Downloaded from orbit.dtu.dk on: Dec 20, 2017

Technical University of Denmark

Finite Volume Methods for Incompressible Navier-Stokes Equations on Collocated Grids with Nonconformal Interfaces

Kolmogorov, Dmitry; Zhu, Wei Jun; Sørensen, Niels N.; Sørensen, Jens Nørkær; Shen, Wen Zhong

Publication date:
2014

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Kolmogorov, D., Zhu, W. J., Sørensen, N. N., Sørensen, J. N., & Shen, W. Z. (2014). Finite Volume Methods for Incompressible Navier-Stokes Equations on Collocated Grids with Nonconformal Interfaces. DTU Wind Energy.

DTU Library
Technical Information Center of Denmark

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Finite Volume Methods for Incompressible Navier-Stokes Equations on Collocated Grids with Nonconformal Interfaces

DTU Vindenergi
PhD Rapport 2014

Dmitry K. Kolmogorov

DTU Wind Energy PhD-0040

Maj 2014

DTU Wind Energy
Department of Wind Energy

Forfatter(e): Dmitry K. Kolmogorov

Titel: Finite Volume Methods for Incompressible Navier-Stokes Equations on Collocated Grids with Nonconformal Interfaces

Department: Wind Energy

DTU Wind Energy PhD-0040

Maj 2014

Projektperiode:
01.02.2011-09.05.2014

Uddannelse:
Doctor of Philosophy

Område:
Wind Energy

Vejledere:
Wen Z. Shen
Niels N. Sørensen
Jens N. Sørensen

Sider: 98
Tabeller: 7
Referencer: 97

Technical University of Denmark
Department of Wind Energy
Nils Koppels Allé
Building 403
2800 Kgs. Lyngby
Denmark
Phone 45254340

mahj@dtu.dk
www.vindenergi.dtu.dk

FINITE VOLUME METHODS FOR INCOMPRESSIBLE NAVIER-STOKES EQUATIONS ON COLLOCATED GRIDS WITH NONCONFORMAL INTERFACES

Developments and applications

DMITRY K. KOLMOGOROV

Ph.D. thesis
Department of Wind Energy
Tehcnical University of Denmark

May 2014

Thesis title:

Finite Volume Methods for Incompressible Navier-Stokes Equations
on Collocated grids with Nonconformal Interfaces

SUPERVISORS:

Lektor Wen Z. Shen

Professor Niels N. Sørensen

Professor Jens N. Sørensen

© Dmitry K. Kolmogorov 2014

ABSTRACT

Direct numerical solutions of the Navier-Stokes equations using Computational Fluid Dynamics methods are recognized as some the most advanced and accurate methods for prediction of flows around wind turbines. The ability of these methods to capture the dynamics of the complex flow properties appearing in the immediate vicinity of a wind turbine rotor makes them invaluable tools in the field of wind energy. Since direct computations of a fully resolved flow around a wind turbine are computationally expensive, a typical requirement for a good CFD method is that it is able to predict the flow field efficiently without jeopardizing the accuracy. In this thesis, some fundamental developments of direct CFD methods are presented to provide a platform for the development of sliding grid method for wind turbine computations. As one of the most prospective CFD methods for incompressible wind turbine computations, collocated grid-based SIMPLE-like algorithms are developed for computations on block-structured grids with nonconformal interfaces. A technique to enhance both the convergence speed and the solution accuracy of the SIMPLE-like algorithms is presented. The erroneous behavior, which is typical for some commonly used mass flux interpolations, is estimated, and a new interpolation technique, which eliminates these errors, is developed together with fully consistent SIMPLE-like algorithms. For the algorithms, both the accuracy and the convergence rate are shown to be higher than standard versions of the SIMPLE algorithm. The new technique is implemented in an existing conservative 2nd order finite-volume scheme flow solver (EllipSys), which is extended to cope with grids with nonconformal interfaces. The behavior of the discrete Navier-Stokes equations is discussed in detail and the developed technique, which exhibits both low implementation costs and high efficiency of the numerical scheme, is presented. A Geometric Multigrid method of the EllipSys flow solver is fully extended to block-structured grids with nonmatching blocks. An Optimized Schwarz method employed for the Incomplete Block LU relaxation scheme is shown to possess several optimal conditions, which enables to preserve high efficiency of the multigrid solver on both conformal and nonconformal grids. The developments are done using a parallel MPI algorithm, which can handle multiple numbers of interfaces with multiple block-to-block connectivity.

ABSTRACT (IN DANISH)

Direkte numerisk løsning af Navier-Stokes ligninger ved hjælp af Computational Fluid Dynamics (CFD) er anerkendt som en af de mest avancerede og præcise metoder til forudsigelse af luftstrømninger omkring vindmøller. Evnen af disse metoder til at indfange dynamikken i de komplekse strømninger, som optræder i umiddelbar nærhed af en vindmøllerotor, har gjort dem til uvurderlige værktøjer til forudsigelse af lokale vindfelter. Da direkte beregninger af en fuldt opløst strømning omkring en vindmølle er beregningsmæssigt dyre, er et typisk krav til en god CFD metode, at den er i stand til at forudsige strømningsfeltet effektivt og hurtigt uden at gå på kompromis nøjagtigheden. I nærværende afhandling præsenteres og udvikles grundlaget til at benytte glidende beregningsnet (sliding meshes) i forbindelse med løsningen af de inkompressible Navier-Stokes ligninger. Metoden er baseret på SIMPLE-teknikken, som videreudvikles til at beregne blok-strukturerede beregningsnet med ikke-konforme grænseflader. De fejl, som typisk optræder i forbindelse med masseflux-interpolationer er analyseret, og har ledt til udviklingen af en ny teknik til eliminering af fejlene. Den udviklede algoritme giver både en højere nøjagtighed og en bedre konvergens end 'standard' SIMPLE algoritmer. Den nye teknik er implementeret i en eksisterende 2. ordens finite-volume Navier-Stokes løser (EllipSys), som er udvidet til at behandle beregningsnet baseret på ikke-konforme grænseflader. I forbindelse med valideringen af den nye teknik, analyseres og diskuteres de opnåede numeriske løsninger i detaljer. Det vises ligeledes hvordan teknikken, i forbindelse med implementeringen i EllipSys koden, er i stand til at inkludere en effektiv multigrid-teknik på ikke-konforme, blok-strukturerede beregningsnet. Dette er gjort ved hjælp af en parallel MPI algoritme, som er i stand til at håndtere grænseflader med flere blok -til- blok forbindelser.

PUBLICATIONS

Journal articles

- Dmitry K. Kolmogorov, Wen Z. Shen, Niels N. Sørensen, Jens N. Sørensen. "Fully Consistent SIMPLE-Like Algorithms on Collocated Grids." Numerical Heat Transfer Part B: Fundamentals, Vol. 67, 2015, p. 101-123.

Conference proceedings

- Dmitry K. Kolmogorov, Wen Z. Shen, Niels N. Sørensen, Jens N. Sørensen. "Fully consistent CFD methods for incompressible flow computations." Journal of Physics: Conference Series (Online), Vol. 524, 012128, 2014.
- Dmitry K. Kolmogorov, Niels N. Sørensen, Wen Zhong Shen, and Jens Nørkær Sørensen. "Multigrid technique and Optimized Schwarz method on block-structured grids with discontinuous interfaces." Proceedings of 21st AIAA Computational Fluid Dynamics Conference, 2013.

Truly knows the law of life only one
who does what he thinks is the law of life.

— Leo N. Tolstoy, Path of Life, 1910

ACKNOWLEDGMENTS

This thesis is the result of three years of work from February 1, 2011 to May 9, 2014 at the Department of Wind Energy, Technical University of Denmark in Copenhagen, Denmark.

First and foremost, my deep and sincere gratitude goes to my supervisors, Wen Zhong Shen, Jens Nørkær Sørensen and Niels N. Sørensen, for this guidance and support during these years.

In particular, I would like to thank my supervisor Jens Sørensen, his opened and positive mind set created an excellent working atmosphere at our department. The door of his office was always opened for discussions and the guidance he gave me was truly invaluable.

I what to thank my first supervisor Wen Zhong Shen for our discussions and help in writing this thesis. We sometimes had opposite points of view, which enabled us to look at problems from different perspectives.

Special thanks to Niels N. Sørensen for his endless encouraging support for these years. During our meetings, you always gave me very constructive suggestions and asked questions, which led to the heart of problems.

I also wish to express my thanks to Robert Mikkelsen for providing me with technical and helpful literature. Special thanks to my colleges, PhD students and stuff at the Department of Mechanical Engineering and Department of Wind Energy, especially Ruth Svane Vestergård and Marianne Hjorthede Arbirk.

I express my deepest and warmest gratitude to my beloved wife Maria, without who this work would not have been possible.

Thank you!

Dmitry K. Kolmogorov
Copenhagen, May 2014

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AD	Actuator Disk	2
ADL	Actuator Disk/Line	2
AL	Actuator Line	2
AMG	Algebraic Multigrid	6
BEM	Blade Element Momentum	1
CFD	Computational Fluid Dynamics	1
CG	Conjugate Gradient	9
CS	Classical Schwarz	10
CV	Control volume	54
DES	Detached Eddy Simulation	3
DDES	Delayed Detached Eddy Simulation	3
GMRES	Generalized Minimal Residual	9
GMRESR	Restarted Generalized Minimal Residual	67
LES	Large Eddy Simulation	2
MG	Multigrid	
MPI	Message Passing Interface	51
IBLU	Incomplete Block LU	12
ILU	Incomplete LU	
OpenMP	Open Multi-Processing	63
OS	Optimized Schwarz	11
PC	Pressure Correction	58
PISO	Pressure implicit with splitting of operator	8
PS	Parameterized Standard	20
SIMPLE	Semi-Implicit Method for Pressure Linked Equations	8
SIMPLEC	"Consistent" SIMPLE	8
SIMPLER	Revised SIMPLE	
RANS	Reynolds Averaged Navier-Stokes	2
VAWT	Vertical Axis Wind Turbine	4

LIST OF NOTATIONS

	Greek symbols	
α	size of velocity underrelaxation parameter	19
β	parameter used to ensure compatibility with SIMPLE ($\beta = 0$) or SIMPLEC ($\beta > 0$) algorithms	19
γ	parameter used to ensure compatibility with SIMPLE ($\gamma = 0$) or SIMPLEC ($\gamma = 1$) algorithms	19
λ	Robin parameter	69
μ	dinamic viscosity	3
ρ	density	3
τ	length of time step	18
	Subcripts	
K	neighbor cell center counter	18
k	control volume edge counter	18
P	control volume center	18
	Superscripts	
m	subiteration counter	19
n	time step counter	18
	Lower-case symbols	
$\vec{b}_P^m$	vector of explicit terms grouped together	19
$\vec{s}_v$	vector accounting the volumetric forces	3
$\vec{v}$	velocity vector	3
a_k	nondiagonal term of matrix of pressure-correction equation	63
a_p	diagonal term of matrix of pressure-correction equation	63

f mass flux in expression of continuity equation 19

p pressure 3

t time 3

Apper-case symbols

A_P diagonal term accounting for the discrete convective and diffusion terms of matrix of momentum equations 18

$\widetilde{A_P}$ $A_P/\alpha + 1.5A_P^V$ 19

$\widetilde{\widetilde{A_P}}$ $A_P(1/\alpha - \gamma) + 1.5A_P^V$ 20

A_P^V $1.5\rho \mathrm{d}V_P/\tau$ 18

$\mathrm{d}V_P$ control volume size 18

$\partial\Omega$ control volume edges shared with the cells of the original domain 51

$\partial\Omega_\mathrm{I}$ control volume edge segments shared with the cells of the neighbor domain . . 51

$\vec{S}$ vector of source terms 18

A_K nondiagonal term accounting for the discrete convective and diffusion terms of matrix of momentum equations 18

C mass flux in expression of momentum equations 52

F^{CV} finite volume flux accounting for convection terms of momentum equations . . . 52

F^{DF} finite volume flux accounting for diffusion terms of momentum equations 52

F^P finite volume flux accounting for pressure-gradient term of momentum equations 52

$N(k)$ center of the neighbor cell sharing the cell face k 57

1 INTRODUCTION

1.1 INTRODUCTION

Wind energy is recognized as the most prospective, inexhaustible and wasteless source of energy in short- and long- term outlook. The diameter of the largest wind turbine exceeds the wingspan of the largest aircraft, whereas the height of the tower exceeds the length of a football field. The tremendous development of wind turbines in the last decades is the result of a collaborative work between the structural, mechanical, environmental and electrical engineers and the researchers in applied and fundamental sciences.

Increasing of the size of wind turbines creates new challenges which have to be faced when predicting the wind energy cost. Indeed, the aerodynamics of wind turbines is inherently unsteady and three-dimensional. Wind turbines experience a variety of fatigue loads induced from the atmospheric turbulence, the wind shear of the atmospheric boundary layer and yaw operation. Predicting extreme operating conditions, such as high wind speed, strong gusts and emergency shutdown become essential when new wind turbines are designed.

In mechanical engineering and applied sciences Computational Fluid Dynamics (CFD) methods are recognized as the most accurate methods for the prediction of flow fields over wind turbines. CFD methods are invaluable tools for prediction of complex three-dimensional effects appearing on the wind turbine rotor, but are nevertheless computationally costly. Since the direct numerical computations of flows around wind turbines are still computationally expensive, a typical requirement for a good CFD method is the ability of predicting the flow field on relatively coarse computational grids both accurately and efficiently.

CFD methods for wind turbine computations have gained a substantial development in the last two decades, as described in the following section.

1.2 CFD METHODS FOR WIND TURBINE AERODYNAMICS

Numerical models for predicting the wind turbine aerodynamics are divided into three categories, such as far wake-, near wake- and direct rotor/tower modeling.

The simplest approach used for the description of the far wake aerodynamics is the analytical kinematic method which exploits the self-similar nature of the wake to obtain expressions for the velocity deficit and the turbulence intensity. Alternatively, the Blade Element Momentum (BEM) method by Glauert [25] is based on the global momentum balance together with the blade element theory to calculate aerodynamic blade characteristics. In the method the wake behind a wind turbine is modeled using the quazi 1D momentum theory. In spite of the simplicity of the BEM method it is still popular. By adding a series of corrections and semi empirical models matching with experimental data can be achieved using the BEM theory. Alternatively, vortex-lattice, vortex-particle methods are based on inviscid incompressible flow and model the blades using lifting lines/surfaces with blade elements, whereas the wake is modeled using vorticity sheets of vorticity particles. Similarly to the vortex-lattice, vortex-particle methods a more accurate approach exist, called as panel method, which models the blade geometry using a surface mesh and in the vicinity of the blade a boundary layer code is used, whereas the flow in the rest of the domain is considered as inviscid and incompressible. A detail comparison between the methods above can be found in the review papers such as Hansen and Madsen [30], Sanderse et al. [69], Hansen et al. [31], Snel [74, 75], Crespo et al. [12] and Vermeer et al. [87].

Similarly to the BEM theory, the aerodynamic loads can be determined from Actuator Disk (AD) and Actuator Line (AL) methods using tabulated airfoil data. But contrary to the BEM method, the Actuator Disk/Line (ADL) methods solve Navier-Stokes equations on a computational grid in the near and the far wake regions using Reynolds Averaged Navier-Stokes (RANS)/Large Eddy Simulation (LES) turbulence models. In the method the rotor geometry is not resolved as the rotor is modeled as source terms in the momentum equations.

Direct numerical methods applied to rotor/tower computations together and the ADL methods are commonly refereed to Computational Fluid Dynamics (CFD) methods. In the CFD methods the Navier-Stokes equations are solved to provide the most accurate prediction of the wind turbine aerodynamics.

In this thesis the direct CFD methods will be considered and some fundamental developments will be presented providing a framework for further developments of the CFD methods for accurate and efficient wind turbine computations. In the rest of the chapter, the CFD methods commonly employed for full rotor computations are discussed and the most prospective methods are pointed out.

1.3 DIRECT CFD METHODS FOR FULL ROTOR COMPUTATIONS

The prediction of three-dimensional turbulent flows over wind turbines is the task which can be solved using the full rotor computations with direct CFD methods. The direct CFD methods are the most accurate tool for this type of problems, but still the full rotor computations is a challenging task. In the computations boundary-fitted grids are commonly used and the complex turbulent flow is captured in the vicinity of the rotor blades, tower and nacelle. As the blades are rotated relatively to the fixed earth and the tower, the full rotor computations require special techniques for treating the moving boundaries. During the last decade, several approaches have been successfully used, among which the most prospective methods are overset grid and sliding grid methods.

Wind turbine aerodynamics is commonly assumed to be governed by incompressible Navier-Stokes equations as below:

$$\vec{\nabla} \cdot \vec{v} = 0 \tag{1.1a}$$

$$\rho \frac{\partial \vec{v}}{\partial t} + \rho(\vec{\nabla} \cdot \vec{v})\vec{v} - \mu \nabla^2 \vec{v} = -\vec{\nabla} p + \vec{s}_v \tag{1.1b}$$

where ρ denotes density, $\vec{v}$ is the velocity vector, p is pressure, μ is the dynamic viscosity and $\vec{s}_v$ is the vector accounting the volumetric forces. To solve the Navier-Stokes equations at a high Reynolds number, various direct methods have been exploited during the last decades, among which the most popular methods are based on space- or time-averaged Navier-Stokes equations (RANS and LES) or the hybrid RANS/LES (Detached Eddy Simulation (DES), Delayed Detached Eddy Simulation (DDES)) methods. When the methods are applied to the direct wind turbine computations, different techniques can be used, as described below.

In 1999 the flow over the NREL Phase II rotor was computed using compressible overset grid method by Duque et al. [16]. The wind turbine with zero yaw was modeled with the nacelle and tower and promising results were shown for the rotor/tower interaction.

Later, accurate predictions of the NREL Phase IV rotor were performed by Sørensen et al. [80] and Johansen et al. [38] who studied 3D aerodynamic effects as a function of wind speeds by using the multiblock finite volume, incompressible flow solver EllipSys3D with a rotor-only configuration. Collocated grid arrangement was employed and the pressure-velocity coupling was ensured by using SIMPLE-like algorithm with the Rhie and Chow [65] interpolation method. The CFD approach has shown to have good qualitative and quantitative agreement with experimental measurements.

To decrease the computational costs Xu and Sankar [92] proposed an approach where small zones surrounding each blade were modeled using 3D unsteady Navier-Stokes equations, whereas the rest of the domain was treated using a significantly less expensive full potential solver. Rather than modeling the entire rotor Pape and Lecanu [62] modeled a single blade omitting the tower and nacelle using compressible cell-centered finite volume method exploited in elsA software developed by ONERA.

An extensive wind turbine aerodynamic study was presented by Duque et al. [17], who performed computations of the NREL Phase VI turbine with the NASA compressible overset grid-based finite differences Overflow-D solver. Various characteristics of the wind turbine performance were discussed including shaft power, normal force and pressure coefficient.

To predict the rotor-tower interactions, computations of the fully resolved rotors were performed by Zahle et al. using incompressible overset grid method in [96]. For the computations collocated grid-based PISO scheme was employed and to preserve the pressure-velocity coupling the Rhie and Chow [65] interpolation was used. To ensure convergence of the multigrid method which is used for the solution of the pressure-correction equation, explicit flux correction was employed to ensure mass flux conservation between the overlapping domains. Using the method unsteady interactions between the rotor blades and the tower were captured and good agreement with experimental data was shown.

A recent application of the overset grid method for computation of the NREL phase VI wind turbine was presented by Li et al. [44] using incompressible finite-difference scheme with PISO algorithm. Good predictions of power and thrust, force coefficient and local pressure coefficient were reported.

Contrary to the overset grid methods, the sliding grid/nonconformal grid concept was also employed for wind turbine computations. Compared to the overset grid, in the sliding grid method the grids do not overlap: one domain is usually set to be stationary whereas another domain slides along the interface between the domains.

Gomez-Iradi et al. in [26] employed a compressible sliding grid method for computations of the flows around NREL Phase IV. The finite-volume method using structured grids was used and to couple the domains at the sliding interface a nonconservative interpolation was used based on the so-called ghost cell interpolation.

Contrary to the compressible sliding grid method, the incompressible sliding grid method for computations of wind turbine aerodynamics was demonstrated only for the Vertical Axis Wind Turbine (VAWT) and using commercial flow solvers. An example of the flow computations over a VAWT are presented by Yao et al. [93] and D'Alessandro et al. [14], where the commercial FLUENT flow solver based on the SIMPLEC algorithm was used. Using the other commercial flow solver StarCD, results of the computations based on incompressible sliding grid method with the PISO scheme were demonstrated by Guerri et al. [28].

1.3.1 *Sliding and overset grid methods*

Based on the overview in the previous section, it can be concluded that the overset grid and the sliding grid methods are one of the most advanced CFD methods, which were used for full rotor computations of wind turbines in the last decade.

The two methods are based on boundary fitted grids, which are capable of resolving the boundary layers around moving rotor blades and steady tower and nacelle. A typical grid for the methods is composed of two parts: one grid associated with the rotor is moving with the rotor, whereas the other grid associated with the tower, nacelle and ground is fixed.

To enable the relative motion between the rotor and the tower the overset grid and the sliding grid methods are based on the decomposition of a global domain into two distinct parts:

- In the sliding grid method or the method on moving grids with nonmatching blocks, the global domain is decomposed into a system of nonoverlapping grids, such that these grids are able to slide along a common interface. At the interface the flow field is transferred by conservative or nonconservative interpolation, see e.g. Gonzalez et al. [27], Usera et al. [85], Basara et al. [3], Steijl and Barakos [83].

- In the overset or Chimera grid method, the global domain is decomposed into a system of geometrically simple overlapping grids. To transfer the flow field between these domains, boundary information is exchanged in the overlapped regions via interpolation of the flow field available in the neighbor domains, see e.g. Meakin and Gomez [52, 53], Zahle [95].

Main strengths and weaknesses of the sliding and the overset grid methods are discussed in the following.

The overset method, originally developed for Euler equations, has been first applied to compressible Navier-Stokes equations by Atta [2], Steger et al. [82] and Buningt [7]. Several overset solvers were developed since then, such as the compressible NASA code OVERFLOW-D developed by Meakin and Gomez [52, 53] and the compressible/incompressible NASA Langley's code FUN3D solver. When dealing with the incompressible Navier-Stokes equations using the overset grids, special techniques preserving convergence of the Poisson pressure-correction equation are often required. One of the first incompressible overset grid methods was presented by Chesshire and Henshaw [10] and Henshaw [32]. In their work a damping term was added to the pressure-correction equation for the stability purpose. In Henshaw [33] the overset grid method was used with a multigrid method. The overlapped grids were incorporated on all grid levels of the multigrid method, providing high convergence rate of the solver. Incompressible overset method was also employed by Hubbard and Chen in [34] for the computations of flows over a cylinder and a turbulent flow over a submarine configuration. The SIMPLER/PISO scheme was used and mass fluxes at the overset interface were corrected explicitly to maintain the mass conservation before the pressure-correction is solved. A fractional step with overset grid methods was presented by Zang and Street [97] and Burton and Eaton [8] for incompressible flow computations. To preserve the convergence of the pressure Poisson equation, an explicit correction of fluxes between overlapped grids was used, such that the discrete continuity equation is satisfied. Similar mass flux correction was employed by Zahle [95] to preserve the convergence of a geometric multigrid method.

Sliding grid method, contrary to the overset grid method, does not rely on the overlap between the domains. Strong coupling at the interfaces is commonly assured by using a conservative scheme as described by Lilek et al. in [49] or Ferziger and Perić in [20]. An example of sliding grid method can be consider by a so-called clicking grid method, which was presented in Wechsler et al. [88], where incompressible multiblock solver with collocated grid arrangement was used and to ensure the pressure-velocity coupling SIMPLE algorithm was employed based on the Rhie and Chow interpolation. A transient turbulent flow induced by a pitched-blade turbine in a baffled stirred tank was computed using two grids: one grid was rotated with the impeller, whereas the other grid was fixed to the baffles. Due to the clicking grid method the grid at the interface was matched at each time step, which ensured the simplicity of the algorithm. The method of Wechsler et al. had two limitations: the grid along the interface had to be equidistant and the time step had to be set such that the grid continuity at the interface is preserved at every time step.

Alternatively to Wechsler et al., Basara et al. [3] presented an unstructured grid-based incompressible finite-volume method using the collocated grid arrangement on sliding grids. The Navier-Stokes equations were solved in the absolute reference frame using a generic conservation law similarly to the fully moving grids by Demirdžić and Perić [15].

Finally, Usera et al. used a block-structured finite volume method using Algebraic Multigrid (AMG) method on sliding grids. The sliding grid method is based on the conservative finite-volume method treating the nonconformal interface similarly to Lilek et al. [49]. Navier-Stokes equations in the inertial reference frame were discretized and SIMPLE algorithm was used with the Rhie and Chow interpolation.

Sliding grid method has found wide applications in incompressible computations using commercial software, such as FLUENT, StarCD, CFX. Computations using SIMPLE/SIMPLEC algorithms were employed with conservative finite-volume schemes at the sliding interface in centrifugal pump and baffled tanks stirred by impeller as can be seen in Gonzalez et al. [27], Li et al.[45], Jaworski et al. [37] etc.

It can be concluded that, whereas the sliding grid methods have found applications in simulation of flows in baffled stirred tanks and centrifugal pumps, applications to the incompressible wind turbine computations was limited by computations of flows around a VAWT only.

Based on the discussion presented above, the advantages and disadvantages of the sliding grid method in comparison to the overset grid methods can be summarized as:

- Governing equations can be treated in a fully conservative form, ensuring good solution accuracy.
- Only one cell overlap can be used compared to big overlapping regions in the overset grids, preserving high solution efficiency.
- Faster solution of the pressure-correction can be ensured, compared to the explicit flux correction method needed for the overset method.
- Fast multigrid method can be used, contrary to the overset methods solving the pressure-correction equation on the finest grid level only.

The disadvantages are:

- A complex grid generation is required involving body-fitted grids, which are connected to the grid in the rest of the domain by means of an interface.
- Strong coupling at the interface requires an accurate and usually conservative schemes, for which the implementation cost is higher than for the overset grids.

Combining the advantages of the sliding grid method together with the numerous results using commercial flow solvers, it can be seen that the SIMPLE-based incompressible sliding/nonconformal grid method is one of the most prospective methods for wind turbine computations.

In this thesis we will follow the sliding grid concept and present development of an accurate and efficient incompressible finite-volume/multigrid method for applications on block-structured nonoverlapping nonconformal grids. The method is based on a newly developed SIMPLE-like algorithm with the mass flux interpolation enhancing both the convergence rate and the solution accuracy of the algorithm. An efficient extension of the block-structured multigrid solver of EllipSys code will be presented for applications on nonconformal and nonoverlapping block-structured grids. Contrary to incompressible overset grid methods, where solution of multigrid solver is complected, the method presented in this thesis enables solution of the pressure-correction equation on all grid levels of the multigrid solver for the computational cost, which is truly *identical* to the cost on conformal nonoverlapping grids.

Development of the accurate and efficient computational scheme requires that: 1) the interfaces are treated fully implicitly and conservatively; 2) the pressure velocity coupling is ensured in the whole domain including the interfaces; 3) specially designed domain decomposition ensuring high convergence rate of the multigrid solver on grids with nonconformal interfaces is used.

Below, we will discuss the main developments that proceeded this work in the field of incompressible conformal/nonconformal SIMPLE-like methods and formulate their main difficulties, challenges and development prospects.

1.4 SIMPLE-LIKE ALGORITHMS

Contrary to staggered grid arrangement, the collocated grid arrangement is known to be advantageous when non-orthogonal body-fitted grids are employed. In the sliding/nonconformal grid method body-fitted grids are used and it is natural to discretize the incompressible Navier-Stokes equations on collocated grids. It is known, that the vast majority of the incompressible CFD tools are based on SIMPLE-like algorithms. The main complexity of the algorithms, when applied on grids with the collocated arrangement, is the well known problem of the pressure-velocity decoupling. To overcome the problem, most of the methods used today in commercial and open source codes are based on the so-called momentum interpolation methods, initially proposed by Rhie and Chow [65].

1.4.1 *Pressure-velocity coupling*

Momentum interpolation, originally developed by Rhie and Chow [65] in 1983, laid a foundation for various computational codes on collocated grids. The simplicity and efficiency of the interpolation became the reason that most of the commercial and the open source codes on collocated grids are still based on the interpolation.

Since the last two decades various modifications of the momentum interpolations have been presented in order to ensure an accurate solution on collocated grids. Originally the Rhie-Chow interpolation was developed for steady flow computations and is known to possess a dependence of velocity underrelaxation parameter at convergence. The problem with the dependence was solved in [51, 57], nevertheless it was later shown in [72, 94] that if the Rhie-Chow interpolation is used for unsteady flow computations, pressure wiggles appear for small time steps. An interpolation method for unsteady flow computations free from the pressure wiggles was later proposed independently by Choi [11] and Shen et al. [72]. Note that the method of Shen et al. possesses the same properties as the method of Choi, but contrary to the Choi's method, it is based on second order scheme in time. As shown in [42] both methods possess a dependence of time step and relaxation parameter at convergence, but the magnitude of the dependencies in practical applications has not been estimated yet.

To overcome the difficulty, several methods, which are independent of time step and relaxation parameter, were proposed in [13, 46, 63, 94]. Nevertheless, comparative analysis of the methods is rarely met in literature. Nowadays in spite of the existence of the time step independent methods, the standard methods of Choi and Shen et al. are widely used as can be seen for example in [1, 40, 41, 54].

In most of the literature, when incompressible collocated grid-based SIMPLE-like algorithms are employed, interpolation methods are often chosen rather to ensure the pressure-velocity coupling, than to ensure compatibility with a specific type of the SIMPLE algorithm (Semi-Implicit Method for Pressure Linked Equations (SIMPLE) [64], "Consistent" SIMPLE (SIMPLEC) [86] and Pressure implicit with splitting of operator (PISO) [36]). There exist an interpolation method of Shen et al. [73] which represents the only interpolation specially adopted for the SIMPLEC algorithm. The interpolation was stated to be consistent with the SIMPLEC algorithm, but the advantage of the consistency is rather unknown. Moreover, the interpolation is time step dependent at convergence and also depends of an additional parameter, which is used to ensure the convergence to steady state.

1.4.2 *SIMPLE algorithms on nonconformal grids*

The SIMPLE-like algorithms, when applied on nonconformal grids, have to be able to solve the incompressible Navier-Stokes equations on grids, which don't match at the common interfaces. Among finite volume methods there exist conservative and nonconservative approximations of the governing equations on the interfaces. The principles of the conservative finite volume scheme on nonconformal block-structured grids were presented in [47, 48] by Lilek et al. and Seidl [71]. Application of the incompressible conservative finite-volume scheme on unstructured grids with nonconformal interfaces can be found in Basara et al. [3]. Nonconservative scheme at the nonconformal interface is rarely met in literature when SIMPLE-like algorithms are employed. Examples of the nonconservative scheme application can be found only in compressible flow computations e.g. by Gomez-Iradi et al. [26] and Steijl and Barakos [83].

