mathbf{u}^k)) \cdot \mathbf{u}_3^* \\ (\mathbf{B}_1^* \cdot \mathbf{u}_1^* + \mathbf{B}_2^* \cdot \mathbf{u}_2^* + \mathbf{B}_3^* \cdot \mathbf{u}_3^*) \end{bmatrix}$$

and

$$\begin{array}{lcllcl} \mathbf{A} & = & [A_{ij}]_{1\leq i\leq n^u, 1\leq j\leq n^u}, & \mathbf{A}^* & = & [A_{ij}]_{1\leq i\leq n^u, n^u+1\leq j\leq n^u_D}, \\ \mathbf{N}(\mathbf{u}) & = & [N_{ij}(\mathbf{u})]_{1\leq i\leq n^u, 1\leq j\leq n^u}, & \mathbf{N}^*(\mathbf{u}) & = & [N_{ij}(\mathbf{u})]_{1\leq i\leq n^u, n^u+1\leq j\leq n^u_D}, \\ \mathbf{B}_m & = & [B_{mij}]_{1\leq i\leq n^p, 1\leq j\leq n^u}, & \mathbf{B}^*_m & = & [B_{mij}]_{1\leq i\leq n^p, n^u+1\leq j\leq n^u_D}, \\ \mathbf{C} & = & [C_{ij}]_{1\leq i\leq n^u, 1\leq j\leq n^u}, & \mathbf{C}^* & = & [C_{ij}]_{1\leq i\leq n^u, n^u+1\leq j\leq n^u_D}, \\ \mathbf{f}_{cm} & = & [\mathbf{C}|\mathbf{C}^*]\cdot[f_{mi}]_{1\leq i\leq n^u_D}, & & & \end{array} \tag{94}$$

and $m = 1, \ldots, d$. Note that the Neumann outflow boundary condition (26) for velocity and pressure is chosen to be zero in this work. The system of $(d \cdot n_u + n_p)$ linear equations for $(d \cdot n_u + n_p)$ unknowns is called Galerkin system. It should be noted that similarly to continuous case, velocity coefficient vector is unique, but solvability with respect to the pressure is problematic. Detailed discussion about solvability of the matrix system can be found in [35]. In our case, where the Neumann outflow condition is applied and so the volume of the fluid flowing in the domain is equal to the volume of the fluid flowing out, the solvability is unique.

Another important issue is to find appropriate finite dimentional spaces to guarantee the stability of the numerical solution, which is discussed in the following section.

In the case of unsteady Navier–Stokes and RANS problems, the Galerkin approach is used identically to the steady Navier–Stokes problem with additional term which represents approximation of the time derivative. Similarly to the weak formulation of the unsteady Navier–Stokes and RANS problems, the discrete form of the RANS problem is given in Appendix B from which the derivation of the discrete Navier–Stokes problem is straightforward and thus it is not included in this thesis.

Solution of the discrete problem

In this paragraph, we briefly introduce the approaches of solving system of equations resulting from the Galerkin discretization of the flow problem. Note that the spatial discretization of all flow problems mentioned above (steady and unsteady Navier–Stokes, Reynolds–Averaged Navier–Stokes problems) using an implicit time discretization result in nonsymmetric linear systems of saddle-point type. A lot of attention has been devoted to the solution of the large saddle-point linear systems, see e.g. [9]. Since the saddle-point problem does not arise only in the context of the incompressible fluid flow, researchers from various fields (e.g. linear elasticity and constrained optimization) are interested in the solution methods for this problem.

Generally, two approaches were developed for the solution of the saddle-point problems, coupled and decoupled (segregated). A straightforward way is to assemble the whole linear system computed for coefficients of both velocity and pressure approximations. On the other hand, decoupled methods compute the velocity and pressure solution separately. Thanks to decoupling, we solve several smaller systems instead of one large system in the particular steps. An example of such method is the class of pressure–correction methods (cf. e.g. [104]), where the momentum equations are solved first to obtain an intermediate velocity field using the pressure approximation from the previous step. However, this velocity field does not fulfill the condition of incompressibility. Next, the pressure correction is computed from a Poisson equation and the intermediate velocity field is projected onto the space of divergence-free vector fields. SIMPLE algorithm [94] is an well known example of the pressure–correction scheme. Although the smaller and thus less time and memory consuming systems are solved, the convergence of the decoupled methods is often very slow.

The coupled approach is applied for solving the linear systems in this thesis. Direct and iterative methods are two groups of the solution methods for systems of linear algebraic equations. Direct methods [99, 75] are based on a factorization of the system matrix. They are robust, give the exact solution of the system of equations in exact arithmetic, but they are very computational consuming. Thus, solving linear system of equations using direct methods is almost not feasible for large system matrices.

On the other hand, iterative methods are less robust, give usually less accurate solution than direct solvers and main disadvantage is that they require many iterations to converge. The major classes are the stationary, nonstationary and multilevel methods. The stationary methods are very simple to implement, they are fast, but their convergence is usually very slow and moreover it is not guaranteed for general matrices. The stationary methods are expressed by a simple scheme, which does not change from iteration to iteration. The four main stationary methods are the Jacobi method, Gauss–Seidel method and SOR (successive overrelaxation) method [75], which are nowadays used only as preconditioners or smoothers for the other classes of iterative methods in practical computations.

The most typical representatives of the nonstationary iterative methods are Krylov subspace methods, [101, 77, 99]. They belong to the most commonly used iterative methods in practice. Krylov subspace methods were constructed such that they converge to the exact solution of the system of linear algebraic equations in finite number of iterations (so called finite termination property), however, only in exact arithmetic. In finite arithmetic, the convergence behavior is much more complicated and thus Krylov methods generally converge after an infinite number of iterations. Actually, the distribution of spectrum of the system matrix strongly affects the convergence rate. To achieve fast convergence, it is necessary to use some preconditioning technique, [100, 66, 99]. The preconditioner should be easy to construct and the linear system with the preconditioned matrix should be effectively solvable.

The preconditioner can be either algebraic, e.g. incomplete LU factorization, or based on knowledge of the problem origin, discretization, matrix structure etc. The latter class of preconditioners, including the so called block preconditioners, is the most suitable for solving linear systems associated with the incompressible Navier–Stokes equations. Examples of the block preconditioner are LSC (least-squares commutator) and AL (augmented Lagrangian), see [33].

The study and implementation of the iterative methods with a good choice of the preconditioner is not a trivial task and thus it is beyond the scope of this thesis. Then, the direct solver with LU decomposition is employed for the time being.

4.2 Isogeometric analysis

The continuous Galerkin method was introduced to discretize the system of partial differential equations. It approximates the unknown functions of the boundary (or initial-boundary) value problem over the computational domain as defined in Section 4.1.3, i.e.,

$$\mathbf{u}_h = \sum_{j=1}^{n^u} (u_{1j}, u_{2j}, u_{3j})^T \varphi_i^u + \sum_{j=n^u+1}^{n_D^u} (u_{1j}^*, u_{2j}^*, u_{3j}^*)^T \varphi_j^u, \tag{95}$$

$$p_h = \sum_{j=1}^{n^p} p_j \varphi_j^p. \tag{96}$$

Then, the discrete Galerkin formulation resulted in a system of algebraic equations (93) for the unknown coefficients $\mathbf{u}_j, j = 1, \ldots, n^u$ of the velocity approximation and $p_j, j = 1, \ldots, n^p$ of the pressure approximation. The bases of the discrete spaces V^h, V_0^h and W^h have not been specified so far. Different choice of the basis functions leads to a different Galerkin type method. Finite element method (FEM) is the most common Galerkin type method that was developed to solve complex elasticity and structural analysis problems related to aerospace and civil engineering and the earliest paper on FEM was proposed by Richard Courant. Nowadays, FEM benefits in solving fluid structure interaction, heat transfer and other engineering disciplines.

Finite element method is based on a partition of the computational domain into a finite number of smaller parts, called finite elements. If the computational domain is an interval $\Omega = (a, b)$, then the elements are subintervals $K_i = (x_{i-1}, x_i), i = 1, \ldots, N$ with $a = x_0 < x_1 < \ldots < x_{N-1} < x_N = b$, such that $\Omega = \bigcup_{i=1}^{N} K_i$. Then the basis functions φ_j^u, φ_j^p of the solution spaces are piecewise polynomial functions on a partition of the domain Ω, which equals to 1 at the given point x_i and equals to zero at the rest of the points. The advantage is that the basis functions have narrow support which leads to sparse matrix of the algebraic system. The finite element basis functions are generally C^0 continuous, since the higher order regularity is achieved only in the interior points of the elements, but it is violated at the boundary of every element. In 2D case, we can generalize the previous idea such that polygons represent finite elements (usually triangles or rectangles) and the number of piecewise polynomial basis functions corresponds to the number of vertices of the partition of the computational domain.

However, the numerical modeling of the fluid flow is based on isogeometric analysis (IgA) in this thesis, which is a recently developed Galerkin type approach in [50]. IgA shares a lot of features with FEM and it is even usually understood as the modification of FEM such that another basis functions are chosen in the Galerkin approximation (95). Contrarily to FEM, IgA is closely related to the description of geometry and takes inspiration from Computer Aided Design (CAD), which allows exact geometry representation. Indeed, the computational domain with boundary represented as B-spline/NURBS objects (i.e. polynomial/rational parametric objects introduced in the following paragraph) can be discretized exactly and then the isoparametric approach is applied, that is, the solution spaces of the velocity and pressure approximation are generated by the same basis functions which represent the geometry. This is the main advantage of IgA, which cannot be reached by FEM polynomial description of the boundaries.

NURBS objects

In the following, we introduce the B-spline/NURBS generation of a geometry, which is the core of the isogeometric analysis, since the same basis functions are used for geometry representation and approximation of a problem solution. This is only a brief introduction to the NURBS object theory, which is necessary to understand their relation to the isogeometric analysis, but the reader is referred to [97, 50] for the geometry details.

The basis for understanding B-spline curves (surfaces) are splines, which are piecewise polynomial functions with high continuity. In the NURBS theory, this is extended to even piecewise rational functions. Both, B-spline and NURBS objects are defined by the B-spline basis functions. Let us consider n be the number of basis functions and consider a knot vector $\mathbf{A} = (\alpha_1, \alpha_2, \dots, \alpha_{n+q+1})$, such that

$$\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_{n+q+1} \tag{97}$$

then the B-spline basis functions of the degree q are defined recursively by

$$N_i^q(\alpha) \;=\; \frac{\alpha - \alpha_i}{\alpha_{i+q} - \alpha_i} N_i^{q-1}(\alpha) + \frac{\alpha_{i+q+1} - \alpha}{\alpha_{i+q+1} - \alpha_{i+1}} N_{i+1}^{q-1}(\alpha) \tag{98}$$

starting with piecewise constants

$$N_i^0(\alpha) \;=\; \left\{ \begin{array}{ll} 1 & \alpha_i \leq \alpha < \alpha_{i+1} \\ 0 & \text{otherwise.} \end{array} \right. \tag{99}$$

Considering (97), a denominator in B-spline basis definition (98) can be zero. In this case, the appropriate fraction is set to zero. Knot vector is a non-decreasing sequence of real numbers which determines the distribution of a parameter on the corresponding curve/surface. The knots can be located uniformly or non-uniformly in the interval $[\alpha_1, \alpha_{n+q+1}]$. B-spline basis functions of degree q are C^{q-1} continuous in general. Knot repeated k times in the knot vector decreases the continuity of B-spline basis functions by $k-1$. Support of B-spline basis functions is local. It is nonzero only on the interval $[\alpha_i, \alpha_{i+q+1}]$ in the parametric space and each B-spline basis function is non-negative, i.e.,

$$N_i^q(\alpha) \geq 0, \quad \forall \alpha \in [\alpha_1, \alpha_{n+q+1}]. \tag{100}$$

Examples of the B-spline basis are shown in Figure 1 and 2. In the case that the first and last knot of the knot vector repeat $q+1$ times, the knot vector is said to be open. The open knot vectors are standard in isogeometric analysis and they are also used in our numerical experiments. Basis functions formed from the open vectors lead to an interpolation at the ends of the parametric space, which is beneficial in the real computations since a boundary condition needs to be fulfilled. However in the interior of the parametric domain, the behaviour of the B-spline basis functions is not automatically interpolatory, but only in the case of C^0 continuity at any interior knot. This is the difference between B-spline and finite element basis. Nevertheless, the piecewise constant and linear (i.e. for $q = 0$ and 1) B-spline basis functions are the same as the piecewise constant and linear finite element basis functions. However, considering Figures 1 and 2, it is evident that the finite element and B-spline basis functions differ for $q \geq 2$. The partition of unity is remaining property of B-spline basis functions, i.e

$$\sum_{i=1}^{n} N_i^q(\alpha) = 1, \quad \forall \alpha \in [\alpha_1, \alpha_{n+q+1}]. \tag{101}$$

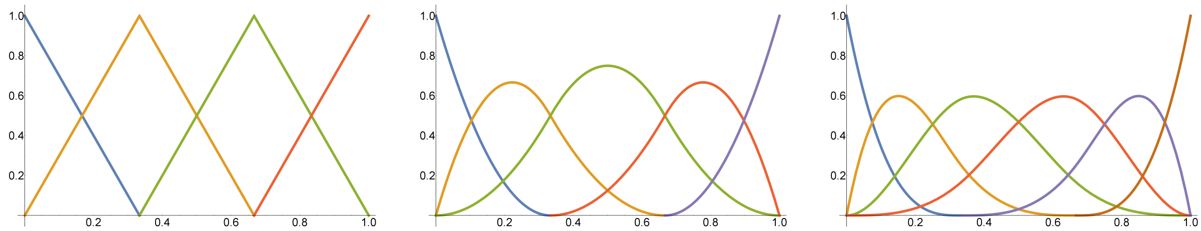


Figure 1: Linear basis for knot vector $A = (0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1)$ (left), quadratic basis for knot $A = (0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1)$ (middle), cubic basis for knot vector $A = (0, 0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1)$ (right).

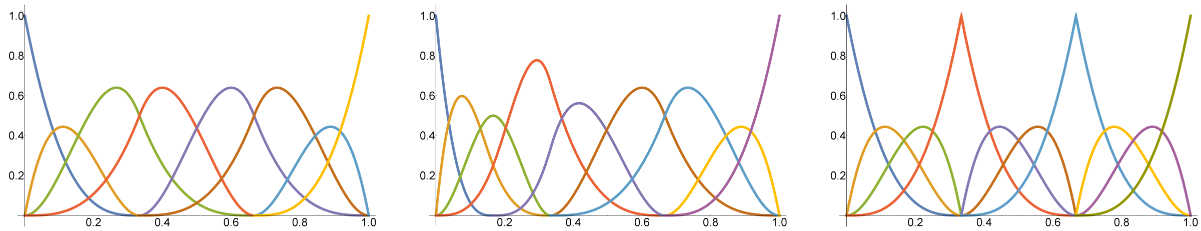


Figure 2: Cubic basis for knot vector $A = (0, 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1, 1)$ (left), $A = (0, 0, 0, 0, \frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1, 1)$ (middle), $A = (0, 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1, 1)$ (right).

B-spline curve in $\mathbb{R}^d$ is then a vector function of one parameter (i.e. mapping of a line segment into d-dimensional space) and it is given as a linear combination of B-spline basis functions

$$\mathbf{C}(\alpha) = \sum_{i=1}^{n} \mathbf{P}_i N_i^q(\alpha), \tag{102}$$

where $\mathbf{P}_i, i = 1, \ldots, n$ are the coefficients of the linear combination, called control points. The control points create control polygon, which is a polyline whose vertices are the control points. The control polygon can be reached as a piecewise linear B-spline curve (i.e. $\mathbf{C}(\alpha)$ from (102) for $q = 1$) since linear B-spline basis give interpolation of the control points.

However, the control points are not interpolated by B-spline curves for general basis degree $q \geq 2$ as already mentioned above. According to the curve and B-spline basis definitions, note that a change of one control point $\mathbf{P}_i$ affects the curve only locally, but a wider surrounding is affected with increasing degree of the basis functions.

The B-spline curves inherit the properties from the basis functions, i.e. for example the B-spline curves have continuous derivatives of order $q-1$ in the absence of repeated knots or control points. Note that each inner knot should not be repeated more than q times, because the curve would become discontinuous otherwise. More details about the B-spline basis and B-spline curves can be found in [97, 50, 54].

The B-spline representation can describe a wide range of curves, but the B-splines are piecewise polynomials, thus the conic sections (except parabola) cannot be represented exactly by B-splines, [50]. In this case, a rational description is required, i.e. NURBS (Non-Uniform Rational B-Spline) representation. To construct a rational curve, every control point $\mathbf{P}_i$ is assigned to a weight w_i. Then the NURBS curve of degree q is given by a parametrization

$$\mathbf{C}(\alpha) = \frac{\sum_{i=1}^{n} w_i \mathbf{P}_i N_i^q(\alpha)}{\sum_{i=1}^{n} w_i N_i^q(\alpha)} = \sum_{i=1}^{n} \mathbf{P}_i R_i^q(\alpha), \tag{103}$$

where

$$R_i^q(\alpha) = \frac{w_i N_i^q(\alpha)}{\sum_{j=1}^{n} w_j N_j^q(\alpha)} \tag{104}$$

are NURBS basis functions. Note that some of the weights can be zero, however, at least one weight must be nonzero to make the fraction well defined. Note that if the weights satisfy $w_i = a$, $i = 1, 2, \ldots, n$, where a is a nonzero constant, then the NURBS becomes B-spline.

Although the NURBS curves provide exact construction of more complicated shapes (like circle and other conic sections), the memory requirements of the evaluation of rational functions are too high for practical computations. Then, we restrict ourselves to polynomial B-spline description in this thesis. However, the reader is referred to [97, 50, 54] for more details about NURBS objects.

In the real applications, we are rather interested in B-spline surfaces/solids. A surface/solid is a vector function of two/three parameters $(\alpha,\, \beta)/(\alpha,\, \beta,\, \gamma)$. There are several ways how to represent the surfaces/solids, however, the tensor product is the simplest and standard approach for B-splines. Then, the B-spline surface ([50]) of degree q is determined by a control net (instead of control polygon for B-spline curves) of control points $\mathbf{P}_{i,j}, i = 1, 2, \ldots, n, j = 1, 2, \ldots, m$ and two knot vectors $\mathbf{A} = (\alpha_1, \ldots, \alpha_{n+q+1})$, $\mathbf{B} = (\beta_1, \ldots, \beta_{m+q+1})$ and it is given by a parametrization

$$\mathbf{S}(\alpha, \beta) = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{P}_{i,j} N_i^q(\alpha) M_j^q(\beta) = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{P}_{i,j} B_{ij}^q(\alpha, \beta) = \sum_{j=1}^{n^u} \mathbf{P}_j B_j^q(\alpha, \beta), \tag{105}$$

where $n^u = n \cdot m$, $N_i^q(\alpha)$, $M_j^q(\beta)$ correspond to the knot vectors $\mathbf{A}$, $\mathbf{B}$, respectively, and they are defined by (98) - (99) and $B_{ij}^q(\alpha, \beta)$ is the corresponding tensor product B-spline basis $N_i^q(\alpha)M_j^q(\beta)$. Note that the degree of the basis can differ in each parameter, but B-splines of equal degree are applied in our numerical experiments and thus we denote them identically by q. The B-spline solid representation is an extension of the B-spline surfaces and thus it is not considered here for simplicity.

Isoparametric concept

In the previous, we introduced the geometry representation by the B-spline description, i.e. we prepared computational domain, in which the fluid flow simulation is considered. A B-spline mesh is defined by the product of knot vectors, i.e. $\mathbf{A} \times \mathbf{B}$ in two dimensions. The span of the knots subdivides the domain into elements. Then, Galerkin discretization of an (initial-)boundary value problem is considered such that the isoparametric concept is required. It means that the discrete solution spaces of the approximation of the unknown quantities are generated by the same B-spline basis functions which represent the geometry. The coefficients of the linear combination of the basis functions are then the unknowns - control variables and we follow in the same way as described in Section 4.1 which leads to the system of linear algebraic equations.

Theory of B-spline/NURBS objects directly implies that it is not possible (or reasonable, in some cases) to describe an object of arbitrary topology by one B-spline/NURBS object [27]. One of the typical examples is a rectangular plate with two holes. The reason for this lies in a regular control net describing B-spline/NURBS object, which is composed of $m \times n$ (in 2D) control points.

Thus, if the isogeometric analysis is used for numerical solving of partial differential equations, it is usually necessary to decompose a computational domain into subdomains, which are suitable for description by one B-spline/NURBS object. Such a domain is then composed of multiple B-spline/NURBS patches and will be called multipatch domain in the following. Then, any solver based on isogeometric analysis working on multipatch domains has to be able to join B-spline/NURBS patches along their interfaces into one computational domain.

Generally, multipatch domains are usually distinguished by the type of connection of patches into:

- conforming B-spline/NURBS meshes – patches have along a common interface the same elements (discretization) and the same control nets describing this interface (see Figure 3 (top)),
- nested B-spline/NURBS meshes – discretization of the common interface of left patch is obtained as a uniform refinement of a discretization of the interface of the right patch, or vice versa (see Figure 3 (middle)),
- non-conforming B-spline/NURBS meshes – description of the common interface between patches is completely independent on both patches (see Figure 3 (bottom)).

Conforming meshes can be handled easily – it is enough to identify the corresponding control points determining the common interface on both patches. The case of nested meshes is more complex, but it is still possible to find a direct relation between control points determining a common interface on both patches. The most complex case is then represented by non-conforming meshes which is usually handled differently (e.g. discontinuous Galerkin method applied only along the interfaces).

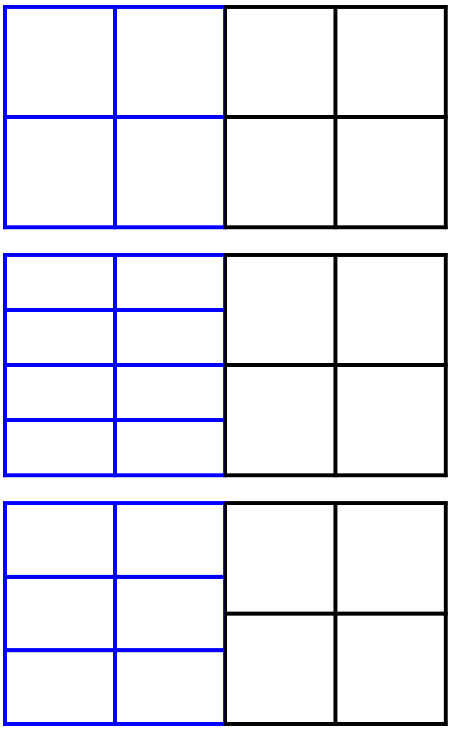


Figure 3: Examples of multipatch domains with different type of connection. Top: Conforming NURBS meshes; Middle: Nested NURBS meshes; Bottom: Non-conforming NURBS meshes.

Note that the geometry description usually leads to very coarse meshes for practical computations. It means that quite coarse control net is commonly sufficient to represent a geometry exactly, but it is not satisfactory for the numerical purposes, especially in the case of simulation of the turbulent flow in our case. Then, the mesh refinement strategies have to be employed. It was mentioned that the computational mesh is dependent on the knot vectors, i.e. the increasing number of the distinct knots in the knot vectors leads to finer grid. This can be reached by knot insertion known as h-refinement (similarly to FEM) such that the knots are inserted (possibly even multiple times) without changing the B-spline object geometrically or parametrically. The h-refinement of the B-splines can be implemented by the commonly known algorithms, which can be found in [97, 50]. Compared to finite element method, h-refinement is generally easy to perform in isogeometric analysis. Once the computational domain is constructed, the knot insertion is automatic process and the user only have to decide the number of refinement. Moreover, the B-spline object is still represented exactly at each refinement level. Contrarily, the process of refinement is much more complicated in finite element analysis and requires manual treatment. Note that h-refinement of the B-spline requires increasing number of the control points and thus number of basis functions, which leads to higher number of degrees of freedom, i.e. larger algebraic system of equations needs to be solved. Considering the basis in Figure 2a (which is cubic basis with C^1 continuity because of repeated knots $\frac{1}{3}$ and $\frac{2}{3}$), a refined basis obtained by inserting new knot $\frac{1}{6}$ into the knot vector can be seen in Figure 2b.

In some cases, it is required to elevate the degree of the basis functions, e.g. to achieve higher numerical accuracy in solving partial differential equations. The degree elevation (p-refinement) is possible to perform without any changes of the geometry representation or parametrization again. The knot vector changes so that the multiplicity of every knot is increased by 1. Thus, the resulting B-spline object preserve the order of the continuity. Similarly to h-refinement, the number of control points in the knot vector is increased applying the p-refinement. Figures 2a shows an example of a cubic B-spline basis with C^1 continuity obtained from the quadratic basis in Figure 1b by degree elevation. In Figure 2c, the cubic basis with C^0 continuity is seen obtained from basis in Figure 2a by knot insertion of the inner knots repeatedly.

It is also possible to increase both degree of the basis and its continuity. It is shown in Figure 1, where we start with linear basis ($q = 1$) for a knot vector $U = (0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1)$ with C^0 continuity. If we repeat the first and last knot once, resp. two times, we get quadratic basis ($q = 2$) for a knot vector $U = (0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1)$ with C^1 continuity, resp. cubic basis ($q = 3$) for a knot vector $U = (0, 0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1)$ with C^2 continuity.

It is possible to combine degree elevation (p-refinement) and knot insertion (h-refinement), but note that the processes of knot insertion and degree elevation do not commute. It is recommended to elevate the degree of the basis on the coarsest level first and then insert knots into the knot vector. This results in less basis functions of a higher order of continuity than if we first refine the knot vector and then elevate the degree. This is called k-refinement, which has no similarity to the finite element method. More details about the treatment of the mesh refinement can be found in [97, 50].

4.3 Stability

The disadvantage of many discretization methods is that they suffer from two main sources of instabilities. One is an incompatibility of pressure and velocity such that the discretization of velocity and pressure has to satisfy special requirements. Indeed, the so called inf–sup condition (or Ladyženskaja–Babuška–Brezzi condition (LBB)) has to be fulfilled to guarantee the stability described in Section 4.3.1. Not satisfied LBB condition leads to the instability which usually appears as oscillations, primarily in the approximate pressure solution. To avoid the restriction following from the inf–sup condition, the additional stabilization technique can be employed, which is also discussed in Section 4.3.1.

The other source of instability is the possible dominance of the advective term over the diffusion term in the problem of Navier–Stokes equations, i.e. in case of the turbulent flow. Also, the system of RANS equations is closed with the turbulence model, which is the advection dominated system of time–dependent advection–diffusion–reaction equations with the source term. Often, the reaction term is significant in some parts of the domain and thus the problem becomes reaction dominated, too. Moreover, most of the turbulence models are problems with non-constant coefficients and usually problems with nonlinear terms, which makes the problem even more challenging.

The basic difficulties are the interior and boundary layers in the solution of the advection dominated problems, where the solution gradients are very large. These layers usually arise from the discontinuous, non-smooth or rapidly changing data, which is typical for the turbulent flow, or generally for advection dominated problems. The width of the sharp layers is usually smaller than the mesh size and thus the layers cannot be resolved properly. It leads to unwanted spurious (nonphysical) oscillations in the numerical solution, which causes loss of accuracy and stability.

The stabilization techniques have to be employed to improve the stability in the case of dominated advection/reaction. But the resulting method should be able to resolve the sharp layers and to prevent the occurrence of the spurious oscillations at the same time to get an accurate numerical solution.

Most of the stabilization techniques follow from the study of a simple steady linear advection–diffusion problem. Note that the numerical solution of this problem is equally difficult if the advection is dominant and the numerical solution suffers from the numerical instabilities for the same reason as mentioned above. Thus, the numerical treatment of the linear advection–diffusion equation is still the subject of extensive research, see e.g. [57, 60, 58, 2].

In case of the finite difference or finite volume methods, one approach is to replace the central difference, which approximates the advection term, by forward or backward difference. The direction of characteristics decides if the forward or backward difference will be chosen. This leads to the so called upwind scheme. The upwind finite difference technique is one of the first stabilization approaches, which removed the oscillations. However, the accuracy remains poor, because too much numerical diffusion was added.

Another option of the stabilization is to use methods, which approximate solution in a special way such that the numerical oscillations are suppressed, e.g. discontinuous Galerkin method (DG) and also TVD methods. The DG method is widely used in practice, such as for solving compressible and incompressible flow with high Reynolds numbers, but also for the shallow water equations, turbomachinery or magneto–hydrodynamics, [23, 22, 5, 87]. The DG method combines features of finite element method and finite volume method. The solution is represented as a FEM approximation within each element and the advection terms are resolved similarly as in FVM. Thus, the DG method computes high order accurate solution and it benefits in its ability to handle with the complicated geometries and hanging nodes. Besides, the importance of DG method is appreciated in situations with steep gradients or shocks. Moreover, DGFEM can handle with the nonconforming meshes.

The DG method usually produces non–physical oscillations near the shocks if higher order approximation is used. Then a slope limiter has to be applied to ensure the stability of the scheme, [49]. Any TVD limiter is usually used, [45]. However, the high order accuracy of the scheme can be lost, especially in multi-dimensional spaces, for more details see [49, 21].

Nevertheless, we concentrate on the solution of the Navier–Stokes/RANS problem using the isogeometric analysis, which is based on the continuous Galerkin discretization. Thus, any of the previous stabilization approaches cannot be applied and we need to follow the techniques developed for the classical finite element method and extended for the isogeometric analysis.

It is well known that the continuous Galerkin discretization is, in simple settings, equivalent to the central finite difference approximation, which gives rise to spurious oscillations in advection dominated problems, see e.g. [16, 57, 58].

Various stabilization techniques have been developed to diminish the numerical difficulties that affect the Galerkin discretization. First approaches were employed in the one-dimensional case such that they imitated the idea of the upwind techniques used in finite differences (cf. [57, 58, 16]). Later, the upwind type stabilizations were derived for multi-dimensional problems. But the upwind techniques are known for their over-diffusive character, which gives typically the solution without the presence of the numerical oscillations but with poor accuracy. Moreover, the upwind methods are not consistent since only the advection term is modified and these methods are only first order accurate. Detailed discussion about the upwind techniques can be found e.g. in [57, 16].

The Streamline Upwind/Petrov–Galerkin (SUPG) method proposed in [16] has become the most popular stabilization method. It eliminates almost all the difficulties of the previous methods for the advection dominated problems. By the construction, the SUPG stabilization introduces numerical diffusion only along streamlines, such that the element residual of the equation is added to the discrete weak form and multiplied by a test function weighted by a suitable stabilization parameter. Note that the SUPG method is consistent since the weighted residual is added to the Galerkin discretization, which results in higher order accurate method.

The amount of the added numerical diffusion is not affected only by the parameters of the given stabilization method but also by the choice of the basis functions. Moreover, the spurious oscillations are usually observed in the vicinity of the sharp layers in the numerical solution computed with the SUPG method, especially in the multi-dimensional and nonlinear problems. The SUPG method is neither monotone nor monotonicity preserving (stated e.g. in [57]). From this follows that it cannot be guaranteed that the SUPG stabilization will be sufficient and thus the numerical oscillations may appear.

Also, the discrete maximum principle is very important property, which is not automatically guaranteed. On the other hand, the maximum principle holds for the continuous steady advection-diffusion problem assuming certain requirements for the boundary condition and the source term.

Only some of the stabilization methods satisfying the discrete maximum principle were constructed, but usually under limited assumptions, see e.g. [24]. Thus, the monotone method with high order accuracy or method that guarantees the discrete maximum principle is very difficult to develop. The stabilization methods must introduce sufficient artificial diffusion in the sharp layers and the additional diffusion must quickly vanish in the regions where the solution is smooth or where the advection is small. According to the literature, only a nonlinear method can avoid the numerical oscillations in the vicinity of the sharp layers (i.e. add sufficient artificial diffusion) and give a high order accuracy in the regions where the solution is smooth, [24].

Numerous additional stabilization methods have been introduced to suppress the remaining spurious oscillation of the SUPG solution. These techniques are called discontinuity capturing, shock capturing or the most commonly, spurious oscillations at layers diminishing (SOLD) methods, see e.g. [59, 60, 57]. Some of them only eliminate the numerical oscillations, but other are developed such that the resulting method is monotone or guarantees a discrete maximum principle.

Additional stabilization terms are added to the SUPG discrete form using the SOLD methods. Most of the SOLD methods are nonlinear and consistent since the additional terms usually depend on the unknown numerical solution in a nonlinear way and on the discrete element residual. Some of the SOLD methods provide satisfactory results such that they reduce the numerical oscillations with a sufficiently high accuracy. On the other hand, some SOLD methods add too much numerical diffusion and hence the attained accuracy of the methods is poor.

Although many SOLD methods have been developed, those of them that guarantee the discrete maximum principle or preserve monotonicity are usually constructed for limited application, e.g. for conforming linear triangular finite elements. An example of such a method is the Mizukami-Hughes method, [59], which is a monotone method of first order and satisfies the discrete maximum principle with high accuracy. But so far, it is not clear how to generalize this method.

Other methods have been introduced which guarantee the discrete maximum principle. For example, a nonlinear additional SOLD method is studied in [18], where sufficient conditions are derived to satisfy the discrete maximum principle, but only with application to linear elements on strictly acute triangulations.

The most published and popular SOLD method in the literature is the crosswind artificial diffusion (cf. e.g. [58, 59, 18, 28, 24]). As mentioned above, the SUPG method introduces numerical diffusion only in the direction of the streamlines usually with sufficient accuracy. Thus, an additional diffusion added only in the crosswind direction is considered as an appropriate technique. Another popular method is the isotropic method which however adds numerical diffusion in all directions and hence to much numerical diffusion is usually added.

The main focus of the current research is on improvements of the SOLD methods, which consists in the control of the amount of the added numerical diffusion. This is usually influenced by the choice of a stabilization parameter, which is contained in the stabilization terms. But the formulas for the parameters are not optimal for general problem and hence they are usually constructed for a particular discrete problem with an unclear way for generalization, e.g. the crosswind diffusion developed for the linear elements on triangulations, [59, 18].

Another SOLD method is the edge stabilization, [19, 59, 60], which adds a term dependent on the jump of the solution gradient over the element boundaries. Some of the improvements of the edge stabilization include the element residual dependency in addition. Nevertheless, the drawback of the method is that it is defined for the linear simplicial finite elements.

The SOLD stabilization techniques are usually introduced for a simple steady advection–diffusion problem with constant coefficients, [57, 58, 50, 59, 60, 63, 2]. There are also papers where the unsteady linear advection dominated problems are studied, [24, 107]. Fewer number of the literature is focused on steady or unsteady advection–diffusion–reaction equation with a source term, [18, 62, 76] and only a few papers can be found which employ the stabilization methods for the nonlinear problems, [28, 107]. But the study of this class of the SOLD methods is rarely seen for the incompressible turbulent flow problems simulated by solving RANS equations, [92]. Note that almost all of the mentioned literature apply the finite element method using linear basis functions and only some of them compare higher degree (quadratic usually) elements. Moreover, there are lots of stabilization methods which are defined only on the linear elements as mentioned above. Hence, we are limited in selection of the stabilization methods since the isogeometric analysis with higher order of the spatial discretization on multipach domain with locally refined meshes is used in this thesis.

An alternative method to SUPG is the Galerkin/Least–squares (GLS) method which extends the idea of the SUPG method by appending element residual in least-squares form to the Galerkin formulation, [59, 38, 75]. The additional terms to the Galerkin form enhance the stability without degrading accuracy. For both SUPG and GLS methods, the second derivatives are required, which are not neglected since the test functions are considered of higher degree using isogeometric analysis. The GLS method can be found to be employed for the advection–diffusion problem, also for the incompressible flow, but usually for the Stokes problem. Nevertheless, the application to the incompressible/compressible Navier–Stokes equations can be found, too. A drawback of the GLS method is the number of additional terms, which have to be evaluated in each iteration.

There are even more stabilization methods which can be applied to improve the stability of the numerical solution (e.g. residual-free bubble finite element methods, local projection stabilization schemes and continuous interior penalty method, variational multiscale method) and they are discussed e.g. in [2]. But all the mentioned techniques are based on modification of the classical Galerkin formulation by adding some stabilization terms.

Besides these techniques, there are other approaches, which are constructed on the algebraic level. The most popular is the predictor-corrector flux-corrected transport (FCT) method, which is a high order method introduced in [72].

The idea of the FCT algorithm is as follows. First, a low order scheme is employed to incorporate enough numerical diffusion such that all the undershoots and overshoots are suppressed in the whole computational domain. This step enforces the positivity of the resulting solution. It is based on an analysis of matrix properties which has to be satisfied to preserve positivity. Algebraic flux correction (AFC) is a standard low-order method used in the FCT algorithm, [68, 86, 55]. Second step is to apply a limited anti-diffusive fluxes to the solution obtained by the low-order method, which results in the high-order solution such that no unwanted spurious oscillations occur.

Another high-resolution FCT scheme is based on the so called blending of the low- and high-order flux approximation using a weighting factor (proposed in [122]). The high-order scheme is used in the regions where the solution is smooth and the low-order scheme is applied along the sharp layers.

Compared to the above mentioned stabilization methods, the predictor-corrector FCT scheme does not include any stabilization parameter, which is challenging to set for a particular advection dominated problem. On the other hand, the anti-diffusive flux dependent on the unknown solution is the most common choice of the FCT algorithm, which leads to high computational cost. In order to reduce the computational requirements, various linearization techniques have been developed. The reader is referred e.g. to [72, 73, 78, 70, 69] for more details.

FCT techniques are mostly restricted to the linear finite elements. There exist extensions to higher order finite elements, but its implementation is very hard due to negative function values of the finite element bases. It leads to negative entries of the mass matrix and hence the discrete maximum principle cannot be guaranteed. Isogeometric analysis is not affected by this problem since B-splines are all non-negative and form the partition of unity. It makes the isogemetric analysis a perfect tool for extending the FCT framework to higher order functions, [78, 64].

The Babuška–Brezzi condition is presented in the following section where the suggestions how to satisfy LBB condition are given. Also, SUPG and SOLD stabilizations are formulated for the isogeometric analysis and then employed in the numerical experiments in Section 5. We also implemented AFC method into our incompressible solver. But the resulting solution is extremely over-diffusive as only low-order AFC method was applied. Thus, the AFC results are not compared in the numerical experiments with the SOLD methods in Section 5. We tackle FCT techniques in the future work.