One of the issues in regards to solution of incompressible Navier-Stokes equations on nonconformal grids, is to ensure accurate and fast solution of the continuity equation. Due to incompressibility constrain, the convergence rate of the solution of the pressure-correction equation strictly depends of the type of the mass interpolation at the nonconformal interfaces. There are two possible types of the mass flux interpolation on the interfaces: conservative and nonconservative. The nonconservative interpolation can be considered by taking an example of a block-structured finite-volume method with so-called ghost or virtual cells [83] used to transfer the data between the grid blocks. If the blocks are nonmatching, the flow field variables in the ghost cells can still be found using interpolation from the nearest cell centers of the neighbor domain. Without any special treatment, the resulted finite-volume fluxes or mass fluxes are not consistent: the sum of the fluxes from one side of the interface does not equal to the sum of the fluxes from the other side of the interface. Nevertheless, this approach is easy to implement as was exploited in compressible flow methods by Gomez-Iradi et al. [26] and Steijl and Barakos [83]. But in this case the strong coupling between domains is violated which can be become a serious problem for incompressible flow methods. When the nonconservative approach is used, artificial source term in the continuity equation appears and explicit correction is required, thus decreasing the convergence rate. This type of the problem encountered widely in incompressible overset grid computations as can be seen in Zahle [95].

It is concluded, that the nonconservative flux interpolation in incompressible flow computations is probably not the most optimal solution when overset grids or grids with nonconformal interfaces are used. Contrary to the nonconservative approach the conservative approach is free from this drawback. But, when the conservative scheme is employed, accurate computation of the fluxes at the nonconformal interfaces is required together with a local grid adaptation at the interface, preserving the mass flux conservation both locally and globally, as can be seen in Basara et al. [3].

1.5 MULTIGRID METHODS

Solution of the continuity equation is the crucial part of the SIMPLE algorithm, which become also important when grids with nonconformal interfaces are employed. According to the SIMPLE algorithm, a Poisson-type pressure-correction equation has to be solved. As the direct solution of the pressure-correction equation is computationally expensive, the natural choice to solve it is to use iterative methods. As known, the basic iteration methods such as Gauss-Seidl method, Jacoby method, ILU factorization and others can easily cope with high frequencies of the errors, whereas for the low frequencies a significant decrease or even stagnation of the convergence rate is observed for the methods. Krylov type methods, such as Conjugate Gradient (CG) and Generalized Minimal Residual (GMRES) methods, have better convergence for the low frequencies, but due to the high computational complexity of the mehtods the general computational cost is still high enough. An alternative to the Krylov type methods can be using a preconditioner, which may drastically speed up their performance.

To enhance the convergence speed of iterative methods, the basic iteration methods and the preconditioned Krylov-type methods can be combined in the so-called multigrid method. Contrary to the standard iteration methods, the idea of the multigrid method is to solve governing equations using a series of coarsened grids, on which the "coarsened" solution is solved and then is interpolated back to the original grid. The advantage of the method is that by using the coarse grids it is possible to eliminate long wave frequencies in a much shorter computational time.

Starting from the pioneer work of Fedorenko in 1964 [19], first practical application of the multigrid methods was presented by Brandt in [5, 6]. In 1976 multigrid method was reinvented by Hackbusch [29], who developed theoretical foundation and extended the multigrid to a wide range of problems. The main advantage of the multigrid is that the its computational cost is nearly independent of the number of grid nodes. By using the method significant enhancement of the convergence speed can be achieved, which was reported by many authors, see e.g. the review papers of Wesseling and Oosterlee [91] and Wesseling [89].

A typical multigrid method consists of the following components:

- operator of relaxation to smooth errors on coarse grids
- operators of coarsening and prolongation between grid levels
- operator for solution of the problem on the coarsest grid level

To enable performance of the multigrid method on multiprocessor computers, domain decomposition methods are often required. State of the art developments in this field are discussed in the following section.

1.6 OPTIMIZED SCHWARZ METHODS

To apply computational methods on multiprocessor computers there is a common approach to use a Classical Schwarz (CS) domain decomposition method [70], where the global domain of the original problem is subdivided into overlapped subdomains and then each of the subdomain problems is solved in parallel using Dirichlet transition condition at the internal boundaries. When solving pressure Poisson equation the problem of the CS method is that it leads to convergence rate which is not uniform in respect to frequency: the high frequency components converge rapidly, whereas low frequency components converge only slowly. Moreover, the convergence rate of the method strongly depends on the size of the overlap between the subdomains.

Contrary to the CS method, in 1990 Lions [50] proposed a method, which uses a non-overlapping domain decomposition with Robin boundary conditions at the internal boundaries. Lions theoretically proved the convergence of an elliptic equation with Robin boundary conditions for any positive Robin parameter. It was found later, that the convergence rate of the method depends significantly on size of the Robin parameter. To define the optimal Robin parameter several methods were proposed and analyzed in Gander [21], Gander and Golub [24] and the optimal transition operators between domains such as Steklov-Poincare or Neumann-Dirichlet map were intensively studied in Gander [21], Gander et al. [23], Nataf et al. [60]. Local approximations of the optimal operators led to the Optimized Schwarz (OS) methods which appeared to be faster than the CS method, as seen e.g. in [21, 59]. The problem of most known OS methods is twofold: they rely on cumbersome Fourier transformation, and they are only proven optimal for simple geometries and uniform grids[59]. For engineering purposes today the most appropriate approach is to use the Robin transition condition, which is a first order approximation of the transition operator of the Optimized Schwarz method.

When the Robin transition condition is used, there are several conditions which the Robin parameter has to fulfill. It was shown by Gander, Nataf in [21, 59] that on a uniform square grid for the two-subdomain problems the optimal value of Robin parameter has to scale as $O((Lh)^{-1/2})$, where h is a cell size and L is the characteristics length of the domain. However, when several domains are connected at one cross point this method leads to divergence. It was recently shown by Gander and Kwok in [22] that the convergence rate can be restored if Robin parameter at the cross points scales as $O(h^{-1})$.

Nevertheless, the choice of the Robin parameter ensuring robust and efficient performance of domain decomposition methods is an open question nowadays.

In literature there exist other transition conditions, which are different from the Robin condition. One of the alternatives is a generalized Schwarz alternating method which was proposed by Tang in [84]. In the method a weighted average of the Dirichlet and Neumann interface conditions was used and it was shown that a good choice of the weighting parameter leads to a significant speedup of the algorithm. Similarly in [66, 67] Rice et al. proposed and analyzed a transition condition, where Dirichlet problem was solved at every odd step and Neumann problem was solved at every even step. The boundary values and normal derivatives were computed based on a convex combination of the previous solutions on adjacent domains. The OS method was also shown to be closely related to the absorbing boundary conditions in [18] by Engquist and Zhao, where Robin and second order transition condition were analyzed and the the optimal interface parameters were determined asymptotically.

We conclude that nowadays only general scaling of the optimal Robin parameter is known. For instance, for the optimal convergence of Poisson equation, the Robin parameter in the regions away and close to the grid cross points has to scale as $O((h)^{-1/2})$ and $O(h^{-1})$, respectively. Specific definition of the Robin parameter has not been presented yet, therefore practical applications of the method is very limited in literature. Nevertheless, the Robin transition condition has a high potential in the scope of the multigrid methods, as by setting the Robin parameter properly, the range of low error frequencies can be eliminated faster. As elimination of the low frequency components is the bottleneck of the multigrid methods, employing the domain decompositions with Robin boundary conditions may serve as an acceleration technique of the multigrid methods.

1.7 ELLIPSYS2D/3D SOLVER

Developments done in the thesis are based on the platform of the incompressible finite-volume/block-structured EllipSys2D/3D code originally developed in DTU/RISØ by Jess Michelsen and Niels Sørensen [55, 77]. The code uses 2$^{\text{nd}}$ order backward-difference scheme in time and 2$^{\text{nd}}$ order central difference scheme in space, except for convective fluxes that are discretized with the QUICK upwinding scheme.

The discrete Navier-Stokes equations are treated in the prediction-correction form, such that momentum equation are solved at the prediction step and the continuity equation is solved at the corrector step using SIMPLE algorithm. The momentum equations are solved using two step Gauss Seidel/Jacobi solver, whereas the continuity equation is solved by multigrid solver. Collocated grid arrangement is employed and pressure-velocity coupling is ensured by using Rhie and Chow interpolation method.

The multigrid method of EllipSys2D/3D code is based on block-structured grid with one finest- and four successively coarsened- grids, which are constructed by removing every second point in each direction. The matrix of the pressure-correction on the coarse grids is composed using an intergrid flux conservation method of Michelsen [56]. Bilinear interpolation operator is employed to transfer solution from coarse grids to finer grid levels. On the coarsest grid level the pressure-correction equation is solved exactly using the Incomplete Block LU (IBLU)-preconditioned Conjugate Gradient (CG) method of Sonneveld et al. [76]. To smooth errors on coarse grid levels, an IBLU relaxation scheme is used based on a weakly overlapped two-step Optimized Schwarz (OS) method with Robin/Dirichlet interface condition on the internal boundaries between the grid blocks.

1.8 OUTLINE

As discussed in the previous sections, the collocated grid-based SIMPLE-like algorithms on grids with nonconformal interfaces have high potential for full rotor wind turbine computations. Solutions at the two sides of the interfaces have to be implicitly coupled due to lack of the grid overlap on the interfaces. It is essential then to ensure that the schemes used on this type of grids are conservative.

When grids are non-staggered, the pressure-velocity coupling can be ensured by using the so-called momentum interpolation. It is rarely seen and discussed in literature how a certain type of interpolations may influence the solution accuracy and the convergence speed of the SIMPLE-like algorithms. Thus, *"How to choose the momentum interpolation for a certain type of SIMPLE-like algorithms?"*,

- this is still an open question, for which we will try to give an answer in the first part of this thesis.

When the pressure-correction equation in SIMPLE-like algorithm is solved using multigrid method, it is a common requirement to have a robust and efficient relaxation scheme (or smoother). It was also discussed in Sec. 1.6, that a domain decomposition using Optimized Schwarz (OS) method is a prospective accelerator of the multigrid method. This will be confirmed in this thesis by employing the OS method of the EllipSys flow solver, which will be then extended to nonoverlapping grids with nonconformal interferences.

The main purpose of this thesis is to:

Develop efficient and accurate finite-volume and multigrid methods for incompressible Navier-Stokes equations on collocated grids with nonconformal interfaces and implement them in the EllipSys flow solver

The main requirements for the finite-volume method on grids with nonconformal interfaces, which will be fulfilled in this thesis, are:

- Mass flux interpolation on collocated grids is consistent, resulting to a time step independent solution at convergence
- SIMPLE-like algorithm is consistent with the mass flux interpolation
- Approximation is conservative with fully implicit treatment of nonconformal interfaces
- Mass fluxes are computed equally from both sides on nonconformal interface
- Efficient solution of the Poisson type pressure-correction equation is enabled by using Multigrid solver
- Algorithm is parallellized for applications on computers with distributed memory

As was discussed in the introduction, the system of the pressure-correction equations on grids with nonconformal interfaces has matrix which is sparse and irregular: neither the number of elements per row nor the bandwidth is constant. To solve the system, the original multigrid method in EllipSys code will be modified and full power of the multigrid will be preserved by fulfilling the following requirements:

- Solution procedure is enabled at all multigrid levels
- Coarse grid solver is able to solve the pressure-correction equation in the global domain

- Efficient domain decomposition is used at intermediate grid levels for relaxation scheme

The rest of the thesis is composed of three parts:

First, the standard mass flux interpolations on conformal grids with collocated arrangement are considered. A compatibility condition between the interpolations and the SIMPLE-algorithms is presented. Optimal parameters ensuring high efficiency of the methods are identified and recommendations for practical use are given. New interpolation method, independent of time step and relaxation parameter at convergence, is presented together with fully consistent SIMPLE-like algorithm. Both the accuracy and the convergence rate of the algorithm are shown to be higher, than of the a standard SIMPLE-like algorithm.

Second, a conservative 2nd order finite-volume scheme of the EllipSys2D code is extended to grids with nonconformal interfaces. Discretisation of the governing equations is discussed in detail and technique preserving low implementation costs and high efficiency of the scheme is presented. Parallelization technique is shown, ensuring performance on grids with nonconformal interfaces with multiblock connectivity. A Geometric Multigrid solver with Optimized Schwarz (OS) domain decomposition method is presented for applications on grids with nonconformal interfaces. The potential of the OS method is demonstrated in theory and in practice. Finally, the finite volume scheme and the multigrid solver are verified on the nonconformal grids, where good accuracy and efficiency of the computational scheme are demonstrated.

Third, concluding remarks and recommendations for the practical use are given, including the prospects of employing the developed computational scheme in sliding grid and overset grid methods.

Part I

MOMENTUM INTERPOLATION METHODS ON COLLOCATED GRIDS

In order to increase convergence rate of SIMPLE-like algorithms on collocated grids, a compatibility condition between the algorithms and the mass flux interpolations is presented. Interpolation methods existing in literature are classified as either being fully compatible with SIMPLE algorithm or being fully compatible with SIMPLEC algorithm. Numerical results of unsteady flow computations show that employing the SIMPLEC algorithm with the SIMPLEC-compatible interpolation results in a convergence rate up to 30% higher compared to the standard SIMPLEC algorithm. It is shown then that an appropriate choice of interpolation methods for the SIMPLE-like algorithm algorithm may not only enhance the convergence rate, but may also increase the solution accuracy of the algorithm. New interpolation method, independent of time step and relaxation parameter at convergence, is presented. It is shown that the SIMPLEC algorithm based on the new interpolation is about 10-25% more efficient than the standard SIMPLEC algorithm. The new interpolation is also shown to result in accurate solution, independent of time step at convergence.

2 STANDARD INTERPOLATION METHODS ON COLLOCATED GRIDS

2.1 INTRODUCTION

Semi-Implicit Method for Pressure Linked Equations (SIMPLE) and lots of its modifications, such as SIMPLE [64], SIMPLEC [86] and PISO [36], were originally developed for staggered grids, where mass flux interpolation is not necessary. Then the algorithms became widely used on grids with collocated grid arrangement, where mass flux interpolation is required. In most computational codes used in engineering application the choice of SIMPLE-like algorithm and the choice of mass flux interpolation method have been done independently [20, 90].

An exception from this practice is the SIMPLEC algorithm, proposed by Shen et al. [73], where an interpolation method specially adopted for the SIMPLEC algorithm was employed. The method is known to be dependent of an additional parameter β, which is used to ensure the convergence to steady state. In comparison to the original Rhie-Chow method the method of Shen et al. was shown to successfully preserve the pressure-velocity coupling on collocated grids, but the advantages of the method in comparison to other popular methods, such as Choi [11] or Majumdar [51], are rather uncertain.

In this chapter it is shown that in order to achieve a high efficiency of SIMPLE-like algorithms, the mass flux interpolation has to be used *consistently* with the algorithms. As representatives of two different interpolations, the standard interpolation methods of Choi and Shen et al. are considered and are formulated in a unified form. To enhance the convergence rate of the SIMPLE-like algorithms, a compatibility condition between the methods and the SIMPLE-algorithms is presented. Optimal parameters ensuring high efficiency of the methods are identified and the recommendations for practical use are given.

This chapter is organized as follows. First, the momentum and continuity equations are discretized and a parameterized form of the interpolation methods of Choi and Shen et al. is introduced. Second, a compatibility condition between interpolation methods and the SIMPLE-like algorithms is presented. Interpolation methods are classified as either fully compatible with SIMPLE algorithm or fully compatible with SIMPLEC algorithm. An interpolation method enhancing the convergence rate of the SIMPLEC algorithm will be pointed out. Finally, several computational experiments of steady and unsteady flows are presented.

2.2 DISCRETE GOVERNING EQUATIONS

In this section the momentum equations are discretized using the second-order difference scheme in time and second order central difference scheme in space, except for convective fluxes that are discretized with the QUICK upwinding scheme, implemented in EllipSys2D/3D [55, 77] solver. The continuity equation is discretized in a general conservative form based on the collocated grid arrangement.

2.2.1 *Momentum equations*

Using a second order backward difference scheme in time, the system of the momentum equations on collocated grids is expressed in a control volume P in the following form:

$$\begin{cases} \widehat{A_P} u_P^{n+1} + \sum\limits_{K=E,W,N,S} A_K u_K^{n+1} + (\partial p/\partial x)_P^{n+1} \mathrm{d}V_P = S_P^{x,n} + A_P^V (2u_P^n - 0.5u_P^{n-1}), \\ \widehat{A_P} v_P^{n+1} + \sum\limits_{K=E,W,N,S} A_K v_K^{n+1} + (\partial p/\partial y)_P^{n+1} \mathrm{d}V_P = S_P^{y,n} + A_P^V (2v_P^n - 0.5v_P^{n-1}) \end{cases} \tag{2.1}$$

where $\sum_{K=E,W,N,S}$ denotes the sum over the neighbor cell centers, as seen in Fig. 2.1, the term $\widehat{A_P}$ is defined as:

$$\widehat{A_P} = A_P + 1.5 A_P^V \tag{2.2}$$

and A_P , A_K are the diagonal and nondiagonal terms accounting for the discrete convective and diffusion terms of matrix of the momentum equations, A_P^V is the coefficient of the time derivative and equal to $1.5\rho \mathrm{d}V_P/\tau$, where τ is time step and $\rho \mathrm{d}V_P$ is control volume mass. The source terms $S_P^{x,n}$ and $S_P^{y,n}$ contain all explicitly treated terms in x- and y- directions, respectively.

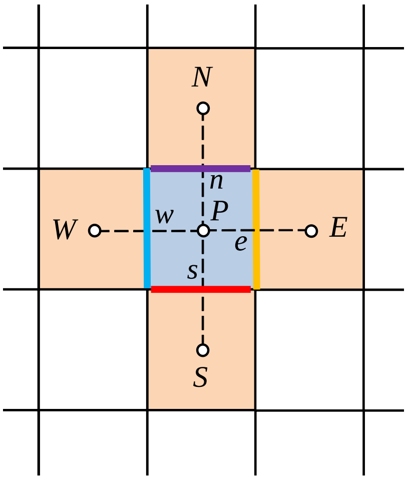


Figure 2.1: Compass notations for the neighbor cell centers and for the control volume faces.

Grouping together the two momentum equations of Sys. (2.1), the vector form of the momentum equations is obtained:

$$\widehat{A_P} \vec{v}_P^{n+1} + \sum_{K=E,W,N,S} A_K \vec{v}_K^{n+1} + \vec{\nabla} p_P^{n+1} \mathrm{d}V_P = \vec{S}_P^n + A_P^V (2\vec{v}_P^n - 0.5\vec{v}_P^{n-1}) \tag{2.3}$$

The velocity $\vec{v}_P$ in Eq. (2.3) usually has to be underrelaxed. The underrelaxation is applied only to the spatial term A_P similarly to [13, 72, 73]. The resulting momentum equations are written as follows:

$$\widetilde{A_P} \vec{v}_P^{m+1} + \sum_{K=E,W,N,S} A_K \vec{v}_K^{m+1} + \vec{\nabla} p_P^{n+1} \mathrm{d}V_p = \vec{b}_P^m \tag{2.4}$$

where all explicit term were grouped in $\vec{b}_P^m$:

$$\vec{b}_P^m = \vec{S}_P^m + A_P^V(2\vec{v}_P^n - 0.5\vec{v}_P^{n-1}) + (1/\alpha - 1)A_P\vec{v}_P^m \tag{2.5}$$

and $\widetilde{A_P}$ is defined as:

$$\widetilde{A_P} = A_P/\alpha + 1.5A_P^V \tag{2.6}$$

where α is the velocity underrelaxation parameter. The superscripts m and n are the subiteration- and time step- counters, respectively, such that the solution at time step $n+1$ is obtained at convergence. In order to compute the flow field at subiteration step $m+1$, the coefficients A_P, A_K and $\widetilde{A_P}$ are taken from the former subiteration step m. In the notations of the coefficients the superscript counter m is dropped for simplicity.

2.2.2 *Continuity equation*

The discrete continuity equation essentially represents the fact that the sum of mass fluxes through the control volume faces equals zero:

$$\sum_{e,w,n,s} f_k = 0 \tag{2.7}$$

where f denotes mass flux and the subscript k corresponds to control volume face positions. On collocated grids the cell face mass fluxes are not typically available. To identify the fluxes one of the widely used method is the momentum interpolation method originally proposed by Rhie and Chow [65].

2.3 PARAMETERIZED STANDARD (PS) INTERPOLATION

In the literature there exist many modifications of the Rhie and Chow interpolation, among which we consider two methods, proposed by Shen et al. [72] and Shen et al. [73]. Contrary to the popular interpolation of Choi [11] for unsteady flow computations, the two methods possess some advantages as described below.

The two methods of Shen can be formulated in a unified parameterized form. For this, one should first subtract a term $\gamma A_P\vec{v}_P^{m+1}$from the left and the right hand sides of Eq. (2.4), then add and subtract a term $\beta A_P\vec{v}_P^m$ from the right hand side of Eq. (2.4), where γ and β are constant parameters described below. After these manipulations the cell centered velocity can be expressed as below:

$$\vec{v}_P^{m+1} = \frac{A_P^V}{\widetilde{\widetilde{A_P}}}(2\vec{v}_P^n - 0.5\vec{v}_P^{n-1}) + (1/\alpha - 1 - \beta)\frac{A_P}{\widetilde{\widetilde{A_P}}}\vec{v}_P^m + \frac{\vec{h}_P^{m+1}}{\widetilde{\widetilde{A_P}}} - \frac{\mathrm{d}V_P}{\widetilde{\widetilde{A_P}}}\vec{\nabla}p_P^{m+1} \tag{2.8}$$

where the term $\widetilde{\widetilde{A}}_P$ and vector $\vec{h}_P^{\,m+1}$ are:

$$\widetilde{\widetilde{A}}_P = A_P(1/\alpha - \gamma) + 1.5A_P^V \tag{2.9}$$

$$\vec{h}_P^{m+1} = \sum_{K=E,W,N,S} A_K \vec{v}_K^{m+1} - \gamma A_P \vec{v}_p^{m+1} + \beta A_P \vec{v}_p^{m} + \vec{S}_P^{m} \tag{2.10}$$

Then applying the interpolation similar to the one described in [72], the following expression for the mass flux at some cell face k is obtained:

$$\begin{aligned} f_k^{m+1} = {} & \overline{\left[\frac{A^V}{\widetilde{\widetilde{A}}}\right]}_k \widetilde{f}_k^n + (1/\alpha - 1 - \beta)\overline{\left[\frac{A}{\widetilde{\widetilde{A}}}\right]}_k f_k^m \\ & + \overline{\left[\frac{\vec{h}^{m+1}}{\widetilde{\widetilde{A}}}\right]}_k \cdot d\vec{S}_k - \overline{\left[\frac{\mathrm{d}V}{\widetilde{\widetilde{A}}}\right]}_k \vec{\nabla} p_k^{m+1} \cdot d\vec{S}_k \end{aligned} \tag{2.11}$$

where $\overline{[\ \ ]}_k$ denotes linear interpolation from cell centers to cell face k [1]. In (2.11) the term $\widetilde{f}_k^n$ indicates $2f_k^n - 0.5f_k^{n-1}$, where f_k^n, f_k^{n-1}, f_k^m are the fluxes available from the former time steps n, $n-1$ and subiteration m. To ensure the pressure-velocity coupling on collocated grids the pressure force term $\vec{\nabla} p_k^{m+1} \cdot d\vec{S}_k$ in (2.11) has to be discretized based on 1δ pressure difference as in the original Rhie-Chow method [65].

The interpolation method defined in Eq. (2.11) will be called as Parameterized Standard (PS) interpolation. Using the parameters γ and β the *PS* method may be used in two forms which correspond to two interpolation methods existing in literature:

1. For $\gamma = 0$ and $\beta = 0$ the *PS* method exactly corresponds to the first method of Shen et al. [72]. In its turn Shen's method is similar to the widely used method of Choi [11], but contrary to Choi's method it is based on the second order backward difference scheme in time.

2. For $\gamma = 1$ and $\beta > 0$ the *PS* method exactly corresponds to another method of Shen et al. [73]. The method is rarely met in literature, but as shown later it may have several advantages over the first method. According to the results shown in Sec. 2.5, the *PS* method with and may enhance the convergence rate of the SIMPLEC algorithm up to 35%, compared to the *PS* method in the first form above.

According to Sec. 2.4.4, the *PS* method in the first and the second form is fully compatible with SIMPLE and SIMPLEC algorithms, respectively. Therefore, the first and the second forms of the *PS* method will be further referred to as PS_E and PS_C methods, respectively (see Table. 2.1), where the subscripts E and C refer to the last letters of the words "SIMPLE" and "SIMPLEC".

[1] For example on a uniform grid $\overline{[\phi]}_e = (\phi_E + \phi_P)/2$, where ϕ is a representative variable.

Forms of PS interpolation	Corresponding γ and β	Compatible with	Similar interpolations in literature
PS_E	$\gamma = 0, \beta = 0$	SIMPLE	Shen et al. [72], Choi [11]
PS_C	$\gamma = 1, \beta > 0$	SIMPLEC	Shen et al. [73]

Table 2.1: Notations for the SIMPLE- and SIMPLEC- compatible forms of PS interpolation.

Using the PS_E method accurate solution on collocated grids can be obtained and the pressure-velocity coupling is ensured for any time step, τ, including the infinitely small τ- values [72]. The method, in fact, has the disadvantage that at convergence to steady state the corresponding fluxes become weakly dependent of τ and α [13, 42, 46, 63, 94]. To get rid of the time step dependence several modifications of the PS_E method were developed in [13, 42, 46, 63].

The dependence of τ and α in the PS_E method arises from the interpolation of $\widetilde{\widetilde{A}}_P$ term, which contains both τ and α at convergence. Indeed, at convergence the cell centered velocities and the cell face mass fluxes become constant and (2.11) for the PS_E method becomes:

$$f_k = \frac{\overline{\left[\frac{\vec{h}}{\widetilde{\widetilde{A}}}\right]_k} \cdot d\vec{S}_k - \overline{\left[\frac{\mathrm{d}V}{\widetilde{\widetilde{A}}}\right]_k} \vec{\nabla} p_k \cdot d\vec{S}_k}{\overline{\left[\frac{A}{\widetilde{\widetilde{A}}}\right]_k}} \tag{2.12}$$

where f_k denotes to the mass flux at convergence. In notations of the variables in (2.12), the superscripts m and n are dropped due to convergence to the steady state. In (2.12) the dependences of τ and α are maintained in the $\widetilde{\widetilde{A}}_P$ term, as seen in (2.9).

Similarly to the PS_E method, the PS_C method is also τ- and α- dependent at convergence. But contrary to the PS_E method, it may result in steady state solution independent of τ in the special case where underrelaxation is not applied. Indeed, in this case the term becomes equal to $1.5\rho \mathrm{d}V_P/\tau$ and the steady state flux in (2.11) becomes independent of τ:

$$f_k = \frac{\overline{\left[\frac{\vec{h}}{\rho \mathrm{d}V}\right]_k} \cdot d\vec{S}_k - \frac{1}{\rho} \vec{\nabla} p_k \cdot d\vec{S}_k}{\overline{\left[\frac{A}{\rho \mathrm{d}V}\right]_k}} \tag{2.13}$$

The most important advantage of the PS_C method over the PS_E method is satisfaction of the compatibility condition with the SIMPLEC algorithm. By satisfying the condition (explained in detail in Sec. and verified in Sec.) the convergence rate of the SIMPLEC algorithm may be increased with up to 35%. In spite of the advantage in speed, there still exist a few unresolved issues relating to the PS_C method:

First, the method has the mass fluxes at convergence dependent on the additional parameter, which is employed to ensure *the finiteness* of the fluxes at convergence, as can be seen from (2.13). The dependence opens a question: "what is the optimum value of β, ensuring robustness, good accuracy and high convergence rate of the PS_C method?" The question was partially answered in the work of Shen et al. [73], whereas in this thesis an exhaustive analysis for the choice of the β will be performed.

Second, it should be noted that in real life applications applying underrelaxation (i.e. using $\alpha < 1$) may become necessary as the matrix of unsteady momentum equations is diagonally dominant only at small time steps. Moreover, if highly skewed grids are employed the velocity underrelaxation becomes unavoidable and the PS_C method becomes dependent of both the time step τ and the relaxation parameter α.

Finally, as was discussed above, for both the PS_E and the PS_C methods weak dependencies of time step and relaxation parameter exist. Today, the magnitude of the dependencies in real life applications is still unknown. In this thesis we will perform a deliberate analysis of the dependencies and present two new time step independent interpolation methods, which modify the original PS_C method.

2.3.1 *On the role of the parameter β in the PS^c method*

The PS_C method discussed in the previous section involves the usage of an additional positive parameter to ensure the finiteness of the fluxes at steady state (see Eq. (2.13)). For the PS_C method the requirement for the β parameter to be positive is important as only in this case the pressure-velocity coupling can be ensured. To see it, one may first rewrite the momentum equations, Eq. (2.4), at steady state as:

$$-\sum_{K=E,W,N,S} A_K\vec{v}_K - A_P\vec{v}_P = \vec{\nabla}p_P\mathrm{d}V_P \tag{2.14}$$

where the source term $\vec{S}_P$ was dropped for simplicity. Then, using Eqs. (2.10) and (2.14) the expression of the steady state flux of the PS_C method can be rewritten as:

$$f_k = \frac{1}{\beta}\frac{-\left[\overline{\left[\frac{\mathrm{d}V}{\overset{\approx}{A}}\right]_k}\vec{\nabla}p_k\cdot d\vec{S}_k - \overline{\left[\frac{\mathrm{d}V}{\overset{\approx}{A}}\vec{\nabla}p\right]_k}\cdot d\vec{S}_k\right]}{\overline{\left[\frac{A}{\overset{\approx}{A}}\right]_k}} + \frac{\overline{\left[\frac{A}{\overset{\approx}{A}}\vec{v}\right]_k}\cdot d\vec{S}_k}{\overline{\left[\frac{A}{\overset{\approx}{A}}\right]_k}} \tag{2.15}$$

The meaning of the β parameter can be seen clearly if we approximate equation above as below:

$$f_k \approx -\frac{1}{\beta}\overline{\left[\frac{\mathrm{d}V}{A}\right]}_k \left[\vec{\nabla} p_k - \overline{\left[\vec{\nabla} p\right]}_k\right] \cdot d\vec{S}_k + \overline{[\vec{v}]}_k \cdot d\vec{S}_k \tag{2.16}$$

Remark, that the term $\left[\vec{\nabla} p_k - \overline{\left[\vec{\nabla} p\right]}_k\right] \cdot d\vec{S}_k$ in (2.16) is the widely known Rhie-Chow correction term preventing the solution from check-board pressure distributions on collocated grids. Therefore, in (2.16), the inverse of β is seen to play the role of ensuring the pressure velocity coupling on collocated grids. The choice of β, which ensures both the finiteness of mass fluxes at convergence and the pressure velocity coupling on collocated grids, will be presented later in the result section.

2.4 COMPATIBILITY CONDITION

In this section interpolation methods on collocated grids will be classified either as being 1) fully compatible with SIMPLE algorithm or 2) fully compatible with SIMPLEC algorithm. To author's knowledge, the compatibility condition, presented below, has not been presented in literature. We will show later that the SIMPLE-like algorithms, which satisfy the condition, have a faster convergence rate than those not satisfying this condition.

2.4.1 *System of Navier-Stokes equations*

In this section the system of the Navier-Stokes equations composed of momentum equation, Eq. (2.4), and continuity equation, Eq. (2.7), will be reformulated. As an example of momentum interpolation methods, the Parameterized Standard (*PS*) interpolation, Eq. (2.11), will be employed to solve the continuity equation and derive the compatibility condition between the SIMPLE-like algorithms and the momentum interpolations on collocated grids.

First, the discrete continuity equation, Eq. (2.7), is reformulated. For this, a discrete divergence operator D_P is introduced, which computes the divergence at some control volume p by linear velocity interpolation:

$$D_P(\vec{v}) = \sum_{k=e,w,n,s} \rho\overline{[\vec{v}]}_k \cdot d\vec{S}_k$$

where $\overline{[\quad]}_k$ is linear interpolation from the cell centers to cell face k. By adding and subtracting $D_P(\vec{v}^{m+1})$ from the discrete continuity equation, Eq. (2.7), the following discrete continuity equation is obtained:

$$D_P(\vec{v}^{m+1}) + \underbrace{\sum_{k=e,w,n,s} \left[f_k^{m+1} - \rho\overline{[\vec{v}^{m+1}]}_k \cdot d\vec{S}_k\right]}_{I_P} = 0 \tag{2.17}$$

where the second term in Eq. (2.17) is denoted as I_P. A concrete form of the term depends on the interpolation method used for determining f_k^{m+1}. Independently of the choice of the interpolation method, I_P can be referred to a correction term, ensuring the pressure-velocity coupling on collocated grids. To see this, the PS method, Eq. (2.11), will be used below.

Using the fluxes of the PS method and the cell centered velocity, Eq. (2.8), the term I_P in (2.17) can be rearranged as:

$$I_P = B_P(p^{m+1}) - d_P^m \tag{2.18}$$

where the pressure terms are grouped into $B_P(p^{m+1})$:

$$B_P(p) = \sum_{k=e,w,n,s} \left[\overline{\left[\frac{\mathrm{d}V}{\widetilde{\widetilde{A}}} \vec{\nabla} p \right]}_k \cdot d\vec{S}_k - \overline{\left[\frac{\mathrm{d}V}{\widetilde{\widetilde{A}}} \right]}_k \vec{\nabla} p_k \cdot d\vec{S}_k \right] \tag{2.19}$$

and the remaining terms are arranged in d_P^m :

$$\begin{aligned} d_P^m \quad = & \quad \sum_{k=e,w,n,s} \left[\overline{\left[\frac{A^V}{\widetilde{\widetilde{A}}} \widetilde{\vec{v}^n} \right]}_k \cdot d\vec{S}_k - \overline{\left[\frac{A^V}{\widetilde{\widetilde{A}}} \right]}_k \widetilde{f_k^n} \right] \\ + & \quad (1/\alpha - 1 - \beta) \sum_{k=e,w,n,s} \left[\overline{\left[\frac{A}{\widetilde{\widetilde{A}}} \widetilde{\vec{v}}^m \right]}_k \cdot d\vec{S}_k - \overline{\left[\frac{A}{\widetilde{\widetilde{A}}} \right]}_k \widetilde{f_k^m} \right] \end{aligned} \tag{2.20}$$

where the velocity vector $\widetilde{\vec{v}^n}$ denotes $2\vec{v}^n - 0.5\vec{v}^{n-1}$.

The momentum equation, Eq. (2.4), and the continuity equation, Eq. (2.17), rearranged using Eq. (2.18), formulate the system of the Navier-Stokes equations on collocated grids as below:

$$D_P(\vec{v}^{m+1}) + B_P(p^{m+1}) \quad = d_P^m \tag{2.21a}$$

$$\widetilde{A_P} \vec{v}_P^{m+1} + \sum_{K=E,W,N,S} A_K \vec{v}_K^{m+1} + \vec{\nabla} p_P^{m+1} \mathrm{d}V_P \quad = \vec{b}_P^m \tag{2.21b}$$

It can be noted that the system of equations, Eqs. (2.21), is the general system of the Navier-Stokes equations on collocated grids, which may be solved by various iterative methods, such as Fractional Step, pressure-correction or other methods. In the continuity equation, Eq. (2.21a), the term B_p is the Rhie-Chow correction term ensuring pressure-velocity coupling on collocated grids, whereas the term d_P^m is the term ensuring pressure-velocity coupling at small time steps.

2.4.2 *Prediction-correction form of the Navier-Stokes equations*

With the system of the reformulated Navier-Stokes equations, Eqs. (2.21), we may show that to achieve high convergence rate of SIMPLE-like algorithms on collocated grids, a proper choice of interpolation methods has to be done. For this, a prediction-correction form of the Navier-Stokes equations is obtained by splitting the velocity and the pressure into two parts, namely a prediction and a correction part as below:

$$\vec{v}^{m+1} = \vec{v}^{*,m+1} + \vec{v}^{c} \qquad (2.22a)$$

$$p^{m+1} = p^{m} + p^{c} \qquad (2.22b)$$

Using Eqs. (2.22) the system of equations, Eqs. (2.21), is solved in two steps. At the first step the momentum equations, Eq. (2.21b), are solved[2] for a prediction velocity $\vec{v}^{*,m+1}$ using the pressure p^m available from the previous subiteration m:

$$\widetilde{A_P}\vec{v}_P^{*,m+1} + \sum_{K=E,W,N,S} A_K\vec{v}_K^{*,m+1} + \vec{\nabla}p_P^m \mathrm{d}V_P = \vec{b}_P^m \qquad (2.23)$$

To obtain an equation for the second step, Eq. (2.23) is subtracted from Eqs. (2.21) and the following system for the velocity and pressure corrections is obtained:

$$\widetilde{A_P}\vec{v}_P^{c} + \sum_{K=E,W,N,S} A_K\vec{v}_K^{c} + \vec{\nabla}p_P^c \mathrm{d}V_P = 0 \qquad (2.24a)$$

$$D_P(\vec{v}^c) + B_P(p^c) = \sum_{k=e,w,n,s} f_k^{*,m+1} \qquad (2.24b)$$

In the continuity equation, Eq. (2.24b), the right hand side was expressed through the mass flux predictions, which are determined by the prediction form of the mass fluxes, Eq. (2.11), as follows:

$$\begin{aligned} f_k^{*,m+1} = {} & \overline{\left[\frac{A^V}{\widetilde{\widetilde{A}}}\right]_k} \widetilde{f}_k^n + (1/\alpha - 1 - \beta)\overline{\left[\frac{A}{\widetilde{\widetilde{A}}}\right]_k} f_k^m \\ & + \overline{\left[\frac{\vec{h}^{*,m+1}}{\widetilde{\widetilde{A}}}\right]_k} \cdot d\vec{S}_k - \overline{\left[\frac{\mathrm{d}V}{\widetilde{\widetilde{A}}}\right]_k} \vec{\nabla}p_k^m \cdot d\vec{S}_k \end{aligned} \qquad (2.25)$$

where the prediction term $\vec{h}_P^{*,m+1}$ is defined as:

$$\vec{h_P}^{*,m+1} = \sum_{K=E,W,N,S} A_K\vec{v}_K^{*,m+1} - \gamma A_P\vec{v}_p^{*,m+1} + \beta A_P\vec{v}_p^m + \vec{S}_P^m \qquad (2.26)$$

[2] It can be noted that Eq. (2.21b) is composed of two equations in 2D (or three equations in 3D) and may be solved for each of the velocity components separately using iterative methods such as Gauss-Seidel, Jacobi etc.

One may verify that the discrete continuity equation Eq. (2.24b) is equivalent to the original continuity equation Eq. (2.21a) To solve the system of equations, Eqs. (2.24), a pressure-correction equation has to be obtained in a way as described in the following section.

2.4.3 *Pressure-correction equation*

It is straightforward to obtain an equation for the pressure correction by expressing the velocity correction from Eq. (2.24a), and then substituting the velocity in Eq. (2.24b). However the resulted matrix of pressure correction equations obtains a rather complex structure. Below, we will use an alternative approach, similar to the SIMPLE and the SIMPLEC algorithms, where the first equation in the system of equations, Eqs. (2.24), is replaced by the following simplified equation:

$$L_P \vec{v}_P^c + \vec{\nabla} p_P^c \mathrm{d}V_P = 0 \tag{2.27}$$

where $L_P = \widetilde{A_P}$ for the SIMPLE [64] algorithm and $L_P = \widetilde{A_P} - A_P$ for the SIMPLEC [86] algorithm.

In spite of the widely spread assumption that the SIMPLE and the SIMPLEC algorithms differ by the L_P only (see e.g. Ferziger and Perić [20] and Wesseling [90]), we will show below that on collocated grids different interpolation methods have to be employed with SIMPLE and SIMPLEC algorithms to ensure a high efficiency of the algorithms. By expressing the velocity correction, $\vec{v}_P^c$, from Eq. (2.27) and then substituting it in Eq. (2.24b), we obtain the following pressure-correction equation:

$$-\sum_{k=e,w,n,s} \overline{\left[\frac{\mathrm{d}V}{\widetilde{\widetilde{A}}}\right]_k} \vec{\nabla} p_k^c \cdot d\vec{S}_k + \underbrace{D_P((\widetilde{\widetilde{A}}^{-1} - L^{-1})\vec{\nabla} p^c \mathrm{d}V)}_{\texttt{Compatibility term}} = \sum_{k=e,w,n,s} f_k^{*,m+1} \tag{2.28}$$

The pressure correction equation, Eq. (2.28), is the final equation, which has to be solved to obtain the pressure-correction, p^c. The equation is different from the widely used pressure-correction equations in literature due to existence of the second term in the equation Eq. (2.28).

It can be noted, that Eq. (2.28) is valid for the standard *PS* interpolation method (see Eq. (2.3)). The second term in Eq. (2.28) is not equal to zero in general. A specific condition when the term equals to zero will be discussed in the next section.

2.4.4 *Compatibility condition*

The second term of the pressure-correction equation, Eq. (2.28), here referred to as compatibility term, is zero if the following condition is fulfilled:

$$L_P^{-1} - \widetilde{\widetilde{A_P}}^{-1} = 0 \tag{2.29}$$

As shown below, fulfilling the condition may enhance the convergence rate of the SIMPLE-like algorithms.

One may consider different interpolation methods with SIMPLE or SIMPLEC algorithm. If Eq. (2.29) is fulfilled for some interpolation with the SIMPLE algorithm than the compatibility term in the corresponding pressure-correction equation becomes zero. Therefore, we will refer such interpolation as interpolation fully compatible with SIMPLE algorithm (or SIMPLE-compatible interpolation). Alternatively, if Eq. (2.29) is fulfilled for the SIMPLEC algorithm than the interpolation is referred to as fully compatible with SIMPLEC algorithm (or SIMPLEC-compatible interpolation). Using the compatibility condition, Eq. (2.29), it can be verified that the PS_E interpolation and the similar interpolations of Choi [11] are SIMPLE-compatible. The original interpolation of Rhie and Chow [65] and lots of its modifications presented in [35, 39, 42, 43, 46, 51, 57, 61, 94], are also SIMPLE-compatible. However, in literature these methods were applied in both SIMPLE and SIMPLEC algorithms, regardless of the fact that the interpolations are not satisfying the compatibility condition, Eq. (2.29), with the SIMPLEC algorithm. To see this, one may consider the PS_E interpolation, which is a second order alternative of the widely spread interpolation of Choi [11] (see Eq. (2.3)). Employing the PS_E interpolation with the SIMPLEC algorithm results in the pressure-correction equation, Eq. (2.28), with the compatibility term not equal to zero. In this case, Eq. (2.28) can be solved using several possible approaches:

1. The first approach is to solve Eq. (2.28) exactly. Nevertheless, this requires using a solver which can handle the matrixes with rather complex structures.
2. The second approach is to treat the compatibility term explicitly and solve Eq. (2.28) iteratively. However, such approach can be computationally expensive.
3. The third approach is the widely used approach (as seen e.g. in Ferziger and Perić [20] and Wesseling [90]), where instead of Eq. (2.28) the following equation is solved:

$$\sum_{k=e,w,n,s} \overline{\left[\frac{\mathrm{d}V}{L}\right]_k} \vec{\nabla} p_k^c \cdot d\vec{S}_k = \sum_{k=e,w,n,s} f_k^{*,m+1} \tag{2.30}$$

which is a typical pressure-correction equation for the SIMPLEC algorithm on collocated grids [20, 39, 61, 90]. Nevertheless, it can be noted that Eq. (2.30) *approximates* the original pressure-correction equation, Eq. (2.28), which may consequently make the convergence rate of the SIMPLEC algorithm slower.

Instead of solving Eq. (2.30), a proper choice of interpolation method for the SIMPLEC algorithm may be used to simplify the corresponding pressure-correction equation. Using the compatibility condition in Eq. (2.29), it can be verified that the PS_C interpolation (see Eq. (2.3)) is SIMPLEC-compatible. For the SIMPLEC algorithm employed with this interpolation, the compatibility term in the corresponding pressure-correction equation becomes inherently zero. Therefore, no approximation of the pressure-correction equation is required, contrary to the approximations needed to obtain the pressure-correction equation, Eq. (2.30).

The summary about the compatibility between the PS interpolations and the SIMPLE-like algorithms is given in Table. 2.1. In the result section below, the SIMPLEC algorithm, which is based on the SIMPLEC-compatible interpolation, is shown to possess a convergence rate up to 35% higher compared to the standard SIMPLEC algorithm.

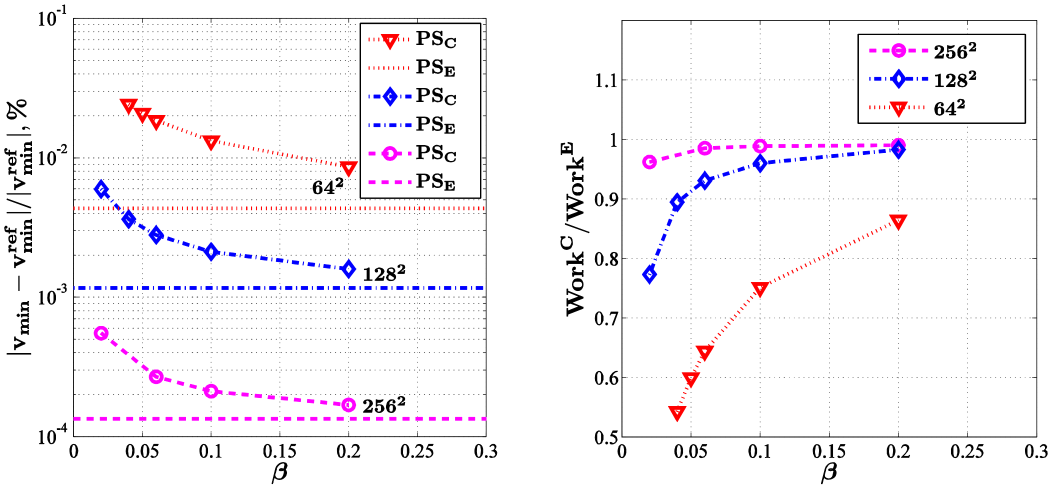
2.5 RESULTS

In this section it is shown that appropriate choice of interpolation method for SIMPLEC algorithm may increase the convergence rate of the algorithm. For this the SIMPLEC-compatible interpolation method of Shen et al. [73] will be compared with the SIMPLE-compatible interpolation of Choi [11]. These two methods, respectively referred to as the PS_C and the PS_E methods (see Table. 2.1), will be tested in steady and unsteady flow computations. First, as the PS_C interpolation depends on some positive parameter β (see Secs. 2.3 and 2.3.1), an optimal value of the parameter will be identified to ensure the robustness of the methods. Second, it will be shown that in unsteady flow computations the SIMPLEC-compatible interpolation enhances the convergence rate of the SIMPLEC algorithm. Two test cases: the lid-driven cavity flows and the Taylor-Green vortex problem will be studied. To simplify notations, different SIMPLEC algorithms will be identified by those interpolation methods, which the algorithms are based on. For instance, the SIMPLEC algorithm with the PS_C interpolation will be simply designated as PS_C.

The convergence criteria in the computations presented below is the residual to be below 10^{-8} for steady computations and 10^{-6} for unsteady computations. For the tests the velocity underrelaxation parameter of $\alpha = 0.9$ is used.

2.5.1 *Lid-driven cavity flow*

Flow in a lid-driven cavity at $Re = 1000$ is computed on a series of non-uniform grids with different resolutions of from 64^2 to 256^2 cells. The computational effort for computing the flow field is measured in number of multigrid cycles needed for solving the pressure-correction equation. The SIMPLEC-compatible PS_C method, is used to identify the optimal value of the β parameter. For this, the PS_C method is compared to the PS_E method, which is independent of β (see Sec. 2.3). A work ratio of the methods is computed as $Work^C/Work^E$, where $Work^C$ and $Work^E$ are the workloads of the PS_C and the PS_E methods, respectively. The work ratio is plotted for various grid resolutions in Fig. 2.2b, where it is seen that the PS_C method with β value of [0.04, 0.06] has up to 40% lower work load. To compare the errors of the methods, the minimum vertical velocity along the horizontal line across the center of the cavity is measured and the errors are computed against the reference solution in Botella and Peyret [4]. As seen in Fig. 2.2a,



(a) Error.

(b) Work load

Figure 2.2: Comparison of relative error and work load for the PS_C and the PS_E methods in a lid-driven cavity flow at Re=1000. $Work^C$ - work of the PS_C method, $Work^E$ - work of the PS_E method.

the SIMPLEC-compatible method gives slightly higher errors than the PS_E method. However, the PS_C method becomes advantageous with respect to both speed and accuracy when it is applied for unsteady flow computations, as will be shown in the following sections.

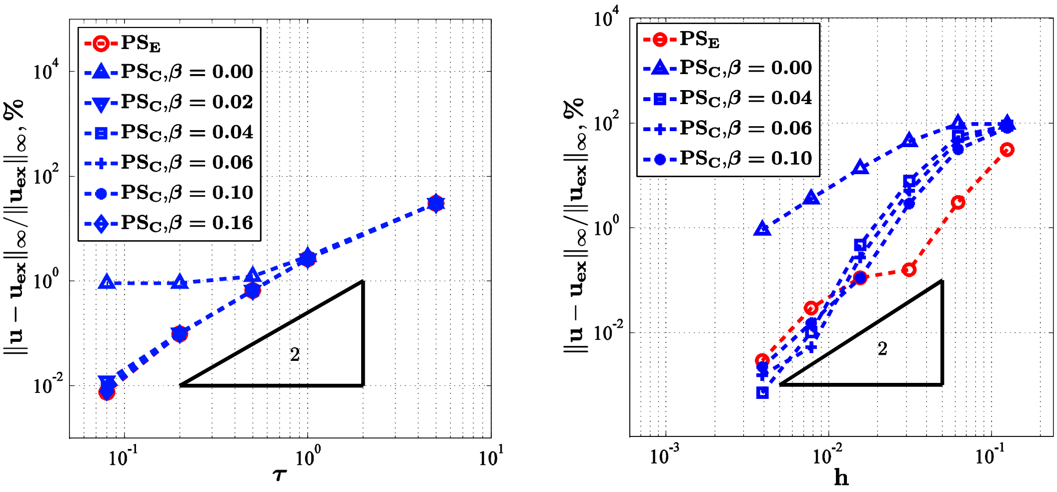
2.5.2 *Taylor-Green vortex*

The Taylor-Green vortex is a test case with an exact solution of the Navier-Stokes equations:

$$
\begin{aligned}
u(x,y,t) &= -sin(\pi x)cos(\pi y)e^{-2\pi^2 t/Re} \\
v(x,y,t) &= cos(\pi x)sin(\pi y)e^{-2\pi^2 t/Re} \\
p(x,y,t) &= 0.25(cos(2\pi x) + sin(2\pi y))e^{-4\pi^2 t/Re}
\end{aligned}
$$

The computational domain consists of a square of $[-\pi, \pi]$ with periodic boundary conditions in both directions. The PS_C and the PS_E methods are used to compute the flow field at Re = 10. The time integration is performed from $t = 0$ to $t = 10$. At time $t = 10$ the total kinetic energy in the domain has dropped to about 2% of the energy in the initial state. Equidistant grids with different resolutions of from 8^2 to 256^2 cells are used and the error is computed against the exact solution at $t = 10$.

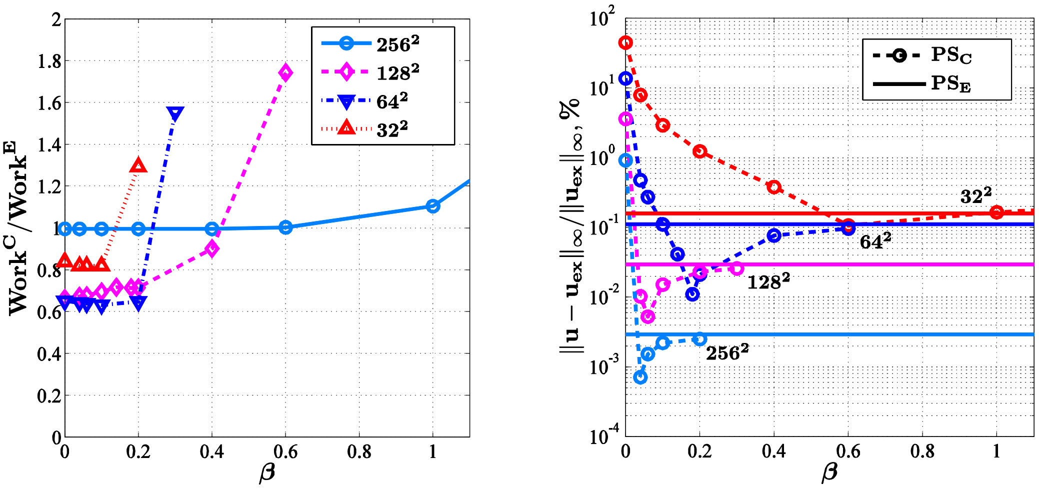
The convergence of the temporal error performed on the grid of 256^2 cells is shown in 2.3a. It is seen that for both PS_C and PS_E methods the second order convergence in time is preserved, except for the PS_C method with $\beta = 0$, where error stagnation appears.



(a) Temporal error.

(b) Spatial error.

Figure 2.3: Convergence of temporal and spatial errors using PS_E and PS_C methods for the Taylor-Green problem.



(a) Work load.

(b) Error.

Figure 2.4: Dependence of work load and relative error on β for the PS_C and the PS_E methods on grids with 32^2, 64^2, 128^2 and 256^2 cells for the Taylor-Green problem. $Work^C$ - work of the PS_C method, $Work^E$ - work of the PS_E method.

The reason of the error stagnation for $\beta = 0$ is that the spatial error contribution becomes *non-negligible*. To check the spatial error, the grid convergence study at a constant and sufficiently small time step $\tau = 0.05$ is plotted in Fig. 2.3b. It is seen that the spatial error in the case of $\beta = 0$ is much higher than that for $\beta > 0$.

The work load is measured using the number of multigrid cycles. The efficiency dependence on the β parameter of the PS_C method is shown in Fig. 2.4a, where the work ratio of the PS_C and the PS_E methods is plotted. It is seen that within the set of $\beta \in [0, 0.1]$, the efficiency is nearly independent of β. Moreover, for nearly all tested grid resolutions the SIMPLEC-compatible method with $\beta \in [0, 0.1]$ has a convergence rate of about 30% higher than the PS_E method, as seen in Fig. 2.4a.

Although the optimum value of the β parameter is not known in advance, for general applications of the PS_C method, a β value of 0.04 should be used. First, for this β the convergence of the PS_C method is preserved at all grid resolutions. Second, as seen in Fig. 2.4a and Fig. 2.4b, this value of β ensures that both speed and accuracy of the PS_C method are higher than those of the PS_E method when an asymptotic range is reached.

2.6 conclusions

In order to increase the convergence rate of SIMPLE-like algorithms on collocated grids, a compatibility condition between the algorithms and the mass flux interpolations has been presented. Interpolation methods existing in literature were classified as either being fully compatible with SIMPLE algorithm or being fully compatible with SIMPLEC algorithm. It was shown, that if the SIMPLE-like algorithms are based on the interpolations fully compatible with the algorithms, a higher convergence rate can be obtained.

Among the existing methods, the PS_C interpolation method of Shen et al. [73] was shown to be the only method fully compatible with the SIMPLEC algorithm. The method was formulated in a parameterized form (PS interpolation), which enabled us to employ the method as either SIMPLE- or SIMPLEC- compatible interpolation (PS_E or PS_C interpolations, respectively).

As the PS_C interpolation depends on an additional β parameter, the optimal value for β was found to be equal to 0.04. Numerical results of unsteady flow computations shown that employing the SIMPLEC algorithm with the SIMPLEC-compatible interpolation results in a convergence rate up to 30% higher compared to the standard SIMPLEC algorithm. Based on the performed computational tests, the recommended parameter values for the PS interpolation are given in Table. 2.2.

Interpolation	Employed in algorithm	Recommended γ and β
PS	SIMPLE	$\gamma = 0, \beta = 0$
	SIMPLEC	$\gamma = 1, \beta = 0.04$

Table 2.2: Recommended γ and β for the PS method employed in SIMPLE or SIMPLEC algorithms.

In spite of the advantages of the PS_C method, when employed in SIMPLEC algorithm, it was shown that it results in solutions, dependent of time step and relaxation parameter at convergence. The influence of the time step dependence on the solution accuracy will be investigated in the next chapter, where two new SIMPLEC-compatible interpolation methods, independent of time step and relaxation parameter, will be presented.

3

NEW INTERPOLATION METHODS ON COLLOCATED GRIDS

3.1 INTRODUCTION

In order to ensure an accurate solution on collocated grids various modifications of the Rhie and Chow momentum interpolation have been presented since the last two decades. For the steady flow computations there is a common practice to employ Majumdar's [51] modification of the Rhie-Chow method, whereas for the unsteady flow computations, interpolation methods of Shen et al. [72] and Choi [11] are employed, as can be seen for example in [1, 40, 41, 54]. The methods of Shen and Choi are known to be time step- and relaxation parameter- dependent at steady state [42]. Nowadays, possible influence of the dependence on the solution accuracy in a typical practical application is still not known.

To overcome the problem with the solution dependence of time step, several interpolation methods were proposed in [13, 63, 94], but the choice between the methods for practical applications is rather uncertain. In the literature, the time step independent methods were mostly used for SIMPLE algorithm and similarly to the standard methods, compatibility between the methods and other types of the SIMPLE algorithm (SIMPLEC, PISO, SIMPLER) was not discussed.

In this chapter, two new interpolation methods, which are independent of time step and relaxation parameter at convergence, are presented. Contrary to other time step-independent methods, the two methods are shown to be fully compatible with the SIMPLEC algorithm. Numerical results of unsteady flow computations, presented in Sec. 3.7, show that the SIMPLEC algorithm with one of the new interpolations ensures convergence rate, which is up to 25% higher than the convergence rate of standard SIMPLEC algorithm. The new interpolation is also shown to result in more accurate solutions at steady state compared to other interpolations.

This chapter is organized as follows. First, two new interpolation methods are introduced and compared with some popular existing interpolations. One of the new methods is shown to be the most accurate, compared to other new method and the methods in literature. Second, the new interpolation, is shown to fulfill compatibility with SIMPLE-like algorithms, thus enhancing the convergence rate of the algorithms. Third, the pressure-correction equation is obtained and several methods to solve the equation are discussed. Fourth, results of various computational experiments of steady and unsteady flows are presented with recommendations for practical use.

3.2 TWO NEW INTERPOLATIONS INDEPENDENT OF TIME STEP AT CONVERGENCE

Most of the existing interpolation methods, which result in solutions independent of the time step τ and the relaxation parameter α, were constructed based on the interpolation method of Choi [11] or Shen et al. [72].

Recall, that in the previous chapter the PS_E interpolation of Shen et al. [72] was shown to be SIMPLE-compatible, whereas the PS_C interpolation of Shen et al. [73] was shown to be the only SIMPLEC-compatible interpolation in literature. By employing the SIMPLEC algorithm with the SIMPLEC-compatible interpolation higher convergence rate was obtained compared with standard SIMPLEC algorithm.

Two new interpolation methods will be presented below, which preserve this advantage of the PS_C interpolation, but contrary to the PS_C interpolation are independent of time step and relaxation parameter at convergence. To develop these methods, the unified form of the PS_E and the PS_C interpolations (see the PS method in Chapter. 2) will be used.

3.2.1 *The PI^0 interpolation*

The first new interpolation method is obtained from Eq. (2.11) by using the following approximation:

$$\overline{\left[\frac{\phi}{\tilde{\tilde{A}}}\right]}_k \Leftarrow \overline{\left[\frac{\phi}{A^V}\right]}_k \frac{1}{1.5 + (1/\alpha - \gamma)\overline{[A/A^V]}_k}$$

where ϕ is a representative variable. By using the approximation in the expression of the mass flux in Eq. (2.11), a new method is obtained:

$$\begin{aligned} f_k^{m+1} = \chi_k\Big(& \tilde{f}_k - \frac{\tau}{\rho}\vec{\nabla} p_k^{m+1} \cdot d\vec{S}_k + \overline{\left[\frac{\vec{h}^{m+1}}{A^V}\right]}_k \cdot d\vec{S}_k \\ & + (1/\alpha - 1 - \beta)\overline{\left[\frac{A}{A^V}\right]}_k f_k^m\Big) \end{aligned} \tag{3.1}$$

where $\overline{[\ \ ]}_k$ denotes linear interpolation from cell centers to cell face k and χ_k is defined as:

$$\chi_k = \frac{1}{1.5 + (1/\alpha - \gamma)\overline{[A/A^V]}_k} \tag{3.2}$$

The interpolation method, Eq. (3.1), is independent of time step and relaxation parameter at convergence (see Sec. for the proof). It is formulated in the parameterized form using constant parameters γ and β, therefore the interpolation, Eq. (3.1), will be further referred to as Parameterized and Independent of both time step and relaxation parameter (PI^0) method. The superscript 0 in the method notation is used due to the reasons explained in the Sec. 3.3. Before the PI^0 method is analyzed, a second new interpolation method is introduced below.

3.2.2 *The PI^∞ interpolation*

The second new interpolation method is obtained from Eq. (2.11) by using the following approximation:

$$\overline{\left[\frac{\phi}{\tilde{\tilde{A}}}\right]}_k \Leftarrow \overline{\left[\frac{\phi}{A}\right]}_k \frac{1}{1.5\overline{[A^V/A]}_k + (1/\alpha - \gamma)}$$

where ϕ is a representative variable. Using the approximation above, Eq. (2.11) is transformed into the equation for the second new interpolation method:

$$\begin{aligned} f_k^{m+1} = \psi_k(&\overline{\left[\frac{A^V}{A}\right]}_k \tilde{f}_k - \overline{\left[\frac{\mathrm{d}V}{A}\right]}_k \vec{\nabla} p_k^{m+1} \cdot d\vec{S}_k + \overline{\left[\frac{\vec{h}^{m+1}}{A}\right]}_k \cdot d\vec{S}_k \\ &+ (1/\alpha - 1 - \beta) f_k^m) \end{aligned} \tag{3.3}$$

where ψ_k is defined as:

$$\psi_k = \frac{1}{1.5\overline{[A^V/A]}_k + (1/\alpha - \gamma)} \tag{3.4}$$

The interpolation method, Eq. (3.3), is also independent of time step and relaxation parameter at convergence. It will be further referred to as Parameterized and Independent of both time step and relaxation parameter (PI^∞) method, where the superscript ∞ in the method notation is used due to the reasons explained in Sec. 3.3.

3.3 COMPARISON WITH STANDARD INTERPOLATIONS

Similarly to the PS interpolation of Shen (see Sec. 2.3), each of the PI^0 and the PI^∞ interpolations above has two forms, namely the form with $\gamma = 0, \beta = 0$ and the form with $\gamma = 1, \beta > 0$ as described below:

1. For $\gamma = 0$ and $\beta = 0$ the PI methods (PI^0 and PI^∞) are similar to two interpolation methods of Pascau (see PICTURETWO and PICTURE in [63], respectively), but contrarily to Pascau's methods, the PI methods are based on second order backward difference in time. In its turn, the PICTURE method of Pascau is similar to Cubero's interpolation in [13].