4.3.1 LBB (Ladyženskaja–Babuška–Brezzi) condition

Ladyženskaja–Babuška–Brezzi condition is a sufficient condition for the discrete saddle point problem to have a unique solution that depends continuously on the input data, i.e. well-posedness problem. According to the Galerkin discretization mentioned in the previous section, we assume V^h to be the discrete velocity space and W^h the discrete pressure space. Then, the LBB condition states that there is a constant $\gamma > 0$ which is not dependent on the mesh such that, [15, 89, 116, 36, 35],

$$\inf_{q\in W^h} \sup_{\mathbf{u}\in V^h} \frac{\int\limits_\Omega q\nabla\cdot\mathbf{u}}{\|\mathbf{u}\|_{H^1(\Omega)}\,\|q\|_{L^2(\Omega)}} \geq \gamma. \tag{106}$$

It is well known that many discretizations of the saddle point problems are unstable which leads to the spurious oscillations of the velocity and primarily pressure approximations (see the spurious checkerboard pressure mode [116]). But only few combinations of discretization satisfy the LBB condition. In practice, it is common to apply an extension of the classical Taylor–Hood finite elements (see e.g. [116, 36, 35]) for the isogeometric analysis to satisfy the inf–sup condition (106), i.e., to choose $(p+1)$-degree of the B-spline basis functions for the velocity approximation and p-degree of the B-spline basis functions for the pressure approximation. The reader is referred e.g. to [15, 89] for more details, discussion and basic theory on the saddle point problems and their numerical analysis.

In the case of classical finite element method, second order approximation for velocity and first order approximation for pressure is the most commonly used combination satisfying (106), see [35, 104, 17, 12]. In case of finite volume method, LBB condition is satisfied by different discretization. Specifically, the scalar pressure quantity is located in the vertex of the mesh but the vector velocity quantity is located in the center of sides.

To avoid this restriction of the discretization, the additional stabilization can be implemented. This can be ensured by the mentioned GLS stabilization method which suppress spurious oscillations caused by both, not satisfied LBB condition and dominating advective term, [38]. This makes it possible to use equal order approximations for the velocity and pressure. If the SUPG method is rather used, the additional Pressure Stabilized/Petrov–Galerkin (PSPG) method (cf. e.g. [110, 12]) is recommended to employ, because the SUPG method does not circumvent LBB condition. There are variety of approaches discussed e.g. in [74, 91, 116, 36]. Nevertheless, the unequal degree of the basis functions is chosen in this thesis.

4.3.2 SUPG stabilization

This and the following subsections are devoted to the stabilization techniques, which suppress the nonphysical spurious oscillations caused by the advection/reaction dominated character of the equation. In our applications, we have to deal with advection dominated system of Navier–Stokes equations, advection/reaction dominated turbulence model and if needed also advection dominated system of RANS equations. Because of different formulations of the mentioned problems, the stabilization methods are introduced on a general time–dependent advection–diffusion–reaction (ADR) equation with a source term and general non-constant coefficients. The application to the specific problem is then analogous.

The unsteady ADR problem is considered in the form

$$\begin{aligned}
\frac{\partial \phi}{\partial t} + \mathbf{b}(\phi,\mathbf{x},t)\cdot\nabla\phi - \nabla\cdot[D(\phi,\mathbf{x},t)\nabla\phi] + r(\phi,\mathbf{x},t)\phi &= f(\phi,\mathbf{x},t) \quad \text{in } \Omega\times(0,T),\\
\phi(\mathbf{x},t) &= g_D(\mathbf{x}) \quad \text{on } \partial\Omega_D\times[0,T],\\
\mathbf{n}\cdot D(\phi,\mathbf{x},t)\nabla\phi(\mathbf{x},t) &= g_N(\mathbf{x}) \quad \text{on } \partial\Omega_N\times[0,T],\\
\phi(\mathbf{x},0) &= \phi_0(\mathbf{x}) \quad \mathbf{x}\in\Omega
\end{aligned} \tag{107}$$

where $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$, $\mathbf{b} = \mathbf{b}(\phi,\mathbf{x},t)$ is an advection coefficient satisfying the condition $\nabla\cdot\mathbf{b} = 0$, $D = D(\phi,\mathbf{x},t) > 0$, $\forall(\mathbf{x},t)\in\Omega\times[0,T]$ is a diffusion coefficient, $r = r(\phi,\mathbf{x},t)$ is a reaction coefficient and $f = f(\phi,\mathbf{x},t)$ is a source term. Note that the advection coefficient is $\mathbf{b} = \mathbf{u}$ for all the mentioned problems.

The semi-discrete formulation of ADR problem (107) is determined as in Section 4.1, which is to find $\phi^{n+1}\in H^1(\Omega)$ for all $n = 0,\dots,N-1$, where $\phi^N = \phi(t_N) = \phi(T)$, s.t. for all $\psi \in H_0^1(\Omega)$ it holds

$$\int\limits_\Omega \frac{\phi^{n+1}-\phi^n}{\Delta t}\psi + \int\limits_\Omega (\mathbf{b}(\phi^{n+1},\mathbf{x},t^{n+1})\cdot\nabla\phi^{n+1})\psi + \int\limits_\Omega D(\phi^{n+1},\mathbf{x},t^{n+1})\nabla\phi^{n+1}\cdot\nabla\psi$$
$$+\int\limits_\Omega r(\phi^{n+1},\mathbf{x},t^{n+1})\phi^{n+1}\psi = \int\limits_{\partial\Omega_N} g_N(\mathbf{x})\psi + \int\limits_\Omega f(\phi^{n+1},\mathbf{x},t^{n+1})\psi. \tag{108}$$

For simplicity, the zero Neumann condition is assumed in the following, i.e., $g_N = 0$. Moreover, the natural outflow condition is cosidered in our numerical experiments, for which the zero Neumann condition is used. The procedure is similar to spatial discretization of the turbulence model, i.e., the Galerkin discretization is applied and B-splines are used as the basis functions of the corresponding finite dimensional search $V^h \subset H^1(\Omega)$ and test spaces $V_0^h \subset H_0^1(\Omega)$. A Galerkin problem arising from (108) reads: find $\phi_h^{n+1} \in V^h$ such that

$$\frac{1}{\Delta t}\int\limits_\Omega \phi_h^{n+1}\psi_h + \int\limits_\Omega (\mathbf{b}(\phi_h^{n+1},\mathbf{x},t^{n+1})\cdot\nabla\phi_h^{n+1})\psi_h + \int\limits_\Omega D(\phi_h^{n+1},\mathbf{x},t^{n+1})\nabla\phi_h^{n+1}\cdot\nabla\psi_h +$$
$$+\int\limits_\Omega r(\phi_h^{n+1},\mathbf{x},t^{n+1})\phi_h^{n+1}\psi_h = \int\limits_\Omega f(\phi_h^{n+1},\mathbf{x},t^{n+1})\psi_h + \frac{1}{\Delta t}\int\limits_\Omega \phi_h^n\psi_h \tag{109}$$

for all $\psi_h \in V_0^h$. Note that the resulting reaction term is

$$\int\limits_\Omega \left(\frac{1}{\Delta t} + r(\phi_h^{n+1},\mathbf{x},t^{n+1})\right)\phi_h^{n+1}\psi_h. \tag{110}$$

Using the same procedure, the fractional step θ-method can be derived. This temporal discretization can be found e.g. in [62]. But only the backward Euler method is employed in this thesis, hence the general θ-scheme is not mentioned here for simplicity. The linearization strategy is analogous to the turbulence problem such that the Galerkin problem (109) is solved iteratively, however, the formulation is not mentioned here for simplicity.

To construct the SUPG method, the weighted element residual is added to the Galerkin problem (109), i.e.,

$$\sum_{i_e=1}^{n_{el}}\int\limits_{Q_{i_e}} \tau_S R(\phi_h^{n+1},\mathbf{x},t^{n+1})\mathbf{b}(\phi_h^{n+1},\mathbf{x},t^{n+1})\cdot\nabla\psi_h, \tag{111}$$

where

$$\begin{aligned} R(\phi_h^{n+1}) &= \frac{\phi_h^{n+1}-\phi_h^n}{\Delta t} + \mathbf{b}(\phi_h^{n+1},\mathbf{x},t^{n+1})\cdot\nabla\phi_h^{n+1} - \nabla\cdot[D(\phi_h^{n+1},\mathbf{x},t^{n+1})\nabla\phi_h^{n+1}] + \\ &\quad + r(\phi_h^{n+1},\mathbf{x},t^{n+1})\phi_h^{n+1} - f_h(\phi_h^{n+1},\mathbf{x},t^{n+1}) \end{aligned} \tag{112}$$

is the element residual, n_{el} is the number of elements Q_{i_e}, τ_S is a suitably chosen nonnegative SUPG stabilization parameter, which is explained below. But the functions from V^h and V_0^h have to satisfy different assumptions such that the second derivative exists. Thus, the functions from the subspaces V^h and V_0^h are assumed to be infinitely smooth inside each element, which is satisfied using the B-spline basis functions in our applications. Alternatively, the subspaces can be extended for the functions from $H^2(\Omega)$. For classical linear finite elements and B-splines of the first order, the second derivative is zero within each element. However, this term cannot be neglected if higher order basis functions are used, which are typically used in isogeometric analysis.

Note that the SUPG method is consistent as the SUPG term (111) is in the form of weighted residual, i.e., the solution of (107) is a solution of the variational problem.

The choice of the stabilization parameter τ_S is a challenging part of the SUPG method. Most of the literature deals with the linear steady advection–diffusion equation, thus the stabilization parameters are only advection dependent or advection and diffusion dependent. Fewer number of papers introduces the stabilization parameter which takes the reaction into account, [18, 62, 76, 28, 107, 26]. There are some texts dealing with higher order basis functions, see e.g. [76, 24, 50, 107]. But the formulas for the stabilization parameter are not optimal for general problem. Often, they are constructed for a particular discrete problem with an unclear way for generalization (cf. e.g. [59, 18]).

We compare four variants of the SUPG stabilization parameter introduced e.g. in [59, 60, 62, 26, 24]. First variant was proposed for one-dimensional advection–diffusion problem with constant source term using piecewise linear finite elements on uniformly refined mesh

$$\tau_S^0 = \frac{h}{2\,\|\mathbf{b}\|}\xi(Pe) \quad \text{with } \xi(Pe) = \coth Pe - \frac{1}{Pe}, \tag{113}$$

where h is a characteristic size of the element and Pe is the local Péclet number

$$Pe = \frac{\|\mathbf{b}\|\,h}{2D}. \tag{114}$$

The parameter τ_S^0 was shown to give nodally exact solution. A detail discussion about the parameter τ_S^0 can be found e.g. in [59, 76, 26, 24]. A simplification of this parameter is

$$\tau_S^1 = \frac{h}{2\,\|\mathbf{b}\|}. \tag{115}$$

It is well known that parameter τ_S^1 adds too much numerical diffusion and that the function ξ in (113) is the so called upwind function, which reduces the amount of the additional numerical diffusion in the smooth regions of the computational domain. Nevertheless, both parameters are compared in the numerical tests since we use the isogeometric analysis. Only few studies of stabilization techniques exist using IgA on single patches or multipatches with regular mesh (e.g. [76, 107, 50]) and to our best knowledge there is only one paper solving the turbulent incompressible flow on multipatches with local refinement for high Reynolds numbers, [92].

If the diffusion coefficient is constant, it is sufficient to apply the previous stabilization parameters for advection dominated problem (Navier–Stokes problem with $D = \nu$). However, if the diffusion coefficient is not constant (RANS equations with $D = \nu + \nu_T$ and k, resp. ω, equation of the SST turbulence model (70) with $D = \sigma_k \nu_T + \nu$, resp. $D = \sigma_\omega \nu_T + \nu$), the following stabilization parameter is recommended (cf. e.g. [107, 28])

$$\tau_S^2 = \left(\left(\frac{2\,\|\mathbf{b}\|}{h} \right)^2 + 9 \left(\frac{4D}{h^2} \right)^2 \right)^{-\frac{1}{2}}. \tag{116}$$

In the case of unsteady problem, we can apply ([107])

$$\tau_S^3 = \left(\left(\frac{2\,\|\mathbf{b}\|}{h} \right)^2 + 9 \left(\frac{4D}{h^2} \right)^2 + \left(\frac{2}{\Delta t} \right)^2 \right)^{-\frac{1}{2}} \tag{117}$$

and if the reaction r is nonzero in (107), the parameter should take the reaction into account, i.e.,

$$\tau_S^4 = \left(\left(\frac{2\,\|\mathbf{b}\|}{h} \right)^2 + 9 \left(\frac{4D}{h^2} \right)^2 + \left(\frac{2}{\Delta t} \right)^2 + \|r\|^2 \right)^{-\frac{1}{2}}. \tag{118}$$

The parameters τ_S^i, $i = 0, 1, \ldots, 4$ are generally functions of the evaluation points in the element.

The element length computation is another key component of the stabilization methods. A simple choice is to use the length of the diagonal or maximum size length of the element. Usually, it is recommended to apply the element size in the advection direction in advection dominated case, [59, 107]. If the problem is also reaction dominated such that reaction even dominates advection in some parts of the domain, the appropriate choice of the element length h is not clear, see e.g. [62]. Because the reaction can be dominant in solving turbulence model, the length of element diagonal is used in our computations similarly to [76]. The alternatives of h computation for isogeometric analysis will be studied in the future work.

For higher order B-splines, the following modification of the stabilization parameter τ_S^0 can be used instead

$$\tau_S^{0,deg} = \frac{h}{2deg\,\|\mathbf{b}\|} \xi(Pe), \tag{119}$$

where deg is the degree of the basis functions. The modification of the rest of the stabilization parameters $\tau_S^{1,deg}$, $\tau_S^{2,deg}$, $\tau_S^{3,deg}$ and $\tau_S^{4,deg}$ is straightforward.

As already mentioned, the SUPG stabilization adds numerical diffusion acting only in the flow direction. Moreover, it is well known that the solution computed with the SUPG formulation does not preclude the presence of spurious oscillations of small magnitudes in the vicinity of the sharp gradients. The small overshoots and undershoots resulting from the SUPG method are caused by the fact that SUPG method is neither monotone nor monotonicity preserving method. In the context of the incompressible flow applications, the sharp gradients are inseparable part of the numerical simulation. Thus, it is usually necessary to modify the SUPG formulation by adding additional stabilization terms, which results in SOLD methods. In the following, we introduce additional crosswind diffusion and isotropic artificial diffusion methods. These methods can be also employed as individual stabilizations without the SUPG formulation (see e.g. [18]), which is not a common approach. First, the inconsistent SUPG method is formulated.

4.3.3 Inconsistent SUPG stabilization

Usually, the B-spline basis functions of higher degree are chosen to approximate the numerical solution using IgA and hence the second derivatives in (112) are not zero in any element as well as the pressure gradient in the Navier–Stokes equations. It follows that the whole residual of the discrete problem has to be evaluated in (111). Thus, the inconsistent SUPG stabilization (denoted by I-SUPG in this thesis) is now formulated and it is then compared in the numerical experiments. Employing the I-SUPG stabilization, the following artificial diffusion term is added to the formulation of a discrete problem, i.e.,

$$\sum_{i_e=1}^{n_{el}} \int\limits_{Q_{i_e}} \tau_S(\mathbf{b}(\phi_h^{n+1}, \mathbf{x}, t^{n+1}) \cdot \nabla \phi_h^{n+1})(\mathbf{b}(\phi_h^{n+1}, \mathbf{x}, t^{n+1}) \cdot \nabla \psi_h) \tag{120}$$

where τ_S is considered as any of the SUPG parameters (113) – (119) in this thesis. However, this type of I-SUPG stabilization leads usually to over-diffusive solution. On the other hand, all the numerical oscillations are typically eliminated using (120).

It can be beneficial to consider the analogy to the classical finite elements applied to the advection–diffusion equation or to Navier–Stokes equations, where the second order and first order basis functions are usually used for the velocity and pressure approximations in spatial discretization. Thus, it leads to zero diffusion and pressure terms and nonzero advection and time derivative terms. In this analogy, we can apply an inconsistent SUPG stabilization with the time derivative term in the form

$$\sum_{i_e=1}^{n_{el}} \int\limits_{Q_{i_e}} \tau_S \left(\frac{\phi_h^{n+1} - \phi_h^n}{\Delta t} + \mathbf{b}(\phi_h^{n+1}, \mathbf{x}, t^{n+1}) \cdot \nabla \phi_h^{n+1} \right) (\mathbf{b}(\phi_h^{n+1}, \mathbf{x}, t^{n+1}) \cdot \nabla \psi_h) \tag{121}$$

and denoted by IT-SUPG in the experiments. Although, this formulation is inconsistent (i.e. the solution of (107) is no longer a solution of the variational problem), we have good experience with the latter (121). Moreover, the research is still devoted to the inconsistent approaches and the most common local projection stabilization (LPS) is applied e.g. in [62, 3] for advection–diffusion/advection–diffusion–reaction equation, and it is also employed for the incompressible flow e.g. in [1].

4.3.4 Crosswind stabilization

As mentioned above, the SUPG stabilization can be assumed to add numerical diffusion in the streamline direction and the crosswind diffusion is employed to reduce the spurious oscillations in the crosswind direction. The crosswind diffusion term is constructed such that the resulting stabilization is consistent, i.e. the additional diffusion is proportional to the element residual, moreover, it must vanish in the regions where the advection is small and the solution is smooth to keep the accuracy.

The additional crosswind stabilizing term is written as (cf. e.g. [59, 58])

$$\sum_{i_e=1}^{n_{el}} \int_{Q_{i_e}} \tau_{cw} P^{\perp} \nabla \phi_h^{n+1} \cdot \nabla \psi_h, \tag{122}$$

where τ_{cw} is the crosswind stabilization parameter, which represents the relative amount of the crosswind diffusion added, and $P^{\perp}$ is an orthogonal projector defined by

$$P^{\perp} = \mathbf{I} - \frac{\mathbf{b} \otimes \mathbf{b}}{\|\mathbf{b}\|^2}. \tag{123}$$

If the magnitude of the advection coefficient equals zero, then $P^{\perp} = \mathbf{0}$. The form of the stabilization parameter τ_{cw} is very difficult to define and is derived similarly to SUPG stabilization parameter. Many formulas for τ_{cw} have been proposed, but the following variant [58, 59] is used in our numerical experiments

$$\tau_{cw} = \frac{\tau_S \|\mathbf{b}\|^2 \|R(\phi_h)\|}{\|\mathbf{b}\| \|\nabla \phi_h\| + \|R(\phi_h)\|}, \tag{124}$$

where τ_S is one of the SUPG parameters (113) – (119).

4.3.5 Isotropic stabilization

Compared to the crosswind, isotropic diffusion is added in all directions. To find the proper amount of the additional artificial diffusion is even more difficult problem for isotropic diffusion if it is combined with SUPG method, since numerical diffusion in the streamline direction was already added from the SUPG part. Thus, isotropic diffusion can be over-diffusive with an inappropriate stabilization parameter.

The isotropic stabilization term is written as

$$\sum_{i_e=1}^{n_{el}} \int_{Q_{i_e}} \tau_{iso} \nabla \phi_h^{n+1} \cdot \nabla \psi_h, \tag{125}$$

where the two variants of the stabilization parameter τ_{iso} (cf. e.g. [58, 59, 62]) are considered

$$\tau_{iso}^1 = \max\big(0, \alpha h^\beta \|R(\phi_h)\| - D\big), \tag{126}$$

with constants α and β specified in the numerical experiments and

$$\tau_{iso}^2 = \frac{h \|R(\phi_h)\|}{2 \|\nabla \phi_h\|}. \tag{127}$$

The nonlinear terms arised from the additional isotropic or crosswind stabilization are treated together with the rest of the problem formulation, since the Navier–Stokes problem, RANS problem and turbulence models are systems of the nonlinear equations. Thus, the Picard's method (introduced in Section 4.1.2) is employed for the whole stabilized form.

5 Numerical experiments

In this section, we present results of some numerical experiments. The fluid flow simulation is obtained from an in-house isogeometric incompressible flow solver implemented in C++ within a framework of the G+Smo library. G+Smo is an open-source object-oriented templated C++ library based on abstract classes for geometry, discretization bases, assemblers, solvers etc. For more information about the library, see the documentation [81]. The linear algebra tools available in G+Smo are mostly inherited from the Eigen library [44].

We implemented the incompressible solver including solution of Stokes, Navier–Stokes and RANS problems in 2D and 3D. The possible choice of periodic boundary conditions or simulation of the fluid flow in rotating machines using rotating frame of reference is also included in the framework of our solver in the G+Smo library. In the case of RANS solver, two–equation turbulence models are implemented, namely Wilcox two–equation $k - \omega$ turbulence model (called LRN turbulence model in the experiments for simplicity) and SST turbulence model introduced in Section 3.2.3. The solver for the advection–diffusion–reaction equation with general non-constant coefficients is also integrated into the G+Smo library.

The numerical experiments were executed on a Linux computer (Ubuntu 16.04.1, 64-bit) with an Intel® Core™ i5-4310U processor @ 2.00GHz with four cores and 16GB of RAM.

First, we focus on laminar fluid flow and after we simulate the turbulent flow. However, the numerical simulation of the turbulent flow is very complex problem leading to a nonlinear advection dominated problem. We solve RANS equations closed with the system of the time-dependent advection–diffusion–reaction equations with the source (production) term to simulate the turbulent flow. A characteristic feature of solutions of this problem is

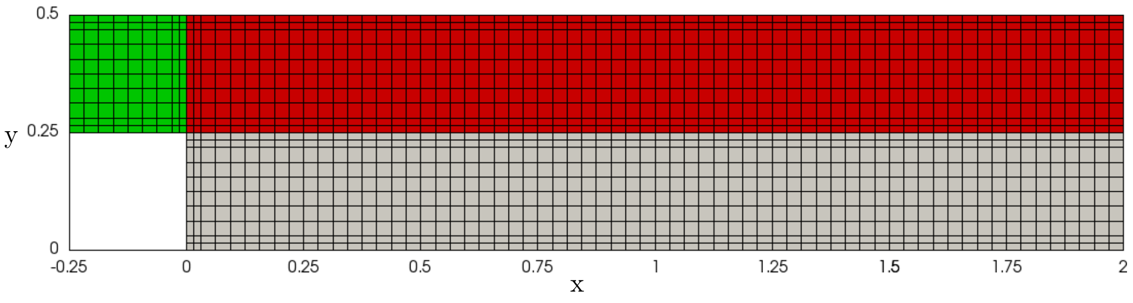


Figure 4: B-spline geometry representation of 2D backward facing step and an example of the computational mesh with 16032 DOFs.

the presence of sharp layers. Numerical methods are required which are able to resolve the sharp layers properly and also which prevent the spurious oscillations that are a typical negative side effect of the problem with small diffusion. Thus, before the RANS problem is solved, we focus on study of the numerical stability of the advection–diffusion (AD) such that we employ and compare the stabilization techniques introduced in Section 4.3. In our opinion, it is important to understand the numerical difficulties of the AD problem and then to apply observations to more complicated case as the solution of the RANS equations closed with turbulence model.

Running a flow simulation of the nonlinear Navier–Stokes, resp. RANS equations closed with turbulence model, means to solve the resulting linear system several times in each time step to let the Picard iterations converge. The stopping criterion of the Picard iterations reflects relative change of the solution between two consequent Picard iterations. The toleration of the relative error is set to 10^{-4}. The number of Picard iterations of the flow solver needed to converge increases with the decreasing value of the fluid viscosity. Typically, high number of the inner iterations is required during first time steps because of the significant changes of the initial condition. On the other hand, it usually decreases as the numerical solution converges to the steady state.

The convergence of the inner iterations of the turbulence model is usually very slow. The time step size can be decreased to reduce the number of the iterations. But higher number of time iterations has to be computed instead. A common approach in the turbulent flow computations is to add the maximum number of Picard iterations in one time step for the stopping condition. In our applications, we set maximum 5 Picard iterations for unsteady Navier–Stokes problem, at most 3 iterations for RANS nonlinear system and maximum 5 inner iterations for the turbulence model.

The system matrix is sparse, nonsymmetric and usually very large. The direct solver with LU decomposition is applied for solving linear systems for the time being.

The geometries in Figures 4 – 6 are the B-spline representations of the computational domains used in the experiments. An example of the computational mesh is also shown for all the geometries. We consider only conforming meshes in this thesis. In our experiments, we consider equal degree of the B-spline basis functions in all parameters as mentioned above. The degree is denoted by q and it follows from the degree of the B-splines used for geometry representation. But note that we elevate the degree of the basis functions

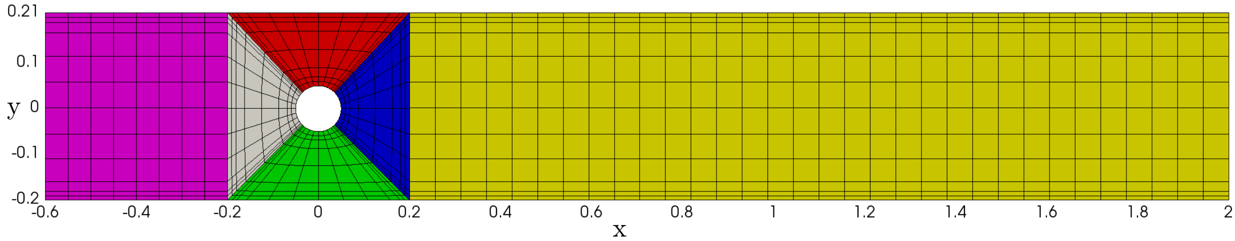


Figure 5: B-spline geometry representation of channel domain with an obstacle and an example of the computational mesh with 9062 DOFs.

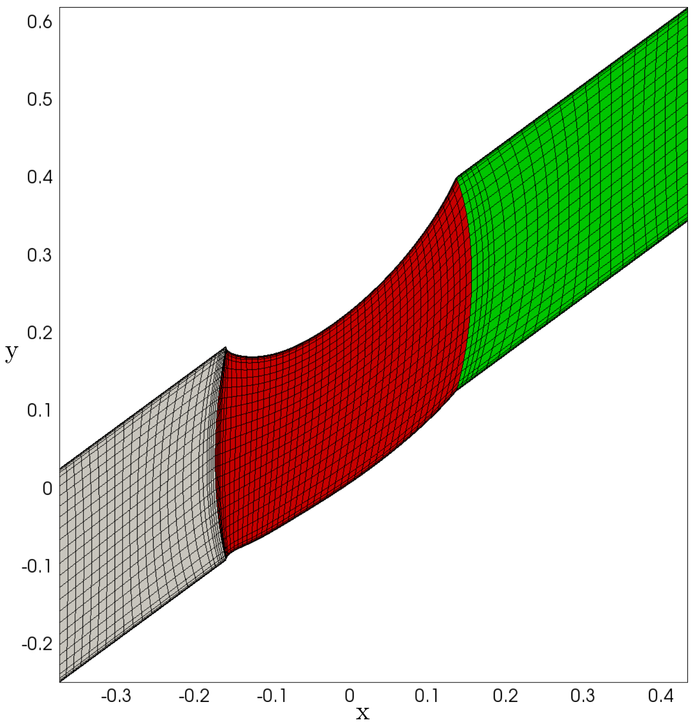


Figure 6: B-spline geometry representation of the blade cascade and an example of the computational mesh with 24349 DOFs.

for velocity by one to satisfy the LBB condition. By elevation, we assume that the degree for velocity approximation increase, but the continuity (regularity) remains the same as described in Section 4.2.

Figure 4 represents the geometry of the 2D backward facing step using the B-splines of degree $q = 2$. It consists of three patches such that the individual patches are described as B-spline surfaces. Uniform refinement with additional local refinement is considered in the Figure 4 resulting in 16032 degrees of freedom (DOFs) in total for velocity and pressure.

Local refinement is very important tool mostly for turbulent flow near the walls if we solve RANS problem closed with turbulence model without wall functions. In this thesis, we consider knot insertion by local refinement explained in Section 4.2 and displayed in Figures 4 – 6. However, in the context of the tensor product B-splines, insertion of a knot into one knot vector causes refinement spreading over the whole control grid, which thus can create a big amount of unnecessary control points. The requirement of conforming mesh causes local refinement also in unsuitable interior parts of the domain, usually along the interfaces, see Figures 4 and 5. Alternative refinement techniques can be applied, e.g. T-splines or truncated hierarchical B-splines (THB-splines), but it is a non-trivial task which we will study in the future.

The inlet Dirichlet condition is set on the left boundary at $x = -0.25$. The zero Neumann condition is set on the right boundary at $x = 2$ and homogeneous Dirichlet condition is considered at the rest of the boundaries.

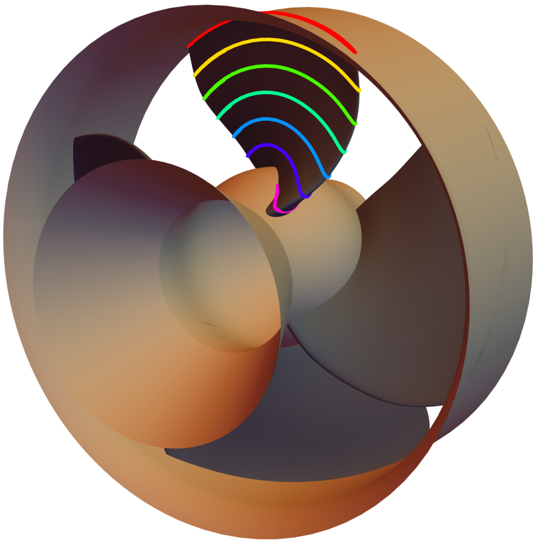


Figure 7: Runner wheel with seven highlighted cylindrical slices.

The fluid flow simulation around a circle obstacle is the second problem which we study. The B-spline geometry representation is shown in Figure 5. The geometry consists of six patches such that the degree of the B-spline basis is $q = 3$. Note that the B-spline representation is used for the circle obstacle, thus the description of the circle is not exact, but the approximation error is very low. The radius of the circle is 0.05 and the centre of the circle is $S = [0, 0]$. The width of the channel is 0.41. Note that the position of the centre of the obstacle is not in the half of the channel width, see Figure 5.

In this case, we show an example of mesh with 9062 DOFs that was generated using both uniform and local refinement. Inlet Dirichlet condition is set on the left boundary at $x = -0.6$. The zero Neumann condition is set on the right boundary at $x = 2$ and walls are considered at the rest of the boundaries (i.e. homogeneous Dirichlet condition for velocity. The boundary conditions for k and ω are explained in Section 5.4).

Blade cascade is the last 2D geometry, which we consider for numerical experiments of the fluid flow simulation, see Figure 6. The B-spline geometry representation consists of three conforming patches with B-spline basis of the degree $q = 3$. The middle (red) patch displayed in Figure 6 represents the part of the domain between two neighboring blade profiles. Blade cascade is constructed such that the upper and bottom boundaries of the first (grey) and third (green) patches are set to be periodic. Note that the periodic sides are parallel, hence we simply define an interface between two corresponding sides and treat the periodic condition in the same way as interface.

The remaining boundaries correspond to the inflow (left boundary) and outflow (right boundary). An example of the mesh can be also seen in Figure 6 with 24349 DOFs. Uniform and local refinement is used again. In this case, the local refinement along the interfaces is not necessary, however, the fluid flow simulation around the leading edge of the profile is very complicated as the velocity gradients are very steep in this area. Thus, it is beneficial to refine this problematic area in both directions to obtain the most regular elements rather than narrow long elements. A drawback is again the spreading of the refinement along the whole interface because of the mentioned tensor product B-splines.

The size of the channel (length and pitch) and the parameters describing the blade profiles are considered from the real data, i.e. the blade profile in Figure 6 is a chosen unfolded cylindrical slice of the runner blade of the Kaplan turbine shown in Figure 7.

The zero Dirichlet boundary condition for velocity is prescribed at the walls (blade profile) and zero Neumann condition is set at outflow boundary. The inlet velocity boundary condition is constant with the tangent direction to the leading edge of the blade profile and magnitude 8.1786, i.e. $\mathbf{u}_{in} = (7.765, -0.28272)$.

One more periodic part of the domain is displayed in Figure 8 in order to see the resulting unfolded slice of the runner blade. The results will be plotted in this periodically extended domain in the following sections.

The aim of this thesis is the fluid flow around the 2D blade profile in a channel employing the stabilization techniques to suppress the numerical oscillations without degrading accuracy. This kind of 2D computation could be used for a simplified turbine blade shape optimization, since the 3D blade geometry is built from several 2D profiles.

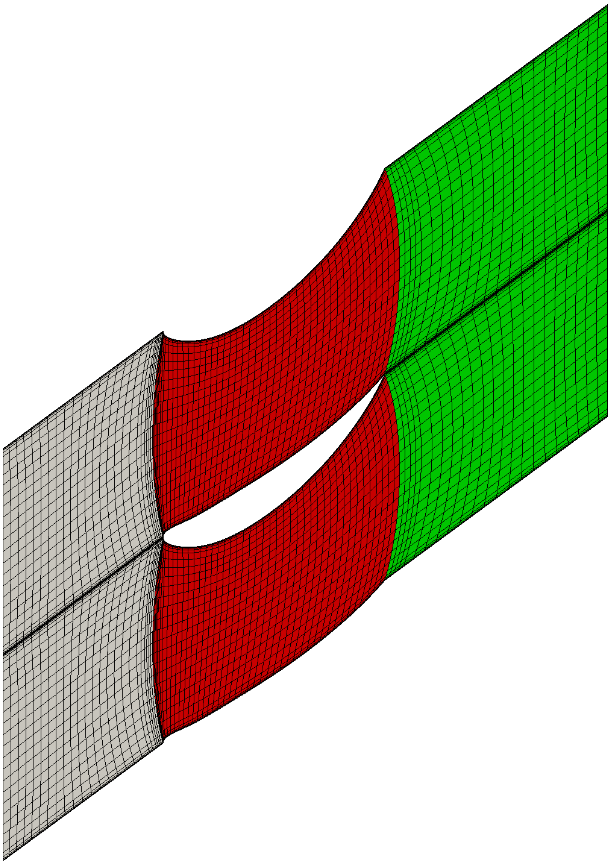


Figure 8: Periodically extended geometry in Figure 6.

The test cases are divided into three parts. First, the numerical solution of the steady and unsteady Navier–Stokes equations is presented, i.e. the laminar flow is simulated. Before the turbulent flow is simulated using RANS equations, the stabilization techniques are studied for the advection–diffusion equation.

5.1 Laminar flow

In this section, the laminar incompressible flow is simulated solving Navier–Stokes problem. In the case of laminar flow, diffusion term dominates advection and the stabilization techniques are not usually required. First example is the laminar flow in the backward facing step. We prescribe a parabolic velocity profile with the maximum of 1 at the inlet boundary, zero velocity on the walls and Neumann condition at the outlet as mentioned in the previous section. The converged numerical solution of the steady Navier–Stokes problem is displayed in Figure 9 for viscosity $\nu = 0.0015$. Note that the channel was lengthened to fit the whole bottom vortex into the computational area. The resulting problem has 20066 DOFs. Zero initial guess was chosen for the Picard iterations, which is equally set for all the experiments solving steady Navier–Stokes problem.

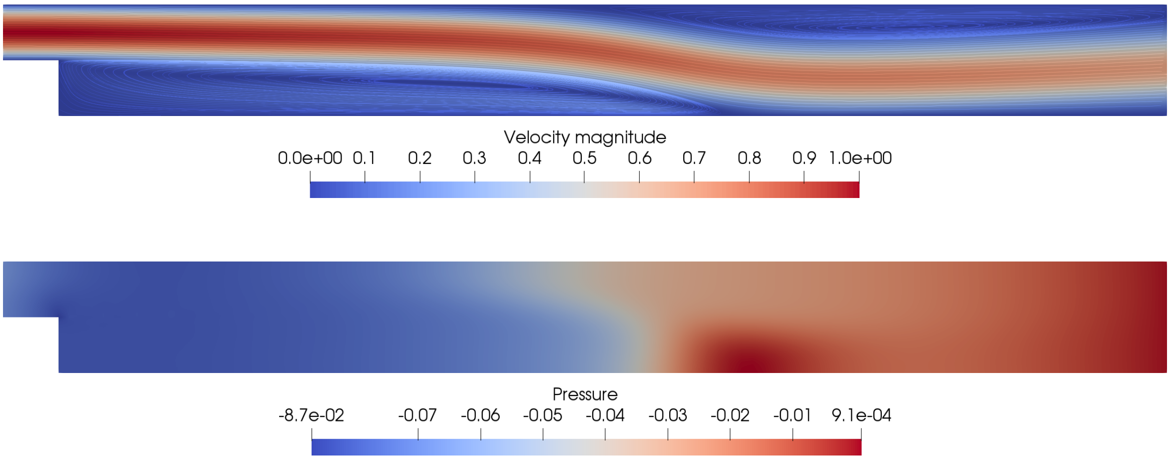


Figure 9: Streamlines and velocity magnitude (top) and pressure solution (bottom) of steady Navier–Stokes for backward facing step flow with viscosity $\nu = 0.0015$.

In the second experiment, we consider a steady flow around the blade profile with viscosity $\nu = 0.01$. The resulting velocity and pressure distributions are presented in Figure 10.

Figures 11 and 12 show the velocity and pressure solutions of the steady Navier–Stokes problem in channel with circle obstacle for $\nu = 0.02$ and $\nu = 0.002$. Parabolic velocity profile with maximum 1 is prescribed at the inlet boundary.

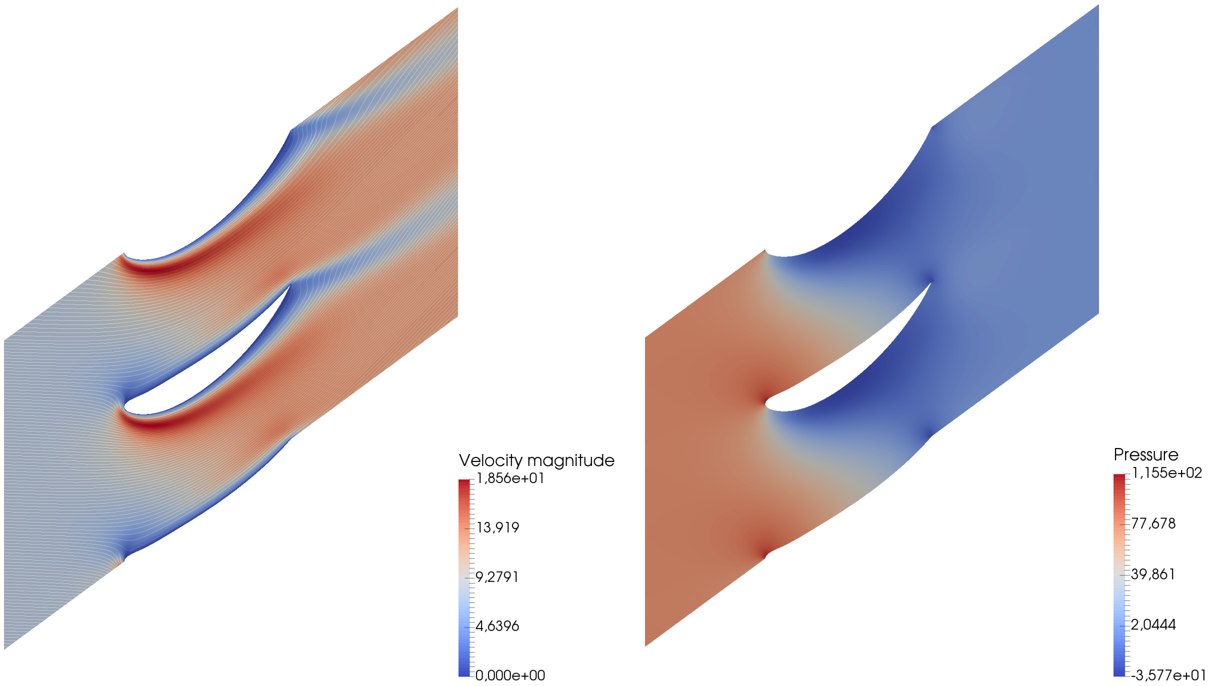


Figure 10: Streamlines and velocity magnitude (left) and pressure solution (right) of steady Navier–Stokes in blade cascade with viscosity $\nu = 0.01$.