2. For $\gamma = 1$ and $\beta > 0$ the PI methods are new interpolation methods. As will be seen later, contrary to other time step independent methods, the PI methods in this form are fully compatible with SIMPLEC algorithm. According to the results presented later, the PI^0 method with $\gamma = 1$ and $\beta = 0.04$ may enhance the convergence rate of the SIMPLEC algorithm up to 25%, compared to the time step independent methods existing in literature.

According to Sec. 3.5, the PI methods in the first and the second form above are fully compatible with SIMPLE and SIMPLEC algorithms, respectively. Therefore, the PI methods in the first and the second forms will be further referred to as PI_E and PI_C methods, respectively (see Table. 3.1).

Forms of PI^0 and PI^∞ interpolations	Corresponding γ and β	Compatible with	Similar interpolations in literature (if exist)
PI_E^0	$\gamma = 0, \beta = 0$	SIMPLE	Pascau [63] (PICTURETWO)
PI_E^∞			Pascau [63] (PICTURE), Cubero and Fueyo [13]
PI_C^0	$\gamma = 1, \beta > 0$	SIMPLEC	
PI_C^∞			

Table 3.1: Notations for the SIMPLE- and SIMPLEC- compatible forms of PI^0 and PI^∞ interpolations.

The steady state solutions of the discrete continuity equation, Eq. (2.7), based on the mass fluxes interpolated using either PI^0 or PI^∞ methods, are independent of both time step τ and velocity relaxation parameter α. Indeed, taking the condition that the flow reached the steady state, the flux for the PI^0 method, Eq. (3.1), becomes:

$$f_k = \frac{\overline{\left[\frac{\vec{h}}{\rho dV}\right]_k} \cdot d\vec{S}_k - \frac{1}{\rho}\vec{\nabla} p_k \cdot d\vec{S}_k}{(1-\gamma+\beta)\overline{\left[\frac{A}{\rho dV}\right]_k}} \tag{3.5}$$

whereas the flux for the PI^∞ method, Eq. (3.3), is equal to:

$$f_k = \frac{\overline{\left[\frac{\vec{h}}{A}\right]_k} \cdot d\vec{S}_k - \overline{\left[\frac{dV}{A}\right]_k} \vec{\nabla} p_k \cdot d\vec{S}_k}{(1-\gamma+\beta)} \tag{3.6}$$

As seen, in Eqs. (3.5) and (3.6) and the fluxes are independent of both τ and α.

By comparing the PI methods, Eqs. (3.1) and (3.3), with the standard PS method, Eq. (2.11), it can be proved that:

- The fluxes of the PI^0 method at steady state equal to the fluxes of the PS method at steady state, if the PS method is employed with an infinitely small time step, $\tau \rightarrow 0$.
- The fluxes of the PI^∞ method at steady state equal to the fluxes of the PS method at steady state, if the PS method is employed with an infinitely large time step, $\tau \rightarrow \infty$.

A strict proof of the two statements above is presented in Appendix A. Using the two statements and knowing that discretization errors drop down at decreasing time steps, we can make the following conclusion:

- The PI^0 method is expected to give a higher accuracy than both the PS and the PI^∞ methods.

In support of the statements above, the results in Sec. 3.7 will show the superiority of the PI^0 method in comparison to both the PI^∞ and the PS methods.

3.4 PRESSURE-CORRECTION EQUATIONS

Similarly to the derivation of the pressure-correction equation for the PS interpolation (see Sec. 2.4) the pressure-correction equations for the new PI^0 and PI^∞ interpolations can be derived. Omitting the intermediate steps, these final equations are given below.

For SIMPLE-like algorithms using the PI^0 interpolation, the unified form of the pressure-correction equation is:

$$-\sum_{k=e,w,n,s} \chi_k \overline{\left[\frac{\tau}{\rho}\right]}_k \vec{\nabla} p_k^c \cdot d\vec{S}_k + \underbrace{D_p((\widetilde{\widetilde{A}}^{-1} - L^{-1})\vec{\nabla} p^c \mathrm{d}V)}_{\texttt{Compatibility term}} = \sum_{k=e,w,n,s} f_k^{*,m+1} \tag{3.7}$$

where the term $L_P = \widetilde{A_P}$ for the SIMPLE algorithm, $L_P = \widetilde{A_P} - A_P$ for the SIMPLEC algorithms, and the operator D_p computes the divergence of any vector field $\vec{\phi}$ as:

$$D_p(\vec{\phi}) = \sum_{k=e,w,n,s} \rho\overline{[\vec{\phi}]}_k \cdot d\vec{S}_k$$

In Eq. (3.7), χ_k is defined by Eq. (3.2) and the prediction mass fluxes $f_k^{*,m+1}$ are defined by Eq. (3.1).

The pressure-correction equation for SIMPLE-like algorithms using the PI^∞ interpolation is:

$$- \sum_{k=e,w,n,s} \psi_k \overline{\left[\frac{\mathrm{d}V}{A}\right]}_k \vec{\nabla} p_k^c \cdot d\vec{S}_k + \underbrace{D_p((\widetilde{\widetilde{A}}^{-1} - L^{-1})\vec{\nabla} p^c \mathrm{d}V)}_{\texttt{Compatibility term}} = \sum_{k=e,w,n,s} f_k^{*,m+1} \qquad (3.8)$$

where ψ_k is defined by Eq. (3.4) and the mass flux predictions $f_k^{*,m+1}$ are determined by Eq. (3.3).

The pressure correction equations, Eq. (3.7) and Eq. (3.8) are different from the widely used pressure-correction equations in literature due to the existence of the second term in the equations. This term referred as compatibility term can be naturally nullified if a proper choice of interpolation method is done, as will be discussed in the following section.

3.5 COMPATIBILITY CONDITION

Instead of solving the pressure-correction equations, Eqs. (3.7) and (3.8), a proper choice of interpolation method may be used first to ensure that the compatibility term in the equation is equal to zero. Similarly to Sec. 2.4.4 this can be ensured by fulfilling the following condition:

$$L_p^{-1} - \widetilde{\widetilde{A}}_P^{-1} = 0 \qquad (3.9)$$

Recall, that if Eq. (3.9) is fulfilled for SIMPLE algorithm, the interpolation is referred to as SIMPLE-compatible. Alternatively, if Eq. (3.9) is fulfilled for SIMPLEC algorithm, the interpolation is referred to as SIMPLEC-compatible.

Using the compatibility condition above, it can be checked that the PI_E^0, PI_E^∞ interpolations and the similar interpolations of Pascau [63] and Cubero and Fueyo [13] are SIMPLE-compatible. Alternatively, the PI_C^0 and the PI_C^∞ interpolations are SIMPLEC-compatible. The summary about compatibility of the PI^0 and PI^∞ interpolations with the SIMPLE and the SIMPLEC algorithms is given in Table. 3.1.

The advantage of fulfilling the compatibility with SIMPLE-like algorithms can be seen by taking the example of the SIMPLEC algorithm using PI^0 interpolation, which is discussed in the following section.

3.6 SIMPLEC ALGORITHM BASED ON PI^0 INTERPOLATION

When a SIMPLEC algorithm is employed with a SIMPLEC-compatible interpolation, no additional approximation of the pressure-correction equation is required. This is shown below in the context of two different algorithms: 1) the SIMPLEC algorithm with the PI_C^0 interpolation and 2) the SIMPLEC algorithm with the PI_E^0 interpolation.

3.6.1 PI_C^0-based SIMPLEC algorithm

If the SIMPLEC algorithm is employed with the PI_C^0 interpolation, the compatibility term in the corresponding pressure-correction equation, Eq. (3.7), is inherently zero. Indeed, for the SIMPLEC algorithm $L_P = (1/\alpha - 1)A_P + 1.5A_P^V$, whereas $\widetilde{\widetilde{A}}_P = (1/\alpha - 1)A_P + 1.5A_P^V$ due to definition of the PI_C^0 interpolation. Therefore, the compatibility condition is fulfilled, i.e. the PI_C^0 interpolation is SIMPLEC-compatible and the the pressure-correction equation, Eq. (3.7), simplifies to:

$$- \sum_{k=e,w,n,s} \frac{\tau/\rho}{1.5 + (1/\alpha - 1)\overline{[A/A^V]}_k} \vec{\nabla} p_k^c \cdot d\vec{S}_k = \sum_{k=e,w,n,s} f_k^{*,m+1} \tag{3.10}$$

where $f_k^{*,m+1}$ are determined by the PI_C^0 method. It can be noted that if the system of equations is composed of the pressure-correction equation Eq. (3.10), the system has a simple matrix with of 5 diagonals in 2D or 7 diagonals in 3D, and may be solved by any standard Poisson solver.

3.6.2 PI_E^0-based SIMPLEC algorithm

Alternatively, if the SIMPLEC algorithm is employed with the PI_E^0 interpolation, the compatibility term in the corresponding pressure-correction equation is not zero. Indeed, due to definition of the PI_E^0 method $\widetilde{\widetilde{A}}_P = A_P/\alpha + 1.5A_P^V$, therefore $L_P \neq \widetilde{\widetilde{A}}_P$ and the corresponding pressure-correction equation has the following form:

$$\begin{aligned} &- \sum_{k=e,w,n,s} \frac{\tau/\rho}{1.5 + (1/\alpha)\overline{[A/A^V]}_k} \vec{\nabla} p_k^c \cdot d\vec{S}_k \\ &+ \sum_{k=e,w,n,s} \overline{\left[\vec{\nabla} p^c \mathrm{d}V (\widetilde{\widetilde{A}}^{-1} - L^{-1})\right]}_k \cdot \ d\vec{S}_k = \sum_{k=e,w,n,s} f_k^{*,m+1} \end{aligned} \tag{3.11}$$

where $f_k^{*,m+1}$ is determined by the PI_E^0 method. If the system of equations is composed using the pressure-correction equation Eq. (3.11), the matrix of the system is not simple. The system can be simplified, if the PC equation, Eq. (3.11), is approximated further using the following assumption:

$$- \sum_{k=e,w,n,s} \frac{\tau/\rho \vec{\nabla} p_k^c}{1.5 + R^\alpha \overline{[A/A^V]}_k} \cdot d\vec{S}_k \approx - \sum_{k=e,w,n,s} \overline{\left[\frac{\vec{\nabla} p^c dV}{1.5A^V + R^\alpha A}\right]}_k \cdot d\vec{S}_k \tag{3.12}$$

where an auxiliary variable R^α was used for brevity: $R^\alpha = 1/\alpha$ or $R^\alpha = 1/\alpha - 1$. The simplification above assumes that the interpolated gradient of the pressure-correction is nearly identical to the directly computed gradient (with A/A^V term being interpolated only). Using the approximation above, Eq. (3.11) is approximated to:

$$- \sum_{k=e,w,n,s} \frac{\tau/\rho}{1.5 + (1/\alpha - 1)\overline{[A/A^V]}_k} \vec{\nabla} p_k^c \cdot d\vec{S}_k = \sum_{k=e,w,n,s} f_k^{*,m+1} \tag{3.13}$$

The pressure-correction equation, Eq. (3.13), is the final pressure-correction equation for the SIMPLEC algorithm based on the PI_E^0 interpolation. The equation has a simple structure and can be solved using any standard Poisson solver, similarly to the pressure-correction equation, Eq. (3.10), for the PI_C^0 interpolation[1]. But contrarily to Eq. (3.10) the equation for the PI_E^0 method *approximates* the original pressure-correction equation, Eq. (3.7), which may decrease the convergence rate of the SIMPLEC algorithm. Indeed, the computational results in Sec. 3.7 show that the PI_C^0-based SIMPLEC algorithm results in a convergence rate 10-25% higher than that of the PI_E^0-based SIMPLEC algorithm.

3.7 RESULTS

In this section it is shown that an appropriate choice of interpolation method for the SIMPLEC algorithm may not only enhance the convergence rate, but also increase the solution accuracy of the algorithm. For this, the SIMPLEC algorithms will be used with the PS, PI^0 and PI^∞ interpolation methods. By taking an example of the SIMPLEC-compatible PI_C^0 interpolation method, it will be shown first that in comparison to other time step independent method of Pascau [63], the PI_C^0 method results in a higher efficiency of the SIMPLEC algorithm. Second, the class of the PI^0 interpolation methods (the PI_C^0 and the PI_E^0), will be shown to ensure the highest solution accuracy in comparison to the popular interpolation of Choi [11] and the PICTURE interpolation of Pascau [63]. Finally, for the standard interpolations of Choi [11] and Shen et al. [73] the magnitude of the time step dependence at convergence will be estimated in a typical turbulent flow computations.

Four test cases: roll-up of a shear layer vortex, flows around circular cylinder, flows in lid-driven cavity and turbulent flow around NACA64618 airfoil will be studied. To simplify notations, different SIMPLEC algorithms will be identified by those interpolation methods, which the algorithms are based on. For instance, the SIMPLEC algorithm with the PI_C^0 interpolation will be simply designated as PI_C^0.

The convergence criteria in the computations presented below, is set to be the residual below 10^{-8} for steady flow computations and 10^{-6} for unsteady flow computations. For the tests the velocity underrelaxation parameter $\alpha = 0.8$ is used.

3.7.1 *Roll-up of a shear layer vortex*

In this section a shear layer roll-up test case [58] is computed at $Re = 100$. The new SIMPLEC-compatible PI_C^0 method with $\beta = 0.04$ is used to enhance the convergence rate of the SIMPLEC algorithm. To compare with the method, the PI_E^0 interpolation is employed as a representative of Pascau's interpolation [63], which is non-fully compatible with SIMPLEC algorithm (see Table. 3.1 in Sec. 3.2.2).

[1] The equations, Eq. (3.10) and Eq. (3.13), are not identical as the mass flux predictions in the equations are defined differently.

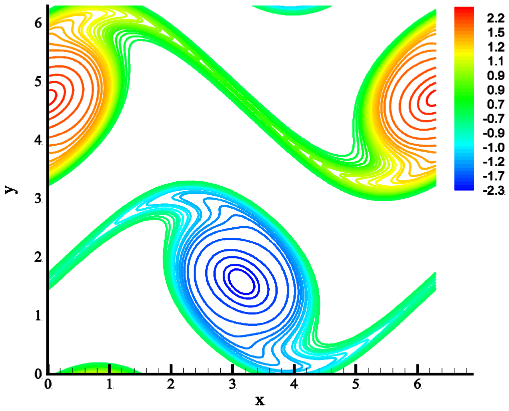
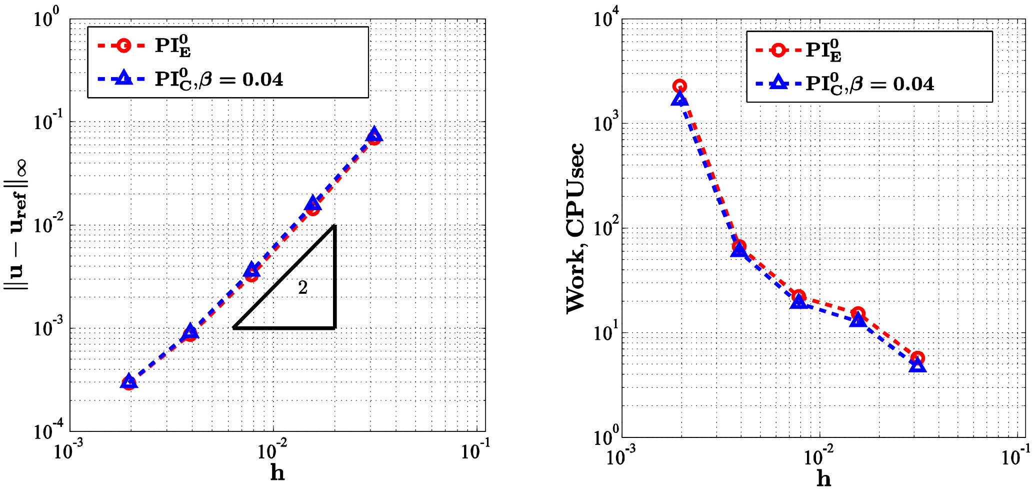


Figure 3.1: Vorticity field of a shear layer roll-up at $Re = 100$ and $t = 8$.

The flow is initialized in the domain of $[0, \pi]$ with periodic boundary conditions,

$$u = \begin{cases} tanh(\frac{y-\pi/2}{\delta}), y \leq \pi & v = \epsilon sin(x) \\ tanh(\frac{3\pi/2-y}{\delta}), y > \pi & \delta = \pi/15, \epsilon = 0.05 \end{cases}$$



(a) Error.

(b) Work load

Figure 3.2: Comparison of spatial error convergence and work loads measured in CPU seconds for the PI_C^0 and the PI_E^0 methods in the shear layer roll-up test case.

An example of the flow field at $Re = 100$ at a dimensionless time $t = 8$ is shown in Fig. 3.1. The solutions are computed at t = 4 and the errors are measured in comparison to the reference solution, which is obtained using the PI_C^0 method on a fine grid of 512^2 cells and a small time step $\tau = 0.005$. The computations are performed on a sequence of the successively coarsened grids. The largest CFL number measured on the finest grid equals to 0.2. For each of the coarse grids, the error is computed by interpolating the reference solution on the coarse grid and subtracting the solution from that grid. As seen from Figs. 3.2a and Fig. 3.2b, second order spatial accuracy is obtained and the error tolerances of the new method and the PI_E^0 method are nearly identical.

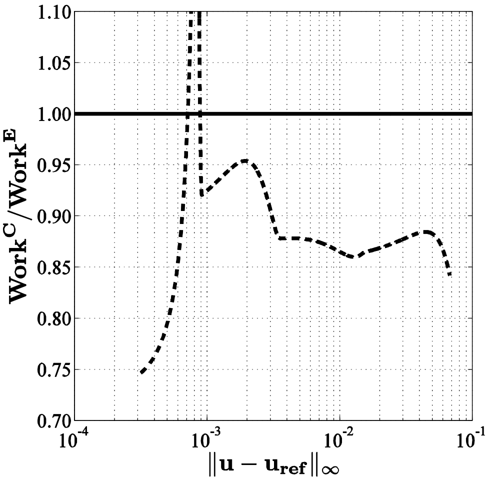
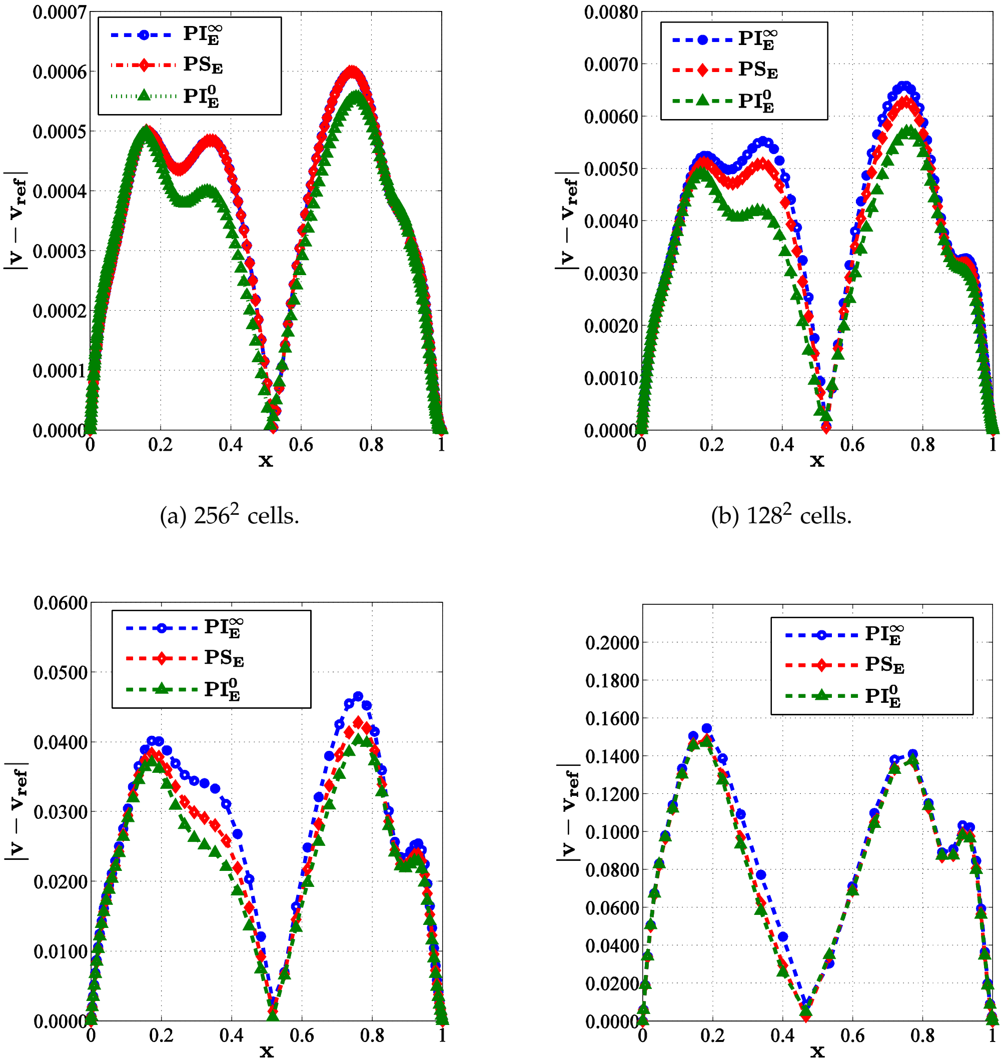


Figure 3.3: Dependence of work load on accuracy for the shear layer roll-up test case. In the ratio $Work^C/Work^E$, the terms $Work^C$ and $Work^E$ denote the work loads of the PI_C^0 method and PI_E^0 method, respectively.

The corresponding work load of the methods, measured in CPU seconds, is plotted in Fig. 3.2b. The efficiencies of the methods can be compared if the work loads are compared for the same accuracy levels. Therefore, the work load for each of the method is found first as a function of the error. Then the ratio of the work loads of the PI_C^0 and PI_E^0 methods is plotted in Fig. 3.3. As seen in the figure, the work load of the PI_C^0 method is up to 25% lower compared to the work load of the PI_E^0 method. These results correlate well with the discussions in Sec. 2.4.4 and Sec. 3.5, where the SIMPLEC-compatible interpolations (such as PS_C, PI_C^0 and PI_C^∞) were supposed to enhance the convergence rate of the SIMPLEC algorithm.

3.7.2 *Lid-dirven cavity flows*

In this section the flow in a lid driven cavity is computed at $Re = 1000$ on a sequence of non-uniform grids with various resolutions. The test case is used to verify that the PI^0 method is the most accurate in comparison to the PS and PI^∞ methods. The methods are used in SIMPLE-compatible forms, i.e. as the PS_E , PI_E^0 and PI_E^∞ methods, which correspond, respectively, to Choi's interpolation and the two interpolations of Pascau (see Table. 3.1 in Sec. 3.2.2). The convergence properties of the methods are compared by the errors of the vertical velocity along the horizontal line across the center of the cavity. The reference solution is computed on a grid with 512^2 cells. The steady solutions are computed using the constant time step $\tau = 1$ on successively coarsened grids. For each of the coarse grids, the error is computed by interpolating the reference solution on the coarse grid and subtracting the solution from that grid. From the error distribution along the horizontal line, shown in Fig. 3.4, it is seen that the PI_E^0 method is the most accurate method compared to both the PI_E^∞ and the PS_E methods, which correlates well with the conclusion made in Sec. 3.3.



(a) 256^2 cells.

(b) 128^2 cells.

(c) 64^2 cells.

(d) 32^2 cells.

Figure 3.4: Velocity error distribution along the horizontal line at $y = 0.5$ across the lid-driven cavity on grids with different resolution.

3.7.3 *Steady flow around a circular cylinder*

As the next test case the steady flow around a circular cylinder is computed at $Re = 40$. In the test case the PS_C interpolation of Shen et al. [73] and the two new PI_C^0 and PI_C^∞ methods (see Table. 3.1 in Sec. 3.2.2) are used with $\beta = 0.04$. The drag error dependence

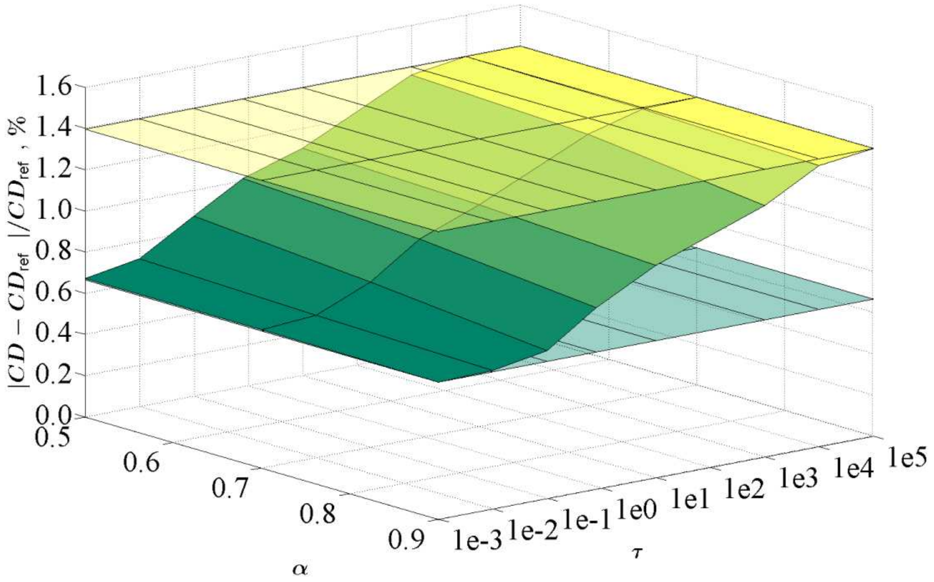
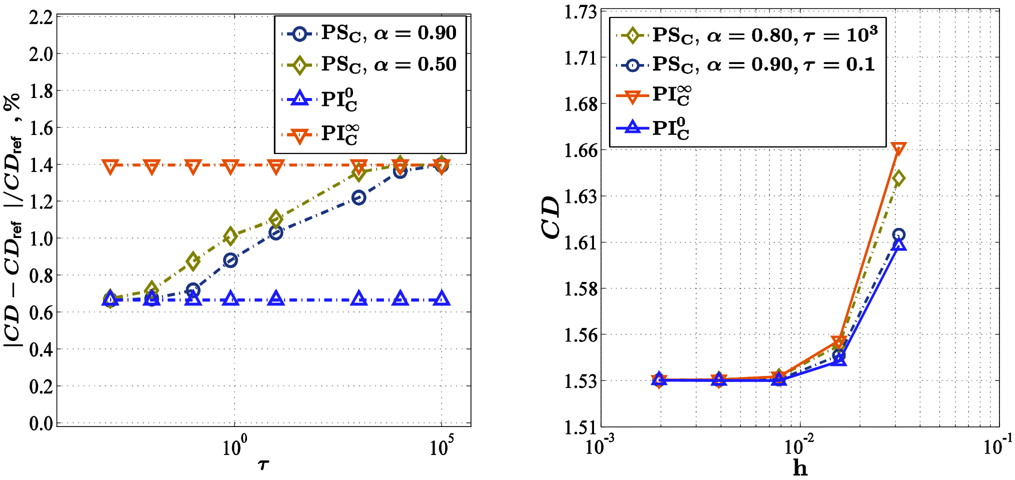


Figure 3.5: Dependence of relative error on both time step τ and relaxation parameter α for flows around a circular cylinder using three interpolation methods: PS_C (curved surface), PI_C^0 (bottom plane) and PI_C^∞ (top plane).

of τ and α on a grid with 64 x 32 cells is plotted in Fig. 3.5, where the drag value of 1.532 computed on the grid with 512 x 256 cells is used as the reference solution. It is seen that the solution of the PS_C method asymptotically tends to the solutions of the PI_C^0 and the PI_C^∞ methods for infinitely small and large time steps, respectively. This behavior is in exact agreement with the analytical conclusions made in Sec. 3.3. From Figs. 3.5 and 3.6a it is also seen that the dependence of the standard method on the relaxation parameter α is less significant than that on the time step τ. The error dependence of τ and α on a grid with 64 x 32 cells plotted in Fig. 3.6a and the grid convergence plotted in Fig. 3.6b show that the dependence on time step of the standard method may contribute up to 50% of the error. Contrary to the PS_C method the PI_C^0 and the PI_C^∞ methods result in solutions, which are independent of τ and α, but as seen in Fig. 3.6b, the accuracy of the PI_C^0 method is higher than of the other methods. These results are in good agreement with the conclusion made in Sec. 3.3 that the PI^0 method results in the highest accuracy compared to both the PS and the PI^∞ methods.

3.7.4 *Turbulent flow around a NACA 64618 airfoil*

The turbulent flow around NACA 64618 airfoil at zero angle of attack is computed at $Re = 1.6 \cdot 10^6$ using SIMPLEC algorithm. The grid of O-type is used with boundaries placed in a distance of 20 chords from the airfoil. Two standard interpolation methods, the PS_E and the PS_C, are used as the representatives of Choi's [11] and Shen et al.'s [73] interpolation methods, resulting to the solutions, dependent of time step at convergence.

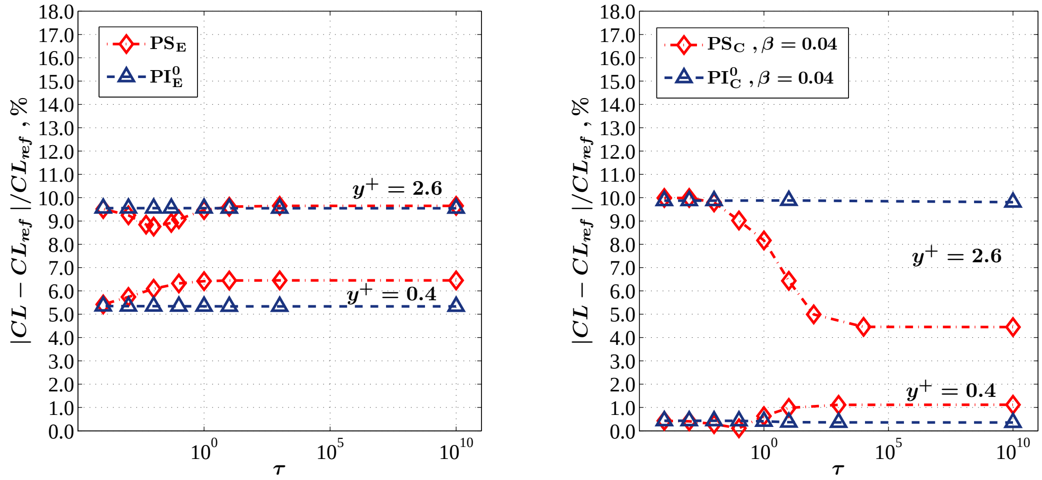


(a) Dependence of relative error on time step τ for flows around a circular cylinder using three interpolation methods: PS_C (curved lines), PI_C^0 (bottom line) and PI_C^∞ (top line).

(b) Dependence of drag coefficient (CD) on grid resolution for flows around a circular cylinder at Re = 40.

Figure 3.6: Solution dependence on time step and relaxation parameter.

To compare with the standard methods, the time step independent methods, the PI_E^0 and the PI_C^0, are used.



(a) PS_E^0 and PI_E^0 methods.

(b) PS_C^0 and PI_C^0 methods.

Figure 3.7: Dependence of relative errors of the standard PS methods in the turbulent flow computations around a NACA 64618 airfoil on grids with 64 x 32 and 128 x 64 cells.

The $k - \omega$ SST turbulence model is used on two relatively coarse grids with 64 x 32 and 128 x 64 cells. For the two grids the maximum y^+ at points one cell away from the airfoil equals to 2.6 and 0.4, respectively. To measure the time step dependence of the standard methods, the lift coefficient is compared against an experimental value in [9], which is equal to 0.44.

As seen from Figs. 3.7a and 3.7b, contrary to the PI^0 method, the solutions of the PS methods are dependent of time step at convergence. For the PS methods the change of the error due to the time step dependence is about 1-2% on the grid with $y^+ = 0.4$, whereas for the grid with $y^+ = 2.6$ the error variations may achieve up to 5.5%, as seen from Fig. 3.7b.

It should be noted, that a typical grid set-up for the turbulent flow computations is based on a grid with 128 cells in the normal direction. Results of computations on a grid with 256 x 128 cells using the standard and the PI^0 methods are shown in Fig. 3.8. It is seen from the figure that on such fine grid, the time step dependence of the error contributes about 0.5% into the total error.

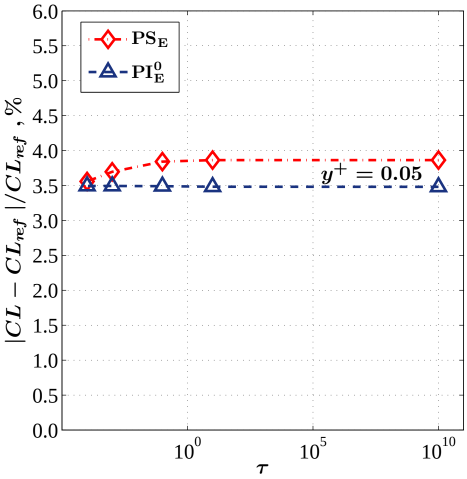


Figure 3.8: Dependence of relative error on time step in the flow computations around NACA 64618 airfoil on the grid with 256 x 128 cells.

The solution, obtained using the PI^0_E method on the grid with 256 x 128 cells is seen to be about 4% different from the experimental value. The solution is less accurate than the solution obtained on the coarser grid using the PI^0_C method with $\beta = 0.04$. This is explained by the fact that for the steady state problems, the PI^0_C method with $\beta = 0.04$ may result in superconvergence similarly to the PS_C method with $\beta = 0.04$ as was shown in Chap. 2.

3.8 CONCLUSIONS

Two new interpolation methods, independent of time step and relaxation parameter at convergence, were presented. The methods were formulated in a parameterized form, which allows employing the methods in either SIMPLE- or SIMPLEC- compatible forms. It was shown that the SIMPLEC algorithm based on the new SIMPLEC-compatible PI^0_C

interpolation is about 10-25% more efficient than the SIMPLEC algorithm based on Pascau's [63] interpolation.

It was shown that the appropriate choice of interpolation methods for the SIMPLEC algorithm may not only enhance the convergence rate, but also increase the solution accuracy of the algorithm. The PI^0 methods (i.e. the PI^0_E and the PI^0_C methods) were shown to result in the most accurate solutions compared to the popular interpolation of Choi [11] and the PICTURE interpolation of Pascau [63].

For the standard interpolation of Shen et al. [73] analysis of the solution dependence of time step and relaxation parameter was presented. Numerical experiments shown that for the method the solution dependence of the relaxation parameter is much less than the dependence of time step.

Magnitude of the solution dependence on the time step for the popular interpolation of Choi [11] and interpolation of Shen et al. [73] was also estimated. Using a typical turbulent test case, it was shown that the magnitude of the time step dependence may achieve 5% of the solution value. For the test case the PI^0 interpolations were shown to result in accurate solutions, which are independent of time step.

It was demonstrated that on coarse grids, the standard interpolations result in much higher non consistent solution behavior than on fine grids. The inconsistency may become crucial in 3D computations, where employing the fine grids becomes computationally demanding.

Based on the analysis and the computational tests presented in this chapter, the PI^0 interpolation method is recommended for practical use. By setting appropriate parameters γ and β, the PI^0 method can be employed as either fully compatible with SIMPLE algorithm or fully compatible with SIMPLEC algorithm. The recommended values of the parameters γ and β for the method are given in Table. 3.2.

Interpolation	Employed in algorithm	Recommended γ and β
PI^0	SIMPLE	$\gamma = 0, \beta = 0$
	SIMPLEC	$\gamma = 1, \beta = 0.04$

Table 3.2: Recommended γ and β for the PI^0 method employed in SIMPLE or SIMPLEC algorithms.

Part II

SIMPLE-LIKE ALGORITHMS ON COLLOCATED GRIDS WITH NONCONFORMAL BLOCK INTERFACES

An extension of 2$^{\text{nd}}$ order finite-volume method of EllipSys2D code to grids with nonconformal blocks is presented. To link velocity and pressure, the Navier-Stokes equations are solved using SIMPLE-like algorithms. Discretization of pressure-correction equation, ensuring simplicity and efficiency of the algorithm, is discussed. The finite volume method is parallellized ensuring fast performance on grids with arbitrary block-to-block connections. Computational result show, that the method results in solution as accurate as on conformal grids. To enable solution of the Poisson pressure-correction equation the multigrid method of the cell-centered/block-structured EllipSys2D code is extended to the block-structured grids with nonconformal interfaces. An Optimized Schwarz (OS) domain decomposition method using Robin boundary conditions for IBLU relaxation scheme is presented in the frame of the multigrid method on discontinuous grids. The Robin parameter, depending on grid geometry and grid discontinuity at block interfaces, is shown to fulfill conditions, required to ensure high convergence of the OS method. Results from computations on grids with nonconformal blocks show, that the modified multigrid method preserves high efficiency of the original multigrid method of EllipSys solver.

4

FINITE VOLUME METHOD AND SIMPLE-LIKE ALGORITHM ON NONCONFORMAL GRID INTERFACES

4.1 INTRODUCTION

In this chapter, the momentum equations are discretized on nonconformal grids using second-order difference scheme in time and second order central difference scheme in space, except for convective fluxes that are discretized with the QUICK upwinding scheme. Approximation of the convective flux at discontinuous interfaces is based on auxiliary node interpolation, whereas the diffusion flux is discretized based on the projection of the velocity gradient on the normal direction as will be described below. When the control-volume fluxes are discretized the deferred correction method is applied.

The discretisation is described with emphasize on the accuracy, constraints on the convergence rate, conservative properties and simplicity of the method of implementation. The Pressure-Correction equation is solved using multigrid method and to preserve its efficiency on grids with discontinuous interfaces the discretisation is based on a mass conservative approach.

To ensure the performance of the computational scheme on parallel computers, the scheme is parallellized using Message Passing Interface (MPI) and the communication tables, which are built by a specially designed preprocessor. The parallel algorithm is able to treat numbers of interfaces with multiple block-to-block connectivity.

The chapter is composed as follows. First, finite volume method is applied to the Navier-Stokes equations on grids with nonconformal blocks and the notations used to discretize the equations are introduced. Second, convection, diffusion and pressure-gradient terms of the Navier-Stokes equations are discretized in a conservative form. Third, pressure-correction equation is discussed with the emphasis on the accuracy, implementation costs and constraints on the convergence rate when applied on discontinuous grids. Finally, parallel algorithm and the preprocessor are described. Verification of the finite volume method is presented in the next chapter together with the tests of the modifgied multigrid method.

4.2 FINITE-VOLUME METHOD

Using finite-volume method and applying the divergence theorem to the integral form of the Navier-Stoke equations at some control volume adjacent to a nonconformal interface, (see Fig. 4.1a), the following equations are obtained:

$$\int\limits_{\partial\Omega\cup\partial\Omega_{\mathrm{I}}} \rho\vec{v}\cdot\vec{n}\mathrm{d}S = 0 \tag{4.1a}$$

$$\frac{\partial}{\partial t}\int\limits_{\Omega}\rho\vec{v}dV + \int\limits_{\partial\Omega\cup\partial\Omega_{\mathrm{I}}} \vec{F}^{CV} = \int\limits_{\partial\Omega\cup\partial\Omega_{\mathrm{I}}} \vec{F}^{DF} + \int\limits_{\partial\Omega\cup\partial\Omega_{\mathrm{I}}} \vec{F}^{P} + \vec{S} \tag{4.1b}$$

where Ω is the size of the control volume, surface integrals are taken over the set of conformal $\partial\Omega$ and non conformal $\partial\Omega_{\mathrm{I}}$ faces. The finite volume fluxes: convective, diffusive and pressure gradient fluxes are defined, respectively, as below:

$$\vec{F}^{CV} = \begin{pmatrix} uC \\ vC \end{pmatrix}, \quad \vec{F}^{DF} = \begin{pmatrix} \mu\vec{\nabla}u\cdot\vec{n}\mathrm{d}S \\ \mu\vec{\nabla}v\cdot\vec{n}\mathrm{d}S \end{pmatrix} \quad \vec{F}^{P} = \begin{pmatrix} -p\vec{n}\cdot\vec{e}_1\mathrm{d}S \\ -p\vec{n}\cdot\vec{e}_2\mathrm{d}S \end{pmatrix} \tag{4.2}$$

where C is the mass flux defined as $C = \rho\vec{v}\cdot\vec{n}\mathrm{d}S$, $\mathrm{d}S$ is the surface length and the vector $\vec{n}$ is the unit vector normal to the surface. In expressions of the diffusion fluxes, F^{DF}, vectors $\vec{e}_1$ and $\vec{e}_2$ are the basis vectors. Finally, the source term $\vec{S}$ represents body forces and other source terms if exist.

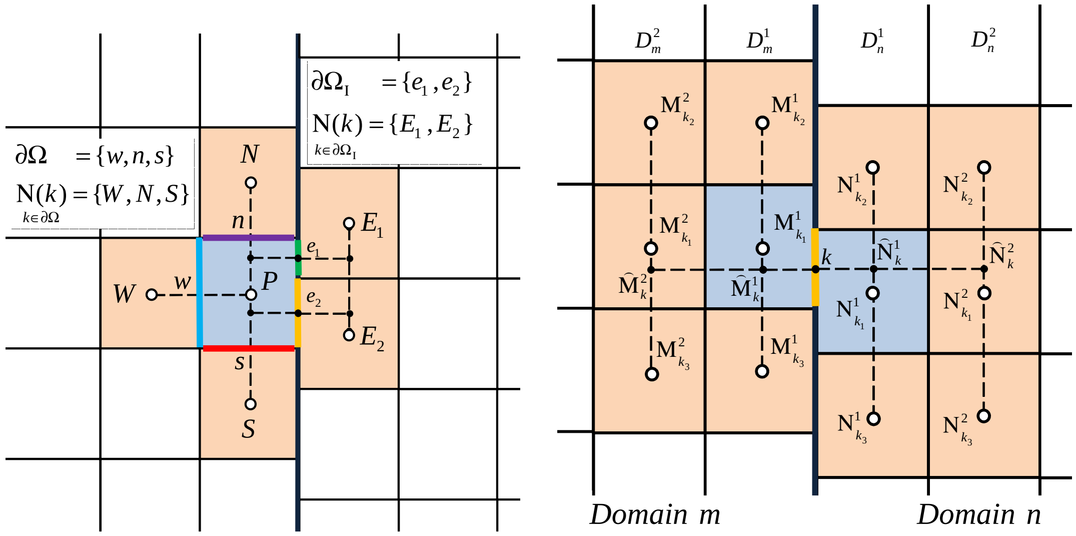
As seen from above, to compute the finite volume fluxes either values or gradients of the flow field are needed to be known at control volume faces. The conservative finite volume discretisation on grids with the discontinuous interfaces can be preserved, if for any control volume the flux through the corresponding interface cell face is expressed as the sum of fluxes through the segments which the face consists of. To discretize Eqs. (4.1a) and (4.1b), several notations are introduced as below.

4.3 NOTATIONS

Assume that two domains D_m and D_n are connected with some interface I_{mn} such that the cells at the interface do not necessarily match as seen in Fig. 4.1b. The two grid layers nearest to the interface I_{mn} in the domain D_m are called D_m^1 and D_m^2 respectively. Applying the same convention for the domain D_n, the grid layers nearest to the interface are called D_n^1 and D_n^2. As the grid is not continuous at the interface, then for some cell belonging to the first grid layer D_m^1 there may in general be more than one neighbor cells at the other side of the interface, belonging to the grid layer D_n^1 of the domain D_n.

For a cell belonging to domain D_m connected to a cell belonging to domain D_n at the discontinuous interface, we consider the common segment of the interface called k as in Fig. 4.1b. To define the flow field values at the segment k, auxiliary nodes are introduced as shown below.

If we consider the domain D_m, then the two auxiliary nodes $\widehat{M_k^1}$, $\widehat{M_k^2}$ may be introduced along the line normal to the segment k and placed inside the grid layers D_m^1 and D_m^2 such that the value at the node $\widehat{M_k^1}$ may be linearly interpolated based on the values at the nearest cell centers $M_{k_1}^1$, $M_{k_2}^1$, $M_{k_3}^1$, belonging to the layer D_m^1, and the value at the node $\widehat{M_k^2}$ may be linearly interpolated based on the values at the nearest cell centers $M_{k_1}^2$, $M_{k_2}^2$,



(a) Example of control volume at discontinuous interface. Faces w, n, s belong to set of conformal faces $\partial\Omega$, the segments e_1 and e_2 belong to set of interface segments $\partial\Omega_I$.

(b) Auxiliary nodes $\widehat{M^1_k}$, $\widehat{M^2_k}$, $\widehat{N^1_k}$ and $\widehat{N^2_k}$ are defined along the line normal to the segment k inside grid layers D^1_m, D^2_m, D^1_n and D^2_n respectively. For example the value at the node $\widehat{M^1_k}$ may be linearly interpolated based on values at nodes $M^1_{k_1}$, $M^1_{k_2}$, $M^1_{k_3}$ of the grid layer D^1_m.

Figure 4.1: General concept of conservative discretisation on nonconformal grid.

$M^2_{k_3}$, belonging to the layer D^2_m. In the definitions $M^j_{k_i}$ belongs to grid layer D^j_m where the superscript $j = \overline{1,2}$ depends on normal to interface direction and changes from 1 for first grid layer to 2 for the second grid layer, while the subscript $i = \overline{1,3}$ depends on tangential direction and changes from 1 for the first nearest cell center to 3 for the third nearest cell center within the same grid layer as shown in Fig. 4.1b.

Similar notations are used in the domain D_n, thus at all the auxiliary nodes the corresponding values are linearly interpolated as:

$$\begin{aligned} \varphi_{\widehat{M^j_k}} &= \alpha_{k_1^j}\varphi_{M^j_{k_1}} + \alpha_{k_2^j}\varphi_{M^j_{k_2}} + \alpha_{k_3^j}\varphi_{M^j_{k_3}}, \text{ for } j = \overline{1,2} \text{ in the domain } D_m \\ \varphi_{\widehat{N^j_k}} &= \beta_{k_1^j}\varphi_{N^j_{k_1}} + \beta_{k_2^j}\varphi_{N^j_{k_2}} + \beta_{k_3^j}\varphi_{N^j_{k_3}}, \text{ for } j = \overline{1,2} \text{ in the domain } D_n \end{aligned} \tag{4.3}$$

where the coefficients $\alpha_{k_i^j}$,$\beta_{k_i^j}$ are the coefficients of polynomial interpolation at side D_m and D_n respectively.

4.4 DISCRETE MOMENTUM EQUATIONS

To preserve conservation properties of the finite volume method not only inside of the computational sub-domains, but also at the nonconformal interfaces, the finite-volume fluxes in Eqs. (4.2) have to be defined in a conservative manner. General ideas about the discretization of Navier-Stokes equations on block-structured grids with discontinuous interfaces can be found in the work of Lilek [47, 48] and Seidl[71], whereas in this section a detailed discretization of momentum equations on grids with discontinuous interfaces is presented. The discrete pressure-correction equation as the most important part for the SIMPLE-like algorithm will be discussed in Sec. 4.5.

System of the momentum equations in Eq. (4.1b) is composed of two equations in 2D (and three equations in 3D). Without loss of generality the first equation of the system (i.e. the equation for the u velocity component) is considered below. For this, first components of the finite volume fluxes, Eq. (4.2), are considered and denoted simply as F^{CV}, F^{DF}, F^{P} for the convection, diffusion and the pressure terms, respectively.

4.4.1 *Convective terms*

Consider a Control volume (CV) adjacent to discontinuous interface as shown in Fig. 4.1a. Recall, that the CV has two sets of the control volume faces: the set of conformal faces ($\partial\Omega$) and the set of the nonconformal faces ($\partial\Omega_{\mathrm{I}}$), corresponding to the interface. The convective finite volume flux F^{CV} can be expressed as the sum of the convective fluxes through each of the segments, which the face consists of, as below:

$$\int\limits_{\partial\Omega_{\mathrm{I}}} F^{CV} = \int\limits_{\partial\Omega_{\mathrm{I}}} uC = \sum_{k\in\partial\Omega_{\mathrm{I}}} u_k C_k = \sum_{k\in\partial\Omega_{\mathrm{I}}} F_k^{CV} \tag{4.4}$$

Then, using the second order upwind scheme each of the convective fluxes F_k^{CV} can be expressed as:

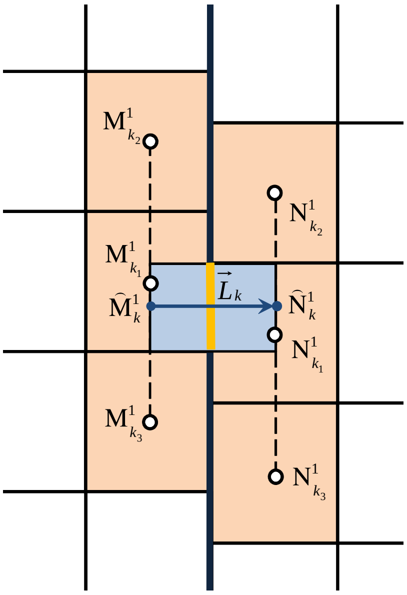


Figure 4.2: Discretisation of diffusion fluxes.

$$\begin{aligned} F_k^{CV} = \quad & u_{\widehat{M_k^1}} C_k^+ + u_{\widehat{N_k^1}} C_k^- + \frac{1}{2} C_k^+ (u_{\widehat{M_k^1}} - u_{\widehat{M_k^2}}) \\ & + \frac{1}{2} C_k^- (u_{\widehat{N_k^1}} - u_{\widehat{N_k^2}}) \end{aligned} \tag{4.5}$$

where $d\vec{S}_k$ is the normal vector with the length dS_k and the fluxes C_k^+, C_k^- equal to $(C_k + |C_k|)/2$ and $(C_k - |C_k|)/2$, respectively. Then employing the polynomial interpolation Eq. (4.3) at the auxiliary nodes $\widehat{M_k^1}$, $\widehat{M_k^2}$, $\widehat{N_k^1}$ and $\widehat{N_k^2}$, the convective flux at the segment k becomes:

$$F_k^{CV} = \sum_{i=\overline{1,3}} u_{M^1_{k_i}} \alpha^1_{k_i} C_k^+ + \sum_{i=\overline{1,3}} u_{N^1_{k_i}} \beta^1_{k_i} C_k^- + \frac{1}{2} C_k^+ (u_{\widehat{M_k^1}} - u_{\widehat{M_k^2}}) + \frac{1}{2} C_k^- (u_{\widehat{N_k^1}} - u_{\widehat{N_k^2}}) \tag{4.6}$$

Expression above can be linearized, such that the mass flux C is taken from the former time step, whereas for the velocity vector $\vec{v}$ the following approach is used.

The convection flux $\vec{F}_k^{CV}$ depends on the velocity values at the auxiliary nodes $\widehat{M_k^1}$, $\widehat{M_k^2}$, $\widehat{N_k^1}$ and $\widehat{N_k^2}$, which are not directly available and treated as follows. The last two of the four terms in the Eq. (4.6) depend on the values at twelve cell center nodes and are treated explicitly. Therefore, the two terms will contribute to the source terms $\vec{S}$. The first and the second terms in the expression are the sum over three nodal values. And only those values which are stored in the cells having the segment k in common are treated implicitly, which results in a compact computational stencil. As seen in Fig. 4.1b these cells are $M^1_{k_1}$, $N^1_{k_1}$ and the expression for the convective flux at the segment k becomes:

$$\begin{aligned} F^{CV}_k = {} & u^{n+1}_{M^1_{k_1}} \alpha^1_{k_1} C^+_k + u^{n+1}_{N^1_{k_1}} \beta^1_{k_1} C^-_k + \sum_{i=\overline{2,3}} u^n_{M^1_{k_i}} \alpha^1_{k_i} C^+_k + \sum_{i=\overline{2,3}} u^n_{N^1_{k_i}} \beta^1_{k_i} C^-_k \\ & + \frac{1}{2} C^+_k (u^n_{\widehat{M^1_k}} - u^n_{\widehat{M^2_k}}) + \frac{1}{2} C^-_k (u^n_{\widehat{N^1_k}} - u^n_{\widehat{N^2_k}}) \end{aligned} \tag{4.7}$$

where the first two terms are treated semi-implicitly and the remaining terms - explicitly. Thus, for the considered control volume at the interface the contributions from the segment k to the diagonal and non-diagonal matrix coefficients of the momentum equations A_P and A_k become as follows:

$$A_P \Leftarrow \alpha^1_{k_1} C^+_k \text{ and } A_k \Leftarrow \beta^1_{k_1} C^-_k \tag{4.8}$$

while the contribution from the convective flux through the segment k to the source term is:

$$S^u \Leftarrow - \sum_{i=\overline{2,3}} u^n_{M^1_{k_i}} \alpha^1_{k_i} C^+_k - \sum_{i=\overline{2,3}} u^n_{N^1_{k_i}} \beta^1_{k_i} C^-_k - \frac{1}{2} C^+_k (u^n_{\widehat{M^1_k}} - u^n_{\widehat{M^2_k}}) - \frac{1}{2} C^-_k (u^n_{\widehat{N^1_k}} - u^n_{\widehat{N^2_k}})$$

Finally, it can be noted that extension from the second order upwind approximation to a higher order scheme such as the QUICK upwinding scheme may be obtained analogously using the similar notations as was used above.

4.4.2 *Diffusion terms*

On the nonconformal interface the diffusion flux is computed based on a scalar product of the gradient of the velocity component and the normal vector at segment k of the interface:

$$\int\limits_{\partial\Omega_I} F^{DF} = \int_{\partial\Omega_I} \mu \vec{\nabla} u \vec{n} dS = \sum_{k \in \partial\Omega_I} \mu_k \vec{\nabla} u_k \vec{n}_k dS_k = \sum_{k \in \partial\Omega_I} F^{DF}_k \tag{4.9}$$

Let's $\vec{L}_k$ be a vector connecting two auxiliary nodes $\widehat{N^1_k}$ and $\widehat{M^1_k}$. Then the diffusive finite volume flux F^{DF}_k at the segment k of the discontinuous interface may be discretized as:

$$F^{DF}_k = \mu_k \frac{\vec{\nabla} u_k \vec{L}_k}{|d\vec{S}_k||d\vec{L}_k|} dS^2_k \tag{4.10}$$

where the product $|\mathrm{d}\vec{S}_k||\mathrm{d}\vec{L}_k|$ represents the area of the staggered control volume at the segment k shown in Fig. 4.2.

Noting that $\vec{\nabla} u_k \vec{L}_k = u_{\widehat{N^1_k}} - u_{\widehat{M^1_k}}$ (see Fig. 4.2) the flux above is transformed as:

$$F_k^{DF} = \mu_k \frac{\beta^1_{k_1} u^{n+1}_{N^1_{k_1}} - \alpha^1_{k_1} u^{n+1}_{M^1_{k_1}}}{|\mathrm{d}\vec{S}_k||\mathrm{d}\vec{L}_k|} \mathrm{d}S^2_k + \mu_k \frac{\sum\limits_{i=\overline{2,3}} \beta^1_{k_i} u^n_{N^1_{k_i}} - \sum\limits_{i=\overline{2,3}} \alpha^1_{k_i} u^n_{M^1_{k_i}}}{|\mathrm{d}\vec{S}_k||\mathrm{d}\vec{L}_k|} \mathrm{d}S^2_k \tag{4.11}$$

where only those terms which are stored in the cells having the segment k in common are treated implicitly, whereas other terms are treated explicitly.

Thus, for the considered control volume at the interface the contributions from the segment k to the diagonal and non-diagonal matrix coefficients of the momentum equations A_P and A_k become as follows:

$$A_P \Leftarrow \mu_k \frac{\alpha^1_{k_1} \mathrm{d}S^2_k}{|\mathrm{d}\vec{S}_k||\mathrm{d}\vec{L}_k|} \text{ and } A_k \Leftarrow -\mu_k \frac{\beta^1_{k_1} \mathrm{d}S^2_k}{|\mathrm{d}\vec{S}_k||\mathrm{d}\vec{L}_k|} \tag{4.12}$$

whereas the contribution to the source term from the diffusion flux through the segment k is:

$$S^u \Leftarrow -\mu_k \frac{\sum\limits_{i=\overline{2,3}} \beta^1_{k_i} u^n_{N^1_{k_i}} - \sum\limits_{i=\overline{2,3}} \alpha^1_{k_i} u^n_{M^1_{k_i}}}{|\mathrm{d}\vec{S}_k||\mathrm{d}\vec{L}_k|} \mathrm{d}S^2_k$$

4.4.3 *Pressure force terms*

Finally, the pressure force terms at the side of the nonconformal interface is approximated as:

$$\int\limits_{\partial\Omega_{\mathrm{I}}} F^P = -\int_{\partial\Omega_{\mathrm{I}}} p\vec{\hat{n}} \cdot \vec{\hat{e}}_1 \mathrm{d}S = -\sum_{k\in\partial\Omega_{\mathrm{I}}} \overline{[p]}_k \vec{\hat{n}}_k \cdot \vec{\hat{e}}_1 \mathrm{d}S_k = \sum_{k\in\partial\Omega_{\mathrm{I}}} F^P_k \tag{4.13}$$

where to obtain the pressure value at the center of the segment k interpolation operator $\overline{[\quad]}_k$ was employed. This operator, contrary to the operator on conformal grids, computes the segment center values using two consecutive interpolations:

- First, polynomial interpolation from the twelve cell centers to the auxiliary nodes $\widehat{M^1_k}$, $\widehat{M^2_k}$, $\widehat{N^1_k}$ and $\widehat{N^2_k}$ as seen in Fig. 4.1b.
- Second, polynomial interpolation from the auxiliary nodes to the segment center k.

This finishes approximations of the finite volume fluxes in x direction. Approximation of the finite volume fluxes in y direction can be obtained in the similar way. The resulted system of discrete momentum equations will be formulated in the next section.

4.4.4 *Discretized momentum equations*

In a control volume adjacent to the discontinuous interface the resulting discretized two-dimensional momentum equations have the following form:

$$\widehat{A_P} u_P^{n+1} + \sum_{k \in \partial\Omega \cup \partial\Omega_{\mathrm{I}}} A_{N(k)} u_{N(k)}^{n+1} + \sum_{k \in \partial\Omega \cup \partial\Omega_{\mathrm{I}}} \overline{[p^{n+1}]}_k \vec{n}_k \cdot \vec{e}_1 \mathrm{d}S_k = S_P^{x,n} + A_P^V \widetilde{u}^n, \quad (4.14\mathrm{a})$$

$$\widehat{A_P} v_P^{n+1} + \sum_{k \in \partial\Omega \cup \partial\Omega_{\mathrm{I}}} A_{N(k)} v_{N(k)}^{n+1} + \sum_{k \in \partial\Omega \cup \partial\Omega_{\mathrm{I}}} \overline{[p^{n+1}]}_k \vec{n}_k \cdot \vec{e}_2 \mathrm{d}S_k = S_P^{y,n} + A_P^V \widetilde{v}^n \quad (4.14\mathrm{b})$$

where $\widehat{A_P}$ is composed of spatial and time terms: $\widehat{A_P} = A_P + 1.5A_P^V$, whereas the auxiliary velocity vector $\widetilde{\vec{v}^n}$ denotes the velocities at the former time steps:

$$\widetilde{\vec{v}^n} = \begin{pmatrix} \widetilde{u}^n \\ \widetilde{v}^n \end{pmatrix} = \begin{pmatrix} 2u^n - 0.5u^{n-1} \\ 2v^n - 0.5v^{n-1} \end{pmatrix} \quad (4.15)$$

Contrary to the momentum equations on conformal grids (see Eq. (2.1) in Sec. 2.2.1) the sum in the system of equations, Eqs. (4.14), is taken over the set of conformal cell faces $\partial\Omega$ and the set of control volume edges $\partial\Omega_{\mathrm{I}}$ lying on the nonconformal interface I. In Eqs. (4.14) and in the equations below the term $N(k)$ denotes the neighbor cell center, such that for $k \in \partial\Omega$, $N(k)$ corresponds to the centers of the neighbor cells in the same domain, whereas for $k \in \partial\Omega_{\mathrm{I}}$, $N(k)$ corresponds to the neighbor cell centers in the neighbor domain (see Fig. 4.1a).

Applying velocity underrelaxation in Eqs. (4.14) results in:

$$\widetilde{A_P} u_P^{m+1} + \sum_{k \in \partial\Omega \cup \partial\Omega_{\mathrm{I}}} A_{N(k)} u_{N(k)}^{m+1} + \sum_{k \in \partial\Omega \cup \partial\Omega_{\mathrm{I}}} \overline{[p^{m+1}]}_k \vec{n}_k \cdot \vec{e}_1 \mathrm{d}S_k = b_P^{x,m}, \quad (4.16)$$

$$\widetilde{A_P} v_P^{m+1} + \sum_{k \in \partial\Omega \cup \partial\Omega_{\mathrm{I}}} A_{N(k)} v_{N(k)}^{m+1} + \sum_{k \in \partial\Omega \cup \partial\Omega_{\mathrm{I}}} \overline{[p^{m+1}]}_k \vec{n}_k \cdot \vec{e}_2 \mathrm{d}S_k = b_P^{y,m} \quad (4.17)$$

where the superscripts m and n are the subiteration- and time step- counters, respectively. The diagonal term of the resulting matrix of the momentum equations, $\widetilde{A_P}$, contains the underrelaxation parameter as in Eq. (2.6), whereas vector $\vec{b}_P^m = (b_P^{x,m}, b_P^{y,m})^T$ contains all explicit terms and is defined as in Eq. (2.5).

The contributions of the interface cell faces, namely $A_{N(k)}$, where $k \in \partial\Omega_{\mathrm{I}}$, make the matrix of the system of momentum equations, Eq. (4.16), irregular. To solve the system, the Block-Jacoby/Gauss-Seidl solver originally implemented in EllipSys flow solver on conformal grids is modified according to Ferziger and Perić [20, sec. 8.6.5], where ILU-type solver on nonconformal grids was considered. The modifications involve nullifying the interface contributions in the iteration matrix and then adding the contributions into the solution during the residual update. These modifications of the solver are inherent to the domain decomposition method and preserve implicit treatment of the interface. As will be seen in Chapter. 5, the convergence rate on nonconformal grids similar to the convergence rate on the regular conformal grids is ensured.

4.5 DISCRETE CONTINUITY EQUATION

Solution of the Pressure Correction (PC) equation is one of the main issues of SIMPLE-like algorithms on collocated grids. An approximation of the PC equation on grids with nonconformal interfaces is described below.

Similarly to the discrete continuity equation on conformal grids, the discrete continuity equation on nonconformal grids essentially represents the fact that the sum of mass fluxes through the control volume faces (see Fig. 4.1a) equals zero:

$$\sum_{k \in \partial\Omega \cup \partial\Omega_{\mathrm{I}}} f_k = 0 \tag{4.18}$$

where the subscript k corresponds to the position of control volume face, including the set of interface segments $\partial\Omega_{\mathrm{I}}$. As the mass fluxes are not typically available at the edges of the control volume, a common approach is to apply the momentum interpolation methods similar to the methods discussed in Sec. 2.3 and Sec. 3.