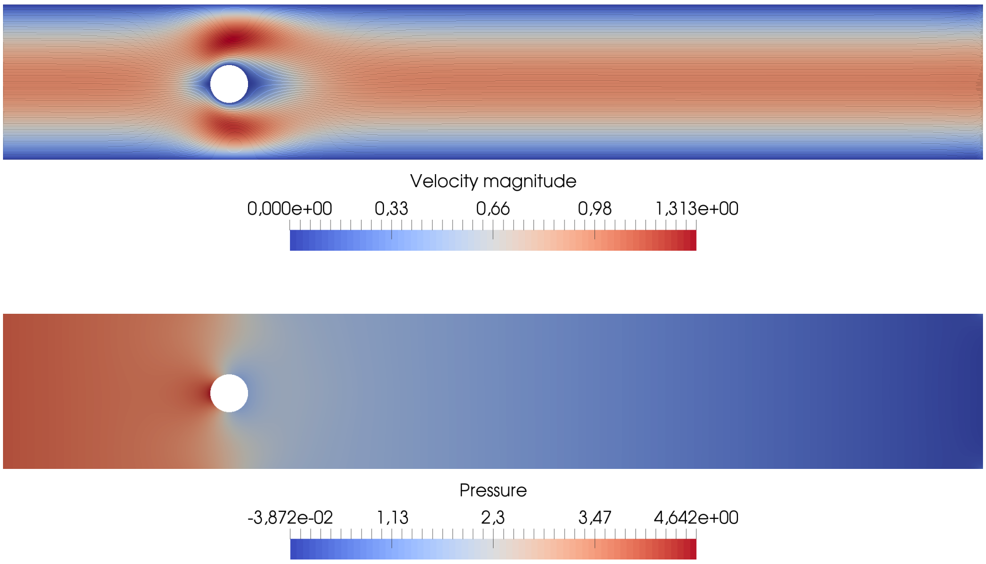


Figure 11: Streamlines and velocity magnitude (top) and pressure solution (bottom) of steady Navier–Stokes flow around circle obstacle with viscosity $\nu = 0.02$.

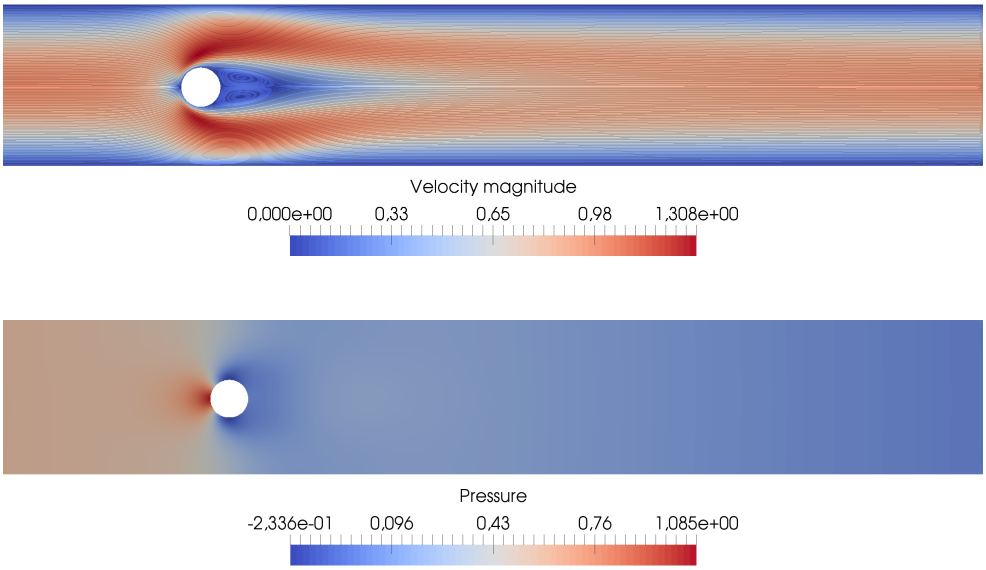


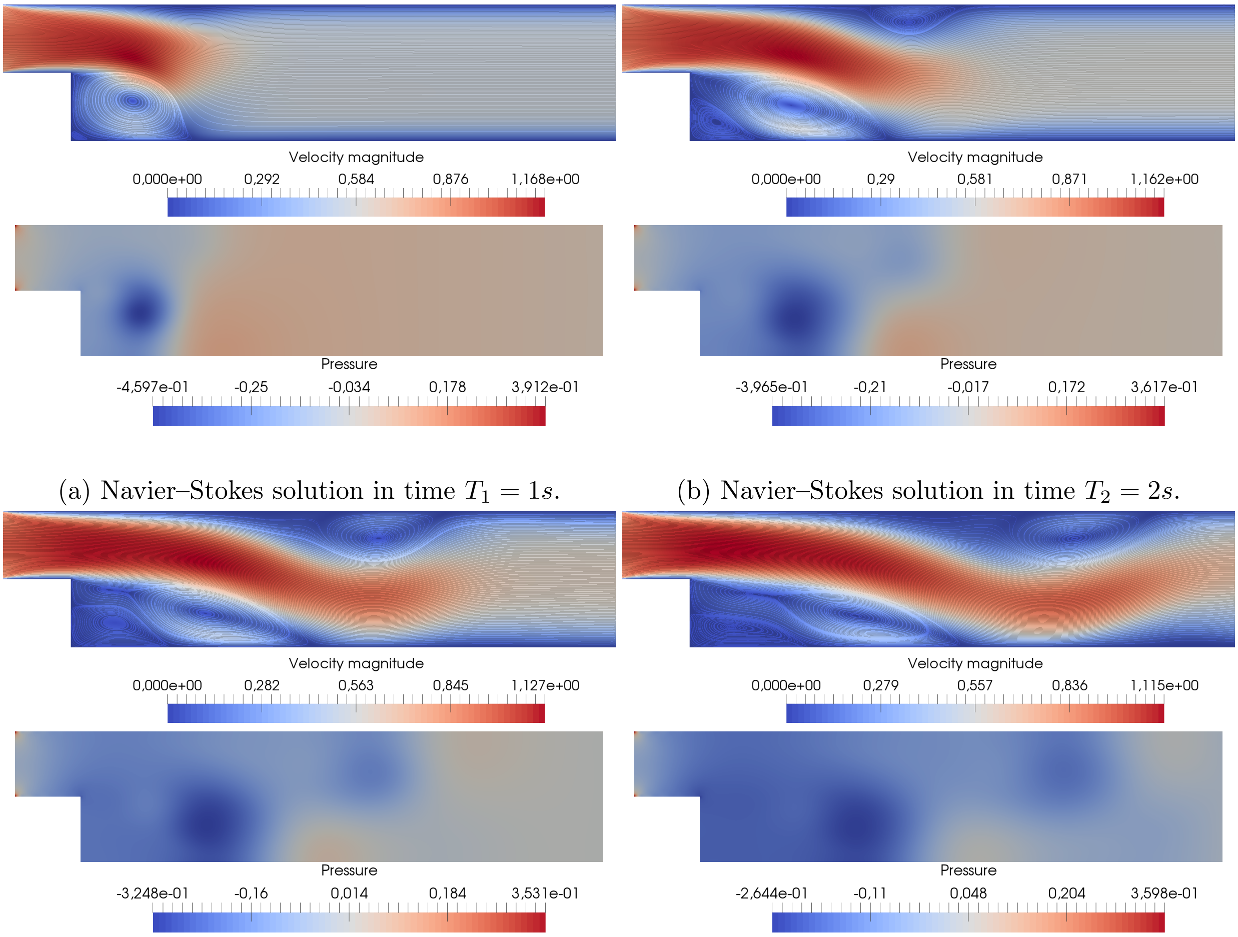
Figure 12: Streamlines and velocity magnitude (top) and pressure solution (bottom) of steady Navier–Stokes flow around circle obstacle with viscosity $\nu = 0.002$.

The unsteady Navier–Stokes equations are solved in the following experiments. In Figure 13 we can see velocity and pressure solutions of Navier–Stokes equations with viscosity $\nu = 10^{-3}$ and $\Delta t = 0.01$ in four different time moments $T_1 = 1s$, $T_2 = 2s$, $T_3 = 3s$, $T_4 = 4s$ for backward facing step. In the subsequent time steps, the vortices connect and start to stretch. The numerical solution then converges to the steady state similar to solution in Figure 9. In this experiment, different velocity profile was chosen at the inlet boundary. In case of decreasing viscosity, the parabolic velocity profile at inlet is not suitable. One approach is to extend the part of the channel in front of the step and set parabolic velocity profile at the inlet boundary. The solution is then developed in the extended channel. An alternative is to set the velocity profile obtained from the corresponding individual channel flow solved in advance. Another approach is to use a general formula, which approximates velocity profile in the channel for given viscosity. In [106], the velocity profile at the position $\bar{x}$ (i.e. the magnitude of the second velocity component $\mathbf{u} = (u_1, u_2)$ in 2D) is expressed as

$$u_2(\bar{x}, y) = u_{max} \left(1 - \frac{y}{\bar{y}}\right)^{\frac{1}{\bar{n}}}, \tag{128}$$

where u_{max} is maximal value of the velocity profile, $\bar{y}$ is half of the channel height and $\bar{n} = 1.03 \ln(Re) - 3.6$. This approximation of the velocity profile is used in the experiment, where the Reynolds number is computed $Re = \frac{0.25 u_{max}}{\nu}$ (the inlet channel height is 0.25 in this example).

The fluid viscosity was chosen sufficiently high in the previous examples such that the corresponding problems were not advection dominated or the computational mesh was fine enough. Hence, the resulting solutions were stable and no additional stabilization was



(a) Navier–Stokes solution in time $T_1 = 1s$.

(b) Navier–Stokes solution in time $T_2 = 2s$.

(c) Navier–Stokes solution in time $T_3 = 3s$.

(d) Navier–Stokes solution in time $T_4 = 4s$.

Figure 13: Time evolution of Navier–Stokes velocity and pressure solutions for $T_1 = 1s$, $T_2 = 2s$, $T_3 = 3s$, $T_4 = 4s$, $\nu = 10^{-3}$.

required. If we are interested in the turbulent flow simulation with small viscosity, the system of RANS equations is solved closed with the turbulence model since we are not able to resolve the Navier–Stokes equations on the corresponding fine mesh. As mentioned above, the turbulence model is a system of advection–diffusion–reaction equations, which has to be stabilized. A motivating results are shown in Figure 14, where the RANS velocity solution is compared for solving SST turbulence model without (left) and with (right) stabilization terms (SUPG method with τ_S^2) for $\nu = 10^{-5}$ and $\Delta t = 10^{-3}$. Obviously, the stabilization of the turbulence model affects significantly the RANS solution. Moreover, the time iterations of the RANS solutions diverge in later time moment when no stabilization scheme is used. Hence, the stabilization techniques are first studied for a simple advection diffusion equation in the next section and then the observations are applied to stabilization of the turbulence model.

Certainly, finer grid with not so narrow elements (especially close to the walls of the blade profile) could be used to suppress the numerical oscillations in some parts of the domain, but our aim is to study the stabilization techniques on meshes which are not regular, generally curvilinear and possibly insufficiently refined. The reason is to approach the flow simulation in 3D geometries, for which we are usually forced to solve the problems on coarser meshes because of the computational requirements.

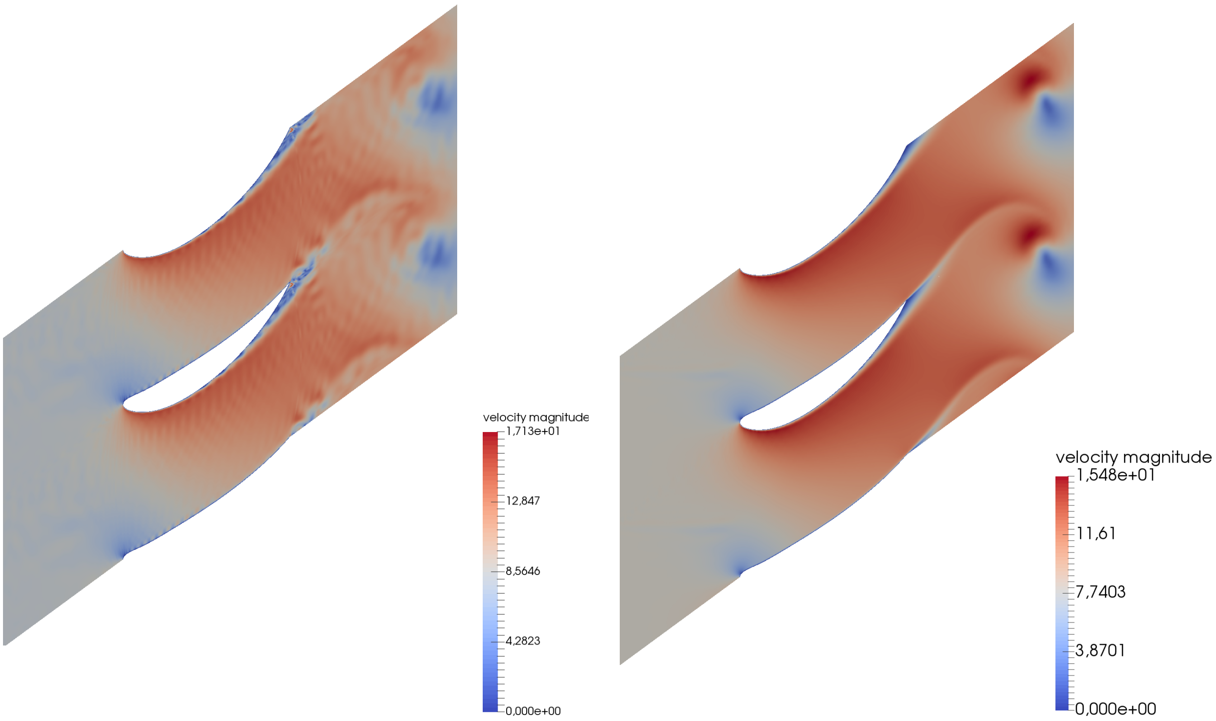


Figure 14: Velocity solution of RANS equations closed with SST turbulence model in blade cascade with viscosity $\nu = 10^{-5}$ in time $T = 0.03s$. SST without stabilization (left) SST with SUPG stabilization and τ_S^2 (right).

5.2 Advection–diffusion–reaction equation

As mentioned in the previous section, the numerical stability of the advection–diffusion equation (107) is studied before the turbulent flow is simulated solving RANS equations. The resulting systems of equations are also solved using LU direct solver.

5.2.1 Example 1

We start with an unsteady linear advection–diffusion initial boundary value problem with constant coefficients where

$$\mathbf{b} = (2, -3), \quad D = 10^{-5}, \quad r = 0, \quad f = 0. \tag{129}$$

Linear B-spline geometry representation of the computational domain with a grid used in experiments is shown in Figure 15. We assume an inflow boundary at the left side of the

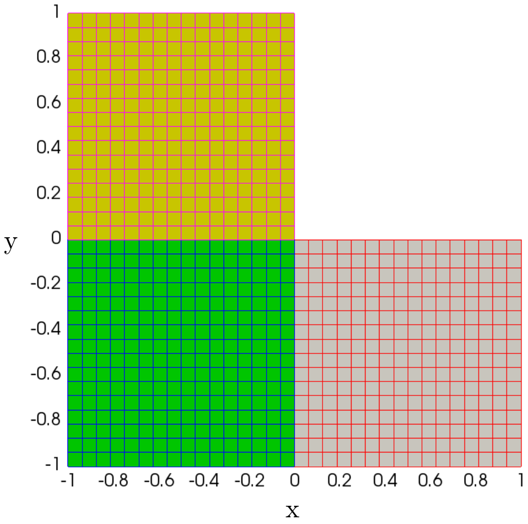


Figure 15: B-spline geometry representation of the L-shape domain with a regular computational grid.

upper left patch, i.e.,

$$\phi(-1, y, t) = 1, \quad y \in (0, 1), t \in [0, T]. \tag{130}$$

At the right side of the lower right patch, the outflow Neumann condition is considered

$$\frac{\partial \phi(1, y, t)}{\partial n} = 0, \quad y \in (-1, 0), t \in [0, T]. \tag{131}$$

The zero Dirichlet condition is set at the rest of the domain boundaries and the initial condition is

$$\phi(x, y, 0) = 0, \quad (x, y) \in \Omega. \tag{132}$$

As mentioned above, our aim is to simulate the fluid flow in the blade cascade for which third degree of the B-splines are used to represent the geometry and hence the turbulent kinetic energy and turbulent specific dissipation are approximated by the third degree B-spline basis. Analogously, the following numerical experiments are computed using the third degree B-spline basis for the numerical solution of the AD problem. Then the resulting discrete problem has 918 DOFs (k-refinement is used for this example, i.e., the linear basis representing the geometry is elevated on the coarsest level and then the knots are inserted into the knot vectors to get the resulting mesh, see Section 4.2).

The time step is chosen $\Delta t = 10^{-3}$ and the numerical solutions are plotted in time $T = 3s$, unless otherwise stated.

Since the advection–diffusion problem (107) with (129) – (132) is advection dominated, an oscillating numerical solution can be expected if no stabilization is used, see Figure 16. Note that the interfaces between the patches are visible, especially the horizontal one. Figures 17 and 18 compare the SUPG (left) and the SOLD adding crosswind diffusion (right) approaches.

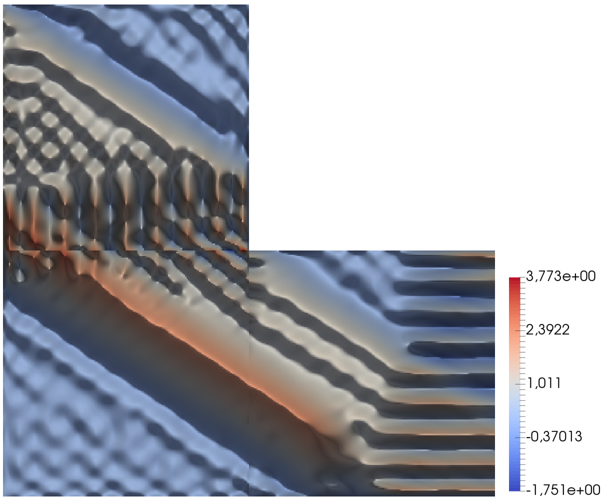


Figure 16: Numerical solution obtained without the employment of the stabilization methods (3D plot top view).

- Stabilization parameter τ_S^2 is used in this example for both SUPG and SOLD terms.
- It can be seen that the SUPG numerical diffusion is sufficient only in the streamline direction and thus the SOLD method is necessary to reduce the spurious oscillations in the crosswind direction.
- The numerical solution oscillates at the outflow boundary using SUPG and the oscillations are eliminated with SOLD.
- Although, the numerical oscillations are still evident along the interfaces, they are

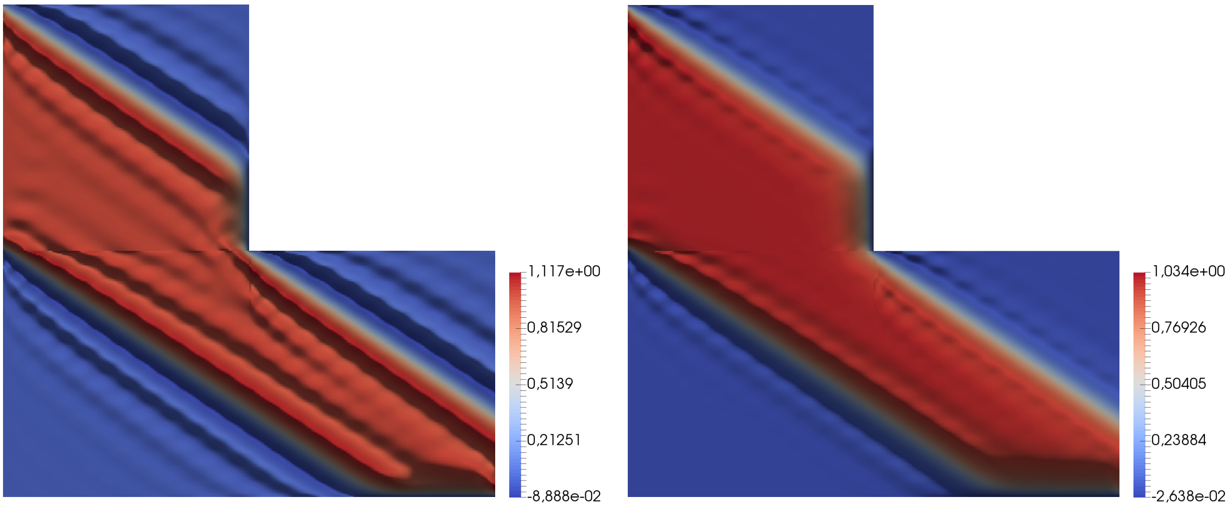


Figure 17: Numerical solution for the SUPG method with τ_S^2 (left) and the SOLD method with additional crosswind diffusion both with τ_S^2 (right), 3D plot (top view).

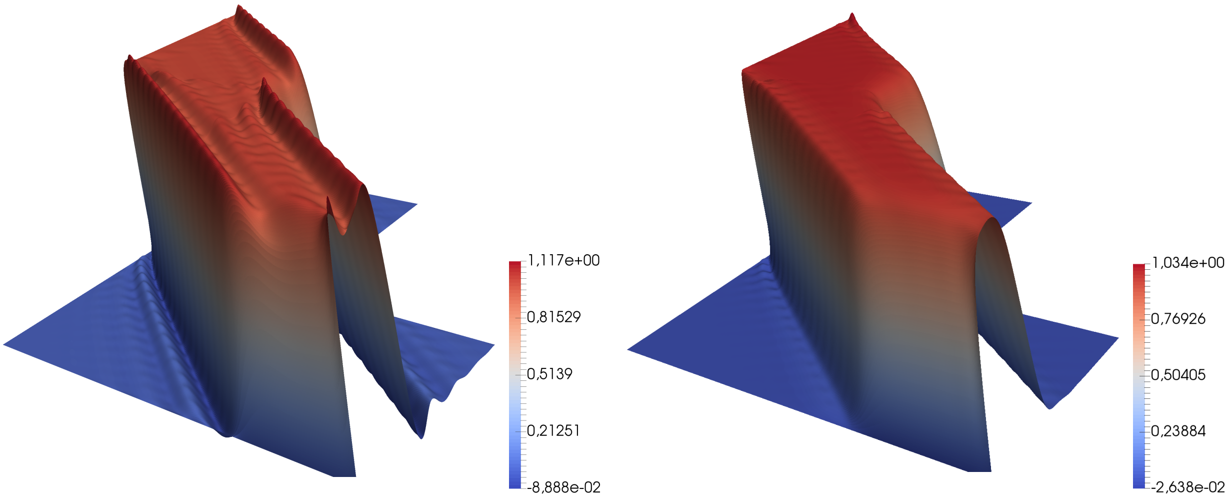


Figure 18: Results from Figure 17 plotted from a different viewpoint.

less visible for the SOLD method. The numerical difficulties along the interfaces depend certainly on the rate of the solution change. The direction of the flow crossing the interface is also important. Moreover, the point of discontinuity $[-1, 0]$ in the boundary condition is also located at the horizontal interface, which makes the problem even more difficult.

In the following, the results are compared in the form of the section along the line $[-1; -0.2]$ - $[0; 0.8]$.

Figure 19 compares the SUPG method with all the stabilization parameters (113) - (119).

- The results are almost identical for the parameters τ_S^0-τ_S^2 and $\tau_S^{0,deg}$-$\tau_S^{2,deg}$ since we solve linear problem with constant coefficients and small diffusion.

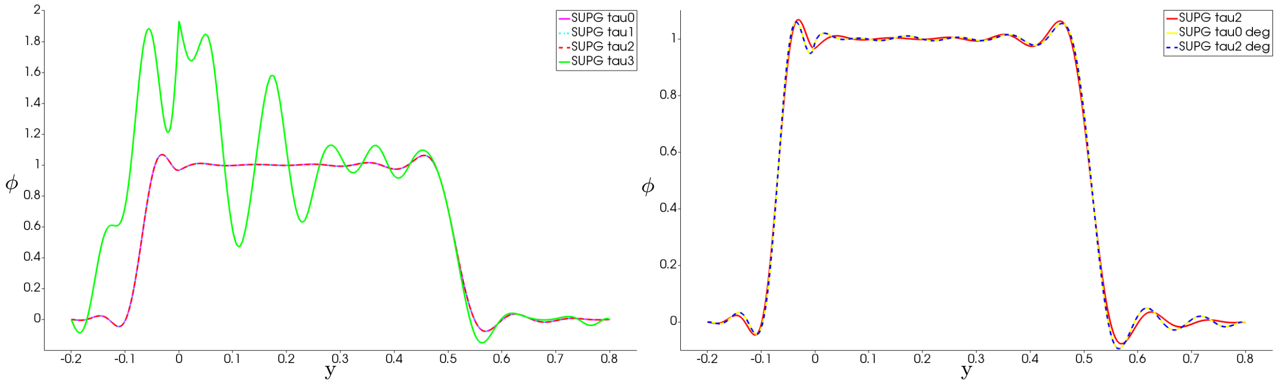


Figure 19: Plot over line, comparison of the SUPG method using various stabilization parameters τ_S (left: τ_S^0, τ_S^1, τ_S^2, τ_S^3, right: τ_S^2, $\tau_S^{0,deg}$, $\tau_S^{2,deg}$).

- It appears that the stabilization parameter τ_S^2 ($\tau_S^{2,deg}$) is not beneficial. Nevertheless, we see this parameter promising for the problems with non-constant diffusion coefficient such that these stabilization terms add less numerical diffusion, resp. more numerical diffusion, in the parts of the domain, where the diffusion coefficient has higher, resp. lower, value. Then, we rather compare the subsequent results with the stabilization parameter τ_S^2 ($\tau_S^{2,deg}$).

- On the other hand, the amount of the additional numerical diffusion is insufficient using SUPG with the parameter τ_S^3, which leads to an unstable solution.

- SUPG using the parameters with basis degree dependency $\tau_S^{0,deg}$, $\tau_S^{2,deg}$ leads to less diffusive results.

The numerical results of the SOLD methods (which add the crosswind or isotropic additional artificial diffusion) with various stabilization parameters are studied in Figures 20 – 22.

- $\alpha = 1.5$ and $\beta = 1.9$ are used for the isotropic diffusion with parameter τ_{iso}^1 (indicated by iso1 in the figure).

- The various stabilization parameters τ_S have similar effect for the SOLD and SUPG approaches, i.e., the employment of the stabilization parameter τ_S^3 or stabilization parameter with the basis degree dependency exhibit oscillations at both the sharp layers and interior layers.

- The most promising results are given by the SUPG method with additional crosswind diffusion using τ_S^2 for both methods or using $\tau_S^{2,deg}$ for SUPG and τ_S^2 for crosswind.

- Unfortunately, the behavior of the numerical solution is rather poor near the interface at $y = 0$ than near the other sharp layer around $y = 0.5$, where all the spurious oscillation are effectively eliminated.

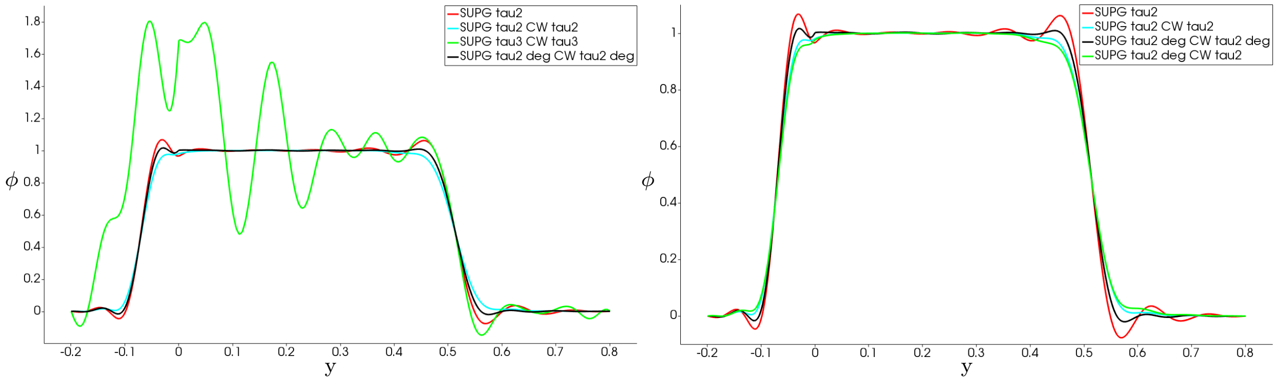


Figure 20: Comparison of the SUPG method and the SOLD method adding crosswind diffusion using various stabilization parameters τ_S chosen the same for both SUPG and crosswind.

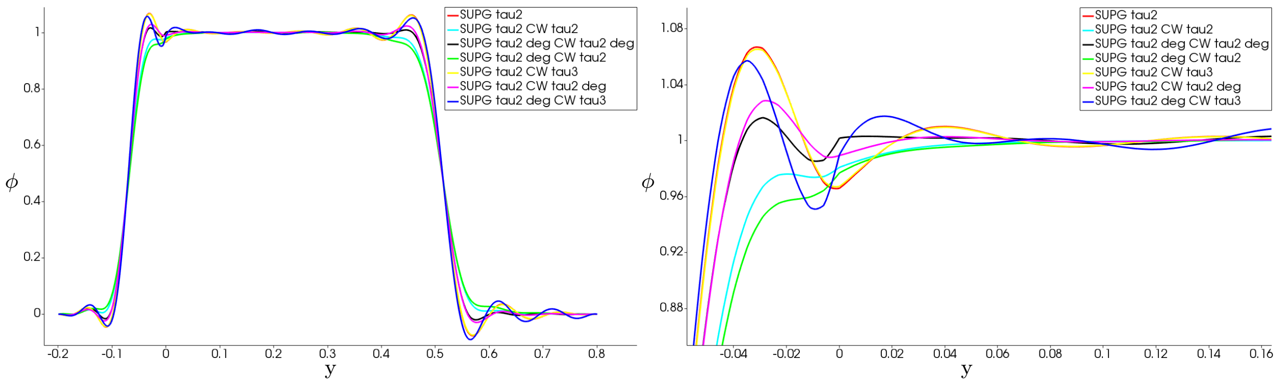


Figure 21: Comparison of the SUPG method and the SOLD method adding crosswind diffusion using various stabilization parameters τ_S chosen differently for SUPG and crosswind in some cases (right: zoom-in).

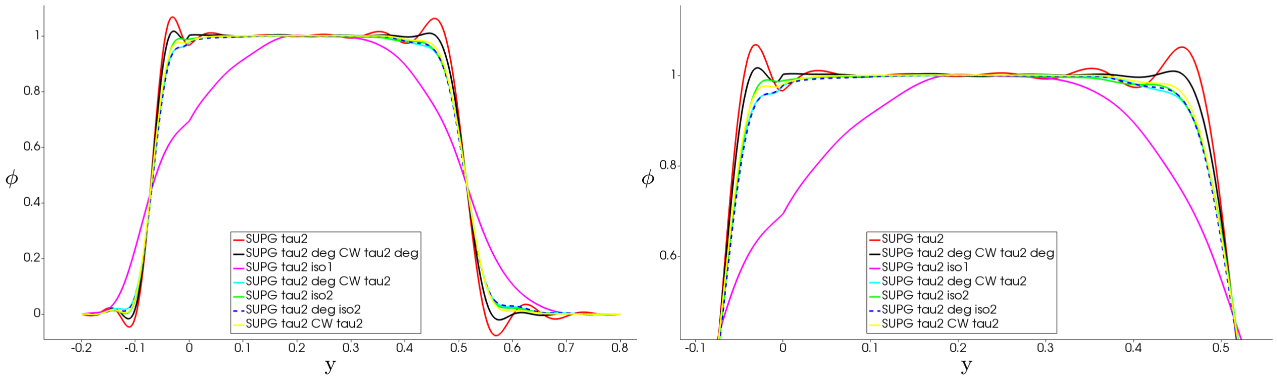


Figure 22: Comparison of the SUPG method, the SOLD method adding crosswind or isotropic diffusion using various stabilization parameters τ_S and τ_{iso} (right: zoom-in).

- The SOLD method which adds crosswind diffusion using $\tau_S^{2,deg}$ (for both SUPG and crosswind methods) leads to acceptable result except for the region near the interface.
- SOLD which adds isotropic diffusion is too diffusive for τ_{iso}^1.
- Parameter τ_{iso}^2 (indicated by iso2) leads to similar results as the SOLD method with crosswind diffusion. SOLD using τ_S^2 for SUPG and τ_{iso}^2 for isotropic diffusion is effective near the interface compared to additional crosswind scheme.

The following experiment is focused on the comparison of the consistent and inconsistent SUPG method. Two types of the inconsistent SUPG terms (I-SUPG and IT-SUPG) were introduced by (120) and (121). The numerical results are displayed in Figure 23 applying these approaches.

- The I-SUPG method produces too much numerical diffusion, even more than SOLD with isotropic diffusion τ_{iso}^1 in this experiment. The amount of the additional numerical diffusion can be again reduced using different stabilization parameter.

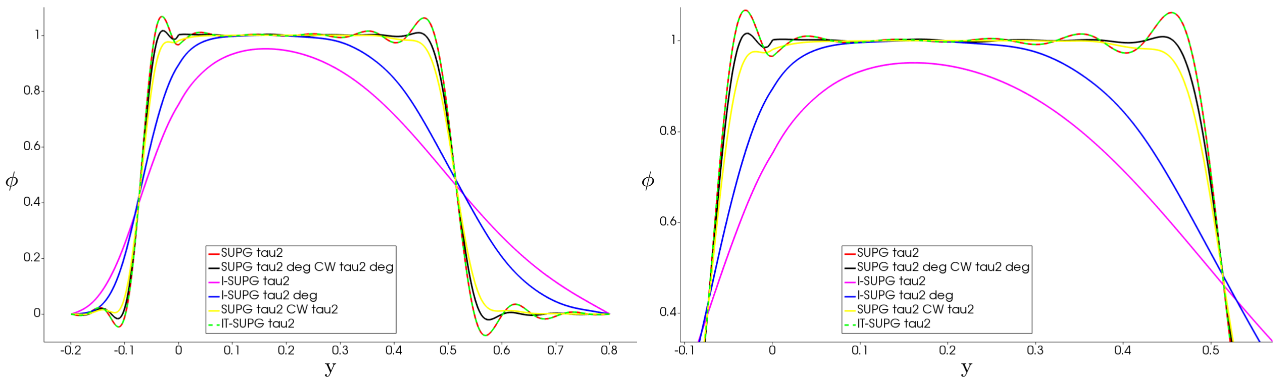


Figure 23: Comparison of the consistent and inconsistent SUPG methods (right: zoom-in).

- The IT-SUPG method gives almost identical result with the consistent SUPG method. Indeed, the advection–diffusion equation is solved with dominated advection and small constant diffusion. Hence, the diffusion term could be omitted in the element residual, which implies the IT-SUPG approach.
- If the stabilization parameter τ_S^3 is used, the numerical solution oscillates again, but it is not displayed in the figure.

We study the influence of the local refinement along the interfaces on the numerical solution in the following. This numerical experiment is very important since all the meshes are locally refined for the fluid flow simulation as mentioned in Section 5.

The mesh displayed in Figure 24 is used for numerical solution. In Figure 25, we can compare the SUPG and SOLD stabilizations with and without local refinement.

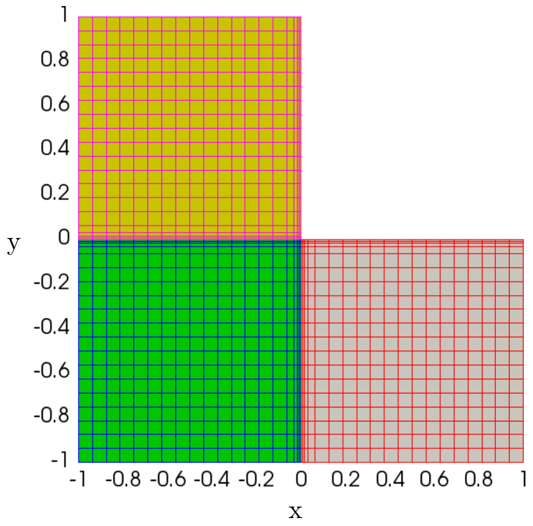


Figure 24: Local refinement of the grid in Figure 15 along the interfaces.

- The numerical solutions differ a lot near the interface. Unfortunatelly, the results on the locally refined mesh involve oscillations with higher magnitude.
- On the other hand, the numerical solutions are identical in the parts of the domain far from the interface.

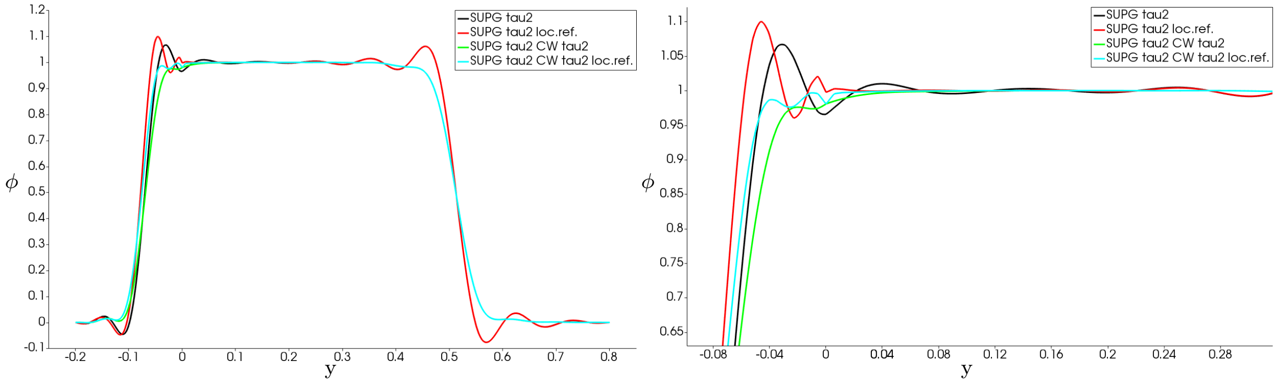


Figure 25: Numerical solution computed on the regular and locally refined meshes for the SUPG method and the SOLD method adding crosswind diffusion with the stabilization parameter τ_S^2 (right: zoom-in).

Now, we want to show the influence of the uniform refinement.

- Figures 26 – 29 display the differences of the numerical solution computed on four meshes using various stabilization techniques. GRID1 is the mesh displayed in the Figure 15 above. GRID0 is a coarser grid such that we use one less uniform refinement than in the case GRID1, then the problem has 270 DOFs.
- GRID2 and GRID3 are finer meshes such that we use one and two more uniform refinements compared to the basic mesh GRID1. The resulting problems have 3366 and 12870 DOFs.
- The numerical solution is less smeared on finer grid. But we cannot generally observe that the numerical oscillations reduce on finer grid.
- Also, the 3D results (top view) computed using the SOLD methods with various stabilization parameters τ_S are displayed in Figure 30.
- The problematic areas are clearly visible in the figures, e.g. the area near the discontinuous Dirichlet boundary condition, where the flow immediately meets the interface between two patches. Or the areas where the numerical solution meets the walls.
- Certainly, the method using the stabilization parameter τ_S^j with a higher label j leads to less diffusive result and thus we can expect more dominant spurious oscillations in the problematic parts of the domain.

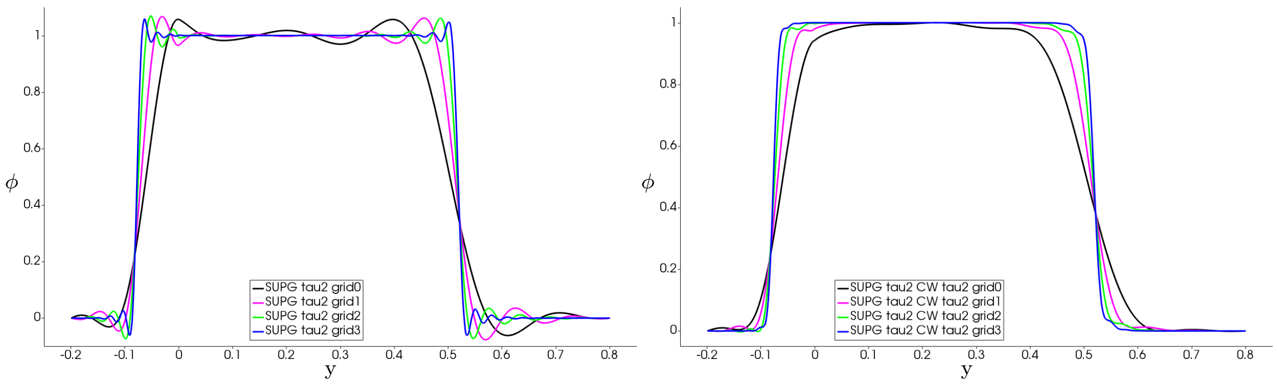


Figure 26: Numerical solution computed on four uniformly refined meshes for the SUPG method using τ_S^2 (left) and for the SOLD method adding crosswind diffusion with parameter τ_S^2 for both SUPG and crosswind (right).

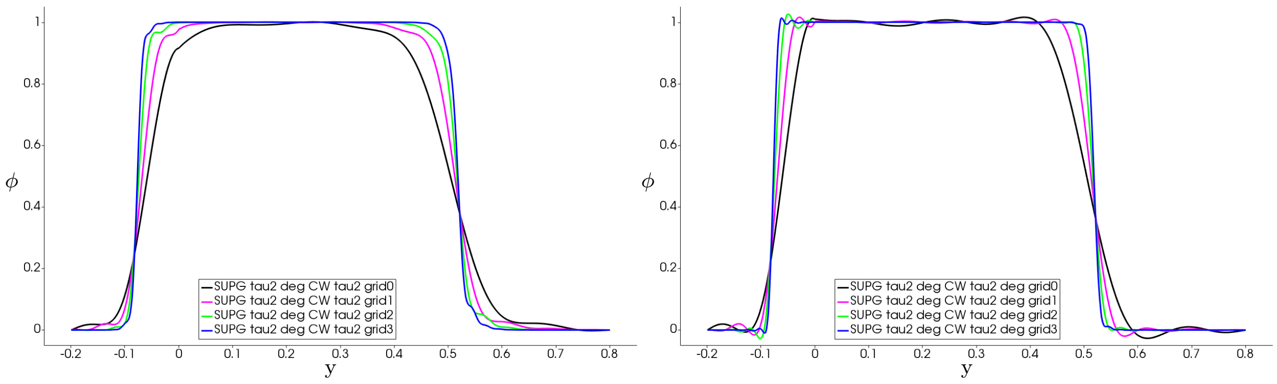


Figure 27: Numerical solution computed on four uniformly refined meshes using the SOLD method with additional crosswind diffusion and the parameter $\tau_S^{2,deg}$ for SUPG and τ_S^2 for crosswind (left) and the parameter $\tau_S^{2,deg}$ for both SUPG and crosswind (right).

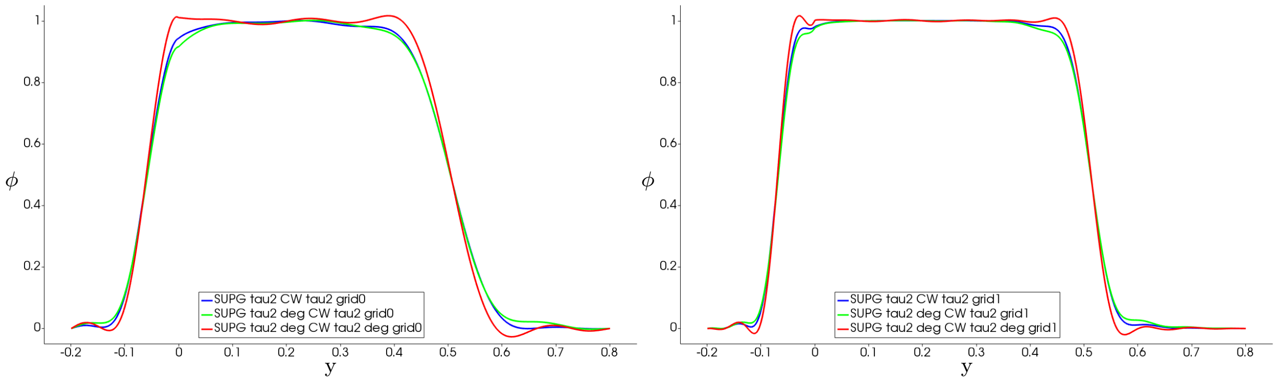


Figure 28: Comparison of the stabilization methods on GRID0 (left) and GRID1 (right).

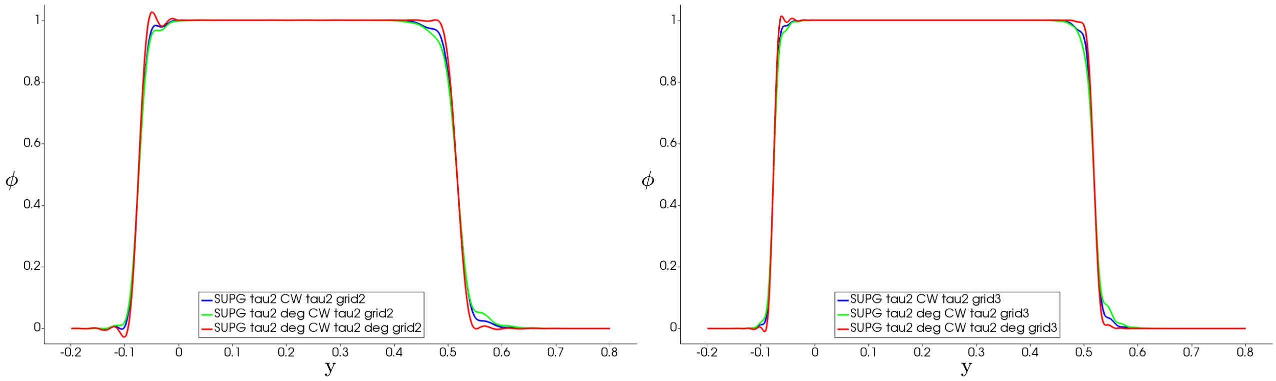


Figure 29: Comparison of the stabilization methods on GRID2 (left) and GRID3 (right).

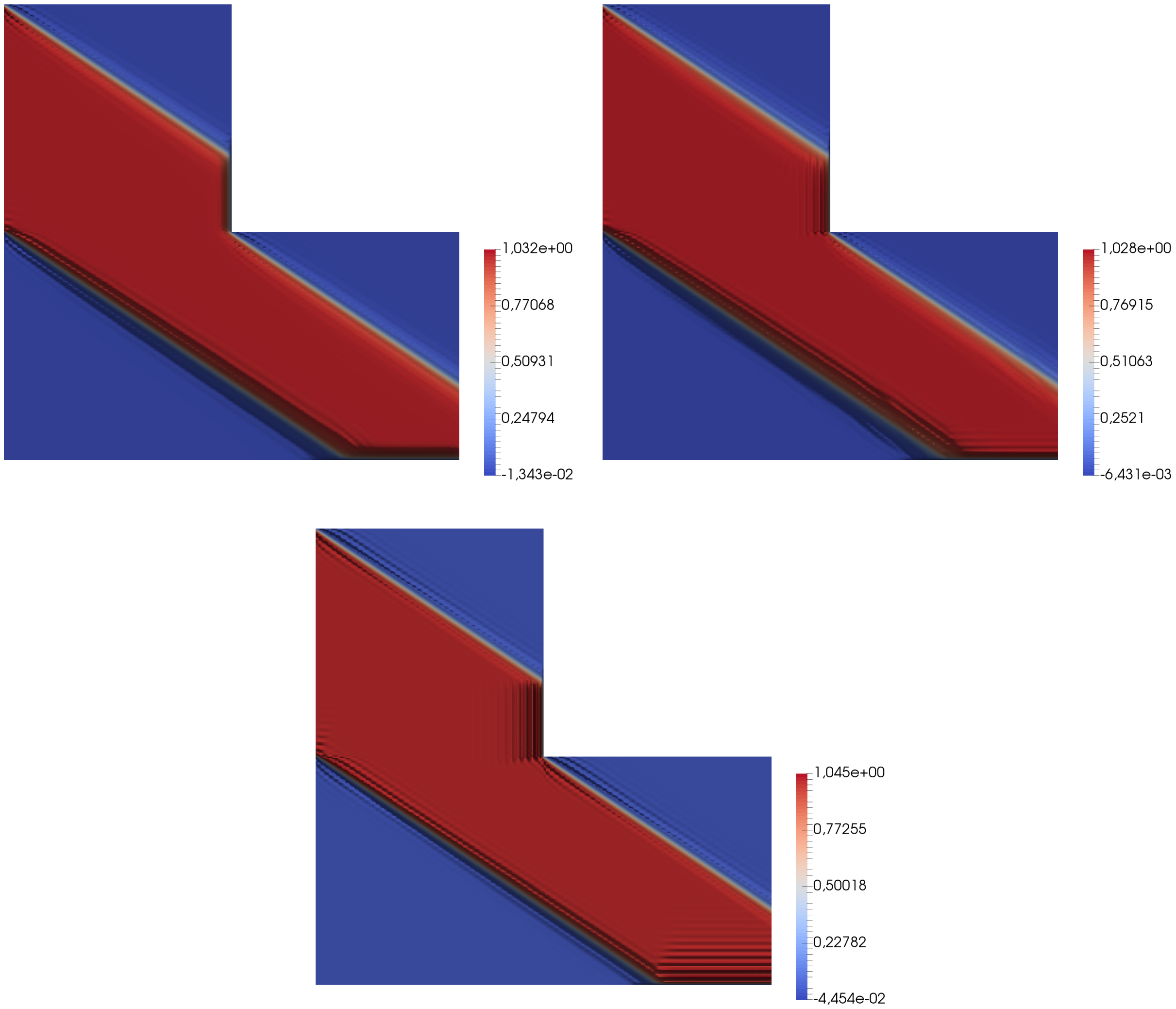


Figure 30: Numerical solution computed on GRID3 using the SOLD method with the additional crosswind diffusion and stabilization parameter τ_S^2 for both SUPG and crosswind (top left), $\tau_S^{2,deg}$ for SUPG and τ_S^2 for crosswind (top right) and $\tau_S^{2,deg}$ for both SUPG and crosswind (bottom), 3D plot (top view).

Let us recall that the stabilization parameters τ_S in (113)–(119) include the element length computation h. In the previous experiments, the length of the element diagonal was used for h. Now, we compare the results computed using h as the maximum length of the element sides, see Figures 31 and 32.

- Stabilization method introduces less numerical diffusion using h as maximum length of the element sides. This can be expected since the computational meshes consist only of the rectangular elements in this section and thus the element diagonal is longer than the maximum element side.
- Then, the maximum element side is beneficial in the areas far from the interfaces, but it has a negative effect on the results near the interface.

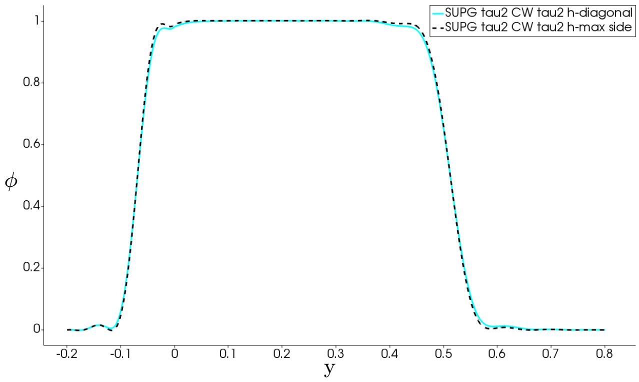


Figure 31: Comparison of the element diameter computation on the regular mesh.

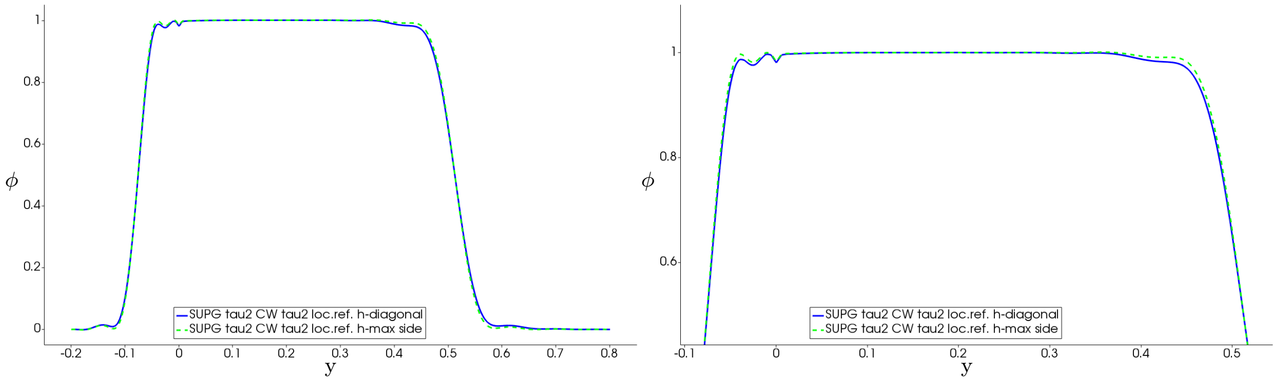


Figure 32: Comparison of the element diameter computation on the locally refined mesh (right: zoom-in).

In Figure 33, we give the numerical results for various magnitudes of the diffusion coefficient from $D = 10^{-2}$ to $D = 10^{-6}$. The same time step size is still used for all experiments, i.e. $\Delta t = 10^{-3}$.

- The results are very similar except $D = 10^{-2}$.

- The numerical solution is almost identical for $D = 10^{-4}$, $D = 10^{-5}$ and $D = 10^{-6}$.

- We get similar observations if we compute advection–diffusion problem without any stabilization, see Figure 34.

- If we compare the results with and without stabilization for $D = 10^{-2}$, Figure 34 (right), we can see that both results are very close to each other, but the stabilization method eliminates the slight numerical instabilities.

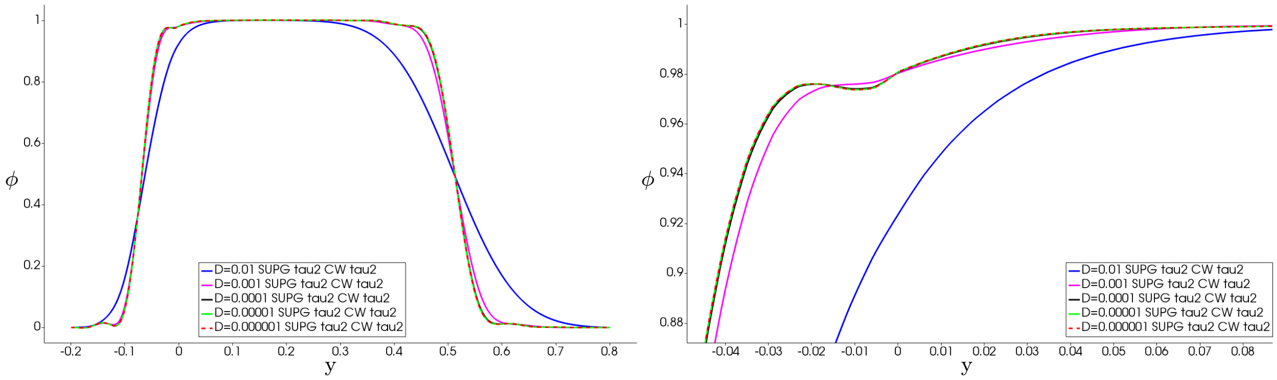


Figure 33: Numerical solution for various magnitude of the diffusion coefficient using the SOLD method (right: zoom-in).

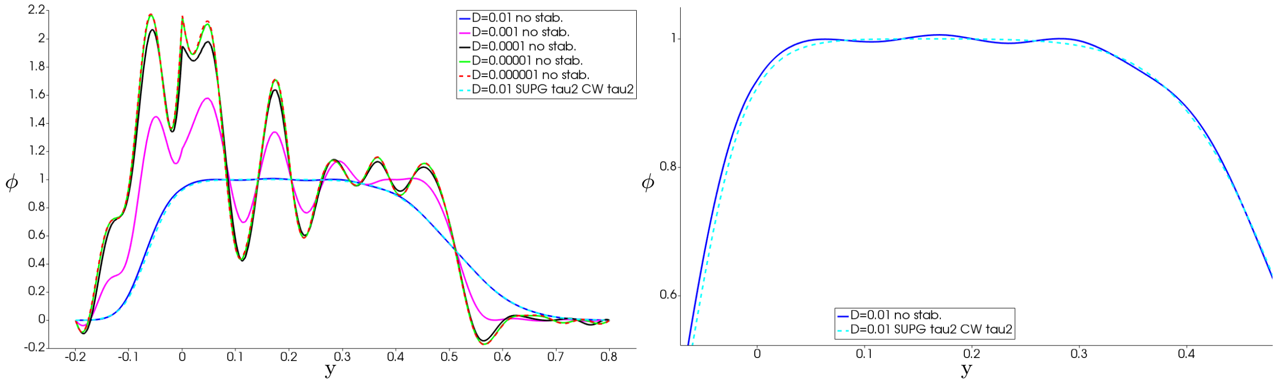


Figure 34: Numerical solution for various magnitudes of the diffusion coefficient without using any additional stabilization terms, compared to the SOLD method (right: zoom-in of some results).

Third degree B-spline basis was used in the previous experiments. In the rest of this section, we focus on study of the dependency of the quality of the numerical solution on the various degrees of the B-spline basis functions to show the advantages and disadvantages of the higher order spatial approximation. Note that the higher order B-splines indicate advantages in the literature including the motivation to the isogeometric analysis (see e.g. [50]). But the experiments are usually computed on a single patch domain with regular mesh. We thus want to find out whether the same behavior can be achieved in our computational problems.

- The degree of the B-spline basis functions is increasing from 1 to 6 in Figures 35 – 37 such that also the continuity of the basis functions increases. Nevertheless, the continuity at the interfaces remains C^0.
- The sections of the numerical solution along the lines $[-1; -0.2]$ - $[0; 0.8]$ and $[1, 0]$ – $[1, -1]$ are shown in Figures 38 and 39.
- The SUPG method using the stabilization parameter τ_S^2 is employed.
- The higher degree of the basis functions (q) tends to reflect better the discontinuous character of the solution, but the SUPG stabilization is not sufficient to keep the numerical solution smooth.
- Thus, we do not observe the mentioned positive effect of the higher degrees of the basis functions in this case. Contrarily, the magnitude of the undershoots and overshoots tends to increase with increasing q in the interior.
- Only, the overshoots are reduced with higher q in the left corner on the outflow boundary, see Figure 39 (left).

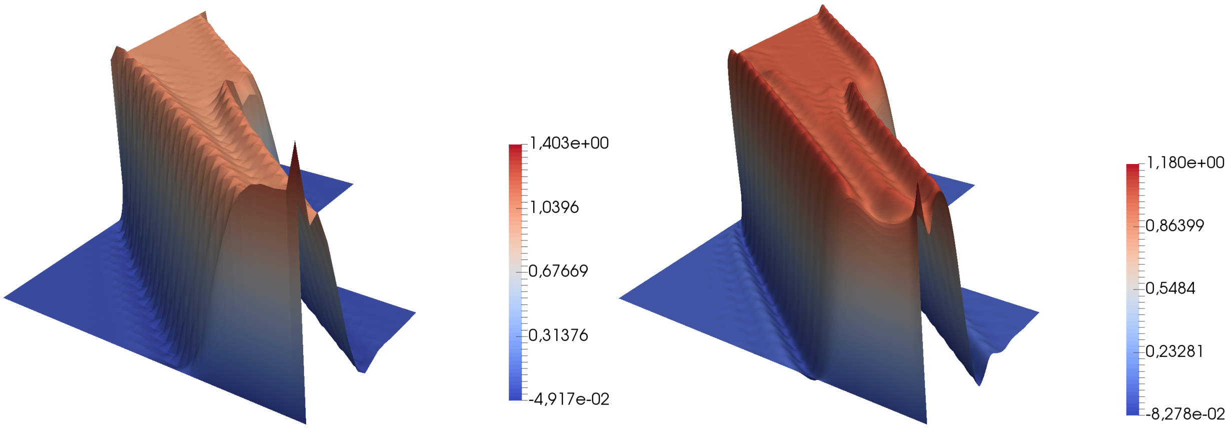


Figure 35: Numerical solution for the B-splines basis functions of the degree $p = 1$ with C^0 continuity (left) and $p = 2$ with C^1 continuity (right) using the SUPG method with τ_S^2.

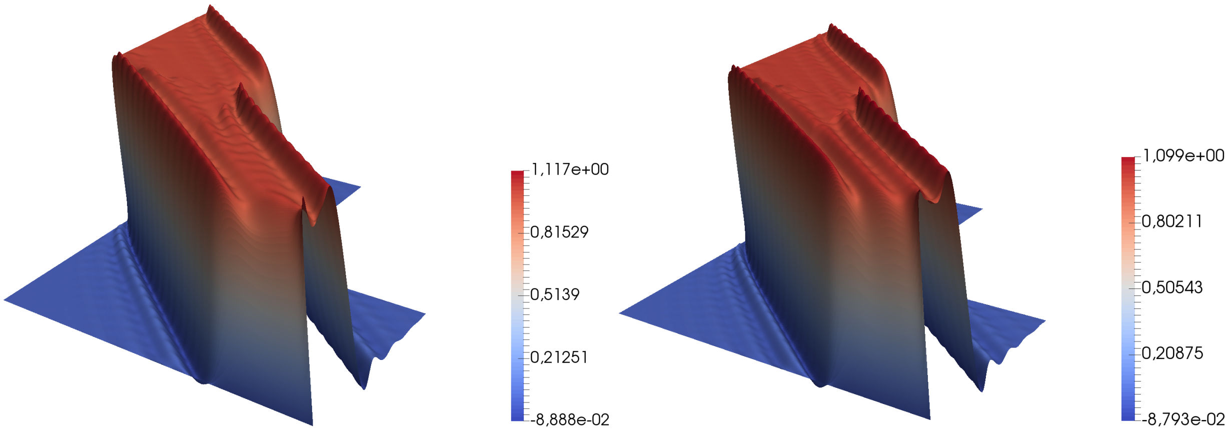


Figure 36: Numerical solution for the B-splines basis functions of the degree $p = 3$ with C^2 continuity (left) and $p = 4$ with C^3 continuity (right) using the SUPG method with τ_S^2.

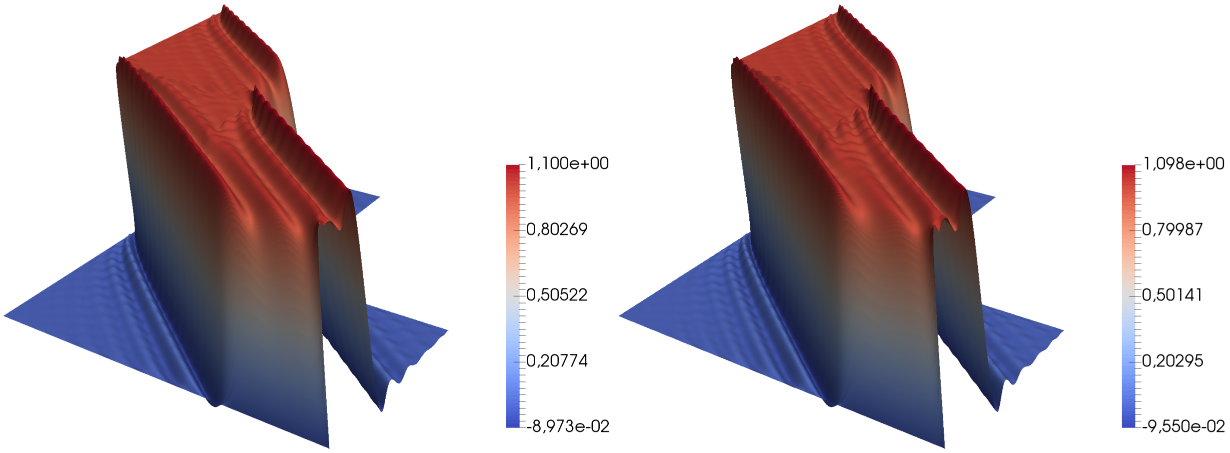


Figure 37: Numerical solution for the B-splines basis functions of the degree $p = 5$ with C^4 continuity (left) and $p = 6$ with C^5 continuity (right) using the SUPG method with τ_S^2.

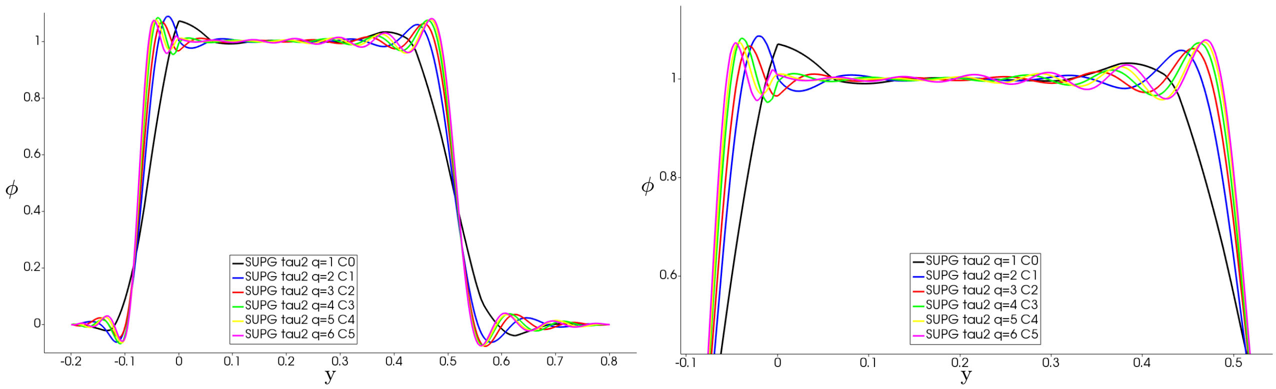


Figure 38: Sections through the numerical solution for the B-splines basis functions of the degree from $p = 1$ to $p = 6$ with the continuity from C^0 to C^5 using the SUPG method with τ_S^2 (right: zoom-in).

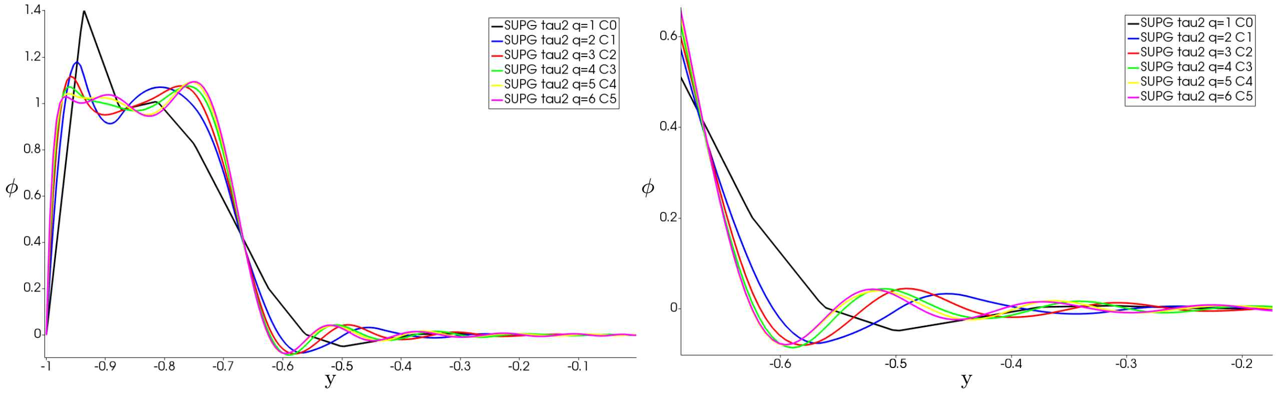


Figure 39: Sections through the numerical solution for the B-splines basis functions of the degree from $p = 1$ to $p = 6$ with the continuity from C^0 to C^5 using the SUPG method with τ_S^2 (right: zoom-in).

- The situation is different, if we solve the problem employing the SOLD method (SUPG with additional crosswind diffusion with τ_S^2 for both), see the results in Figures 40 – 47.

- Except for the surroundings of the interface, we can notice the promising behavior of the numerical solution with increasing degree q.

- Nevertheless, we can see the tendency to the slightly oscillating behavior, which is typical for higher order approximations.

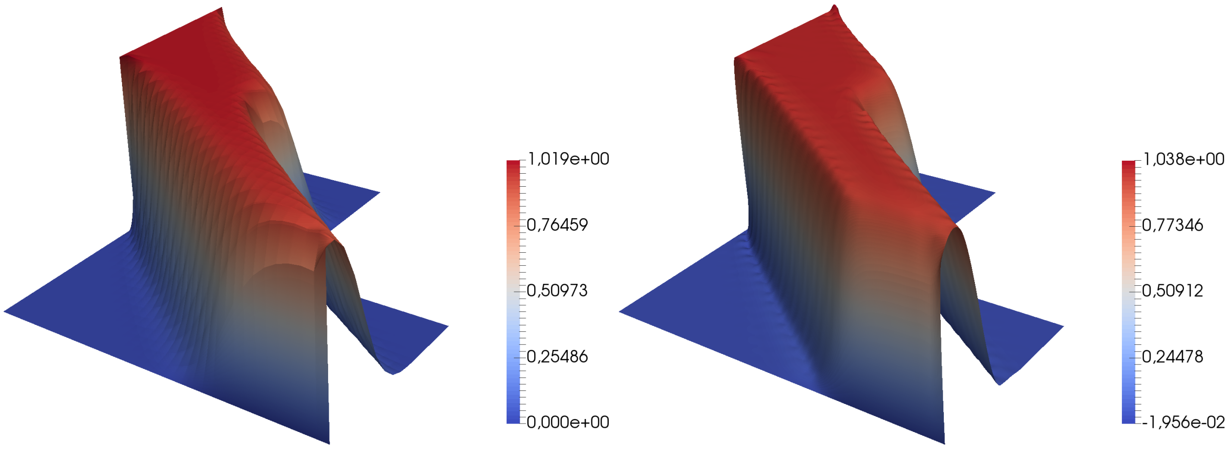


Figure 40: Numerical solution for the B-spline basis functions of the degree $p = 1$ with C^0 continuity (left) and $p = 2$ with C^1 continuity (right) using the SOLD method with additional crosswind diffusion and τ_S^2.

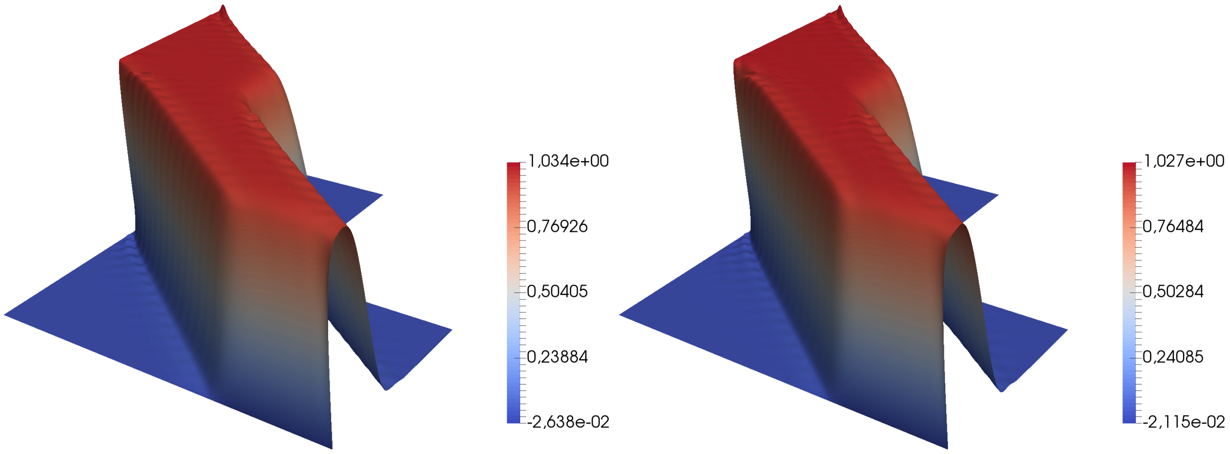


Figure 41: Numerical solution for the B-spline basis functions of the degree $p = 3$ with C^2 continuity (left) and $p = 4$ with C^3 continuity (right) using the SOLD method with additional crosswind diffusion and τ_S^2.

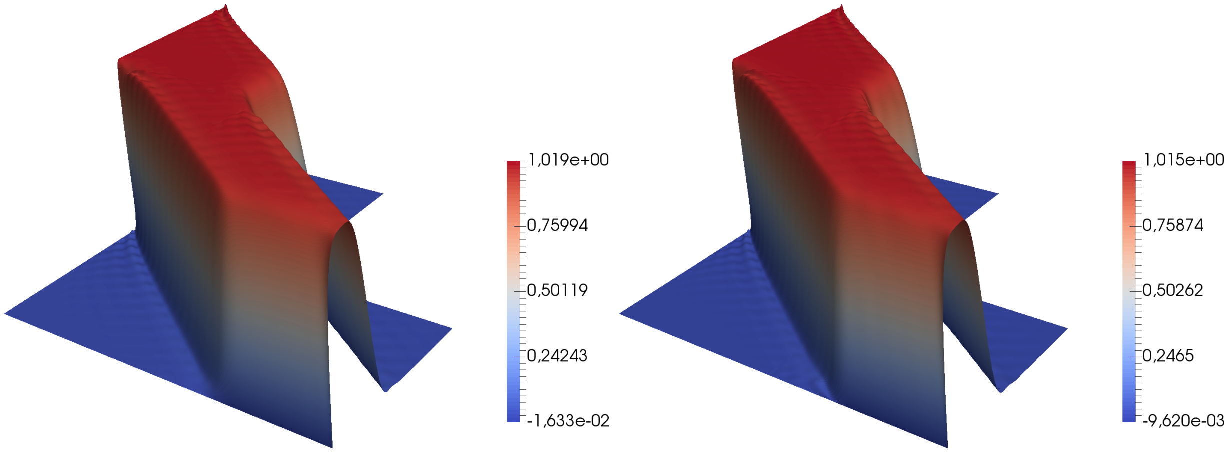


Figure 42: Numerical solution for the B-spline basis functions of the degree $p = 5$ with C^4 continuity (left) and $p = 6$ with C^5 continuity (right) using the SOLD method with additional crosswind diffusion and τ_S^2.

Last experiment is similar to the latter one. The degree of the B-spline basis is increased, but the continuity of the basis remains C^0 in each patch. The results are shown in Figure 48, where the degree of the B-spline basis increases from 1 to 6. The section of the result obtained using basis of the third degree with C^2 continuity is added for comparison. This representative was chosen according to the basis degree used for the computations in the blade profile geometry.

- The influence of the degree elevation to the numerical solution is opposite to the previous case, where both degree of the basis and its order of continuity were increased.
- Except for $q = 1$, the smearing of the layers is larger with increasing degree q.
- Evidently, the results with C^0 continuity dominates over the computations with higher order continuity for the layer near the interface.

- On the other hand, the right layer of the section is less smeared using basis with higher continuity.

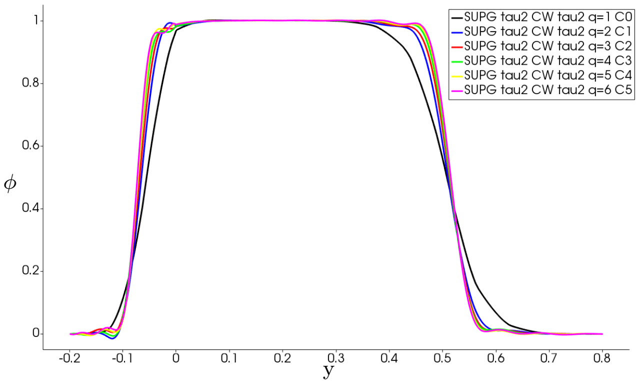


Figure 43: Sections through the numerical solution for the B-spline basis functions of the degree from $p = 1$ to $p = 6$ with the continuity from C^0 to C^5 using the SOLD method with additional crosswind diffusion and τ_S^2.