The interpolation at the discontinuous interface is not as straightforward as on conformal grids. Regardless the specific choice of the momentum interpolation method there are two general approaches to obtain the fluxes at the interface:

- Conservative
- Nonconservative

The conservative approach is based on mass flux interpolation at each segment of the control volume edge, which is lying on the interface. The advantage of this approach is that it is conservative and ensures strong connectivity between the domains separated by the interface.

The nonconservative approach is based on so-called ghost or virtual cells. In this approach the cell center values of the neighbor domain are interpolated to the ghost cell center using polynomial interpolation. The advantage of the approach is that it is simple and relatively easy to implement. Nevertheless, the strong coupling between domains is violated, as the approach is not conservative:

- First, the approach results in the sum of mass fluxes along one side of the interface becomes not equal to the sum of the fluxes from the other side. As the incompressible fluid is considered, the resulting mass flux defect may significantly deteriorate the convergence rate of the incompressible solver.
- Second, to obtain the value at some ghost cell center, the nearest cell center values have to be used for interpolation. As the overlap of the ghost cell with the neighbor domain is small (contrary to e.g. the overset grid method, where the the fringe cells are surrounded by the numbers of cells in the neighbor domain, see e.g. Zahle [95]), high accuracy of the interpolation may not be guaranteed.

- Third, using the ghost cells the pressure velocity coupling may not be easily ensured at the interface. This may potentially result in pressure wiggles, especially when highly nonconformal grids (with high grid mismatch ratio at the interface) are used.

Contrary to the nonconservative approach the conservative approach is free from these drawbacks and ensures strong coupling between the domains. Nevertheless, when conservative scheme is employed the system of pressure-correction equations has matrix which is sparse and irregular: neither the number of elements per row nor the bandwidth is constant. This requires careful change of the Multigrid solver as described in Chap.5. In the following sections, the *PS* momentum interpolation technique (see Sec. 2.3) will be used to couple velocity and pressure in the whole domain, including the nonconformal interface.

4.5.1 *Mass flux interpolation*

Recall that in collocated grid-based SIMPLE-like algorithms the pressure is introduced in the continuity equation through the momentum equations. The solution of the momentum equations is stored in cell centers and is computed based on the pressure values available from the previous subiteration. As the solution of the momentum equations is not divergence free, to ensure the continuity of the flow field the solution of the momentum equations is expressed at cell faces in the expressions of the mass fluxes. Then the pressure-correction equation is solved to find the correction mass fluxes, which allows to correct the pressure and the velocity to satisfy the continuity equation. But as the flow field after that does not satisfy the momentum equations the procedure repeats again until convergence.

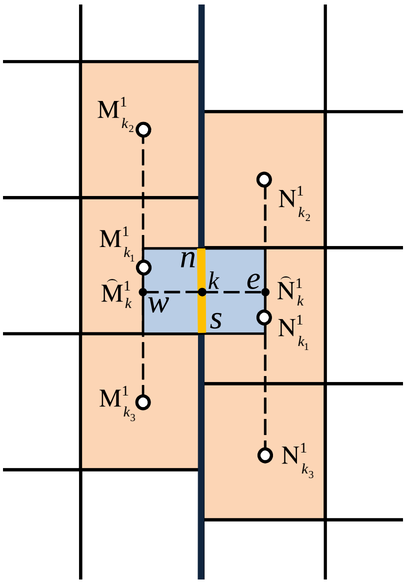


Figure 4.3: Staggered CV at segment *k*.

To preserve pressure-velocity coupling the pressure gradient in the expression of cell face velocity is computed directly, not through velocity interpolation [65]. Here the interpolation of Shen et al. [73] is employed in the parameterized form (see the *PS* interpolation in Sec. 2.3). In Sec. 2.3 this interpolation was described on regular grids, whereas in this section the PS_C interpolation is presented on grids with nonconformal blocks.

As the solution of momentum equations is stored in cell centers, then to define the mass fluxes at cell faces or cell face segments at rhe discontinuous interface the interpolation in necessary.

Similarly to conformal grids, the velocity at the center of the control volumes adjacent to nonconformal interface is:

$$\vec{v}_P^{m+1} = \frac{A_P^V}{\widetilde{\widetilde{A}}_P}\widetilde{\vec{v}}_P^{\,n} + (1/\alpha - 1 - \beta)\frac{A_P}{\widetilde{\widetilde{A}}_P}\vec{v}_P^{m} + \frac{\vec{h}_P^{m+1}}{\widetilde{\widetilde{A}}_P} - \frac{\mathrm{d}V_P}{\widetilde{\widetilde{A}}_P}\vec{\nabla} p_P^{m+1} \tag{4.19}$$

where $\widetilde{\widetilde{A}}_P$ and $\widetilde{\vec{v}}^{\,n}$ are defined in Eq. (2.9) and Eq. (4.15), repsectively, and the auxiliary vector $\vec{h}_P$ is defined as:

$$\vec{h}_P^{m+1} = \sum_{k\in\partial\Omega\cup\partial\Omega_{\mathrm{I}}} A_{N(k)}\vec{v}_{N(k)}^{m+1} - \gamma A_P \vec{v}_p^{m+1} + \beta A_P \vec{v}_p^{m} + \vec{S}_P^{m} \tag{4.20}$$

which contains the sum over the sets of both conformal $\partial\Omega$ and nonconformal $\partial\Omega_{\mathrm{I}}$ faces. Interpolation to conformal faces was discussed in Sec. 2.3, whereas here the *PS* interpolation is used to obtain the mass flux expression at some segment k of the nonconformal interface:

$$\begin{aligned} f_k^{m+1} = {} & \overline{\left[\frac{A^V}{\widetilde{\widetilde{A}}}\right]}_k \widetilde{f}_k^{\,n} + (1/\alpha - 1 - \beta)\overline{\left[\frac{A}{\widetilde{\widetilde{A}}}\right]}_k f_k^{m} + \\ & \overline{\left[\frac{\vec{h}^{m+1}}{\widetilde{\widetilde{A}}}\right]}_k \cdot d\vec{S}_k - \overline{\left[\frac{\mathrm{d}V}{\widetilde{\widetilde{A}}}\right]}_k \vec{\nabla} p_k^{m+1} \cdot d\vec{S}_k \end{aligned} \tag{4.21}$$

The mass flux above is defined at the face segment k. Regardless of the fact, that it has the same form as the expression for conformal grid, it differs from the expression on conformal grids in the two aspects:

- First, the operator $\overline{[\quad]}_k$ computes the segment center value using consecutive polynomial interpolations from four neighbor cell centers in the first domain and from four cell centers in second domain, see e.g. Fig. 4.1b.
- Second, the pressure force term in Eq. (4.21) is approximated as the pressure difference in the direction normal to the segment k (see Fig 4.3) as below:

$$-\vec{\nabla} p_k^{m+1} \cdot \mathrm{d}\vec{S}_k \overline{\left[\frac{\mathrm{d}V}{\widetilde{\widetilde{A}}_P}\right]}_k = a_k\left(p_{\widehat{N}_k^1}^{m+1} - p_{\widehat{M}_k^1}^{m+1}\right) \tag{4.22}$$

where the pressure coefficient a_k is defined as:

$$a_k = -\frac{\mathrm{d}S_k}{\mathrm{d}L_k}\overline{\left[\frac{\mathrm{d}V}{\widetilde{\widetilde{A}}_P}\right]}_k \tag{4.23}$$

where $\mathrm{d}L_k$ is the distance between the auxiliary nodes $\widehat{N}_k^1$ and $\widehat{M}_k^1$ and zero skewness of the cells at the interface is assumed.

As the values at the auxiliary nodes are not directly available they are interpolated using polynomial interpolation as in the Eq. (4.3) using three neighbor values as seen in Fig. 4.3.

For cells with rectangular shape and small skewness the term with pressure gradient is approximated with second order accuracy, whereas higher order approximations of the surface integral are possible, for example, by using the Simpson rule.

Using the approach above for computing the mass fluxes at the nonconformal interface, it is ensured that the sum of the the mass fluxes along the one side of the interface is equal to the sum of the fluxes along the the other side of the interface. Thus, the conservative approach guarantees that the continuity equation on nonconformal grids can be solved efficiently as shown in the result section.

4.5.2 *Pressure-correction equation*

Similarly to the pressure-correction equation on conformal grids, the pressure-correction equation on nonconformal grids can be obtained as below:

$$-\sum_{k\in\partial\Omega\cup\partial\Omega_{\mathrm{I}}}\overline{\left[\frac{\mathrm{d}V}{\tilde{\tilde{A}}}\right]}_k\vec{\nabla}p_k^c\cdot d\vec{S}_k+\underbrace{D_P((\tilde{\tilde{A}}^{-1}-L^{-1})\vec{\nabla}p^c\mathrm{d}V)}_{\texttt{Compatibility term}}=\sum_{k\in\partial\Omega\cup\partial\Omega_{\mathrm{I}}}f_k^{*,m+1}\tag{4.24}$$

where the operator D_P computes the divergence of any vector field $\vec{\phi}$ as below:

$$D_P(\vec{\phi})=\sum_{k\in\partial\Omega\cup\partial\Omega_{\mathrm{I}}}\rho\overline{[\vec{\phi}]}_k\cdot d\vec{S}_k$$

and the mass flux prediction $f_k^{*,m+1}$ is defined by the prediction form of Eq. (4.21):

$$\begin{aligned}f_k^{*,m+1}=\overline{\left[\frac{A^V}{\tilde{\tilde{A}}}\right]}_k\widetilde{f}_k^n+(1/\alpha-1-\beta)\overline{\left[\frac{A}{\tilde{\tilde{A}}}\right]}_k f_k^m+&\\ \overline{\left[\frac{\vec{h}^{*,m+1}}{\tilde{\tilde{A}}}\right]}_k\cdot d\vec{S}_k-\overline{\left[\frac{\mathrm{d}V}{\tilde{\tilde{A}}}\right]}_k\vec{\nabla}p_k^m\cdot d\vec{S}_k&\end{aligned}\tag{4.25}$$

with $\vec{h}_P^{*,m+1}$ being the prediction of $\vec{h}_P^{m+1}$ in Eq. (4.20):

$$\vec{h}_P^{*,m+1}=\sum_{k\in\partial\Omega\cup\partial\Omega_{\mathrm{I}}}A_{N(k)}\vec{v}_{N(k)}^{*,m+1}-\gamma A_P\vec{v}_p^{*,m+1}+\beta A_P\vec{v}_p^m+\vec{S}_P^m\tag{4.26}$$

If the *PS* interpolation is used consistently with the SIMPLE-like algorithms (PS_E interpolation is used with SIMPLE algorithm or PS_C interpolation is used with SIMPLEC algorithm), the pressure-correction equation, Eq. (4.24), reduces to the following form:

$$\sum_{k\in\partial\Omega\cup\partial\Omega_{\mathrm{I}}} f_k^c = \sum_{k\in\partial\Omega\cup\partial\Omega_{\mathrm{I}}} f_k^{*,m+1} \tag{4.27}$$

where the correction flux f_k^c is defined as:

$$f_k^c = -\overline{\left[\frac{\mathrm{d}V}{\widetilde{\widetilde{A}}}\right]_k} \vec{\nabla} p_k^c \cdot d\vec{S}_k \tag{4.28}$$

In the case of the nonconformal interface, i.e. for $k \in \partial\Omega_{\mathrm{I}}$, the correction flux f_k^c may be expressed using the pressure force approximation in Eqs. (4.22) and (4.23) as:

$$f_k^c = a_k \left(p^c_{\widehat{N_k^1}} - p^c_{\widehat{M_k^1}} \right) \tag{4.29}$$

The correction fluxes can be further approximated as the order of the correction flux approximation may influence only the convergence rate, but not the solution accuracy.

4.5.3 *Correction flux approximation*

The mass flux corrections in Eq. (4.27) depend on pressure correction values at the auxiliary nodes $\widehat{N_k^1}$ and $\widehat{M_k^1}$. There are several different choices about the auxiliary node interpolation, which result in different methods in composing the computation matrix:

- First choice is 1st order interpolation, i.e. using the value of the nearest cell center node as the value at the auxiliary node.
- Second choice is 2nd order polynomial interpolation, which was used for interpolating auxiliary nodes for the mass flux predictions. Contrary to the fist approach, this approach results in much more irregular shape of the computational matrix for the PC equation.

In the current work the first approach is preferred. In this case, even though the correction flux is defined with relatively low accuracy, it does not influence the accuracy of the solution. This may require a higher number of sub-iterations at each time-step for the transient solver or iterations for the steady solver.

Thus, in the approximation of the correction flux the pressure correction values $p^c_{\widehat{N_k^1}}$ and $p^c_{\widehat{M_k^1}}$ at auxiliary nodes $\widehat{N_k^1}, \widehat{M_k^1}$ (Fig. 4.3) are replaced by the cell center values $p^c_{N^1_{k_1}}$ and $p^c_{M^1_{k_1}}$, respectively. The resulting expression for the correction mass flux, f_k^c, becomes as:

$$f_k^c = a_k \left(p^c_{N^1_{k_1}} - p^c_{M^1_{k_1}} \right) \tag{4.30}$$

The final form of the pressure correction equation becomes:

$$a_p p_P^c + \sum_{k \in \partial\Omega \cup \partial\Omega_I} a_k p_{N(k)}^c = \sum_{k \in \partial\Omega \cup \partial\Omega_I} f_k^{*,m+1} \tag{4.31}$$

where $a_p = -\sum_{k \in \partial\Omega \cup \partial\Omega_I} a_k$ and the sum $\sum_{k \in \partial\Omega \cup \partial\Omega_I}$ is taken over the neighbor cell faces as seen in Fig. 4.1a.

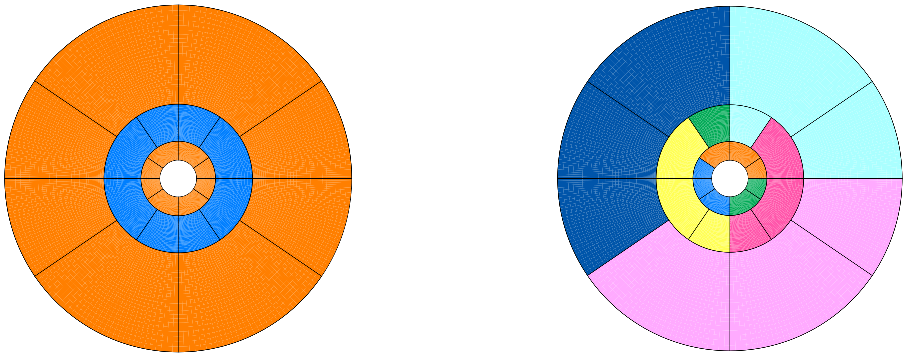
When the pressure correction is solved the pressure is corrected by the newly found and under relaxed pressure correction.

We may summarize that for each segment at a discontinuous interface the mass fluxes equally contribute to the source terms on both sides of the interface. It enables us to solve the PC equation efficiently, as will be seen in the Chapter. 5.

4.6 PARALLELIZATION AND PREPROCESSOR

It is known that many computational methods for application to wind turbine computations are computationally expensive, therefore parallelization of the methods using MPI or Open Multi-Processing (OpenMP) libraries is usually required.

In comparison with the computations on regular grids, the parallel algorithms for applications on grids with nonconformal blocks have to be able to ensure the data exchange between the nonconformal blocks. In such domains, the connections between blocks and the connections between processes are not of regular structure, which complicates the development of the parallel algorithm.



(a) Domain separated by two interfaces into three subdomains.

(b) Blocks distributed over 8 processes and separated by two interfaces.

Figure 4.4: Example of a domain consisting of 24 grid blocks, which are separated by two intermediate interfaces, for computation of flows around a circular cylinder.

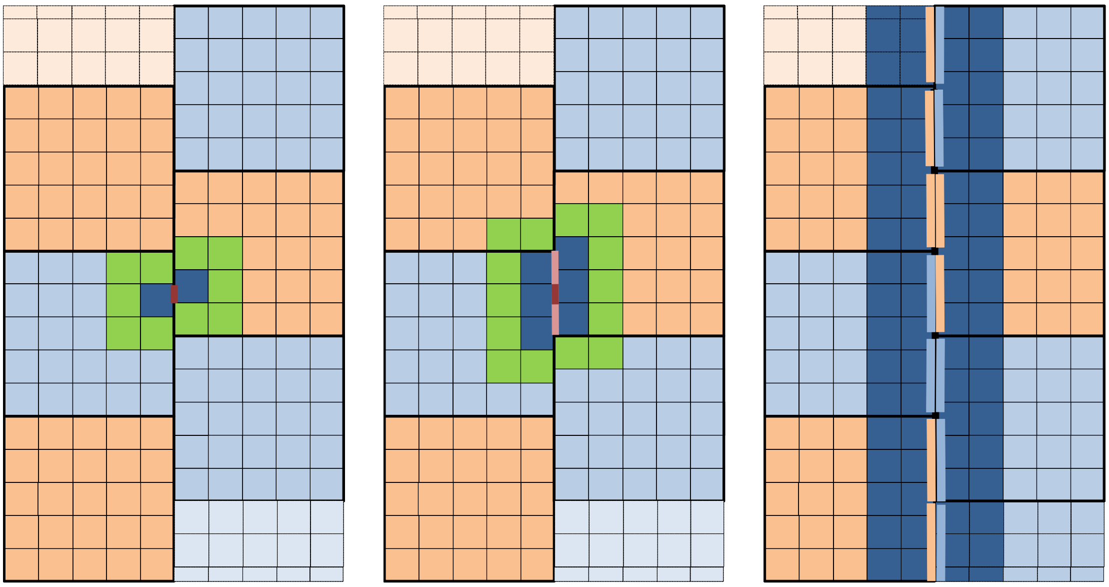
Using a test problem of flows around a circular cylinder it is possible to demonstrate the multiprocess computation in the domain with nonconformal blocks. The computational domain (see Fig. 4.4a) is divided by two interfaces into three subdomains (see

Fig. 4.4b). The grid blocks of the domain were originally distributed over 8 processes and then the middle subdomain was rotated anti-clockwise 90 degrees, such that the grid on the interfaces became discontinuous (see e.g. Fig. 5.10a). This test problem demonstrates clearly, that the connections between the processes on discontinuous grids may be considerably irregular.

To cope with the irregular interprocessor connectivity, the parallel MPI-based algorithm of EllipSys2D solver [55, 77], which was originally developed for conformal block-structured grids, is modified as described below.

To ensure high efficiency of the data exchange between the nonconformal blocks, the flow field data stored in the block-structured format of the EllipSys2D/3D code is translated into an interface format (see Fig. 4.5), composed of four interface units as below:

- Cell edge segment s_e^c - part of a cell face shared with a neighbor cell (see Fig. 4.5a)
- Block face segment s_f^b - part of a block face shared with a neighbor block (see Fig. 4.5b)
- Group of cell edge segments s_e^c belonging to s_f^b (see Fig. 4.5b)
- Group of block face segments s_f^b belonging to each processor (see Fig. 4.5c)


(a) Cell edge segment.

(b) Block face segment.

(c) Interface.

Figure 4.5: Interface data structure of a planar interface.

A preprocessor is used to build the interface structure together with communication tables, which are used during the computations. First, the preprocessor identifies the interfaces between grid blocks and creates groups of blocks and groups of processes sharing the same interfaces. A unique MPI communicator is assigned for each process group, such that during the computations the data is broadcasted within each group using the own MPI communicator. Using this algorithm, domains with multiple interfaces can be handled, such that one process may handle more than one interface.

To compose the communication tables, a two-step fast neighbor search is used: at the first step the neighbor blocks are found and at the second step the neighbor cells are identified within these neighbor blocks.

As was shown in Sec. 4.3, computations on nonconformal interface require flow field to be interpolated using the cells placed within the first two grid layers on each side of the interface. For interpolations between two blocks at the interface (see Fig. 4.5b) , data from the cells belonging to neighbor blocks is also required. Therefore, in the current implementation, double layer buffer along the *whole* interface is allocated (see Fig. 4.5c) by each process, which exchanges data at the interface. During computations the data from all processes is gathered using the MPI_ALLGATHERV routine. This approach simplifies building the communications tables between the processes, whereas the computational overhead is negligible, as the interface data is 1-dimensional in 2D and 2-dimensional in 3D.

5 MULTIGRID AND OPTIMIZED SCHWARZ METHODS ON NONCONFORMAL GRID INTERFACES

5.1 INTRODUCTION

Multigrid methods are one of the most powerful tools for solution of Poisson type Pressure Correction (PC) equations. The matrix of the PC equations when discretized on grids with nonconformal interfaces, has irregular structure: neither the number of elements per row, nor the bandwidth is constant. Extending multigrid method to grids with nonconformal interfaces is not a straightforward task, as the full power of the multigrid method can be obtained only if all parts of it: relaxation, prolongation, coarsening and coarse grid solution are able to accurately handle the irregular structure of the computational matrix and the nonconformal interface structure. Moreover, care has to be taken, when the mass fluxes, placed in the right hand side of the pressure-correction equation, are transferred between the grid level of the multigrid solver.

In this chapter the multigrid method in the EllipSys flow solver originally developed by Jess Michelsen in Department of Fluid Mechanics, DTU for conformal block-structured grids is extended to the block-structured grids with nonconformal blocks. In the original multigrid method an Incomplete Block LU (IBLU) factorization [76] is used as the error smoother with weakly overlapped domain decomposition, where the overlap in one layer of cells is used to update residual after each inner iteration. Coarse grid correction is based on a IBLU-preconditioned Conjugate Gradient (CG) method. The domain decomposition applied for the IBLU relaxation scheme is based on the Optimized Schwarz (OS) method of Lions [50]. The robustness and efficiency of the original multigrid method have been proven in practice since 1999 for both 2D and 3D engineering application, as exploited in the EllipSys2D/3D flow solver [79, 81].

Each operator of the multigrid solver, such as the operators of relaxation, coarsening, prolongation and the coarse grids solver, will be considered separately and the procedure to extend them to nonconformal grids is presented. First, the Optimized Schwarz method employed in the IBLU relaxation scheme, will be shown to fulfill some optimal conditions ensuring the high convergence rate of the multigrid method. Then, the OS method will be extended to nonconformal grids, ensuring the convergence rate as fast as on conformal grids. Modified operators of coarsening and prolongation will be presented later together with the modified IBLU-preconditioned Conjugate Gradient (CG) and Restarted Generalized Minimal Residual (GMRESR) methods. Finally, the modified multigrid method will be tested together with the finite volume method developed in

Chap. 4. Computational result of flow fields around a circular cylinder on grids with two nonconformal interfaces will be presented.

5.2 OPTIMIZED SCHWARZ METHOD

As discussed in the introduction part of this thesis, the Optimized Schwarz method using Robin transition condition is a prospective method for a multigrid method as by setting properly the Robin parameter, the range of low error frequencies can be eliminated faster and the convergence rate of the multigrid method can be increased. From literature, only optimal scaling of the Robin parameter is known: in the regions away and close to the grid cross points it has to scale as $O((h)^{-1/2})$ and $O(h^{-1})$, respectively. Specific definition of the Robin parameter has not been presented yet, therefore practical applications of the method is very limited in literature.

In the next sections a Robin parameter employed in EllipSys flow solver is described. On block-structured grids the parameter is shown to fulfill the optimal scaling at cross points. Then this approach is extended to block-structured grids with nonmatching blocks.

5.2.1 *Geometry dependent Robin parameter in Optimized Schwarz method*

In this section a geometry dependent Robin parameter, proposed by Jess Michelsen in the scope of a two step relaxation scheme, is described. Using some recent theory it is shown that the parameter scales optimally at block interfaces, ensuring robustness and effectiveness of the Robin transition condition.

One may consider an interface between two grid domains as shown in Fig. 5.1. For the control volume at the interface the algebraic Pressure Correction (PC) equation of the collocated grid-based SIMPLE-like algorithm is written as:

$$a_P p_P^{c,l+1} + \sum_{k=e,w,n,s} a_k p_{N(k)}^{c,l+1} = S_P \tag{5.1}$$

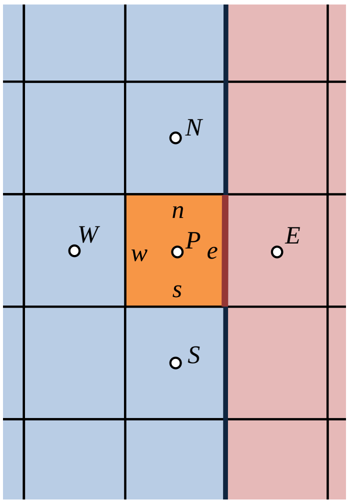


Figure 5.1: Control volume at block interface.

where p^c is the pressure correction, a_P and a_k are diagonal and nondiagonal terms of the matrix of the pressure-correction equations, the sum $\sum_{k=e,w,n,s}$ is taken over the cell faces and $N(k)$ denotes the corresponding neighbor cell centers $N(k) = \{E, W, N, S\}$ as seen in Fig. 5.1 and S_P is the control volume mass residual $\sum\limits_{k=e,w,n,s} f_k^*$, where f_k^* is the mass flux prediction. The superscript l denotes the counter of the inner iterations, used to find the pressure-correction for the certain mass residual S_P.

For the control volume, shown in Fig. 5.1, the Robin boundary condition at the east cell face e may be written as below:

$$a_e \left(\frac{\partial p^{c,l+1}}{\partial n} \right)_e + \lambda_e p_e^{c,l+1} = a_e \left(\frac{\partial p^{c,l}}{\partial n} \right)_e + \lambda_k p_e^{c,l} \tag{5.2}$$

where λ_e is some positive Robin parameter.

The discrete form of the equation is:

$$a_e \frac{p_E^{c,l+1} - p_P^{c,l+1}}{h} + \lambda_e \frac{p_E^{c,l+1} + p_P^{c,l+1}}{2} = g_e^l \tag{5.3}$$

where g_e^l denotes the right hand side of the Eq. (5.2) and depends on the former subiteration counter l, h is the grid size normal to segment k. As we are interested in the contribution from the boundary conditions to the diagonal term of the computational matrix, then we need the neighbor cell value $p_E^{c,l+1}$ to be expressed through the value at the considered cell $p_P^{c,l+1}$ as:

$$p_E^{c,l+1} = p_P^{c,l+1} B_e + g_e^l / (\frac{a_e}{h} + \frac{\lambda_e}{2}) \tag{5.4}$$

The parameter B_e may be expressed through the dimensionless parameter $\widehat{\lambda}_e = \frac{\lambda_e h}{a_e}$ as:

$$B_e = \frac{1 - \widehat{\lambda}_e/2}{1 + \widehat{\lambda}_e/2} \tag{5.5}$$

And the contribution to the diagonal term of the computational matrix of the PC equation becomes:

$$a_p = - \sum_{k=w,n,s} a_k - a_e B_e \tag{5.6}$$

From the Eq. (5.5) and Fig. 5.2a it is seen that when the Robin parameter $\widehat{\lambda}$ changes from zero to plus infinity the boundary parameter B changes from the Neumann BC region with $B = 1$ to the Dirichlet BC region with $B = -1$ correspondingly.

Choosing the optimal Robin parameter $\widehat{\lambda}$ is equivalent to choosing the boundary parameter B. From the Fig. 5.2a the parameter $\widehat{\lambda}$ is taken in the vicinity of the Neumann region, such that for small $\widehat{\lambda}$ the corresponding boundary parameter B can be expressed using Taylor series as follows below:

$$B_e \approx 1 - \widehat{\lambda}_e \tag{5.7}$$

From the behavior of the parameter B in Fig. 5.2a it is seen that for $\widehat{\lambda}_e$ lower than 1/2 the parameter B_e is close to a linear function of $\widehat{\lambda}_e$. In that range of $\widehat{\lambda}_e$ Eq. 5.7 may be considered as accurate enough. Michelsen proposed to define the Robin parameter $\widehat{\lambda}_e$ as a weighted sum of the inverse relative distances to the nearest cross point d_1/dd and the opposite block side d_1/d_2:

$$\widehat{\lambda}_e = \frac{d_1}{dd} + \frac{d_1}{d_2} \tag{5.8}$$

where, as seen on Fig. 5.2b, d_1 is the half cell size at the interface, dd is the distance from the cell face center to the cross point of the nearest neighbor block and d_2 is the distance from the cell face to the opposite block side.

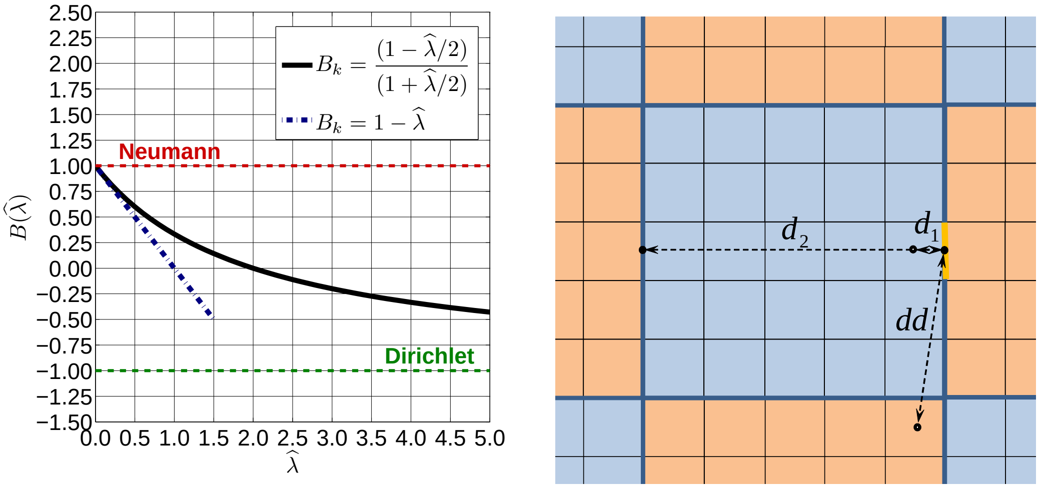
It can be checked that on grids with square cells, as the boundary cell position approaches to a cross point, the parameter $\widehat{\lambda}_e$, following Eq. (5.8), increases but does not exceed 1/2. Thus, at any position at block interface Eq. (5.7) is accurate enough. But for the grids stretched in the direction normal to the interface, the parameter $\widehat{\lambda}_e$ may exceed 1/2 due to the contribution from the inverse distance to the opposite block boundary. Therefore, to preserve the behavior of the boundary condition similar to the Neumann condition the parameter $\widehat{\lambda}_e$ is banded by 1/2, resulting in the parameter B_e to be limited from below by 0.5:

$$B_e = \max(0.5, 1 - \frac{d_1}{dd} - \frac{d_1}{d_2}) \tag{5.9}$$

To ensure the identity of the boundary parameter for each side of the interface, B_e is averaged with the value at other side of the interface as below:

$$B^{AV} = \frac{1}{2}\left(B_e + B_w\right) \tag{5.10}$$

where B_w is the boundary parameter B determined in the neighbor control volume at the opposite side of the interface.



(a) For small parameter $\widehat{\lambda}$ the boundary condition corresponds to Neumann region. $\widehat{\lambda}$ is taken in the vicinity of the Neumann region where $\widehat{\lambda} < 1/2$.

(b) On each side of the interface the Robin parameter is computed based on the local distance to the nearest cross point and the opposite block side.

Figure 5.2: Determining and scaling of the Robin parameter.