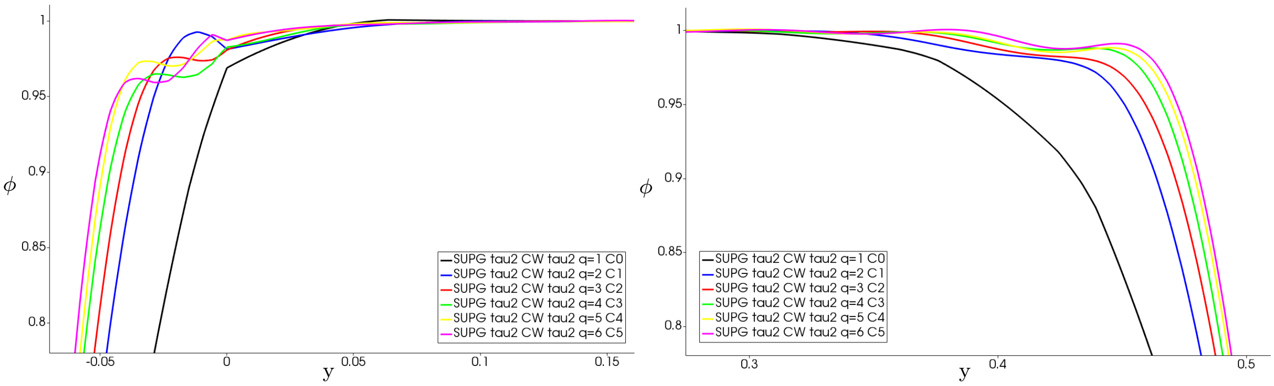


Figure 44: Zoom-in of the upper left and upper right part of the results in Figure 43.

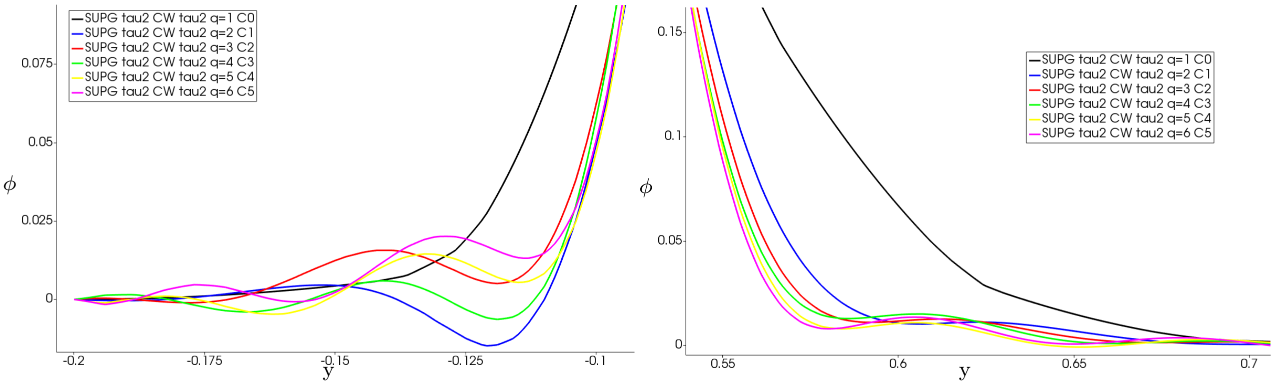


Figure 45: Zoom-in of the bottom left and bottom right part of the results in Figure 43.

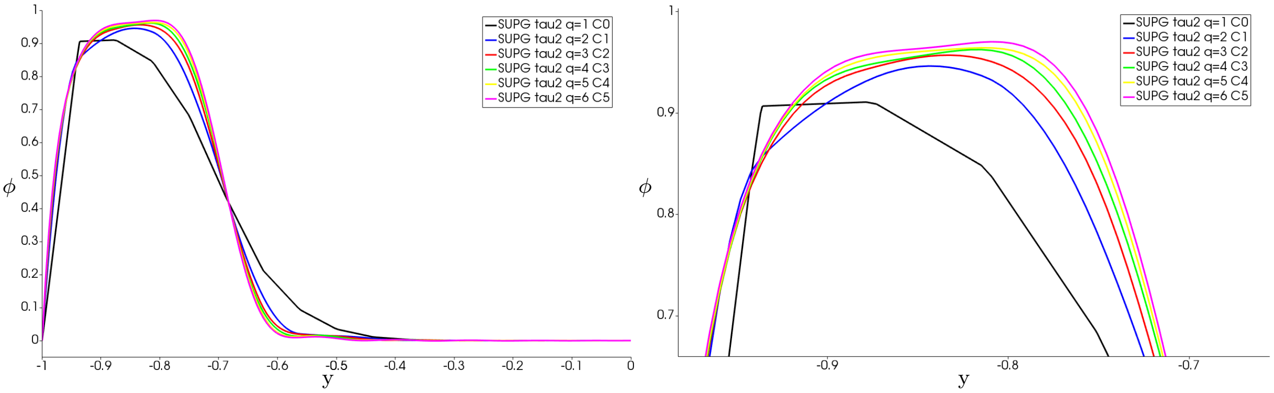


Figure 46: Sections through the numerical solution for the B-spline basis functions of the degree from $p = 1$ to $p = 6$ with the continuity from C^0 to C^5 using the SOLD method with additional crosswind diffusion and τ_S^2 (right: zoom-in).

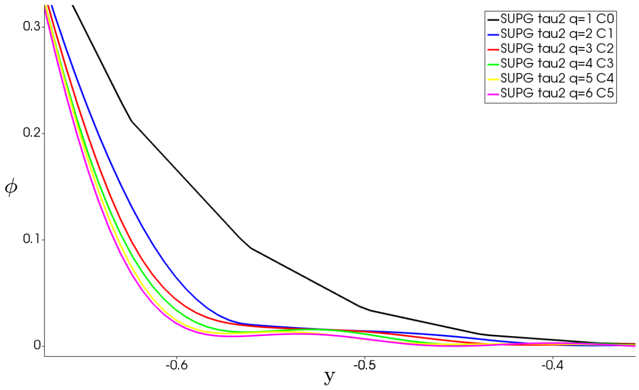
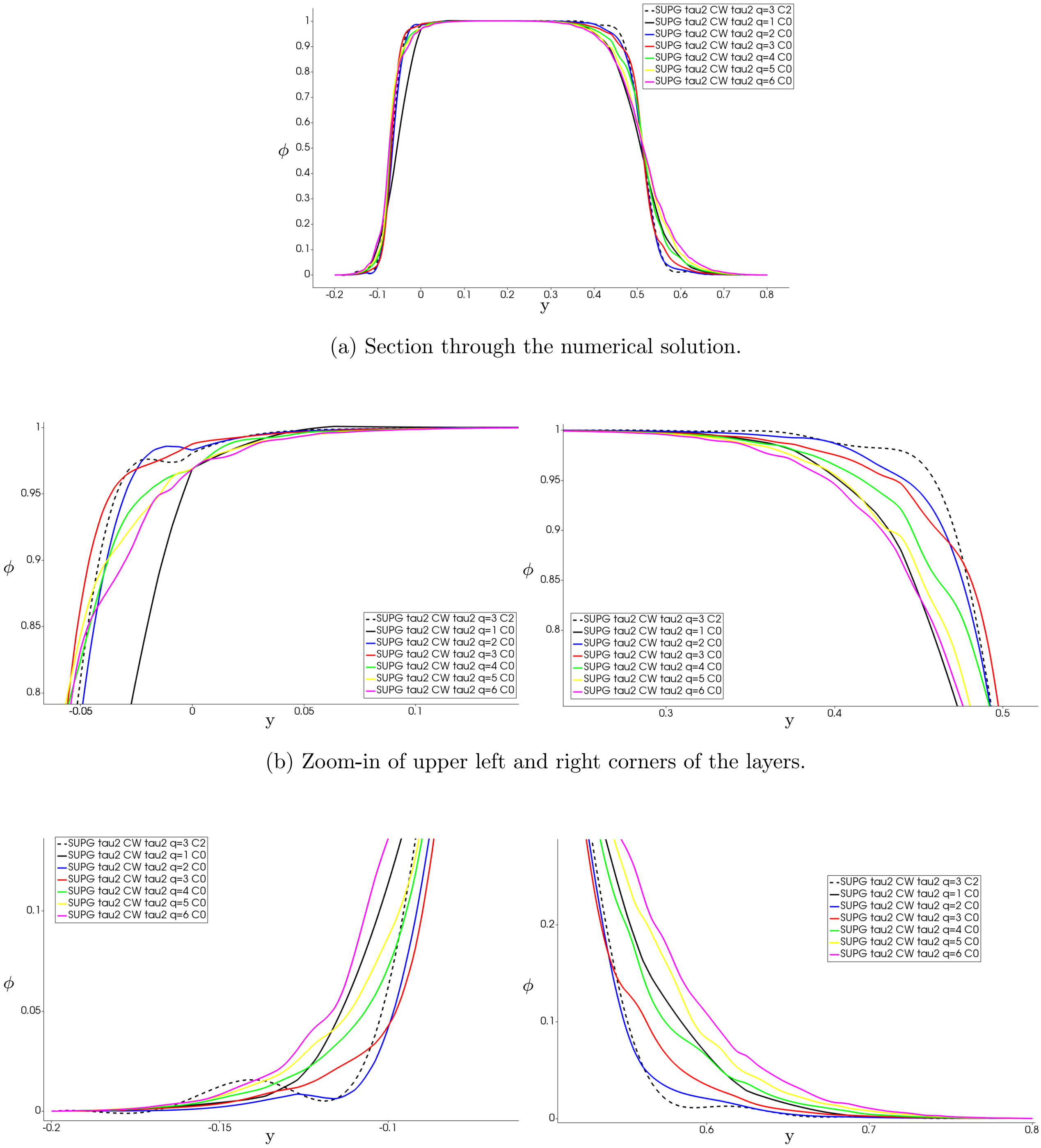


Figure 47: Zoom-in of the bottom right part of the results in Figure 46.



(a) Section through the numerical solution.

(b) Zoom-in of upper left and right corners of the layers.

(c) Zoom-in of bottom left and right corners of the layers.

Figure 48: Sections thought the numerical solution for the B-spline basis functions of the degree from $q = 1$ to $q = 6$ with C^0 continuity using the SUPG method with additional crosswind diffusion for τ_S^2.

5.2.2 Example 2

Second example is devoted to solving the advection–diffusion equation again, but the computational domain is the blade cascade displayed in Figure 6. We prescribe the initial and boundary conditions similar to our target problem for the turbulence kinetic energy k since we want to approach the setting of the turbulence model. The inlet boundary condition is

$$\phi(-0.373118, y, t) = 0.1449, \quad y \in (-0.246578, 0.028311), t \in [0, T] \tag{133}$$

and the Neumann condition at outlet

$$\frac{\partial \phi(0.433544, y, t)}{\partial n} = 0, \quad y \in (0.34605, 0.620939), t \in [0, T]. \tag{134}$$

The zero Dirichlet condition is set at the walls representing the blade profile and the periodic conditions are set at the rest of the boundaries. The initial condition is considered as the expanded inlet condition into the whole domain, i.e.,

$$\phi(x, y, 0) = 0.1449, \quad (x, y) \in \Omega. \tag{135}$$

The advection coefficient is still constant in the whole domain and is chosen to be equal to the inlet velocity profile, which is considered for the fluid flow simulation (see Section 5), i.e. $\mathbf{b} = (7.765, -0.28272)$. The diffusion coefficient $D = 10^{-5}$, $\Delta t = 10^{-3}$ and $T = 0.2$, unless otherwise stated.

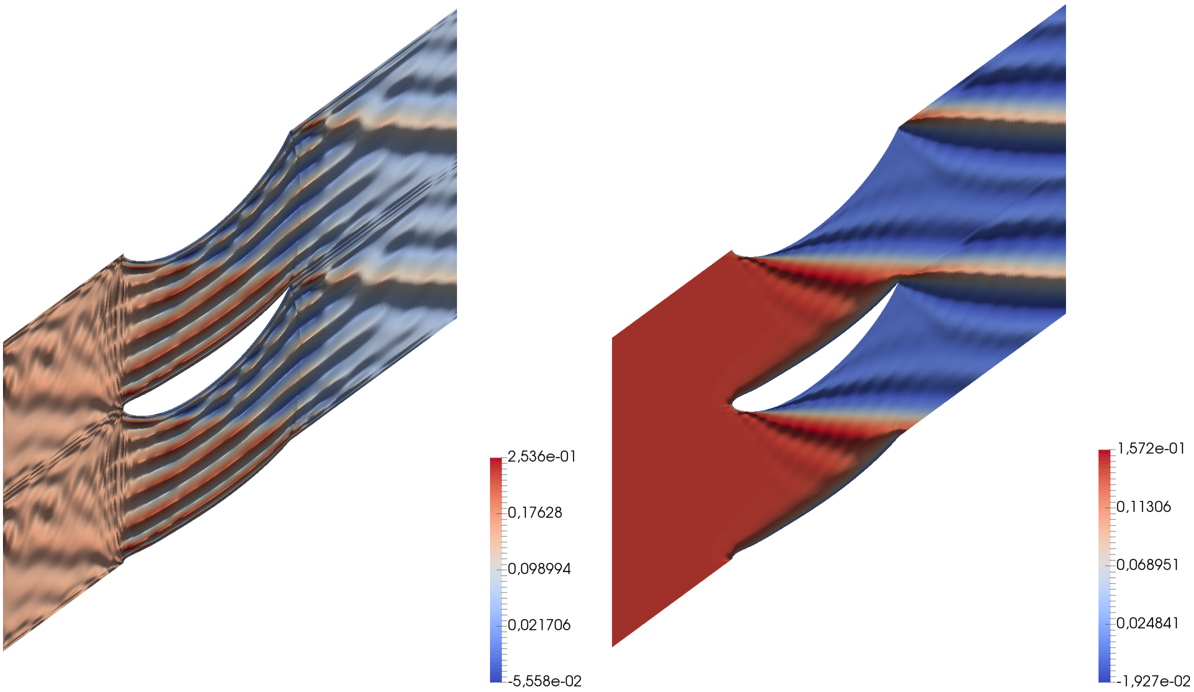


Figure 49: Numerical solution computed without using the stabilization methods (left) and applying the SUPG method with τ_S^2 (right), 3D plot (top view).

The numerical solution of the given advection–diffusion problem is shown in Figure 49, which oscillates since no stabilization technique is used (left) and the oscillations are reduced using the SUPG stabilization (right) with parameter τ_S^2. Similarly to the first example in Section 5.2.1, the interfaces are visible, mainly in the case without stabilization.

The following Figures 50 – 56 compares the results using various stabilization techniques in the form of the section along the y direction at $x = -0.0768$. Note that the section is not crossing any interface in this example.

- The stabilization methods with various parameters τ_S have similar effect to the numerical solution as in the previous test case except for the stabilization parameter τ_S^3.

- Remember that the results using τ_S^3 were close to the solution computed without stabilization in Section 5.2.1. In this numerical example, stabilization terms also add less numerical diffusion for τ_S^3, but the resulting solution does not oscillates, only some remaining spurious oscillation are observed.

- Generally, the parameters with the degree dependency and τ_S^3 are beneficial in the left part of the cross-section since they give less numerical diffusion.

- However, it leads to the over/under-shoots of higher magnitudes, see Figures 51 and 52.

- We thus need to find a compromise such that the left part is not significantly smeared and the right part of the cross-section is not oscillating.

- The crosswind diffusion offers the possibility to reduce the numerical oscillations such that the results are not excessively smeared, e.g., if we choose the stabilization parameter τ_S^j with higher value of the index j or if we choose the parameter with the degree dependency as can be seen in Figures 53 – 55.

- The influence of the isotropic diffusion (with $\alpha = 1.5$ and $\beta = 1.9$ for τ_{iso}^1) can be observed in Figure 56.

- The results are not satisfactory for τ_{iso}^1, since too much numerical diffusion is added in the left part of the graph and the added numerical diffusion is not sufficient in the right part of the graph.

- Additional isotropic diffusion for τ_{iso}^2 is also rather poor in this experiment as the numerical solution is smeared.

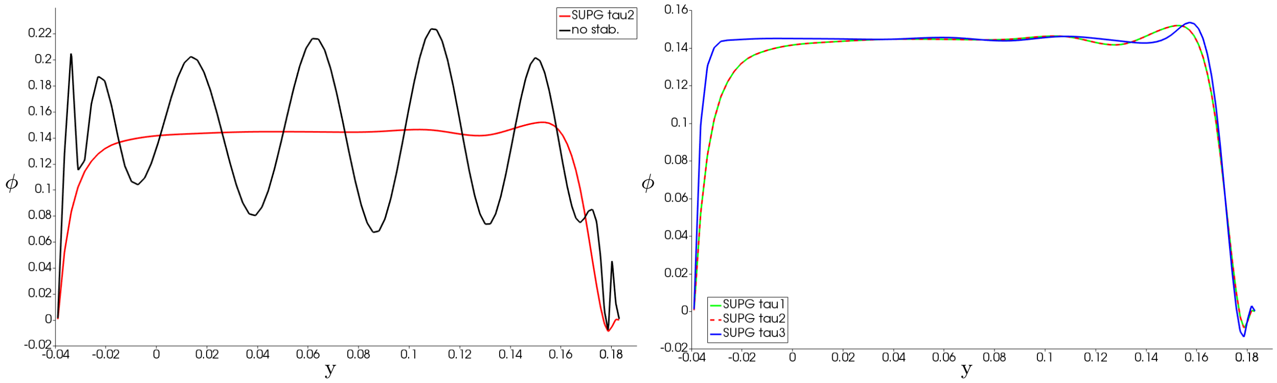


Figure 50: Sections through the numerical solution, comparison of the results computed with and without stabilization (left), SUPG method using various stabilization parameters τ_S (right).

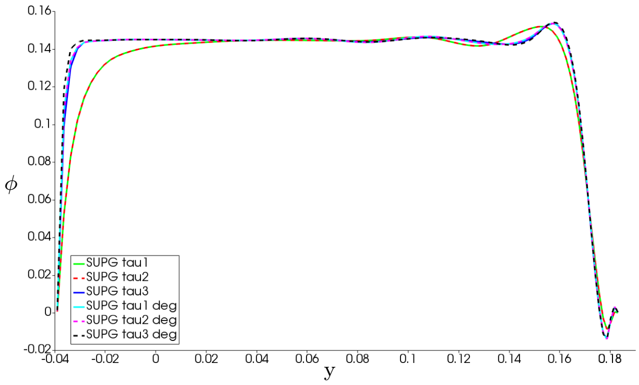


Figure 51: Sections through the numerical solution, comparison of the SUPG method using various stabilization parameters τ_S.

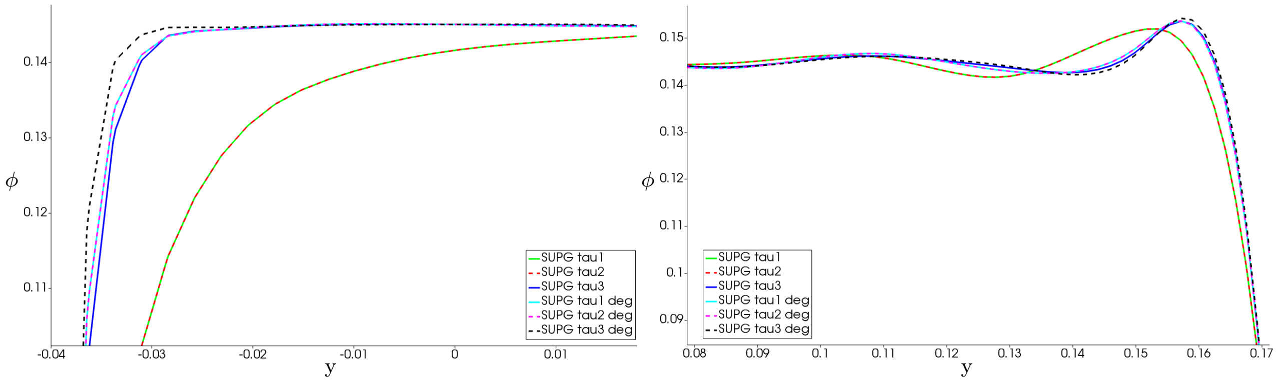


Figure 52: Zoom-in of the upper left and upper right part of the results in Figure 51.

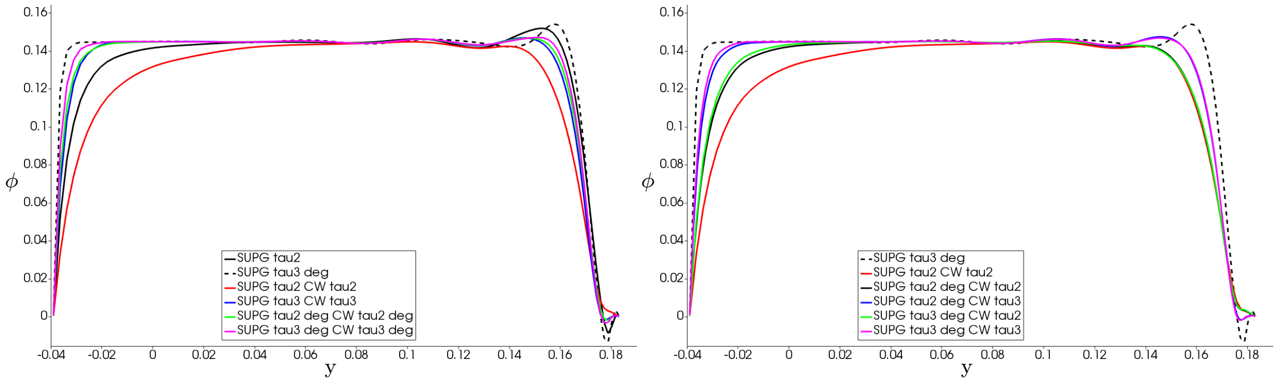


Figure 53: Comparison of the SUPG method and the SOLD method with the additional crosswind diffusion using various stabilization parameters τ_S.

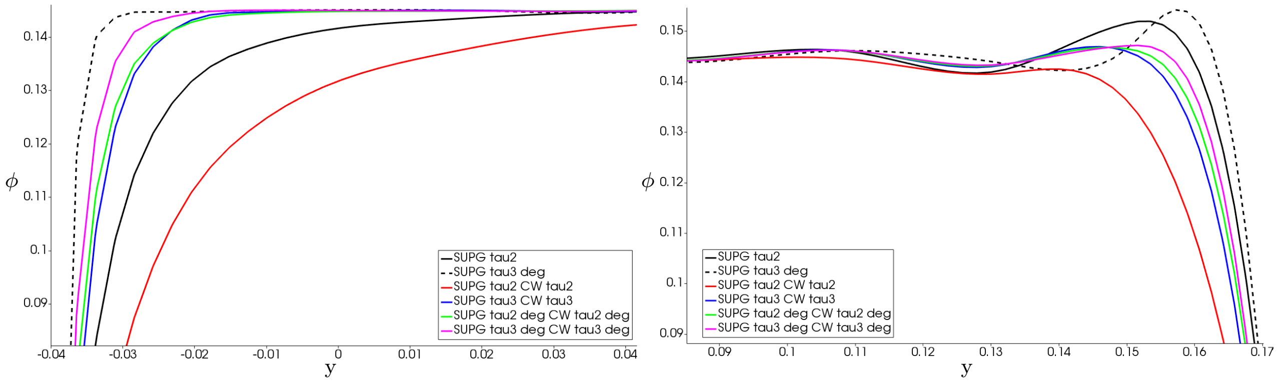


Figure 54: Zoom-in of the upper left and upper right part of the results in Figure 53 (left).

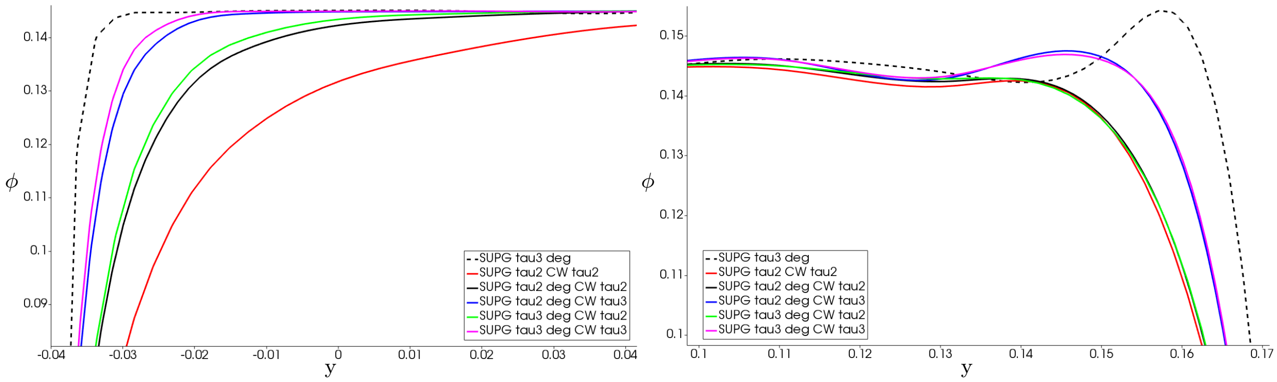


Figure 55: Zoom-in of the upper left and upper right part of the results Figure 53 (right).

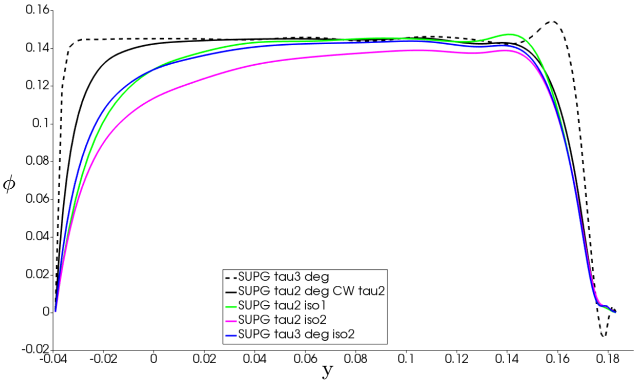


Figure 56: Comparison of the SUPG method, the SOLD method with additional crosswind or isotropic diffusion using various stabilization parameters τ_S and τ_{iso}.

In Section 5.2.1, we did not presented the influence of the time step size on the quality of the numerical solution, because the effect was not significant. The reason is that we displayed the steady state of the numerical solution. Moreover, the SUPG or SOLD methods using the parameter τ_S^3 led to an unstable solution and thus we used the other parameters for computations, which are not time dependent. Here, we are interested in the behavior of the numerical solution in time for which the results are not steady states. Thus, the time step affects the results more significantly, even if we employ the stabilization parameter τ_S^2, see Figure 57.

Figure 58 shows the differences of the numerical solution computed using the SOLD method for various time step sizes and for τ_S^3. We can see that the time step size has a significant influence on the quality of the numerical solution.

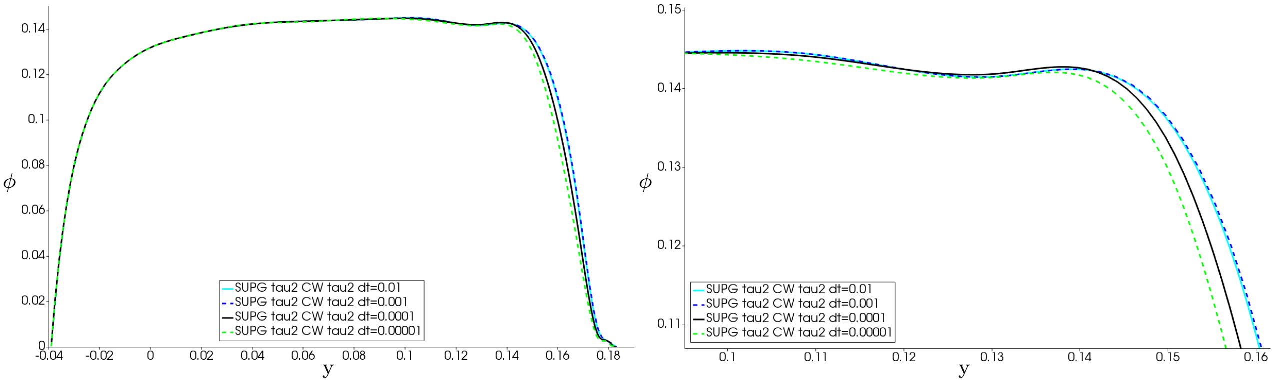


Figure 57: Numerical solution computed for various time step sizes using the SOLD method with the parameter τ_S^2 (right: zoom-in).

The results of the last experiment of this subsection are shown in Figure 59, in which we compare the stabilization techniques with some selected stabilization parameters. In this case, the section through the numerical solution is performed at $x = 0.433544$. We can repeatedly confirm the same influence of the various stabilization parameters on the resulting numerical solution. The SOLD method with the parameter $\tau_S^{2,deg}$ seems to be a convenient choice.

5.3 Summary

We studied the SUPG and SOLD methods applied for solving unsteady advection dominated advection–diffusion equation with constant coefficients on regular and irregular mesh. The most convenient technique appears to be the SOLD method which adds crosswind artificial diffusion, which suppress the spurious oscillations of the numerical solution. Higher degree spatial approximation provides positive effect, if the SOLD method is employed. However, this observation does not hold for general problem in our applications. But note that even the nonlinear schemes as AFC and FCT provide smearing of the numerical solution with increasing degree of the B-splines, see [64].

The drawback is the application to multipatch problems, which follows only C^0 continuity of the numerical solution at the interfaces. Therefore, the solution gradients are

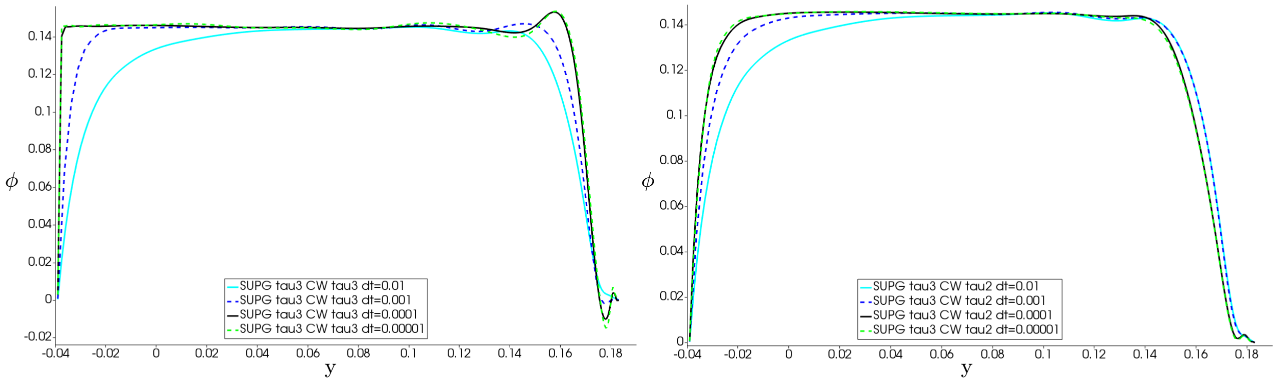


Figure 58: The numerical solution computed for various time step sizes using the SOLD method with different stabilization parameters τ_S.

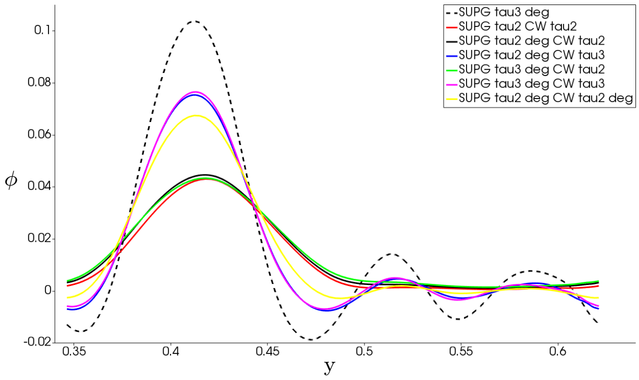


Figure 59: Section at $x = 0.433544$ using various stabilizations.

discontinuous at the interfaces. Hence, the abrupt changes in the solution near or at the interfaces usually amplify the numerical oscillations, which make the problem of numerical stabilization even more challenging.

The key component is the choice of the stabilization parameter. The most promising approach seems to be the mentioned SOLD method adding crosswind diffusion with the stabilization parameters τ_S^2, $\tau_S^{2,deg}$. Also, τ_S^3, $\tau_S^{3,deg}$ can be applied if the numerical solution does not involve discontinuities. In some cases, it is beneficial to use different variant of the parameter τ_S for SUPG and crosswind diffusion.

The employed stabilization schemes are neither monotone nor monotonicity preserving. Thus, the oscillation-free numerical solution is not guaranteed.

The mentioned observations are applied for the stabilization of the turbulence model in the following section. Certainly, the problem of advection–diffusion–reaction turbulence model with the production terms is much more complicated. We studied the stabilization techniques for the ADR equation with reaction and source terms with a constant coefficients. Also, we tested this equation with non-constant coefficients (not changing in time), which were set as the real coefficients of the k equation of the turbulence model in specific time. But the observations are similar to the experiments in this and previous sections. The stabilization parameter τ_S^4 in (118) provides negligible advantages in our applications. But the importance of this parameter is appreciated to the turbulence model, where the reaction and production terms are dominant over the advection in some parts of the computational domain. Moreover, the terms are time dependent and nonlinear. We thus show the influence of the parameter τ_S^4 for the turbulence model coupled with RANS equations in the following section.

5.4 Turbulent flow

We present some numerical results obtained by our RANS solver. The main goal of this section is the simulation of the fluid flow in the blade cascade. The time evolution of the fluid flow over the circle obstacle is also solved and the numerical solutions are presented in Appendix C. First, it is important to mention, how the boundary conditions for the turbulence model are imposed.

We set the boundary conditions according to the literature (e.g. [42, 116, 30, 31, 32]). The $k - \omega$ model is used in the thesis (primarily SST model and Wilcox LRN in some comparisons), which can be resolved up to the walls. At the inflow boundary, the turbulent kinetic energy satisfies

$$k = \frac{3}{2} (IU)^2 , \tag{136}$$

where I is the turbulence intensity and U is the mean velocity already mentioned in (7) in Section 2. The turbulence intensity is usually expected $I \leq 5\%$. $I = 4\%$ in the experiments of this section. The inlet condition for specific dissipation can be considered as

$$\omega = \frac{\sqrt{k}}{l} \tag{137}$$

using the estimation of the length scale l. An alternative is to set the turbulence viscosity ν_T at inlet such that the required eddy viscosity ratio $\frac{\nu_T}{\nu}$ is satisfied. The eddy viscosity ratio expresses the influence of the turbulent viscosity to the molecular viscosity. The relation (137) is rather recommended for the internal flows than external flows, but the viscosity ratio should be controlled to be sure that a reasonable value for ω is set. The viscosity ratio $\frac{\nu_T}{\nu} = 10$ in the experiments, unless otherwise stated.

The normal gradients of k and ω are assumed to be zero at the outflow boundary, i.e. the zero Neumann boundary condition is set. The Dirichlet boundary condition for the turbulent kinetic energy at solid walls is zero ($k = 0$). However, one problem occurs for ω at walls since (see the relation (62) in Section 3.2.3)

$$\omega = \frac{\epsilon}{k} \tag{138}$$

goes to infinity. Thus, ω is assumed to reach high limited value, which is usually problem dependent. A partially universal procedure is to apply the relation for ω derived for the wall function approach (cf. e.g. [42, 116, 30, 31, 32]), i.e.,

$$\omega = \frac{6\nu}{\beta y^2}. \tag{139}$$

To approximate the boundary condition at walls for ω, this relation is used such that y is considered as the normal distance of the first grid point to the nearest wall. It is sometimes common to approximate analogically the boundary condition for k as a small value close to zero ($k \approx 0$).

The homogeneous Dirichlet boundary condition for the mean velocity is prescribed at the walls and zero Neumann condition is set at outflow boundary. The inlet velocity boundary condition is constant with the tangent direction to the leading edge of the blade profile and magnitude 8.1786, i.e. $\bar{\mathbf{u}}_{in} = (7.765, -0.28272)$.

Two types of initial conditions are considered for velocity, k and ω in this work. First type is to set constant value in the whole internal part of the computational domain, which is zero for velocity and expanded inlet condition for k and ω in the whole interior part of the domain.

Second type is a non-constant initial condition. A Stokes problem or steady Navier–Stokes problem is solved assuming higher viscosity (i.e. laminar flow is considered for the identical setting of the boundary conditions) such that the numerical scheme converges to the steady state. The resulting numerical solution is set as the initial condition for the RANS unsteady discrete problem.

The initial condition for the turbulence model can be set analogically, i.e., a separated steady discrete problem of the turbulence model is solved iteratively until converges. For this purpose, the corresponding initial condition for the velocity field (computed in the previous step by solving Stokes or steady Navier–Stokes problem) is used for evaluation of the terms of the steady turbulence model.

An alternative is to solve only the steady Stokes/Navier–Stokes problem for the setting of the initial condition for RANS problem and to apply the mentioned constant initial condition for k and ω. The chosen initial condition is specified in the numerical examples.

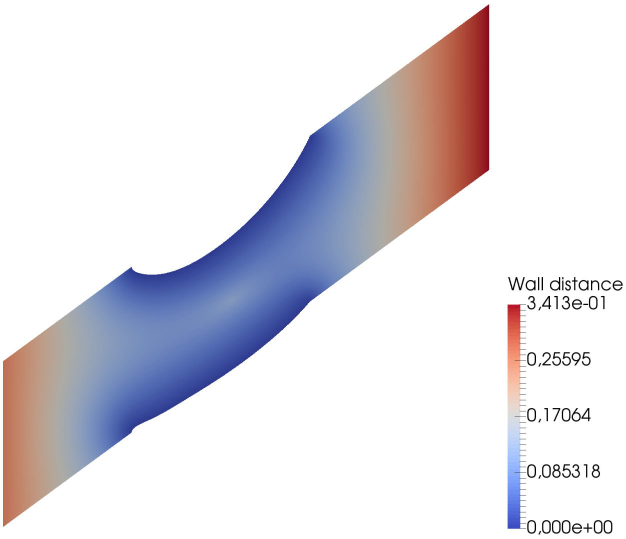


Figure 60: Approximation of the wall distance.

The discrete RANS problem is solved decoupled from the discrete turbulence model. In the first time step, the Picard RANS problem is solved until the iterations of the numerical solution of $\mathbf{u}$ and p converge (or until the maximum number of the Picard iterations is achieved as mentioned in Section 5). Then, we continue solving the discrete turbulence model such that the computed velocity and pressure solutions are used for evaluation of the turbulence model terms. This sequence is repeated in each time step.