The Robin transition condition defined in Eqs. (5.10) and (5.8) fulfills several optimal conditions, summarized below:

- The Robin parameter is strictly positive, as required by Lions.
- The Robin parameter increases as the distance to the cross points decreases, which correlates well with the optimal asymptotic of Robin parameter by Gander [21].
- The Robin parameter fulfills the condition described by Gander and Kwok in [22], according to which the convergence of elliptic equations can be ensured if the Robin parameter scales at the cross points as $O(h^{-1})$.

The boundary parameter in (5.10) ensures that the Robin transition condition is similar to a Neumann condition leading to the faster convergence of low frequency components, which becomes advantageous in the scope of multigrid methods.

5.2.2 *Two step relaxation scheme*

The geometry dependent Robin parameter, described above, was used by Jess Michelsen to develop an Incomplete Block LU (IBLU) relaxation scheme of the multigrid solver in EllipSys2D/3D code. Michelsen parallelized the scheme using a two step OS method, according to which the global domain is subdivided into weakly overlapped subdomains, where overlap in only one cell layer is used only. At the boundaries of the subdomains boundary conditions are imposed and the relaxation scheme is solved in two steps, similarly to the method of Rice et al. in [66], but contrary to Rice et al., simple Dirichlet problem is solved at every even step, whereas at the odd step is solved using the geometry dependent Robin boundary described in the previous section.

The two step OS method has proven its value since 1999 as the backbone of the EllipSys2D and 3D solvers, as was exploited in the numerical experience in [78, 79, 81]. In Sec. 5.2.1 it was shown that the Robin interface operator of the OS method fulfills several conditions required for robust and efficient performance of the domain decomposition.

The theoretical foundation presented in Sec. 5.2.1 enables us to develop a Robin transition condition on grids with nonconformal blocks, which preserves the same efficiency as on conformal grids. In the next section the two step OS method will be extended to grids with nonconformal interfaces together with the coarse grid solver, prolongation and coarsening operators of the multigrid solver.

5.2.3 *Extension to nonconformal block-structured grids*

In this thesis nonconformal interfaces are treated as sub-domain block-boundaries of the domain decomposition and the contributions in the computational matrix from a neighbor domain are taken into account during the update of the residual vector. Besides residual update, the boundary conditions contribute to those diagonal terms of the computational matrix which correspond to cells at internal continuous or discontinuous boundaries. In this section it is shown how the contributions into diagonal terms may be approximated for Robin boundary conditions.

One may consider an interface between two grid blocks, which cells do not necessarily match at the interface. For the control volume adjacent to the interface (see Fig. 4.1a) the algebraic Pressure Correction (PC) equation is written as:

$$a_P p_P^{c,l+1} + \sum_{k\in\partial\Omega\cup\partial\Omega_\mathrm{I}} a_k p_{N(k)}^{c,l+1} = S_P \tag{5.11}$$

where recall that $\partial\Omega$ and $\partial\Omega_\mathrm{I}$ are the sets of conformal and nonconformal cell faces, respectively. For some control volume face segment k lying on the interface, i.e. $k \in \partial\Omega_\mathrm{I}$ (see Fig. 5.3), the Robin boundary condition can be formulated as:

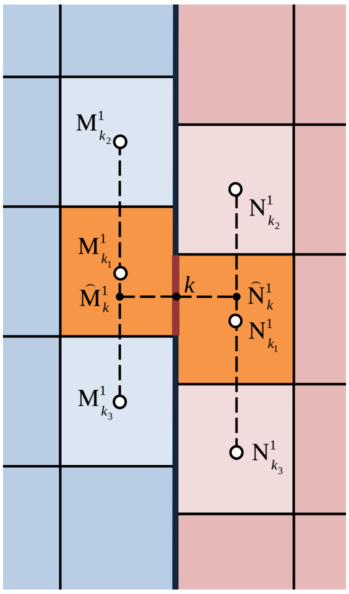


Figure 5.3: Approximation of Robin BC on nonconformal interface.

$$a_k \left(\frac{\partial p^{c,l+1}}{\partial n}\right)_k + \lambda_k p_k^{c,l+1} = a_k \left(\frac{\partial p^{c,l}}{\partial n}\right)_k + \lambda_k p_k^{c,l} \tag{5.12}$$

where k corresponds to the center of the face segment.

The equation may be written in discrete form as:

$$a_k \frac{p^{c,l+1}_{N^1_{k_1}} - p^{c,l+1}_{M^1_{k_1}}}{h} + \lambda_k \frac{p^{c,l+1}_{N^1_{k_1}} + p^{c,l+1}_{M^1_{k_1}}}{2} = g_k^l \tag{5.13}$$

Here the assumption that pressure-correction values at nodes $N^1_{k_1}$ and $M^1_{k_1}$ may represent the pressure-correction values at the auxiliary nodes $\widehat{N^1_{k_1}}$ and $\widehat{M^1_{k_1}}$ was used similarly to the assumption used for approximating the correction fluxes f_k^c in the Sec. 4.5.3. The neighbor cell center value $p^{c,l+1}_{N^1_{k_1}}$ is expressed through the value at the considered cell center $M^1_{k_1}$:

$$p^{c,l+1}_{N^1_{k_1}} = p^{c,l+1}_{M^1_{k_1}} B_k + g_k^l / (\frac{a_k}{h} + \frac{\lambda_k}{2}) \tag{5.14}$$

where denoting $\frac{\lambda_k h}{a_k}$ by $\widehat{\lambda}_k$ the parameter B_k after expanding in Taylor series, similarly to Eq. (5.7), is written as:

$$B_k \approx 1 - \widehat{\lambda}_k \tag{5.15}$$

The contribution to the diagonal term of the computational matrix of the PC equation on nonconformal interface becomes:

$$a_p = -\sum_{k\in\partial\Omega} a_k - \sum_{k\in\partial\Omega_I} a_k B_k \tag{5.16}$$

where the second sum in the RHS of Eq. (5.16) is the sum over the set of the cell face segments lying on the nonconformal interface (see Fig. 4.1a).

The description of the method on grids with nonconformal interfaces is finished as follows. On each side of the interface the boundary parameter B is computed at each

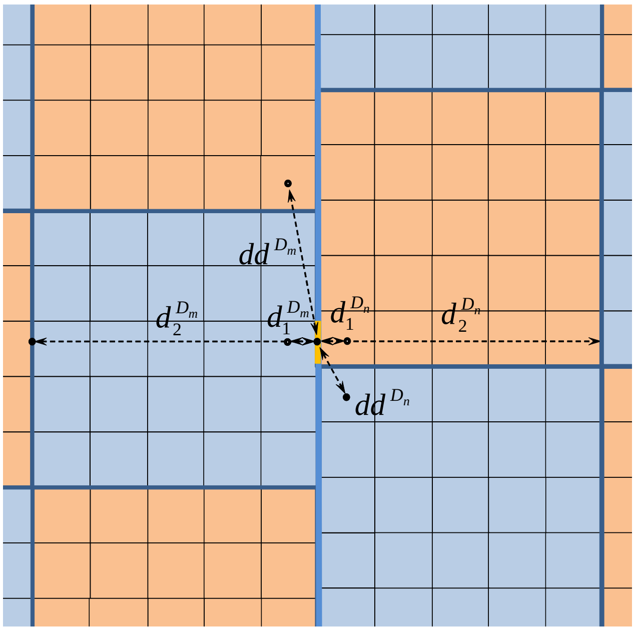


Figure 5.4: Robin parameter on the nonconformal interface.

interface segment (see Fig. 5.4) based on the local distance to the nearest cross point and the distance to the opposite block face using Eq. (5.9). Then the computed parameter B_k is averaged with the parameter B_k computed at the opposite side of the interface. In the current method we choose the weighting coefficient in Eq. (5.10) to be equal to $1/2$.

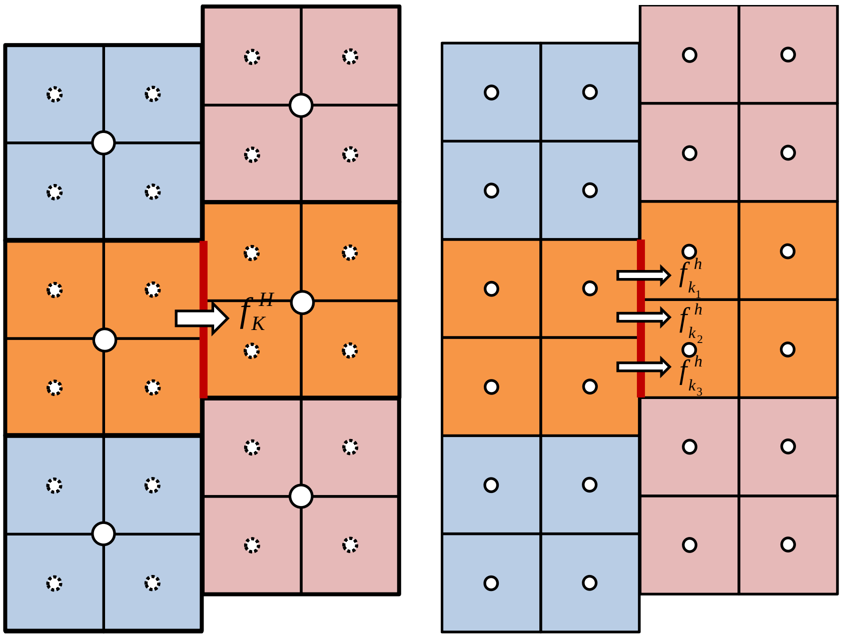
The results of flow computations, presented in Sec. 5.5.2, show that this boundary transition condition adopted to nonconformal grids as above is as effective as the transition condition on conformal grids.

5.3 OPERATORS OF RESTRICTION AND PROLONGATION

In this section operators of restriction and prolongation on grids with nonconformal blocks are presented. To ensure a consistency with the original multigrid solver, the coarse grid matrix and the restriction operator on grids with nonconformal interfaces are constructed using a modified intergrid mass conservation method, originally proposed by Michelsen [56] for conformal grids. Bilinear interpolation is used at conformal and nonconformal interfaces for the prolongation operator.

5.3.1 *Coarse grid operator*

To solve the system of the Pressure Correction (PC) equations on coarse grids of the multigrid method, the matrix of the system has to be composed on all coarse grid levels. An intergrid mass conservation method originally proposed in [56] and described in [77] for conformal grids, is modified here to compose the coarse grid matrix on grids with nonconformal interfaces.



(a) Flux at face segment K on coarse grid.

(b) Fluxes at face segments k_1, k_2 and k_3 on fine grid corresponding to the coarse grid segment K.

Figure 5.5: Intergrid flux conservation in the multigrid method.

To formulate the coarse grid matrix of the PC equation, first consider an interface between two grid blocks, which cells do not match at the interface. As was shown in Chap. 4, for a control volume adjacent to the interface (see e.g. Fig. 4.1b), the discrete Poisson Pressure Correction (PC) equation is given by Eq. (4.31), where the matrix terms, coming from the interface are defined by Eq. (4.23). To compose the matrix terms on coarse grid levels of the multigrid solver, the pressure-correction equation, Eq. (4.31), is written first on the finest grid level in the following compact form:

$$\sum_{k \in \partial\Omega^h} f_k^{c,h} + \sum_{k \in \partial\Omega_I^h} f_k^{c,h} = S_p^h \tag{5.17}$$

where superscript h indicates the fine grid level, S_p^h is the mass flux residual at control volume P, which is the sum of the mass flux predictions $f_k^{*,h}$:

$$S_p^h = \sum_{k \in \partial\Omega^h} f_k^{*,h} + \sum_{k \in \partial\Omega_I^h} f_k^{*,h} \tag{5.18}$$

and the correction mass flux $f_k^{c,h}$ is determined as:

$$f_k^{c,h} = a_k^h \Delta p_k^{c,h} \tag{5.19}$$

where Δp_k is the pressure difference at the cell face k.

One may consider the fine grid shown in Fig. 5.5b and assume that the cell face k in Eq. (5.19) corresponds to one of the cell faces k_1, k_2 or k_3. The set of these segments compose one coarse grid segment K, as seen in Fig. 5.5a.

Thus the flux at the coarse grid segment K equals to sum of the fluxes on the finer grid level as follows:

$$f_K^{c,H} = \sum_{k \in K} f_k^{c,h}$$

where superscript H indicates the coarse grid level and the sum is taken over all the fine grid segments, which the coarse grid segment K consists of. The equation above can be rewritten as follows:

$$a_K^H \Delta p_K^{c,H} = \sum_{k \in K} a_k^h \Delta p_k^{c,h}$$

Assuming that the pressure gradients on the fine and the coarse grids are equal, the following equality $\Delta p_K^{c,H} \approx 2\Delta p_k^{c,h}$ is satisfied and the coarse grid coefficient a_K^H is obtained as below:

$$a_K^H = 1/2 \sum_{k \in K} a_k^h \tag{5.20}$$

The diagonal term of the pressure-correction equation on the coarse grid is computed as:

$$a_P^H = -\sum_{K \in \partial\Omega^H} a_K^H - \sum_{K \in \partial\Omega_I^H} a_K^H \tag{5.21}$$

Equations, Eq. (5.20) and (5.21), define the coarse grid matrix terms, accounting for the diagonal and the nondiagonal terms of the pressure Poisson equation at the nonconformal interface.

Similarly to the mass flux corrections, the mass flux residual S_P^H at the coarse grid level is computed as the sum of the mass flux predictions at the fine grid level. For this, the contribution to the residual from the cell face segment K at the nonconformal interface is computed as the sum of the corresponding mass flux predictions at the finer grid level:

$$f_K^{*,H} = \sum_{k \in K} f_k^{*,h} \tag{5.22}$$

Thus, using Eq. (5.22) the coarse grid mass flux predictions can be computed and the mass flux residual at the coarse grid S^H is computed similarly to Eq. (5.18) and the pressure-correction equation at the coarse grid level is obtained as below:

$$a_P^H p_P^{c,H} + \sum_{K\in\partial\Omega^H\cup\partial\Omega_{\mathrm{I}}^H} a_K^H p_{N(K)}^{c,H} = S^H \tag{5.23}$$

The PC equation for the remaining coarse grids is obtained recursively and therefore not shown here.

5.3.2 *Prolongation operator*

To prolong the solution from coarse to fine grids, interpolation is required. This is achieved using a bilinear interpolation on the nonconformal interfaces as schematically shown in Fig. 5.6.

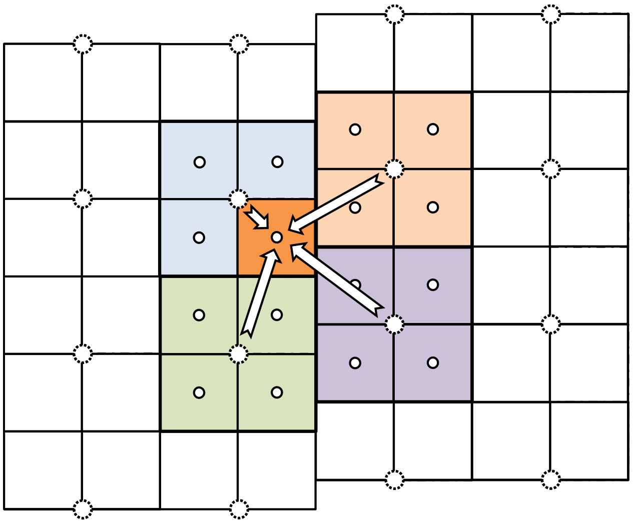


Figure 5.6: Prolongation operator based on bilinear interpolation using the values at the 4 nearest coarse grid cells.

5.4 COARSE GRID SOLVER

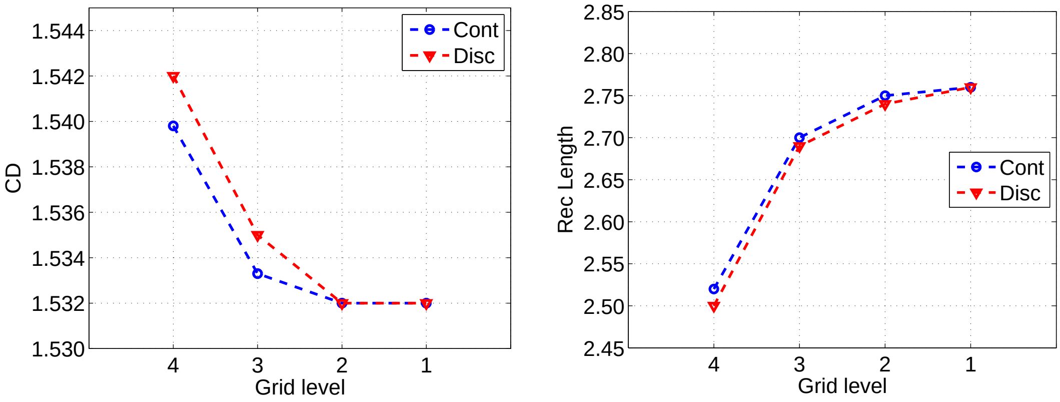
The IBLU-preconditioned Conjugate Gradient (CG) method, employed in EllipSys2D/3D code on the coarsest grid level, is solved in the global domain. To extend its performance on grids with nonconformal blocks, the coarse grid PC equation, defined by Eq. (5.23), is used. During the inner iterations of the CG solver, the interface matrix terms in Eq. (5.23) are taken into account during the update of the residual vector:

$$r_P^H = S_p^H - \sum_{K\in\partial\Omega^H\cup\partial\Omega_{\mathrm{I}}^H} a_K^H \Delta p_k^H \tag{5.24}$$

In addition to the modified CG method, the Restarted Generalized Minimal Residual (GMRESR) method of Saad and Schultz [68] is implemented in the EllipSys code and adopted for both conformal and nonconformal grids. To enhance its convergence speed the IBLU preconditioner is used.

5.5 results

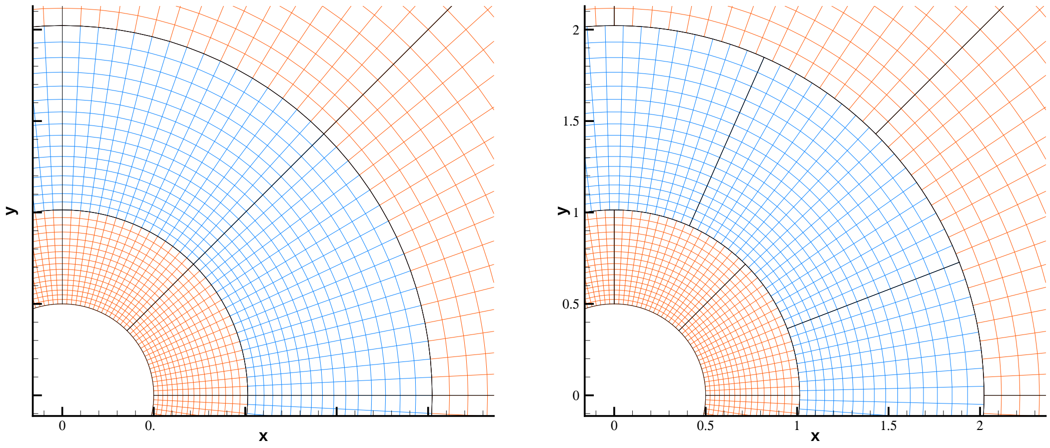
In the result section the finite volume method and the multigrid method extended to grid with nonconformal blocks are verified. First, the finite volume scheme is tested using steady and unsteady flow computations around circular cylinder at Re=40 and Re=100 on grids with two nonconformal interfaces. Second, the multigrid method and the Optimized Schwarz method is tested on the nonconformal grids using the steady flow around cylinder at Re=40.



(a) Drag coefficient

(b) Recirculation length

Figure 5.7: Results of computations of flows around circular cylinder at Re=40 on conformal and nonconformal grids of 1) 512x256 cells, 2)256x128 cells, 3) 128x64 cells and 4) 64x32 cells.



(a) Continuous grid with 128x64 cells.

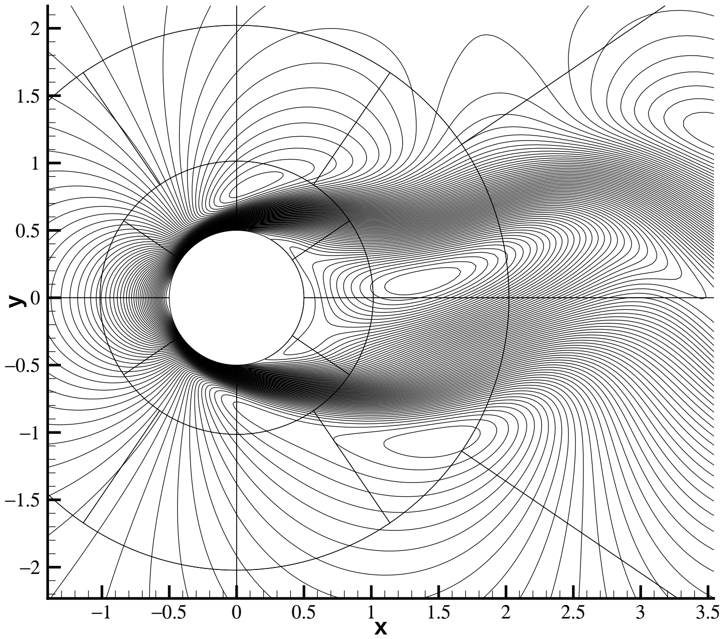
(b) Discontinuous grid with 128x64 cells.

Figure 5.8: Continuous and discontinuous grids with equidistant cells along interface.

The computational domain ranges 30 cylinder diameters and is covered by an O-mesh. Two cylindrical discontinuous interfaces are placed at distances of 0.5 and 1.5 of cylinder diameters away from the cylinder surface. The positions of the interfaces are deliberately placed in the area of high pressure gradients. SIMPLEC algorithm is used with the PS_c interpolation, which is fully compatible with the SIMPLEC algorithm (see Sec. 2.6). The steady- and unsteady- state solutions are achieved when the residuals are reduced by a factor of 10^8 and 10^4, respectively. The finest computational grid is composed of 256 cells in the radial direction and 512 cells in the tangential direction, whereas three successive coarser grids are constructed by removing every second point in both directions. On the finest grid the cell height at the wall is 0.02 cylinder diameter. The dimensionless time step is based on the cylinder diameter and the free stream velocity.

5.5.1 *Verification of finite volume method*

In this section results of computations of steady and unsteady flows around a circular cylinder at a Reynolds number of $Re = 40$ and $Re = 100$ are presented. For the unsteady flow computations the Strouhal number, mean, max and min values of drag and lift coefficients are compared at different grid levels and time instances.



(a) u velocity component contours.

Figure 5.9: Continuity of the flow field for $Re = 100$ on nonconformal grid with the maximum grid mismatch ratio along interface equal to 4.

First, for the steady state flow a grid convergence study is shown in Fig. 5.7a, indicating good agreement between the continuous and discontinuous setup, see Fig. 5.8a and Fig. 5.8b.

Second, for the unsteady computations of flows around a cylinder at $Re = 100$ the time step convergence analysis is made on the conformal grid with 256x128 cells. The results are presented in Table 5.2.

From the Table 5.2 it is seen that the difference of the solutions based on $\mathrm{dt} = 0.02$ and $\mathrm{dt} = 0.04$ is less than 0.5%. Taking the time step equal to 0.02 the grid dependence analysis for the unsteady computations on grid with discontinuous interfaces is made and results are shown in Table 5.1. The grid mismatch at the interface is equal to two cells per cell. An example of the nonconformal grid with 128x64 cells is shown in Fig. 5.8b.

Continuity of the flow field for $Re = 100$ is examined on a nonconformal grid with maximum grid mismatch ratio at the interface of 16 as shown on Fig. 5.10c.

Contours of pressure u velocity component on grid with grid mismatch ratio equal to 4 are presented in Fig. 5.9, whereas the pressure contours on grids with the grid mismatch ratio equal to 4 and 16 are shown in Figs. 5.10a and 5.10d, respectively. The flow field continuity is seen to be well satisfied in the whole domain including the areas where two nonconformal interfaces are placed.

Grid	St	$\overline{\mathrm{CD}}$	$max(\mathrm{CD})$	$min(\mathrm{CD})$	$max(\mathrm{CL})$	$min(\mathrm{CL})$
64x32	0.1346	1.2132	1.2146	1.2119	0.1128	-0.1128
128x64	0.1565	1.2988	1.3056	1.2919	0.2600	-0.2599
256x128	0.1636	1.3310	1.3400	1.3219	0.3169	-0.3169
512x256	0.1646	1.3319	1.3405	1.3233	0.3136	-0.3136

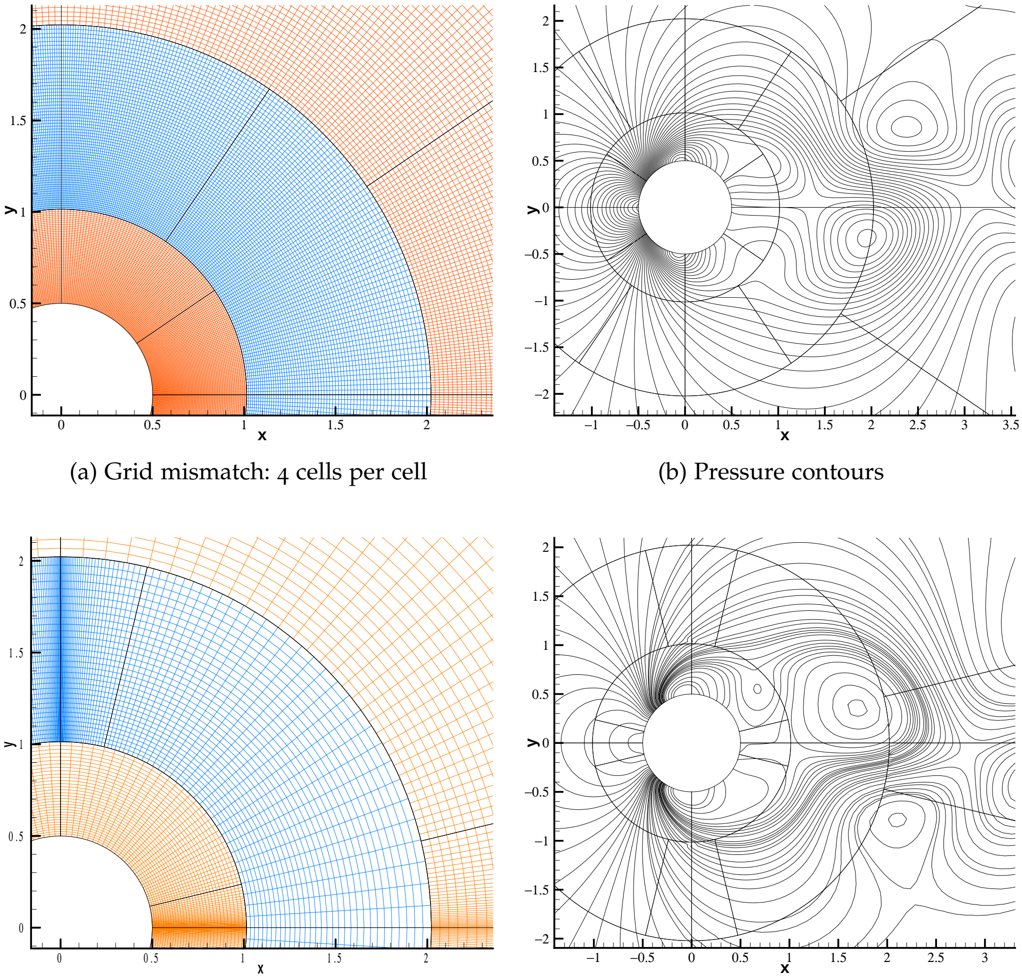
Table 5.1: Solution dependence on grid resolution on nonconformal grids. Grid mismatch at interfaces of each grid is set to be 2 cells per cell.

dt	St	$\overline{\mathrm{CD}}$	$max(\mathrm{CD})$	$min(\mathrm{CD})$	$max(\mathrm{CL})$	$min(\mathrm{CL})$
0.16	0.1575	1.3069	1.3125	1.3014	0.2509	-0.2509
0.08	0.1635	1.3297	1.3379	1.3214	0.3070	-0.3070
0.04	0.1647	1.3350	1.3440	1.3260	0.3216	-0.3216
0.02	0.1647	1.3351	1.3441	1.3261	0.3233	-0.3233

Table 5.2: Solution dependence on time step for computations at $Re = 100$ on conformal grid with 256x128 cells.

5.5.2 *Multigrid performance*

Computations of flows around circular cylinder at Re=40 on grids with nonconformal blocks are used in this section to test the modified multigrid method, described in Secs. 5.2-5.4. For this, O-mesh is used with two cylindrical interfaces (see Fig. 5.10a), where the intermediate domain was rotated to 90 degrees, thus making discontinuous the grid at the interfaces. The computational grid is composed of 256 cells in the radial direction and 512 cells in the tangential direction. The multigrid performance is tested by solving the Pressure Correction equation where PS_c interpolation, fully compatible with SIMPLEC algorithm, is used.



(a) Grid mismatch: 4 cells per cell

(b) Pressure contours

(c) Grid mismatch: 16 cells per cell

(d) Pressure contours

Figure 5.10: Pressure contours on weakly and strongly discontinuous grids.

The multigrid method is used here to solve the Pressure Correction equation in the scope of the SIMPLEC algorithm, described in Chapter. 4.

Recall, that the Incomplete Block LU (IBLU) factorization scheme, used as the relaxation operator in the current multigrid method, is solved using Robin/Dirichlet boundary

Grid type	IBLU as solver		Multigrid	
	CS	OS	CS	OS
Continuous	17619	6036	328	229
Discontinuous	17451	5991	331	231
	Iterations		Cycles	

Table 5.3: Convergence rate of schemes based on Classical Schwarz (CS) and Optimized Schwarz (OS) methods. Two schemes are used: 1) IBLU factorization as the single-grid solver; 2) Multigrid method with the IBLU factorization as relaxation scheme. Computational efforts are compared for the first and the second methods using the total number of iterations and the total number of multigrid cycles, respectively.

conditions: with Robin BC at each odd step and with Dirichlet BC at each even step (see Sec. 5.2.2). This relaxation scheme, here referred to as Optimized Schwarz (OS)-based scheme, is compared with the Classical Schwarz (CS)-based scheme, i.e. the scheme where Robin/Dirichlet steps are simply replaced with the Dirichlet/Dirichlet steps.

The both approaches are compared using the IBLU factorization either as single grid solver or relaxation operator of the Multigrid method.

First, the IBLU is used as a single grid solver. The total number of iterations needed for the computations is shown in Table 5.3. It is seen that on both continuous and discontinuous grids the OS-based solver is about 3 times as efficient as the CS-based solver.

Second, the IBLU is applied as relaxation scheme of the multigrid method, described in Secs. 5.2-5.4. In Table 5.3 by comparing the OS-based multigrid with the CS-based multigrid on both continuous and discontinuous grids, it is seen that the OS-based multigrid method is about 1.4 times more efficient than the CS-based multigrid method. The advantage in speed of the OS domain decomposition over the CS domain decomposition when multigrid is employed is not as high as when IBLU solver is used as single grid solver. This is explained by the fact that in the multigrid method the Conjugate Gradient method employed in the global domain on the coarsest grid (see Sec. 5.4) provides good reduction of long wave error frequencies.

It is seen from Table 5.3 that the OS methods on continuous grids are as efficient as the OS methods on discontinuous grids. This fact shows that the new OS method for the nonconformal grids, presented in Sec. 5.2.3, preserves high efficiency of the original EllipSys2D/3D solver.