The wall distance y of the computational nodes is required to evaluate the coefficients of the SST turbulence model. Usually, an additional differential equation is solved to approximate the distance to the nearest wall rather than search algorithms, which can be difficult to apply in complex geometries. The three well known differential equations used to compute wall distance are the Eikonal, Hamilton-Jacobi and Poisson equations.

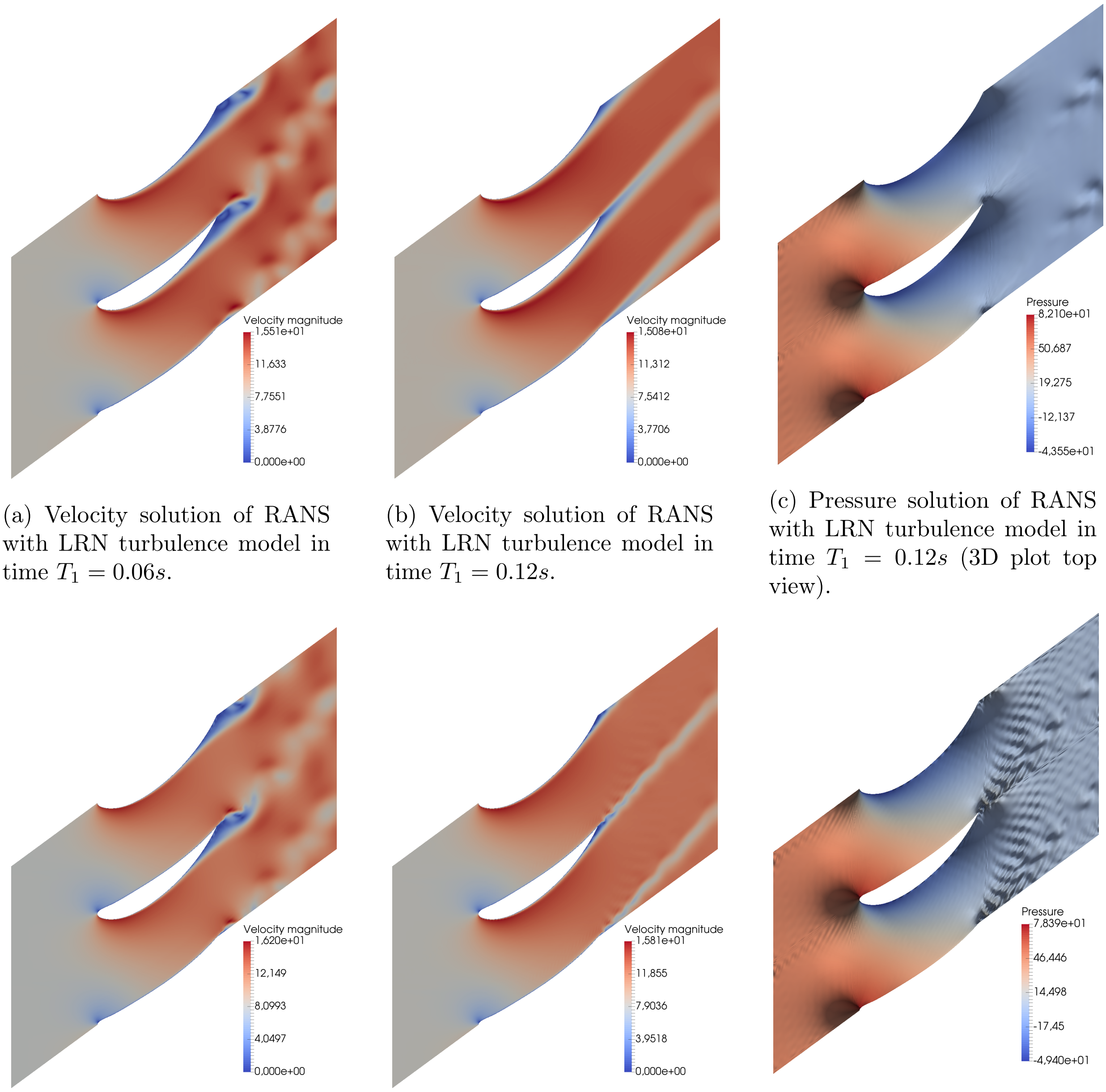
In our case, the result of the additional Poisson equation is used to approximate the wall distance, i.e.,

$$\Delta \Psi = -1, \quad \text{in } \Omega \tag{140}$$

is the Poisson equation for which the zero Dirichlet condition is set at the walls of interest and zero Neumann condition on the rest of the boundaries. The wall distance is then approximated as (cf. [113])

$$y = -\left\| \nabla \Psi \right\| + \sqrt{\left\| \nabla \Psi \right\|^2 + 2\Psi}. \tag{141}$$

The wall distance approximation is computed only once at the beginning of the RANS solver since the geometry does not change over time. The resulting wall distance is shown in Figure 60 for the blade profile geometry. Very fine grid is used to solve the Poisson equation (140) in order to get the most accurate wall distance approximation. Although the computational requirements are high using fine grid, the Poisson equation is solved only once as mentioned.



(a) Velocity solution of RANS with LRN turbulence model in time $T_1 = 0.06s$.

(b) Velocity solution of RANS with LRN turbulence model in time $T_1 = 0.12s$.

(c) Pressure solution of RANS with LRN turbulence model in time $T_1 = 0.12s$ (3D plot top view).

(d) Velocity solution of RANS with SST turbulence model in time $T_1 = 0.06s$.

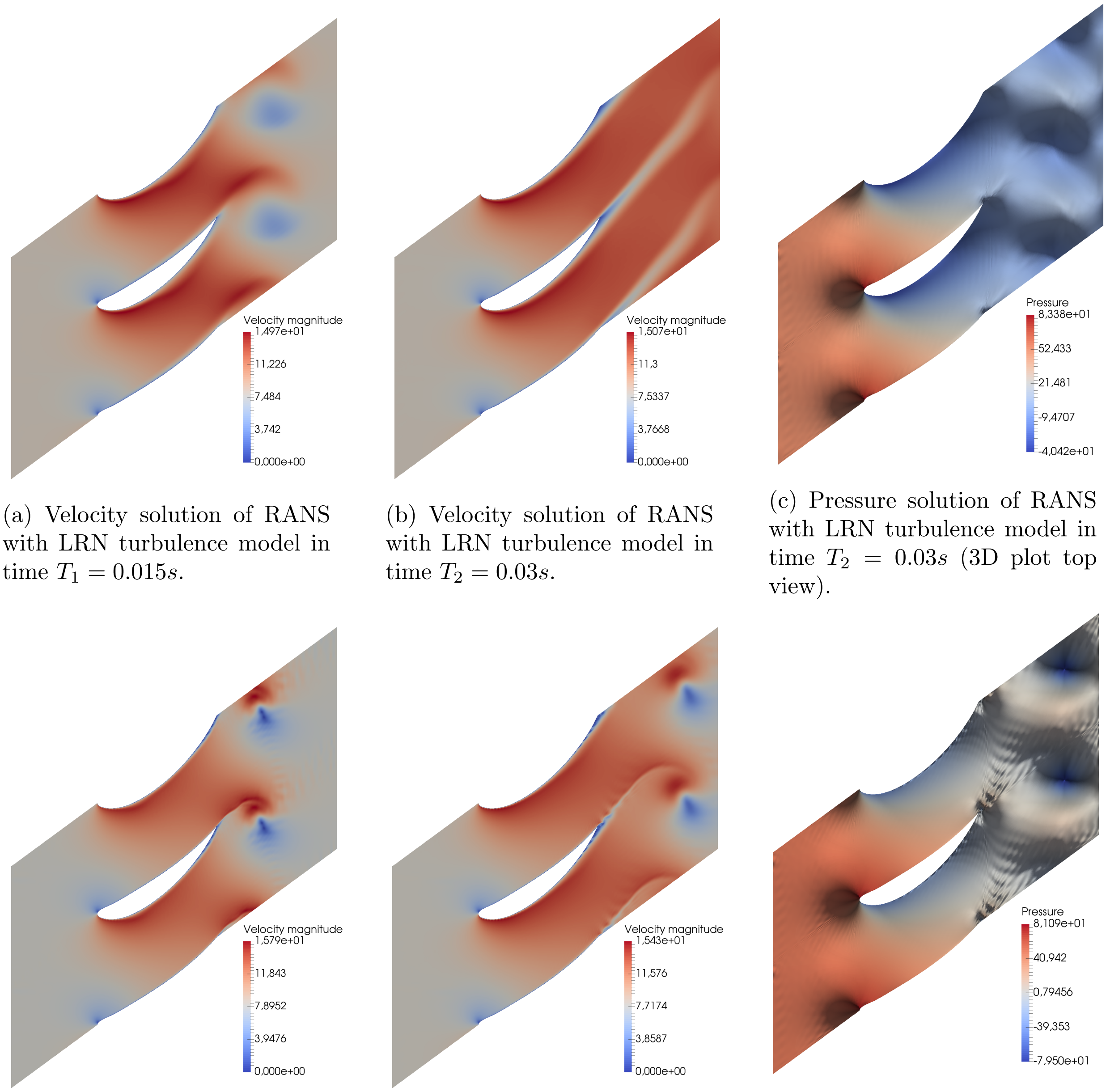
(e) Velocity solution of RANS with SST turbulence model in time $T_1 = 0.12s$.

(f) Pressure solution of RANS with SST turbulence model in time $T_1 = 0.12s$ (3D plot top view).

Figure 61: Comparison of LRN and SST turbulence models for $T_1 = 0.06s, T_2 = 0.12s$. Solutions of steady problems set as initial condition.

The computational mesh in Figure 6 is used, which satisfies $y^+ < 10$. The simulation was carried out for a fluid with viscosity $\nu = 10^{-5}$ with time step $\Delta t = 10^{-4}$.

If the stabilization techniques does not completely eliminate the spurious oscillations, special treatment is required to prevent the nonphysical negative values of the variables. A negative slope linearization or other positivity preserving techniques can be found e.g. in [69, 93]. The main idea is to limit the coefficients and source term of the linearized



(a) Velocity solution of RANS with LRN turbulence model in time $T_1 = 0.015s$.

(b) Velocity solution of RANS with LRN turbulence model in time $T_2 = 0.03s$.

(c) Pressure solution of RANS with LRN turbulence model in time $T_2 = 0.03s$ (3D plot top view).

(d) Velocity solution of RANS with SST turbulence model in time $T_1 = 0.015s$.

(e) Velocity solution of RANS with SST turbulence model in time $T_2 = 0.03s$.

(f) Pressure solution of RANS with SST turbulence model in time $T_2 = 0.03s$ (3D plot top view).

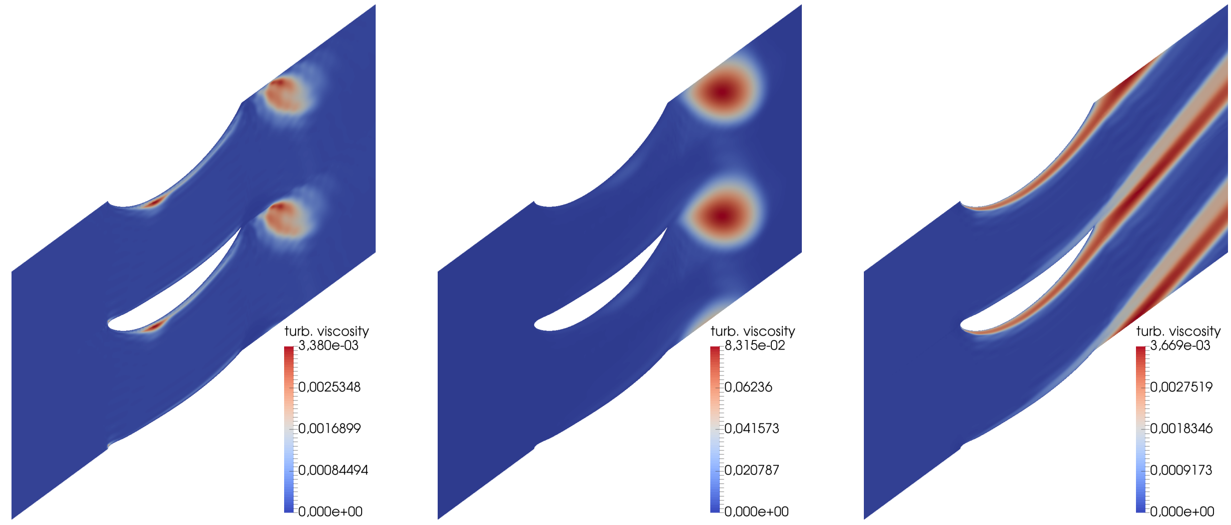
Figure 62: Comparison of LRN and SST turbulence models for $T_1 = 0.015s, T_2 = 0.03s$. Constant initial conditions.

equations from below, which is also employed in our fluid flow solver.

The LRN (Wilcox low Reynolds number $k-\omega$ turbulence model) and SST turbulence models are compared in Figures 61 - 63. The solutions of the steady problems are set as the initial condition for both RANS and turbulence model problems in Figure 61, whereas the constant initial value is prescribed in Figures 62 and 63. The SOLD method which adds crosswind artificial diffusion is employed in the rest of the experiments for the turbulence model. Parameter τ_S^4 is chosen for both the SUPG and crosswind terms in this section, unless otherwise stated.

The LRN turbulence model produces much more turbulence viscosity, see Figure 63 (middle and right). It leads to numerical solution with no or little numerical instabilities, which is more evident in the case of constant initial condition, Figures 62 (top) and 63 (middle and right). Almost identical results are obtained if the problem is solved without any stabilization techniques for LRN model, but the results are not presented here. The reason is the mentioned big amount of ν_T added to the RANS diffusion coefficient $(\nu+\nu_T)$. From the numerical point of view, the turbulence viscosity plays a role of the additional artificial diffusion. The resulting approximation of the fluid flow is laminar, which prevents the transition to turbulence.

On the other hand, the SST turbulence model produces less turbulent viscosity ν_T, which leads to better predictions of the turbulence. Also, the SST model for RANS computations is one of the most common turbulence models for the real applications in the literature. The price to be paid is the presence of the spurious oscillations, see Figures 61 and 62 (bottom) and Figure 63 (left). Moreover, the time iterations of the numerical



(a) Eddy viscosity ν_T of the SST turbulence model in time $T_1 = 0.015s$.

(b) Eddy viscosity ν_T of the LRN turbulence model in time $T_1 = 0.015s$.

(c) Eddy viscosity ν_T of the LRN turbulence model in time $T_2 = 0.12s$.

Figure 63: Comparison of ν_T of LRN and SST turbulence models for $T_1 = 0.015s, T_2 = 0.12s$. Constant initial conditions.

solution diverge if none of the stabilization approaches is employed for the SST model (as already mentioned in the motivation example in Section 5.1).

Thus, the SST $k-\omega$ turbulence model is rather used to predict the turbulence in the rest of the experiments. However, the suitable stabilization scheme has to be applied to suppress the numerical oscillations without an excessive smearing.

The influence of the SUPG and SOLD methods with various stabilization parameters used for the system of unsteady advection–diffusion–reaction $k-\omega$ equations is similar to the results obtained in Section 5.2. It means that the SUPG method reduces the unphysical oscillations only in the streamline direction and a convenient artificial diffusion has to be added. The additional crosswind diffusion also leads to most promising results for turbulence model. The stabilization parameters τ_S affect the amount of the added artificial diffusion in the same way as observed in Section 5.2. The artificial isotropic diffusion (with both parameters τ^0_{iso} and τ^1_{iso}) was also applied to the SST model for comparison, but this approach does not provide satisfactory results similarly to observations in Section 5.2. We thus focus on numerical study of the additional crosswind diffusion for the SOLD method in this section.

We presented the RANS velocity and pressure solutions for solving SST model with and without stabilization terms in Section 5.1. It follows from the results that the stabilization techniques used for turbulence model can strongly affect the RANS solutions. Hence, the choice of the stabilization parameters is also very important. Similarly to the simple advection–diffusion equation, the parameters τ^0_S and primarily τ^1_S leads to numerical results which are extremely diffusive. Also, the parameter τ^2_S is not satisfactory since the turbulence model is usually reaction dominated in some areas of the computational domain, but τ^2_S is not reaction dependent.

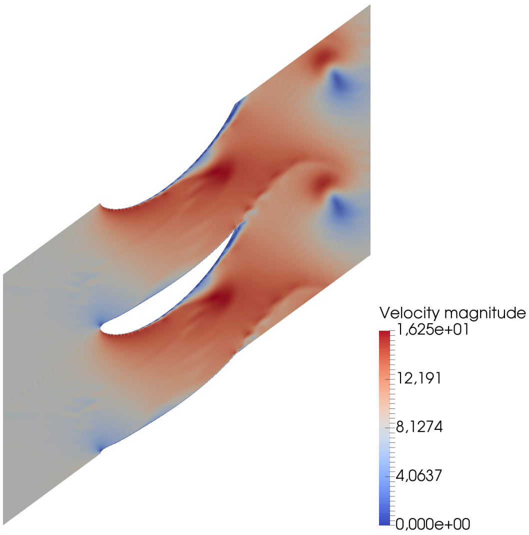


Figure 64: RANS velocity solution in time $T = 0.03s$. SOLD parameters: τ^4_S for SUPG, τ^2_S for crosswind.

Hypothetically, the mentioned stabilization parameters could be used in order to get oscillation-free solution though smeared. However, the resulting time iterations of the RANS solution diverge. According to our consideration, the reason is just the unsuitable stabilization parameter, which leads to diffusive result. This can be observed in Figures 64 and 65, where the SOLD method adding crosswind diffusion is used with τ_S^4 for SUPG and τ_S^2 for crosswind. Indeed, the reaction and production terms play an important role in the boundary layer. If too much artificial diffusion is added, the solution is excessively smeared to the areas outside the boundary layer, see the time evolution of the turbulent kinetic energy in Figure 65. The smearing spreads fast along the interface and the arised mass of

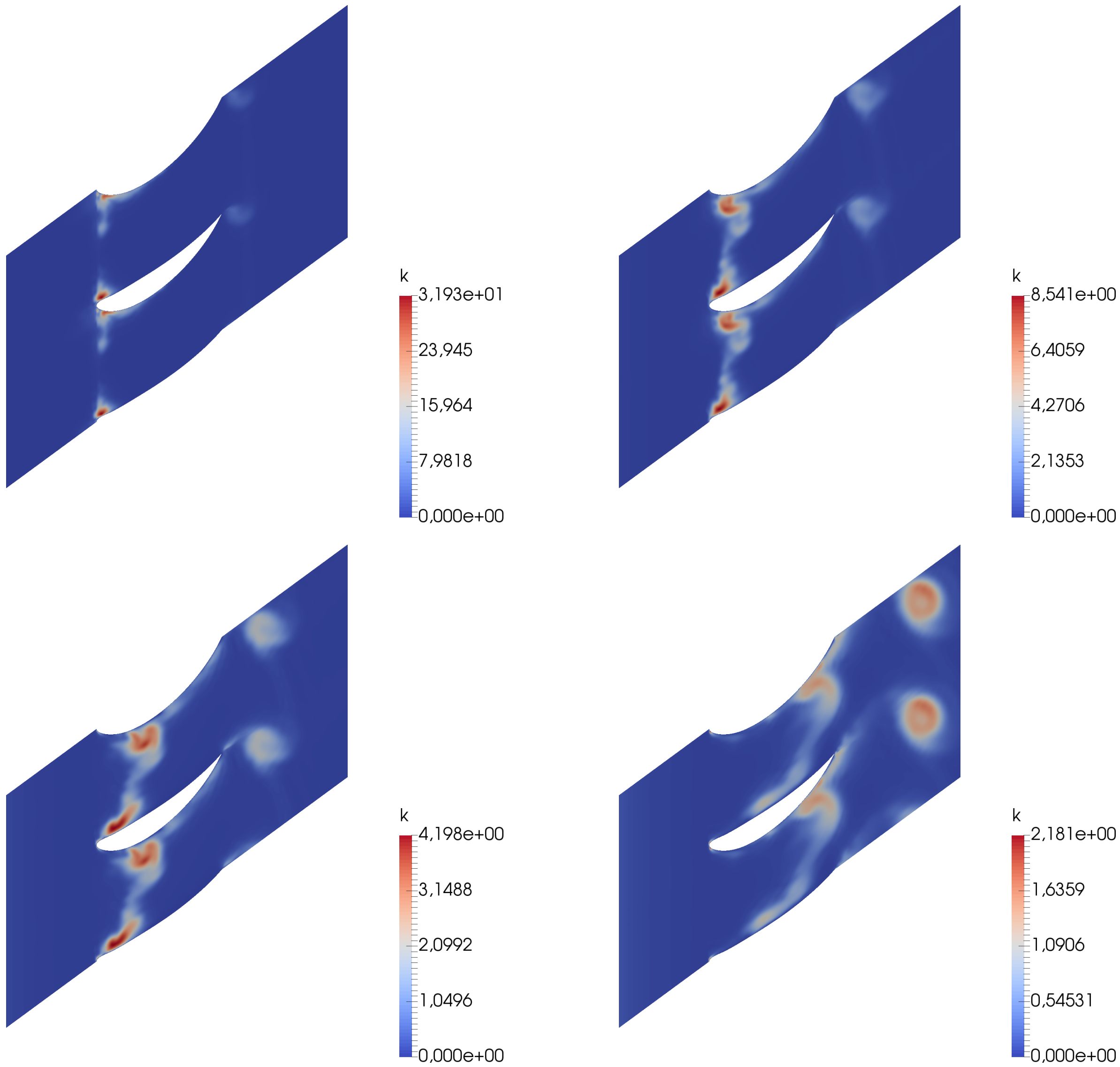


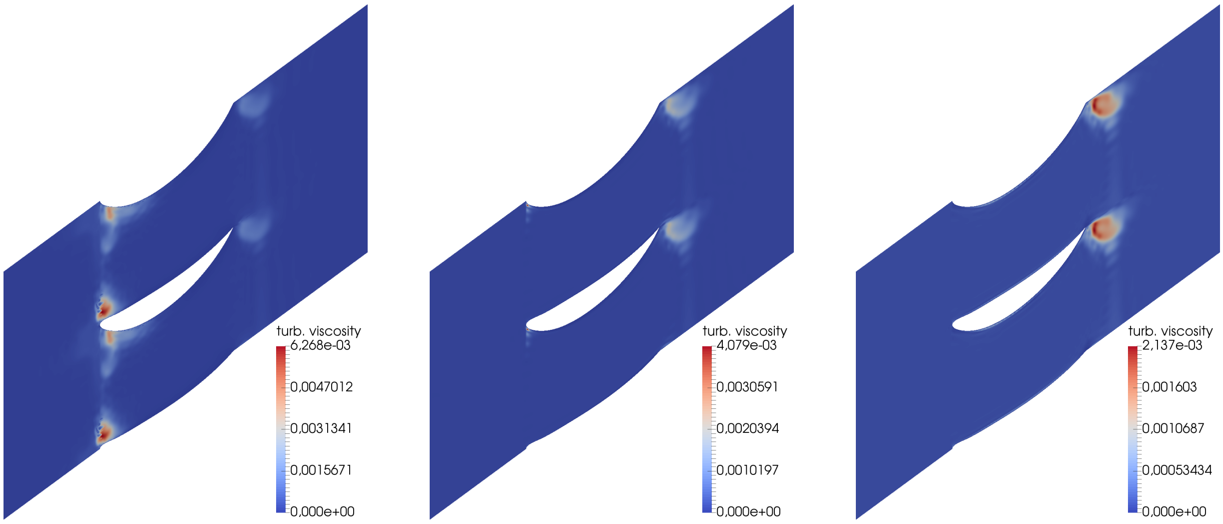
Figure 65: Time evolution of turbulence kinetic energy solution for $T_1 = 0.005s$, $T_2 = 0.01s$, $T_3 = 0.015s$, $T_4 = 0.03s$ from top left to bottom right. SOLD parameters: τ_S^4 for SUPG, τ_S^2 for crosswind.

the kinetic energy travels through the domain. At the moment, when it crosses the second interface and wake at the end of the profile, the numerical solution blows-up. If the time step size is decreased, the blow-up time is unchanged. The same observations are obtained also using other stabilization parameters, which do not take the reaction coefficient into account. It is primarily for the crosswind diffusion, which smears the solution near the leading edge of the blade profile in the orthogonal direction to the wall. If only SUPG stabilization is used for the turbulence model, the time iterations of the solution does not diverge although the numerical solution is polluted by the spurious oscillations all over the domain mainly in the crosswind direction.

Figure 66 compares the distribution of the turbulent kinetic energy for various stabilization parameters for the SOLD method. We can see the importance of the parameter dependence on the reaction term. Nevertheless, the remaining numerical instabilities arises from behavior of the numerical solution along the interfaces as presented in Section 5.2.1 and apparently from the insufficiently refined mesh (such that the elements are now very narrow, curvilinear and irregular) near the walls.

To sum up, the stabilization approaches are needed for the discrete Galerkin problem of the SST turbulence model to get non-divergent time iterations of the numerical solution. The requirement is a stabilization scheme which does not introduce too much artificial diffusion primarily in the crosswind direction in the regions with dominated reaction term. Hence, the SOLD method adding crosswind diffusion with parameter τ_S^4 is recommended despite the remaining spurious oscillations which arises from the numerical difficulties along the interfaces (where the order of the solution continuity is decreased to C^0).

A drawback is that the numerical instabilities in the k and ω solutions propagate into



(a) SOLD parameters: τ_S^4 for SUPG, τ_S^2 for crosswind.

(b) SOLD parameters: τ_S^4 for SUPG, τ_S^3 for crosswind.

(c) SOLD parameters: τ_S^4 for SUPG, τ_S^4 for crosswind.

Figure 66: Comparison of ν_T of the SST turbulence models in time $T = 0.005s$ for various SOLD parameters.

the RANS velocity and pressure solutions. Thus, to suppress the remaining unphysical oscillations, a stabilization terms are required to be added also to the discrete RANS problem. Unfortunately, the results employing the SUPG or SOLD approaches for RANS equations are not satisfactory in our applications to the flow in blade cascade so far. Only the I-SUPG and IT-SUPG stabilizations (introduced in Section 4.3.3) provide non-divergent time iterations of the RANS solution.

Considering the results of the advection–diffusion equation in Section 5.2.1, the I-SUPG term (120) introduces equally too much numerical diffusion in application to RANS equa-

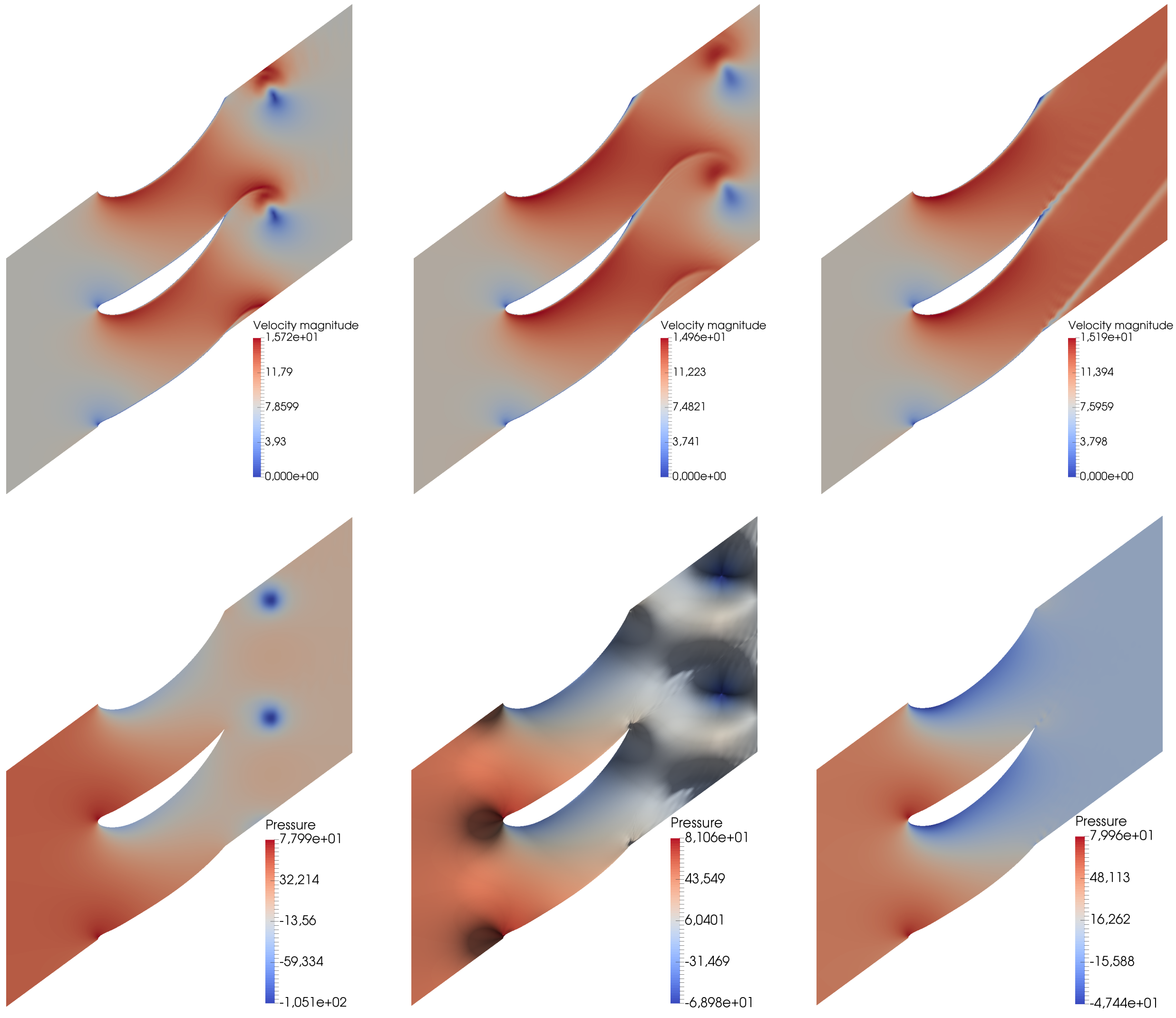


Figure 67: Time evolution of RANS velocity (upper row) and pressure (bottom row) solutions using SST turbulence model for $T_1 = 0.015s$ (left col), $T_2 = 0.03s$ (middle col), $T_3 = 0.12s$ (right col). SOLD method (SUPG with additional crosswind) for SST model and IT-SUPG method for RANS equations. Stabilization parameters: τ_S^4 for both SUPG and crosswind, τ_S^3 for IT-SUPG, (3D plot of the middle bottom result: top view).

tions. The velocity and pressure approximations are smeared similarly to the results shown in Figure 62 for the LRN turbulence model and hence they are not displayed. On the other hand, the IT-SUPG method provides satisfactory results, see Figure 67. Compared to the RANS solutions in Figure 62 (bottom row), almost all the numerical oscillations are eliminated, which is most evident for the RANS pressure solution. The remaining spurious oscillations again arises from the C^0 continuity across the interfaces. Although the inconsistent method was applied, the resulting RANS solution are only slightly smeared (see the pressure minimum value in time $T = 0.03s$).

Several other numerical experiments for different stabilization parameters τ_S of the SOLD method are shown in Appendix C.

The spurious oscillations can be also reduced to some extent, if we set higher value of the eddy viscosity ratio $\frac{\nu_T}{\nu}$ at the inlet boundary. As a consequence, the resulting turbulent viscosity has globally higher value in the whole computational domain. In our consideration, the locally added artificial diffusion introduced from the mentioned stabilization schemes is rather used than to globally increase the diffusion coefficient.

6 Conclusions and future work

Isogeometric analysis is a very powerful tool for the analysis of partial differential equations and optimization processes since it gives the possibility to represent the geometry exactly. Moreover, it allows to avoid the time-consuming step of mesh generation and manual treatment of the mesh refinement. We have presented numerical investigations for IgA applications to unsteady advection/reaction dominated problems. A characteristic feature of numerical solutions of the mentioned problems is the presence of sharp layers, where the solution gradients are very large. Then the continuous Galerkin discretizations give rise to unwanted nonphysical oscillations.

This thesis project resulted in the extension of the common stabilization approaches for finite elements to higher order basis functions on multipatch domains with general curvilinear meshes including local refinement (within the meaning of tensor product refinement).

The SUPG and SOLD methods have been applied to suppress the spurious oscillations without excessive smearing of the numerical solution. The choice of the stabilization parameters is the most challenging part of the stabilization techniques. Numerical results reveal their capabilities in solving unsteady advection–diffusion equations and RANS equations closed with SST turbulence model in highly advective situations. Special effort has been devoted to the problems on multipatch domains with irregular and curvilinear meshes.

Higher degree spatial approximation provides positive effect, if the SOLD method is employed. However, this observation does not hold for all our numerical experiments. The drawback is the application to the multipatch problems, which follows only C^0 continuity of the numerical solution across the interfaces. Therefore, the solution gradients are discontinuous at the patch boundaries. Hence, the abrupt changes in the solution along the interfaces usually amplify the numerical oscillations, which makes the problem of numerical stabilization even more challenging.

For all the numerical experiments, the SOLD method which adds the artificial crosswind diffusion appears to be the most convenient technique with an appropriate choice of the stabilization parameter. It is hard to suggest an optimal parameter for general advection–diffusion equation and even more complicated is the choice of the parameter for complex problems. According to our observations, we recommend parameter with dependency on all the coefficients of the given equation.

The importance of the stabilization parameter with reaction dependency is significant for the turbulence model, where the reaction terms are dominant over the advection in some parts of the computational domain. This parameter should be used especially in the case of SOLD method application despite the remaining spurious oscillations, which mostly arises from the numerical difficulties along the interfaces (where the order of the solution continuity is decreased to C^0). As a consequence, the numerical instabilities of the solution of the turbulence model propagate into the resulting RANS solution. The IT-SUPG stabilization method provides satisfactory results, but the SUPG and SOLD techniques for RANS equations leads to unstable solutions in our applications so far. Thus, the stabilization techniques for the RANS equations should be investigated in more detail.

The numerical solutions of the advection dominated problems were obtained from an in-house IgA solver implemented in C++ within a framework of the G+Smo library.

Since the spatial discretization of the Navier–Stokes and RANS equations using an implicit time discretization results in nonsymmetric linear systems of saddle-point type, employment of the iterative methods with a good choice of preconditioner for the whole system is necessary for industrial application, which is not a trivial task. Thus, the direct solver with LU decomposition is applied for the time being. But the efficiency is a significant limitation of the current code and hence only two dimensional problems were studied.

The work of this thesis is motivated by the problem of automatic shape optimization of runner blades in water turbines in order to improve utility quantities as the turbine efficiency. The flow in the hydraulic turbine is assumed to be very fast and thus the turbulent behavior of the flow is expected. Moreover, if we are interested in shape optimization, the flow near boundary must be properly simulated to get reasonable solution.

Considering the above goal, it is necessary to improve the numerical simulation of the turbulent flow, especially near the boundary. The near wall treatment is very problematic and important issue. The turbulence models are resolved up to the walls, since the wall function approach is not employed so far. A drawback is very fine grid near the solid walls. In the future work, either we have to improve the local refinement near the walls using special refinement techniques, e.g. truncated hierarchical B-splines, or the wall function approach has to be integrate into our fluid flow solver. For the same reasons, the mesh along the wake flow past the blade profile deserves more care. But without the true local refinement (not tensor product knot insertion) techniques, the treatment of the long narrow elements has to be covered.

The length of the element diagonal was chosen for evaluation of the stabilization parameters and compared with the maximum size length of the element. The alternatives suitable for the isogeometric analysis applied on advection/reaction dominated problems will be studied in the future work.

Further research will be also concentrated on AFC and FCT nonlinear stabilization techniques for high order B-splines. FCT techniques are mostly restricted to the linear finite elements. There exist extensions to higher order finite elements, but its implementation is very hard due to negative function values of the finite element bases. Isogeometric analysis is not affected by this problem since B-splines are all non-negative, which makes IgA a perfect tool for extending the FCT framework to higher order functions. The low-order AFC method has been already implemented into the framework of the G+Smo library, but certainly the resulting numerical solution is excessively smeared. The implementation of the high-resolution FCT scheme needs to be tackled.

Application of different time discretization schemes together with the SOLD methods should be also integrated into our flow solver to get higher order accuracy of the entire space-time discretization.

All of the experiments demonstrated the numerical difficulties along the multipatch interfaces. The rate of the solution change and the direction of the flow crossing the interface extremely affect the numerical oscillations in this areas. In order to handle the decrease in the solution approximation continuity across the interfaces, the discontinuous Galerkin technique is recommended to be employed. The discontinuities are allowed in the numerical solution at the patch boundaries, which can provide advantage since the solution derivatives and the turbulence viscosity are generally discontinuous at the interfaces. This combination of the IgA and DG approaches starts to be frequently used for the compressible flows, but we believe in favourable results also for the incompressible flow. Moreover, DG method is capable to accommodate the non-conforming meshes which are frequently employed in multipatch domains.

References

[1] D. Arndt, G. Lube: *FEM with local projection stabilization for incompressible flows in rotating frames*, NAM-Preprint, 2015.

[2] M. Augustin, A. Caiazzo, A. Fiebach, J. Fuhrmann, V. John, A. Linke, R. Umla: *An assessment of discretizations for convection–dominated convection–diffusion equations*, Comput. Methods Appl. Mech. Engtg., Vol, 200, pp. 3395-3409, 2011.

[3] G. R. Barrenechea, V. John, P. Knobloch: *A local projection stabilization finite element method with nonlinear crosswind diffusion for convection-diffusion-reaction equations*, ESAIM Mathematical Modelling and Numerical analysis, Vol. 47, Issue 5, pp. 1335-1366, 2013.

[4] T. Barth, R. Herbin, M. Ohlberger: *Finite volume methods: foundation and analysis*, Encyclopedia of Computational Mechanics Second Edition, Wiley, 2017.

[5] F. Bassi, S. Rebay: *A high order discontinuous Galerkin method for compressible turbulent flows*, Lecture Notes in Computational Science and Engineering, Vol. 11, pp. 77-88, 2000.

[6] Y. Bazilevs, V. M. Calo, J.A. Cottrell, T. J. R. Hughes, A. Reali, G. Scovazzi: *Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows*, Compup. Methods Appl. Mech. Engrg., Vol. 197, pp. 173-201, 2007.

[7] Y. Bazilevs, T.J.R. Hughes: *Weak imposition of Dirichlet boundary conditions in fluid mechanics*, Computers and Fluids, Vol. 36, pp. 12-26, 2007.

[8] Y. Bazilevs, C. Michler, V.M. Calo, T.J.R. Hughes: *Weak Dirichlet boundary conditions for wall-bounded turbulent flows*, Computer Methods in Applied Mechanics and Engineering, Vol. 196, pp. 4853-4862, 2007.

[9] M. Benzi, G. Golub, J. Liesen: *Numerical solution of saddle point problems*, Acta Numerica, Vol. 14, pp. 1-137, 2005.

[10] L.C. Berselli, T. Iliescu, W.J. Layton: *Mathematics of Large Eddy Simulation of turbulent flows*, Springer, Germany, 2006.

[11] R.de Borst, S.J. Hulshoff, S. Lenz, E.A. Munts, H.van Brummelen, W.A. Wall: *Multiscale methods in computational fluid and solid mechanics*, European Conference on Computational Fluid Dynamics, ECCOMAS CFD, 2006.

[12] M. Braack, E. Burman, V. John, G. Lube: *Stabilized finite element methods for the generalized Oseen problem*, Comput. Methods Appl. Mech. Engrg., Vol. 196, pp. 853-866, 2007.

[13] P. Bradshaw: *An introduction to turbulence and its measurement*, Pergamon Press, Oxford, 1971.

[14] J. Bredberg: *On the wall boundary condition for turbulence models*, Chalmers University of Technology, Internal Report, Sweden, 2000.

[15] A. Bressan, G. Sangalli: *Isogeometric discretizations of the Stokes problem: stability analysis by the macroelement technique*, IMA Journal of Numerical Analysis, Vol. 33, Issue 2, pp. 629-651, 2013.

[16] A.N. Brooks, T.J.R. Hughes: *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering, Vol. 32, pp. 199-259, 1982.

[17] M. Burger: *Numerical methods for incompressible flow*, Lecture Notes, UCLA.

[18] E. Burman, A. Ern: *Nonlinear diffusion and discrete maximum principle for stabilized Galerkin approximations of the convection–diffusion–reaction equation*, Comput. Methods Appl. Mech. Engrg., Vol. 191, Issue 35, pp. 3833-3855, 2002.

[19] E. Burman, P. Hansbo: *Edge stabilization for Galerkin approximations of convection–diffusion–reaction problems*, Comput. Methods Appl. Mech. Engrg., Vol. 193, Issues 15-16, pp. 1437-1453, 2004.

[20] I. B. Celik: *Introductory turbulence modeling*, Lecture Notes, West Virginia University, 1999.

[21] B. Cockburn: *Discontinuous Galerkin methods for convection dominated problems*, Lecture Notes, University of Minnesota, USA, 1999.

[22] B. Cockburn, G. Kanschat, D. Schoetzau: *Local discontinuous Galerkin methods for the Oseen equations*, Mathematics of Computation, Vol. 73, No. 246, pp. 569-593, 2003.

[23] B. Cockburn, G. Kanschat, D. Schoetzau, C. Schwab: *Local discontinuous Galerkin methods for the Stokes system*, SIAM J. Numer. Anal., Vol. 40, pp. 319-343, 2002.

[24] R. Codina: *A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection-diffusion equation*, Computer Methods in Applied Mechanics and Engineering, Vol. 110, Issues 3-4, pp. 325-342, 1993.

[25] R. Codina: *Comparison of some finite element methods for solving the diffusion-convection-reaction equation*, Computer Methods in Applied Mechanics and Engineering, Vol. 156, pp. 185-210, 1998.

[26] R. Codina: *On stabilized finite element methods for linear system of convection-diffusion-reaction equations*, Computer Methods in Applied Mechanics and Engineering, Vol. 188, pp. 61-82, 2000.

[27] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs: *Isogeometric Analysis: Toward Integration of CAD and FEA*, John Wiley & Sons, Ltd, 2009.

[28] Y. Chai, J. Ouyang: *Appropriate stabilized Galerkin approaches for solving two-dimensional coupled Burgers' equations at high Reynolds numbers*, Computers and Mathematics in Application, in Press, 2019.

[29] J.C. Chrispell, V.J. Ervin, E.W. Jenkins: *A fractional step θ-method for convection-diffusion problems*, Clemson University, Clemson, 2008.

[30] P.A. Davidson: *Turbulence: An introduction for scientists and engineers*, Oxford University Press, 2004.

[31] L. Davidson: *An introduction to turbulence models*, Chalmers University of Technology, Sweden, 2011.

[32] L. Davidson: *Fluid mechanics, turbulent flow and turbulence modeling*, Lecture Notes in MSc courses, Chalmers University of Technology, Sweden, 2013.

[33] S. Deck: *Recent improvements in the zonal Detached Eddy Simulation (ZDES) formulation*, Theor. Comput. Fluid Dyn., Vol. 26, pp. 523-550, 2012.

[34] H. C. Elman, D.J. Silvester, A. J. Wathen: *Iterative methods for problems in computational fluid dynamics*, Report CS-TR-3675, UMIACS-TR-96-58, 1996.

[35] H. C. Elman, D.J. Silvester, A. J. Wathen: *Finite Elements and Fast Iterative Solvers*, With Applications in Incompressible Fluid Dynamics, Second Edition, Oxford University Press, 2014.

[36] V.J. Ervin, E.W. Jenkins: *The LBB Condition for the Taylor-Hood P2 – P1 and Scott-Vogelius P2 – discP1 Element Pairs in 2-D*, Technical Report, Clemson University, USA, 2011.

[37] M. Feistauer: *Mathematical methods in fluid dynamics*, Longman Scientific & Technical, Harlow, 1993.

[38] T.P. Fries, H.G. Matthies: *A review of Petrov–Galerkin stabilization approaches and an extension to meshfree methods*, Informatik-Bericht Nr. 2004-01, Technical University Braunschweig, Germany, 2004.

[39] E. Furbo, J. Harju, H. Nilsson: *Evaluation of turbulence models for prediction of flow separation at a smooth surface*, Report in scientific computing advanced course, Uppsala Universitet, Sweden, 2009.

[40] E. FURBO: *Evaluation of RANS turbulence models for flow problems with significant impact of boundary layers*, Master thesis, Uppsala Universitet, Sweden, 2010.

[41] T. GELHARDA, G. LUBEA, M.A. OLSHANSKIIB, J.-H. STARCKEA: *Stabilized finite element schemes with LBB-stable elements for incompressible flows*, Journal of Computational and Applied Mathematics, Vol. 177, pp. 243-267, 2005.

[42] J.F. GERBEAU, C. FARHAT: *The finite element method for fluid mechanics*, Standford University, 2009.

[43] F. F. GRINSTEIN, L.G. MARGOLIN, W.J. RIDER: *Implicit Large Eddy Simulation*, Cambridge University Press, 2007.

[44] G. GUENNEBAUD, B. JACOB, ET AL.: Eigen v3, `http://eigen.tuxfamily.org`, 2010.

[45] C. HIRSCH: *Numerical computation of internal and external flows*, Volume 1 Fundamentals of Computational Fluid Dynamics, Second Edition, Elsevier, 2007.

[46] J. HOFFMAN, C. JOHNSON: *Blow up of incompressible Euler solutions*, Numerical Mathematics, 2006.

[47] J. HOFFMAN, C. JOHNSON: *Computational turbulent incompressible flow: applied mathematics body and soul*, Volume 4, Springer, Berlin, 2006.

[48] J. HOFFMAN, C. JOHNSON: *Resolution of d'Alembert Paradox*, Journal of Mathematical Fluid Mechanics, Vol. 12, pp. 321-334, 2010.

[49] H. HOTEIT, P. ACKERER, R. MOSÉ, J. ERHEL, B. PHILIPPE: *New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes*, Research Report RR-4491, 2002.

[50] T.J.R. HUGHES, J.A. COTTRELL, Y. BAZILEVS: *Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry, and Mesh Refinement:*, Computer Methods in Applied Mechanics and Engineering, Vol. 194, pp. 4135-4195, 2005.

[51] T.J.R. HUGHES, G. SCOVAZZI, L.P. FRANCA: *Multiscale and stabilized methods*, Encyclopedia of Computational Mechanics, 2004.

[52] R.I. ISSA: *Solution of the implicitly discretised fluid flow equations by operator–splitting*, J. Computational Phys., Vol. 62, pp. 40-65, 1985.

[53] R.I. ISSA, A.D. GOSMAN, A.P. WATKINS: *The computation of compressible and incompressible recirculating flows by a non–iterative implicit scheme*, J. Computational Phys., Vol. 62, pp. 66-82, 1986.

[54] F. JEŽEK: *Geometrické a Počítačové Modelování*, Pomocný učební text, ZČU, Plzeň, 2009.

[55] A. Jha, V. John: *A study of solvers for nonlinear AFC discretizations of convection–diffusion equations*, Computers and Mathematics with Applications, Vol. 78, Issue 9, pp. 3117-3138, 2019.

[56] V. John, S. Kaya: *A finite element variational multiscale method for the Navier–Stokes equations*, Journal on Scientific Computation, Vol. 26, Issue 5, pp. 1485-1503, 2005.

[57] V. John, P. Knobloch: *A comparison of spurious oscillations at layers diminishing (SOLD) methods for convection–diffusion equations – PART I*, Preprint Nr. 156, FR 6.1 - Mathematik, Universität des Saarlandes, Saarbrück, 2005.

[58] V. John, P. Knobloch: *A computational comparison of methods diminishing spurious oscillations in finite element solutions of convection–diffusion equations*, In: J.Chleboun, K. Segeth, T. Vejchodský (eds.) Proceedings of the International Conference Programs and Algorithms of Numerical Mathematics 13, Academy of Science of the Czech Republic, pp. 122-136, 2006.

[59] V. John, P. Knobloch: *On spurious oscillations at Layers Diminishing (SOLD) Methods for Convection–Diffusion Equations: Part I – A review*, Comput. Methods Appl. Mech. Engrg., Vol. 196, pp. 2197-2215, 2007.

[60] V. John, P. Knobloch: *On spurious oscillations at layers diminishing (SOLD) methods for convection–diffusion equations: Part II – Analysis for P_1 and Q_1 finite elements*, Comput. Methods Appl. Mech. Engrg., Vol. 197, Issues 21-24, pp. 1997-2014, 2008.

[61] V. John, P. Knobloch, S. B. Savescu: *A posteriori optimization of parameters in stabilized methods for convection–diffusion problems - Part I*, Comput. Methods Appl. Mech. Engrg., Vol. 200, pp. 2916-2929, 2011.

[62] V. John, E. Schmeyer: *Finite element methods for time-dependent convection–diffusion–reaction equations with small diffusion*, Comput. Methods Appl. Mech. Engrg., Vol. 198 (3), pp. 475-494, 2008.

[63] V. John, L. Schmumacher: *A study of isogeometric analysis for scalar convection–diffusion equations*, Applied Mathematics Letters, Vol. 27, pp. 43-48, 2014.

[64] A. Jaeschke: *Isogeometric analysis for compressible flows with applicationin turbomachinery*, Master thesis, TU Delft, Netherlands, 2015.

[65] G. Kalitzin, G. Medic, G. Iaccarino, P. Durbin: *Near–wall behaviour of RANS turbulence models and implications for wall functions*, Journal of Computational Physics, Vol. 204, pp. 265-291, 2005.

[66] C.M. Klaij, C. Vuik: *SIMPLE–type preconditioners for cell–centered, colocated finite volume discretization of incompressible Reynolds–Averaged Navier–Stokes equations*, Int. J. Numer. Meth Fluids, Vol. 17, pp. 830-849, 2013.

[67] M. Kozubková: *Modelování proudění tekutin*, Technická univerzita Ostrava, 2008.

[68] D. Kuzmin, S. Turek: *Flux correction tools for finite elements*, Journal of Computational Physics, Vol. 175, Issue 2, pp. 525-558, 2002.

[69] D. Kuzmin, R. L"ohner S. Turek: *Flux-Corrected Transport: Principles, Algorithms, and Applications*, Second Edition, Scientific Computation, Springer, 2012.

[70] D. Kuzmin, M. Möller, S. Turek: *High-resolution FEM-FCT schemes for multidimensional conservation laws*, Comput. Methods Appl. Mech. Engrg., Vol. 193, pp. 4915-4946, 2004.

[71] D. Kuzmin, O. Mierka: *On the implementation of the $k - \epsilon$ turbulence model in incompressible flow solvers based on a finite element discretization*, University of Dortmund, Germany, 2006.

[72] D. Kuzmin: *Explicit and implicit FEM-FCT algorithms with flux linearization*, Journal of Computational Physics, Vol. 228, Issue 7, pp. 2517-2534, 2009.

[73] D. Kuzmin: *Linearity-preserving flux correction and convergence acceleration for constrained Galerkin schemes*, Journal of Computational and Applied Mathematics, Vol. 236, Issue 9, pp. 2317-2337, 2012.

[74] H. P. Langtangen, K.-A. Mardal, R. Winther: *Numerical Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics (No. 31), Cambridge University Press, 2004.

[75] M.G. Larson, F. Bengzon: *The Finite Element Method: Theory, Implementation, and Practice*, Springer, 2010.

[76] R. Li, Q. Wu, S. Zhu: *Proper orthogonal decomposition with SUPG-stabilized isogeometric analysis for reduced order modelling of unsteady convection-dominated convection-diffusion-reaction problems*, Journal of Computational Physics, Vol. 387, pp. 280-302, 2019.

[77] J. Liesen, Z. Strakoš: *Krylov subspace methods: principles and analysis*, Numerical Methods and Scientific Computation, Oxford University Press, Oxford, 2013.

[78] C. Lohmann, D. Kuzmin, J.N. Shadid, S. Mabuza: *Flux-corrected transport algorithms for continuous Galerkin methods on high order Bernstein finite elements*, Journal of Computational Physics, Vol. 344, pp. 151-186, 2017.

[79] K.A. Mardal, H.P. Langtangen: *Mixed Finite Elements*, Advanced Topics in Computational Partial Differential Equations - Numerical Methods and Diffpack Programming, Springer, 2003.

[80] L.G. Margolin: *Modeling Turbulent Flow with Implicit LES*, In Proceedings of the Joint Russian–American Five Laboratory Conference on Computational Mathematics/Physics, Vienna, Austria, 2005.

[81] A. Mantzaflaris and others (see website): G+Smo (Geometry plus Simulation modules) v0.8.1, `http://github.com/gismo`, 2018.

[82] J. M. McDonough: *Introductory Lectures on Turbulence Physics, Mathematics and Modeling*, University of Kentucky, 2007.

[83] S. Míka, P. Přikryl, M. Brandner: *Speciální Numerické Metody: Numerické Metody řešení okrajových úloh pro diferenciální rovnice:* 1. vyd. Plzeň, Vydavatelský servis, 2006.

[84] P. Moin, K. Mahesh: *Direct numerical simulation: A tool in turbulence research*, Annu. Rev. Fluid Mech., Vol. 30, pp. 539-578, 1998.

[85] B. Mohammadi, O. Pironneau: *Analysis of the k-epsilon turbulence model* Wiley, New York, 1993.

[86] M. Möller: *Algebraic flux correction for nonconforming finite element discretizations of scalar transport problems:* Computing, Vol 95, Issue 5, pp. 425-448, 2013.

[87] A. Montlaur: *High–order discontinuous Galerkin methods for incompressible flows:* Doctoral thesis, Escola Politècnica Superior de Castelldefels, Barcelona, Spain, 2009.

[88] F. Moukalled, L. Mangani, M. Darwich: *The finite volume method in computational fluid dynamics:*, An advanced introduction with OpenFOAM and MATLAB, Springer, 2016.

[89] P.N. Nielsen, A. R. Gersborg, J. Gravesen, N. L. Pedersen: *Discretizations in isogeometric analysis of Navier–Stokes flow*, Comp. Meth. in Appl. Mech. and Eng., Vol. 200, Issues 45-46, pp. 3242-3253, 2011.

[90] R. H. Nichols: *Turbulence Models and Their Application to Complex Flows*, University of Alabama at Birmingham, Revision 4.01, 2014.

[91] H. Nishikawa: *First, second, and third order finite–Vvolume schemes for advection–diffusion*, 21st AIAA Computational Fluid Dynamics Conference, 24 - 27 June, San Diego, California, American Institute of Aeronautics and Astronautics Paper AIAA, pp. 2013-2568, 2013.

[92] K. Nordanger, R. Holdahl, A. M. Kvarving, A. Rasheed, T. Kvamsdal: *Implementation and comparison of three isogeometric Navier–Stokes solvers applied to simulation of flow past a fixed 2D NACA0012 airfoil at high Reynolds number*, Comput. Methods Appl. Mech. Engrg., Vol. 284, pp. 664-688, 2015.

[93] S.V. Patankar: *Numerical Heat Transfer and Fluid Flow*, McGraw–Hill, New York, 1980.

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

[95] J.W. Peterson, A.D. Lindsay, F. Kong: *Overview of the Incompressible Navier–Stokes simulation capabilities in the MOOSE Framework*, Advances in Engineering Software, Vol. 119, pp. 68-92, 2018.

[96] O. Petit: *Towards Full Predictions of the Unsteady Incompressible Flow in Rotating Machines, Using OpenFOAM*, Ph.D. thesis, Chalmers University of Technology, Sweden, 2012.

[97] L. Piegl, W. Tiller: *The NURBS book*, Springer Verlag, Berlin, 1997.

[98] A. Quarteroni, A. Valli: *Numerical Approximation of Partial Differential Equations*, Springer, Germany, 1994.

[99] M. ur Rehman: *Fast iterative methods for incompressible Navier–Stokes equations*, Ph.D. thesis, Delft University of Technology, Netherlands, 2010.

[100] M. ur Rehman, T. Geenen, C. Vuik, G. Segal, S.P. MacLachlan: *On iterative methods for the incompressible Stokes problem*, Int. J. Numer. Meth. Fluids, Vol. 65, pp. 1180-1200, 2011.

[101] Y. Saad: *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.

[102] S. Schmidt, V. Schulz: *Shape derivatives for general objective functions and the incompressible Navier–Stokes equations*, Control and Cybernetics, Vol. 39, No. 3, pp. 677-713, 2010.

[103] A. Segal, M. ur Rehman, C. Vuik: *Preconditioners for incompressible Navier–Stokes solvers*, Numer. Math. Theor. Meth. Appl., Vol. 3, No. 3, pp. 245-275, 2010.

[104] A. Segal: *Finite Element Methods for the Incompressible Navier–Stokes Equations*, Burgerscentrum, Research School for Fluid Mechanics, Delft University of Technology, 2012.

[105] A. Smith, D. Silvester: *Implicit algorithms and their linearization for the transient incompressible Navier–Stokes equations*, IMA Journal of Numerical Analysis, Vol. 17, pp. 527-545, 1997.

[106] J. Štigler: *Analytical velocity profile in tube for laminar and turbulent flow*, Engineering Mechanics, Vol. 21, Issue 6, pp. 371-379, 2014.

[107] K. Takizawa, T.E. Tezduyar, Y. Otoguro: *Stabilization and discontinuity-capturing parameters for space–time flow computations with finite element and isogeometric discretizations*, Computational Mechanics, Vol. 62, Issue 5, pp. 1169-1186, 2018.

[108] E.J.D. Tebbens, I. Hnětynková, M. Plešinger, Z. Strakoš, P. Tichý: *Analýza Metod pro Maticové výpočty*: Základní Metody, Matfyzpress, Praha, 2012.

[109] H. Tennekes: *Turbulent flow in two and three dimensions*, Bull. Amer. Meteor. Soc., Vol. 59, pp. 22-28, 1978.

[110] T.E. Tezduyar, S. Mittal, S.E. Ray, R. Shih: *Incompressible flow computations with stabilized bilinear and linear equal–order–interpolation velocity–pressure elements*, Computer Methods in Applied Mechanics and Engineering, Vol. 95, pp. 221-242, 1992.

[111] T.E. Tezduyar, S. Sathe: *Stabilization parameters in SUPG and PSPG formulations*, Journal of Computational and Applied Mechanics, Vol. 4, No. 1, pp. 71-88, 2003.

[112] T.E. Tezduyar: *Computation of moving boundaries and interfaces and stabilization parameters*, International Journal for Numerical Methods in Fluids, Vol. 43, pp. 555-575, 2003.

[113] P. G. Tucker, C. L. Rumsey, R. E. Bartels, R. T. Biedron: *Transport equation based wall distance computations aimed at flows with time-dependent geometry*, NASA Report, NASA-TM 2003-212680, 2003.

[114] S. Turek, D. Kuzmin: *Algebraic Flux Correction III. Incompressible Flow Problems*, Ergebnisberichte Angew. Math. 270, University of Dortmund, Germany, 2004.

[115] O. Verhoeven: *Trailing Edge Noise Simulations using IDDES in OpenFOAM*, Master Thesis, Delft University of Technology, 2011.

[116] H. K. Versteeg, W. Malalasekera: *An Introduction to Computational Fluid Dynamics, The Finite Volume Method*, Second Edition, Pearson Education, 2007.

[117] W. Vieser, T. Esch, F. Menter: *Heat Transfer Predictions Using Advanced Two Equation Turbulence Models*, CFX Technical Memorandum, AEA Technology, Germany, 2002.

[118] A.A. van der Vorst, C. Vuik: *The superlinear convergence behaviour of GMRES*, J. Comp. Appl. Math., Vol. 48, pp. 327-341, 1993.

[119] C. Vuik, A. Saghir, G.P. Boerstoel: *The Krylov accelerated SIMPLE(R) method for flow problems in industrial furnaces*, Int. J. Numer. Meth. Fluids, Vol. 33, pp. 1027-1040, 2000.

[120] Q. Wang: *Uncertainty Quantification for Unsteady Fluid Flow Using Adjoint–Based Approaches*, PhD thesis, Stanford University, 2008.

[121] D.C. Wilcox: *Turbulence Modeling for CFD*, DCW Industries, 2nd edition, California, 1998.

[122] S. T. Zalesak: *Fully multidimensional flux-corrected transport algorithms for fluids*, Journal of Computational Physics, Vol. 31, Issue 3, pp. 335-362, 1979.

[123] CFD online: `http://www.cfd-online.com/`.

List of publications

Articles in journals

B. Bastl, M. Brandner, J. Egermaier, K. Michálková, E. Turnerová: *IgA-Based Solver for turbulence modelling on multipatch geometries*, Advances in Engineering Software, Vol. 113, pp. 7-18, 2017.

B. Bastl, M. Brandner, J. Egermaier, K. Michálková, E. Turnerová: *Isogeometric analysis for turbulent flow*, Mathematics and Computers in Simulation, Vol. 145, pp. 3-17, 2018.

Conference papers

B. Bastl, M. Brandner, J. Egermaier, K. Michálková, E. Turnerová: *Isogeometric Analysis for Fluid Flow Problems*, In: J. Chleboun, P. Přikryl, K. Segeth, T. Šístek (eds.): Programs and Algorithms of Numerical Mathematics. Proceedings of Seminar. Dolní Maxov, June 8-13, 2014. Institute of Mathematics AS CR, Prague, pp. 23-31, 2015.

B. Bastl, M. Brandner, J. Egermaier, H. Horníková, K. Michálková, J. Šourek, E. Turnerová: *Comparison of coupled and decoupled solvers for incompressible Navier-Stokes equations solved by isogeometric analysis*, In: E.H. van Brummelen, A. Corsini, S. Perotto, G. Rozza, (eds.), Numerical Methods for Flows: FEF 2017 Selected Contributions 132, Springer, 2018 - Accepted.

B. Bastl, M. Brandner, J. Egermaier, H. Horníková, K. Michálková, E. Turnerová: *Gradient-free and gradient-based methods for shape optimization of water turbine blade*, In: J. Chleboun, P. Küs, P. Přikryl, M. Rozložník, K. Segeth, Jakub Šístek, T. Vejchodský (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Hejnice, June 24-29, 2018. Institute of Mathematics CAS, Prague, pp. 15-26, 2019.

Other publications

B. Bastl, M. Brandner, J. Egermaier, K. Michálková, E. Turnerová: *Modely nestlačitelného proudění a jejich numerické řešení*, Výzkumná zpráva č. NTIS-VP5a-002/2013, Plzeň, 2013.

B. Bastl, M. Brandner, J. Egermaier, K. Michálková, E. Turnerová: *Turbulentní modely pro nestlačitelné proudění*, Výzkumná zpráva č. NTIS-VP5a-002/2014, Plzeň, 2014.

B. Bastl, M. Brandner, K. Michálková, E. Turnerová: *Implementace řešiče Navier-Stokesových rovnic založeného na isogeometrické analýze*, Výzkumná zpráva č. NTIS-VP5a-003/2014, Plzeň, 2014.

B. Bastl, M. Brandner, J. Egermaier, K. Michálková, E. Turnerová: *Numerical modelling of turbulent flows*, In Proceedings of Seminář Numerické Analýzy, Institute of Computer Science AS CR, Praha, pp. 9-12, 2015.

B. Bastl, M. Brandner, J. Egermaier, K. Michálková, E. Turnerová: *Metody tvarové optimalizace*, Výzkumná zpráva č. NTIS-VP5a-002/2015, Plzeň, 2015.

A Weak formulation and linearization of the unsteady problem

In the following, the weak formulation of the Reynolds-Averaged Navier–Stokes equations closed with k–ω turbulence model is derived similarly to the case of the steady Navier–Stokes problem in Section 4.1. However, special attention has to be invested to the time-derivative of the mean velocity solution. We introduced several turbulence models in Section 3. Note that all of the turbulence models are advection-diffusion-reaction equations, which differ in the form of the coefficients from the mathematical point of view. Thus, only the Wilcox's two equation k–ω turbulence model (66) is considered here to close the RANS problem and the derivation of the weak formulation of RANS with another turbulence model is straightforward. Before the weak form is derived, the closed RANS problem is formulated.

In the unsteady case, we are looking for a velocity field $\bar{\mathbf{u}}(\mathbf{x},t) : \bar{\Omega} \times [0,T] \to \mathbb{R}^d$, a pressure $p(\mathbf{x},t) : \bar{\Omega} \times [0,T] \to \mathbb{R}$, a turbulent kinetic energy $k(\mathbf{x},t) : \bar{\Omega} \times [0,T] \to \mathbb{R}$ and turbulent specific dissipation $\omega(\mathbf{x},t) : \bar{\Omega} \times [0,T] \to \mathbb{R}$ such that the initial boundary value problem is satisfied

$$\begin{aligned}
\frac{\partial \bar{\mathbf{u}}}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} - \nabla \cdot \left[(\nu + \nu_T)(\nabla \bar{\mathbf{u}} + \nabla \bar{\mathbf{u}}^T)\right] + \nabla \bar{p} &= -\frac{2}{3}\nabla k, \quad \text{in } Q,\\
\nabla \cdot \bar{\mathbf{u}} &= 0, \quad \text{in } Q\\
\frac{\partial k}{\partial t} + \bar{\mathbf{u}} \cdot \nabla k - \nabla \cdot \left[\left(\nu + \sigma_k \alpha^* \frac{k}{\omega}\right) \nabla k\right] + \beta^* \omega k &= P_k, \quad \text{in } Q\\
\frac{\partial \omega}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \omega - \nabla \cdot \left[\left(\nu + \sigma_\omega \alpha^* \frac{k}{\omega}\right) \nabla \omega\right] + \beta \omega^2 &= \gamma \frac{\omega}{k} P_k + \frac{\sigma_d}{\omega} \nabla k \cdot \nabla \omega, \quad \text{in } Q\\
\bar{\mathbf{u}}(\mathbf{x},0) &= \bar{\mathbf{u}}_0(\mathbf{x}), \quad \text{in } \Omega,\\
k(\mathbf{x},0) &= k_0(\mathbf{x}), \quad \text{in } \Omega,\\
\omega(\mathbf{x},0) &= \omega_0(\mathbf{x}), \quad \text{in } \Omega,\\
\bar{\mathbf{u}} &= \mathbf{g}, \quad \text{on } \partial\Omega_D \times (0,T),\\
\nu \mathbf{n} \cdot (\nabla \bar{\mathbf{u}} + \nabla \bar{\mathbf{u}}^T) - \mathbf{n}\bar{p} &= \mathbf{0}, \quad \text{on } \partial\Omega_N \times (0,T),\\
k &= g^k, \quad \text{on } \partial\Omega_D \times (0,T),\\
\omega &= g^\omega, \quad \text{on } \partial\Omega_D \times (0,T),\\
\mathbf{n} \cdot \nabla k &= 0, \quad \text{on } \partial\Omega_N \times (0,T),\\
\mathbf{n} \cdot \nabla \omega &= 0, \quad \text{on } \partial\Omega_N \times (0,T),
\end{aligned} \tag{142}$$

where $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$, $Q = \Omega \times (0,T)$ and the coefficients in the k–ω equations are given in Section 3.2.3. A classical solution of the unsteady RANS problem closed with the two–equation turbulence model is $(\bar{\mathbf{u}}, \bar{p}, k, \omega)$, if it satisfies (142) and $\bar{\mathbf{u}} \in C^2(\bar{Q})$, $\bar{p} \in C^1(\bar{Q})$, $k \in C^2(\bar{Q})$ and $\omega \in C^2(\bar{Q})$.

To treat the time dependence of the RANS problem, we apply the semidiscrete method such that the discretization of the time variable is given first. For this purpose, the RANS and $k-\omega$ equations are rewritten in a more general form as

$$\begin{aligned}
\frac{\partial \bar{\mathbf{u}}}{\partial t} &= R(\mathbf{x}, t, \bar{\mathbf{u}}, \nabla\bar{\mathbf{u}}, \nabla\bar{p}, \nabla k, \nu_T), \\
\frac{\partial k}{\partial t} &= R^k(\mathbf{x}, t, k, \nabla k, \omega, \bar{\mathbf{u}}, \nabla\bar{\mathbf{u}}), \\
\frac{\partial \omega}{\partial t} &= R^\omega(\mathbf{x}, t, \omega, \nabla\omega, k, \nabla k, \bar{\mathbf{u}}, \nabla\bar{\mathbf{u}}).
\end{aligned} \tag{143}$$

Now, we choose a time step Δt and discretize the time interval $[0, T]$ as $0 = t_0 < t_1 < t_2 < \ldots < t_N = T$, where $t_n = n \cdot \Delta t$, $n = 0, 1, \ldots, N$. Then we denote an approximation of the velocity and pressure at a given time t_n by $\bar{\mathbf{u}}^n = \bar{\mathbf{u}}^n(\mathbf{x}) \approx \bar{\mathbf{u}}(\mathbf{x}, t_n)$, $\bar{p}^n = \bar{p}^n(\mathbf{x}) \approx \bar{p}(\mathbf{x}, t_n)$ and similarly k^n and ω^n. We look for the approximations $\bar{\mathbf{u}}^{n+1}$, $\bar{p}^{n+1}$, k^{n+1} and ω^{n+1} applying the θ–method, which yields

$$\begin{aligned}
\frac{\bar{\mathbf{u}}^{n+1} - \bar{\mathbf{u}}^n}{\Delta t} &= \theta R(\mathbf{x}, t^{n+1}, \bar{\mathbf{u}}^{n+1}, \nabla\bar{\mathbf{u}}^{n+1}, \nabla\bar{p}^{n+1}, \nabla k^{n+1}, \nu_T^{n+1}) + \\
&\quad +(1-\theta) R(\mathbf{x}, t^n, \bar{\mathbf{u}}^n, \nabla\bar{\mathbf{u}}^n, \nabla\bar{p}^n, \nabla k^n, \nu_T^n), \\
\frac{k^{n+1} - k^n}{\Delta t} &= \theta R^k(\mathbf{x}, t^{n+1}, k^{n+1}, \nabla k^{n+1}, \omega^{n+1}, \bar{\mathbf{u}}^{n+1}, \nabla\bar{\mathbf{u}}^{n+1}) + \\
&\quad +(1-\theta) R^k(\mathbf{x}, t^n, k^n, \nabla k^n, \omega^n, \bar{\mathbf{u}}^n, \nabla\bar{\mathbf{u}}^n), \\
\frac{\omega^{n+1} - \omega^n}{\Delta t} &= \theta R^\omega(\mathbf{x}, t^{n+1}, \omega^{n+1}, \nabla\omega^{n+1}, k^{n+1}, \nabla k^{n+1}, \bar{\mathbf{u}}^{n+1}, \nabla\bar{\mathbf{u}}^{n+1}) + \\
&\quad +(1-\theta) R^\omega(\mathbf{x}, t^n, \omega^n, \nabla\omega^n, k^n, \nabla k^n, \bar{\mathbf{u}}^n, \nabla\bar{\mathbf{u}}^n),
\end{aligned} \tag{144}$$

where $\theta \in [0, 1]$. We obtain the fully explicit scheme for $\theta = 0$, the choice $\theta = 1$ results in the fully implicit scheme and the Crank–Nicolson scheme is obtained for $\theta = 0.5$. However, the fully implicit case is considered in the following paragraphs for simplicity, i.e.,

$$\begin{aligned}
\frac{\bar{\mathbf{u}}^{n+1} - \bar{\mathbf{u}}^n}{\Delta t} - \nabla \cdot \left[(\nu + \nu_T^{n+1})(\nabla\bar{\mathbf{u}}^{n+1} + (\nabla\bar{\mathbf{u}}^{n+1})^T)\right] + & \\
+\bar{\mathbf{u}}^{n+1} \cdot \nabla\bar{\mathbf{u}}^{n+1} + \nabla\bar{p}^{n+1} &= -\frac{2}{3}\nabla k^{n+1}, \\
\nabla \cdot \bar{\mathbf{u}}^{n+1} &= 0, \\
\frac{k^{n+1} - k^n}{\Delta t} + \bar{\mathbf{u}}^{n+1} \cdot \nabla k^{n+1} - \nabla \cdot \left[\left(\nu + \sigma_k \tilde{\nu}_t^{n+1}\right) \nabla k^{n+1}\right] + & \\
+\beta^{*,n+1} \omega^{n+1} k^{n+1} &= P^{n+1}, \\
\frac{\omega^{n+1} - \omega^n}{\Delta t} + \bar{\mathbf{u}}^{n+1} \cdot \nabla\omega^{n+1} - \nabla \cdot \left[\left(\nu + \sigma_\omega \tilde{\nu}_t^{n+1}\right) \nabla\omega^{n+1}\right] + & \\
+\beta^{n+1} \left(\omega^{n+1}\right)^2 - \frac{\sigma_d^{n+1}}{\omega^{n+1}} \nabla k^{n+1} \cdot \nabla\omega^{n+1} &= \gamma^{n+1} \frac{\omega^{n+1}}{k^{n+1}} P^{n+1}.
\end{aligned} \tag{145}$$

Notice that we used a substitution

$$\tilde{\nu}_t = \alpha^* \frac{k}{\omega} \tag{146}$$

to simplify expressions in the equations (145).

Weak formulation

In order to derive the weak formulation of the semidiscrete problem, we proceed similarly to the derivation of the weak form of the stationary Navier–Stokes in Section 4.1, i.e., we define solution and test function spaces as follows

$$\begin{aligned} V &= \{\mathbf{u} \in H^1(\Omega)^d \,|\, \mathbf{u} = \mathbf{g} \text{ on } \partial\Omega_D\}, \\ V_0 &= \{\mathbf{v} \in H^1(\Omega)^d \,|\, \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega_D\}, \\ \hat{V} &= \{k \in H^1(\Omega) \,|\, k = g^k \text{ on } \partial\Omega_D\}, \\ \tilde{V} &= \{\omega \in H^1(\Omega) \,|\, \omega = g^\omega \text{ on } \partial\Omega_D\}, \\ \hat{V}_0 &= \{w \in H^1(\Omega) \,|\, w = 0 \text{ on } \partial\Omega_D\}. \end{aligned} \tag{147}$$

The weak formulation of the closed RANS problem in semidiscrete form is to find $\bar{\mathbf{u}}^{n+1} \in V$, $\bar{p}^{n+1} \in L^2(\Omega)$, $k^{n+1} \in \hat{V}$ and $\omega^{n+1} \in \tilde{V}$, satisfying $\bar{\mathbf{u}}^0 = \bar{\mathbf{u}}_0(\mathbf{x})$, $k^0 = k_0(\mathbf{x})$, $\omega^0 = \omega_0(\mathbf{x})$ and

$$\begin{aligned} &\frac{1}{\Delta t}(\bar{\mathbf{u}}^{n+1}, \mathbf{v}) + ((\nu + \nu_T^{n+1})\nabla\bar{\mathbf{u}}^{n+1}, \nabla\mathbf{v}) + (\bar{\mathbf{u}}^{n+1} \cdot \nabla\bar{\mathbf{u}}^{n+1}, \mathbf{v}) - \\ &\qquad -(\nabla \cdot \mathbf{v}, p^{n+1}) + (\nu_T^{n+1}(\nabla\bar{\mathbf{u}}^{n+1})^T, \nabla\mathbf{v}) = \frac{1}{\Delta t}(\bar{\mathbf{u}}^n, \mathbf{v}) + \frac{2}{3}(\nabla \cdot \mathbf{v}, k^{n+1}), \\ &\qquad\qquad (\nabla \cdot \bar{\mathbf{u}}^{n+1}, q) = 0, \\ &\frac{1}{\Delta t}(k^{n+1}, w) + (\bar{\mathbf{u}}^{n+1} \cdot \nabla k^{n+1}, w) + ((\nu + \sigma_k\tilde{\nu}_t^{n+1})\nabla k^{n+1}, \nabla w) + \\ &\qquad +(\beta^{*,n+1}\omega^{n+1}k^{n+1}, w) = \frac{1}{\Delta t}(k^n, w) + (P^{n+1}, w), \\ &\frac{1}{\Delta t}(\omega^{n+1}, w) + (\bar{\mathbf{u}}^{n+1} \cdot \nabla\omega^{n+1}, w) + (\left(\nu + \sigma_\omega\tilde{\nu}_t^{n+1}\right)\nabla\omega^{n+1}, \nabla w) + (\beta^{n+1}\left(\omega^{n+1}\right)^2, w) \\ &\qquad -(\frac{\sigma_d^{n+1}}{\omega^{n+1}}\nabla k^{n+1} \cdot \nabla\omega^{n+1}, w) = \frac{1}{\Delta t}(\omega^n, w) + (\gamma^{n+1}\frac{\omega^{n+1}}{k^{n+1}}P^{n+1}, w), \end{aligned} \tag{148}$$

for all $\mathbf{v} \in V_0$, $q \in L_2(\Omega)$ and $w \in \hat{V}_0$, where the L^2 scalar products in Ω are defined as

$$\begin{aligned} (\mathbf{u}, \mathbf{v}) &= \int_\Omega \mathbf{u} \cdot \mathbf{v}\, \mathrm{d}\Omega, && \text{for vector functions,} \\ (\sigma, \tau) &= \int_\Omega \sigma : \tau\, \mathrm{d}\Omega, && \text{for second-order tensors} \\ (p, q) &= \int_\Omega pq\, \mathrm{d}\Omega, && \text{for scalar functions.} \end{aligned} \tag{149}$$

To treat the nonlinearity in the advective term of the unsteady problem, we employ Picard's method in the same way as described in Section 4.1.2 at every time step, i.e., the problem is solved iteratively at each time step and the non-linear term is linearized using the solution from previous Picard's iteration step, [35].