5.6 conclusions

A 2nd order finite volume method on collocated grids with nonconformal blocks was presented. To link velocity and pressure, the Navier-Stokes equations were solved using

SIMPLEC algorithm with the PS_C interpolation method. Pressure-correction equation was discretized on the nonconformal interfaces such that the simplicity and the accuracy of the algorithm is preserved. Algorithm for the parallel data exchange based on the MPI libraries was presented for computations on the grids with nonconformal blocks. Computational experiments using the SIMPLEC algorithm shown, that the developed computational scheme, results in solutions as accurate as on conformal grids.

The multigrid method of the cell-centered/block-structured EllipSys2D code was extended to the block-structured grids with nonconformal interfaces. Based on an intergrid mass conservation, coarse grid operator was modified and simple structure of the computational matrix on coarse grids was preserved. Prolongation operator as well as Krylov type methods were modified to ensure their performance on the nonconformal grids.

An Optimized Schwarz (OS) domain decomposition using Robin boundary conditions for IBLU relaxation scheme was presented in the frame of multigrid method on discontinuous grids. The presented Robin parameter, depending on grid geometry and grid discontinuity at block interfaces, was shown to fulfill several conditions, required to ensure high convergence of the OS method.

Results from computations of laminar flows around a circular cylinder on grids with nonconformal blocks shown, that the modified IBLU relaxation scheme and the multigrid method preserve high efficiency of the original relaxation scheme and the original multigrid method of EllipSys solver. It was demonstrated that compared to the Classical Schwarz domain decomposition the Optimized Schwarz domain decomposition increases performance of a single grid - and multigrid - solvers up to 3 times and 1.4 times, respectively.

6 CONCLUSIONS AND RECOMMENDATIONS

6.1 CONCLUSIONS

The main goal of this thesis, as stated in Chapter 1, was to develop efficient finite-volume and multigrid methods for incompressible Navier-Stokes equations on collocated grids with nonconformal interfaces and implement them in the EllipSys flow solver. As was presented in this thesis a computational method for applications on grids with nonconformal interfaces was developed using:

1. Conservative finite volume method with implicit interface treatment

2. Geometric Multigrid method enabled at all grid levels

3. Optimized Schwarz (OS)-based relaxation scheme in the Multigrid method

4. Fully consistent SIMPLE-like algorithm to couple velocity and pressure

In the method a strong solution coupling is preserved at nonconformal interfaces using fully implicit and conservative treatment, the sparse and irregular matrix of the pressure-correction equation is handled at all grid levels of the Multigrid solver, which convergence speed is enhanced using Optimized Schwarz-based Incomplete Block LU relaxation scheme, whereas the pressure-velocity coupling is ensured by the newly developed Fully Consistent SIMPLE-like algorithm resulting in faster and more accurate solutions on grids with the collocated arrangement. This comprehensive computational method presents a robust, accurate and efficient platform, which can be used for further development of Sliding Grid method for full rotor/tower computations.

A summary on the main developments, which were presented in each chapter of this thesis, and the recommendations for applications and further developments are summarized below.

6.1.1 *Momentum interpolation*

It was shown first that both the solution accuracy and the convergence rate of collocated grids-based SIMPLE-like algorithms are strongly dependent of choice of the mass flux interpolation method on collocated grids. We derived a compatibility condition between the SIMPLE-like algorithms and the momentum interpolation methods, which enhances the convergence rate of the algorithms. Most of the existing interpolation were shown to be fully consistent with SIMPLE algorithm. A new Fully Consistent SIMPLEC algorithm was developed. Using this algorithm it was shown that appropriate choice of interpolation methods may not only enhance the convergence rate, but also increase the solution accuracy of the SIMPLE-like algorithm on the collocated grids. New interpolation, which is time step- and relaxation parameter- independent at convergence, was developed. It was shown to result in a higher convergence rate of the SIMPLEC algorithm, compared to other interpolation methods.

Using a typical turbulent test case, it was shown that the magnitude of the time step dependence for some standard momentum interpolations, may achieve 5% of the solution value. The new interpolation was proved to be free from this error. When coarse grids are employed the erroneous behavior of the standard interpolations was shown to increase significantly. For this type of grids, the new PI^0 interpolation is recommended as the most accurate. The parameters given in Table. 3.2 can be used for the practical applications of the new interpolation.

6.1.2 *Finite-Volume/Multigrid methods*

A second order finite volume method of EllipSys flow solver was extended to grids with nonconformal interfaces. An implicit and conservative treatment was used to preserve good accuracy and strong solution coupling at the interfaces. The Geometric Multigrid method was also extended to the nonconformal grids preserving high efficiency and robustness of the original solver. IBLU-preconditioned Krylov-type methods, prolongation and coarse grid operators were successfully extended to block-structured grids with nonconformal interfaces. The IBLU relaxation scheme with Optimized Schwarz method, enhancing the convergence rate of the scheme was discussed. This finite-volume/multigrid method was shown to be robust, accurate and efficient, and presents a platform for further development of Sliding Grid method for full rotor/tower computations, which is discussed below.

6.1.3 *Optimized Schwarz domain decomposition method*

Advantages of using the Optimized Schwarz (OS) domain decomposition method were demonstrated in incompressible flow computations. Compared to the Classical Schwarz (CS) domain decomposition a two-step domain decomposition method with Robin transition condition was shown to enhance the performance of single grid - and multigrid - solvers 3 times and 1.4 times, respectively. This OS method was extended to grids with nonconformal interfaces and the same increase in efficiency compared to the CS domain decomposition was shown. This demonstrates two facts:

- Robin boundary condition at internal boundaries can be very beneficial in incompressible flow computations

- The extension of the Robin transition condition to nonconformal interfaces, which was presented in this thesis, retains the high efficiency of the Robin transition condition on conformal interfaces

6.2 RECOMMENDATIONS

The finite-volume method developed in this thesis has shown reliable results on stationary grids with nonconformal interfaces. The goal of this thesis was also to provide a platform for further development of the finite-volume method for applications on moving grids with nonconformal interfaces (sliding grids). Most of the components of the developed method have good prospects of further extension to the sliding grid method as considered below.

6.2.1 *Finite-Volume/Multigrid method extension to sliding grids*

To extend the implicit and conservative finite-volume method, developed in this thesis, to sliding grids, the most evident choice is the technique by Demirdžić and Perić [3, 15], where the equations are solved using a fully conservative procedure in an arbitrary moving control volumes. To enable grid motion the convective terms have to be corrected and the local grid adaptation can be done at the sliding interface as e.g. described by Basara et al. [3]. The remaining parts of the current finite-volume method do not require additional changes.

The multigrid method, when extended to moving grids, does not require any additional changes. Combining this with the high convergence rate of the multigrid method obtained on grids with nonconformal interfaces, as was shown Sec. 5.5.2, it can be concluded that high efficiency of the multigrid solver can be preserved on moving grids also.

6.2.2 *Optimized Schwarz domain decomposition extension to moving grids*

The Optimized Schwarz (OS) method, developed in this thesis for applications on grids with nonconformal interfaces, demonstrates a prospective of this domain decomposition not only for sliding grid, but also for overset grid methods. Extension of the OS method to sliding grid is straightforward and the same high efficiency of the multigrid method is expected.

It is known, that in incompressible overset grid method the solution of the multigrid solver is not straightforward due to complicated solution of the pressure-correction equation on the coarsest grid level. Therefore, the straightforward choice is to exclude the solution on the coarsest grid level from the multigrid solver. In this case fast elimination of low error frequencies is not guaranteed and the convergence rate is decreased.

As a remedy for this problem, the OS method with Robin transition condition in the overlapped regions, can be used for the relaxation scheme employed on intermediate grid levels. As the Robin boundary transition condition with properly adjusted Robin parameter is known to enhance elimination of the low error frequencies, the performance of the multigrid solver can be enhanced even without the coarse grid correction.

This demonstrates an example of high prospective of the OS method in incompressible flow computations.

6.2.3 *Momentum interpolation extension to moving grids*

The Fully Consistent SIMPLE-like algorithm developed in this thesis, has been demonstrated to result in higher convergence rate and better accuracy compared to standard SIMPLE-like algorithms. This was demonstrated on stationary grids and unfortunately its extension to moving grids is not straightforward. When the moving grids are considered, employing the new PI^0 interpolation or the popular interpolations of Choi or Shen et al. becomes problematic, as these interpolations rely on the mass fluxes from the former time steps. On the moving grids, these former mass fluxes are not available, therefore instead of the interpolations of Choi or Shen et al., Majumdar's type interpolation can be used, which doesn't rely on the former time step fluxes, but uses the fluxes from the former subiteration. In spite of the fact, that this method can not be considered as fully consistent, it must be the most reliable method among the existed interpolation methods for the moving grids. Moreover, the technique enhancing convergence speed of the SIMPLE-like algorithms by fulfilling a compatibility condition, presented in this thesis, can be used for this interpolation in the moving grid setup.

Part III

APPENDIX

A

THE PI^0, PI^∞ AND PS INTERPOLATIONS

In Sec. 3.3 it was shown that the flux of the PI^0 method at steady state satisfies Eq. (3.5). We will show below that the same expression can be obtained for the PS method at convergence to steady state if an infinitely small time step is employed. In Sec. 3.3 it was also shown that the flux of the PI^∞ method at steady state satisfies Eq. (3.6). The same expression can also be obtained for the PS method at convergence to steady state if an infinitely large time step is employed.

Proof.

The flux of the PS method, defined by Eq. (2.11), at convergence of subiterations (i.e. when $f_k^{m+1} = f_k^m = f_k^{n+1}$ are satisfied) gives the following form:

$$f_k^{n+1} = \frac{\overline{\left[\frac{\mathrm{d}V}{\widetilde{\widetilde{A}}}\right]_k} \widetilde{f_k^n} + \overline{\left[\frac{h^{\vec{n}+1}}{\widetilde{\widetilde{A}}}\right]_k} \cdot d\vec{S}_k - \overline{\left[\frac{\mathrm{d}V}{\widetilde{\widetilde{A}}}\right]_k} \vec{\nabla} p_k^{n+1} \cdot d\vec{S}_k}{\overline{\left[\frac{A(1-\gamma+\beta)+1.5A^V}{\widetilde{\widetilde{A}}}\right]_k}} \tag{A.1}$$

where $\vec{h}_P^{n+1}$ is:

$$\vec{h}_p^{\,n+1} = \sum_{E,W,N,S} A_{nb}\vec{v}_{nb}^{n+1} - \gamma A_p \vec{v}_p^{n+1} + \beta A_p \vec{v}_p^{n+1} + \vec{S}_p^{n+1} \tag{A.2}$$

Using the definition of $\widetilde{f_k^n} = 1.5 f_k^{n+1} - 0.5 f_k^n$, the flux, Eq. A.1, at steady state (i.e. when $f_k^{n+1} = f_k^n = f_k^{n-1} = f_k$ the equalities are satisfied) becomes equal to:

$$f_k^{n+1} = \frac{\overline{\left[\frac{\vec{h}}{\widetilde{\widetilde{A}}}\right]_k} \cdot d\vec{S}_k - \overline{\left[\frac{\mathrm{d}V}{\widetilde{\widetilde{A}}}\right]_k} \vec{\nabla} p_k \cdot d\vec{S}_k}{(1-\gamma+\beta)\overline{\left[\frac{A}{\widetilde{\widetilde{A}}}\right]_k}} \tag{A.3}$$

It can be noted, that for the steady state flux, Eq. (A.3), no assumption on time step τ has been made yet.

Now, by considering an infinitely small time step, τ, the term $\widetilde{\widetilde{A}}_P$ scales as $1.5A_P^V$ and Eq. (A.3) becomes:

$$f_k = \frac{\overline{\left[\dfrac{\vec{h}}{\rho\,\mathrm{d}V}\right]_k} \cdot d\vec{S}_k - \dfrac{1}{\rho}\vec{\nabla} p_k \cdot d\vec{S}_k}{(1-\gamma+\beta)\,\overline{\left[\dfrac{A}{\rho\,\mathrm{d}V}\right]_k}}$$

which is the same as expression for the flux of the PI^0 method at steady state in Eq. (3.5).

On the other hand, the term $\widetilde{\widetilde{A}}_P$ at an infinitely large time step, τ, scales as $(1/\alpha-\gamma)$ and the flux of the PS method at steady state, defined by Eq. (A.3), gives the following form:

$$f_k = \frac{\overline{\left[\dfrac{\vec{h}}{A}\right]_k} \cdot d\vec{S}_k - \overline{\left[\dfrac{\mathrm{d}V}{A}\right]_k}\vec{\nabla} p_k \cdot d\vec{S}_k}{(1-\gamma+\beta)}$$

which is the same as expression for the flux of the PI^∞ method at steady state in Eq. (3.6).

This completes the proof.

BIBLIOGRAPHY

[1] R Abbasi, A Ashrafizadeh, and A Shadaram. A comparative study of finite volume pressure–correction projection methods on co–located grid arrangements. *Computers & Fluids*, 2013.

[2] Essam H Atta. Component-adaptive grid interfacing. *AIAA paper*, 81:0382, 1981.

[3] B Basara, A Alajbegovic, and D Beader. Simulation of single-and two-phase flows on sliding unstructured meshes using finite volume method. *International journal for numerical methods in fluids*, 45(10):1137–1159, 2004.

[4] O Botella and R Peyret. Benchmark spectral results on the lid-driven cavity flow. *Computers & Fluids*, 27(4):421–433, 1998.

[5] Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977. ISSN 00255718, 10886842.

[6] Achi Brandt. Multi-level adaptive technique (mlat) for fast numerical solution to boundary value problems. In *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, pages 82–89. Springer, 1973.

[7] PG Buningt. A 3-d chimera grid embedding technique. 1985.

[8] Tristan M Burton and John K Eaton. Analysis of a fractional-step method on overset grids. *Journal of Computational Physics*, 177(2):336–364, 2002.

[9] Jiangtao Cheng, Weijun Zhu, Andreas Fischer, Néstor Ramos García, Jesper Madsen, Jin Chen, and Wen Zhong Shen. Design and validation of the high performance and low noise cqu-dtu-ln1 airfoils. *Wind Energy*, 2013.

[10] G Chesshire and William D Henshaw. Composite overlapping meshes for the solution of partial differential equations. *Journal of Computational Physics*, 90(1):1–64, 1990.

[11] Seok Ki Choi. Note on the use of momentum interpolation method for unsteady flows. *Numerical Heat Transfer: Part A: Applications*, 36(5):545–550, 1999.

[12] A Crespo, J Hernandez, and S Frandsen. Survey of modelling methods for wind turbine wakes and wind farms. *Wind energy*, 2(1):1–24, 1999.

[13] Ana Cubero and Norberto Fueyo. A compact momentum interpolation procedure for unsteady flows and relaxation. *Numerical Heat Transfer, Part B: Fundamentals*, 52 (6):507–529, 2007.

[14] V. D'Alessandro, S. Montelpare, R. Ricci, and A. Secchiaroli. Unsteady aerodynamics of a savonius wind rotor: a new computational approach for the simulation of energy performance. *ENERGY*, 35(8):3349–3363, 2010. ISSN 03605442. doi: 10.1016/j.energy.2010.04.021.

[15] I Demirdžić and M Perić. Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries. *International Journal for Numerical Methods in Fluids*, 10(7):771–790, 1990.

[16] Earl PN Duque, CP Van Dam, and Shannon C Hughes. Navier-stokes simulations of the nrel combined experiment phase ii rotor. In *EWEC-CONFERENCE-*, pages 79–84, 1999.

[17] Earl PN Duque, Michael D Burklund, and Wayne Johnson. Navier-stokes and comprehensive analysis performance predictions of the nrel phase vi experiment. *Journal of Solar Energy Engineering*, 125(4):457–467, 2003.

[18] Bjorn Engquist and Hong-Kai Zhao. Absorbing boundary conditions for domain decomposition. *Applied numerical mathematics*, 27(4):341–365, 1998.

[19] Radii Petrovich Fedorenko. The speed of convergence of one iterative process. *USSR Computational Mathematics and Mathematical Physics*, 4(3):227–235, 1964.

[20] Joel H Ferziger and Milovan Perić. *Computational methods for fluid dynamics*, volume 3. Springer Berlin, 2002.

[21] Martin J Gander. Optimized schwarz methods. *SIAM Journal on Numerical Analysis*, 44(2):699–731, 2006.

[22] Martin J Gander and Felix Kwok. Best robin parameters for optimized schwarz methods at cross points. *SIAM Journal on Scientific Computing*, 34(4):A1849–A1879, 2012.

[23] Martin Jakob Gander, Laurence Halpern, Frédéric Nataf, et al. Optimal convergence for overlapping and non-overlapping schwarz waveform relaxation. 1999.

[24] MJ Gander and GH Golub. A non-overlapping optimized schwarz method which converges with arbitrarily weak dependence on h. 2002.

[25] H Glauert. Aerodynamic theory: A general review of progress, volume iv, chapter division l, airplane propellers, 1963.

[26] S Gomez-Iradi, R Steijl, and GN Barakos. Development and validation of a cfd technique for the aerodynamic analysis of hawt. *Journal of Solar Energy Engineering*, 131(3):031009, 2009.

[27] Jose Gonzalez, Joaquın Fernández, Eduardo Blanco, and Carlos Santolaria. Numerical simulation of the dynamic effects due to impeller-volute interaction in a centrifugal pump. *Journal of Fluids Engineering*, 124(2):348–355, 2002.

[28] Ouahiba Guerri, Anas Sakout, and Khedidja Bouhadef. Simulations of the fluid flow around a rotating vertical axis wind turbine. *Wind Engineering*, 31(3):149–163, 2007.

[29] Hackbusch. On the convergence of a multi-grid iteration applied to finite element equations. Technical report, Report 77-8, Universität zu Köln, 1977.

[30] Martin OL Hansen and Helge Aagaard Madsen. Review paper on wind turbine aerodynamics. *Journal of fluids engineering*, 133(11):114001, 2011.

[31] Martin Otto Laver Hansen, Jens Nørkær Sørensen, S Voutsinas, Niels Sørensen, and H Aa Madsen. State of the art in wind turbine aerodynamics and aeroelasticity. *Progress in aerospace sciences*, 42(4):285–330, 2006.

[32] William D Henshaw. A fourth-order accurate method for the incompressible navier-stokes equations on overlapping grids. *Journal of computational physics*, 113(1):13–25, 1994.

[33] William D Henshaw. On multigrid for overlapping grids. *SIAM Journal on Scientific Computing*, 26(5):1547–1572, 2005.

[34] BJ Hubbard and Hamn-Ching Chen. A chimera scheme for incompressible viscous flows with applications to submarine hydrodynamics. *AIAA paper*, (94-2210), 1994.

[35] Muhammad Ijaz and NK Anand. Co-located variables approach using implicit runge-kutta methods for unsteady incompressible flow simulation. *Numerical Heat Transfer, Part B: Fundamentals*, 54(4):291–313, 2008.

[36] Raad I Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational physics*, 62(1):40–65, 1986.

[37] Z Jaworski, ML Wyszynski, IPT Moore, and AW Nienow. Sliding mesh computational fluid dynamics-a predictive tool in stirred tank design. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, 211(3):149–156, 1997.

[38] Jeppe Johansen, NN Sorensen, JA Michelsen, and S Schreck. Detached-eddy simulation of flow around the nrel phase-vi blade. In *ASME 2002 Wind Energy Symposium*, pages 106–114. American Society of Mechanical Engineers, 2002.

[39] Peter Johansson and Lars Davidson. Modified collocated simplec algorithm applied to buoyancy-affected turbulent flow using a multigrid solution procedure. *Numerical Heat Transfer, Part B Fundamentals*, 28(1):39–57, 1995.

[40] Ramji Kamakoti and Wei Shyy. Fluid-structure interaction for aeroelastic applications. *Progress in Aerospace Sciences*, 40(8):535–558, 2004.

[41] Ramji Kamakoti, Wei Shyy, Siddharth Thakur, and Bhavani Sankar. Time dependent rans computation for an aeroelastic wing. *AIAA*, 886:2004, 2004.

[42] Yasuo Kawaguchi, Wen-Quan Tao, and Hiroyuki Ozoe. Checkerboard pressure predictions due to the underrelaxation factor and time step size for a nonstaggered grid with momentum interpolation method. *Numerical Heat Transfer: Part B: Fundamentals*, 41(1):85–94, 2002.

[43] V Kazemi Kamyab, AH Van Zuijlen, and H Bijl. Higher order implicit time integration schemes to solve incompressible navier-stokes on co-located grids using consistent unsteady rhie-chow. 2012.

[44] Yuwei Li, Kwang-Jun Paik, Tao Xing, and Pablo M Carrica. Dynamic overset cfd simulations of wind turbine aerodynamics. *Renewable Energy*, 37(1):285–298, 2012.

[45] Zhifeng Li, Dazhuan Wu, Leqin Wang, and Bin Huang. Numerical simulation of the transient flow in a centrifugal pump during starting period. *Journal of fluids engineering*, 132(8):081102, 2010.

[46] FS Lien and MA Leschziner. A general non-orthogonal collocated finite volume algorithm for turbulent flow at all speeds incorporating second-moment turbulence-transport closure, part 1: Computational implementation. *Computer methods in applied mechanics and engineering*, 114(1):123–148, 1994.

[47] Z Lilek, S Muzaferija, M Peric, and & Seidl S. An implicit finite-volume method using non-matching blocks on structured grid. *Num. Heat Transfer Part B*, 32:385–401, 1988.

[48] Željko Lilek, Samir Muzaferija, Milovan Perić, and Volker Seidl. Computation of unsteady flows using nonmatching blocks of structured grid. *Numerical Heat Transfer*, 32(4):403–418, 1997.

[49] Željko Lilek, Samir Muzaferija, Milovan Perić, and Volker Seidl. An implicit finite-volume method using nonmatching blocks of structured grid. *Numerical Heat Transfer*, 32(4):385–401, 1997.

[50] Pierre-Louis Lions. On the schwarz alternating method. iii: a variant for nonoverlapping subdomains. In *Third international symposium on domain decomposition methods for partial differential equations*, volume 6, pages 202–223. SIAM, Philadelphia, PA, 1990.

[51] S Majumdar. Role of underrelaxation in momentum interpolation for calculation of flow with nonstaggered grids. *Numerical Heat Transfer*, 13(1):125–132, 1988.

[52] Robert Meakin and R Gomez. On adaptive refinement and overset structured grids. In *13th AIAA Computational Fluid Dynamics Conf*, 1997.

[53] Robert L Meakin and Andrew M Wissink. Unsteady aerodynamic simulation of static and moving bodies using scalable computers. In *Proc. 14th AIAA Computational Fluid Dynamics Conference*, 1999.

[54] Jure Mencinger and Iztok Žun. On the finite volume discretization of discontinuous body force field on collocated grid: Application to vof method. *Journal of Computational Physics*, 221(2):524–538, 2007.

[55] J. A. Michelsen. Basis3d - a platform for development of multiblock pde solvers. Technical report, AFM 92-05, Technical University of Denmark, Denmark, 1992.

[56] JA Michelsen. Mesh-adaptive solution of the navier-stokes equations. *MULTIGRID METHODS III*, 98:301–312, 1991.

[57] TF Miller and FW Schmidt. Use of a pressure-weighted interpolation method for the solution of the incompressible navier-stokes equations on a nonstaggered grid system. *Numerical Heat Transfer, Part A: Applications*, 14(2):213–233, 1988.

[58] Michael L Minion and David L Brown. Performance of under-resolved two-dimensional incompressible flow simulations, ii. *Journal of Computational Physics*, 138(2):734–765, 1997.

[59] Frédéric Nataf. Optimized schwarz methods. In *Domain Decomposition Methods in Science and Engineering XVIII*, pages 233–240. Springer, 2009.

[60] Frédéric Nataf, Francois Rogier, and Eric de Sturler. Optimal interface conditions for domain decomposition methods. *CMAP (Ecole Polytechnique)*, 1994.

[61] J Papageorgakopoulos, G Arampatzis, D Assimacopoulos, and NC Markatos. Enhancement of the momentum interpolation method on non-staggered grids. *International journal for numerical methods in fluids*, 33(1):1–22, 2000.

[62] A Le Pape and J Lecanu. 3d navier–stokes computations of a stall-regulated wind turbine. *Wind Energy*, 7(4):309–324, 2004.

[63] Antonio Pascau. Cell face velocity alternatives in a structured colocated grid for the unsteady navier–stokes equations. *International Journal for Numerical Methods in Fluids*, 65(7):812–833, 2011.

[64] S.V. Patanakar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere, New York, 1980.

[65] CM Rhie and WL Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA journal*, 21(11):1525–1532, 1983.

[66] John R Rice, P Tsompanopoulou, and E Vavalis. Interface relaxation methods for elliptic differential equations. 1997.

[67] JR Rice, P Tsompanopoulou, and E Vavalis. Fine tuning interface relaxation methods for elliptic differential equations. *Applied numerical mathematics*, 43(4):459–481, 2002.

[68] Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

[69] B Sanderse, SP Pijl, and B Koren. Review of computational fluid dynamics for wind turbine wake aerodynamics. *Wind Energy*, 14(7):799–819, 2011.

[70] H.A. Schwarz. Uber einige abbildungsaufgaben. mathematik. *Journal reine angew*, 70:105 – 120, 1869.

[71] V Seidl, M Peric, and S Schmidt. Space-and time-parallel navier-stokes solver for 3d block-adaptive cartesian grids. In *Parallel Computational Fluid Dynamics: Proc*, volume 95, pages 557–584, 1995.

[72] Wen Zhong Shen, Jess A Michelsen, Niels N. Sørensen, and Jens Nørkær Sørensen. Improved rhie-chow interpolation for unsteady flow computations. *AIAA journal*, 39(12):2406–2409, 2001.

[73] Wen Zhong Shen, Jess A Michelsen, Niels N Sørensen, and Jens Nørkær Sørensen. An improved simplec method on collocated grids for steady and unsteady flow computations. *Numerical Heat Transfer: Part B: Fundamentals*, 43(3):221–239, 2003.

[74] H Snel. Review of the present status of rotor aerodynamics. *Wind Energy*, 1(s 1): 46–69, 1998.

[75] Herman Snel. Review of aerodynamics for wind turbines. *Wind Energy*, 6(3):203–211, 2003.

[76] P Sonneveld, P Wesseling, and PM De Zeeuw. Multigrid and conjugate gradient methods as convergence acceleration techniques. In *Multigrid methods for integral and differential equations*, volume 3, pages 117–168. Oxford UP, Oxford, 1985.

[77] Niels N Sørensen. *General purpose flow solver applied to flow over hills*. PhD thesis, Technical University of DenmarkDanmarks Tekniske Universitet, Risø National Laboratory for Sustainable EnergyRisø Nationallaboratoriet for Bæredygtig Energi, Wind Energy DivisionAfdelingen for Vindenergi, Aeroelastic DesignAeroelastisk Design, 1995.

[78] Niels N Sørensen. Cfd modelling of laminar-turbulent transition for airfoils and rotors using the γ- model. *Wind Energy*, 12(8):715–733, 2009.

[79] Niels N Sørensen, JA Michelsen, and S Schreck. Navier–stokes predictions of the nrel phase vi rotor in the nasa ames 80 ft× 120 ft wind tunnel. *Wind Energy*, 5(2-3): 151–169, 2002.

[80] Niels N Sørensen, JA Michelsen, and S Schreck. Navier–stokes predictions of the nrel phase vi rotor in the nasa ames 80 ft× 120 ft wind tunnel. *Wind Energy*, 5(2-3): 151–169, 2002.

[81] NN Sørensen and JA Michelsen. Aerodynamic predictions for the unsteady aerodynamics experiment phase-ii rotor at the national renewable energy laboratory. *AIAA paper*, 37, 2000.

[82] Joseph L Steger, F Carroll Dougherty, and John A Benek. A chimera grid scheme. 1983.

[83] R Steijl and G Barakos. Sliding mesh algorithm for cfd analysis of helicopter rotor–fuselage aerodynamics. *International journal for numerical methods in fluids*, 58(5):527–549, 2008.

[84] Wei Pai Tang. Generalized schwarz splittings. *SIAM Journal on Scientific and Statistical Computing*, 13(2):573–595, 1992.

[85] G Usera, A Vernet, and JA Ferré. A parallel block-structured finite volume method for flows in complex geometry with sliding interfaces. *Flow, Turbulence and Combustion*, 81(3):471–495, 2008.

[86] JP Van Doormaal and GD Raithby. Enhancements of the simple method for predicting incompressible fluid flows. *Numerical heat transfer*, 7(2):147–163, 1984.

[87] LJ Vermeer, Jens Nørkær Sørensen, and A Crespo. Wind turbine wake aerodynamics. *Progress in aerospace sciences*, 39(6):467–510, 2003.

[88] K Wechsler, M Breuer, and F Durst. Steady and unsteady computations of turbulent flows induced by a 4/45 pitched-blade impeller. *Journal of fluids engineering*, 121(2): 318–329, 1999.

[89] P. Wesseling. Multigrid methods in computational fluid dynamics. *Zeitschrift Fur Angewandte Mathematik Und Mechanik*, 70(5):T337–47, 1990. ISSN 00442267, 15214001.

[90] Pieter Wesseling. *Principles of computational fluid dynamics*, volume 29. Springer, 2009.

[91] Pieter Wesseling and Cornelis W Oosterlee. Geometric multigrid with applications to computational fluid dynamics. *Journal of Computational and Applied Mathematics*, 128(1):311–334, 2001.

[92] Guanpeng Xu and Lakshmi N Sankar. Development of engineering aerodynamics models using a viscous flow methodology on the nrel phase vi rotor. *Wind Energy*, 5(2-3):171–183, 2002.

[93] Ji Yao, Jianliang Wang, Weibin Yuan, Huimin Wang, and Liang Cao. Analysis on the influence of turbulence model changes to aerodynamic performance of vertical axis wind turbine. *Procedia Engineering*, 31:274–281, 2012.

[94] Bo Yu, Wen-Quan Tao, Jin-Jia Wei, Yasuo Kawaguchi, Toshio Tagawa, and Hiroyuki Ozoe. Discussion on momentum interpolation method for collocated grids of incompressible flow. *Numerical Heat Transfer: Part B: Fundamentals*, 42(2):141–166, 2002.

[95] Frederik Zahle. *Wind turbine aerodynamics using an incompressible overset grid method*. PhD thesis, Technical University of DenmarkDanmarks Tekniske Universitet, Risø National Laboratory for Sustainable EnergyRisø Nationallaboratoriet for Bæredygtig Energi, 2007.

[96] Frederik Zahle, Niels N Sørensen, and Jeppe Johansen. Wind turbine rotor-tower interaction using an incompressible overset grid method. *Wind Energy*, 12(6):594–619, 2009.

[97] Y Zang and RL Street. A composite multigrid method for calculating unsteady incompressible flows in geometrically complex domains. *International journal for numerical methods in fluids*, 20(5):341–361, 1995.

DTU Vindenergi er et institut under Danmarks Tekniske Universitet med en unik integration af forskning, uddannelse, innovation og offentlige/private konsulentopgaver inden for vindenergi. Vores aktiviteter bidrager til nye muligheder og teknologier inden for udnyttelse af vindenergi, både globalt og nationalt. Forskningen har fokus på specifikke tekniske og videnskabelige områder, der er centrale for udvikling, innovation og brug af vindenergi, og som danner grundlaget for højt kvalificerede uddannelser på universitetet.

Vi har mere end 240 ansatte og heraf er ca. 60 ph.d. studerende. Forskningen tager udgangspunkt i ni forskningsprogrammer, der er organiseret i tre hovedgrupper: vindenergisystemer, vindmølleteknologi og grundlagsskabende forskning.

Technical University of Denmark
Department of Wind Energy
Nils Koppels Allé
Building 403
2800 Kgs. Lyngby
Denmark
Phone 46 77 50 85

info@vindenergi.dtu.dk
www.vindenergi.dtu.dk