Assume that the weak solution at the n-th time step $(\bar{\mathbf{u}}^n, \bar{p}^n, k^n, \omega^n) \in V \times L^2(\Omega) \times L^2(\Omega) \times L^2(\Omega)$, $n = 0, 1, \ldots$, is known. To linearize the problem (148) at the $(n+1)$-th time step using Picard's method, consider an initial guess $(\bar{\mathbf{u}}^{n+1,0}, \bar{p}^{n+1,0}, k^{n+1,0}, \omega^{n+1,0}) \in V \times$

$L^2(\Omega) \times \hat{V} \times \tilde{V}$ and a sequence of iterates $(\bar{\mathbf{u}}^{n+1,1}, \bar{p}^{n+1,1}, k^{n+1,1}, \omega^{n+1,1})$, ..., $(\bar{\mathbf{u}}^{n+1,m}, \bar{p}^{n+1,m}, k^{n+1,m}, \omega^{n+1,m}) \in V \times L^2(\Omega) \times \hat{V} \times \tilde{V}$. Now, let $(\bar{\mathbf{u}}^{n+1,m}, \bar{p}^{n+1,m}, k^{n+1,m}, \omega^{n+1,m})$ be known from the previous iteration step. Then we look for $\bar{\mathbf{u}}^{n+1,m+1} \in V$, $\bar{p}^{n+1,m+1} \in L^2(\Omega)$, $k^{n+1,m+1} \in \hat{V}$, $\omega^{n+1,m+1} \in \tilde{V}$, $m = 0, 1, \ldots$, such that for all $\mathbf{v} \in V_0$, $q \in L^2(\Omega)$ and $w \in \hat{V}_0$ satisfying

$$\begin{array}{rcl}
\frac{1}{\Delta t}(\bar{\mathbf{u}}^{n+1,m+1} - \bar{\mathbf{u}}^n, \mathbf{v}) + & & \\
+((\nu + \nu_T^{n+1})\nabla \bar{\mathbf{u}}^{n+1,m+1}, \nabla \mathbf{v}) - (\nabla \cdot \mathbf{v}, p^{n+1,m+1}) + & & \\
+(\bar{\mathbf{u}}^{n+1,m} \cdot \nabla \bar{\mathbf{u}}^{n+1,m+1}, \mathbf{v}) + (\nu_T^{n+1}(\nabla \bar{\mathbf{u}}^{n+1,m+1})^T, \nabla \mathbf{v}) & = & \frac{2}{3}(\nabla \cdot \mathbf{v}, k^{n+1}), \\
(\nabla \cdot \bar{\mathbf{u}}^{n+1,m+1}, q) & = & 0, \\
\frac{1}{\Delta t}(k^{n+1,m+1} - k^n, w) + ((\nu + \sigma_k \tilde{\nu}_t^{n+1,m})\nabla k^{n+1,m+1}, \nabla w) + & & \\
+(\bar{\mathbf{u}}^{n+1} \cdot \nabla k^{n+1,m+1}, w) + ((\beta^*)^{n+1,m}\omega^{n+1,m}k^{n+1,m+1}, w) & = & (P^{n+1,m}, w), \\
\frac{1}{\Delta t}(\omega^{n+1,m+1} - \omega^n, w) + ((\nu + \sigma_\omega \tilde{\nu}_t^{n+1,m})\nabla \omega^{n+1,m+1}, \nabla w) + & & \\
+(\bar{\mathbf{u}}^{n+1} \cdot \nabla \omega^{n+1,m+1}, w) + (\beta^{n+1,m}\omega^{n+1,m}\omega^{n+1,m+1}, w) - & & \\
-(\frac{\sigma_d^{n+1,m}}{\omega^{n+1,m}}\nabla k^{n+1,m} \cdot \nabla \omega^{n+1,m+1}, w) & = & (\gamma^{n+1,m}\frac{\omega^{n+1,m}}{k^{n+1,m}}P^{n+1,m}, w),
\end{array} \tag{150}$$

Similarly to the Picard's iteration of the steady Navier–Stokes problem, the solution from the previous time step is chosen to be the initial guess of the Picard's iteration of the RANS problem (150), i.e. $\bar{\mathbf{u}}^{n+1,0} = \bar{\mathbf{u}}^n, p^{n+1,0} = p^n$, $k^{n+1,0} = k^n, \omega^{n+1,0} = \omega^n$. Note that the given initial condition from (142) is considered as the initial guess for the Picard's iteration at the first time step. Another approach how to deal with the nonlinearity is to use the solution from the previous time step to evaluate the nonlinear terms, which is possible when using a sufficiently small time step Δt.

B Discrete unsteady formulation

To discretize the closed RANS problem (150) by means of the Galerkin method, we proceed in the same way as in the derivation of the discrete steady Navier–Stokes problem given in Section 4.1.3. Define finite dimensional subspaces $V^h \subset V, V_0^h \subset V_0$, $W^h \subset L_2(\Omega)$, $\hat{V}^h \subset \hat{V}, \hat{V}_0^h \subset \hat{V}_0$ and $\tilde{V}^h \subset \tilde{V}$ together with their bases. We look for discrete solution $\bar{\mathbf{u}}_h \in V^h$, $\bar{p}_h \in W^h$, $k_h \in \hat{V}_h$ and $\omega_h \in \tilde{V}_h$ of the closed RANS problem such that

$$
\begin{aligned}
\frac{1}{\Delta t}(\bar{\mathbf{u}}_h^{n+1,m+1}, \mathbf{v}_h)+ & \\
+((\nu + \nu_T^{n+1})\nabla \bar{\mathbf{u}}_h^{n+1,m+1}, \nabla \mathbf{v}_h) - (\nabla \cdot \mathbf{v}_h, p_h^{n+1,m+1})+ & \\
+(\bar{\mathbf{u}}_h^{n+1,m} \cdot \nabla \bar{\mathbf{u}}_h^{n+1,m+1}, \mathbf{v}_h) + (\nu_T^{n+1}(\nabla \bar{\mathbf{u}}_h^{n+1,m+1})^T, \nabla \mathbf{v}_h) &= \frac{1}{\Delta t}(\bar{\mathbf{u}}_h^{n}, \mathbf{v}_h) + \\
&+\frac{2}{3}(\nabla \cdot \mathbf{v}_h, k_h^{n+1}), \\
(\nabla \cdot \bar{\mathbf{u}}_h^{n+1,m+1}, q_h) &= 0, \\
\frac{1}{\Delta t}(k_h^{n+1,m+1}, w_h) + ((\nu + \sigma_k \tilde{\nu}_t^{n+1,m})\nabla k_h^{n+1,m+1}, \nabla w_h)+ & \\
+(\bar{\mathbf{u}}_h^{n+1} \cdot \nabla k_h^{n+1,m+1}, w_h) + ((\beta^*)^{n+1,m}\omega_h^{n+1,m} k_h^{n+1,m+1}, w) &= \frac{1}{\Delta t}(k_h^{n}, w) + (P^{n+1,m}, w_h), \\
\frac{1}{\Delta t}(\omega_h^{n+1,m+1}, w_h) + ((\nu + \sigma_\omega \tilde{\nu}_t^{n+1,m})\nabla \omega_h^{n+1,m+1}, \nabla w_h)+ & \\
+(\bar{\mathbf{u}}_h^{n+1} \cdot \nabla \omega_h^{n+1,m+1}, w_h) + (\beta^{n+1,m}\omega_h^{n+1,m}\omega_h^{n+1,m+1}, w_h)- & \\
-(\frac{\sigma_d^{n+1,m}}{\omega_h^{n+1,m}}\nabla k_h^{n+1,m} \cdot \nabla \omega_h^{n+1,m+1}, w_h) &= \frac{1}{\Delta t}(\omega_h^{n}, w_h) + \\
&+(\gamma^{n+1,m}\frac{\omega_h^{n+1,m}}{k_h^{n+1,m}}P^{n+1,m}, w_h),
\end{aligned}
\tag{151}
$$

for all $\mathbf{v}_h \in V_0^h$, $q_h \in W^h$ and $w_h \in \hat{V}_0^h$. Let us write the velocity solution $\bar{\mathbf{u}}_h$ as a linear combination of basis functions $\varphi_j^{\bar{u}} \in V^h$ and the pressure solution $\bar{p}_h$ as a linear combination of basis functions $\varphi_j^{\bar{p}} \in W^h$. In 3D, the approximate solution has the form

$$
\begin{aligned}
\bar{\mathbf{u}}_h &= \sum_{j=1}^{n^{\bar{u}}} (\bar{u}_{1j}, \bar{u}_{2j}, \bar{u}_{3j})^T \varphi_j^{\bar{u}} + \sum_{j=n^{\bar{u}}+1}^{n_D^{\bar{u}}} (\bar{u}_{1j}^*, \bar{u}_{2j}^*, \bar{u}_{3j}^*)^T \varphi_j^{\bar{u}}, \\
\bar{p}_h &= \sum_{i=i}^{n^{\bar{p}}} \bar{p}_j \varphi_j^{\bar{p}},
\end{aligned}
\tag{152}
$$

where the coefficients $\bar{\mathbf{u}}_j^*, j = n^{\bar{u}}+1, \dots, n_D^{\bar{u}}$ are fixed and the second velocity term in (152) represents the interpolation of the Dirichlet boundary condition. Similarly, the functions of turbulent kinetic energy k_h and specific turbulent dissipation ω_h can be written as a linear combination of basis functions such that we employ basis functions equal to pressure basis functions and noted as $\varphi_j^{k\omega}$. Then

$$\begin{aligned} k_h &= \sum_{j=1}^{n^{k\omega}} k_j \varphi_j^{k\omega} + \sum_{j=n^{k\omega}+1}^{n_D^{k\omega}} k_j^* \varphi_j^{k\omega}, \\ \omega_h &= \sum_{j=1}^{n^{k\omega}} \omega_j \varphi_j^{k\omega} + \sum_{j=n^{k\omega}+1}^{n_D^{k\omega}} \omega_j^* \varphi_j^{k\omega}. \end{aligned} \tag{153}$$

Remember that the type of boundary conditions is assumed to be the same at each boundary for k and ω according to (142). Thus, the number of coefficients, which corresponds to the Dirichlet boundary condition is the same for k and ω and it is marked by $n_D^{k\omega}$.

The Picard's iteration is performed in every time step as mentioned in the previous section. However, the matrix formulations are written without treatment of the nonlinearity to keep the following text well readable. For the same reason, the matrix forms of the RANS and k–ω model are written separately and the RANS matrix formulation is mentioned first, i.e. suppose that the solution from the previous time step is known, then we look for the vector of velocity and pressure coefficients $(\bar{\mathbf{u}}_1^{n+1}, \bar{\mathbf{u}}_2^{n+1}, \bar{\mathbf{u}}_3^{n+1}, \bar{p}^{n+1}) \in \mathbb{R}^{n^{\bar{u}}} \times \mathbb{R}^{n^{\bar{u}}} \times \mathbb{R}^{n^{\bar{u}}} \times \mathbb{R}^{n^{\bar{p}}}$ such that the following system of equations in three dimensions is satisfied

$$\begin{bmatrix} \mathbf{M}_1 & \mathbf{E}_{12} & \mathbf{E}_{13} & -\mathbf{B}_1^T \\ \mathbf{E}_{21} & \mathbf{M}_2 & \mathbf{E}_{23} & -\mathbf{B}_2^T \\ \mathbf{E}_{31} & \mathbf{E}_{32} & \mathbf{M}_3 & -\mathbf{B}_3^T \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{u}}_1^{n+1} \\ \bar{\mathbf{u}}_2^{n+1} \\ \bar{\mathbf{u}}_3^{n+1} \\ \bar{\mathbf{p}}^{n+1} \end{bmatrix} = \frac{1}{\Delta t} \begin{bmatrix} \mathbf{C} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{u}}_1^{n} \\ \bar{\mathbf{u}}_2^{n} \\ \bar{\mathbf{u}}_3^{n} \end{bmatrix} + \frac{2}{3} \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \mathbf{F}_3 \\ \mathbf{0} \end{bmatrix} \mathbf{k}^{n+1} + \tag{154}$$

$$+ \begin{bmatrix} -(\frac{1}{\Delta t}\mathbf{C}^* + \mathbf{A}^* + \mathbf{N}^*(\bar{\mathbf{u}}^{n+1}) + \mathbf{E}_{11}) \cdot \bar{\mathbf{u}}_1^* - \mathbf{E}_{12} \cdot \bar{\mathbf{u}}_2^* - \mathbf{E}_{13} \cdot \bar{\mathbf{u}}_3^* \\ -(\frac{1}{\Delta t}\mathbf{C}^* + \mathbf{A}^* + \mathbf{N}^*(\bar{\mathbf{u}}^{n+1}) + \mathbf{E}_{22}) \cdot \bar{\mathbf{u}}_2^* - \mathbf{E}_{21} \cdot \bar{\mathbf{u}}_1^* - \mathbf{E}_{23} \cdot \bar{\mathbf{u}}_3^* \\ -(\frac{1}{\Delta t}\mathbf{C}^* + \mathbf{A}^* + \mathbf{N}^*(\bar{\mathbf{u}}^{n+1}) + \mathbf{E}_{33}) \cdot \bar{\mathbf{u}}_3^* - \mathbf{E}_{31} \cdot \bar{\mathbf{u}}_1^* - \mathbf{E}_{32} \cdot \bar{\mathbf{u}}_2^* \\ -(\mathbf{B}_1^* \cdot \bar{\mathbf{u}}_1^* + \mathbf{B}_2^* \cdot \bar{\mathbf{u}}_2^* + \mathbf{B}_3^* \cdot \bar{\mathbf{u}}_3^*) \end{bmatrix} \tag{155}$$

where

$$\mathbf{M}_i = \tfrac{1}{\Delta t}\mathbf{C} + \mathbf{A} + \mathbf{N}(\bar{\mathbf{u}}^{n+1}) + \mathbf{E}_{ii} \tag{156}$$

and

$$\begin{aligned} \mathbf{A} &= [A_{ij}]_{1\le i\le n^{\bar{u}},1\le j\le n^{\bar{u}}}, & \mathbf{A}^* &= [A_{ij}]_{1\le i\le n^{\bar{u}},n^{\bar{u}}+1\le j\le n_D^{\bar{u}}}, \\ \mathbf{N}(\bar{\mathbf{u}}) &= [N_{ij}(\bar{\mathbf{u}})]_{1\le i\le n^{\bar{u}},1\le j\le n^{\bar{u}}}, & \mathbf{N}^*(\bar{\mathbf{u}}) &= [N_{ij}(\bar{\mathbf{u}})]_{1\le i\le n^{\bar{u}},n^{\bar{u}}+1\le j\le n_D^{\bar{u}}}, \\ \mathbf{B}_l &= [B_{lij}]_{1\le i\le n^{\bar{p}},1\le j\le n^{\bar{u}}}, & \mathbf{B}_l^* &= [B_{lij}]_{1\le i\le n^{\bar{p}},n^{\bar{u}}+1\le j\le n_D^{\bar{u}}}, \\ \mathbf{C} &= [C_{ij}]_{1\le i\le n^{\bar{u}},1\le j\le n^{\bar{u}}}, & \mathbf{C}^* &= [C_{ij}]_{1\le i\le n^{\bar{u}},n^{\bar{u}}+1\le j\le n_D^{\bar{u}}}, \\ \mathbf{E}_{kl} &= [E_{klij}]_{1\le i\le n^{\bar{u}},1\le j\le n^{\bar{u}}}, & \mathbf{E}_{kl}^* &= [E_{klij}]_{1\le i\le n^{\bar{u}},n^{\bar{u}}+1\le j\le n_D^{\bar{u}}}, \\ \mathbf{F}_l &= [F_{lij}]_{1\le i\le n_d^{\bar{u}},1\le j\le n^{k\omega}}, \end{aligned} \tag{157}$$

$$
\begin{aligned}
A_{ij} &= \left((\nu+\nu_T)\nabla\varphi_j^{\bar{u}}, \nabla\varphi_i^{\bar{u}}\right), \\
N_{ij}(\bar{\mathbf{u}}) &= \left(\left(\sum_{l=1}^{n_D^{\bar{u}}}(\bar{u}_{1l},\bar{u}_{2l},\bar{u}_{3l})\varphi_l^{\bar{u}}\right)\cdot\nabla\varphi_j^{\bar{u}}, \varphi_i^{\bar{u}}\right), \\
B_{lij} &= \left(\nabla\varphi_j^{\bar{u}}\cdot\mathbf{e}_l, \varphi_i^{\bar{p}}\right), \\
C_{ij} &= \left(\varphi_j^{\bar{u}}, \varphi_i^{\bar{u}}\right), \\
E_{klij} &= \left(\nabla\varphi_j^{\bar{u}}\cdot\mathbf{e}_k, \nabla\varphi_i^{\bar{u}}\cdot\mathbf{e}_l\right), \\
F_{lij} &= \left(\varphi_j^{k\omega}, \nabla\varphi_i^{\bar{u}}\cdot\mathbf{e}_l\right)
\end{aligned}
\tag{158}
$$

Here, the block matrices $\mathbf{E}_{kl}, k \neq l$ are off-diagonal blocks. As the Picard's iteration is performed in every time step, the blocks $\mathbf{E}_{kl}$ can be treated explicitely to keep the off-diagonal blocks of the system matrix zero.

Now the matrix formulation of the turbulence model is written, i.e. also suppose that the k and ω solution from the previous time step is known. Then, we look for the vector of the coefficients $(k^{n+1}, \omega^{n+1}) \in \mathbb{R}^{n^{k\omega}} \times \mathbb{R}^{n^{k\omega}}$ such that the following system of equations in three dimensions is satisfied

$$
\begin{bmatrix} \mathbf{M}^k & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^\omega \end{bmatrix}
\begin{bmatrix} \mathbf{k}^{n+1} \\ \omega^{n+1} \end{bmatrix}
= \frac{1}{\Delta t}\begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} \end{bmatrix}
\begin{bmatrix} \mathbf{k}^{n} \\ \omega^{n} \end{bmatrix}
+ \begin{bmatrix} \mathbf{F}^k \\ \mathbf{F}^\omega \end{bmatrix} +
\tag{159}
$$

$$
+\begin{bmatrix} -(\frac{1}{\Delta t}\mathbf{C}^{k\omega *} + \mathbf{A}^{k*} + \mathbf{N}^{k\omega *}(\bar{\mathbf{u}}^{n+1}) + \mathbf{D}^{k*}(\omega^{n+1}))\cdot\mathbf{k}^* \\ -(\frac{1}{\Delta t}\mathbf{C}^{k\omega *} + \mathbf{A}^{\omega *} + \mathbf{N}^{k\omega *}(\bar{\mathbf{u}}^{n+1}) + \mathbf{D}^{\omega *}(\omega^{n+1}) + \mathbf{T}^{\omega *}(k^{n+1},\omega^{n+1}))\cdot\omega^* \end{bmatrix}
\tag{160}
$$

where

$$
\begin{aligned}
\mathbf{M}^k &= \tfrac{1}{\Delta t}\mathbf{C}^{k\omega} + \mathbf{A}^k + \mathbf{N}^{k\omega}(\bar{\mathbf{u}}^n) + \mathbf{D}^k(\omega^{n+1}), \\
\mathbf{M}^\omega &= \tfrac{1}{\Delta t}\mathbf{C}^{k\omega} + \mathbf{A}^\omega + \mathbf{N}^{k\omega}(\bar{\mathbf{u}}^n) + \mathbf{D}^\omega(\omega^{n+1}) + \mathbf{T}^\omega(k^{n+1},\omega^{n+1})
\end{aligned}
\tag{161}
$$

and

$$
\begin{array}{lcllcl}
\mathbf{C}^{k\omega} & = & [C_{ij}^{k\omega}]_{1\le i\le n^{k\omega},1\le j\le n^{k\omega}}, & \mathbf{C}^{k\omega *} & = & [C_{ij}^{k\omega}]_{1\le i\le n^{k\omega},n^{k\omega}+1\le j\le n_D^{k\omega}}, \\
\mathbf{A}^{k} & = & [A_{ij}^{k}]_{1\le i\le n^{k\omega},1\le j\le n^{k\omega}}, & \mathbf{A}^{k*} & = & [A_{ij}^{k}]_{1\le i\le n^{k\omega},n^{k\omega}+1\le j\le n_D^{k\omega}}, \\
\mathbf{A}^{\omega} & = & [A_{ij}^{\omega}]_{1\le i\le n^{k\omega},1\le j\le n^{k\omega}}, & \mathbf{A}^{\omega *} & = & [A_{ij}^{\omega}]_{1\le i\le n^{k\omega},n^{k\omega}+1\le j\le n_D^{k\omega}}, \\
\mathbf{N}^{k\omega}(\bar{\mathbf{u}}) & = & [N_{ij}^{k\omega}(\bar{\mathbf{u}})]_{1\le i\le n^{k\omega},1\le j\le n^{k\omega}}, & \mathbf{N}^{k\omega *}(\bar{\mathbf{u}}) & = & [N_{ij}^{k\omega}(\bar{\mathbf{u}})]_{1\le i\le n^{k\omega},n^{k\omega}+1\le j\le n_D^{k\omega}}, \\
\mathbf{D}^{k}(\omega) & = & [D_{ij}^{k}(\omega)]_{1\le i\le n^{k\omega},1\le j\le n^{k\omega}}, & \mathbf{D}^{k*}(\omega) & = & [D_{ij}^{k}(\omega)]_{1\le i\le n^{k\omega},n^{k\omega}+1\le j\le n_D^{k\omega}}, \\
\mathbf{D}^{\omega}(\omega) & = & [D_{ij}^{\omega}(\omega)]_{1\le i\le n^{k\omega},1\le j\le n^{k\omega}}, & \mathbf{D}^{\omega *}(\omega) & = & [D_{ij}^{\omega}(\omega)]_{1\le i\le n^{k\omega},n^{k\omega}+1\le j\le n_D^{k\omega}}, \\
\mathbf{T}^{\omega}(k,\omega) & = & [T_{ij}^{\omega}(k,\omega)]_{1\le i\le n^{k\omega},1\le j\le n^{k\omega}}, & \mathbf{T}^{\omega *}(k,\omega) & = & [T_{ij}^{\omega}(k,\omega)]_{1\le i\le n^{k\omega},n^{k\omega}+1\le j\le n_D^{k\omega}}, \\
\mathbf{F}^{k} & = & [F_{i}^{k}]_{1\le i\le n^{k\omega}}, & & & \\
\mathbf{F}^{\omega} & = & [F_{i}^{\omega}]_{1\le i\le n^{k\omega}}, & & &
\end{array}
\tag{162}
$$

$$
\begin{aligned}
C_{ij}^{k\omega} &= \left(\varphi_j^{k\omega}, \varphi_i^{k\omega}\right), \\
A_{ij}^{k} &= \left((\nu+\sigma_k\tilde{\nu}_t)\nabla\varphi_j^{k\omega}, \nabla\varphi_i^{k\omega}\right),
\end{aligned}
$$

$$
\begin{aligned}
A_{ij}^{\omega} &= \big((\nu + \sigma_\omega \tilde{\nu}_t)\nabla\varphi_j^{k\omega}, \nabla\varphi_i^{k\omega}\big)\,, \\
N_{ij}^{k\omega}(\bar{\mathbf{u}}) &= \left(\left(\sum_{l=1}^{n_D^{\bar{u}}}(\bar{u}_{1l},\bar{u}_{2l},\bar{u}_{3l})\varphi_l^{\bar{u}}\right)\cdot\nabla\varphi_j^{k\omega}, R_i^{k\omega}\right), \\
D_{ij}^{k}(\omega) &= \left(\beta^*\left(\sum_{l=1}^{n_D^{k\omega}}\omega_l\varphi_l^{k\omega}\right)\varphi_j^{k\omega}, \varphi_i^{k\omega}\right), \\
D_{ij}^{\omega}(\omega) &= \left(\beta\left(\sum_{l=1}^{n_D^{k\omega}}\omega_l\varphi_l^{k\omega}\right)\varphi_j^{k\omega}, \varphi_i^{k\omega}\right), \\
T_{ij}^{\omega}(k,\omega) &= \left(\frac{\sigma_d}{\sum_{l=1}^{n_D^{k\omega}}\omega_l\varphi_l^{k\omega}}\left(\sum_{l=1}^{n_D^{k\omega}}k_l\nabla\varphi_l^{k\omega}\right)\cdot\nabla\varphi_j^{k\omega}, \varphi_i^{k\omega}\right), \\
F_i^{k} &= \big(P(\bar{\mathbf{u}},\nu_T), \varphi_i^{k\omega}\big)\,, \\
F_i^{\omega} &= \left(\gamma\frac{\sum_{l=1}^{n_D^{k\omega}}\omega_l\varphi_l^{k\omega}}{\sum_{l=1}^{n_D^{k\omega}}k_l\varphi_l^{k\omega}}P(\bar{\mathbf{u}},\nu_T), \varphi_i^{k\omega}\right).
\end{aligned}
\tag{163}
$$

C Extension of numerical experiments

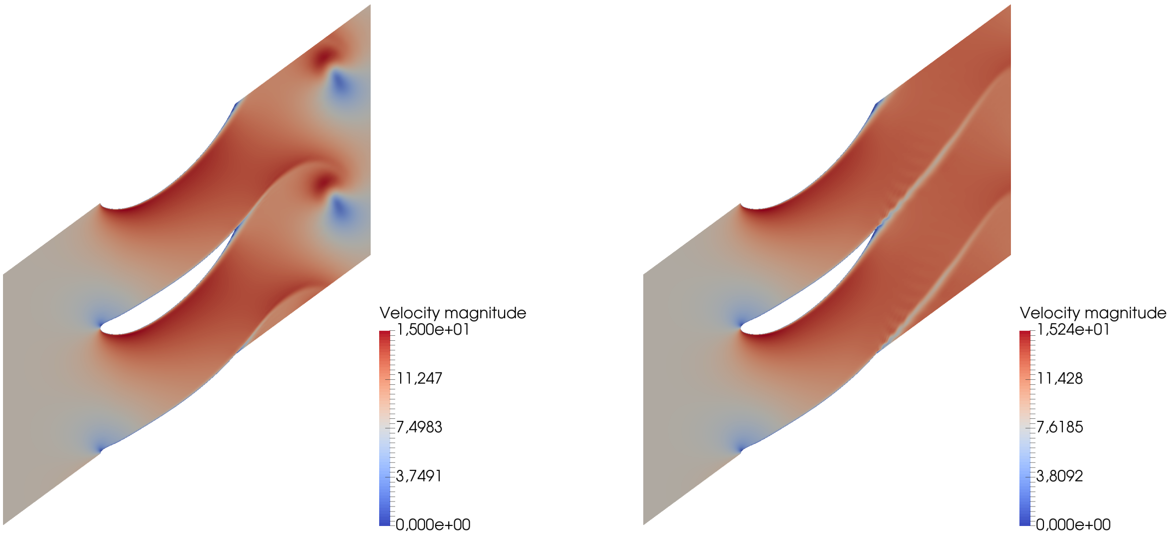


Figure 68: RANS velocity solution for $\nu = 10^{-6}$ in time $T_1 = 0.03s$ (left) and $T_2 = 0.06s$ (right). SOLD method (SUPG with additional crosswind) for SST model and IT-SUPG for RANS equations. Stabilization parameters are τ_S^4 for both SUPG and crosswind, τ_S^3 for IT-SUPG. Constant initial condition, $\Delta t = 0.0001$.

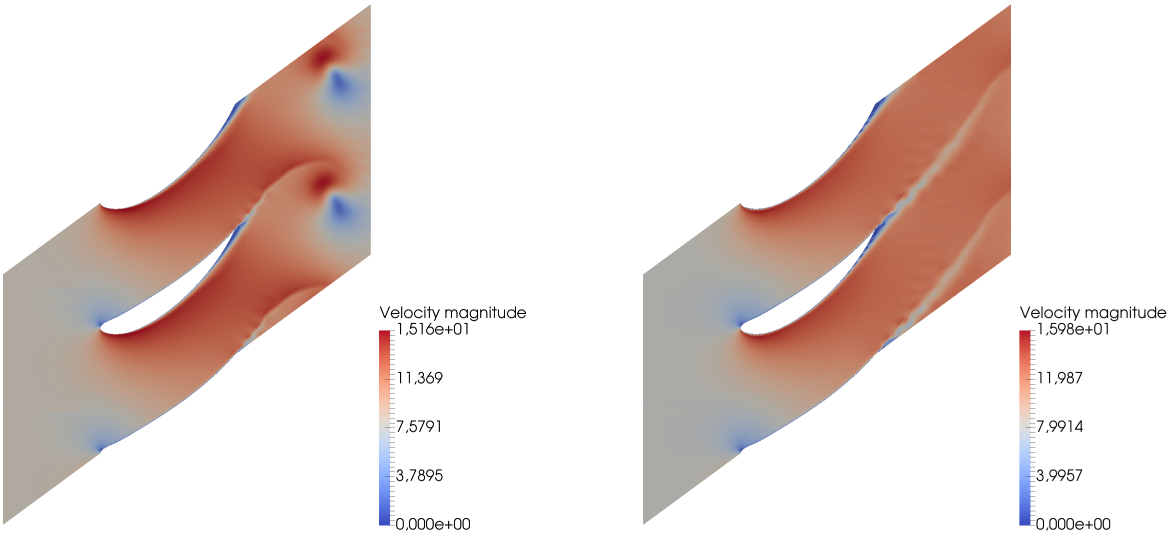


Figure 69: RANS velocity solution in time $T_1 = 0.03s$ (left) and $T_2 = 0.06s$ (right). SOLD method (SUPG with additional crosswind) for SST model. Stabilization parameters are τ_S^4 for SUPG, τ_S^3 for crosswind. Constant initial condition, $\nu = 10^{-5}$, $\Delta t = 0.0001$.

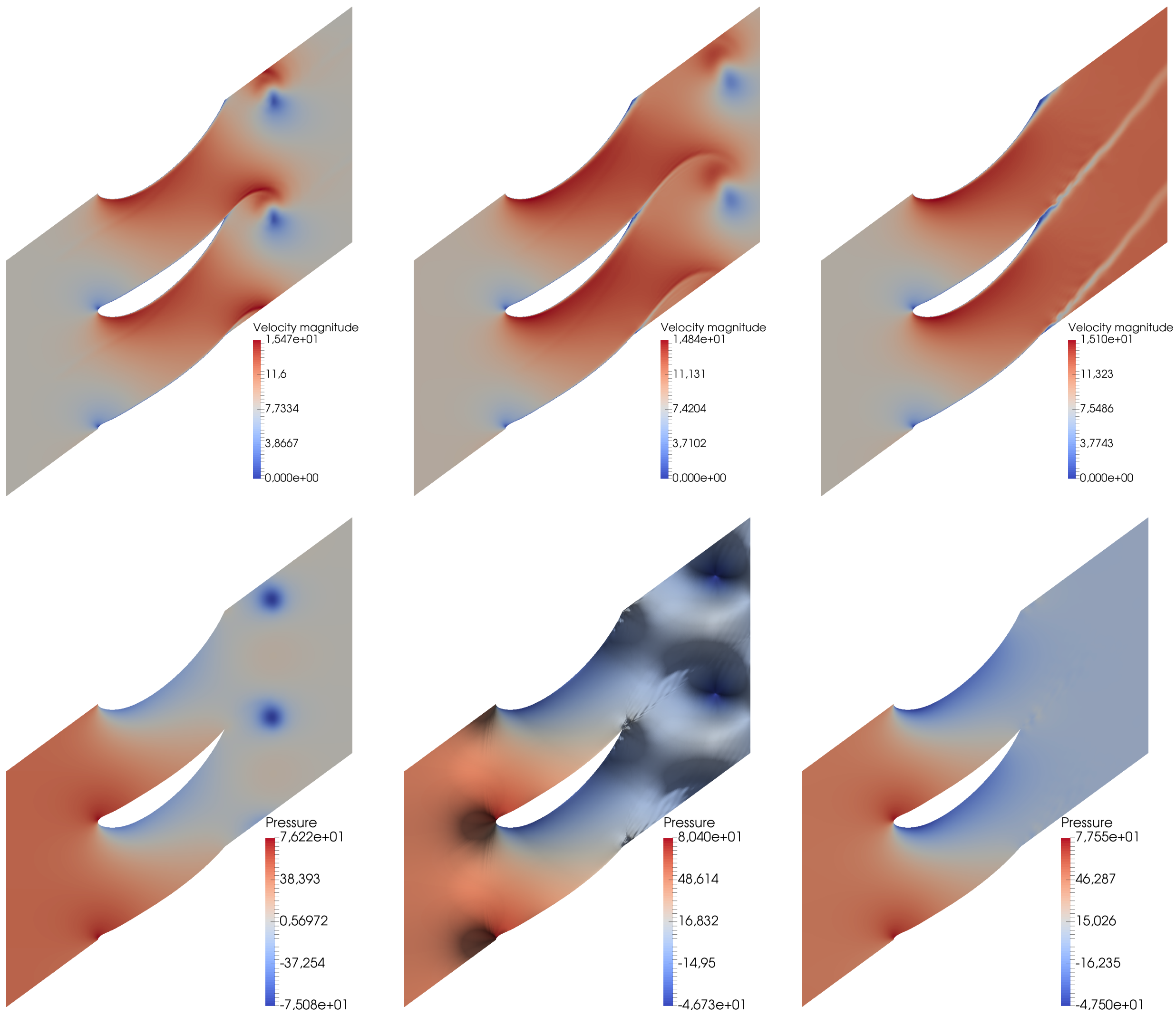
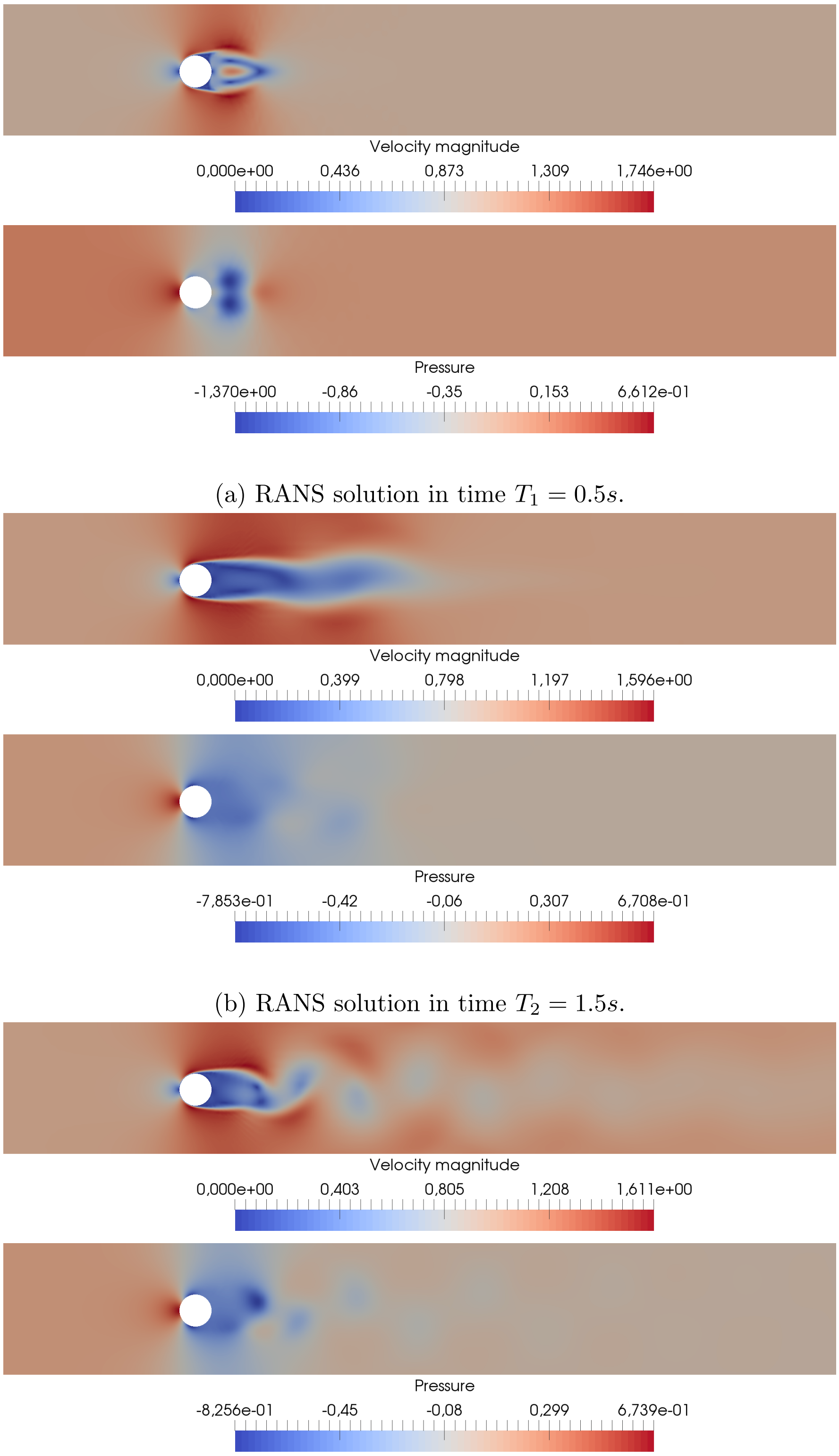


Figure 70: Time evolution of RANS velocity (upper row) and pressure (bottom row) solutions using SST turbulence model for $T_1 = 0.015s$ (left col), $T_2 = 0.03s$ (middle col), $T_3 = 0.12s$ (right col). SOLD method (SUPG with additional crosswind) for SST model and IT-SUPG method for RANS equations. Stabilization parameters: τ_S^4 for both SUPG and crosswind, $\tau_S^{2,deg}$ for IT-SUPG. Constant initial condition, $\nu = 10^{-5}$, $\Delta t = 0.0001$, (3D plot of the middle bottom result: top view).



(a) RANS solution in time $T_1 = 0.5s$.

(b) RANS solution in time $T_2 = 1.5s$.

(c) RANS solution in time $T_3 = 3.5s$.

Figure 71: Time evolution of RANS velocity and pressure solutions for $T_1 = 0.5s$, $T_2 = 1.5s$, $T_3 = 3.5s$, $\nu = 10^{-4}$, $\Delta t = 10^{-3}$. RANS equations and SST turbulence model without stabilization terms. Constant initial condition.